

TRABAJO FIN DE MASTER

METODOLOGIA DE MODELADO DE ROBOTS HIPERREDUNDANTES ACTUADOS POR CABLES

TRABAJO FIN DE MASTER
PARA LA OBTENCIÓN DEL
TÍTULO DE **MASTER EN**
AUTOMÁTICA Y ROBÓTICA

JUNIO 2022

Diego Cerrillo Vacas

DIRECTOR DEL TRABAJO FIN DE MASTER:
Antonio Barrientos Cruz

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES



POLITÉCNICA

TRABAJO FIN DE MÁSTER

para la obtención del título de Máster en Automática y
Robótica

**Metodología de Modelado de Robots Hiperredundantes
Actuados por Cables**



PRESENTADO POR
Diego Cerrillo Vacas

Tutor
Antonio Barrientos Cruz

Madrid, 2022

AGRADECIMIENTOS

A la hora de cerrar mi paso por la Escuela Técnica Superior de Ingenieros Industriales, me entra una cierta nostalgia. Han sido seis años en los que he crecido como persona, me he formado como ingeniero, y he tenido la oportunidad de conocer a mucha gente de mucho valor. Se da la coincidencia de que, teniendo dos TFM que entregar, redacto dos mensajes de agradecimiento... Imagino que es una señal de que hay mucho que agradecer.

El primer agradecimiento, sin duda alguna, es a Dios, que me ha puesto en este camino. Que en su momento me inspiró el deseo de hacerme ingeniero, y en concreto industrial. Que hace dos años suscitó en mí el deseo de hacer un doble máster. Y que hoy me sigue llamando a hacer de mí un buen cristiano, una buena persona... y un buen ingeniero.

En este trabajo tengo que agradecer, sin duda alguna, a Antonio Barrientos. Antonio ha sido para mí un verdadero Maestro. Trabajar contigo ha sido un auténtico lujo. Tu entrega por tus alumnos, tu entusiasmo por la robótica y por la docencia, tu saber hacer como investigador y como profesor... Todo eso ha sido testimonio para mí estos últimos años. Gracias por todas las oportunidades recibidas para trabajar contigo, y con el grupo de investigación. Gracias por el acompañamiento, y por la oportunidad de descubrir bajo tu tutoría qué significa investigar. Ojalá a lo largo de la vida pueda devolver una parte de lo que siento haber recibido estos años.

Gracias también a los otros miembros del grupo de investigación. Jaime, Chrystian, David, Jorge, Celia... Han sido momentos duros, especialmente estos últimos días, pero sin duda alguna, siempre sabía que podía contar con vuestra ayuda. Sois gente maravillosa, de gran valía, y estoy convencido de que seguiréis haciendo grandes cosas en el departamento.

Gracias a mi familia y a mi comunidad, que me ha soportado incluso en los momentos de gran oscuridad con mucha paciencia y con mucho cariño. Es un lujo saber que, independientemente de los obstáculos a los que te enfrentas siempre puedes volver con ellos y saber que estás en casa.

Gracias a los amigos que con tesón se preocupan por el estado de mis robots y mis trabajos. Poder salir y tomar una cerveza o cantar en un karaoke es siempre un motivo de descanso y de disfrutar. Muchas gracias por estar siempre ahí.

Por último, gracias a la ETSII. Gracias por estos seis años de formación, de crecimiento. Gracias a todas las personas que forman parte de ella y se esfuerzan sinceramente para hacerla un lugar mejor día tras día. Gracias a todos los profesores que sacáis rato para investigar, enseñar, atender a los alumnos, acompañar... La ETSII ya forma parte de mí.

RESUMEN

Cuando un robot tiene un número de actuadores independientes notablemente superior a los seis grados de libertad necesarios para definir la posición y orientación de un sólido rígido en el espacio, se denomina **robot hiperredundante**. Este conjunto de robots, cuya importancia en el mundo académico ha crecido en las últimas décadas, es mucho más complejo de modelar que un robot clásico. Las metodologías tradicionales no son suficientes para plantear las ecuaciones matemáticas de los robots hiperredundantes.

Por ello, en este trabajo se realiza una revisión general del modelado cinemático de robots hiperredundantes actuados por cables (o RHACs). El número de publicaciones disponible y las discrepancias entre la nomenclatura empleada por los distintos autores hace difícil para un investigador que se introduce en el tema saber cómo abordar en un principio este paso.

El objetivo principal de este Trabajo Fin de Máster conducente a los estudios del Máster en Automática y Robótica busca sistematizar los distintos enfoques e hipótesis que se usan actualmente para el modelado de RHACs y proponer una metodología que, contemplando los modelos más frecuentes en la bibliografía actual, proponga cuál de ellos podría ser más apropiado para la morfología particular del robot que se desee modelar.

Para ello, el trabajo comienza revisando algunos **conceptos** relativos a la robótica hiperredundante, definiéndolos a partir del criterio de Martín-Barrio [23], y proponiendo algunos ejemplos de robots que pertenezcan a esta categoría, destacando entre ellos a aquellos robots pertenecientes al Centro de Automática y Robótica.

Posteriormente, en el capítulo de métodos y herramientas se propone un **marco teórico** para comprender la estructura general de un modelo hiperredundante: en la bibliografía los enfoques son muchas veces distintos, y puede ser confuso comprender qué paso se está tratando de resolver si no se dispone de un marco teórico adecuado. En el capítulo se pretende unificar la nomenclatura a seguir a lo largo de todo el trabajo, además de ofrecer un esquema para la cinemática directa y la cinemática inversa donde posteriormente se irán introduciendo los distintos modelos desarrollados.

El siguiente paso que se da en el trabajo es el desarrollo teórico de los modelos. En primer lugar se analizan los dos principales que dependen únicamente de **parámetros cinemáticos**, sin introducir en ningún momento las causas del movimiento: el modelado a partir de **Denavit-Hartenberg** y el modelado según la **hipótesis de curvatura constante** o PCC. Ambos dos son de gran importancia, y constituirán la base para modelar dos grandes grupos de RHACs. Asimismo, se recoge una versión optimizada del PCC que reduce su carga computacional, propuesta recientemente por Barrientos Díez, Dong et Al. [4]

No obstante, estos no son los únicos modelos disponibles en la bibliografía. Por ello, el siguiente capítulo recoge los tres modelos que emplean **parámetros elásticos** más utilizados en el mundo académico para resolver la cinemática de RHACs. Para su uso se requiere de herramientas de cálculo más poderosas (como solvers de ecuaciones no lineales o programas de simulación por elementos finitos), pero su empleo ayuda a obtener modelos más precisos.

A continuación, se revisa un algoritmo de optimización numérica de gran utilidad para resolver una parte de la cinemática inversa de robots hiperredundante: el **Natural CCD**. Su eficiencia y bajo coste computacional lo hace un gran competidor frente a alternativas de inteligencia artificial como redes neuronales o algoritmos genéticos. El algoritmo también permite simplificar algunas tareas de control del robot y realizar algunos movimientos y trayectorias de forma sencilla, a través de ligeras modificaciones del algoritmo denominadas behaviours.

Habiendo realizado una panorámica del estado de la cuestión del modelado de robots hiperredundantes, se procede a proponer una **metodología** para identificar los modelos más apropiados para cada morfología de robot. Se basa en una serie de cuestiones a plantearse, que se deben identificar para el robot, y con las cuales se puede obtener un modelo ajustado a las necesidades concretas. La metodología se aplica posteriormente a **tres ejemplos** de robots del CAR.

Con el fin de no dejar el trabajo en un plano exclusivamente teórico, se ha **validado** un modelo cinemático obtenido empleando la metodología a uno de los robots del CAR: el MACH-I. Se presentan en el capítulo 8 los resultados de esta validación, verificando así que se trata de un modelo apropiado.

La memoria se concluye con la presentación de las conclusiones del trabajo, su impacto social y las posibles líneas de trabajo que se podrían seguir en un futuro. Finalmente, se incluyen también los datos relativos a la organización del trabajo: los presupuestos, la EDP y el diagrama de Gantt del TFM.

Códigos UNESCO

- [330419] Robótica
- [330412] Dispositivos de control
- [120326] Simulación
- [120304] Inteligencia Artificial

Palabras Clave

Robótica hiperredundante actuada por cables, modelos cinemáticos, Denavit-Hartenberg, modelos PCC, metodología, validación, simulación, redes neuronales

ÍNDICE GENERAL

Agradecimientos	I
Resumen	III
Índice General	v
1. Introducción	7
1.1. Motivación	7
1.2. Objetivos del Trabajo	7
1.3. Estructura de la memoria	8
2. Definiciones básicas sobre Robótica Hiperredundante	9
2.1. Definición de robot hiperredundante	9
2.2. Definiciones de robots hiperredundantes actuados por cables	11
3. Métodos y herramientas para el modelado de robots hiperredundantes	15
3.1. Nomenclatura y definiciones básicas para el modelado de un RHAC	15
3.2. Etapas para el modelado cinemático de un RHAC: Cinemática Directa	17
3.3. Etapas para el modelado cinemático de un RHAC: Cinemática Inversa	18
4. Modelos cinemáticos para RHAC	21
4.1. Hipótesis discreta: Método de Denavit-Hartenberg	21
4.1.1. Aplicación de la hipótesis	21
4.1.2. Explicación del método	22
4.2. Modelo PCC - Piecewise Constant Curvature	25
4.2.1. Cinemática directa	25
4.2.2. Cinemática Inversa	34
4.2.3. Cinemática Diferencial	38
4.3. LSK - Cinemática de Segmentos Linealizados	41
5. Modelos cinemáticos para RHAC basados en parámetros de elasticidad	45
5.1. PRB: Pseudo-Rigid Body	46
5.2. Teoría elástica de Cosserat - CRT (Cosserat Rod Theory)	49
5.3. Método de Elementos Finitos (FEM)	52
6. Algoritmo de optimización: Natural - CCD	55
6.1. Planteamiento del problema	55
6.2. Desarrollo del algoritmo	59
6.2.1. Algoritmo CCD	59
6.2.2. Modificación: Natural-CCD	60
6.3. Otros usos del Natural-CCD: Behaviors	61
7. Propuesta de guía para el modelado de RHAC. Ejemplos de aplicación	65
7.1. Presentación de la guía - tutorial	65
7.1.1. Determinación del tipo de robot	66
7.1.2. Hipótesis de curvatura constante	68

7.1.3. Conocimiento sobre las fuerzas: ¿son continuas o discretas?	68
7.1.4. Las velocidades como requisito del modelo	69
7.1.5. Otros criterios: ecuaciones para la longitud de los cables	70
7.2. Ejemplificación sobre robots reales	71
7.2.1. MACH-I: Ejemplo de un robot discreto	71
7.2.2. Ejemplo de robot no PCC: Ruan	76
7.2.3. Ejemplo de robot PCC: Pilory	79
8. Validación de un modelo cinemático sobre robot real: MACH-I	83
8.1. Antecedentes	83
8.2. Arquitectura desarrollada para la validación	86
8.3. Simulador para validación de la cinemática inversa	88
8.4. Desarrollo de la cinemática directa	91
8.5. Pruebas sobre el robot real	96
9. Conclusiones	97
9.1. Conclusiones y aportaciones del trabajo	97
9.2. Impacto social y medioambiental del trabajo	97
9.3. Líneas futuras de trabajo	98
10. Planificación del trabajo	99
10.1. Presupuesto del trabajo	99
10.2. Estructura de Descomposición del Proyecto (EDP)	100
10.3. Planificación temporal del trabajo	101
Bibliografía	103
Indice de Figuras	107
Glosario	110

1 | INTRODUCCIÓN

1.1. Motivación

El modelado de los robots manipuladores tradicionales es de sobra conocido por el mundo de la robótica. El modelado mediante el método de Denavit-Hartenberg, en referencia a quienes asentaron las bases de este método matricial, se sigue estudiando en las escuelas de ingeniería y robótica por su capacidad de sistematizar el modelado de robots seriales.

No obstante, el desarrollo de la robótica ha permitido la creación de robots con un número de actuadores independientes muy superior al mínimo necesario para posicionar y orientar un sólido rígido en el espacio: esta categoría de robots se denominan robots hiperredundantes. Su modelado es mucho más complejo que en el caso de los manipuladores tradicionales: su morfología es muy variable para poder alcanzar este elevado número de grados de libertad.

Ello hace difícil poder generalizar el método de Denavit-Hartenberg a cualquier robot hiperredundante: las hipótesis de las que parte y el modelado como una sucesión serie de barras-articulación no es en general aplicable a este tipo de robots, obligando a los diseñadores a abordar un modelado cinemático inverso y directo particular, atendiendo a las características del mismo.

Ante esta necesidad de dar respuesta a con qué ecuaciones modelar los robots hiperredundantes, el mundo académico ha ido generando distintas soluciones, partiendo de casuísticas particulares, y generando una variedad importante de modelos entre los cuales puede ser difícil identificar el más adecuado para un robot.

En este marco de multiplicidad de modelos, este Trabajo Fin de Máster pretende revisar los distintos modelos que se han ido publicando, identificando sus características y sistematizando una metodología que permita facilitar la elección de un modelo adecuado para ayudar a quien se inicia en el estudio de los robots hiperredundantes.

1.2. Objetivos del Trabajo

Como objetivo principal de este trabajo se ha definido la elaboración de una propuesta de sistematización para el modelado cinemático de robots hiperredundantes actuados por cables, con el resultado final de una guía que introduzca al investigador novel a este campo.

Para poder lograr este objetivo general, se han definido los siguientes objetivos secundarios:

- Analizar la bibliografía existente sobre el modelado cinemático de robots hiperredundantes, recabando información sobre los modelos que se han ido desarrollando y su campo de aplicación.
- Establecer un marco teórico que permita comprender las distintas etapas del modelado cinemático de robots hiperredundantes.
- Presentar un desarrollo sistemático de los distintos modelos, combinando la información de distintas fuentes para dar lugar a una guía única.

- Interpretar dichos modelos desde el marco teórico propuesto.
- Elaborar una guía-resumen que permita a un nuevo investigador, determinar rápidamente qué tipo de modelo es más recomendable para un tipo de robot concreto.
- Aplicar dicha guía resumen a tres robots hiperredundantes que se encuentran en el CAR, para ejemplificar su uso.
- Validar al menos uno de los modelos propuestos para dichos robots

1.3. Estructura de la memoria

La memoria sigue aproximadamente el desarrollo de estos objetivos de forma sistemática. Se comienza presentando un breve estado de la técnica, que desarrolla algunos conceptos relevantes para el resto del trabajo y presenta algunos ejemplos de robots hiperredundantes, clasificándolos en distintas categorías.

Posteriormente, en el capítulo de métodos y herramientas, se presenta el marco teórico fundamental que se empleará en el resto del trabajo. Así, se presentan las distintas etapas del modelado cinemático, tanto directo, como inverso, además de introducir la nomenclatura básica que se seguirá en el resto del trabajo.

A continuación, comienza el desarrollo de los distintos modelos disponibles en la bibliografía. Estos se han separado en dos grandes grupos: los modelos cinemáticos que usan exclusivamente variables no dinámicas, y aquellos que sí introducen los conceptos de fuerza para poder modelar con precisión el robot. Para cada uno de los modelos se introduce el desarrollo de las deducciones más importantes, y se presentan las ecuaciones finales de los mismo. El siguiente capítulo revisa el algoritmo de Natural-CCD, propuesto por Martín-Barrio [22], como recomendación para una de las etapas del modelo cinemático inverso.

Con ello, se termina la presentación de todos los modelos y las herramientas. Se dispone ya de todos los materiales para proponer una guía-tutorial, basada en un esquema gráfico. En el capítulo 7 se desarrolla este concepto, y se aplica su uso a tres robots hiperredundantes disponibles en el CAR. De esta forma se ofrece una mejor comprensión de los planteamientos de la guía.

Finalmente, se propone la validación de uno de los modelos, obtenido según la metodología anterior, sobre el robot real. Para ello, se presenta el conjunto de desarrollos realizados para programar el nuevo modelo cinemático para el robot MACH-I. La validación del nuevo modelo se logra a partir de la comparación con un modelo existente y plenamente validado, ya que es el conjunto de ecuaciones que se usa actualmente para el control del robot.

La memoria se concluye con un breve análisis de las conclusiones más importantes del trabajo y las posibles líneas futuras derivadas del mismo, así como con la presentación del cronograma y los presupuestos del trabajo.

2 | DEFINICIONES BÁSICAS SOBRE ROBÓTICA HIPERREDUNDANTE

2.1. Definición de robot hiperredundante

En este trabajo se va a tratar el modelado cinemático de robots hiperredundantes. No obstante, para ello es necesario definir primero con claridad qué se entiende por *robot hiperredundante*. A pesar de que no existe una única definición clara, aceptada genéricamente por la bibliografía, en este TFM se adoptará el criterio propuesto por Andrés Martín et al. [23], por ser una definición objetiva que permite realizar una discriminación acertada. En el caso de otras clasificaciones, no se ofrecen criterios numéricos para su diferenciación. [8] En este criterio se distinguen tres grandes tipos de robots, según los grados de libertad: robots discretos, robots redundantes y robots hiperredundantes.

Así, en primer lugar, es necesario plantear el número mínimo de grados de libertad (GDL) que son necesarios a la hora de determinar perfectamente el estado de un sólido rígido (esto es, incluyendo la posición y la orientación) en un espacio. Por generalidad, escogiendo el espacio \mathbb{R}^m , este número de GDL se calcula en la ecuación 2.1:

$$GDL_{\mathbb{R}^m} = \frac{m(m+1)}{2} \quad (2.1)$$

No obstante, en la robótica se trabaja esencialmente bien en el plano (es decir, el espacio bidimensional \mathbb{R}^2) o en el espacio tridimensional (\mathbb{R}^3). Para estos casos, y particularizando:

- **Caso 2D:** \mathbb{R}^2 , $m = 2 \implies GDL_{\mathbb{R}^2} = 3$
- **Caso 3D:** \mathbb{R}^3 , $m = 3 \implies GDL_{\mathbb{R}^3} = 6$

A partir de estos valores, ya se puede proponer una clasificación en función de los grados de libertad que el robot tenga:

- **Robots no redundantes:** se denominan así los robots que cuyo número de grados de libertad independientes en el espacio \mathbb{R}^m es menor o igual que el número $GDL_{\mathbb{R}^m}$. De esta forma, se encuentra por debajo, o justo en el número de GDL que se necesitan para posicionar perfectamente el robot, y podrá existir alguna coordenada que quede fijada a partir de las demás. En concreto, si el robot dispone de n GDL, tendrá $m - n$ grados dependientes.
- **Robots redundantes:** se refiere a aquellos robots cuyo número de grados de libertad independientes se encuentra por encima de $GDL_{\mathbb{R}^m}$. Tienen la capacidad de alcanzar un estado cualquiera empleando distintas combinaciones de valores, de forma que existe multiplicidad en las soluciones posibles.
- **Robots hiperredundantes:** son aquellos robots que disponen el doble, o más grados de libertad de los mínimos necesarios $GDL \geq 2 * GDL_{\mathbb{R}^m}$.

Los robots hiperredundantes tienen múltiples ventajas. La multiplicidad en las soluciones permite que el robot se configure en el espacio de trabajo con facilidad, pudiendo evitar posibles

obstáculos que se encuentren en el espacio, así como permitiéndoles realizar determinadas tareas de una cierta complejidad, como puede ser el agarre por envolvimiento del objeto o la inspección de espacios confinados.

Además, esta redundancia en los grados de libertad actuados permite una mayor robustez en la ejecución de las tareas. En el caso de una avería o de mal funcionamiento, otros GDL pueden suplir dicha falta y aún así realizar la tarea.

En el trabajo de Andrés Martín se incluyen también otras clasificaciones [23], de las cuales se incorporan aquí las más relevantes para este TFM:

1. **Robot continuo / discreto**: en función de si el robot está compuesto por un número finito de segmentos rígidos distinguibles, o si se trata de una estructura deformada de forma continua.
2. **Robots infraactuados / no infraactuados**: los robots infraactuados son aquellos en los que existe un mayor número de grados de libertad que de actuadores, de tal forma que cada actuador mueve más de un GDL. Por ejemplo, los robots continuos, por tener infinitos grados de libertad teóricos, siempre van a ser robots infraactuados.
3. **Robots intrínsecos / extrínsecos**: Los robots intrínsecos son aquellos que tienen sus actuadores situados en el propio mecanismo, mientras que los extrínsecos requieren de elementos de transmisión para poder llevar el movimiento al mecanismo.

En la figura 2.1 se encuentra un gráfico resumen publicado en el trabajo de Andrés Martín con dichas clasificaciones, así como algunos ejemplos de robots clasificados por código de colores. [23]

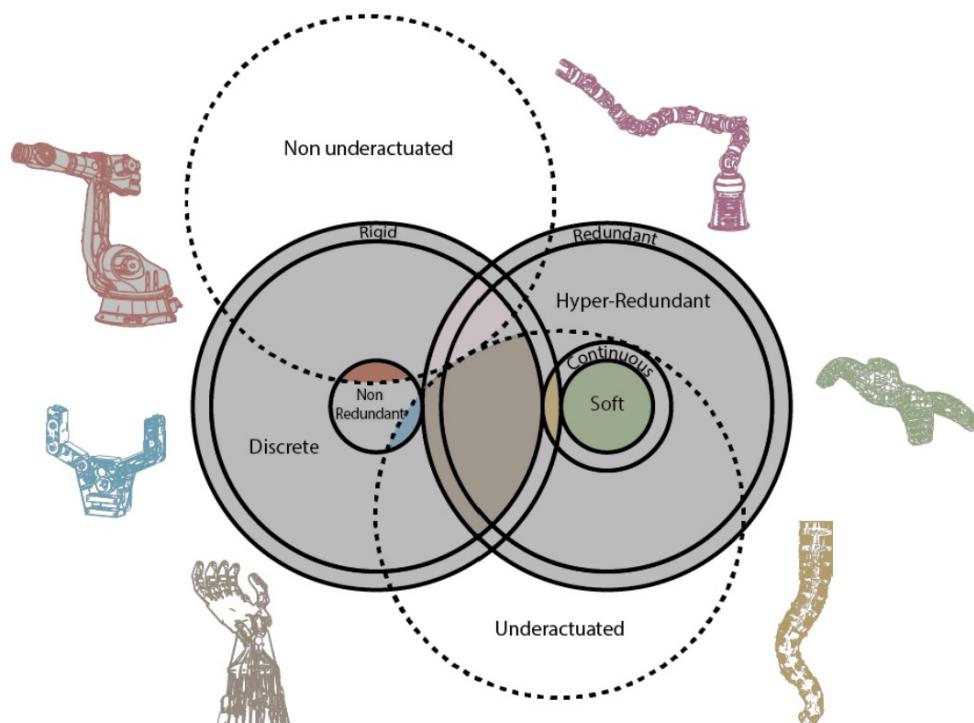


Figura 2.1: Gráfico resumen de clasificación de robots hiperredundantes. [23]

2.2. Definiciones de robots hiperredundantes actuados por cables

Como se ha podido observar en el apartado anterior, la robótica hiperredundante es un campo amplio, que incorpora muchos tipos de robots distintos. Por ello, para posibilitar una investigación sobre el modelado cinemático de un tipo concreto, se ha optado por el conjunto de los **Robots Hiperredundantes Actuados por Cables** (en adelante, **RHAC**).

Los RHAC son un conjunto particular de robots hiperredundantes, generalmente con morfología de brazos robóticos, que están compuestos de una serie de secciones, normalmente delimitadas por discos rígidos, a los que se fijan una serie de cables. A través de variaciones de longitud de los cables, obtenidas por diversos medios (como motores, o en el caso de filamentos SMA, por cambios de temperatura), se consiguen desplazar las secciones para alcanzar la configuración deseada.

En este caso, la hiperredundancia se consigue a partir de la composición de múltiples secciones en serie. Los robots que actúan los cables a través de motores suelen ser, además, robots extrínsecos. De esta forma, se desplaza el peso del motor a la base del robot, actuando el propio cable como correa de transmisión para el movimiento. Indudablemente, eso hace depender la precisión del movimiento de varias fuentes de error (como la tensión del cable, o la forma precisa que adopta), pero permite simplificar la estructura.

Para poder distinguir entre los tipos de RHAC se puede introducir el concepto de **núcleo** o **backbone**, según la grafía inglesa. El núcleo de un RHAC es la estructura interna que soporta el peso propio del robot, así como servir como base para unir los distintos discos que delimitan las secciones del robot. Teniendo este concepto en cuenta, y recuperando algunas clasificaciones del apartado anterior, pueden definirse tres grandes grupos de RHAC, que a su vez servirán para simplificar el proceso de modelado de estos robots. Sus características principales incluyen:

1. **RHAC Discretos**: incluye todos los robots actuados por cables que pertenecen al grupo clásico de los robots hiperredundantes discretos. Se trata del grupo que, históricamente, ha tenido más peso en las orígenes robótica hiperredundante. Su característica principal es la rigidez de las secciones. Así, el robot se puede definir como una sucesión de segmentos completamente rígidos unidos por articulaciones de, generalmente, uno o dos grados de libertad.

Esta rigidez permite la aplicación de técnicas clásicas de modelado en la obtención de su modelo matemático. Dos ejemplos de robots hiperredundantes discretos de diseño más reciente son el MACH-I [23], perteneciente al CAR (ver imagen 2.2a), o el Series II, X125 System, robot comercial fabricado por OC Robotics. [36] (Ver imagen 2.2b).



(a) MACH-I RHAC discreto del C.A.R.
Imagen propia



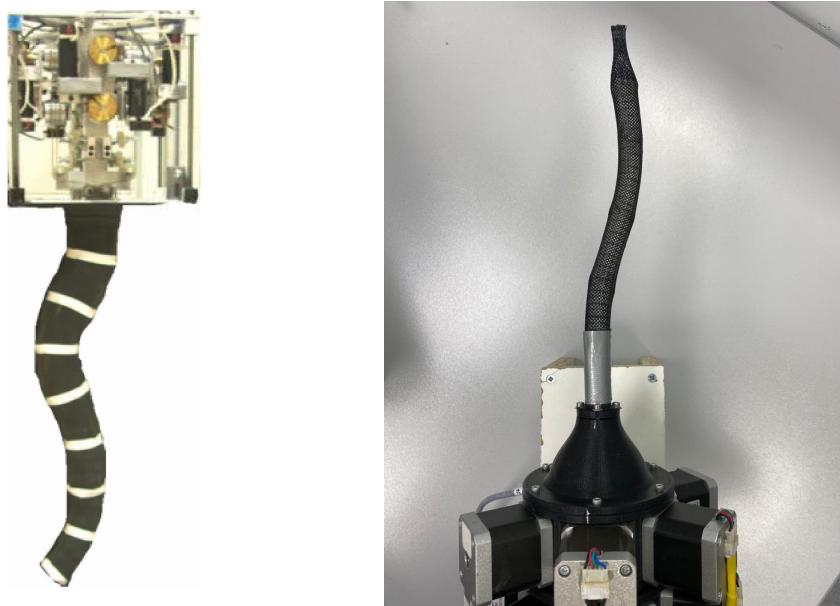
(b) Series II, X125 System: un ejemplo de RHAC discreto comercial. [31]

Figura 2.2: Ejemplos de robots discretos

2. **RHAC continuos no blandos:** Dentro de los robots actuados con cables, pueden encontrarse también un subconjunto de robots que constituyan una curva continua. Incluirá, por tanto, y siguiendo las definiciones de la sección anterior, todos aquellos robots que no dispongan de secciones rígidas distinguibles. El modelado cinemático de este tipo de robots conlleva una mayor dificultad, por motivo de esta continuidad. Se excluyen de este grupo los robots blandos

Es también un grupo de robots con muchos ejemplos, y de una morfología variable: existen algunos con un núcleo flexible que es el que aporta la curvatura continua; otros pueden emplear un número suficiente de discos pasivos por sección como para poder suponer su continuidad; también existen aquellos que emplean discos pivotantes uno sobre otro para generar esa sensación de continuidad... Como se puede observar, la continuidad absoluta es difícil de conseguir, si bien con un número suficiente de GDL tiende a considerarse como tal.

Posibles ejemplos para esta clase de robots se pueden encontrar en la trompa de elefante de Hannan y Walker ([14], ver figura 2.3a) o, en el caso del CAR, en el robot Pylori-I, basado en discos pivotantes (ver figura 2.3b). [29]



(a) RHAC continuo: trompa de elefante. (b) Pilory: un ejemplo de RHAC continuo del CAR. Imagen Propia [14]

Figura 2.3: Ejemplos de robots continuos no blandos

3. **RHAC blandos:** El último tipo de robots a considerar en esta clasificación propuesta es el de los robots blandos. Este tipo de robots se denominan como tal por emplear materiales considerados blandos. Precisamente definir qué se considera por material blando es otra de las discusiones presentes en la bibliografía relativa a robots blandos. En este trabajo se asumirá la conclusión de Majidi, que interpreta como material rígido aquel que supera un módulo de Young de $E_{rigido} \geq 1 \times 10^{10}$, y material blando como el que se encuentra entre $1 \times 10^2 \leq E_{blando} \leq 1 \times 10^6$. [21]

Los robots blandos presentan el caso más complejo para su modelado, por la gran cantidad de fuentes de incertidumbre que se manejan (como puede ser la caracterización precisa del material constitutivo). Generalmente requieren de métodos numéricos para resolver las

ecuaciones constitutivas, o de reducir la precisión a cambio de usar modelos algo más simples. Morfológicamente, muchos emplean mecanismos de movimiento similares a los otros robots continuos.

Ejemplos de robots blandos se pueden encontrar en dos exponentes del Centro de Automática y Robótica: Ruan [40], un robot blando creado por Silvia Terrile (ver figura 2.4a) o Kyma, [24] otro robot con una morfología diferente (ver figura 2.4b).

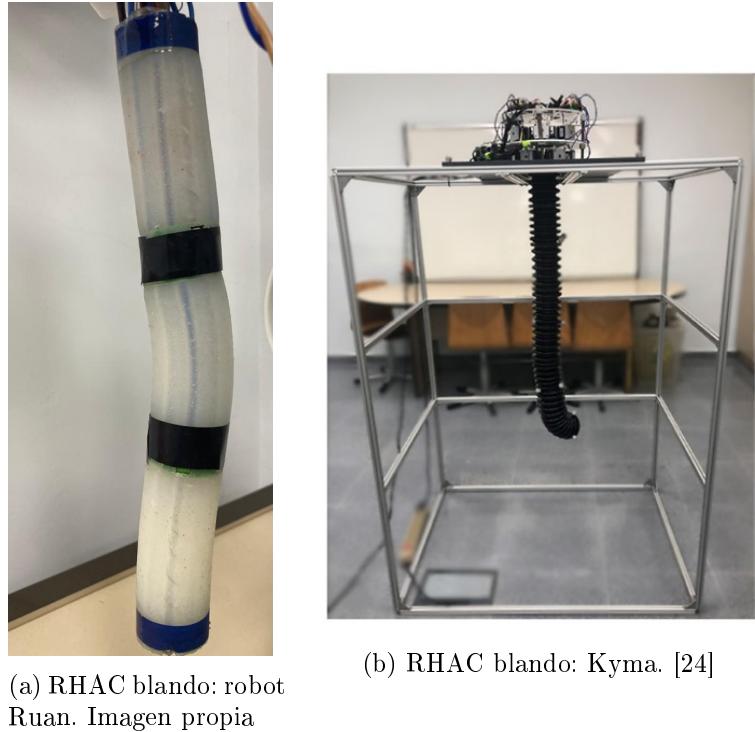


Figura 2.4: Ejemplos de robots continuos blandos

3 | MÉTODOS Y HERRAMIENTAS PARA EL MODELADO DE ROBOTS HIPERREDUNDANTES

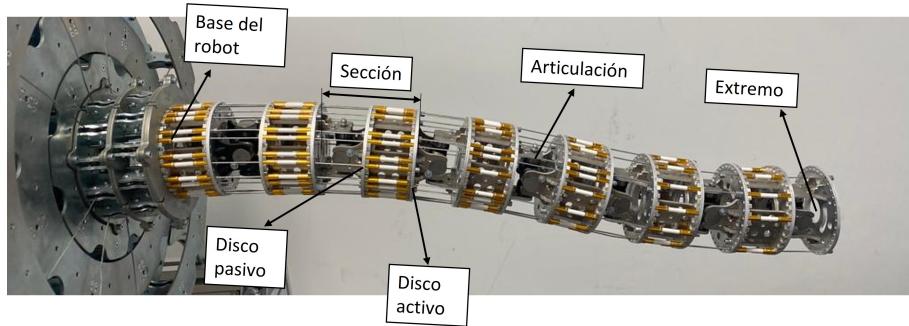
En este capítulo se van a analizar aquellos aspectos generales sobre el modelado de robots hiperredundantes que vertebrarán el resto de contenidos de este trabajo. En concreto, tras plantear la nomenclatura general que se seguirá a lo largo de este trabajo, se indicarán unos marcos teóricos que constituirán el corazón de los modelos analizados. En ese sentido, se distinguirá entre la cinemática directa y la inversa, como objetivos para la obtención del modelo completo.

3.1. Nomenclatura y definiciones básicas para el modelado de un RHAC

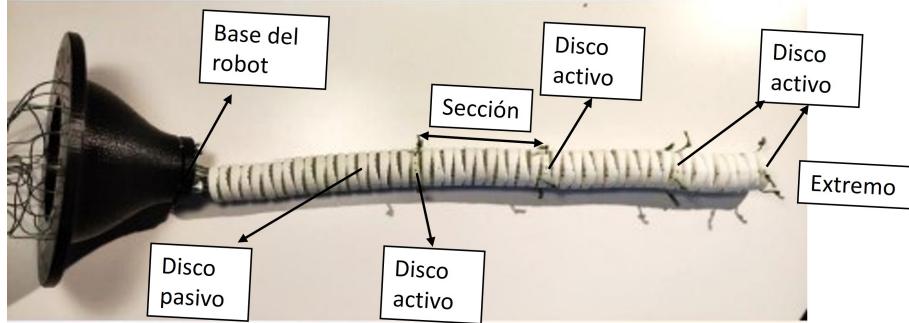
El modelado cinemático es el proceso mediante el cual se obtienen unas relaciones matemáticas, expresadas mediante ecuaciones, entre el estado del robot (referido a su posicionamiento y a su orientación) en cada momento t con aquellas variables correspondientes a la actuación del sistema, que en el caso de los RHAC, se asimila a las longitudes de los cables correspondientes. Es mediante estas variaciones de longitud con las que puede posicionarse el robot en la configuración deseada.

Con el fin de unificar los conceptos que van a ser manejados a lo largo de este TFM, es necesario poner el foco en una serie de definiciones sobre los distintos elementos que configuran a un robot hiperredundante. Nótese que en función de la morfología del robot, un RHAC puede no tener alguno de estos elementos:

1. **Discos activos del robot**: elementos rígidos con la forma de su nombre, insertados a lo largo del núcleo, en los cuales se fija un número determinado de cables sobre los que se puede actuar.
2. **Sección**: porción de robot ubicado entre dos discos activos, que modifica su forma conforme se actúa sobre las longitudes de los cables fijados al disco activo posterior de la misma.
3. **Discos pasivos**: elementos rígidos, insertados a lo largo del núcleo del robot, en los que, a diferencia de los discos activos, no hay ningún cable fijo en ellos que actúe sobre ellos. Su posición depende del núcleo o de las fuerzas ejercidas sobre los discos activos. Se emplean para hacer pasar a los cables por ellos, y dirigir su forma para asemejarla más a la deseada.
4. **Base del robot**: se define como base del robot, y como punto que constituirá el origen del sistema de coordenadas mundo para los modelos propuestos, al punto central de la sección del robot anclada a su estructura. Las secciones serán numeradas desde la base hacia el extremo.
5. **Extremo**: se entiende por extremo del robot de cara a todos los modelos propuestos en este trabajo el punto medio del disco activo posterior de la última sección.



(a) Definiciones sobre el robot MACH-I. Elaboración propia.



(b) Definiciones sobre el robot Pilory. [29] Los discos activos son aquellos donde están fijados los cables

Figura 3.1: Esquema sobre las definiciones geométricas en RHAC.

Con el fin de tener una referencia visual de cada uno de estos elementos, se inserta la figura 3.1. En ella, se puede observar aplicado a un ejemplo de robot real cada uno de estos elementos.

Una vez determinados los elementos geométricos mínimos que son necesarios para las definiciones de los modelos, es necesario también unificar la nomenclatura matemática que se empleará en todo este Trabajo Fin de Máster.

- Se denominará **configuración del robot** a la definición de la forma de toda la estructura del robot, esto es, a la posición de todos los puntos del mismo en el espacio. Conocida la forma del robot, esta determinación se puede realizar conocida la posición de todos los puntos del núcleo (o de un núcleo teórico en caso de carecer de él).
- La determinación de la posición de cada uno de los infinitos puntos de una sección en concreto, se denominará **estado de la sección**. No obstante, en función del tipo de modelo y de las hipótesis simplificativas que se usen, muchas veces será posible determinar perfectamente este conjunto de puntos mediante la utilización de un número finito de parámetros, generalmente representados por un vector.
- Para determinar el número total de secciones del robot, se empleará la letra n .
- La letra k se empleará como subíndice para referirse a una sección en particular.
- El número total de todos los cables del robot se nombrará con f .
- Por extrapolación, el número de cables que corresponden a una sección en particular se nombrará como f_k .
- Todos aquellos parámetros referidos al extremo del robot llevarán el subíndice e .

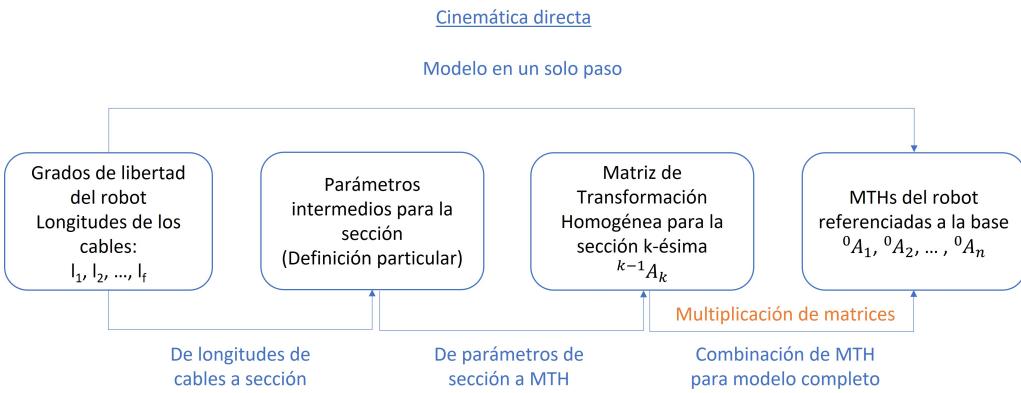


Figura 3.2: Resumen de las distintas etapas para el modelo cinemático directo. Elaboración propia

Empleando esta nomenclatura, a continuación se presentan los marcos teóricos que se van a emplear para el desarrollo de los distintos modelos aplicables a robots hiperredundantes.

3.2. Etapas para el modelado cinemático de un RHAC: Cinemática Directa

La cinemática directa en los RHAC es el problema que consiste en encontrar la relación existente entre los grados de libertad que tiene el robot (en concreto, las longitudes de los cables al hablar de cinemática directa) con la configuración del robot en cada momento. Sean $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$ los estados de cada una de las secciones de un RHAC, representados como un vector de parámetros. En este caso, el problema de cinemática directa se podría definir como encontrar las ecuaciones que relacionan l_1, l_2, \dots, l_n como dato para obtener $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$.

No obstante, aquí la hiperredundancia entra en juego. El hecho de poder encontrar de un solo paso estas ecuaciones, por ejemplo mediante una deducción puramente geométrica, es de una complejidad excesivamente elevada. [42] En general, se trata de abordar el problema empleando otros mecanismos, ya sean métodos numéricos para resolver las ecuaciones, aplicaciones de inteligencia artificial, o, como suele ser más común, mediante la aplicación de modelos simplificadores que permiten obtener un modelo mucho más manejable. [23]. En general, estas hipótesis simplificadoras permiten, además, dividir el modelo cinemático de una forma sistemática en varios pasos.

Además de dividir el problema, la simplificación generalmente implica una representación del estado de la sección a través de un conjunto finito (y generalmente, de un número reducido) de parámetros. Este es el caso, por ejemplo, de la hipótesis de curvatura constante: el hecho de suponer que la sección sigue una forma determinada ya es suficientemente relevante como para poder tener toda la información sobre el estado de la sección a partir de tres parámetros. Generalmente, estos valores luego se trasladan a una de las herramientas matemáticas que se emplean para representar las poses (posición y orientación) de las secciones. En este TFM se van a emplear las Matrices de Transformación Homogéneas (MTHs). Esto se debe a la simplicidad de su uso. Definiendo un sistema de referencia en el origen de la sección y otro al final de la sección, sumado a las hipótesis simplificadoras, se obtiene un modelo completo para la misma con toda la información relevante, que además es muy sencillo de manejar y operar. A lo largo de este trabajo, la nomenclatura que se empleará para representar la MTH para la sección j , referenciada respecto al extremo de la sección i será iA_j .

Así pues, aplicando la división del modelo previamente mencionada, se puede definir un marco generalista para los procesos de modelado cinemático de robots hiperredundantes tal y como se observa en la figura 3.2. En concreto, este será el marco que se empleará a lo largo de todo el TFM. Tal y como se indicaba, el último paso para la obtención del modelo implica la multiplicación de matrices de transformación homogéneas entre sí, con el fin de modificar la referencia de la misma. La concatenación de MTHs permiten obtener la posición y orientación de cada sección respecto a la base del robot. Las otras dos etapas se definirán en función del robot y del modelo escogido para representarlo.

En general, la primera etapa depende completamente del robot que está siendo modelado. Al fin y al cabo, la relación entre la longitud de los cables y los parámetros de la sección dependen de cómo están dispuestos los cables en cada sección, dónde se fijan en los discos de los extremos... Por tanto, se trata de un paso que depende completamente de la pericia del ingeniero para encontrar dicha relación.

El segundo paso suele estar mucho más asociado a la hipótesis simplificadora escogida. En general, cada modelo define una serie de relaciones que transforman los parámetros de la sección más convenientes para su representación con la posición y orientación del extremo (y, por tanto, con la matriz de transformación correspondiente). A lo largo de esta memoria se irán considerando distintos modelos y sus ecuaciones asociadas.

3.3. Etapas para el modelado cinemático de un RHAC: Cinemática Inversa

A continuación, es necesario identificar el proceso equivalente para el problema inverso: la cinemática inversa. En este caso, el objetivo consiste en encontrar las longitudes de los cables que son necesarias para alcanzar una determinada configuración del robot. Una vez más, se trata de un problema de una complejidad ciertamente elevada, especialmente según se incrementan los grados de libertad. Cuando, además, se acumulan secciones, los efectos de unas sobre las otras también complican en mayor medida estos modelos. [42]

La definición concreta del problema cinemático inverso es, indudablemente arbitraria. ¿Qué se entiende por el objetivo final? ¿El usuario del robot va a definir la configuración completa del robot? ¿Únicamente se va a introducir el punto final del extremo? ¿Qué información se dispone del entorno? Todas esas preguntas deben ser resueltas para poder definir bien un modelo inverso completo.

En este caso, se ha decidido optar que, con el fin de contemplar todas las etapas del modelado, en el caso más general se considerará que el usuario sólo introduce una pose final para el extremo robot. Esto es, el modelo cinemático inverso contemplará como valores de entrada las coordenadas espaciales $[x \ y \ z]^t$. La justificación es sencilla: en este caso es necesario aplicar un algoritmo de optimización para encontrar la configuración completa del robot. Si el usuario del robot en algún momento necesitara introducir una configuración de robot, sería tan sencillo como eliminar del modelo temporalmente el primer paso.

Así pues, el desarrollo del modelo cinemático inverso se puede realizar de forma equivalente al modelo directo. El problema global se divide en varias etapas, las cuales se particularizarán en función del robot concreto escogido, así como de la hipótesis simplificadora (ver figura 3.3). Una vez más, este diagrama constituirá el marco teórico para todos los modelos de cinemática inversos estudiados aquí.

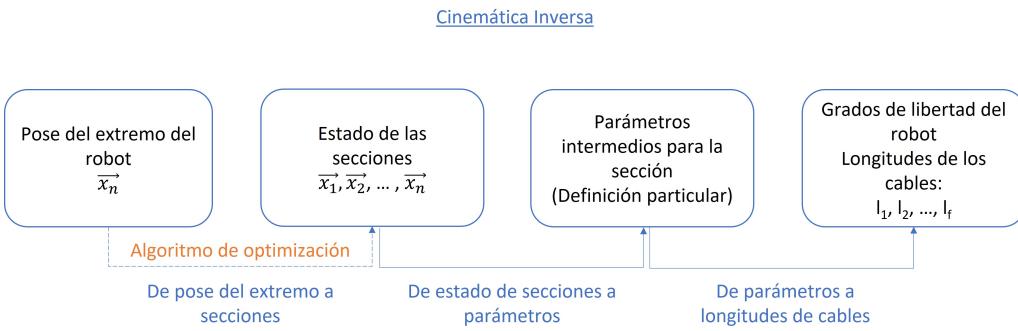


Figura 3.3: Resumen de las distintas etapas para el modelado cinemático inverso. Elaboración propia

El primer paso representa el algoritmo de optimización escogido, y es completamente independiente de otras tareas. Constituye una etapa muy bien estudiada en distintos trabajos académicos, además de poder emplear numerosas técnicas diferentes ([33], [10], [41]). El objetivo del mismo es encontrar una serie de poses, devolviendo una posición y orientación para cada sección del robot, que permitan que el extremo alcance su punto objetivo.

En general, este paso usa algoritmos que, modelando bien el robot, permiten que esta distribución se realice de forma completamente independiente a la definición de los grados de libertad y a la morfología del robot individual. Las secciones se pueden representar por sistemas de referencia, no necesitando mayor información. El único aspecto que sí debe considerarse es la ubicación de dichos sistemas de referencia en el robot real. Un ejemplo de estos algoritmos de optimización podría ser el algoritmo Natural-CCD [22], que se describirá más adelante en el TFM.

Las etapas posteriores del modelado inverso sí dependen de los modelos escogidos y sus etapas simplificadoras, así como del robot particular considerado. En concreto, serán desarrolladas para cada modelo aquí presentado, pero todas ellas incluyen una transformación desde el estado de cada sección hasta una serie de parámetros intermedios. Posteriormente, estos parámetros se relacionan en los modelos inversos con las longitudes de los cables.

4 | MODELOS CINEMÁTICOS PARA RHAC

En este capítulo se comienza el desarrollo de distintos modelos aplicables al modelado de robots hiperredundantes. Cada uno de ellos supone una hipótesis simplificadora que permitirá desarrollar las ecuaciones matemáticas pertinentes. El desarrollo para cada modelo será de forma paralela: primero se aplicará el esquema desarrollado en el capítulo anterior a cada una de las hipótesis simplificadoras; posteriormente se realizará el desarrollo matemático propiamente dicho para cada uno de los casos, presentando tanto el modelo directo como el inverso.

4.1. Hipótesis discreta: Método de Denavit-Hartenberg

4.1.1. Aplicación de la hipótesis

El primer modelo que se propone como posible para los RHAC es, precisamente, el de la hipótesis de robots discretos. Se basa en modelizar el robot como una sucesión de articulaciones de uno o dos grados de libertad, distinguibles perfectamente. Es un modelo extremadamente simple, pero aplicable a muchos robots que, según la clasificación de la sección 2.2 sean RHAC discretos. En particular, este modelo permite aplicar un método extremadamente útil para la obtención de las ecuaciones cinemáticas, por ser una técnica sistemática y muy extendida en el ámbito de la robótica: el algoritmo de Denavit-Hartenberg.

El método, que se expone posteriormente, es aplicable directamente sobre la segunda etapa del marco teórico propuesto para la cinemática directa. Fijando cada uno de los grados de libertad de las articulaciones (también llamados coordenadas generalizadas) como los parámetros de la sección el algoritmo da una relación directa entre parámetros de sección y Matrices de Transformación Homogéneas. Únicamente el primer paso queda sin definir completamente. Tal y como se indicaba en la sección 3.2 es altamente dependiente de la geometría particular del robot, y no se puede dar un método general.

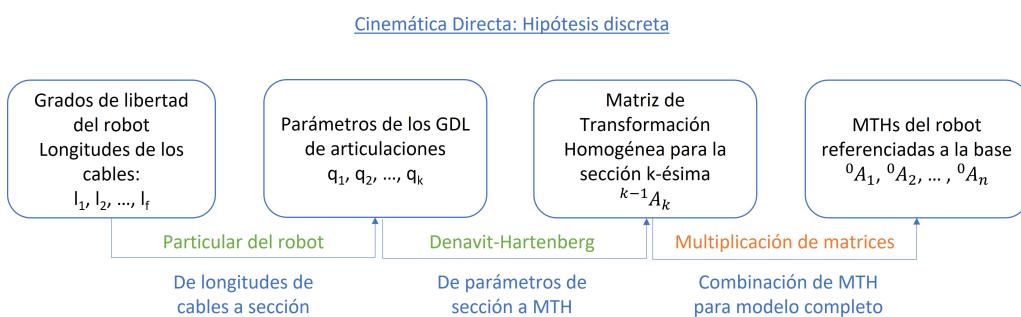


Figura 4.1: Aplicación del marco teórico directo a la hipótesis discreta. Elaboración propia

La cinemática inversa también permite la aplicación de la hipótesis de robot discreto, y el método de Denavit-Hartenberg es aplicable para ayudar en el procedimiento de la obtención de las ecuaciones. En este caso, la relación entre los estados de las secciones y los parámetros se obtiene de forma única y particular para cada robot en concreto.

No obstante, una vez las coordenadas generalizadas se han obtenido, las matrices de transformación homogéneas se pueden obtener para cada sección empleando el método de Denavit-Hartenberg. Aplicando sucesivas transformaciones a los puntos finales de cada cable (es decir, los puntos por donde están fijados o atraviesan cada uno de los discos de las secciones), se puede, mediante resta de vectores, calcular su posición en el espacio y deducir la longitud total. Este procedimiento se puede ver aplicado a un ejemplo concreto en la sección 7.2.1.

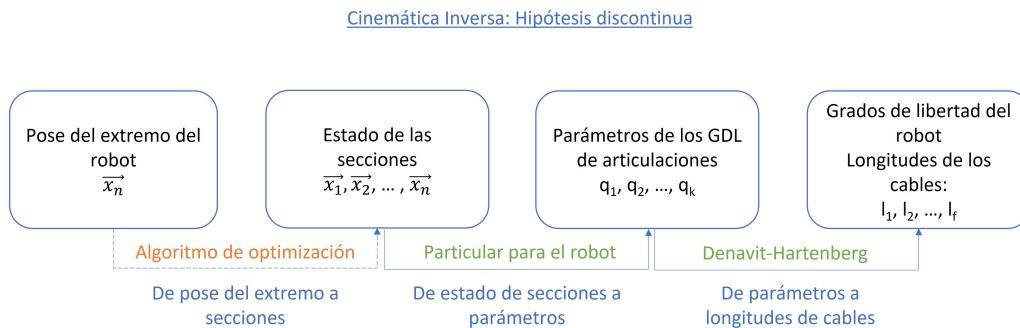


Figura 4.2: Aplicación del marco teórico inverso a la hipótesis discreta. Elaboración propia

4.1.2. Explicación del método

El método de Denavit-Hartenberg es un algoritmo basado en el álgebra matricial, propuesto por Denavit y Hartenberg a mitad del siglo XX para la obtención de un modelo cinemático en una cadena de articulaciones. Se basa en la selección de una serie de sistemas de referencia que permiten obtener una sencilla relación entre ellos, reduciendo los seis grados de libertad de un espacio tridimensional a cuatro transformaciones básicas. [3] A partir de este método es sencillo encontrar un modelo de cinemática directa para robots formados por cadenas cinemáticas discretas.

Estas cuatro transformaciones para dos articulaciones consecutivas $\{S_{i-1}\}$ y $\{S_i\}$ son las siguientes:

- Rotación en torno al eje $z_{i-1} \theta$ grados.
- Traslación a lo largo del eje $z_{i-1} d_i$ unidades de longitud.
- Traslación a lo largo de $x_i a_i$ unidades de longitud.
- Rotación en torno al eje $x_i \alpha_i$ unidades de longitud.

Con estas transformaciones como base, se comprueba que el paso de un sistema de referencia a otro se obtiene a partir de la expresión:

$$\begin{aligned}
 {}^{i-1}A_i &= Rot_z(\theta_i) T(0, 0, d_i) T(a_i, 0, 0) Rot_x(\alpha_i) \\
 &= \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin (\alpha_i) \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)
 \end{aligned}$$

A partir de la ecuación anterior se pueden obtener las posiciones de cada sistema de referencia a partir de la concatenación de matrices de transformación homogéneas. En este sentido, se puede

intuir cómo la extensión de este método a robots hiperredundantes formados por concatenaciones de articulaciones discretas es directa. El método de Denavit-Hartenberg es independiente del número de articulaciones del robot, y es por tanto aplicable.

El algoritmo define una serie de pasos necesarios para poder escoger los sistemas de referencia para cada articulación. Estos pasos se detallan a continuación, obtenidos a partir de [3]:

1. Se numera cada eslabón móvil de 1 a n. Se numera como 0 la base fija del robot.
2. Se numera cada articulación de 1 (más cercana a la base) a n (más lejana). Si existe alguna articulación con más de un grado de libertad, se divide en k articulaciones de 1 solo grado de libertad.
3. Se localiza el eje de cada articulación. Si es rotativa, se toma su eje de giro. Si es prismática, se toma el eje en el que se desplaza.
4. Se coloca el eje z_i sobre el eje de la articulación $i + 1 \forall i = (0, 1, \dots, n - 1)$
5. Se sitúa el sistema $\{S_0\}$ sobre cualquier punto del eje z_0 . Los ejes x_0 e y_0 se sitúan como un sistema dextrógiro.
6. $\forall i = (0, 1, \dots, n - 1)$ se coloca $\{S_i\}$ en la intersección de z_i con la perpendicular común a z_{i-1} y z_i . Si fueran paralelos, se sitúan en la articulación $i + 1$.
7. Se sitúa x_i en la perpendicular común a z_{i-1} y z_i .
8. Se define y_i para formar un sistema dextrógiro.
9. Se sitúa z_i en el extremo del robot para que z_n coincida con z_{n-1} y que x_n sea paralelo a la perpendicular común de z_{i-n} y z_n .

A partir de esta definición de los distintos sistemas de referencia, se pueden obtener los distintos parámetros de Denavit-Hartenberg para cada transformación ${}^{i-1}A_i(\theta_i, d_i, a_i, \alpha_i)$. De nuevo, según se indica en [3] estos obtienen como:

- θ_i : ángulo que debe girar en torno a z_{i-1} para que x_{i-1} y x_i sean paralelos. Se corresponde con el grado de libertad de una articulación rotativa.
- d_i : la distancia, medida a lo largo de z_{i-1} que habría que desplazar $\{S_{i-1}\}$ para que x_i y x_{i-1} queden alineados. Se corresponde con el grado de libertad de una articulación prismática.
- a_i : distancia, medida a lo largo de x_i que habría que desplazar $\{S_{i-1}\}$ para que su origen coincida con $\{S_i\}$.
- α_i : ángulo que habría que girar en torno a x_i para que el nuevo $\{S_{i-1}\}$ coincida con $\{S\}$.

Debe hacerse notar que cada una de las matrices como la que se indica en la ecuación (4.1) es válida para un único grado de libertad. No obstante, como se indicó en la descripción de los RHAC discretos, no es infrecuente encontrarse con robots que utilicen articulaciones de dos grados de libertad. En estos casos, la modelización se logra definiendo dos sistemas de referencia de Denavit-Hartenberg distintos con el mismo origen, y multiplicando ambas matrices entre sí: ${}^{i-1}A_i \cdot {}^{i-1}A_{i1} \cdot {}^{i1}A_i$.

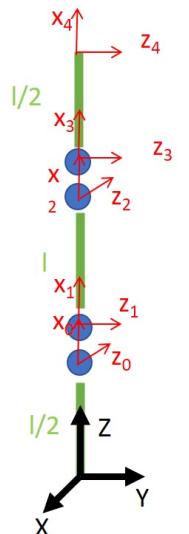
La multiplicación y concatenación de las sucesivas matrices ${}^{i-1}A_i$ permite posteriormente obtener el modelo completo del robot, referenciado a una articulación en concreto. La gran utilidad que tiene emplear este método de representación es que cualquier vector definido localmente

en una sección (como por ejemplo, la posición de los cables), puede transformarse con sencillez entre sistemas de referencia.

A continuación se va a mostrar la aplicación de este algoritmo a un ejemplo de robot con varias articulaciones. La extensión a un robot hiperredundante es, como ya se ha explicado, cuestión de repetir el método para un número mayor de articulaciones. Como aclaración, indicar dos cuestiones. Si se desea cambiar el modelo cinemático a un sistema fijo, diferente al que se obtiene en la aplicación directa del algoritmo, simplemente se incluye una transformación fija entre dicho sistema de referencia y el $\{S_0\}$.

En segundo lugar, es frecuente a la hora de aplicar Denavit-Hartenberg usar esquemas para simplificar el robot. En ese sentido, se va a emplear una representación esquemática en el que, como se explicó anteriormente, a pesar de que una sola articulación tenga dos grados de libertad se van a independizar en el esquema y se van a dibujar dos. Debe recordarse a la hora de obtener los parámetros de Denavit-Hartenberg que espacialmente tienen su origen en el mismo punto.

Se muestra a continuación la aplicación del método de Denavit-Hartenberg a un robot con dos articulaciones de dos grados de libertad:



θ	d	a	α
q_1	0	0	$-\pi/2$
q_2	0	l	$\pi/2$
q_3	0	0	$-\pi/2$
q_4	0	$l/2$	0

(a) Figura esquemática para la aplicación del algoritmo de Denavit-Hartenberg. Elaboración propia

(b) Valores de los parámetros de Denavit-Hartenberg para el robot indicado

Figura 4.3: Aplicación del método de Denavit-Hartenberg a un caso de ejemplo

Finalmente, para obtener el modelo cinemático definitivo, se aplica la ecuación de cada transformación de un sistema de referencia a otro, conjuntamente con la transformación fija de la base, es decir:

$$\begin{aligned}
 {}^B A_0 &= T(0, 0, l/2) Rot_y(-\pi/2) \\
 {}^0 A_1 &= Rot_z(q_1) T(0, 0, 0) T(0, 0, 0) Rot_x(-\pi/2) \\
 {}^1 A_2 &= Rot_z(q_2) T(0, 0, l) T(0, 0, 0) Rot_x(\pi/2) \\
 {}^2 A_3 &= Rot_z(q_3) T(0, 0, 0) T(0, 0, 0) Rot_x(-\pi/2)
 \end{aligned}$$

$${}^3A_4 = \text{Rot}_z(q_4) T(0, 0, l) T(0, 0, 0) \text{Rot}_x(0)$$

$$MTH = {}^B A_0 *^0 A_1 *^1 A_2 *^2 A_3 *^3 A_4 \quad (4.2)$$

4.2. Modelo PCC - Piecewise Constant Curvature

El segundo modelo que se presentará en este trabajo es el modelo PCC o de hipótesis de curvatura constante. La cinemática *Piecewise Constant Curvature* se basa, como su propio nombre indica, en la hipótesis de que el robot se puede dividir en un número finito de arcos de curvatura constante. [42] Es una hipótesis deseable en robots hiperredundantes, y se pueden aplicar con bastante éxito a muchos robots hiperredundantes. Generalmente, las condiciones necesarias para que dicho modelo sea aplicable son, en primer lugar, poder asumir despreciables los efectos de la gravedad, y en segundo lugar, que los discos entre secciones estén fijos en el núcleo central, permitiendo que se doble provocando una fricción despreciable. [38]

Cada uno de los arcos que tienen curvatura constante que posee el robot se corresponderá con una de las secciones del mismo. Al tratarse de una curva plana, puede contenerse el núcleo del robot en un plano, así como al centro de la circunferencia a la cual pertenece el arco.

4.2.1. Cinemática directa

En la figura 4.4 puede observarse la aplicación del marco teórico que se ha establecido para los modelos de cinemática directa. En primer lugar, se deben definir los parámetros que se escogen para reflejar el estado de cada sección. Como se ha indicado, la hipótesis permite modelar el robot siguiendo una curva plana, un arco de circunferencia.

Este arco se puede representar con tres valores distintos. En primer lugar, el ángulo azimutal, ϕ , que representa la orientación del plano que contiene la curva trazada por el robot. En segundo lugar, la longitud total del arco, l . Finalmente, el inverso del radio de la circunferencia que contiene al arco se denomina curvatura κ .

Una vez se han determinado los parámetros de la sección, las dos transformaciones que quedan por determinar son la transformación dependiente del robot (esta es, la que va desde las longitudes de los cables a los parámetros de sección), y la expresión independiente de la morfología concreta (aquella que relaciona los parámetros de sección con una matriz de transformación homogénea). Ambas transformaciones se analizarán a continuación.

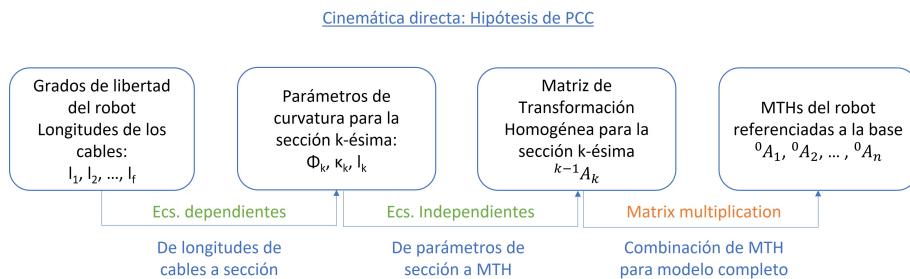


Figura 4.4: Aplicación del marco teórico directo a la hipótesis PCC. Elaboración propia

Ecuaciones Independientes - Método Geométrico

Por simplicidad, se comenzará con la deducción de la segunda transformación, aquella que es independiente del robot. Así, se van a emplear relaciones puramente geométricas para encontrar las ecuaciones matemáticas para una sola sección k [30] [42]. No obstante, en la próxima sección se contemplarán posibles alternativas para llegar a las mismas relaciones. Dado que en todo el proceso se va a referir la deducción a una sección, el subíndice k se omitirá en estas ecuaciones.

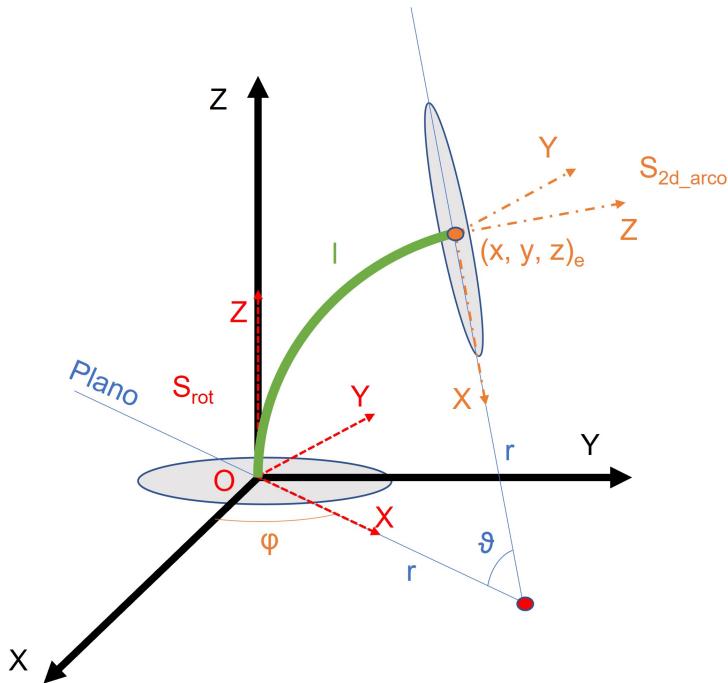


Figura 4.5: Esquema auxiliar para el modelo PCC. Elaboración propia

Así pues, tal y como se indicaba, se definirá un arco tridimensional en el espacio. Asimismo, se tomará como referencia un sistema de referencia centrado en el origen O , que se puede encontrar en el centro del disco inferior de la sección (ver figura 4.5). El eje z se define como tangente a la curva central de la sección en ese punto, ortogonal al propio disco inicial. Los ejes x e y pertenecerán al plano definido por el disco, y ortogonales el uno al otro. El eje x , en concreto, se define arbitrariamente. El eje y , no obstante, se obtiene a partir del dato de que se trata de un sistema dextrógiro.

Para definir el arco en sí mismo, se tienen que emplear los parámetros previamente definidos para la representación del arco: l , ϕ y κ . Éste último, no obstante, se puede transformar de forma sencilla en el ángulo θ , presente en la imagen. El problema así queda reducido en encontrar la relación entre los parámetros κ, ϕ y l y la posición del extremo de la sección x_e, y_e, z_e . La orientación queda definida por la propiedad de que la curva central (esto es, el arco), es siempre perpendicular a la sección trasversal del robot en cada punto.

Este objetivo se puede lograr combinando dos movimientos (véase la ecuación (4.3)): en primer lugar, se define una rotación para el sistema de referencia inicial de ϕ grados, en torno al eje z . De esta forma se obtiene el sistema de referencia S_{rot} , el cual tiene el eje x_{rot} apuntando hacia el centro de la circunferencia que contiene al arco. Esto permite aprovechar una propiedad muy útil: la curva está completamente contenida en el plano definido por los ejes x y z de este sistema de referencia. De esta forma, el resto de la transformación se realiza en un mismo plano.

Es necesario unir una traslación y una rotación para alcanzar el punto final. La traslación define la posición del extremo, mientras que se necesita una rotación para que el eje z permanezca, como se ha definido, tangente a la curva en cada punto y perfectamente perpendicular a la sección transversal. La rotación se realiza en torno al eje y en sentido positivo, de tal manera que el eje x sigue apuntando en todo momento al centro de la circunferencia.

$${}^{k-1}A_k = T_{rot}(\phi) * T_{2d_arco}(\phi, \kappa, l) \quad (4.3)$$

La obtención de $T_{rot}(\phi)$ es muy sencilla. Se trata de una matriz de rotación en torno al eje z , de ϕ grados en sentido positivo.

$$T_{rot} = \begin{bmatrix} \cos \phi & -\sin \phi & 0 & 0 \\ \sin \phi & \cos \phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

En cuanto a la transformación necesaria para representar el arco plano, empleando una única matriz de transformación homogénea T_{2d_arco} es suficiente. Seguir el arco completo implica trasladar el origen de coordenadas del sistema de referencia al extremo del robot, mientras se realiza la rotación previamente descrita. No obstante, es interesante definir la matriz de tal forma que fuera aplicable a cualquier punto del arco, permitiendo así la representación completa de todos los puntos del mismo (teniendo un modelo completo de la configuración del robot).

Para ello, se define la variable auxiliar s , que representará el grado de avance por el arco, y teniendo como valor máximo la longitud final l ($s \in [0, l]$). Así pues, sustituyendo s en cada punto, se podrían obtener distintos sistemas de referencia, siendo el más relevante el situado en el extremo de la sección. En este momento, resulta más útil emplear la variable θ que la propia curvatura.

Para ello, en primer lugar se conoce que la curvatura se define como el inverso del radio de la circunferencia.

$$r = \frac{1}{\kappa} \quad (4.5)$$

A partir de este valor, el ángulo que define la rotación necesaria en torno al eje y (θ) para cada punto, se puede deducir empleando la figura 4.6, tal y como aparece en la ecuación (4.6):

$$\theta = \kappa s \quad (4.6)$$

Una vez más, para obtener la traslación, es interesante mantener la nomenclatura continua, empleando la variable s . Para ello, empleando la trigonometría del ángulo θ , puede deducirse que para cada punto, la traslación total es:

$$t = [r(1 - \cos \theta) \quad 0 \quad r \sin \theta] = [\frac{1 - \cos \kappa s}{\kappa} \quad 0 \quad \frac{\sin \kappa s}{\kappa}] \quad (4.7)$$

Ahora sí, se pueden introducir los dos movimientos (de rotación y de traslación) en una misma matriz de transformación homogénea T_{2d_arco} :

$$T_{2d_arco} = \begin{bmatrix} \cos \kappa s & 0 & \sin \kappa s & \frac{1 - \cos \kappa s}{\kappa} \\ 0 & 1 & 0 & 0 \\ -\sin \kappa s & 0 & \cos \kappa s & \frac{\sin \kappa s}{\kappa} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

Finalmente, las ecuaciones cinemáticas independientes del robot se pueden sacar multiplicando las dos transformaciones que se han explicitado en las ecuaciones (4.8) y (4.7). Con esto, se tiene ya la matriz general ${}^{k-1}A_k(s)$, siendo especialmente interesante su particularización en l :

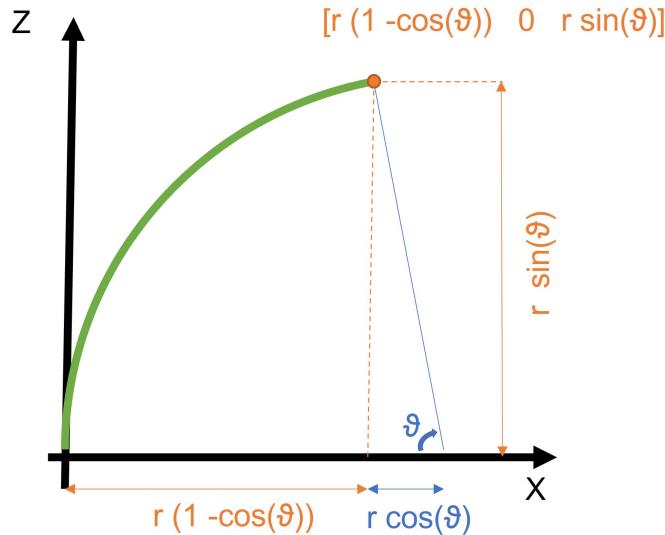


Figura 4.6: Figura auxiliar: plano que contiene al arco. Elaboración propia

$$k^{-1} A_k(l) = \begin{bmatrix} \cos \phi \cos \kappa l & -\sin \phi & \cos \phi \sin \kappa l & \frac{\cos \phi (1 - \cos \kappa l)}{\sin \phi (1 - \cos \kappa l)} \\ \sin \phi \cos \kappa l & \cos \phi & \sin \phi \sin \kappa l & \frac{\sin \phi}{\sin \kappa l} \\ -\sin \kappa l & 0 & \cos \kappa l & \frac{\kappa}{1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.9)$$

Ecuaciones Independientes - Alternativas al método geométrico

Por completitud, y para prevenir posibles confusiones a la hora de investigar toda la bibliografía existente, ha de tenerse en cuenta que a la hora de llegar a la transformación independiente, pueden adoptarse numerosas estrategias distintas. En el apartado anterior se ha desarrollado el llamado método geométrico, por ser el más intuitivo ya que emplea nociones básicas de geometría. No obstante, no es el único caso posible. Las ecuaciones que se obtendrán deberán ser equivalentes a las alcanzadas por el método geométrico, en todo caso.

A continuación, se desarrolla también una alternativa, por ser paradigmática. Puede definirse también la sección del arco continuo, según los modelos PCC, aplicando el algoritmo de Denavit-Hartenberg, a pesar de que los robots de curvatura continua no tienen articulaciones discretas en su cadena cinemática. [42] Un ejemplo reciente de la Universidad Nacional de Singapur ha aplicado de forma eficaz el método de Denavit-Hartenberg para estimar un modelo cinemático para un RHAC flexible y continuo. [20] En ese trabajo se utilizaba la herramienta computacional MATLAB, que está optimizada para el cálculo matricial. Así, se demuestra que es un método eficiente y sencillo.

Para poder realizar la aplicación del método de Denavit-Hartenberg es necesario, en primer lugar, definir la serie de articulaciones ficticias que se emplearán como grados de libertad del robot. A partir de este conjunto de articulaciones, se obtendrán los sistemas de referencia, en aplicación rigurosa del algoritmo. Con ello, se encontrará el conjunto de transformaciones entre dichos sistemas de referencia que, en aplicación de un número finito de parámetros, expresan la sección del robot. Existen varias alternativa para la definición de las articulaciones en la bibliografía, cada una con un desarrollo ligeramente distinto.

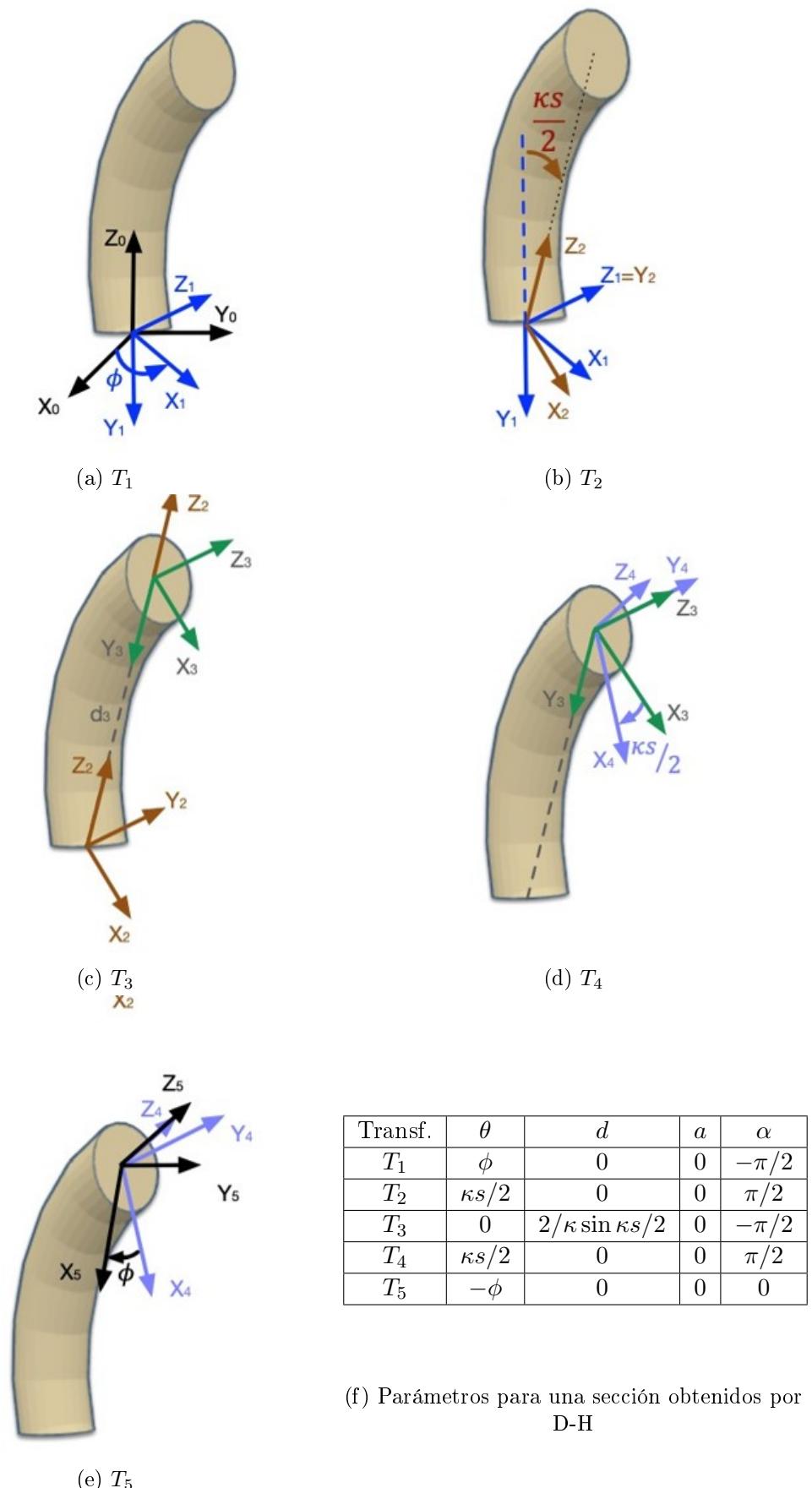


Figura 4.7: Aplicación del Algoritmo de D-H a un robot, aplicando el modelo PCC.
Elaboración propia

Por continuidad con el trabajo mostrado hasta ahora, y dado que las definiciones paramétricas que conlleva este desarrollo dan lugar a una transformación equivalente a la dada en la ecuación (4.9), se va a utilizar la alternativa desarrollada por Walker y Hannan en 2003 [15], modificada posteriormente por Webster [42] con el fin de cuadrar exactamente los sistemas de referencia empleados.

En este trabajo se utilizan cinco articulaciones virtuales para cada sección. Cada una de ellas lleva asociada un sistema de referencia, cuya transición se refleja en las transformaciones representadas, por orden, en las distintas imágenes de la figura 4.7. Las cinco transformaciones se explicitan a continuación.

1. La primera transformación (T1 - figura 4.7a) se emplea, al igual que en el caso anterior, para convertir el problema en un sistema plano: se utiliza una rotación de ϕ en torno al eje z .
2. La segunda transformación (T2 - figura 4.7b) representa una rotación de $\theta_2 = \frac{1}{2}\kappa s$ grados. Es la responsable de que el sistema de referencia apunte hacia el extremo final de la curva.
3. La tercera transformación (T3 - figura 4.7c) introduce la traslación del sistema de referencia. Dado que el eje z ya apunta hacia el extremo por la transformación anterior, ya se puede calcular directamente una distancia para un GDL de traslación: $d_3 = \frac{2}{\kappa} \sin \frac{\kappa s}{2}$
4. La cuarta transformación (T4 - figura 4.7d) aplica una nueva rotación de $\theta_4 = \frac{1}{2}\kappa s$. De esta forma, se consigue que el sistema de referencia obtenido sea tangente a la curva en el punto final, requisito previamente definido.
5. La quinta transformación (T5 - figura 4.7e) se utiliza para devolver el sistema desde el problema bidimensional, nuevamente al problema en el espacio.

Una vez definidas todas estas transformaciones, el resto del problema consiste en la aplicación sistemática del algoritmo desarrollado en la sección 4.1.2 a estos sistemas de referencia. Esta operación resulta en los parámetros de Denavit-Hartenberg que se pueden observar en la tabla de la figura 4.7f. A partir de esta tabla se pueden calcular las ecuaciones matemáticas (es decir, las matrices de transformación homogéneas) y, multiplicándolas entre sí, obtener las ecuaciones finales.

Más allá del algoritmo de Denavit-Hartenberg, existen otras metodologías diversas que pueden ser utilizadas con el fin de obtener las mismas ecuaciones matemáticas: el uso de geometría diferencial [14], los sistemas de referencia de curvatura de Frenet [42], métodos integrales [5]... Cada uno de ellos debe obtener como resultado unas ecuaciones y una matriz de transformación homogénea perfectamente equivalente a la que se ha deducido en este trabajo.

Por ello, todos estos sistemas de deducción de ecuaciones, aunque en la bibliografía a veces no se presenten como tal explícitamente, se corresponden con modelos de curvatura constante o PCC: todos ellos constituyen un único modelo, una única hipótesis simplificadora. El único término restante con el fin de obtener el modelo cinemático completo es la relación existente entre longitudes de cables y los parámetros de curvatura de la sección.

Esto hace que la elección entre estas formas de deducir las ecuaciones sea indiferente de cara a las ecuaciones. En casos como el de Kim [20], la elección es apropiada por disponer de las herramientas matemáticas adaptadas perfectamente a ese enfoque. En el caso de este trabajo, se ha preferido dar preponderancia al método geométrico, por considerarse más intuitivo. No

obstante, es completamente arbitraria la elección de método de obtención de las ecuaciones, y únicamente factores externos pueden aconsejar la adopción de una u otra alternativa.

Ecuaciones Dependientes - Longitud de los cables

Los apartados anteriores han definido ya la relación existente entre los parámetros de curvatura y las matrices de transformación homogénea que sirven para determinar de forma precisa la configuración del robot. Las otras ecuaciones que son necesarias a continuación son aquellas que relacionan las longitudes de los cables con los parámetros de curvatura de la sección k-ésima. Como ya se indicaba en la sección 3.2, se trata de un conjunto de ecuaciones que dependen completamente del robot concreto que se esté pretendiendo modelar.

Esta es la razón por la que no es posible definir un modelo completamente general: en esta etapa es necesario un análisis particularizado, que dependerá en última instancia de la habilidad propia del ingeniero que modele el robot. No obstante, existen ciertas morfologías de robot que son relativamente frecuentes en el mundo académico y en el diseño de robots. En estos casos, es posible encontrar una deducción de sus ecuaciones en la bibliografía, y adaptarlas al caso particular mediante ciertas adaptaciones sencillas.

Los robots con tres o cuatro cables por sección, distribuidos homogéneamente alrededor de la sección. Las ecuaciones han sido estudiadas en numerosos artículos, y existen muchas deducciones disponibles. Por ejemplo, el caso de tres cables por sección, equidistantes respecto al núcleo central del robot y conformando un triángulo equilátero en su disposición (120 deg entre cables) se puede encontrar en el artículo [19]. En el artículo de Webster [42] puede también extraerse la misma demostración para tres y cuatro cables, de forma similar. Los casos de un mayor número de cables por sección no se han estudiado, porque incrementar el número de cables no aumentaría el número de grados de libertad del robot, y complicaría su control.

En concreto, en este trabajo se ha optado por desarrollar el procedimiento de Webster. Para ello, se estudiará el caso de un robot con tres cables activos por sección, distribuidos homogéneamente alrededor del centro de la sección a una distancia equidistante, de radio d (véase la figura 4.8a). En esa misma sección del robot se han incorporado varios discos pasivos, de tal forma que los cables son guiados a lo largo del núcleo central.

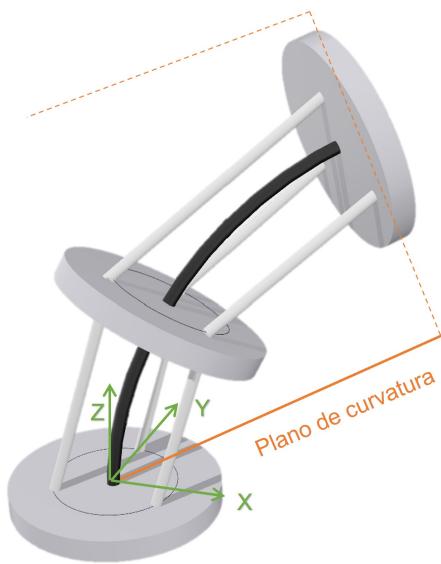
El núcleo admite la hipótesis de curvatura constante, siguiendo el modelo PCC. Así pues, las ecuaciones del arco de curvatura constante son las que determinan la posición de cada uno de los centros de los discos. No obstante, es necesario asumir una hipótesis adicional: los cables se conformarán entre los discos siguiendo una línea recta (ver figura 4.8b).

En la figura 4.8c puede verse la proyección en el plano (en una vista de planta) del disco de una de las secciones del robot. En la imagen se ha marcado el plano de curvatura (el que contiene el arco de circunferencia que define el núcleo del robot). También se ha marcado el plano paralelo que contiene a la curva que traza uno de los cables, al que se le asigna el número 1.

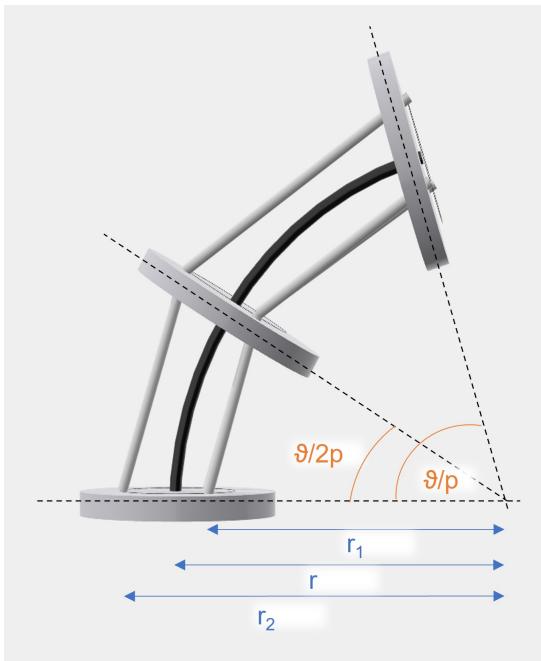
Dada la definición que se hizo en un principio del robot, los tres cables de la sección son equidistantes respecto al centro del disco. La distribución sigue una circunferencia de radio d , y los cables, para los que se usa el subíndice i , homogéneamente distribuidos quedan definidos en su origen mediante el ángulo ϕ_i . Se deduce fácilmente de la figura que:

$$r_i = r - d \cos \phi_i \quad (4.10)$$

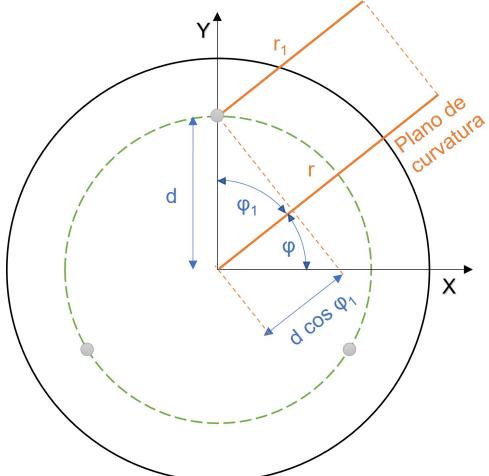
La demostración de esta relación es válida independientemente del número de cables que tenga el robot en la sección. La única hipótesis que requiere esta fórmula es la de equidistancia



(a) Modelo 3D del robot tipo considerado. Elaboración propia



(b) Hipótesis geométrica aplicada para obtener la posición de los cables. Elaboración propia



(c) Vista de planta de una sección del robot, marcando el plano de curvatura. Elaboración propia.

Figura 4.8: Figuras auxiliares para las relaciones geométricas de los cables

y distribución homogénea de los cables.

Regresando a la figura 4.8b, tanto la expresión para la longitud hipotética del núcleo de la sección (l_c), como la fórmula para obtener la longitud de cada uno de los cables l_i pueden ser deducidas. Sea p el número de discos pasivos que hay en una sección

$$l_c = 2 * p * r \sin(\theta/2p) \quad (4.11)$$

$$l_i = 2 * p * r_i \sin(\theta/2p) \quad (4.12)$$

Multiplicando la ecuación (4.10) por $2p \sin(\theta/2p)$, sustituyendo en las ecuaciones (4.11) y (4.12), y, finalmente, resolviendo, se puede sacar la siguiente ecuación:

$$l_c = l_i + d2p * \sin(\theta/2p) * \cos(\phi_i) \quad (4.13)$$

A partir de esta última ecuación se puede deducir una propiedad esencial para las siguientes deducciones. Como se ha indicado previamente, los cables se distribuyen homogéneamente en torno al centro de la sección, de tal manera que la suma total de todos los ángulos que definen su posición, ϕ_i , es igual a cero.

Mediante la aplicación de relaciones trigonométricas, si se suma la ecuación (4.13) para cada uno de los cables, entonces la longitud total del núcleo de la sección se puede calcular como la media aritmética de las longitudes reales de los cables, independientemente del número total de cables por sección f_k .

$$l_c = \sum_{i=1}^{f_k} \frac{l_i}{f_k} \quad (4.14)$$

Regresando al ejemplo particular de los tres cables, se aplica la ecuación (4.13) a dos cables distintos. Dado que las longitudes del núcleo central l_c deben ser las mismas en cada fórmula, se igualan las ecuaciones, y así se obtiene la ecuación (4.15).

$$d2p \sin(\theta/2p) = (l_2 - l_1)/(\cos \phi_1 - \cos \phi_2) \quad (4.15)$$

Este paso se repite para un par distinto de cables (por ejemplo, el 2 y el 3). Posteriormente, ambas ecuaciones se pueden combinar, dando lugar a la relación:

$$\phi = \tan^{-1} \left(\frac{\sqrt{3}(l_2 + l_3 - 2l_1)}{3(l_2 - l_3)} \right) \quad (4.16)$$

Finalmente, los parámetros de curvatura se obtienen a partir de las ecuaciones (4.11) y (4.12) mediante las cuales:

$$\sin(\theta/2p) = l_c/2pr = l_i/2pr_i \quad (4.17)$$

Así, para cada cable i :

$$r_i = \frac{l_i}{\kappa l_c} \quad (4.18)$$

Si se aplica la ecuación (4.10), entonces:

$$\kappa = \frac{(l_c - l_i)}{l_c d \cos(\phi_i)} \quad (4.19)$$

Dada la referencia que se ha definido previamente:

$$\kappa = \frac{l_2 + l_3 - 2l_1}{(l_1 + l_2 + l_3)d \sin(\phi)} \quad (4.20)$$

Con esto se tiene ya una primera ecuación para κ . No obstante, es necesario dejar la ecuación en función de los parámetros iniciales del problema. Para ello, se puede sustituir ϕ empleando la

ecuación ya obtenida (4.16). Utilizando también la ecuación trigonométrica $\sin(\tan^{-1}(y/x)) = y/\sqrt{x^2 + y^2}$:

$$\kappa = \frac{2\sqrt{l_1^2 + l_2^2 + l_3^2 - l_1l_2 - l_1l_3 - l_2l_3}}{d(l_1 + l_2 + l_3)} \quad (4.21)$$

Ahora bien, todavía no se conoce la longitud real del núcleo de la sección. Hasta ahora, únicamente se ha calculado la longitud de un cable hipotético que circulara por el centro. Ahora bien, esto significa que ese cable traza una recta entre cada disco, que no representa la longitud real del núcleo. Para obtener la longitud de la curva, se aplican las definiciones de los parámetros geométricos a la ecuación (4.11):

$$l = \frac{2p}{\kappa} \sin^{-1} \left(\frac{l_c \kappa}{2p} \right) \quad (4.22)$$

Finalmente, se aplican las ecuaciones (4.21) y (4.14):

$$l = \frac{pd(l_1 + l_2 + l_3)}{2\sqrt{l_1^2 + l_2^2 + l_3^2 - l_1l_2 - l_1l_3 - l_2l_3}} \sin^{-1} \left(\frac{\sqrt{l_1^2 + l_2^2 + l_3^2 - l_1l_2 - l_1l_3 - l_2l_3}}{3pd} \right) \quad (4.23)$$

Combinando los dos pasos del modelo cinemático (el paso de los cables a los parámetros de curvatura y el paso de éstos a la configuración del robot), se dispone ya de un modelo cinemático directo completo para los RHAC mediante la aplicación de la hipótesis simplificadora del PCC. No obstante, el modelo se ha desarrollado, como se mencionaba al principio, para una única sección.

El paso de un modelo monosección a un modelo multisección se puede lograr sin excesiva complejidad. Para ello, las transformaciones que se aplican a cada sección se deben combinar, multiplicando las matrices que las representan. No obstante, se debe tener en cuenta que puede ser necesario introducir algún factor corrector, puesto que los cambios en una sección pueden afectar a la forma de otras secciones. En estos casos, existen algoritmos iterativos, como en *tangle-un-tangle* [18] o la posibilidad de aplicar algoritmos de optimización para mejorar el modelo que se tiene.

4.2.2. Cinemática Inversa

En esta sección se va a explicar un posible modelo cinemático inverso para los robots hiper-redundantes a los que se puede aplicar la hipótesis PCC. La metodología que se va a emplear en esta sección parte de la hipótesis de que el estado para cada sección ya se ha fijado (es decir, que se ha aplicado el algoritmo de optimización correspondiente para encontrar esta distribución).

El objetivo de esta sección es, por tanto, encontrar las ecuaciones que relacionan la posición en el espacio de las secciones del robot para una configuración concreta con las longitudes de los cables que dicho posicionamiento requiere. Matemáticamente, son las ecuaciones que dan las longitudes l_1, l_2, \dots, l_f cuando se introducen los parámetros de la sección $[x \ y \ z \ \Phi \ \theta \ \Psi]^t$.

Al igual que en modelos anteriores, una vez superado el paso de definir la posición de las secciones mediante el algoritmo de optimización, se requieren dos transformaciones adicionales. La primera es la relación de los datos de la sección con los parámetros de curvatura, obtenidos a partir de sus definiciones geométricas. Posteriormente hay un segundo paso dependiente del robot concreto que se maneja, de manera equivalente al caso de la cinemática directa. La relación entre parámetros de curvatura y longitudes de cables no es general, y debe ser examinada para

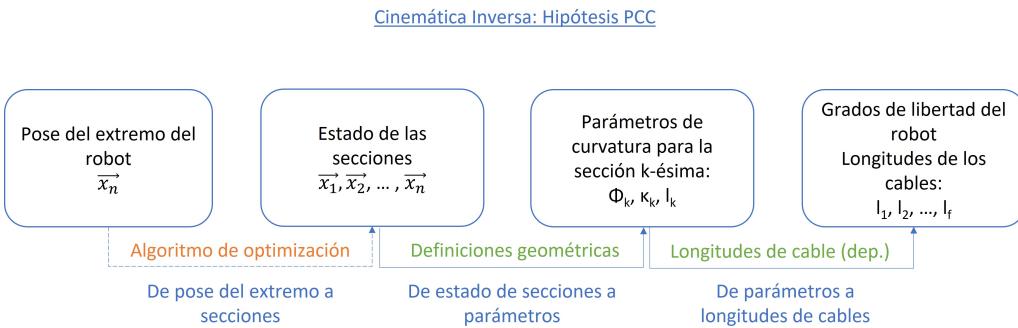


Figura 4.9: Aplicación del marco teórico inverso a la hipótesis PCC. Elaboración propia

cada caso concreto.

En primer lugar, y dada la asunción que se hace de la hipótesis de curvatura constante, es necesario obtener los parámetros de curvatura dadas las posiciones finales de cada una de las secciones. Nótese que, en este caso, el estado de la sección queda perfectamente definido con sólo tres parámetros (el mismo número de parámetros que definen la curvatura). Con las restricciones previamente establecidas, por las cuales la sección es perpendicular al núcleo en cada punto, la orientación queda ya definida por la posición.

Así, el problema se reduce a encontrar los parámetros de curvatura a partir de los datos $[x \ y \ z]^t$ para el extremo de cada sección. Para lograrlo, se va a seguir el método de Neppalli, [30]. Se puede comenzar por el ángulo azimutal ϕ , que se obtiene a partir de una relación sencilla:

$$\phi = \tan^{-1} \left(\frac{y}{x} \right) \quad (4.24)$$

La curvatura se obtiene posteriormente analizando el arco del robot en el plano vertical que lo contiene. En concreto, basta con encontrar el radio, ya que por definición $\kappa = 1/r$. En este caso, el extremo de la sección de robot se puede localizar en las coordenadas $x' = \sqrt{x^2 + y^2}$, $z' = z$, siendo los ejes prima los del plano. Así, el radio se puede encontrar igualando la distancia al centro del arco con el extremo:

$$(\sqrt{x^2 + y^2} - r)^2 + z^2 = r^2$$

Encontrando r e invirtiendo:

$$\kappa = \frac{2\sqrt{x^2 + y^2}}{x^2 + y^2 + z^2} \quad (4.25)$$

Finalmente, para obtener el valor de la longitud del arco l es necesario calcular el ángulo θ . Analizando de nuevo el arco contenido en el plano, los siguientes valores se pueden deducir:

$$\theta = \begin{cases} \cos^{-1}(1 - \kappa\sqrt{x^2 + y^2}), & z > 0 \\ 2\pi - \cos^{-1}(1 - \kappa\sqrt{x^2 + y^2}), & z \leq 0 \end{cases} \quad (4.26)$$

Finalmente, la transformación para obtener l es meramente geométrica: $l = r\theta = \theta/\kappa$.

Al obtener un modelo cinemático inverso, es esencial prestar atención a posibles singularidades que el propio modelo puede tener, y que pueden ser causantes de importantes problemas cuando se traten de aplicar posteriormente las ecuaciones. En este caso concreto pueden aparecer singularidades tanto para κ como para ϕ cuando el extremo del robot se sitúa en el eje vertical [30]:

- ϕ : el eje vertical z se corresponde con el conjunto de los puntos para los cuales $x = 0$ e $y = 0$. En esos casos, cuando el extremo del robot se sitúa perfectamente sobre el eje z , cualquier valor para ϕ es posible.
- κ : en este caso deben tenerse en cuenta dos posibilidades. Siempre que $z \neq 0$, entonces $\kappa = 0$ y $s = z$ pueden utilizarse. Esto presupone que la longitud del robot se puede variar y que el valor de z es positivo, necesitando otros casos un lugar más a fondo. Por otra parte, si $z = 0$, entonces el extremo del robot estaría situado perfectamente en el origen, formando una curva cerrada. También existirían múltiples soluciones desde el punto de vista matemático, siendo un ejemplo de ello $\kappa = 0$, $\phi = 0$, $\theta = 2\pi$. No obstante, ha de tenerse en cuenta que en muchas ocasiones no se trata de una solución mecánicamente viable.

Una vez obtenidos los parámetros de curvatura para una sección, el paso siguiente sería obtener las longitudes de los cables que se necesitan para alcanzar esa configuración determinada. Como ya se ha indicado, es una transformación plenamente dependiente del tipo de robot que se esté utilizando. Así pues, las ecuaciones aquí presentadas no deben considerarse como ciertas para todo RHAC.

No obstante, se desea obtener al menos un modelo cinemático completo. Al igual que en el caso directo, existen varias tipologías de RHAC paradigmáticas, que han sido estudiadas en profundidad en la bibliografía. Por ello, se desarrollará el mismo modelo que en el caso de la cinemática directa: el robot con tres cables homogéneamente distribuidos por sección. Para resolver su modelo, es suficiente con combinar las ecuaciones (4.13) y (4.12), presentadas en la sección 4.2.1, que definían la longitud de los cables. Así, se obtiene que:

$$l_i = 2 * p * r_i \sin(\theta/2p) \quad (4.27)$$

En la bibliografía pueden encontrarse otros múltiples enfoques para obtener esta ecuación. Un ejemplo de ello es el uso de deducciones geométricas, que pueden posteriormente transformarse en esta expresión a partir de la definición de r_i [19]:

$$l_i = 2 * p * \sin\left(\frac{\theta}{2p}\right) * \left(\frac{1}{\kappa} - d \sin(\phi + \phi_i)\right) \quad (4.28)$$

Otro aspecto que debe tenerse en cuenta cuando se trabaja con el modelo cinemático inverso se relaciona con los robots multisección. La extensión de estas ecuaciones de la longitud de los cables puede realizarse, siempre y cuando se siga un orden estricto de cálculo, puesto que se deben tener en cuenta las múltiples secciones que un mismo cable atraviesa. Cuando una sección atravesada por el cable cambia su posición, la longitud del cable también se ve afectada (como cable pasivo), y se deben aplicar las ecuaciones con los parámetros de curvatura correspondientes.

Esto se justifica cuando se observa la figura 4.10. En ella, la segunda sección $S2$ tiene un radio de arco ligeramente menor a la sección $S1$. Además, los cables tienen un desfase respecto a los de la sección $S1$, puesto que atraviesan la sección por agujeros distintos permitiendo su simultaneidad. Claramente, si se desea posicionar la segunda sección, con el fin de establecer las longitudes de los cables, tienen que tenerse en cuenta ambos segmentos de cable: tanto el segmento activo que se fija en $S2$ como su parte pasiva que atraviesa $S1$.

Como ejemplo, para el robot de la figura 4.10, siendo l_{ik} la longitud del cable i para la sección k :

$$l_{i1} = 2 * p_1 * \sin\left(\frac{\theta_1}{2p_1}\right) * \left(\frac{1}{\kappa_1} - d_1 \sin(\phi + \phi_i)\right) \quad (4.29)$$

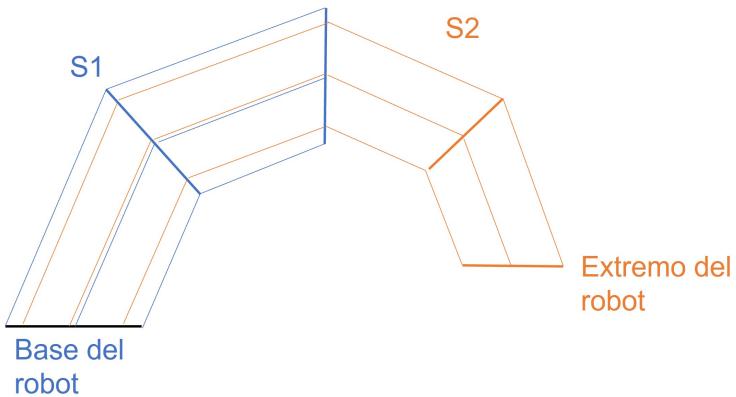


Figura 4.10: Longitudes de los cables para robots multisección. Elaboración propia

$$l_{i'2} = 2 * p_1 * \sin\left(\frac{\theta_1}{2p_1}\right) * \left(\frac{1}{\kappa_1} - d_1 \sin(\phi + \phi_{i'})\right) + 2 * p_2 * \sin\left(\frac{\theta_2}{2p_2}\right) * \left(\frac{1}{\kappa_2} - d_2 \sin(\phi + \phi_{i'})\right) \quad (4.30)$$

En particular, las fórmulas anteriores deben ser aplicadas tres veces. Una para calcular directamente las longitudes de los cables que se fijan en la sección 1: l_1 , directamente como se ha visto en este apartado del trabajo. Posteriormente, se usan esos mismos parámetros de curvatura (los de $S1$), pero ahora modificando d (si fuera necesario) y ϕ_i , para poder aplicarlos a la parte de cable de $S2$ que atraviesa la primera sección. Finalmente, la tercera vez que se aplican son para obtener los parámetros de curvatura y las longitudes de cable de $S2$.

Generalizando para n secciones se puede verificar fácilmente que, en la primera sección del robot, las fórmulas se deben aplicar para cada uno de los cables que intervienen en el robot, manejando el total de las n secciones. En la segunda sección deben aplicarse a los cables de todas las secciones menos la primera. Extrapolando para todas las secciones, y asumiendo f_k cables por sección, el número total de operaciones que deben ser computadas para n secciones es:

$$f_k(n + (n - 1) + \dots + 2 + 1) = f_k \sum_{j=1}^n j = f_k \frac{n(n + 1)}{2}$$

No obstante, debe ser remarcado que, aunque las ecuaciones hayan sido obtenidas geométricamente (y por tanto, en teoría son aplicables a cualquier robot que cumpla con la morfología especificada) no es completamente generalizable. A día de hoy existen muchos desarrollos con robots de longitud variable, esto es, que pueden crecer y hacerse más largos o más cortos en función de la tarea (la longitud se convierte en uno de sus grados de libertad).

Otros ejemplos son los robots blandos: su posibilidad de retraerse provoca cambios en la longitud incompatibles con las hipótesis aceptadas para estas ecuaciones. Para estos casos habría que adaptar las ecuaciones como se exponen aquí.

Por otra parte, la mayor parte de los robots continuos actuales tienen un núcleo rígido. Esto implica que la longitud entre secciones tiene que ser prácticamente rígida. Ahora bien, esto significa que el espacio de estados no es completamente controlable. Uno de sus grados de libertad (la distancia entre secciones) ya está fijada.

Así, no todo el espacio tridimensional puede ser alcanzado por el robot, y esto debe ser tenido en cuenta en la aplicación del modelo cinemático. Por razones semejantes a las esgrimidas en la

sección 3.2, es bueno introducir estas limitaciones en la búsqueda de soluciones.

Finalmente, hay que considerar otro efecto que también afecta a los robots multisección. Existe un cierto acoplamiento entre secciones, de tal manera que un cambio en una de las secciones afectan a la posición de las secciones a su alrededor, incluso cuando aquella no se ha modificado explícitamente. Aunque en algunos casos, estos errores pueden ser despreciables, un algoritmo *tangle / untangle* diseñado por Jones y Walker puede utilizarse para mejorar la precisión. [18]

4.2.3. Cinemática Diferencial

En esta sección se propone una última alternativa a la hora de modelar robots hiperredundantes mediante la hipótesis PCC pura. Se trata del empleo del modelo diferencial para conocer la configuración del robot en cada instante determinado.

El modelo diferencial es el conjunto de ecuaciones que relacionan las velocidades del sistema de referencia global con cada uno de los grados de libertad actuables del robot. En concreto, relacionaría la tasa de cambio de las longitudes de los cables con las velocidades de las articulaciones del robot. La integración de estos valores es la que permite conocer, en última instancia, la posición y la orientación del robot.

La obtención del modelo diferencial cinemático se puede obtener de dos formas distintas. La primera opción puede considerarse al derivar las ecuaciones del modelo cinemático directo. No obstante, no suele ser recomendable abordar así este problema, puesto que la derivación no es computacionalmente eficiente. La otra opción implica utilizar un método similar al seguido en los casos directo e inverso: separar las transformaciones cinemáticas independientes de la tipología de robot y aquellas ecuaciones que sí dependen explícitamente del tipo de robot a emplear.

Así pues, la solución más frecuente es derivar de forma separada ambas transformaciones, descritas ya en la figura 4.4. Esto acaba dando lugar a una expresión específica de las relaciones entre el ratio de cambio de los parámetros de curvatura y las velocidades de los parámetros de curvatura de la sección, y a otra entre éstas últimas y las velocidades de las articulaciones.

Hay múltiples maneras de encontrar la primera matriz Jacobiana. Algunos trabajos, como el de Hannan [15], lo logran mediante la transformación directa de las matrices mediante el método de Denavit-Hartenberg. El problema principal de utilizar estos métodos tradicionales (ver [32]) para cinemáticas de tiempo real es el número extremadamente alto de operaciones que implican, especialmente dado que los puntos de referencia del robot cambian constantemente por su cariz reconfigurable. [6]

Otra alternativa posible al método tradicional, esta vez sí mucho más usado, es el empleo de la teoría de tornillos (en inglés, *screws*). Se trata de una herramienta computacionalmente mucho más eficiente, y que es muy útil para la representación de la información, si bien es menos intuitiva para el ingeniero. [43]

No obstante, en un trabajo por Chembrammel y Kesavadas [6], se propone un algoritmo novedoso para calcular la Jacobiana de un manipulador, a la vez que propone su aplicación a un robot hiperredundante. Se basa en descomponer la Jacobiana como el producto de dos matrices: L y P. A pesar de que se aplica directamente a robots discretos, puede ajustarse de forma sencilla para la hipótesis PCC. El artículo demuestra el algoritmo, y lo aplica a un robot de 2 GdL. En este trabajo se explica su adaptación al PCC.

Para poder usar el algoritmo, se debe definir el producto de Kronecker:

$$A \otimes B = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \otimes \begin{bmatrix} b_{11} & \dots & b_{1n'} \\ \dots & \dots & \dots \\ b_{m'1} & \dots & b_{m'n'} \end{bmatrix} = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \dots & \dots & \dots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & \dots & a_{11}b_{1n'} & \dots & a_{1n}b_{11} & \dots & a_{1n}b_{1n'} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{11}b_{m'1} & \dots & a_{11}b_{m'n'} & \dots & a_{1n}b_{m'1} & \dots & a_{1n}b_{m'n'} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m1}b_{11} & \dots & a_{m1}b_{1n'} & \dots & a_{mn}b_{11} & \dots & a_{mn}b_{1n'} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m1}b_{m'1} & \dots & a_{m1}b_{m'n'} & \dots & a_{mn}b_{m'1} & \dots & a_{mn}b_{m'n'} \end{bmatrix} \quad (4.31)$$

Para ser más claro, se plantea un ejemplo concreto. Siendo I_2 la matriz identidad de dimensión 2, y A otra matriz 2x2:

$$I_2 \otimes A = \begin{bmatrix} 1 \cdot a_{11} & 1 \cdot a_{12} & 0 \cdot a_{11} & 0 \cdot a_{12} \\ 1 \cdot a_{21} & 1 \cdot a_{22} & 0 \cdot a_{21} & 0 \cdot a_{22} \\ 0 \cdot a_{11} & 0 \cdot a_{12} & 1 \cdot a_{11} & 0 \cdot a_{12} \\ 0 \cdot a_{21} & 0 \cdot a_{22} & 1 \cdot a_{21} & 0 \cdot a_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ 0 & 0 & a_{11} & a_{12} \\ 0 & 0 & a_{21} & a_{22} \end{bmatrix} \quad (4.32)$$

Habiendo definido el producto de Kronecker, para comenzar el algoritmo se necesita la matriz de transformación homogénea del extremo de la sección referido a la base de la misma. Se ha calculado en la ecuación (4.9). Las tres variables que se van a usar para calcular el modelo diferencial son: l , ϕ , κ .

Las matrices L_i se calculan según la expresión:

$$L_i = \frac{\partial T_{indep}(q)}{\partial q_i} \quad (4.33)$$

Siendo q_i las coordenadas generalizadas mencionadas previamente. Las tres matrices correspondientes son:

$$L_l = \begin{bmatrix} -\kappa \cos \phi \sin \kappa l & 0 & \kappa \cos \phi \cos \kappa l & \cos \phi \sin \kappa l \\ -\kappa \sin \phi \sin \kappa l & 0 & \kappa \sin \phi \cos \kappa l & \sin \phi \cos \kappa l \\ -\kappa \cos \kappa l & 0 & -\kappa \sin \kappa l & \cos \kappa l \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.34)$$

$$L_\phi = \begin{bmatrix} -\sin \phi \cos \kappa l & -\cos \phi & -\sin \phi \sin \kappa l & ((\cos \kappa l - 1) \sin \phi) / \kappa \\ \cos \phi \cos \kappa l & -\sin \phi & \cos \phi \sin \kappa l & -((\cos \kappa l - 1) \cos \phi) / \kappa \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.35)$$

$$L_\kappa = \begin{bmatrix} -l \cos \phi \sin \kappa l & 0 & l \cos \phi \sin \kappa l & (\cos \phi(\cos \kappa l - 1)) / \kappa^2 + l \cos \phi \sin \kappa l / \kappa \\ -l \sin \phi \sin \kappa l & 0 & l \sin \phi \sin \kappa l & (\sin \phi(\cos \kappa l - 1)) / \kappa^2 + l \sin \phi \sin \kappa l / \kappa \\ -l \cos \kappa l & 0 & -l \sin \kappa l & (l \cos \kappa l) / \kappa - \sin \kappa l / \kappa^2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.36)$$

Así, la matriz L se construye mediante la combinación de las tres submatrices:

$$L = [L_l \ L_\phi \ L_\kappa] \quad (4.37)$$

En el trabajo de Chembrammel y Kesavadas, [6], el algoritmo es general para cualquier punto en el sistema de referencia. Aquí, por conveniencia y paralelismo con los desarrollos directos e inverso, se va a utilizar únicamente el extremo de la sección. Éste se puede definir como el vector origen transformado posteriormente por T_{indep} :

$$p = [0 \ 0 \ 0 \ 1]^T \quad (4.38)$$

El algoritmo propuesto divide la matriz Jacobiana en dos partes: la Jacobiana de la velocidad lineal y la Jacobiana de la velocidad angular. Comenzando con la parte lineal, la matriz P_{linear} se obtiene empleando el producto de Kronecker con la identidad:

$$P_{lin} = I_3 \cdot p = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \quad (4.39)$$

Para la parte angular, otro P_{ang} se obtiene como la inversa de la matriz de transformación homogénea original (es decir, T_{indep}). Finalmente, cada una de las dos Jacobianas (lineal y angular) se pueden encontrar multiplicando ambas matrices por cada parte:

$$J_{lin} = LP_{lin} = \begin{bmatrix} \cos \phi \sin \kappa l & (\sin \phi(\cos \kappa l - 1))/\kappa & (\cos \phi(\cos \kappa l - 1))/\kappa^2 + l \cos \phi \sin \kappa l/\kappa \\ \sin \phi \sin \kappa l & -(\cos \phi(\cos \kappa l - 1))/\kappa & (\sin \phi(\cos \kappa l - 1))/\kappa^2 + l \sin \phi \sin \kappa l/\kappa \\ \cos \kappa l & 0 & (l \cos \kappa l)/\kappa - \sin \kappa l/\kappa^2 \end{bmatrix} \quad (4.40)$$

La Jacobiana angular se obtiene de forma ligeramente diferente. Para cada coordenada generalizada q_i se calcula el producto $L_i P$. Para cada coordenada, se obtiene una componente diferente del producto $L_i P$:

$$J_\omega = \begin{bmatrix} (L_1 P)_{(3,2)} & (L_2 P)_{(3,2)} & (L_3 P)_{(3,2)} \\ (L_1 P)_{(1,3)} & (L_2 P)_{(1,3)} & (L_3 P)_{(1,3)} \\ (L_1 P)_{(2,1)} & (L_2 P)_{(2,1)} & (L_3 P)_{(2,1)} \end{bmatrix} = \begin{bmatrix} -\kappa \sin \phi & 0 & -l \sin \phi \\ \kappa \cos \phi & 0 & l \cos \phi \\ 0 & 1 & 0 \end{bmatrix} \quad (4.41)$$

Con ello, se puede obtener la Jacobiana convencional, simplemente mediante la multiplicación de ambas matrices.

$$J = \begin{bmatrix} J_{lin} \\ J_\omega \end{bmatrix} \quad (4.42)$$

Esto representa el modelo diferencial, aplicado a los parámetros de curvatura, esto es, la parte independiente del modelo para el robot. El trabajo de Chembrammel y Kesavadas [6] también utiliza otras expresiones para poder calcular la derivada de la Jacobiana. Esto sirve para poder realizar una integración numérica de la Jacobiana respecto al tiempo.

Estas expresiones no se van a desarrollar y particularizar para este modelo cinemático en concreto, pero se presentan a continuación, siendo sencillo su cálculo:

$$\dot{L} = \frac{\partial L}{\partial q_i} \quad (4.43)$$

$$\dot{J}_{lin} = \sum_{i=1}^n \frac{\partial J_{lin}}{\partial q_i} \dot{q}_i = \sum_{i=1}^n \frac{\partial L}{\partial q_i} P \dot{q}_i \quad (4.44)$$

$$\dot{J}_\omega = \sum_{i=1}^n \frac{\partial J_\omega}{\partial q_i} \dot{q}_i \quad (4.45)$$

$$\frac{\partial J_{\omega,i}}{\partial q_j} ([(L_i P)_{(3,2)} \quad (L_i P)_{(1,3)} \quad (L_i P)_{(2,1)}])^T \quad (4.46)$$

$$\frac{\partial L_i P}{\partial q_j} = \left(\frac{\partial L_i}{\partial q_j} - L_i T_{indep}^{-1} L_j \right) T_{indep}^{-1} \quad (4.47)$$

Como ya se ha indicado, se trata de un modelo diferencial parcial: solo tiene en cuenta la parte independiente del robot y por ello son expresiones completamente generales y válidas para cualquier robot que acepte la hipótesis PCC. No obstante, este método para calcular Jacobianas también es plausible en el caso de querer generar un modelo completo.

La única modificación que sería necesario es sustituir las relaciones entre los cables y los parámetros de curvatura (tales como las ecuaciones (4.21) y (4.16)) en la matriz de transformación de la sección que se ha empleado para derivar el modelo (ecuación (4.9)). Posteriormente, se deduciría el modelo de nuevo, empleando las longitudes (l_1, l_2, \dots, l_n) como las nuevas coordenadas generalizadas. Esto resultaría en un modelo diferencial completo.

4.3. LSK - Cinemática de Segmentos Linealizados

La hipótesis de curvatura constante es, como se ha podido ver, un modelo extremadamente útil a la hora de simplificar la obtención de ecuaciones cinemáticas de robots hiperredundantes. Sin embargo, también presenta una serie de limitaciones a la hora de tratar de introducirla en determinadas aplicaciones. Una de las más destacadas es el gran coste computacional que conlleva su aplicación, especialmente en aquellos casos de robots con un número elevado de secciones. Ya se ha podido comprobar en la ecuación (4.2.2) el rápido crecimiento del número de cálculos.

Por ello, la aplicación a Sistemas en Tiempo Real puede ser complejo. En función de la arquitectura hardware y software del robot, la respuesta de un modelo cinemático que emplee las ecuaciones ya vistas puede no ser suficiente. Así pues, se está trabajando para encontrar métodos alternativos que eviten estas limitaciones. Un importante ejemplo de ello es la Cinemática de Segmentos Linealizados, que también permite eliminar las singularidades del modelo PCC. [4]

Se trata de un método que deriva del modelo PCC, lo que hace que su diagrama general sea bastante parecido. De hecho, la cinemática LSK mantiene la hipótesis de curvatura constante. El principal cambio que conlleva la aplicación de la hipótesis LSK es la linealización de las ecuaciones que devuelven las longitudes de los cables. Por tanto, la modificación es específica para la transformación dependiente de la geometría del robot.

No obstante, se aplicará el mismo criterio que en los casos anteriores: dada la existencia de robots con una morfología muy semejante hace fácil encontrar las ecuaciones planteadas para muchos de los robots existentes hoy en día. En este caso, se muestra el esquema para la cinemática inversa de un robot con las mismas hipótesis que en el caso de PCC (cables homogéneamente distribuidos a la misma distancia del núcleo). No obstante, se podría aplicar el mismo procedimiento a la cinemática directa o a otras tipologías de robots.

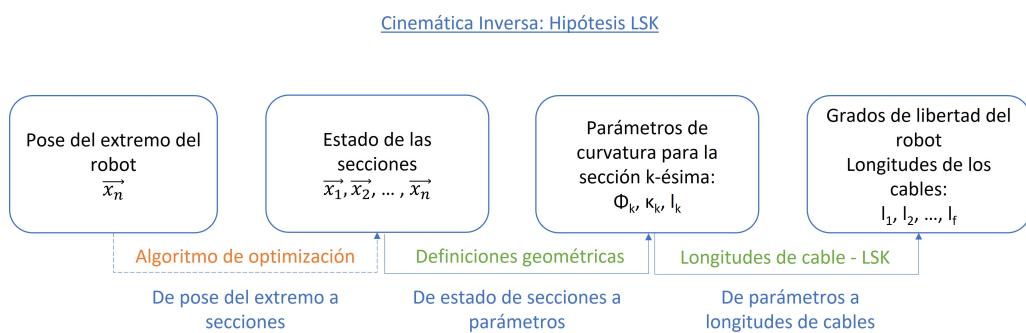


Figura 4.11: Aplicación del marco teórico inverso al modelo LSK. Elaboración propia

Así pues, este método se basa en la linealización de las ecuaciones cinemáticas para cada cable, simplificando el número de cálculos para cada sección y, por tanto, para el robot completo. Para lograrlo, se van a utilizar unas expresiones adaptadas de algunas fórmulas del modelo PCC, siguiendo el método de [4]. Se combinarán las ecuaciones (4.10), (4.13). Aplicando relaciones

trigonométricas:

$$l_i = 2 * \sin\left(\frac{\theta}{2p}\right) * \left(\frac{l_c p}{\theta} - r_c \sin(\phi + \phi_i)\right) \quad (4.48)$$

Para evitar el coste computacional de los cálculos de senos, así como para eliminar la posible singularidad provocada al dividir por cero ($\theta = 0$) y para reducir el número de computaciones necesarias cuando se incrementa el número de secciones, esta será la expresión a linealizar.

La primera pregunta que sería planteable al enfrentarse a este problema es si la linealización es necesaria o no, así como su error (toda linealización supone cometer un error frente a un modelo más realista). Para ello, se comienza calculando la longitud de la fibra neutra del robot, que se convertirá en un ejemplo para posteriores cálculos.

$$l \geq 2 * \sin\left(\frac{\theta}{2p}\right) * \left(\frac{l_c p}{\theta} - r_c \sin(\phi)\right)$$

Los cables se supone que trabajarán exclusivamente a tensiones de tracción, y que no ofrecen ninguna resistencia ante esfuerzos de compresión. De esta forma, se podría alcanzar cualquier punto que requiera menor longitud de cables, de tal manera que se fuerza la aparición de una inecuación. Posteriormente, para la linealización, se aplican las series de MacLaurin para θ (que matemáticamente son las series de Taylor aplicadas en torno al origen, es decir, $\theta = 0$). Como resultado:

$$l \approx l' \geq l_c - r_c \frac{\theta}{p} \quad (4.49)$$

Si se aplicara la linealización a todo el rango posible de ángulos que puede alcanzar el robot se verificaría como la linealización en torno a un punto (como sería, en este caso, el origen) generaría un error excesivo en las ecuaciones para puntos muy alejados respecto a aquel que se empleó en la linealización. No obstante, en la mayor parte de los robots existen varios segmentos de cable en una misma sección, debido a la utilización de discos pasivos para guiarlos.

En este caso, el ángulo total de giro se divide respecto a los discos pasivos, reduciendo así el error acumulado, que se vuelve aceptable. Barrientos y Dong, en su trabajo, demuestran para su desarrollo un error máximo de 0,7%.

Ahora se puede proceder a la obtención de las ecuaciones como tal. Para ello, se linealizan las ecuaciones para cada cable l_i y se aplican una vez más las series de MacLaurin. Con ello se obtiene la siguiente expresión:

$$l'_i \geq -p * r_c * \theta * (\cos(\phi_i) * \cos(\phi) + \sin(\phi_i) * \sin(\phi)) + l_c \quad (4.50)$$

Dado que lo que se necesita realmente es el incremento de la longitud de cable con respecto a una posición neutral, la constante puede eliminarse de la ecuación. Con ello se logra una expresión en diferencias. Además de ello, se puede observar que el sistema queda como el producto entre un determinado factor de escala $K(\theta)$, una matriz que únicamente depende de la geometría del robot $\mathbf{A}(\phi_i)$, y una serie de expresiones trigonométricas que dependen de ϕ :

$$\begin{bmatrix} \Delta l_1 \\ \Delta l_2 \\ \dots \\ \Delta l_f \end{bmatrix} \gtrsim K(\theta) \mathbf{A}(\phi_i) \begin{bmatrix} \cos(\phi_i) \\ \sin(\phi_i) \end{bmatrix} = -r_c \theta \begin{bmatrix} \cos(\phi_1) & \sin(\phi_1) \\ \cos(\phi_2) & \sin(\phi_2) \\ \dots & \dots \\ \cos(\phi_{np}) & \sin(\phi_{np}) \end{bmatrix} \begin{bmatrix} \cos(\phi_i) \\ \sin(\phi_i) \end{bmatrix} \quad (4.51)$$

Estas ecuaciones se corresponden con el modelo cinemático inverso de una única sección. Para obtener el modelo de la cinemática directa, siempre y cuando las longitudes de los cables sean coherentes con una situación real (recordemos que existen relaciones entre ellas que se deben cumplir), se puede inferir que:

$$\begin{bmatrix} \cos(\phi_i) \\ \sin(\phi_i) \end{bmatrix} \lesssim -\frac{1}{r_c \theta} \begin{bmatrix} \cos(\phi_1) & \sin(\phi_1) \\ \cos(\phi_2) & \sin(\phi_2) \\ \dots & \dots \\ \cos(\phi_{np}) & \sin(\phi_{np}) \end{bmatrix}^T \begin{bmatrix} \Delta l_1 \\ \Delta l_2 \\ \dots \\ \Delta l_f \end{bmatrix} \quad (4.52)$$

5 | MODELOS CINEMÁTICOS PARA RHAC BASADOS EN PARÁMETROS DE ELASTICIDAD

Una vez descritos y enunciados los distintos modelos que son frecuentes en la aplicación a robots hiperredundantes, ha de destacarse que los modelos exclusivamente cinemáticos no son la única posibilidad para el modelado de RHAC. La inclusión de variables de fuerzas y momentos permiten también el desarrollo de otros modelos, en los cuales para la cinemática RHAC es frecuente utilizar parámetros de la elasticidad de sus materiales.

En general, para aquellos casos en los que la hipótesis de curvatura constante no es suficiente, se suelen emplear técnicas de modelado por elasticidad para obtener las ecuaciones. Estos desarrollos parten de la información disponible sobre los esfuerzos a los que se ven sometidos los robots para obtener la información sobre la configuración de los mismos. [34] Esto es, las variables ya no parten de las longitudes de los cables, sino de las fuerzas y momentos que experimentan los robots.

En general, se tratan de modelos más precisos, pero a costa de aumentar la complejidad de las ecuaciones. Esto, a su vez, se traduce en un mayor coste computacional a la hora de resolverlos. De hecho, muchos de los modelos no pueden expresarse en una única expresión analítica, sino que se deben resolver mediante métodos numéricos.

Ya se ha mencionado en el apartado anterior que una de las posibles causas que pueden provocar que no sea aplicable la hipótesis PCC es el peso propio del robot: en estos casos, el modelo PCC puede no tener la mínima precisión necesaria como para considerar una buena representación del mismo. Cuando el robot tiene un núcleo continuo, y se dispone de sus propiedades dinámicas, incorporarlas en el modelo suelen ofrecer resultados más precisos que en el caso meramente cinemático.

En particular, aquí se recogen tres modelos de modelado a partir de las fuerzas que actúan sobre los robots que suelen ser frecuentes en el modelado de robots hiperredundantes. Dos de ellos pueden considerarse modelos como tal: se adopta una hipótesis que permite la representación del robot según ciertas ecuaciones matemáticas. No obstante, su solución puede requerir métodos numéricos o de integración. Por otra parte, el último enfoque pretende ser más preciso y realista, si bien requiere un coste computacional elevado.

El primer modelo, denominado PRB (Pseudo-Rigid Body) emplea parámetros discretos para modelar la dinámica del núcleo. [34]. Así, se trata de un conjunto finito de parámetros con los que se puede representar el robot. En cambio, el segundo modelo, denominado *Cosserat Rod Theory* (CRT) se basa en un conjunto infinito de parámetros con los que se modela de forma continua todo el núcleo del robot.

5.1. PRB: Pseudo-Rigid Body

La teoría PRB, o del cuerpo pseudo-rígido, es un modelo que permite realizar cálculos sobre grandes deformaciones en las vigas, a partir de modelarlos como una sucesión de articulaciones de rotación junto con muelles. En particular, el modelo más utilizado para obtener ecuaciones para núcleos de robots hiperredundantes es el propuesto por Su en 2009. [39]

Esta alternativa propone modelar el núcleo de los RHAC como un conjunto de cuatro segmentos rígidos y rectos dispuestos en serie, unidos entre sí por articulaciones rotativas puntuales, así como por tres muelles de torsión para poder modelar cada porción de robot que transcurre entre dos discos consecutivos (ya sean activos o pasivos). Nótese que, al tratarse de un modelo dinámico, utiliza las fuerzas como parámetros de entrada. Por otra parte, dado que se parametriza el núcleo a partir de sus propiedades dinámicas, se orienta particularmente a robots con un núcleo continuo, de un material homogéneo, al menos entre discos.

Para este modelo se va a analizar en concreto el modelo directo. Se asumirá que en estos casos, lo que interesa es conocidas las fuerzas que actúan sobre el robot, cuál es la posición o configuración en la que este termina.

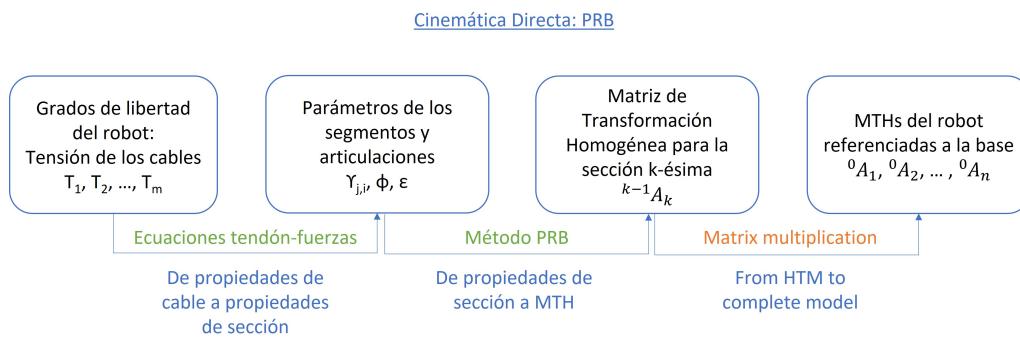


Figura 5.1: Aplicación del marco teórico directo al modelo PRB. Elaboración propia

Como se ha indicado, el modelo se basa en suponer cada sector del robot localizado entre dos discos como un conjunto de segmentos rígidos, sin posibilidad de deformación (en naranja) y articulaciones (en azul), tal y como se observa en la figura 5.2

γ_0	γ_1	γ_2	γ_3
0,125	0,35	0,388	0,136

Cuadro 5.1: Parámetros γ_q para los segmentos rectos de una sección PRB

El modelo propone el empleo de cuatro parámetros q_j que ya se han optimizado en la bibliografía [7]. Puede encontrarse en la tabla 5.1 los valores optimizados para cada uno de ellos. Éstos representan el ratio entre la longitud de cada uno de los segmentos individuales frente a la longitud total de la sección. Ahora, si el ángulo $\theta_{k,i}$ se define como el ángulo del segmento i en la sección k , la matriz de transformación correspondiente en el plano de la sección k se obtiene como:

$$T_{k,i} = \begin{bmatrix} \cos(\theta_{k,i}) & 0 & \sin(\theta_{k,i}) & \gamma_{k,i} l_j \sin(\theta_{k,i}) \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_{k,i}) & 0 & \cos(\theta_{k,i}) & \gamma_{k,i} l_j \cos(\theta_{k,i}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

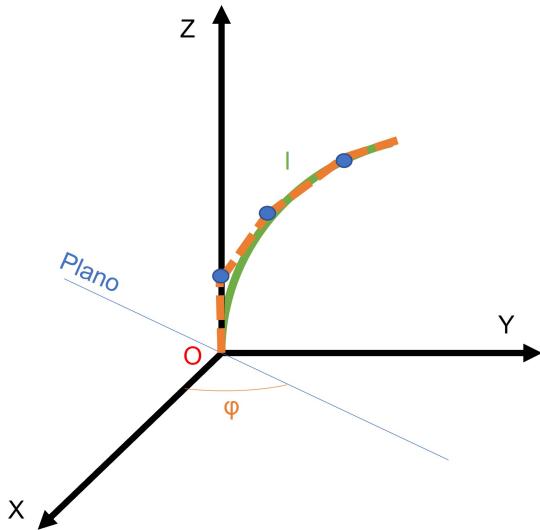


Figura 5.2: Diagrama conceptual para el modelo PRB. Elaboración propia

Así pues, la sección se modela finalmente como una combinación de varios movimientos. En primer lugar, una rotación en torno al eje z , que se denominará R_z . Posteriormente, deben acumularse las transformaciones que se dan por la sucesión de segmentos rígidos. Finalmente, se debe deshacer la primera rotación, para devolver el sistema de coordenadas a la orientación inicial. Ahora bien, puede añadirse un último movimiento, en el caso de que se quieran modelizar también posibles torsiones.

$${}^k A_{k+1} = R_z(\phi_j) T_{k,0} T_{k,1} T_{k,2} T_{k,3} R_z(-\phi_j) R_z(\epsilon) \quad (5.2)$$

Es el momento de realizar varias consideraciones que deberían tenerse en cuenta. Cuando Huang et al. propusieron este modelo, solo se consideraban robots de una única sección (entendiendo sección en este apartado como porción de robot entre dos discos, independientemente de que sean activos o pasivos), que además no estuvieran sometidos a fuerzas exteriores.[16] No obstante, Rao et al. propusieron un modelo modificado, incorporando algunas de estas posibilidades. Por ser este el modelo más completo, se proseguirá con su procedimiento de modelización.

En primer lugar, y denominando $\tau_{j,i}$ como la tensión existente en uno de los cables sobre la sección i , se puede calcular la fuerza total que ejerce el cable j sobre el último disco del robot n . Esto se logra empleando el vector $\overrightarrow{P_{j,n}P_{j,n-1}}$ que se da entre el origen del cable en la sección para dos discos consecutivos.

$$\vec{F}_{j,n} = \tau_{j,n} \frac{\overrightarrow{P_{j,n}P_{j,n-1}}}{\| \overrightarrow{P_{j,n}P_{j,n-1}} \|} \quad (5.3)$$

Para otros discos i , la ecuación se presenta finalmente como:

$$\vec{F}_{j,i} = \tau_{j,i} \frac{\overrightarrow{P_{j,i}P_{j,i-1}}}{\| \overrightarrow{P_{j,i}P_{j,i-1}} \|} + \tau_{j,i+1} \frac{\overrightarrow{P_{j,i+1}P_{j,i}}}{\| \overrightarrow{P_{j,i+1}P_{j,i}} \|} \quad (5.4)$$

La fuerza total que se ejerce sobre un determinado disco se obtiene así al sumar todas las fuerzas externas (que se supondrán ejercidas sobre la punta del robot, por ejemplo, cuando sujetes un determinado peso; por tanto, se considerará para el último disco). Denominando n al último disco, siendo i un disco genérico y f el número total de cables en el robot:

$$\vec{F}_n^T = \sum_{j=1}^f \vec{F}_{j,n} + \vec{F}_{ext} \quad (5.5)$$

$$\vec{F}_i^T = \sum_{j=1}^f \vec{F}_{j,i} + \vec{F}_{i+1} \quad i < n \quad (5.6)$$

A partir de la información aportada por las fuerzas se pueden calcular los momentos involucrados en el sistema. Se denominará $\overrightarrow{O_i O_{q,i}}$ al vector que une la base del disco i con la articulación teórica generada por el modelo PRB j . $\overrightarrow{M_{i,q}}$ será el momento total aplicado al disco i por la articulación j . Con ello, siendo $\overrightarrow{M_{j,i}^c}$ la tensión que el cable j aplica al disco i :

$$\vec{M}_{n,q} = \sum_{j=1}^f \overrightarrow{M_{n,j}^c} + \overrightarrow{O_n O_{q,n}} \times \vec{F}_n^T \quad (5.7)$$

$$\vec{M}_{i,q} = \underline{j} = 1^f \overrightarrow{M_{i,j}^c} + \overrightarrow{O_i O_{q,i}} \times \vec{F}_i^T + \vec{M}_{i+1,q} \quad i < n \quad (5.8)$$

El momento total de la sección \vec{M}_i^T se calcula como:

$$\vec{M}_n^T = \sum_{j=1}^f \overrightarrow{M_{n,j}} + \overrightarrow{O_{n-1} O_n} \times \vec{F}_n^T \quad (5.9)$$

$$\vec{M}_i^T = \sum_{j=1}^f \overrightarrow{M_{i,j}} + \overrightarrow{O_{i-1} O_i} \times \vec{F}_i^T + \vec{M}_{i+1}^T \quad i < n \quad (5.10)$$

Tomando este vector, el ángulo ϕ se puede calcular conocido que el vector \vec{y}_i , perpendicular al plano que contiene la curva, es colineal con el vector $\vec{M}_i^T - \vec{M}_j^T \cdot \vec{z}_i$. Una vez este ángulo se ha calculado, únicamente faltan los datos de los ángulos de las articulaciones para poder tener un modelo completo.

Para obtener los ángulos que ha girado cada articulación, se emplea la relación:

$$||\vec{M}_i^T - \vec{M}_i^T \cdot \vec{z}_i|| = K_{\Theta_q} \frac{EI}{l_i} \theta_q \quad (5.11)$$

Siendo E el módulo de Young del material, I el momento de inercia, l la longitud de la sección, y K_{Θ_q} un parámetro del modelo PRB. En concreto, este parámetro también se ha optimizado en la bibliografía, para asegurar un mejor comportamiento del modelo. Los valores optimizados se pueden encontrar en la siguiente tabla:

K_{Θ_1}	K_{Θ_2}	K_{Θ_3}
3.25	2.84	2.95

Cuadro 5.2: Parámetros K_{Θ_q} optimizados

Se puede obtener una última ecuación que dará lugar al parámetro de torsión ϵ , permitiendo la influencia de fuerzas externas y un número arbitrario de segmentos, casos en los que aparecen este tipo de esfuerzos:

$$GJ\epsilon_i = l_i(i-1) \vec{T}_i \vec{M}_i \vec{z}_B \quad (5.12)$$

El conjunto de estas ecuaciones conforma un modelo en sí mismo. Puede comprobarse que existen cinco variables cinemáticas independientes para cada sección $(\theta_{1,i}, \theta_{2,i}, \theta_{3,i}, \phi_i, \epsilon_i)$, y otras

tres variables dinámicas independientes por sección ($M_{x,i}^T, M_{y,i}^T, M_{z-i}^T$). Por otra parte, se disponen de 8 ecuaciones disponibles por sección:

- 3 ecuaciones en la expresión vectorial (5.10), donde el resto de variables son dependientes de las variables cinemáticas o conocidos.
- Otras 3 ecuaciones disponibles en (5.11), una para cada ángulo de articulación.
- 1 ecuación para la condición de colinealidad entre \vec{y}_i e $\vec{M_i^f} - \vec{M_j^f} \cdot \vec{z}_i$
- Finalmente, 1 ecuación para 5.12

No obstante, no se conoce una forma analítica para desacoplar estas ecuaciones. Esto supone que las ecuaciones tienen que resolverse simultáneamente. Es más, existe un acople entre distintas secciones, por la recurrencia que se genera en las fuerzas entre secciones. Esta interdependencia, sumada a la fuerte no linealidad del conjunto de ecuaciones, fuerzan al usuario a emplear métodos numéricos para resolver el modelo PRB. [44]

5.2. Teoría elástica de Cosserat - CRT (Cosserat Rod Theory)

La teoría elástica de Cosserat, más conocida en inglés como *Cosserat Rod Theory* o CRT, es una de las técnicas que se emplea con mayor frecuencia para modelar vigas continuas. En los robots hiperredundantes continuos puede asemejarse el núcleo de un material elástico con una viga deformable, pudiendo por tanto emplear esta teoría para modelar la deformación que experimenta el robot ante distintas fuerzas.

La teoría se basa en asignar 6 grados de libertad a cada punto del núcleo del robot, mientras que establece ciertas condiciones de contorno que debe cumplir, a partir de las ecuaciones dinámicas. [2]. Esta técnica se ha aplicado con éxito en distintos problemas cinemáticos y para distintas configuraciones de robots [17, 10, 35]. Todos estos trabajos presentan modelos ligeramente modificados, según las hipótesis que se aplican (robots restringidos a un plano, consideración de la fuerza gravitatoria)...

En particular, la teoría de Cosserat proporciona la relación existente entre las fuerzas (o esfuerzos) aplicados sobre el robot y la posición de cada uno de los puntos de su núcleo. Una vez más, que las fuerzas constituyan la entrada a las ecuaciones muestra que se está trabajando con un modelo dinámico. En particular, dadas las diferencias entre modelos ya mencionadas, en este trabajo se va a emplear el desarrollo de Jones [17].

Partiendo de la figura 5.3, se pueden definir la posición y la orientación de la sección i a partir de un vector de posición $\vec{p}(i)$ y una matriz de rotación ${}^{i-1}A_i$. Si se define $\vec{v}(i)$ como la velocidad lineal de una rebanada infinitesimal del núcleo en el extremo de la sección i referido al sistema de referencia de la propia sección, y $\vec{u}(i)$ como la rotación infinitesimal del sistema de referencia ${}^{i-1}A_i$, entonces las derivadas tienen la forma:

$$\dot{\vec{p}} = {}^{i-1}A_i \cdot \vec{v} \quad (5.13)$$

$${}^{i-1}A_i = {}^{i-1}A_i \cdot \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \quad (5.14)$$

Observando la figura 5.3 pueden introducirse 6 variables adicionales, que se emplearán para definir el modelo:

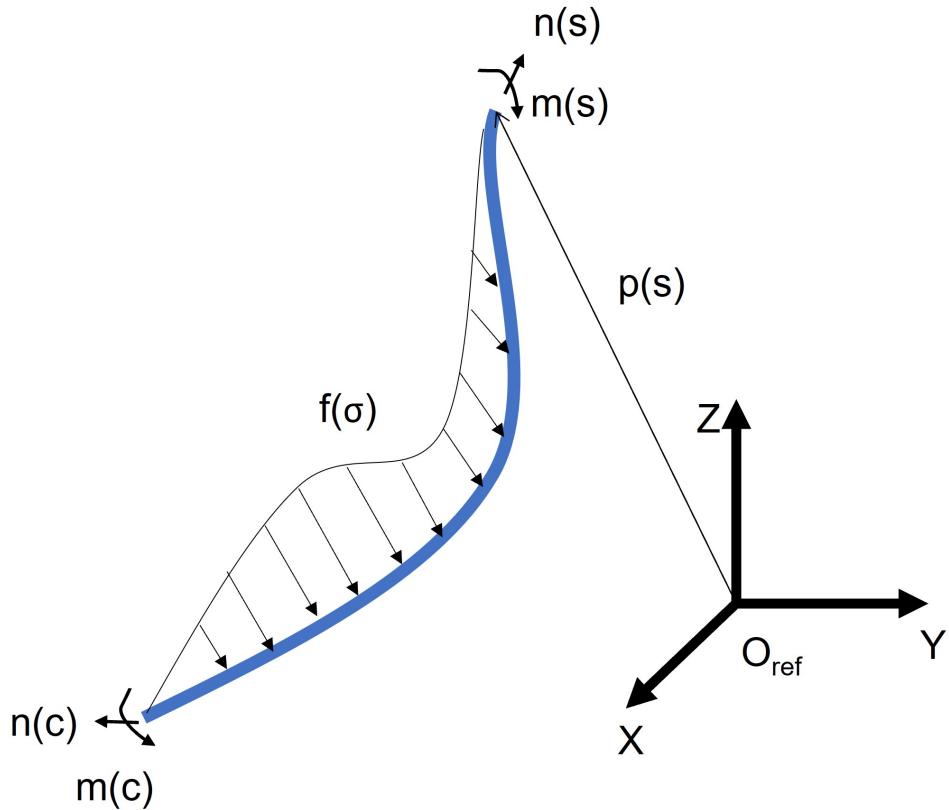


Figura 5.3: Sección teórica de una viga (o núcleo del robot) para explicar la CRT. Elaboración propia

- \vec{v}^* y $\vec{\omega}^*$ representan la velocidad lineal y angular del núcleo en su estado original, esto es, sin deformación.
- $\vec{m}(i)$ y $\vec{n}(i)$ representan los vectores de momento interno y fuerzas internas del núcleo.
- \vec{f} es la distribución de fuerzas que se aplican sobre el robot.
- \vec{l} es la distribución de momentos ejercidos sobre el robot.

Empleando esta nomenclatura, se pueden aplicar las condiciones para el equilibrio. Utilizando $\vec{m}(i)$ e $\vec{n}(i)$ como las variables de estado del sistema:

$$\dot{\vec{n}} = -\vec{f} \quad (5.15)$$

$$\dot{\vec{m}} = -\vec{p} \times \vec{n} - \vec{l} \quad (5.16)$$

Finalmente, es necesario introducir también la caracterización dinámica del material:

$$K_{se}(i) = \begin{bmatrix} GA(i) & 0 & 0 \\ 0 & GA(i) & 0 \\ 0 & 0 & EA(i) \end{bmatrix} \quad K_{bt}(i) = \begin{bmatrix} EI_{xx}(i) & 0 & 0 \\ 0 & EI_{yy}(i) & 0 \\ 0 & 0 & EI_{zz}(i) \end{bmatrix} \quad (5.17)$$

Para obtener un modelo concreto, se introducen estas matrices en el sistema de ecuaciones anterior a partir de las leyes constitutivas del material del núcleo:

$$\vec{v} = K_{se}^{-1} \vec{n} + \vec{v}^* \quad (5.18)$$

$$\vec{u} = K_{bt}^{-1} \vec{m} + \vec{u}^* \quad (5.19)$$

Puede ya definirse el modelo concreto de la Teoría de Cosserat, sin introducir aún ningún elemento de robótica. El modelo completo se corresponde con el conjunto formado por las ecuaciones (5.13) (5.14) (5.15) (5.16) (5.18) y (5.19).

Ahora bien, es posible particularizar algo más el modelo para el caso que ocupa a este trabajo: el modelado de robots hiperredundantes actuados por cables. Para ello, es necesario acoplar también las ecuaciones para los cables en el modelo, considerándolos también como cuerpos que responden a la teoría elástica de Cosserat. [35] Para ello, ha de tenerse en cuenta que tanto las fuerzas como los momentos tienen una componente externa (\vec{f}_e , \vec{l}_e) y una componente provocada por los propios tendones (\vec{f}_t , \vec{l}_t).

El desarrollo de las expresiones para las fuerzas y momentos en función de la tensión de los cables (τ_j) es complejo, y emplea numerosas variables auxiliares y coeficientes que dificultan la exposición clara en esta síntesis. Por ello, no se desarrollará completamente en este trabajo, extrayendo únicamente las conclusiones. En ese sentido, para expresar la solución, se define el operador \hat{w} como:

$$\hat{w} = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \quad (5.20)$$

Utilizando esta herramienta se pueden expresar las ecuaciones que relacionan las fuerzas y los tendones. Se ha de tener cuidado en no confundir la notación: se emplea el escalar f como en todo el trabajo para expresar el número total de cables que tiene el robot, mientras que la magnitud vectorial \vec{f} representa las fuerzas que intervienen. Así, definiendo \vec{r}_j como el vector que va desde el origen del sistema de referencia fijo en la base al punto de origen del cable, se tiene que:

$$\vec{f}_t = - \sum_{j=1}^f \tau_j \frac{\hat{\vec{p}}_j^2}{\|\vec{p}_j\|^3} \vec{p}_j \quad (5.21)$$

$$\vec{l}_t = - \sum_{j=1}^f \tau_j ({}^{i-1}A_i \vec{r}_j) \frac{\hat{\vec{p}}_j^2}{\|\vec{p}_j\|^3} \vec{p}_j \quad (5.22)$$

Estas ecuaciones emplean las variables, que se pueden obtener como función de las variables cinemáticas que intervienen en el modelo:

$$\dot{\vec{p}}_j = {}^{i-1}A_i (\hat{\vec{u}} \vec{r}_j + \dot{\vec{r}}_j + \vec{v}) \quad (5.23)$$

$$\ddot{\vec{p}}_j = {}^{i-1}A_i (\hat{\vec{u}} (\hat{\vec{u}} \vec{r}_j + \dot{\vec{r}}_j + \vec{v}) + \hat{\vec{u}} \dot{\vec{r}}_j + \hat{\vec{u}} \dot{\vec{r}}_j + \ddot{\vec{r}}_j + \dot{\vec{v}}) \quad (5.24)$$

Esto constituye un modelo implícito para los RHAC. No se trata de un modelo analítico que pueda resolverse. En concreto, en el artículo se continúa hasta obtener un sistema de ecuaciones de estado con el que se podría resolver numéricamente el modelo. Así, se requiere de una herramienta de cálculo numérico para poder emplear el modelo CRT.

Esta sección deja clara la complejidad de uso de los modelos dinámicos: se obtienen ecuaciones fuertemente no lineales, que en el caso del CRT necesita ser integrada numéricamente por ordenador. No se dispone de una solución analítica como en los modelos puramente cinemáticos, además de necesitarse un conocimiento exacto de las fuerzas y momentos a los que se ve sometido el robot.

Esta es la razón que justifica que los modelos cinemáticos, aún con sus deficiencias, sigan siendo la alternativa dominante para los robots continuos no blandos. Son modelos de aplicación más sencilla y que requieren una capacidad de cómputo muy inferior.

5.3. Método de Elementos Finitos (FEM)

Los modelos ya mencionados son útiles de cara al modelado de robots continuos, si bien los robots blandos muchas veces tienen materiales muy complejos de modelar, por sus propiedades heterogéneas, que impiden poder generalizar las propiedades del material a todo el núcleo (base de los modelos PRB y CRT). Ante esta realidad existen dos alternativas distintas: la primera es emplear la inteligencia artificial para tratar de encontrar una relación estadística entre los grados de libertad actuables y la configuración del robot en cada momento, a partir de datos recogidos experimentalmente. Lógicamente, se trata de un modelo caja negra, en el que no se conocen las relaciones reales que subyacen en el robot. [24]

No obstante, existe otra alternativa de mayor interés y que también tiene una importante aceptación en el mundo académico actualmente. Se trata del empleo de métodos numéricos, concretamente a partir de la técnica del método de elementos finitos. Ya en 2013 se planteaba el empleo de este algoritmo numérico a través del software SOFA (Simulation Open Framework Architecture) para modelar robots blandos, permitiendo además su empleo en condiciones de Tiempo Real. [11]

En concreto, la técnica de FEM se basa en generar una malla con la forma del robot, formada por un número finito de elementos a los que aplicar una ecuación diferencial individualmente. El método numérico busca la solución de tal manera que se cumplan todas las condiciones de contorno definidas por la interacción entre los distintos elementos que conforman la malla. [12]

En los robots blandos la malla la determinará la forma del robot en cada momento. El número de elementos que conforma la malla es uno de los parámetros esenciales del método. A mayor número de discretizaciones que se realicen, más precisión tendrá el modelo, pero crecerá notablemente el coste computacional de encontrar la solución del mismo. [24]

La aplicación del método FEM frente a las otras alternativas presentadas en este trabajo es clara: el FEM comienza la simulación utilizando la ley constitutiva del material (medida experimentalmente) para calcular la matriz de elasticidad. No obstante, entra en juego un algoritmo iterativo a partir de las restricciones del proceso que, junto a la matriz de elasticidad, es capaz de caracterizar con mayor precisión la relación entre actuadores y configuración del robot.

En SOFA, combinado con el plugin de Soft Robots, es una alternativa muy empleada en la investigación actual con robots blandos. La ecuación diferencial que el sistema utiliza en cada uno de los elementos de la malla es la segunda ley de Newton: [9]

$$M(q) \cdot \ddot{v} = P(t) - F(q, v) + H^T \lambda \quad (5.25)$$

siendo q las coordenadas generalizadas del robot, M la matriz de inercia, v es la velocidad, F son las fuerzas internas que se dan en el robot, H^T es una matriz donde se indican las direcciones

de las restricciones que se aplican al robot y λ es el vector de multiplicadores de Lagrange, que representan las intensidades de las fuerzas de restricción.

Durante la ejecución, el sistema integra esta ecuación mediante un algoritmo de Euler implícito como método de integración numérica. En concreto, tras la aplicación del método de Euler y la linealización de las ecuaciones, la expresión resultante es:

$$\frac{\delta F}{\delta q} dq = P - f_i + H^T \lambda \quad (5.26)$$

Se puede establecer un paralelismo entre esta expresión y un sistema lineal, con un sumando adicional: $A\dot{x} = b + H^T \lambda$. Nótese que en esa ecuación existen dos incógnitas: la derivada de la configuración del robot (que será la que sirva para determinar la posición del robot en cada momento), y la intensidad de las fuerzas de ligadura.

Así, la integración de las ecuaciones se da en dos pasos. En primer lugar se calcula la configuración del robot suponiendo que no existen fuerzas de ligadura (encontrando así la configuración q_{free} para todos los elementos de la malla). En un segundo paso se ejecuta un proceso de optimización, de tal manera que se obtiene como resultado la intensidad para cada fuerza de ligadura (el vector λ). [9]

SOFA, además, incorpora cuatro capas en el modelo. La primera está definida por la malla de elementos finitos que utiliza el programa para resolver las ecuaciones; en ese sentido dependerá de la configuración del usuario. La segunda capa se corresponde con la capa de actuación, que contiene los posibles actuadores del robot. La tercera define las posibles zonas de colisión para el robot, permitiendo la interacción consigo mismo y con su entorno. Finalmente, se encuentra la capa de visualización. [24]

Como se puede observar, es un método potente, y que ha llegado a emplearse incluso a tiempo real con robots blandos, si bien para llegar a respuestas tan rápidas se necesitan métodos que optimicen el número de elementos a calcular. En caso contrario, el coste computacional de emplear este modelo es demasiado elevado.

6 | ALGORITMO DE OPTIMIZACIÓN: NATURAL - CCD

Los dos capítulos anteriores han introducido un conjunto de modelos, tanto basados únicamente en parámetros cinemáticos como basados también en información de la dinámica del robot, que son los más frecuentes en la bibliografía académica para modelar el comportamiento de robots hiperredundantes. Cada uno de ellos tiene una serie de hipótesis asociada que simplifican las ecuaciones del robot de una forma determinada.

Ahora bien tal y como se analizó en los fundamentos para el modelado de los RHAC (sección 3.2), se han definido dos pasos concretos que son completamente generales para cualquier robot que se desee modelar. En primer lugar, el paso de modelos particulares para cada sección, modelados en este TFM a partir de matrices de transformación homogéneas, a la obtención de posibles modelos generales para la configuración del robot. Esto se logra referenciando la posición de cada sección a la base de referencia global del robot mediante la multiplicación de las distintas MTH.

El segundo paso se deriva de la convención adoptada en la sección 3.3 por la cual se consideraría el modelo cinemático inverso como el proceso completo de ecuaciones que obtienen a partir del punto final al que se desea llevar el extremo del robot toda la configuración del mismo. Se justificó en su momento la introducción del paso de punto para el extremo del robot a puntos finales de cada una de las secciones por completitud: si en algún caso se tuviera una configuración a la que se desease llevar el robot, únicamente haría falta obviar ese primer paso del modelo.

Así pues, en este capítulo se va a desarrollar una alternativa para este paso general. Tras plantear brevemente el problema, se mencionarán varias posibilidades, para quedarse finalmente con el algoritmo de optimización escogido para el desarrollo de este TFM, por eficiencia y por experiencia de uso en el grupo de investigación: el natural-CCD.

6.1. Planteamiento del problema

Considérese pues, el objetivo para este capítulo. Dado un robot con n secciones, y un punto al que se desea llevar el extremo del robot $[x \ y \ z]_e^T$, el problema es encontrar un conjunto de configuraciones para las distintas secciones del robot que permitan alcanzar este punto final.

Para ello, el primer paso necesario es definir un modelo sencillo del robot, por el cual se pueda disponer de información simplificada sobre los estados alcanzables para cada sección (y por tanto, del robot en su conjunto), pero sin necesitar los cálculos de alto coste computacional que supone modelar el robot completo. Para ello, la solución típica se basa en definir una serie de articulaciones ficticias para cada sección que puedan imitar de forma lo más cercana posible al robot real.

Cualquiera de los modelos expresados anteriormente pueden trabajar en el modelo cinemático inverso disponiendo de la información de la pose (posición y orientación) de cada una de las secciones. De hecho, esto es lo que se reflejaba como estado de las secciones en la figura 3.3. Posteriormente, se aplicarán una serie de ecuaciones propias de cada uno de los modelos para obtener los parámetros particulares con los que funciona.

Así pues, definir una articulación ficticia para cada sección es una fórmula que permite, sin cálculos excesivamente complejos, modelar el comportamiento de cada una de las secciones. Cada rótula, además, estará unida a su predecesora y su sucesora mediante líneas rectas, por simplicidad. Lo único que se necesitará, pues, es la posición y la orientación del sistema de referencia que se defina para modelar la sección.

Ahora bien, pueden plantearse varias maneras de definir estas articulaciones virtuales, que influirán de forma determinante para la posterior aplicación al robot. Recuérdese que se definirá un sistema de referencia que servirá para representar el extremo de cada sección. La definición de este sistema de referencia y el giro de la articulación generarán, de forma automática, un espacio alcanzable que debe ser coherente con la realidad física del robot.

Esto se puede ver claramente con un modelo bidimensional (por simplicidad) de el robot como una sucesión de articulación-segmento recto. Si se realiza una definición completamente arbitraria, puede darse el caso de que un punto alcanzable en el modelo no sea realmente alcanzable por el robot y viceversa. En la figura 6.1 se puede observar dos casos con segmentos de longitud constante que pueden servir para exemplificar estas diferencias.

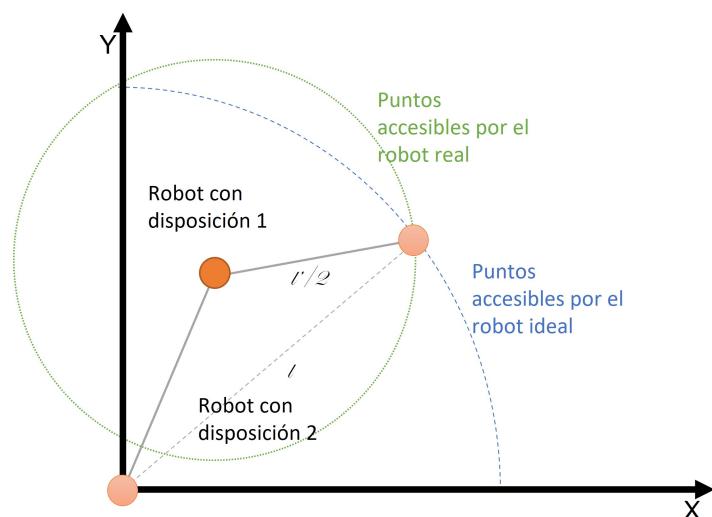


Figura 6.1: Definición de las articulaciones virtuales para cada sección

Con esto en mente, se puede comenzar a plantear el modelado de tipos de robots concretos. Por ejemplo, se puede tomar un robot discreto. En este caso, el robot es manejado por articulaciones discretas, de forma que lo que tiene más sentido es definir las articulaciones para el algoritmo con la misma disposición física que en la realidad. La mayor parte de los robots hiperredundantes tienen las articulaciones a mitad de la sección, teniendo un tramo fijo completamente recto de articulación a articulación.

De esta forma, el sistema de referencia deseado para señalizar cada una de las secciones se puede definir a mitad del segmento recto. Un ejemplo concreto de esta disposición en concreto se puede encontrar en la figura 6.2. Los rectángulos verdes representan el final de la sección, donde se definirá un sistema de referencia con el eje z paralelo al segmento recto. Los círculos naranjas representan, en cambio la posición de las articulaciones.

El modelo en particular puede obtenerse con la única información de la posición de las

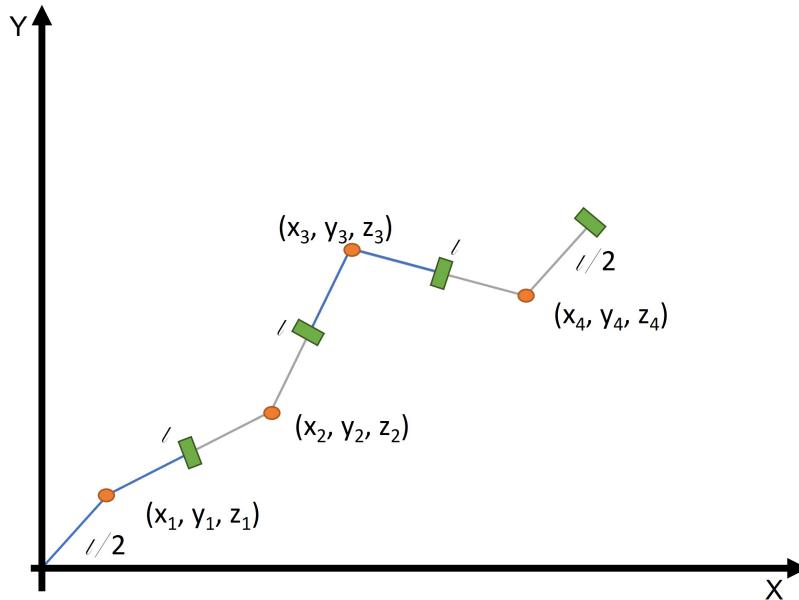


Figura 6.2: Ejemplo de definición de referencias para el robot discreto

articulaciones. Denominando a estas posiciones de las articulaciones $[x_k \ y_k \ z_k]^t$ y para el extremo de la sección k $[x_{ek} \ y_{ek} \ z_{ek}]^t$, la ecuación (6.1) representa este cálculo:

$$\begin{bmatrix} x_{ek} \\ y_{ek} \\ z_{ek} \end{bmatrix} = \begin{bmatrix} \frac{x_{k+1}-x_k}{2} \\ \frac{y_{k+1}-y_k}{2} \\ \frac{z_{k+1}-z_k}{2} \end{bmatrix} \quad (6.1)$$

Los robots continuos deben abordarse utilizando otra técnica. Debe tenerse en cuenta que sus relaciones y movimientos en el espacio son mucho más complejos. Por ejemplo, en la hipótesis PCC no es la distancia final de sección-articulación la que se mantiene constante; es el núcleo del robot el que no modifica su longitud. Por ello, a pesar de que se puede mantener la modelización con articulaciones esféricas, deben incorporarse algunas modificaciones (un grado de libertad en la longitud de los segmentos, con ciertas restricciones). Cada caso para los robots continuos debe ser considerado en particular.

Una vez definido el modelo del robot que se va a utilizar para resolver este aspecto de la cinemática inversa, solo queda determinar el procedimiento concreto para abordarlo. Se trata de intentar aproximar el extremo del robot modelo (mucho más sencillo que el original, o el creado por cualquiera de los modelos cinemáticos para RHAC) lo máximo posible al punto deseado. Enunciado de forma matemática, es un problema de optimización: se desea encontrar la configuración articular que permite minimizar la distancia euclídea del extremo del robot respecto al punto definido.

Es de sobra conocido que la resolución de un problema de optimización se puede realizar con muchos enfoques distintos. Existen muchas herramientas que sirven precisamente para ello. A pesar de la modelización realizada, la obtención de las ecuaciones del modelo particular, su derivación, y la obtención de una solución analítica es, a priori, un método demasiado complejo como para ser eficiente (la hiperredundancia complica las ecuaciones). Por ello, suele recurirse a otras alternativas.

En la bibliografía sobre robots hiperredundantes (y en cualquier ejemplo de problema de

optimización) se pueden encontrar diversas herramientas. No es poco frecuente implementar alguna solución de inteligencia artificial, y alimentarla a partir de datos simulados del modelo para resolver este problema: hoy en día la programación y el entrenamiento de una red neuronal o un algoritmo genético es una posibilidad para nada descartable. También existen distintos métodos numéricos que se pueden emplear, así como el enfoque de la Jacobiana Pseudo-inversa. El caso es minimizar la distancia euclídea previamente definida.

La elección en particular del algoritmo puede depender de múltiples factores. Además de la experiencia y las preferencias personales del responsable del modelo cinemático inverso, pueden tenerse en cuenta otras variables como la capacidad computacional disponible para el algoritmo o los requisitos de tiempos de respuesta. [33, 10, 41]

En particular, en este trabajo se propone el empleo de un algoritmo Natural-CCD, un método iterativo que permite optimizar este tipo de problemas. Éste es una mejora respecto al modelo CCD, cuyas siglas significan *Cyclic Coordinate Descent*. Éste, sin utilizar derivadas de alto coste computacional, es capaz de encontrar un mínimo local para una función basándose en el fenómeno de que, al minimizar utilizando coordinadas generalizadas una por una en cada iteración, la acumulación de éstas supone una minimización global de la función. [41]. En este caso concreto, las coordenadas generalizadas corresponden a los grados de libertad de cada una de las articulaciones que se utilizan para modelar el robot.

La justificación de emplear este algoritmo incluye varios elementos. En primer lugar se trata de un algoritmo probado en el departamento, siendo en concreto la mejora del algoritmo CCD para dar lugar al Natural-CCD de procedencia del grupo de investigación. [22] Asimismo, es un método numérico cuyo funcionamiento es comprensible y muy sencillo, siendo por tanto interesante, al menos su conocimiento, frente a soluciones de tipo más caja negra como podría ser la inteligencia artificial.

Finalmente, se trata de un método de una eficiencia elevada, y de una muy buena precisión. En comparación con otros métodos funciona con muy buenos resultados. Su simplicidad algorítmica y su bajo coste computacional permite que, en concreto, sea implementado para solucionar problemas de sistemas en tiempo real (RTS) [22]. Por todo ello, incorporando el algoritmo Natural-CCD al marco teórico desarrollado en este trabajo, el esquema del modelo cinemático inverso quedaría como se indica en la figura 6.3.

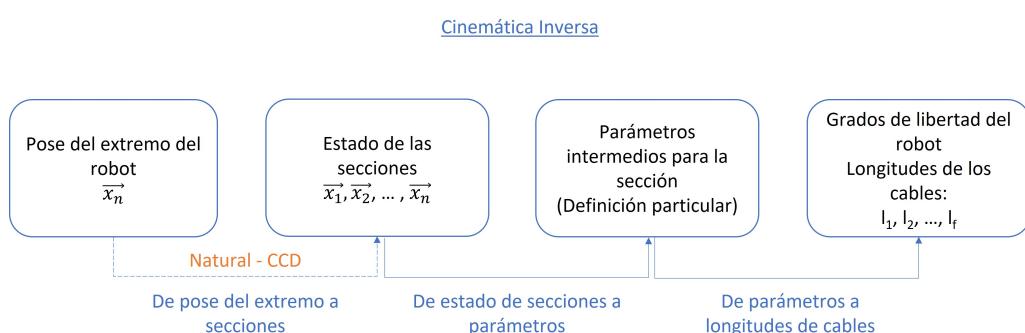


Figura 6.3: Inverse Kinematics using CCD

6.2. Desarrollo del algoritmo

6.2.1. Algoritmo CCD

Una vez definida plenamente la cadena cinemática que servirá como modelo teórico del robot se puede comenzar a desarrollar el algoritmo CCD. Para ello se definen tres puntos: p_c en la articulación concreta que se va a mover en una determinada iteración; p_e se definirá en el extremo del robot y p_f será el punto final, el objetivo que se ha definido como entrada para el modelo cinemático completo.

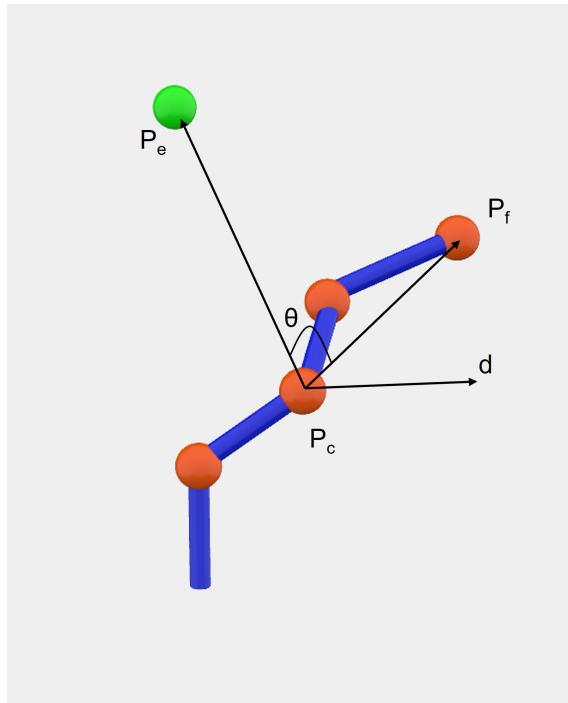


Figura 6.4: Definición de los vectores necesarios para el algoritmo CCD

A partir de estos tres vectores se comienza la optimización. Está claro que para minimizar la distancia euclídea $\|\overrightarrow{p_e p_f}\|$ para una articulación seleccionada, se debe tender a hacer coincidir en dirección y sentido a $\overrightarrow{p_c p_e}$ y $\overrightarrow{p_c p_f}$. La forma de representar esta rotación será mediante la combinación dirección - módulo. Para ello, se necesitan un ángulo θ y un vector \vec{d} que servirá para representar al eje de rotación. La expresión de estos dos valores se obtiene como:

$$\theta = \cos^{-1} \left(\frac{p_e - p_c}{\|p_c p_e\|} * \frac{p_f - p_c}{\|p_c p_f\|} \right) \quad (6.2)$$

$$\vec{d} = \frac{\vec{p}_e - \vec{p}_c}{\|\vec{p}_c \vec{p}_e\|} * \frac{\vec{p}_f - \vec{p}_c}{\|\vec{p}_c \vec{p}_f\|} \quad (6.3)$$

A partir de estas ecuaciones aquí planteadas se desarrolla el algoritmo iterativo. Se define un bucle hasta que se alcance un error menor que el definido por el usuario para la distancia euclídea entre el extremo del robot y el punto deseado. Dentro de ese bucle, se van aplicando (6.2) y (6.3) sucesivamente a cada articulación $i = 1, 2, \dots, n$ para obtener los giros correspondientes representados por los ángulos θ_i y los ejes de los mismos d_i .

Debe tenerse en cuenta que el CCD no acaba aquí. Los giros acumulados acabarán dando una θ_{fi} final para cada articulación, así como el eje de giro total. Ahora bien, estos no son parámetros que se puedan introducir en las ecuaciones previamente deducidas. Es necesario obtener algunas

relaciones geométricas que, dada la hipótesis de que las articulaciones se unen por segmentos rectos, encuentren las posiciones de los finales de sección, parámetros sí contemplados en el modelo cinemático (ver figura 6.3). Posteriormente se pueden introducir las ecuaciones del modelo correspondiente al tipo de robot concreto.

De esta forma se obtiene un algoritmo muy sencillo, computacionalmente eficiente y preciso. Además, como se observa, es de fácil aplicación a cualquier tipo de RHAC, requiriendo muy pocas adaptaciones para cada caso concreto. Así se evitan la necesidad de modelos concretos para cada tipo de robot (como se necesitarían para algunos algoritmos de IA como redes neuronales o genéticos; un entrenamiento particularizado para cada clase de robot).

6.2.2. Modificación: Natural-CCD

No obstante, este método tal y como se ha presentado en el apartado anterior, también tiene una serie de desventajas que deben ser tenidas en cuenta. [22] En primer lugar, no se ha discutido nada sobre las singularidades cinemáticas que pueden aparecer debido al modelo y al diseño del robot. El CCD no puede tenerlas en cuenta para sus cómputos. Es más, tampoco incluye un detector de colisiones que pueda prevenir choques con otros objetos o consigo mismo.

Estos impedimentos podrían resultar en una situación muy inconveniente: desarrollar una planificación para una configuración del robot que en verdad no sea alcanzable. Existe una última desventaja que también se debe considerar: operar en cada giro con valores muy elevados de θ puede ser muy demandante para las articulaciones físicas del robot.

Con el fin de solucionar, o como mínimo, mitigar estos inconvenientes, que afectan de forma negativa al uso práctico y en casos reales del algoritmo CCD, se propuso en 2018 el algoritmo Natural-CCD. [22] El Natural-CCD es un algoritmo derivado directamente del CCD, pero que obtiene mejores resultados. En especial, destaca que mediante su aplicación se consiguen movimientos más fluidos y naturales de los robots.

Se pueden analizar uno por uno los distintos efectos que el Natural-CCD tiene sobre cada una de las desventajas que se han presentado sobre el CCD:

- En primer lugar, para tratar de manejar las posibilidades de alcanzar una singularidad cinemática, se pueden emplear dos tácticas distintas. En ambos casos, se trata de no realizar el movimiento previsto, y saltar directamente a la siguiente articulación que corresponda en el bucle del algoritmo. Es posible que el movimiento de esas otras articulaciones preventivamente que, cuando en la siguiente iteración se vuelva a trabajar con la articulación, ya no se encuentre en posición de entrar en una singularidad

No obstante, también existen casos en los que la singularidad afecta a todo el robot, en vez de a una articulación en particular. Un caso notorio de esta situación es la configuración en la cual todas las articulaciones mantienen al robot perfectamente colineal, y el vector apunta a la meta. En estos casos, se ha previsto una salida de la singularidad: el algoritmo devuelve un eje de giro y un valor de rotación aleatorizados, provocando la salida de la singularidad y permitiendo la continuidad del algoritmo.

- En lo relativo al manejo de colisiones, se trata de implementar una condición de diseño físico en el propio algoritmo. En general, a la hora de planificar la construcción del robot, se aporta en las especificaciones un dato de giro máximo θ_{max} para las articulaciones, con el fin de evitar la colisión consigo mismo. Con ello se informa a los futuros usuarios de los

giros permitidos con el fin de que no se choque con el propio robot.

En los casos en los que el propio constructor no de la información, y se necesite programar en el algoritmo Natural-CCD, se propone en el artículo de Martín et Al. [22] θ_{max} como el ángulo suplementario al ángulo interior de un polígono de N lados, siendo N el número de articulaciones de las que dispone el robot a modelarse para el Natural-CCD. Posteriormente, este límite se integra en el código, verificando que los giros acumulados no lleguen a superar este valor.

- Finalmente, se debe atajar también el problema de los movimientos abruptos de las articulaciones. El Natural-CCD funciona implementando un coefficiente k que limita el valor de θ_i para cada iteración de rotación. En particular, el inverso de la distancia entre el extremo del robot y la posición deseada para el mismo se sugiere como posible coeficiente limitador. No obstante, k puede escogerse arbitrariamente siempre y cuando se encuentre acotado superiormente por 1, manteniendo así la convergencia del algoritmo numérico.

$$k = \frac{1}{|\vec{p_e p_f}|} \quad 0 > k \geq 1 \quad \theta^* = k\theta$$

Otra posible definición para el coeficiente k se basa en tratar de obtener un número específico de ciclos para un movimiento en concreto. Para hacerlo, siendo N el número total de articulaciones y C una constante positiva, se tiene que:

$$k = \frac{C}{N|\vec{p_e p_f}|} \quad 0 > k \geq 1 \quad \theta^* = k\theta$$

Aplicando todas estas modificaciones al CCD, se obtiene como resultado final el algoritmo Natural-CCD. En la figura 6.5 se puede observar un resumen gráfico con el flujoGRAMA del procedimiento.

La aplicación de este algoritmo tiene algunas consecuencias muy interesantes. En primer lugar, a partir del Natural-CCD se derivan movimientos mucho más fluidos y naturales, con menos requiebros o acciones bruscas. De hecho, se considera que el Natural-CCD tiene un cierto comportamiento biomimético. [22]

6.3. Otros usos del Natural-CCD: Behaviors

Ya se ha introducido el concepto del Natural-CCD como un algoritmo que devuelve resultados de un cierto biemimétismo cuando se aplica a la cinemática inversa de robots RHAC. No obstante, no es la única consecuencia derivada del Natural-CCD. Existen unos comportamientos particulares que aparecen a partir de ligeras modificaciones en el algoritmo, que imitan comportamientos de una cierta complejidad y que surgen de forma completamente natural como derivada del Natural-CCD. Éstos se han denominado behaviours. [25]

Los behaviours se definen como movimientos distintivos del robot, que surgen de forma natural cuando se modifican los tres vectores definidos como núcleo de todo el algoritmo CCD: p_c , p_e y p_f . Para la cinemática inversa se habían tomado como puntos relevantes la articulación actuada, el extremo del robot y el objetivo o meta que se había fijado. Ahora bien, existen otros puntos que también pueden considerarse como relevantes y que, de hecho, cambian radicalmente los efectos del algoritmo.

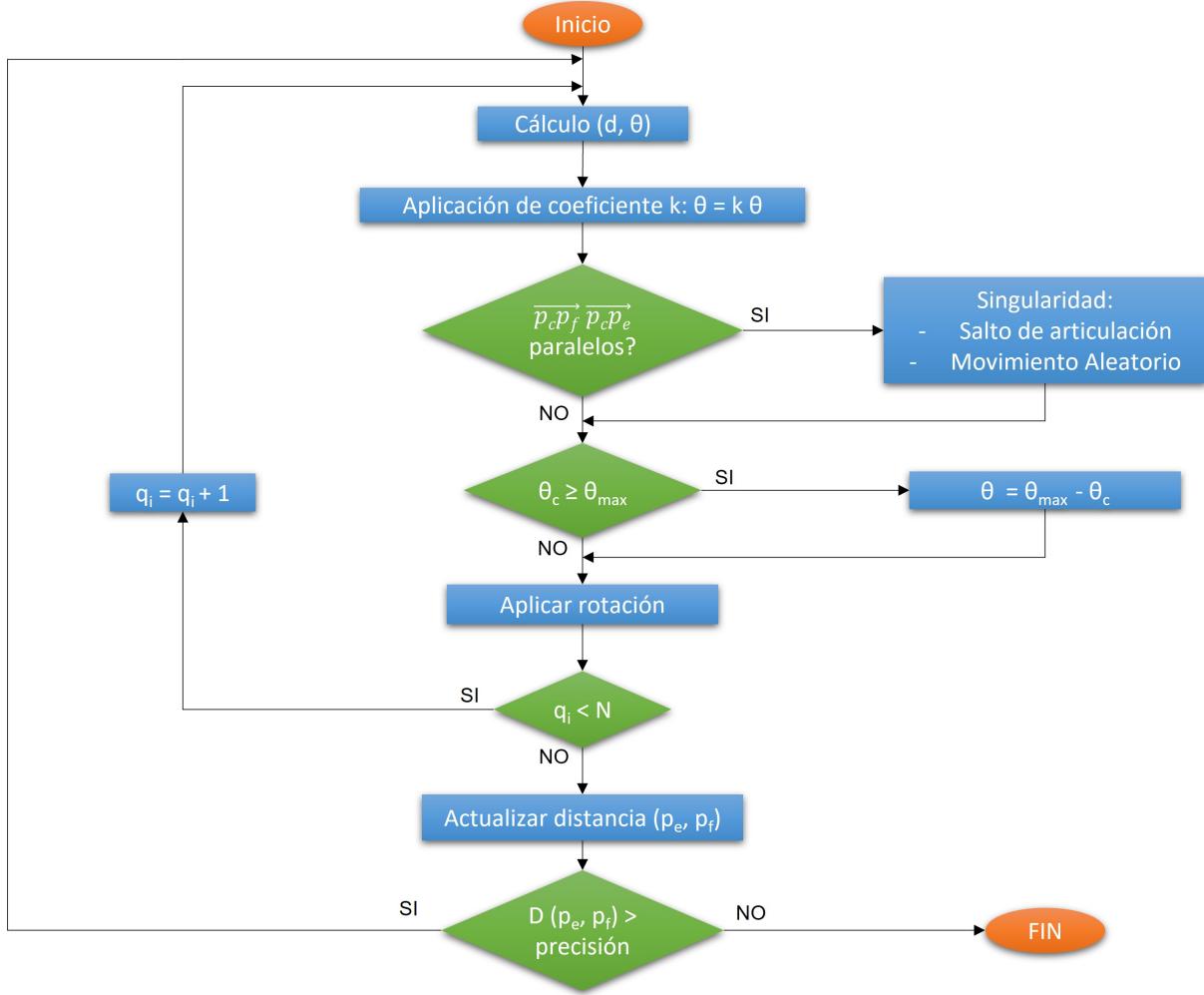


Figura 6.5: Resumen gráfico del flujo del Natural-CCD. Elaboración propia a partir de [22]

Como ejemplo, puede destacarse la utilización de la base del robot como posible punto relevante. Otra opción es, en lugar de considerar solo la articulación actuada, extender la perspectiva y escoger la inmediatamente anterior o posterior a la que se está calculando en la iteración algorítmica. Generalizando completamente, cualquier punto externo podría introducirse en lugar de estos vectores, aunque no todos darán resultados relevantes para el robot.

En el artículo de Martín et Al. se escogen 6 puntos concretos como puntos relevantes. Con ellos, comienza a hacer permutaciones: por ejemplo, escoger la base del robot como p_c , la articulación anterior a la calculada como p_e y la última articulación, el extremo del robot, como p_f . A partir de los seis puntos escogidos, contemplando todas las permutaciones posibles y eliminando aquellas repetidas, da lugar a un total de 40 casos de estudio distinto o 40 behaviours. A su vez, divide estos en dos grupos, uno complementario del otro.

Indudablemente, no todas estas permutaciones presentan el mismo interés de cara al estudio para la aplicación al modelado cinemático de RHACs. De todas ellas, el artículo destaca seis comportamientos de una mayor relevancia que suponen la característica destacable de los behaviours [25]:

- **Atracción:** el robot se ve atraído por un punto destacado determinado por p_e . En concreto, la cinemática inversa tal y como se ha desarrollado aquí pertenece a este behaviour. También se puede analizar aquí el caso de definir una configuración completa, en lugar de un solo

punto, para que el robot la adopte.

- **Repulsión:** se casa del caso contrario, el robot naturalmente se aleja del punto p_e lo máximo posible.
- **Rodear:** este behaviour tiene dos aplicaciones primordiales. En una, el robot se enrolla sobre sí mismo en forma de espiral. En otra, el robot tiende a rodear un obstáculo, agarrándolo independientemente de su forma.
- **Estirado:** sería el complementario al caso de rodear, en el que el robot tendría a recuperar una forma recta.
- **Doblado:** el robot adopta una forma plegada en zig-zag mediante el empleo de este comportamiento. De esta forma, el robot se retrae rápidamente.
- **Circumnutación:** basado directamente en un caso biomimético de las plantas. Se trata de un movimiento helicoidal de forma semejante al crecimiento de algunas plantas trepadoras. Puede generarse, por ejemplo, mediante un objetivo móvil para la cinemática inversa.

Los behaviours no se van a aplicar de forma directa a las ecuaciones de cinemática inversa. Sin embargo, sí pueden introducirse en el primer paso, el algoritmo para determinar la posición de cada sección, con el fin de generar automáticamente movimientos. Esto es, en lugar de simplemente permitir la introducción de puntos de entrada concretos a los que llevar el extremo del robot, sustituyendo para un movimiento dicha entrada por una llamada a un behaviour determinado, se puede generar un movimiento completo.

De hecho, esta integración de los behaviours en la cinemática inversa ya ha sido realizada por el propio Martín-Barrio en su simulador de robots hiperredundantes. Así, se puede modificar el modo cinemática inversa, o se puede elegir un behaviour que de las instrucciones necesarias para que los cables del robot se muevan de tal forma que permitan al robot realizar algunos movimientos, como enrollarse o desenrollarse. [25]

A partir de este conjunto de movimientos variados pueden establecerse diversas estrategias de control que, sin necesidad de conocer la particularidad del robot, con un alto grado de eficiencia computacional y permiten efectuar determinados movimientos de una forma sencilla. Por todo ello, y pese a que en este TFM se va a hacer uso del behaviour atractivo para el extremo del robot, merece la pena conocer el Natural-CCD como posible algoritmo para los RHAC, así como la implementación de los behaviours como posibles órdenes de control para el robot.

7 | PROPUESTA DE GUÍA PARA EL MODELADO DE RHAC. EJEMPLOS DE APLICACIÓN

Con el trabajo desarrollado en los capítulos anteriores, ya se han proporcionado las herramientas necesarias para el modelado cinemático (incluso en algunos casos, dinámico) de los robots hiperredundantes actuados por cables. Tras establecer un marco teórico que simplifica la comprensión de cada modelo, se han desarrollado las ecuaciones a partir de las hipótesis simplificadoras más relevantes en la bibliografía sobre RHAC.

No obstante, en este capítulo se da un paso más: es el momento de ordenar todos esos modelos disponibles de tal forma que sea más sencillo aproximarse al modelado cinemático. En concreto, el objetivo es proponer una guía-tutorial que permita al desarrollador novel hacerse las preguntas adecuadas para seleccionar el tipo de modelo más beneficioso para su morfología concreta de robot.

Para ello, además de la propuesta teórica de la guía-tutorial, se han incorporado en este capítulo tres ejemplos, uno por cada gran grupo de robot hiperredundante, que pretenden ser una muestra de aplicación de este procedimiento propuesto, así como aportar una visión más detallada de los procesos que se necesitan para aplicar concretamente los modelos a los distintos tipos de robot.

7.1. Presentación de la guía - tutorial

En primer lugar cabe plantearse qué criterios seguir para discriminar entre los distintos tipos de RHACs y escoger el modelo cinemático más apropiado. En ese sentido, las decisiones al respecto pueden basarse en numerosos factores subjetivos y objetivos. De cara al desarrollo de esta guía tutorial, se ha tratado de seguir el criterio de la simplicidad.

Como norma general, es deseable mantener al mínimo posible el coste computacional de aplicar modelos complejos. En muchas ocasiones no existe una única respuesta correcta: dependerá de los medios hardware, software e incluso humanos de los que se disponga. Ahora bien, en esta guía-tutorial se ha preferido destacar aquel modelo más sencillo, menos costoso y más comprensible ante posibles alternativas.

El procedimiento de elección de modelos que se ha propuesto se ha plasmado gráficamente en la figura 7.1. Se puede ver cómo se han organizado las distintas hipótesis simplificadoras de una manera sencilla y clara. Así, en una primera aproximación un nuevo diseñador de RHAC puede tener una primera orientación sobre el modelo que sea más conveniente.

Es conveniente, no obstante, realizar un análisis en mayor detalle sobre los distintos pasos y las distintas disyuntivas que se presentan en la guía. Como se puede observar, se fundamenta la elección de modelo en cuatro preguntas sucesivas, representadas en el gráfico como los rombos verdes. En función de la respuesta, escogida particularmente para el robot que se está tratando

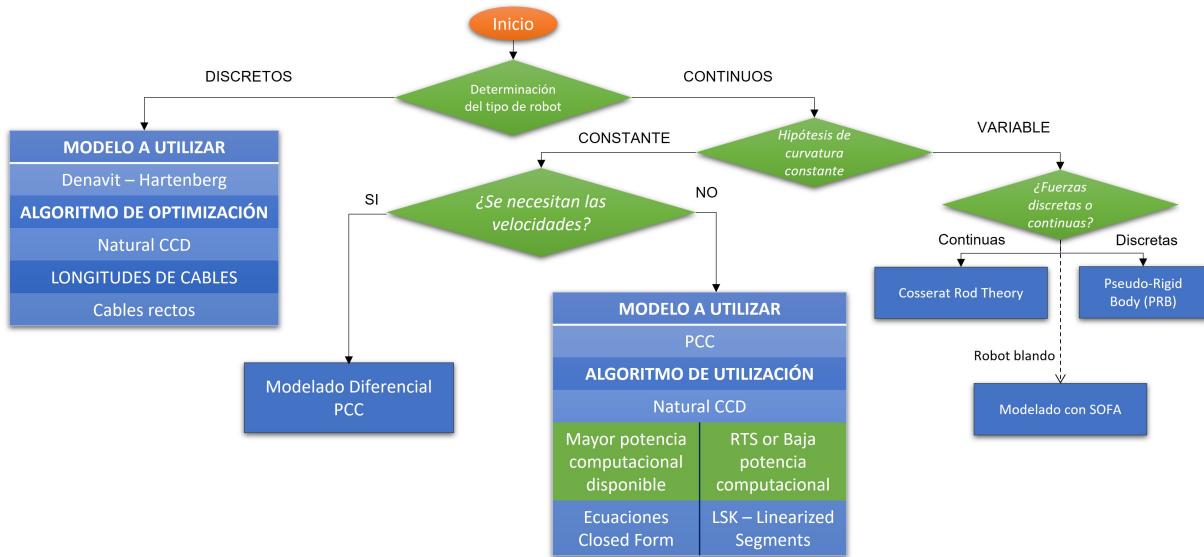


Figura 7.1: Resumen de la propuesta para el modelado de RHACs. Elaboración propia

de modelar, se va procediendo hasta llegar a un conjunto de ecuaciones representados en azul en el gráfico.

Además de las cuatro preguntas principales, que sirven para determinar a grandes rasgos el modelo que determinará la hipótesis simplificadora que será más conveniente aplicar en cada caso, para el modelo PCC en concreto se ofrece una recomendación basada en el principio de simplicidad previamente indicado: se recomiendan unas ecuaciones para los cables a partir de criterios auxiliares, pero ambas opciones son perfectamente válidas y coherentes con el modelo de robot.

En las secciones posteriores se irá detallando aquellos criterios en los que es más importante fijarse, al plantearse las preguntas de la guía tutorial, tratando de facilitar el seguimiento de la misma.

7.1.1. Determinación del tipo de robot

El primer paso que se puede dar de cara a concebir un modelo cinemático de un RHAC se basa, precisamente, en la clasificación dada en la sección 2.2. Es fundamental determinar la tipología del robot, en cuanto a si se puede considerar un robot discreto o continuo; tal y como se ha ido desarrollando en los diferentes modelos planteados, puede obligar o no a emplear un determinado método.

Así pues, el primer paso es determinar se el robot puede considerarse continuo o discreto, en función de las articulaciones y los grados de libertad del robot. Ya se indicó que la definición de continuidad de un robot implica, puramente, que éste disponga de infinitos grados de libertad. Esto en muchos casos es complejo de determinar.

Por ello, se propone una perspectiva distinta, pero más pragmática. De cara al modelado, puede considerarse el robot como discreto siempre y cuando pueda definirse como una cadena cinemática de un número finito de articulaciones de grados de libertad de rotación dispuestas en serie. La razón para imponer esta condición es, sin duda alguna, porque es la naturaleza en sí misma del algoritmo de Denavit-Hartenberg. En robots que cumplan esa condición, la aplicación del método es directa.

En cambio, para robots continuos, como ya se ha visto, la aproximación se plantea de forma distinta. Se ha visto en la sección 4.2 que el modelo cinemático PCC también puede aprovechar el algoritmo de Denavit-Hartenberg para alcanzar las ecuaciones. No obstante, esto no es lo relevante. Las ecuaciones que se alcanzan, de hecho, ya se tienen disponibles. En el caso de los robots continuos lo esencial es conocer las ecuaciones del arco que define el núcleo, puesto que éste es el que va a determinar la posición de los discos pasivos intermedios, afectando por tanto a las ecuaciones dependientes del robot. En este aspecto, el modelo de Denavit-Hartenberg es insuficiente.

Este es precisamente el quid de la cuestión: las longitudes de los cables. Indudablemente, cualquiera de las representaciones dadas puede representar, a través de una matriz de transformación homogénea, la posición y orientación de la sección inmediatamente siguiente. No obstante, en cada uno de los tipos de robots, la conveniencia se justifica, en el caso de los robots discretos, que se permite obtener esta MTH en función de los parámetros articulares reales del robot, y en el caso de los robots continuos, que el algoritmo de Denavit-Hartenberg es insuficiente: las longitudes de los cables dependen directamente de su posición en la curva descrita por el núcleo del robot.

Así pues, a partir de esta pregunta se deriva la primera opción para el modelado cinemático: en el caso de los RHAC discretos, se usará el modelo de Denavit-Hartenberg tanto para los modelos directo e inverso en las ecuaciones independientes del robot. Por otra parte, para la cinemática inversa se usará, como era de esperar, el algoritmo de Natural-CCD.

Finalmente, y a raíz de lo discutido anteriormente, se puede decir que lo fundamental para los cálculos de longitudes de los cables es que se considerará que trazan una línea recta entre los planos de las secciones que determinan el ángulo entre las mismas. Esto contrasta con el caso de curvatura continua, que debe seguir los trazos de los discos pasivos. La diferencia puede observarse en la figura 7.2.

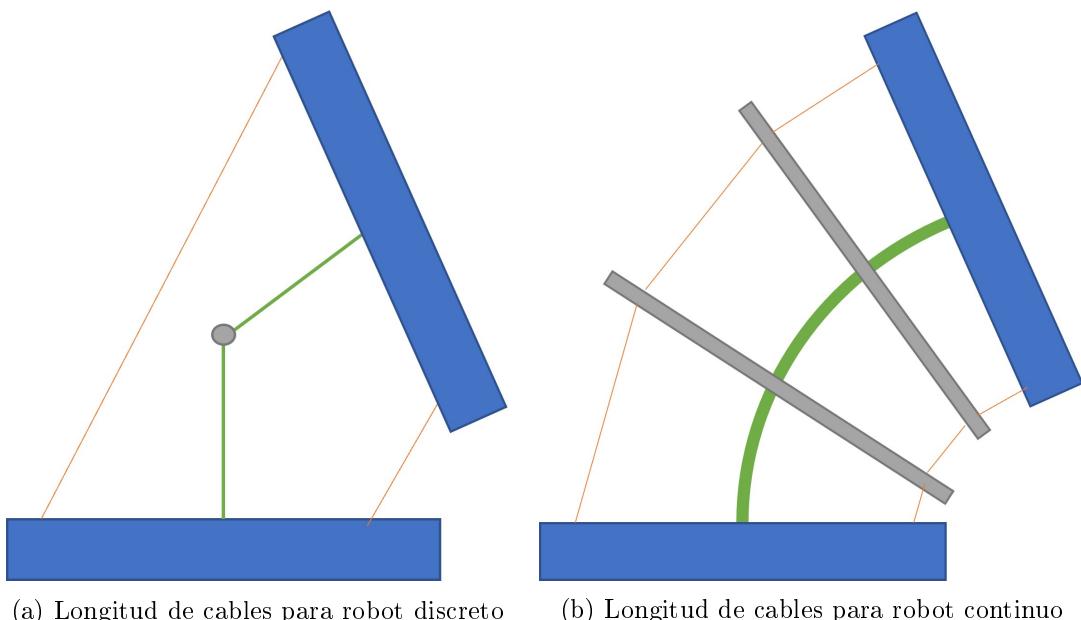


Figura 7.2: Diferencias entre los planteamientos de las longitudes de cables para RHAC discretos y RHAC continuos. Elaboración propia

7.1.2. Hipótesis de curvatura constante

Si en el apartado anterior la conclusión es que se trata de un robot continuo, no se dispone de un modelo preparado para aplicar. En este caso, se debe plantear una nueva pregunta para poder determinar cómo proseguir: ¿Es aceptable aplicar al robot la hipótesis de curvatura constante?

Ya se ha comprobado que la hipótesis de curvatura constante es muy poderosa, y permite utilizar un conjunto de ecuaciones analíticamente obtenidas. En general, es un modelo muy extendido para el trabajo con RHACs, por su sencillez conceptual y matemática.

Ha de destacarse que tener un robot que trace un arco de circunferencia perfecto es imposible de tener; como todo modelo, tendrá una incertidumbre asociada. Así pues, no se trata tanto de cumplir una serie de requisitos que hacen posible aplicar sobre un robot la hipótesis de curvatura constante. El criterio para su aceptación es un criterio eminentemente empírico: comprobar mediante una serie de pruebas si, para el caso concreto de uso de ese robot, la precisión que aporta el modelo PCC es suficiente, o si se debe adoptar otro criterio para la elección del modelo.

Indudablemente, sí existen una serie de condicionantes constructivos que favorecen la aparición de un mayor error, y que pueden considerarse como indicios de cara a considerar el mejor modelo posible. En la sección 5 ya se indicaron algunos de los motivos por los que, en la bibliografía, se ha considerado que el modelo PCC suele fallar. Entre ellos se encuentran un peso específico excesivo en la estructura del robot (impidiendo que sea capaz de soportar su propio peso, y deformando la curva que traza el robot) o el empleo de un material poco resistente.

Como se observa, en estos casos en los que las razones son eminentemente mecánicas, es difícil que un buen controlador sea capaz de corregir el error. No obstante, en la bibliografía no se ha encontrado ningún valor numérico que se haya dado, ni siquiera como referencia, para poder aceptar o rechazar de forma taxativa el empleo de la hipótesis de curvatura constante. Por ello, lo más conveniente para el diseñador del modelo es probar en primer lugar el modelo PCC: su simplicidad frente a los modelos dinámicos lo hace mucho más conveniente si se puede aplicar (como de hecho así se hace en la bibliografía), y en muchos casos el error cometido es despreciable.

7.1.3. Conocimiento sobre las fuerzas: ¿son continuas o discretas?

Por concluir en primer lugar la casuística relacionada con un robot para el que la hipótesis de curvatura constante no sea aceptable, se desarrollará primero esta rama de la guía. Una vez se ha verificado que el modelo PCC no es aplicable al robot, es inmediata la consideración de los modelos dinámicos, desarrollados en el capítulo 5. Ambas alternativas necesitan como variables de entrada las fuerzas que actúan sobre el robot, para poder obtener en última instancia las longitudes de los cables para cada sección.

El conocimiento del que se disponga sobre estas fuerzas será, precisamente, el criterio de elegir entre uno u otro modelo. Recordando brevemente los conceptos sobre los que se fundamentan ambas propuestas, el modelo PRB discretiza de alguna forma la sección en cuatro segmentos rectos unidos por articulaciones de rotación. En las ecuaciones se admitía la presencia de fuerzas externas, pero en todo caso se modelaban en el extremo del robot (o, como mucho, de la sección).

Así pues, para poder seleccionar el PRB como modelo para el robot, se debe poder asumir que todas las fuerzas que actúan sobre el robot pueden asimilarse en la punta de la sección. Esto tiene ciertas implicaciones: si la causa de rechazar la hipótesis PCC es el peso específico excesivo del robot, es complicado que éste pueda rechazarse, y el modelo PRB puede no ser la mejor

alternativa. No obstante, en robots con distancia pequeña entre discos, o con un peso marcado en los discos activos, sí puede ser una posibilidad modelar el peso en los extremos.

Asimismo, también es un conjunto de ecuaciones de extremada utilidad en aquellos casos en los que la razón por la que el robot se desvía del modelo PCC es la recogida de objetos pesados que impiden su asimilación a un arco de circunferencia perfecto. Es, de hecho, el caso para el que se diseñó dicho modelo. [16]

En todo caso, puede ser que todas estas asimilaciones no sean válidas. En muchas ocasiones, tal y como sucede con el caso del peso específico, no es suficiente con considerar fuerzas actuantes en puntos concretos. Es precisamente para estos casos para los que surge la teoría de Cosserat aplicada a los robots. La CRT surge para modelar vigas sometidas a esfuerzos, ya sean discretos o continuos. En el desarrollo de la sección 5.2 las ecuaciones se obtuvieron precisamente suponiendo esfuerzos continuos.

Así, la teoría CRT pese a ser una alternativa computacionalmente más costosa (requiere una integración numérica en el tiempo para el sistema desarrollado) es una buena opción al enfrentarse a problemas en los que actúen fuerzas continuas. A pesar de ello, ha de tenerse en cuenta de que no es infalible. La teoría CRT considera la existencia de un núcleo del robot homogéneo y continuo, asimilable, al fin y al cabo, a una viga.

Existe una serie de robots para los que esta hipótesis puede no ser suficiente: los robots blandos. A pesar de que algunos de ellos sí han llegado a modelarse con la CRT e incluso, en algunos casos, con la PCC, en estos casos sí puede ser más recomendable optar por un simulador capaz de integrar en el espacio las distintas ecuaciones mediante técnicas de Elementos Finitos, tales como SOFA.

Por ello, esta alternativa se ha representado mediante una línea discontinua en el esquema: se trata de reflejar que es una posibilidad típicamente considerada en la bibliografía. No todos los robots blandos se modelan con este método, pero sí es frecuente, y no deja de arrojar resultados algo más precisos que otras hipótesis, tal vez menos ajustadas a la realidad, a costa de un mayor esfuerzo computacional.

Con esto se han cubierto en la propuesta de guía-tutorial los casos de los robots continuos fuera de la hipótesis de curvatura constante.

7.1.4. Las velocidades como requisito del modelo

Regresando al caso en el que es aceptable la hipótesis de curvatura constante, puede concluirse que el resto de disyuntivas a considerar entrarán todas bajo el paraguas del modelo PCC. No obstante, tal y como se ha ido explicando en la sección 4.2 y 4.3 dentro de este modelo existen múltiples enfoques que se deben discriminar.

Una pregunta que puede hacerse en este momento el diseñador es qué variables dispone a la hora de afrontar el modelo. En cualquiera de los casos, a partir de la posición puede aproximarse suficientemente la velocidad, acumulando datos en el tiempo y tratando de aproximarse al límite de la derivada en el tiempo. No obstante, en el caso del modelo PCC se ha presentado el desarrollo, frecuente en la bibliografía, del modelo cinemático diferencial.

Así, para aquellos casos en los que los motores por su naturaleza, o por los sensores de los que se dispone para captar la información cinemática, se dispone de la velocidad de rotación (y, por ende, de la velocidad de enrollado o desenrollado del cable, pudiéndose traducir a \dot{l}) puede

ser más recomendable optar directamente por un modelo diferencial.

Otra opción por la que podría ser interesante el modelo diferencial es cuando la información relevante para el usuario es la velocidad de la punta del robot. En estos casos, en los que se puede querer realizar un control en velocidad, y en los que se desea plantear límites directos a la misma, puede ser más conveniente escoger el modelo diferencial como el que gobierna la cinemática del robot.

De esta forma, se es más fiel a la información real que se tiene. Posteriormente se puede integrar esta información en un ordenador o en el microcontrolador del robot para obtener los datos sobre la posición concreta del robot.

7.1.5. Otros criterios: ecuaciones para la longitud de los cables

Queda una última disyuntiva presentada en el resumen gráfico de la figura 7.1. En el caso de que se escoja un modelo no diferencial (esto es, cinemática directa e inversa empleando posiciones como variables de las ecuaciones), ya se puede fijar la elección del modelo PCC, y el algoritmo Natural CCD como la alternativa de algoritmo de optimización escogida.

No obstante, queda por escoger la ecuación que se empleará para el cálculo de las longitudes de los cables. Ya en la sección 7.1.1 se ha tratado la importancia de las longitudes del modelo y que se trata de una diferencia fundamental entre la consideración de los robots continuos y los discretos (que no suelen tener discos pasivos). También dentro del modelo PCC puede optarse por distintas alternativas de afrontar este aspecto concreto.

En la sección 4.2.2 se introdujeron, para las ecuaciones dependientes del robot, un ejemplo para robots de tres cables de las ecuaciones Closed-Form, que modelan, a partir de la información sobre la curvatura del arco de circunferencia al que se asimila el núcleo, la longitud necesaria de los cables para alcanzar dicha configuración. El desarrollo pretende ajustarse lo máximo posible a la realidad, utilizando fórmulas trigonométricas para deducir los resultados.

El coste computacional de estas funciones trigonométricas fue la causa para introducir un modelo muy reciente que ha permitido abordar este problema: el modelo cinemático basado en segmentos linearizados (LSK). Desarrollado en la sección 4.3 se insiste en que pertenece al gran marco teórico que representa el modelo PCC. No obstante, a partir de linealizar las ecuaciones correspondientes a las longitudes de los cables (toma como origen las ecuaciones Closed-Form) reduce enormemente ese coste computacional.

Esta es la principal razón por la que el criterio a considerar en este caso es el coste computacional. En el punto actual de la guía-tutorial, es el caso de que ambos modelos son válidos. La diferencia fundamental entre ambas alternativas, y que es la que debe guiar al desarrollador al elegir entre ambas es, precisamente, si dispone de potencia computacional suficiente como para ejecutar para todo el robot las ecuaciones Closed-Form, o si es más conveniente optar por el LSK. Por ejemplo, la necesidad de ejecutar el modelo en tiempo real (RTS) puede ser un elemento más que opte por el modelo LSK. Indudablemente, linealizar las ecuaciones tiene un coste en precisión, pero en el caso del LSK ha demostrado ser relativamente pequeño. [4]

Una vez tomada esta decisión, puede considerarse que ya se dispone de la información suficiente como para concretar los modelos para el robot con el que se desee trabajar. El ordenamiento gráfico de todos estos modelos puede resultar extremadamente útil para poder escoger una opción con la que modelar un nuevo robot, recogiendo asimismo la información bibliográfica disponible sobre RHAC y la obtención de sus ecuaciones cinemáticas.

7.2. Ejemplificación sobre robots reales

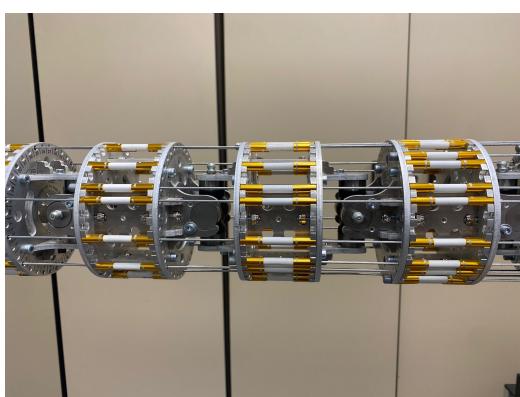
A pesar de ello, puede ser extremadamente útil ver aplicados estos conceptos teóricos a algunos robots reales. De esta forma, las dudas que puedan quedar al plantearse las preguntas propuestas por la guía pueden clarificarse mediante la presentación de ejemplos para los tipos principales de robot. De esta forma, se pueden seguir los mismos pasos viendo su aplicación a distintas morfologías.

Para ello, se han tratado de escoger tres casos representativos de robots disponibles en el CAR: el MACH-I, un robot discreto; Ruan, un robot blando y Pilory, un robot de discos pivotantes. En esta sección se desarrollará el modelo cinemático de cada uno de ellos siguiendo los pasos comentados de la guía-tutorial.

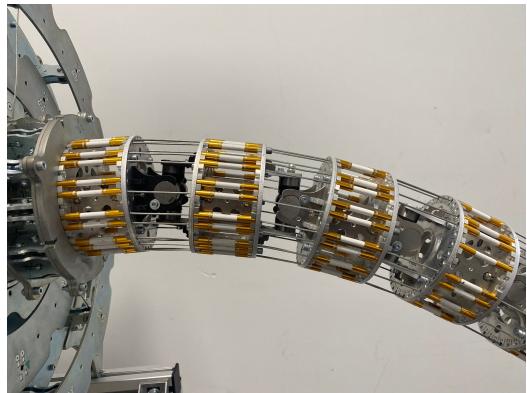
7.2.1. MACH-I: Ejemplo de un robot discreto

El MACH-I es un robot del CAR desarrollado por Iván Rodríguez, y mejorado por David Márquez y Andrés Martín, que crearon una interfaz gráfica, una aplicación para su control mediante realidad virtual y desarrollaron algunos equipos de protección electrónicos para el robot. [37, 26, 1]

Se trata claramente de un robot discreto. Siguiendo el criterio desarrollado en la pregunta sobre la continuidad de los robots, es una cadena cinemática de 7 juntas Cardan, de dos grados de libertad cada una. Así, cumple el requisito necesario para poder ser considerado un robot discreto según los parámetros de este trabajo. Puede verse en las figuras 7.3a y 7.3b los detalles constructivos del robot que lo clasifican como un robot discreto. En ellas también puede comprobarse el trazado recto que siguen los cables entre las secciones.



(a) Detalle de las articulaciones



(b) Detalle de las secciones del robot en posición curvada

Figura 7.3: Detalles del MACH-I que apoyan su clasificación como robot discreto

Así pues, siguiendo el esquema gráfico de la figura 7.1, se observa que ante la respuesta de que se trata de un robot discreto, el MACH-I obtiene la recomendación de modelarlo empleando el algoritmo de Denavit-Hartenberg como marco teórico. La referencia completa del método puede encontrarse en la sección 4.1.2. Es, no obstante, un método sencillo de aplicar.

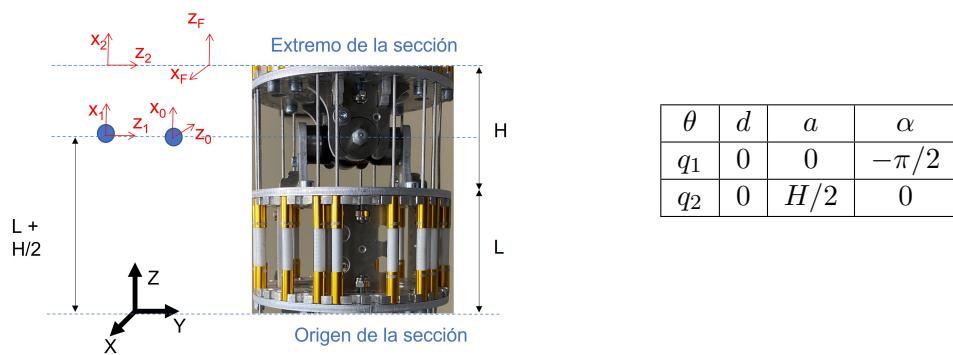
Con el fin de poder ser riguroso con el modelo cinemático, se van a usar los esquemas para robots discretos que se encontraban en las figuras 4.1 y 4.2 como base para el desarrollo completo del modelo. Se puede observar que, tanto en la cinemática directa como en la inversa se encuentra un paso referido a la obtención de la matriz de Denavit-Hartenberg.

Además de este paso, que se concretará a continuación, deben tenerse en cuenta tres elementos más. En primer lugar, el algoritmo de optimización de la cinemática inversa. Por coherencia con el resto de este trabajo fin de máster, se propone emplear el algoritmo Natural-CCD para la resolución del primer paso de las ecuaciones inversas. Por otro lado, se requiere una relación que permita pasar del estado de las secciones a los parámetros articulares del robot, también en la cinemática inversa. Finalmente, en ambos esquemas se observa que se necesitan unas ecuaciones que relacionen las longitudes de los cables con esos mismos parámetros articulares.

Con todos estos pasos necesarios en mente, se puede comenzar la aplicación del algoritmo de Denavit-Hartenberg para este caso concreto. Con el fin de poder abordar correctamente el modelado, se tendrán en cuenta las siguientes consideraciones.

- La base del robot, y su sistema de referencia global se situará en el disco en contacto con la base metálica, coincidente con el primer tramo recto de cable contenido en los cilindros guía.
- El eje z_G será perpendicular al disco inicial de la sección, y paralelo al segmento recto de cable que sigue.
- El eje x_G se situará, mirando al robot desde la punta hacia la base, en dirección horizontal con sentido hacia la izquierda.
- El eje y_G se situará, por coherencia con un sistema de referencia dextrógiro, en vertical y hacia arriba.
- El resto de secciones se situarán de forma que sean paralelos en todo caso a los ejes definidos por el sistema global cuando el robot esté completamente estirado.
- El origen de coordenadas de la sección i estará situado en el centro del disco inmediatamente posterior a la articulación $i + 1$

Aplicando estos criterios, es ahora sencillo aplicar la metodología de Denavit-Hartenberg. Para ello, se deben definir los correspondientes sistemas de referencia que requiere el algoritmo, conforme a las normas de orientación de los ejes que se necesitan cumplir. Un elemento al que hay que prestar atención es a la distancia recta que existe en cada sección, que se debe incorporar posteriormente.



(a) Figura esquemática para la aplicación del algoritmo Denavit-Hartenberg al MACH-I

(b) Parámetros del algoritmo de Denavit Hartenberg para los sistemas de referencia definidos.

Figura 7.4: Aplicación del método Denavit-Hartenberg al MACH-I

Puede comprobarse que la elección de sistemas de referencia y de parámetros de Denavit-Hartenberg que se ha definido para el MACH-I son semejantes a la realizada en la sección 4.1.2. La definición de los sistemas de referencia propios para el algoritmo se puede observar en la figura 7.4a, y sus parámetros asociados se encuentran en la figura 7.4b.

Debe hacerse notar que, propiamente, los sistemas de referencia definidos por el algoritmo solo son los que tienen el subíndice 0, 1 y 2. Así, el paso del sistema 0 al 1 representa un grado de la articulación, y el paso del 1 al 2 la otra articulación. Dado que la orientación de estos sistemas viene predefinida por el propio algoritmo, no cumplen con las directrices dadas previamente para la definición de sistemas de referencia. La forma de abordar esta discrepancia es incorporar posteriormente dos transformaciones de rotación, que permiten definir el sistema de la base de la sección y el sistema final a partir de rotaciones simples y una traslación (ver las ecuaciones 7.1 y 7.2).

$${}^B A_0 = T(0, 0, L + H/2) Rot_y(-\pi/2) \quad (7.1)$$

$${}^2 A_F = Rot_y(\pi/2) Rot_z(\pi/2) \quad (7.2)$$

Sustituyendo los parámetros de Denavit-Hartenberg en la matriz para cada sección 7.3, se dispone con ${}^{il-1} A_k$ la matriz de transformación homogénea de la matriz k-ésima. Por lo explicado en las fórmulas anteriores, se requieren también las transformaciones extras para disponer de la MTH de la sección:

$$S_i = T(0, 0, L + H/2) Rot_y(-\pi/2)^1 A_2^2 A_3 Rot_y(\pi/2) Rot_z(\pi/2) \quad (7.3)$$

Finalmente, el punto final de cada sección se puede referenciar a la sección base, premultiplicando la MTH de la sección por las MTH de todas las secciones anteriores. En concreto, para el extremo del robot se tiene que:

$$S_n = S_1 \cdot S_2 \cdot \dots \cdot S_{n-1} \cdot S_n \quad (7.4)$$

En este momento se dispone ya de las expresiones correspondientes al modelo de Denavit-Hartenberg de la cinemática directa y la inversa. Quedan, como se apuntaba anteriormente, dos pasos del modelo inverso y uno del directo. Comenzando por la cinemática inversa, se necesita poder transformar el estado de las secciones en los parámetros articulares que emplea el algoritmo de Denavit-Hartenberg.

En el caso del MACH-I es sencillo, partiendo de la información dada por el Natural-CCD: teniendo el ángulo de giro y la dirección del eje de rotación, puede emplearse la fórmula de Rodrígues para calcular el nuevo eje z , tras haber experimentado la rotación:

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \sin \theta \begin{bmatrix} 0 & -d_z & d_y \\ d_z & 0 & -d_x \\ -d_y & d_x & 0 \end{bmatrix} + (1 - \cos \theta) \begin{bmatrix} 0 & -d_z & d_y \\ d_z & 0 & -d_x \\ -d_y & d_x & 0 \end{bmatrix}^2 \quad (7.5)$$

Premultiplicando la matriz R con el vector $[0 \ 0 \ 1]^T$ se tiene el nuevo eje z , que se llamará v . El giro del eje z a v se necesita descomponer en un giro en torno al eje X original, para dar lugar al vector intermedio i , seguido por un giro en torno al nuevo eje \bar{y} . Esta descomposición puede observarse en la figura 7.5.

Para el cálculo de los parámetros q_1 y q_2 se plantea un ejercicio básico de geometría: conociendo el vector y los ángulos se pueden determinar mediante trigonometría básica. Se conoce que el vector i es perpendicular al eje x (es resultante del giro de z en torno a ese eje), así como

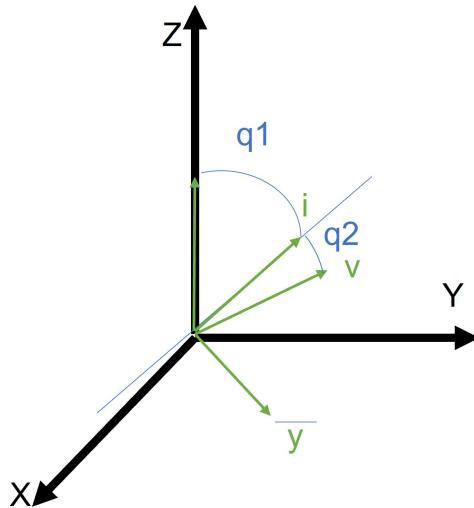


Figura 7.5: Descomposición del giro en los parámetros q_1 y q_2 . Elaboración propia

al eje \bar{y} . Por otra parte, el vector \bar{y} es perpendicular tanto a x como al vector final v .

Con esto en mente, se puede calcular el vector \bar{y} como:

$$\bar{y} = \frac{[1 \ 0 \ 0]^T \times v}{\| [1 \ 0 \ 0]^T \times v \|} \quad (7.6)$$

Finalmente, puede calcularse el vector i como un nuevo producto vectorial, ya que es perpendicular tanto al eje x como al vector \bar{y} :

$$i = [1 \ 0 \ 0]^T \times \bar{y} \quad (7.7)$$

Finalmente, los parámetros angulares q_1 y q_2 se deducen mediante los triángulos que se forman:

$$q_1 = -\arctan 2\left(\frac{\bar{y}_3}{\bar{y}_2}\right) \quad (7.8)$$

$$q_2 = \arccos(v \cdot i) \quad (7.9)$$

El signo de q_2 se modifica en función del signo del producto escalar. Con este conjunto de ecuaciones ya se dispone de la relación entre el Natural-CCD y los parámetros de Denavit-Hartenberg. Para completar la cinemática inversa únicamente se requiere la relación entre éstos y las longitudes de los cables. Esto demuestra ser relativamente sencillo, ya que el empleo de las MTH permite su cálculo directo.

Dado que conocemos las geometrías de los discos que delimitan las secciones, es posible obtener los puntos donde se sitúan los orígenes de los cables en cada sección, y calcular las longitudes de los cables a partir de la diferencia entre vectores. Con la definición de los sistemas de referencia según los convenios adoptados previamente, el cable i para el disco de la primera sección se sitúa en el punto determinado por las expresiones:

$$\phi_i = (i - 1) \cdot \frac{2\pi}{36} + 0,5 \cdot \frac{2\pi}{36} \quad (7.10)$$

$$O_i = \begin{bmatrix} r \cos \phi \\ r \sin \phi \\ 0 \\ 1 \end{bmatrix} \quad (7.11)$$

Nótese que se ha introducido la cuarta componente 1 para que su transformación mediante MTH sea inmediata. En la figura 7.6 puede observarse una construcción geométrica de lo que se desea obtener. Se define el vector P_i^j como el que une el origen del cable de la sección j con su origen para la sección siguiente $j + 1$.

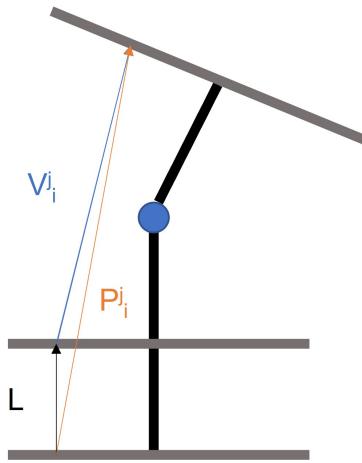


Figura 7.6: Razonamiento geométrico para las longitudes de los cables. Elaboración propia

Éste se puede calcular mediante la MTH de la sección j : S_j .

$$P_i^j = S_j \cdot \begin{bmatrix} 0 \\ d \\ 0 \end{bmatrix} - O_i \quad (7.12)$$

Ahora bien, este vector no refleja fielmente la longitud del cable. No tiene en cuenta que en el MACH-I existen unos tramos rectos de los cables en cada sección, de longitud L . Así, descomponemos la longitud del cable para cada sección en una componente fija L y una componente variable V_i^j . Ésta se calculará como la resta entre el vector P_{ji} y el vector fijo en la dirección del eje z de longitud L :

$$V_i^j = P_i^j - \begin{bmatrix} 0 \\ 0 \\ L \end{bmatrix} \quad (7.13)$$

La longitud final total se calcula como la suma de la componente fija y la componente variable:

$$l_i^j = L + \|V_i^j\| \quad (7.14)$$

Con estas ecuaciones se puede calcular finalmente de forma recursiva la longitud para todos los cables. Teniendo en cuenta lo expuesto en la sección 4.2.2, comenzando por la sección 1 se calculan todas las longitudes de los cables que la atraviesan con su MTH, y así sucesivamente para todas las secciones. Sumando todas ellas, se obtiene la longitud completa para todos los cables.

Hasta aquí se ha desarrollado un modelo cinemático inverso completo. No obstante, también sería necesario conocer en cada momento dónde está el robot, a partir de la información de la posición de los husillos que tensan y destensan los cables. Para esto es imprescindible completar el modelo cinemático directo, al que le faltan las ecuaciones dependientes que relacionan las longitudes de los cables con los parámetros geométricos q_1 y q_2 .

Para ello es necesario aplicar los conocimientos sobre robótica paralela. En la sección se utilizan tres grados de libertad actuables (las longitudes de los cables) para localizar la sección con dos parámetros (los ángulos q_1 y q_2). Por tanto, es un robot con una componente de paralelismo que ha de tenerse en cuenta a la hora de obtener estas ecuaciones.

Siguiendo la guía de Merlet [28] se puede comprobar que el robot tiene tres grados de libertad de traslación (los orígenes de los cables del segundo disco se desplazan en el espacio) permitiendo una rotación libre de los cables. Por tanto, la estructura paralela coincide con un robot PPP-3S. Esta estructura no tiene solución analítica posible: para resolver el sistema se necesitarían métodos numéricos para optimizar a partir del polinomio característico o una solución de Inteligencia Artificial.

En este trabajo, en la sección 8.4 se ha desarrollado un modelo de redes neuronales que permiten establecer la relación estadística entre longitudes de cables y parámetros articulares.

7.2.2. Ejemplo de robot no PCC: Ruan

El segundo ejemplo que se va a considerar en este capítulo es un robot al que no aplica la hipótesis de curvatura constante: el RHAC Ruan. Ruan es un robot que se puede categorizar como robot blando (ver figura 7.7). Su movimiento se logra a través de filamentos SMA que atraviesan sus secciones y que movilizan el robot para adoptar la forma deseada.

No obstante, comenzando la propuesta de la guía tutorial desde el principio, la primera categorización que se debe realizar es si es un robot discreto o continuo. Observando la morfología del robot, se observa que no tiene una cadena cinemática clara de articulaciones de rotación. Así, siguiendo este criterio propuesto, no se puede incluir en la categoría de robots discretos: esto excluye la posibilidad de modelarlo con el método de Denavit-Hartenberg como se ha hecho con el robot MACH-I.

El siguiente interrogante que se plantea es si puede ser modelado como un robot de curvatura constante. Como ya se ha desarrollado en la sección 7.1.2 esta decisión es más bien subjetiva. Lo propio sería tratar de implementar las ecuaciones del modelo PCC y posteriormente comprobar si según el criterio establecido por el desarrollador se cumple.

En este caso concreto, se pueden analizar esas “pistas” que se indicaban que pueden dar algún indicio. Se trata, en primer lugar de un robot blando: esto hace complicado que la precisión del PCC sea ajustada al comportamiento real del robot. Asimismo, la silicona que emplea tiene un peso específico elevado, que deforma la curva que traza el robot en funcionamiento. Por todo ello, se va a optar por considerar que requiere un modelo más preciso.

Siguiendo el procedimiento de la guía tutorial, se presentan nuevas incógnitas que se deben considerar. ¿Cómo son las fuerzas que se conocen que van a actuar sobre el robot? Responder a esta pregunta va a permitir decidir si el modelo a emplear es el PRB o la teoría de Cosserat (CRT). Para ello, se analizan los esfuerzos interiores y exteriores que se dan en el robot.



Figura 7.7: Ruan, un robot blando

Se trata de un robot blando, que, como se ha indicado previamente, dispone de filamentos SMA que actúan de forma discreta sobre los discos del robot, al ser sometido a los cambios de temperatura que generan su movimiento. Esto favorece que se prefiera emplear el método PRB antes que el CRT, por poder considerar sin un alto grado de error que las fuerzas sean discretas sobre los finales de sección.

Indudablemente, hay otro punto a considerar. Tratándose de un robot blando, sería posible, como también se indica en la guía tutorial, optar por un método completamente numérico y usar el simulador SOFA para modelar la posición del robot en cada momento. No obstante, se considera que el robot Ruan no va a realizar tareas de precisión, y que por tanto es suficiente con el modelo PRB, que exigirá algo menos de potencia computacional.

Así pues, se puede proceder a aplicar las ecuaciones del modelo PRB al robot Ruan. Se trata de un robot con tres discos, que separan las secciones entre sí, de tal manera que se deben particularizar las ecuaciones del modelo PRB para calcular las longitudes de los cables a partir de las fuerzas para cada sección. Dado que la deducción pormenorizada de las ecuaciones ya se realizó en la sección 5.1, no parece razonable repetir todo el desarrollo matemático. Así, el conjunto de ecuaciones que se obtiene se puede encontrar en la figura 7.8a.

Estas ecuaciones, como se indicó en la sección 5.1 constituyen el núcleo del modelo cinemático y se deben resolver simultáneamente con un solver de ecuaciones no lineales, con el fin de poder obtener para cada momento todas las variables del modelo.

Sección 1	Sección 2	Sección 3
$\vec{F}_{j_1} = \tau_{j_1} \frac{\vec{o}_{j,1:B}}{\ \vec{o}_{j,1:B}\ } + \tau_{j_2} \frac{\vec{o}_{j,2:1}}{\ \vec{o}_{j,2:1}\ }$	$\vec{F}_{j_2} = \tau_{j_2} \frac{\vec{o}_{j,2:1}}{\ \vec{o}_{j,2:1}\ } + \tau_{j_3} \frac{\vec{o}_{j,3:2}}{\ \vec{o}_{j,3:2}\ }$	$\vec{F}_{j_3} = \tau_{j_3} \frac{\vec{o}_{j,3:2}}{\ \vec{o}_{j,3:2}\ }$
$\vec{F}_1 = \sum_{j=1}^f \vec{F}_{j_1} + \vec{F}_2$	$\vec{F}_2 = \sum_{j=1}^f \vec{F}_{j_2} + \vec{F}_3$	$\vec{F}_3 = \sum_{j=1}^f \vec{F}_{j_3} + \vec{F}_{ext}$
$\vec{M}_{2,q} = \sum_{j=1}^f \vec{M}_2^C + \overrightarrow{O_2 O_{q,2}} \times \vec{F}_2^T + \vec{M}_3^q$		
$\vec{M}_1^T = \sum_{j=1}^f \vec{M}_{1,j} + \overrightarrow{O_B O_1} \times \vec{F}_1^T + \vec{M}_2^T$	$\vec{M}_2^T = \sum_{j=1}^f \vec{M}_{2,j} + \overrightarrow{O_1 O_2} \times \vec{F}_2^T + \vec{M}_3^T$	$\vec{M}_3^T = \sum_{j=1}^f \vec{M}_{3,j} + \overrightarrow{O_2 O_3} \times \vec{F}_3^T$
$\ \vec{M}_2^T - \vec{M}_2^T \cdot z_2\ = k_q \cdot \theta_q \cdot \frac{EI}{l_2}$		$\ \vec{M}_3^T - \vec{M}_3^T \cdot z_3\ = k_q \cdot \theta_q \cdot \frac{EI}{l_3}$
$GJ\varepsilon_1 = l_1 \left({}^B T_1 \vec{M}_1 \right) \vec{z}_B$	$GJ\varepsilon_2 = l_2 \left({}^1 T_2 \vec{M}_2 \right) \vec{z}_B$	$GJ\varepsilon_3 = l_3 \left({}^2 T_3 \vec{M}_3 \right) \vec{z}_B$

(a) Ecuaciones de CRT aplicadas a las secciones de Ruan

Adicionales
$T_{k,i} = \begin{bmatrix} \cos(\theta_{k,i}) & 0 & \sin(\theta_{k,i}) & \gamma_{k,i} l_j \sin(\theta_{k,i}) \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_{k,i}) & 0 & \cos(\theta_{k,i}) & \gamma_{k,i} l_j \cos(\theta_{k,i}) \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $z_i = {}^B A_i \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$ $\overrightarrow{O_{j,i:i-1}} = {}^B A_i \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \overrightarrow{OP_j}$ ${}^k A_{k+1} = R_z(\phi_j) T_{k,0} T_{k,1} T_{k,2} T_{k,3} R_z(-\phi_j) R_z(\epsilon)$ $\overrightarrow{O_i O_{q,i}} = {}^B A_i R_z(\Phi_j) T_{k,0} \dots T_{k,q} R_z(-\Phi_j) R_z(\varepsilon) \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} - {}^B A_i \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ $\overrightarrow{O_i O_{i+1}} = {}^B A_{i+1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} - {}^B A_i \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$

(b) Ecuaciones adicionales para el modelo cinemático de Ruan

Figura 7.8: Modelo cinemático para el robot Ruan

No obstante, para completar las ecuaciones es necesario introducir una serie de ecuaciones adicionales que se incluyen en la figura 7.8b. Además de incluir la expresión concreta que permite obtener las MTH en función de los cálculos resultantes del resto del modelo, se incluyen una serie de ecuaciones geométricas. Al igual que en el robot MACH-I, es necesario definir las distancias entre algunos sistemas de referencia:

- $\vec{O}_{j,i:i-1}$: es el vector que une el origen del cable j para una sección con la siguiente, referido a la primera.
- $O_i \vec{O}_{i+1}$: es el vector que une los centros de los discos que delimita una sección y su consecutiva, que se calcula a partir de la diferencia de matrices.
- $O_i \vec{O}_{q,i}$: es el vector que une el origen de una sección con la articulación ficticia q que define el modelo PRB.

La combinación de estas ecuaciones es la que permite definir el modelo cinemático completo para el robot.

7.2.3. Ejemplo de robot PCC: Pilory



Figura 7.9: Pilory, un RHAC continuo

El último ejemplo que se va a tratar como ejemplo para esta guía-tutorial se corresponde con el robot Pilory, un robot continuo de discos pivotantes desarrollado en el CAR. [29] Una vez más se debe comenzar desde el principio para seguir rigurosamente los pasos del procedimiento.

Para tomar la decisión sobre si es continuo o discreto, es necesario fijarse en la estructura interna del robot, que se puede observar en la figura 7.9b. Es una sucesión de discos unidos por tendones donde un disco pivota sobre el anterior para generar la curvatura. En concreto, el robot tiene cuatro secciones, cuyo disco final es un disco pasivo. Las secciones tienen 20, 12, 12 y 8 discos respectivamente. [29] Cada sección es actuada por cuatro cables.

Utilizando una vez más el criterio definido para determinar la continuidad o no de un robot, no existe una cadena cinemática de articulaciones discretas que permita modelar el robot mediante el método de Denavit-Hartenberg, teniendo que optar por la rama continua del gráfico. No obstante, otra pista que permite descartar este modelo es la discusión sobre las longitudes de los cables que se hizo en la sección 7.1.2. La existencia de discos pasivos complica la posibilidad de obtener las longitudes de los cables directamente con la MTH, necesitando conocer la forma concreta que adoptan los discos entre sí.

La siguiente cuestión a plantear es sobre la aplicación de la hipótesis PCC. Para ello, se pueden analizar los criterios indicativos que se han empleado en otras ecuaciones. Pilory no tiene un núcleo que se doble, reduciendo enormemente el peso del robot, que tiene un peso específico muy bajo, además de reducir los problemas de elasticidad. Los discos tampoco añaden peso al robot, al haber sido construidos mediante impresoras 3D.

Todas estas características parecen apoyar la adopción del modelo PCC. Visualmente, en el movimiento del robot no parece que su curvatura se aleje mucho del arco de una circunferencia para cada sección. Además, como se ha ido discutiendo, merece la pena desarrollar en primer lugar este modelo antes de tratar de utilizar sistemas más complejos. Tampoco es un robot blando, que pueda dirigir más los modelos hacia sistemas dinámicos.

Una vez determinado que se va a emplear un modelo PCC, las decisiones a considerar tienen un cariz algo más arbitrarias. En este caso, se busca un modelo lo más simple posible. La siguiente pregunta que se plantea en la guía-tutorial es sobre las velocidades y su implicación en el modelo. Constructivamente, el robot Pilory utiliza para enrollar los cables motores paso a paso, que en la información que devuelven a los controladores se utiliza la posición (en pasos, fácilmente traducible a posición angular) del motor. De esta forma, se dispone de información en posición, y no existen razones adicionales que parezcan favorecer el empleo de un modelo diferencial.

Con esto queda claro que se va a utilizar un modelo directo-inverso basado en la hipótesis de curvatura continua. No obstante, en esta casuística existía un planteamiento más que hacerse, relativo a las ecuaciones de las longitudes de los cables. Esta pregunta es relativa a la potencia computacional de la que se dispone para realizar los cálculos del modelo cinemático.

Una vez más, se recurre a la información disponible sobre la construcción del robot. [29] Se utilizan unos circuitos drivers que se localizan en el armario electrificado del robot, que gestionan individualmente los motores. En ellos, se ha programado un procedimiento sencillo que, a partir de los comandos recibidos, se generan los impulsos eléctricos necesarios para mover los motores paso a paso. No obstante, los comandos son inmediatos.

Toda la gestión de más alto nivel se realiza de forma externa: el robot debe conectarse a través de un cable USB que establece una conexión serial entre el controlador interno del robot y un ordenador externo, encargado de lanzar los comandos al robot. Los ordenadores externos no tienen tantas limitaciones de computación como un sistema embebido. Por tanto, se considera que el ordenador externo tiene potencia computacional suficiente como para gestionar el modelo cinemático completo, sin necesidad de realizar una aproximación lineal como la propuesta por el modelo LSK.

Con esto, se pueden recurrir a las ecuaciones propias del modelo PCC. En la sección 4.2 se ha realizado el desarrollo matemático detallado de todos los requisitos del modelo. En este caso, las ecuaciones independientes del modelo cinemático directo y del inverso se pueden adoptar tal cual.

En concreto, las ecuaciones que relacionan las longitudes de los cables con los parámetros de curvatura para el modelo inverso son válidas independientemente del robot. También las ecuaciones que definen los parámetros de curvatura con el estado final de la sección (en este caso, del algoritmo Natural-CCD se requiere como datos las coordenadas x , y y z) son válidas sin necesidad de introducir ningún cambio.

Únicamente se deben considerar algunos cambios en las ecuaciones para el modelo directo. En la sección 4.2 se desarrollaron las ecuaciones dependientes del robot suponiendo que cada sección tenía tres cables activos para la misma. El Pilory, como se ha mencionado previamente, tiene cuatro cables por sección. Esto requiere algunos cambios ligeros en las ecuaciones, que se han reflejado en la figura 7.10

Como se puede observar, la longitud del arco que correspondería con un hipotético cable central sigue siendo la media entre las longitudes de todos los cables. Los parámetros de curvatura se modifican siguiendo el mismo desarrollo que el realizado para tres cables, pero realizando una sustitución adicional. Con ello, se puede obtener ϕ y κ , que a su vez permiten calcular la longitud real del núcleo.

Modelo inverso

$$\begin{aligned}\phi &= \tan^{-1} \left(\frac{y}{x} \right) \\ \kappa &= \frac{2\sqrt{x^2 + y^2}}{x^2 + y^2 + z^2} \\ \theta &= \begin{cases} \cos^{-1}(1 - \kappa\sqrt{x^2 + y^2}), & \kappa > 0 \\ 2\pi - \cos^{-1}(1 - \kappa\sqrt{x^2 + y^2}), & z \leq 0 \end{cases}\end{aligned}$$

Modelo directo

$$\begin{aligned}l_c &= \frac{l_1 + l_2 + l_3 + l_4}{4} \\ \ell &= l_i + \theta d \cos \phi_i \\ \phi &= \tan^{-1} \left(\frac{l_4 - l_2}{l_3 - l_1} \right) \\ \kappa(\mathbf{q}) &= \frac{(l_1 - 3l_2 + l_3 + l_4) \sqrt{(l_4 - l_2)^2 + (l_3 - l_1)^2}}{d(l_1 + l_2 + l_3 + l_4)(l_4 - l_2)}\end{aligned}$$

Longitudes de cables

$$l_i = 2 * n * \sin \left(\frac{\theta}{2n} \right) * \left(\frac{1}{\kappa} - d \sin(\phi + \phi_i) \right)$$

$$T_{indep.}(l) = \begin{bmatrix} \cos \phi \cos \kappa l & -\sin \phi & \cos \phi \sin \kappa l & \frac{\cos \phi(1-\cos \kappa l)}{\kappa} \\ \sin \phi \cos \kappa l & \cos \phi & \sin \phi \sin \kappa l & \frac{\sin \phi(1-\cos \kappa l)}{\kappa} \\ -\sin \kappa l & 0 & \cos \kappa l & \frac{\sin \kappa l}{\kappa} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 7.10: Ecuaciones con el modelo cinemático completo para el robot Pilory

El conjunto de todas las ecuaciones de la figura 7.10 constituyen, por tanto, el último modelo cinemático de robots reales del CAR que se proponen en este trabajo.

8 | VALIDACIÓN DE UN MODELO CINEMÁTICO SOBRE ROBOT REAL: MACH-I

En el capítulo anterior se ha realizado una propuesta teórica, en forma de guía-tutorial, con algunos pasos e indicaciones que pueden ayudar a un investigador novel a introducirse en el modelado de robots hiperredundantes, proponiendo tipos concretos de modelo en función de la morfología y las características del robot a modelar. Además, se han propuesto tres ejemplos con robots del CAR que permiten visibilizar la forma en la que esta guía debería ser aplicada.

Los modelos tratados en los capítulos de este trabajo han sido ya verificados reiteradamente en la bibliografía académica, con muchos ejemplos entre los trabajos citados. No obstante, por completitud de este trabajo, se ha querido probar al menos uno de los modelos del capítulo anterior (para no exceder el trabajo previsto para un TFM) en alguno de los robots del CAR.

En concreto, por su robustez y por la disponibilidad de los mismos dentro del departamento, se ha optado por realizar las pruebas en el MACH-I. Para ello, ha sido necesario programar las ecuaciones propuestas en la sección 7.2.1 y proponer una serie de pruebas que comprueben que el modelo cinemático cumple con los estándares mínimos que certifiquen su validez.

8.1. Antecedentes

Para lograr los objetivos introducidos es necesario conocer en primer lugar las herramientas asociadas al MACH-I de las que se disponen inicialmente. El robot, construido inicialmente por Iván Rodríguez [37] es completamente funcional. Mediante los trabajos de Rodríguez, Alcolea y Martín-Barrio [37, 26, 1] se dispone de una documentación abundante sobre el mismo. Tanto a nivel de detalles constructivos, como a nivel software: en un repositorio de GitHub se encuentran recogidos todos los programas que contienen el funcionamiento básico del robot, incluyendo los programas de la Arduino que se encuentran en la torre electrificada del robot y la interfaz gráfica de comunicación que se emplea desde un ordenador externo.

La interfaz gráfica es una herramienta de gran utilidad de cara a manejar el robot. Gestiona internamente las comunicaciones con el robot a través de un canal serial entre Matlab y la Arduino del robot. Dispone de tres pestañas: una para realizar las conexiones con la Arduino (figura ??), otra para realizar la calibración del robot gestionando los motores individualmente (figura ??) y otra pestaña que permite simular y modificar la configuración del robot real (figura 8.1a).

Lógicamente, el empleo de estas herramientas implica que existe un modelo cinemático preparado ya para este robot. Además, su uso combinado con realidad virtual [26] también hace necesario que existan algunas ecuaciones que rijan la configuración del robot para estas aplicaciones. En efecto, analizando la bibliografía disponible del robot, se observa que existe un modelo cinemático inverso programado en la interfaz.

En concreto, las ecuaciones se basan en el uso de dos parámetros: α y β para cada sección. α representa el ángulo respecto al eje x de la sección (definido igual que en la sección 7.2.1) del eje de giro para el disco final respecto al inicial. Por tener juntas Cardan de dos grados de libertad,



Figura 8.1: Interfaz gráfica para el manejo del MACH-I

dicho eje se encontrará siempre en el plano de la sección. El ángulo β representa en cambio el ángulo de giro respecto a ese eje.

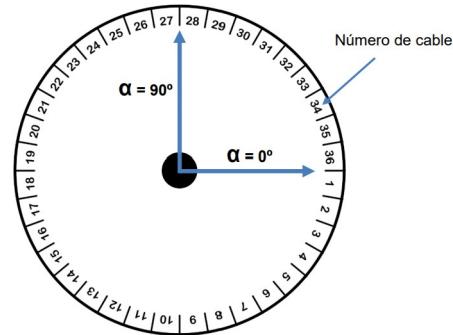


Figura 8.2: Definición de α respecto al sistema de referencia de la sección. [1]

Para obtener la MTH correspondiente a estos valores, se emplea la fórmula de Rodrígues, también introducida en este trabajo en la ecuación (7.5).

Por otra parte, para completar este modelo se utilizó una interfaz distinta para el algoritmo

Natural-CCD. Desarrollada por Martín-Barrio, se trataba originalmente de un simulador que permitía probar el algoritmo con una configuración de robot definida por el usuario (pudiendo elegir distintos tipos de articulaciones y un número arbitrario de secciones). De esta forma, se incorporaba el control para llevar el robot a un punto determinado.

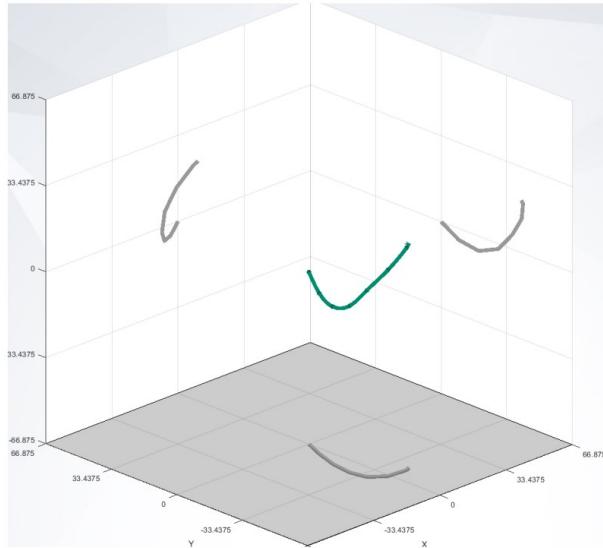


Figura 8.3: Interfaz integrada para usar Natural-CCD con MACH-I. [1]

Así, el conjunto de estas herramientas constituyen un modelo cinemático validado para el MACH-I. Su precisión se ha verificado experimentalmente, y ha permitido realizar aplicaciones reales con él. No obstante, también presenta algunas desventajas: en primer lugar, la interfaz gráfica creada y el modelo cinemático primordial utilizan como parámetros básicos de entrada α y β que son definidos virtualmente y no tienen una traducción directa a una articulación real. Además, las dos interfaces (la que controla la calibración y la que usa el natural CCD) son independientes, y se debe combinar el uso de ambas dos.

Además, el modelo cinemático implementado por ambas interfaces sólo considera las ecuaciones inversas. En ningún caso se ha programado el modelo cinemático directo que permita conocer la posición real del robot en cada momento.

En ese sentido, se quiere proponer la implementación de un modelo cinemático siguiendo la guía tutorial del capítulo anterior. En primer lugar, porque permite sistematizar el modelado cinemático: las ecuaciones actuales del MACH-I siguen una metodología particularizada al robot, mientras que el tutorial pretende abordar ejemplos más generales. En segundo lugar, porque usa variables que tienen un sentido físico real: son los grados que gira cada eje de la junta Cardan, las articulaciones físicas que usa el robot real. Finalmente, porque se va a incorporar un modelo cinemático directo que permita completar el modelo.

No obstante, el modelo cinemático existente resultará de gran utilidad de cara a la validación del nuevo modelo aquí propuesto. Conocida su precisión y habiendo sido verificado, si ambos modelos devuelven los mismos comandos para el robot, podrá considerarse que el modelo aquí propuesto es correcto.

8.2. Arquitectura desarrollada para la validación

Una vez determinado que se iba a implementar un modelo cinemático alternativo al existente, era necesario determinar el enfoque que se iba a adoptar. Como se ha mencionado, existían algunas herramientas ya creadas por otros estudiantes que servían para comunicarse y controlar al MACH-I. Había dos interfaces gráficas (una original para control y calibración y otra para el uso del Natural-CCD). Una posibilidad era aprovechar estas interfaces y sustituir el modelo cinemático existente en ellas. La otra, era construir una arquitectura alternativa desde cero.

La ventaja de aprovechar la arquitectura existente era evidente: gráficamente estaba muy depurada, y muchas funcionalidades ya estaban incorporadas. Sería una construcción sobre capas previamente existentes. No obstante, también tenía sus inconvenientes. El código de ambas interfaces utilizaba un gran número de variables globales, y tenía muchas interdependencias. Esto era especialmente patente en la aplicación para el Natural CCD. El código era tremadamente complejo, puesto que incorporaba muchas funcionalidades para simulación de distintos robots que no eran necesarias para el alcance de este trabajo.

Dada la baja legibilidad de algunas partes del código y que se trataban de aplicaciones muy interdependientes que tenían funcionalidad plena, el riesgo de modificarlas y editar sobre ellas era muy alto. Con el fin de asegurar que se mantuviera ese buen funcionamiento, y dado que exigía modificar muchas funciones que no eran relevantes para este proyecto, finalmente se determinó plantear una arquitectura separada.

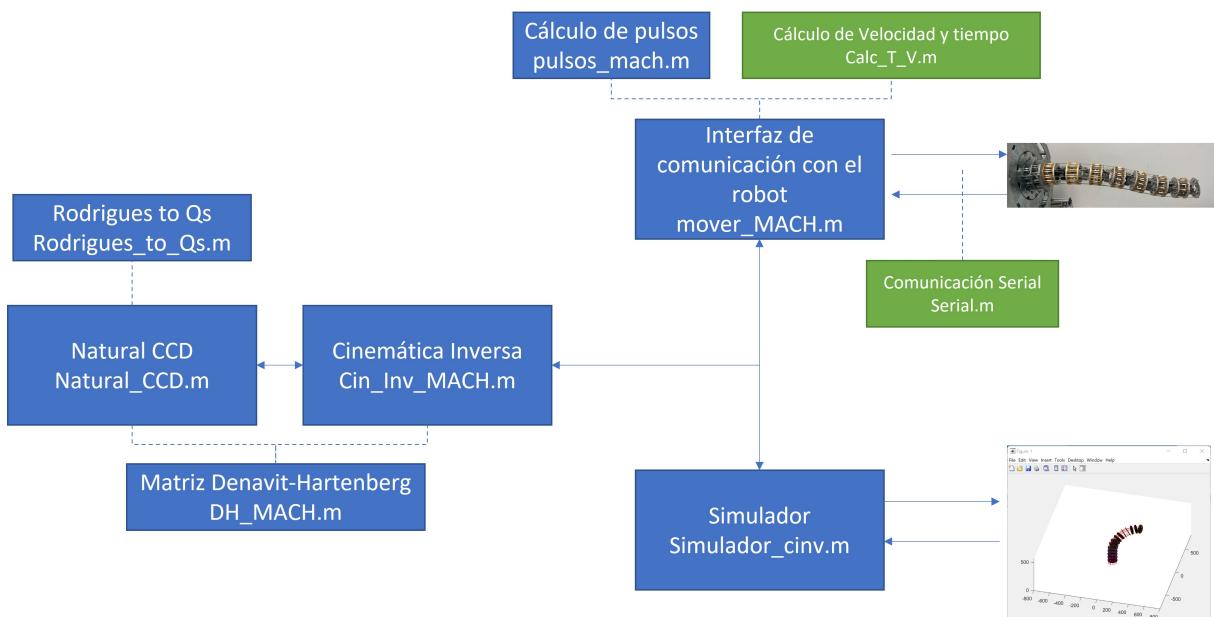


Figura 8.4: Arquitectura de código generada para la validación del modelo cinemático de MACH-I. Elaboración propia

Como el objetivo de este proyecto es únicamente validar el modelo cinemático planteado a partir de la guía-tutorial, el aspecto gráfico no es tan importante, de modo que se planteó una arquitectura sencilla, basada en funciones de Matlab bien organizadas, y que en un futuro permitiera integrarse en las interfaces gráficas si se considerara relevante, pero que fuera un sistema básico para el objetivo de la validación.

Esta arquitectura sencilla se ha recogido en la figura 8.4. En ella se ha marcado en azul

las funciones de Matlab generadas explícitamente por el autor de este trabajo. No obstante, ha habido algunos módulos que sí se han aprovechado en esta nueva arquitectura. En concreto, dado que no se han modificado los programas de los drivers internos de los motores del robot, era especialmente útil aprovechar los scripts que manejaban esa comunicación serial, y que traducían los comandos de los datos de los scripts del modelo a señales que entendiera el robot. Estos módulos, que manejan la conversión a velocidades y tiempos y los envíos y recepción de comunicaciones a la Arduino, se han representado de color verde en el cuadro con la arquitectura.

A continuación, se procede a analizar brevemente la funcionalidad, las entradas y las salidas de cada función/script, con el fin de mejorar la comprensión del desarrollo realizado:

- **Rodrigues to Qs**: esta función de Matlab es un recurso auxiliar que se utiliza en el Natural CCD para convertir la información de θ y el eje de rotación a coordenadas articulares. Las entradas son precisamente θ , un vector del eje de rotación y la matriz de rotación que acumula las orientaciones de iteraciones previas. Las salidas son las coordenadas articulares. La metodología se basa en el juego de proyecciones explicado en la sección 7.2.1.
- **Matriz Denavit-Hartenberg**: se trata de otra función auxiliar, llamada varias veces por las otras funciones. Es la que genera la matriz de Denavit-Hartenberg (su valor de salida), a partir de las coordenadas articulares y teniendo en cuenta en qué sección se halla para las traslaciones correspondientes.
- **Natural CCD**: Función que, recibiendo como entrada el punto final que se desea alcanzar y el tipo de robot que se va a manejar (en este caso la cadena 'MACH'), devuelve los vectores con las coordenadas articulares para cada sección, un error si no se ha logrado alcanzar el punto y el número de iteraciones que ha tardado en converger el algoritmo. En la parte superior del código de la función vienen definidas algunas variables importantes para el manejo del código. Se define el error máximo que admite para la distancia euclídea entre el extremo del robot y el punto deseado, el valor máximo admisible de giro en cada iteración, el número máximo de iteraciones y el ángulo máximo que puede rotar la articulación de la junta Cardan en cualquier dirección.

El algoritmo está basado en la construcción realizada por Andrés Martín-Barrio [22], si bien se han adaptado algunos puntos a la realidad de este proyecto. Por ejemplo, la conversión de los parámetros ángulo-eje de giro a coordenadas articulares, así como la revisión del ángulo máximo admisible para cada articulación.

- **Cinemática Inversa**: Este script, a partir de un punto, devuelve las longitudes, las coordenadas articulares y las matrices de transformación de cada sección que se necesita para que el robot alcance su objetivo. Constituye el script que contiene el modelo cinemático completo del MACH-I conformado a partir de la guía-tutorial.

Es la función encargada de llamar al algoritmo Natural CCD, y discriminar cuando se ha dado un error. A partir de las coordenadas articulares que recibe del algoritmo, emplea las ecuaciones dependientes del MACH-I para la cinemática inversa desarrolladas en la sección 7.2.1, que utilizan las matrices de transformación homogéneas para calcular las longitudes de los cables, teniendo también en cuenta sus tramos rectos.

- **Interfaz de comunicación con el robot**: es un script que retoca aquellos elementos de las interfaces de comunicación desarrollados por Rodríguez y Alcolea [37, 1] para adaptarlos a las necesidades de este modelo. En concreto, aprovecha los módulos creados por ellos para calcular las velocidades de los motores y el tiempo del movimiento, así como la

comunicación serie con la Arduino.

No es una función como tal. Se trata de un script que, definiendo un punto en la parte superior del código, lleva primeramente al robot hasta ese punto. Posteriormente, tras esperar un tiempo, devuelve al robot a su posición original. Estas pruebas son suficientes para la validación del modelo cinemático planteado.

- **Cálculo de pulsos:** dada la infraestructura de comandos definida por los desarrolladores del robot, la Arduino de a bordo no recibe las longitudes de los cables, sino que recibe los pulsos que debe mandar a los motores paso a paso. Esa función existía en la arquitectura original de códigos del MACH-I, pero existe una diferencia fundamental que ha obligado a adaptar el código.

El código original omitía las longitudes fijas de los cables. Esto es, dada una referencia con el robot completamente estirado, solo empleaba las diferencias de longitud que se producían por los cambios de orientación en las secciones, y a partir de ahí generaba los pulsos. En este trabajo, no obstante, se ha preferido mantener el modelo fiel a la realidad, teniendo en cuenta para el modelo la longitud total de los cables, por si este dato fuera de interés para el desarrollador.

Por tanto, ha sido necesario realizar una pequeña adaptación: se han modificado las referencias de las longitudes de los cables con el robot completamente estirados, para asegurar que los comandos de los pulsos se obtengan en el formato definido originalmente en el código de la Arduino de a bordo, permitiendo un buen funcionamiento del robot en su conjunto.

- **Simulador:** en paralelo a la rama que se encarga de las comunicaciones con el robot real, se ha generado también un script de Matlab encargado de simular el comportamiento del MACH-I, para poder observar los resultados del modelo de forma gráfica antes de cargarlos en el robot (previniendo así posibles fallos).

Es un simulador sencillo: a partir de los datos devueltos por la cinemática inversa para el punto que la función admite de entrada, se generan tanto las superficies de los puntos de los discos intermedios como los puntos finales para los cables, dibujando todos ellos en la misma figura. Es una función que tarda un poco en ejecutarse, por la demanda gráfica que supone para el ordenador.

El conjunto de estas funciones de Matlab constituyen todas las herramientas necesarias para validar el modelo cinemático inverso del robot MACH-I. Asimismo, como se verá en la sección 8.4, serán de vital importancia para obtener el modelo cinemático directo para el robot.

8.3. Simulador para validación de la cinemática inversa

Una vez se dispone de todas las herramientas necesarias, es posible realizar los trabajos propiamente dichos que servirán como validación del modelo planteado. Por seguridad con el robot, se ha realizado en primer lugar una validación puramente en simulación. De esta forma, se intenta proteger al robot para el momento de trasladar los comandos de movimiento a la Arduino.

Para ello, el objetivo es el siguiente: verificar si los comandos generados por la combinación de herramientas desarrolladas en el apartado anterior son iguales (o parecidos con un error despreciable) a los comandos obtenidos mediante el empleo de las herramientas originales para el MACH-I. Al tratarse de un modelo cinemático plenamente verificado, se puede tener la confianza

de que sus resultados son válidos, y se pueden introducir sin problema en el robot.

Un primer paso para ello fue el desarrollo del simulador gráfico. El manejo de datos sin interpretar puede ser complicado. En cambio, teniendo una perspectiva visual de la configuración del robot se pueden percibir posibles errores en el código. Para ello, se desarrolló el archivo Simulador_cinv.m. (Ver figura 8.5) Con su ayuda se han corregido algunos errores de código, para que los resultados tengan sentido.

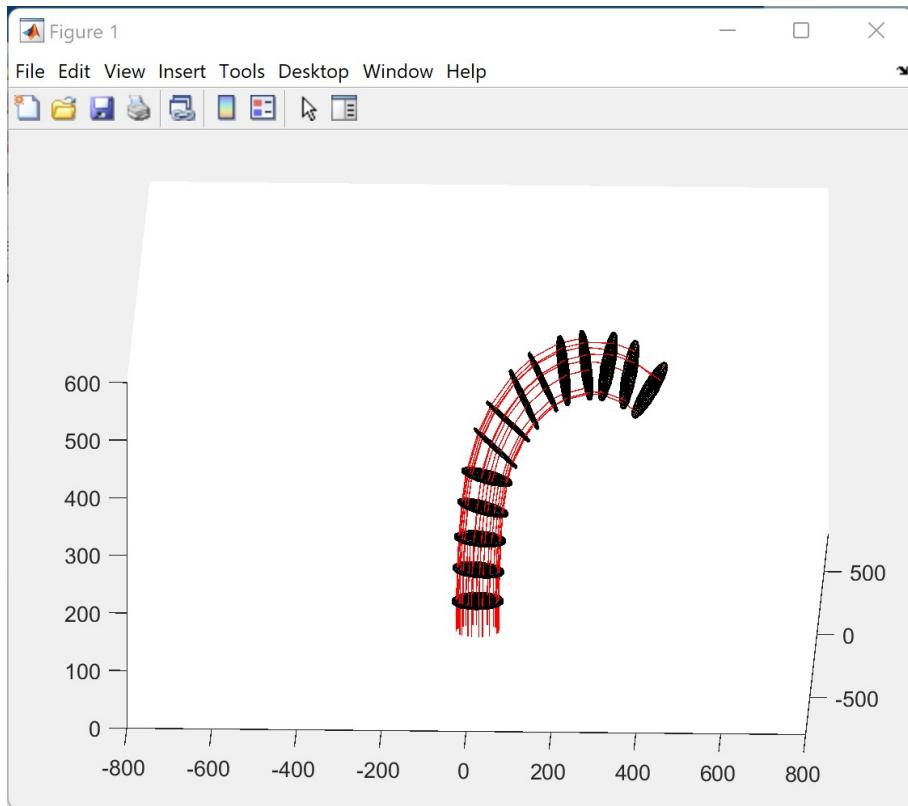


Figura 8.5: Ejemplo de ejecución del simulador del MACH-I para el punto [400 100 400]
Elaboración propia

Asimismo, el simulador es útil para recibir información sobre el campo de trabajo que puede llegar a abarcar el robot. Al haber limitado los ángulos máximos de cada articulación, no todos los puntos serán alcanzables, o el Natural-CCD puede caer en un mínimo local del que no sea capaz de salir. De esta forma, el simulador permite verificar rápidamente estas situaciones, revisando paso a paso la convergencia del algoritmo. Introduciendo un punto objetivo en el simulador antes de mandarlo al robot real se logra dotar al robot de una capa de seguridad adicional ante averías.

El siguiente paso fue la verificación del modelo contra la versión de la cinemática ya existente. Para ello, es necesario comparar los resultados, pero necesitan realizarse algunas adaptaciones. Las funciones existentes del modelo cinemático antiguo devuelven la información sobre los cables directamente en los pulsos que se mandan a los motores. Las longitudes (diferenciales, en el sentido de plasmadas como el cambio de longitud respecto al origen) quedan como variables internas de las funciones.

Por ello, se desarrollaron las funciones que traducían las longitudes a pulsos y viceversa. De esta forma, en el nuevo modelo se acometía también la transformación de longitudes a pulsos para poder comparar entre medidas equivalentes. Esto elimina también las discrepancias exis-

tentes en el criterio de definición de longitudes (en este modelo se considera la longitud total de cable, en anteriores, solo la componente variable). Al considerar diferencias de longitudes, las representaciones se homogeneizan.

Inicialmente se escogió un punto sencillo, en el plano $y = 0$. En los puntos contenidos en ese plano, los valores angulares de giro de la junta Cardan en torno al eje X , es decir, q_1 , serán nulos en todo caso. Para garantizar que la configuración del robot sea el mismo en la prueba de ambos modelos cinemáticos, se ha de buscar una relación entre los parámetros articulares y los parámetros alternativos α, β . Así, denominando z' al vector $[0 \ 0 \ 1]$ transformado por la MTH de la sección:

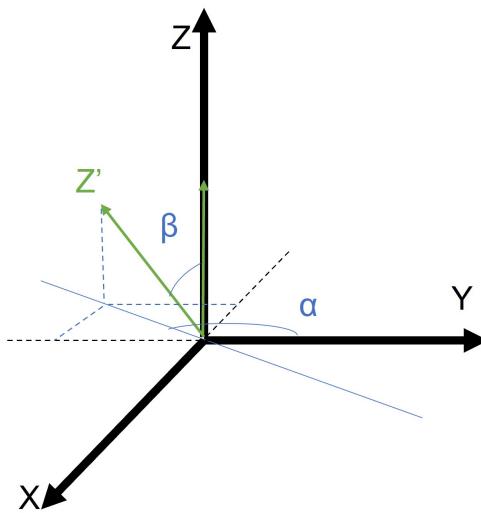


Figura 8.6: Relación entre parámetros de ambos modelos cinemáticos. Elaboración propia

$$\alpha = \arctan \left(\frac{z'(2)}{z'(1)} \right) \quad (8.1)$$

$$\beta = \arctan \left(\frac{\sqrt{z'(2)^2 + z'(1)^2}}{z'(3)} \right) \quad (8.2)$$

Realizando los procedimientos correspondientes con ambos modelos cinemáticos y comparando los pulsos obtenidos con cada uno y convirtiéndolo a longitudes, se obtiene un error máximo (es decir, el mayor error que se comete en cualquiera de los cables) nulo, cuando se realiza un redondeo en el cálculo de los parámetros alfa y beta, para evitar errores por decimales de las funciones trigonométricas.

La comprobación mediante este punto sirvió para detectar un error en el código por un signo, así como para ajustar las longitudes fijas de las secciones, teniendo en cuenta que la diferencia entre los cables estirados y la posición deseada debe ser la misma en ambos modelos. No obstante, esta comprobación es insuficiente para determinar la validez del modelo. Es necesario realizar un análisis más amplio y para una muestra más variada del espacio.

Por ello, empleando el simulador, se generó una muestra aleatoria de 18000 puntos válidos (esto es, se descartan valores no alcanzables durante la generación de la muestra). Empleando estos puntos, se repitió el procedimiento de cálculo con ambos modelos, y se realizó la comparación. Aunque la diferencia entre modelos se haya calculado en pulsos, por los motivos expuestos,

los valores numéricos se han traducido con posterioridad en longitudes, por ser una variable más comprensible.

Se vuelve a analizar como variable determinante el error máximo en un cable: es decir, el mayor error que se comete en cualquiera de los 21 cables que intervienen en el MACH-I. En general, puede observarse que éste se da para las secciones finales del robot, lo que tiene sentido por la acumulación de error que se acumula al sumar todas las secciones. Para los 18000 puntos, haciendo la media del error máximo cometido, se obtiene una media de 0.9436 mm de error máximo para el cable, lo que supone un error menor a 1 mm en todos los cables del robot. El error se puede achacar a las funciones trigonométricas que se emplean y que acumulan un cierto error al acumularse.

Analizando también estadísticamente la mediana de la muestra analizada, se ve que queda en 0.7441 mm. Esto implica que la mayor parte de los puntos analizados están por debajo del error medio, pero que existen algunos valores discordantes que elevan más la media. Un análisis más detallado de los datos revela que el error se incrementa para valores más bajos de z , posiblemente por requerir ángulos más pronunciados. En todo caso, el error máximo de la muestra es de 3.8047 mm, que comparado con la longitud total del cable correspondiente es un error del 0.5 %.

A continuación se muestra el diagrama de caja y bigotes del error cometido en porcentaje respecto a la longitud del cable. Se trata de un análisis que tiene mejor en cuenta los errores generales del modelo: al fin y al cabo, el error máximo bruto

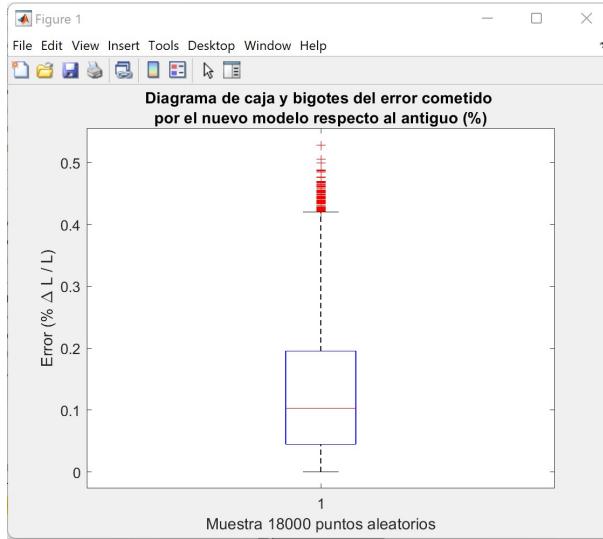


Figura 8.7: Diagrama de caja y bigotes del error cometido en porcentaje para la muestra analizada. Elaboración propia

Con este resultado, puede considerarse que ambos modelos ofrecen respuestas equivalentes para el mismo problema. Dado que el modelo cinemático anterior ya había sido validado con éxito, se puede extraer que el nuevo modelo cinemático inverso es perfectamente válido para el robot MACH-I, solventando con éxito el problema planteado.

8.4. Desarrollo de la cinemática directa

Una vez concluida la programación del modelo cinemático inverso, de una mayor sencillez, se procede a abordar el modelo cinemático directo. De éste, se dispone ya de las expresiones

correspondientes a las matrices de Denavit-Hartenberg de las secciones tomando como entrada los parámetros articulares del robot. No obstante, se necesita también la relación entre las longitudes de los cables y las coordenadas articulares.

Ya se trató en el capítulo anterior las dificultades de lograr esto para el MACH-I. Dado que su estructura equivalente es la de un robot paralelo PPP-3S, no es posible encontrar una ecuación analítica que refleje dicha relación. [28] Por ello, se han de plantear bien métodos numéricos o algoritmos de inteligencia artificial, que sean capaces de optimizar una relación estadística entre ambos conjuntos de variables.

En el departamento ya se ha trabajado y se ha tenido experiencia con el manejo de redes neuronales para resolver cinemáticas de robots paralelos. [13] La elección de este algoritmo frente a otras alternativas, tales como algoritmos genéticos se ha realizado por los buenos resultados de los mismos en materia de regresión, conocimientos previos del autor de este trabajo y la posibilidad de generar fácilmente distintas redes neuronales mediante la herramienta de la Toolbox para Deep Learning de Matlab. [27]

A pesar de que la Toolbox está especialmente preparada para configurar redes neuronales convolucionales para el tratamiento de imágenes, también existe la posibilidad de configurar a mano una arquitectura de red que sirva para predicciones y regresión. De cara a la propuesta de arquitectura, se comenzó trabajando a partir de los consejos recogidos en el trabajo de García Samartín [13]. Así, los parámetros fundamentales para la red neuronal se definieron como sigue:

- **Capa de entrada de parámetros:** la variable de entrada para la relación que se quiere encontrar son las longitudes de los cables. El vector de longitud se ha definido de 36 parámetros (aunque MACH solo tenga actualmente 21), por estar preparado para el caso de que el robot crezca. Para no tener que modificar los datos de cara a introducirlos a la red neuronal, se mantiene esta estructura vectorial, considerando que los valores no utilizados siempre serán cero.

Por otra parte, es de sobra conocido que para mejorar el funcionamiento de las redes neuronales conviene normalizar la información de entrada. Matlab permite configurar esto directamente, y se implementa una normalización en torno a cero (denominada en Matlab como 'z-score').

- **3 capas ocultas:** se utilizan tres capas ocultas, de tipo clasificadoras ('fully-connected layers'). Para calcular el número de neuronas, se manejaron algunos límites orientativos [13]. Siendo N_e el número de neuronas de la capa de entrada, N_c el número de neuronas para las capas ocultas y N_s el número de variables de salida:

$$\begin{aligned} N_c &< 2N_e = 2 * 36 = 72 \\ N_c &\approx \frac{2}{3}N_e + N_s = \frac{2}{3} * 36 + 14 = 38 \end{aligned}$$

Aunque se comenzó probando con capas de 38 neuronas, se verificó que el error era demasiado grande, y finalmente se optó por 60 neuronas por capa oculta.

- **Función de activación:** se ha optado por utilizar la función lineal rectificada o RELU, la opción típica a la hora de afrontar problemas de regresión y de predicción. Esta función de activación se ha incorporado detrás de cada capa oculta.
- **Capas de salida:** para preparar la salida de la red neuronal, se introduce una última capa clasificadora pero con la configuración posterior de la regresión. Matlab dispone directamente de una capa preparada para realizar predicciones numéricas, que se ha configurado

en este caso concreto para obtener los 14 parámetros de salida que constituyen todos los valores articulares del robot.

El resumen completo de la arquitectura de la red neuronal generada por la propia Toolbox de Matlab puede observarse en la figura 8.8.

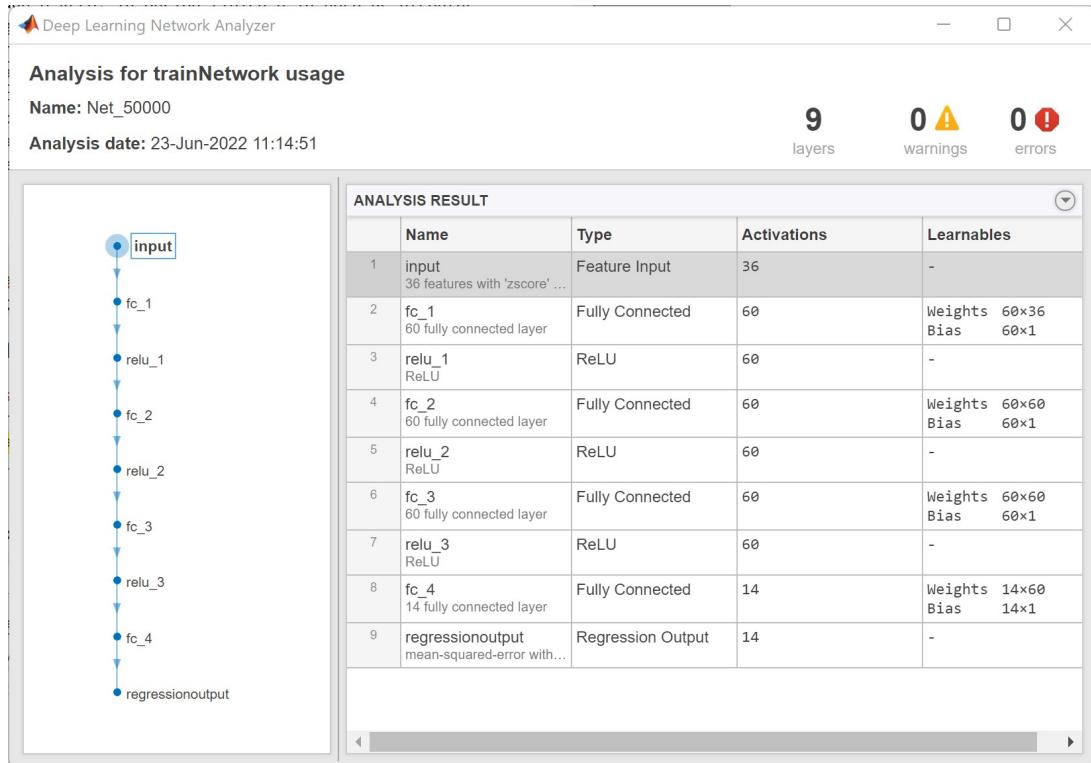


Figura 8.8: Arquitectura para las redes neuronales del modelo directo.

Una vez definida la arquitectura a emplear para la red neuronal, se requiere generar la base de datos para su entrenamiento y validación. La forma escogida para ello es emplear el simulador creado para el modelo cinemático inverso, de forma semejante a la utilizada para la validación de las ecuaciones inversas: generando puntos aleatorios en el espacio de trabajo posible para el robot, se comprueba en primer lugar si son alcanzables o no. Si son alcanzables, se almacena tanto el punto como los valores de la cinemática inversa (las longitudes de los cables y los parámetros articulares q_1 y q_2).

Esta base de datos será el que se emplee para generar los conjuntos de entrenamiento de la red. Así pues, utilizando un script automatizado de Matlab que obtiene puntos aleatorios, se han generado un total de 66103 puntos con sus correspondientes variables cinemáticas. La distribución en el espacio de dichos puntos puede observarse en la figura 8.9.

No obstante, queriendo optimizar el funcionamiento de la red y su validez, se ha querido probar con diferentes tamaños de muestras para el entrenamiento de las distintas redes neuronales. De esta forma, de este conjunto grande, se han tomado subconjuntos aleatorios de distintos tamaños: 1000, 2000, 5000, 10000, 20000, 30000, 40000, 50000 y 60000 puntos. Cada uno de estos subconjuntos ha sido empleado para generar una red neuronal con un entrenamiento distinto, y así poder comparar a posteriori el error de cada una de ellas.

Cada uno de los subconjuntos que se han empleado para el entrenamiento se han dividido, como es frecuente para el manejo de redes neuronales, en tres grupos de puntos. Un 70 % de

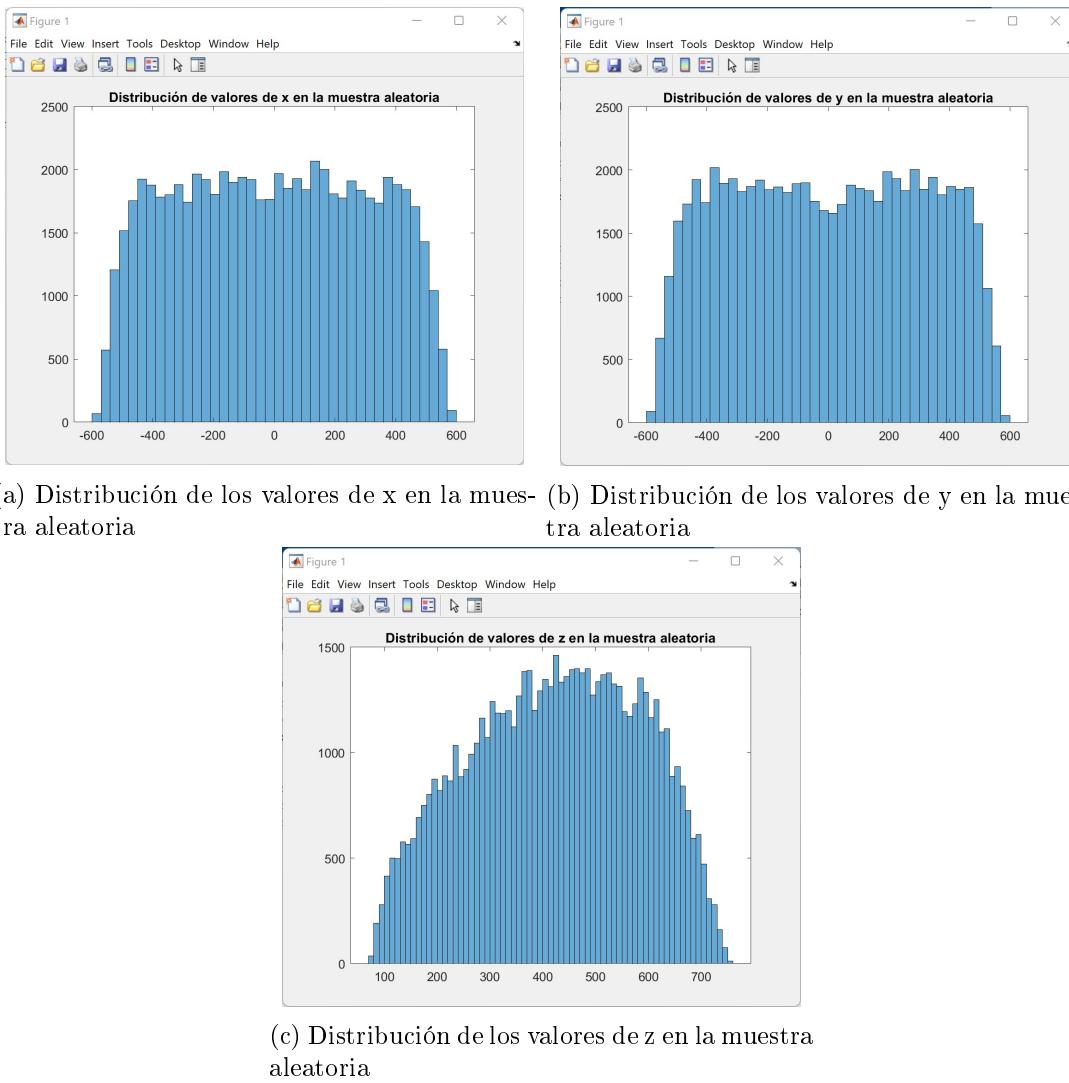


Figura 8.9: Distribución de los puntos para la muestra aleatoria de la red neuronal.

ellos se emplearían para aplicar el algoritmo de entrenamiento de la red. Un 15 % se emplearían para la validación del mismo, mientras que el 15 % restante queda para realizar los test de error posteriores.

Poniendo como ejemplo la red neuronal para 50000 puntos, tras configurar el entrenamiento (se empleó el algoritmo ADAM, de momentos adaptativos, incorporado por Matlab), se procedió a entrenar la red. En la figura 8.10 puede verse el resumen del entrenamiento que genera la Toolbox, en la que se presenta la evolución del error para el conjunto de entrenamiento y el conjunto de validación según avanzan las iteraciones del algoritmo.

Tras realizar el entrenamiento, se procedió al testeо de los resultados con el subconjunto de puntos aún no empleado. Para ello, se utilizó cada una de las redes entrenadas para predecir los valores de los parámetros articulares a partir de la longitud de los cables. Este resultado se comparó mediante el error cuadrático medio (típica medida de error en las regresiones) con los valores reales que se tenían en el conjunto de test.

Este valor es el que se ha empleado para comparar los distintos entrenamientos de la red neuronal entre sí. Conocidos los errores cometidos en el conjunto de test, es posible tener un

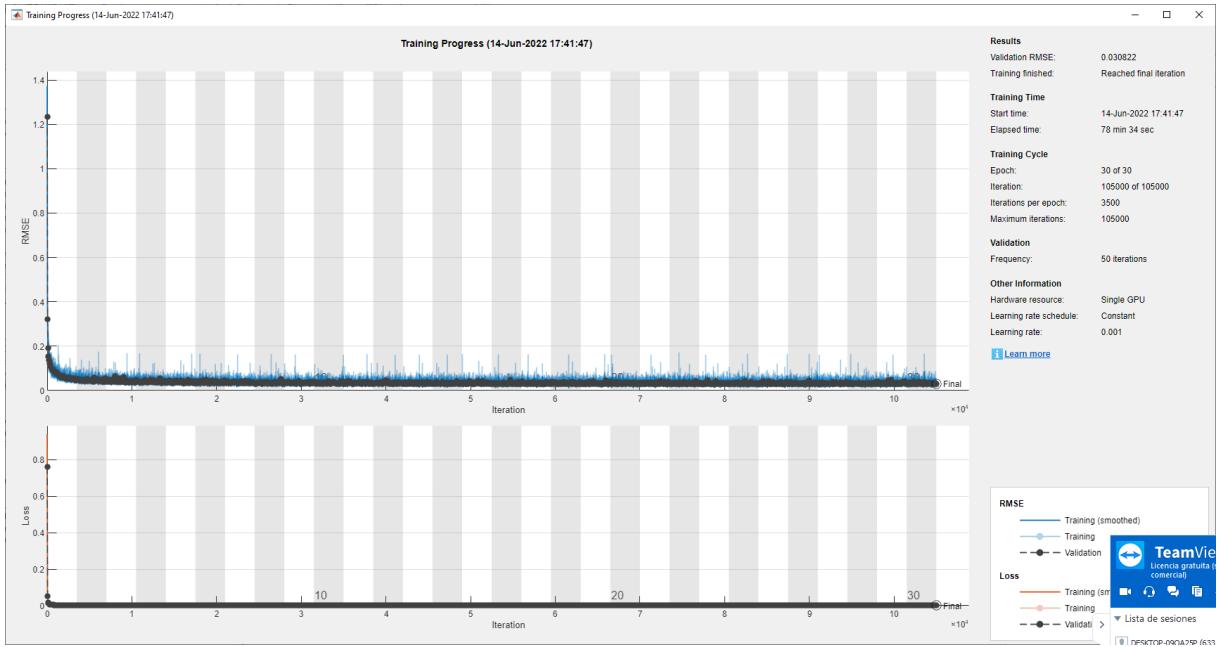


Figura 8.10: Entrenamiento para la red neuronal empleando el subconjunto de 50000 puntos

conocimiento orientativo sobre qué red neuronal es más precisa a la hora de establecer la relación longitudes de cable-parámetros articulares. Así, tras repetir el proceso entrenamiento-validación-prueba para los 9 conjuntos de entrenamiento, se han obtenido los siguientes datos:

Tamaño del conjunto	1000	2000	5000	10000	20000	30000	40000	50000	60000
Error cometido en prueba	3.6247e-04	1.7846e-04	1.6023e-04	3.1343e-04	8.4636e-05	7.3095e-05	7.7106e-05	6.8114e-05	6.9310e-05

Cuadro 8.1: Error cometido por la red neuronal contra el subconjunto de pruebas para distintos tamaños de entrenamiento

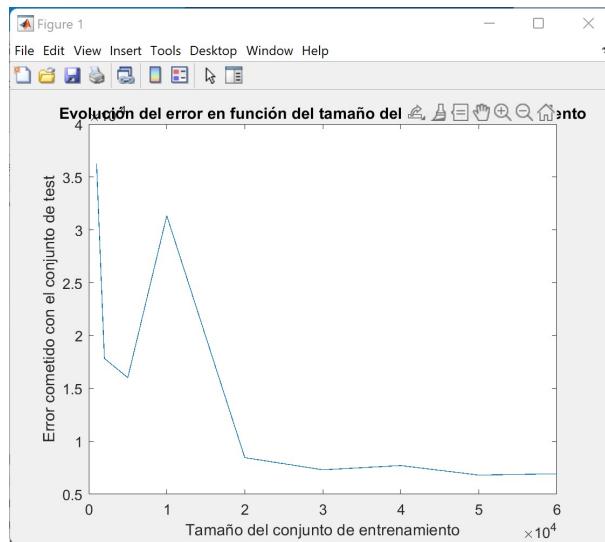


Figura 8.11: Variación del error en función del tamaño del conjunto de entrenamiento

Gráficamente también se puede observar que en los 50000 puntos para entrenamiento se encuentra un mínimo de entre todos los valores probados. Por ser el error tan pequeño (del orden

de las cienmilésimas) se puede considerar aceptable la predicción de esta red neuronal, siendo la escogida de entre todas las entrenadas para este paso del modelo cinemático directo.

Con esta herramienta puede darse por concluida la programación del modelo cinemático: ya se dispone de un modelo cinemático inverso, validado contra el modelo previo del robot. Asimismo, utilizando el simulador para generar una base de datos suficiente, se ha logrado crear una red neuronal que predice con éxito las relaciones entre longitudes de cables y parámetros articulares. Uniendo la red a las expresiones de las matrices de Denavit-Hartenberg programadas en la función de Matlab “DH_MACH” se dispone también de un modelo cinemático directo completo.

8.5. Pruebas sobre el robot real

Una vez se disponen de todas las herramientas cinemáticas creadas, y habiéndolas validado en simulación, se puede proceder a la implementación y prueba de las mismas sobre el robot real. Tal y como se explicó en la sección 8.2, ya se ha preparado la interfaz de comunicación para poder mandar comandos al robot y poder comprobar el buen funcionamiento del modelo.

Las pruebas se comenzaron en el plano descriptivo. Inicialmente, se utilizó la interfaz para mandar al robot puntos sencillos, que permitieran realizar un análisis cualitativo sobre los modelos cinemáticos y su funcionamiento. Por ello, se probaron dos movimientos, en el plano $y = 0$ y $x = 0$ que permitieron verificar que, en efecto, el trabajo realizado era bueno.

No obstante, se pretendía dar un paso más, y obtener datos numéricos que permitieran comprobar la validez del modelo. Para ello, se preparó un setup del optitrack, que permitiera obtener la posición relativa de la punta del robot respecto a su base, lo que constituiría una fuente de información de gran valor a la hora de disponer de un contraste numérico de la validez de los modelos.

Lamentablemente, a la hora de realizar las pruebas, fue necesario utilizar un alargador USB entre el ordenador y la Arduino, para compatibilizar la conexión ethernet al switch donde se conectaban las cámaras (con una longitud de cable reducida) con la conexión a la Arduino del robot. Al lanzar el comando, el movimiento realizado por el robot fue extraño en algunas de las articulaciones, provocando la rotura de uno de los cables del robot.

Contrastando esta información con los constructores del robot, se pudo saber que la conexión serial Matlab-Arduino falla en ocasiones cuando se utiliza este tipo de alargadores, corrompiendo los datos de los comandos, y que ya se había provocado algún movimiento extraño por esta razón anteriormente.

Este suceso provocó la interrupción de las pruebas de validación sobre robot real. Se comenzaron una serie de tareas de reparación del robot con el fin de tratar de culminarlas, pero no ha sido posible presentar el resultado antes de entregar esta memoria. Pese a todo, se considera que la validación en simulación y las pruebas descriptivas han dado una base suficientemente sólida como para considerar validado este nuevo modelo cinemático sobre el MACH-I.

9 | CONCLUSIONES

9.1. Conclusiones y aportaciones del trabajo

El trabajo pretendía aportar una sistematización ordenada de lo que supone el modelado cinemático de robots hiperredundantes. En concreto, se puede considerar que el objetivo principal del trabajo se ha cumplido. Dada mi experiencia personal al introducirme en este tema, en el que frecuentemente me encontraba perdido al leer bibliografía, considero que el esfuerzo ha sido grande a la hora de plantear el estado de esta cuestión.

Se puede considerar que el marco teórico aquí aportado contribuye a ordenar la estructura mental que requiere el modelado de estos robots, simplificando la aproximación de un nuevo investigador. Asimismo, la guía desarrollada es útil para una primera aproximación. Lógicamente, el investigador experimentado puede tener unos requisitos más concretos o desear abordar el problema desde un enfoque particular, pero incluso en este sentido, la guía aporta una serie de preguntas importantes que se deben considerar cuando se trabaja con RHACs.

Finalmente, el trabajo de aplicación y validación sobre robots del CAR es de gran importancia para no dejar este trabajo en un plano exclusivamente teórico. La posibilidad de revisar con un ejemplo los pasos necesarios para aplicar con éxito la guía y el modelo a un robot es siempre enriquecedora y puede simplificar el trabajo.

En cuanto a la validación con el MACH, se considera un éxito en tanto y cuanto el modelado aquí aportado tiene ciertas ventajas: primero, sigue un método sistemático como es Denavit-Hartenberg, que permite presentar unas instrucciones ordenadas para lograr un modelo robusto; en segundo lugar, los parámetros empleados son más parecidos a la realidad constructiva del robot, lo que siempre presenta ventajas de información para el desarrollador.

En conclusión, es posible afirmar que el trabajo consigue abordar con éxito los objetivos del trabajo, y que es una contribución importante para futuras personas que deseen trabajar en el campo de los robots hiperredundantes.

9.2. Impacto social y medioambiental del trabajo

Un trabajo final de máster debe reflejar, además del contenido técnico, una reflexión madura sobre el significado del mismo en el ámbito de la responsabilidad social y medioambiental. Al fin y al cabo, nuestra formación está dirigida a que en un futuro seamos buenos profesionales, lo que incluye este elemento ético.

En el tema social, uno de los impactos más importantes que tiene este trabajo es en el campo educativo. Es un trabajo que busca servir de puerta de entrada a futuros estudiantes y/o investigadores que se aproximen por primera vez al tema. Dado que la robótica hiperredundante y blanda es un campo en auge de la robótica, la contribución de este trabajo puede ayudar a formar de entrada a estos nuevos participantes.

Por otra parte, y con una perspectiva más amplia, los usos de la robótica hiperredundante

suelen ser, por ejemplo, tareas de inspección o acceso a ubicaciones remotas de difícil acceso para operarios humanos. Una contribución en este campo ayuda proteger a prospectivos operarios, previniendo que sea necesario que individualmente accedan a sitios potencialmente peligrosos.

Indudablemente, la robótica es un campo afectado por la discusión ética, y más con el creciente impacto de la inteligencia artificial. No obstante es un ámbito científico de gran relevancia, de forma que se tiene que seguir trabajando en una robótica social, es decir, que sea responsable con las demandas sociales del momento, y que no genere tensiones sociales (por ejemplo, por motivos laborales). En ese sentido, la robótica colaborativa con robots hiperredundantes es posible y puede contribuir en este ámbito.

Respecto al impacto medioambiental, pueden considerarse dos fuentes principales de impacto: la parte constructiva del robot, en su parte proporcional que ha afectado a este trabajo, y el consumo de los recursos energéticos necesarios para poder realizar los cálculos necesarios por ordenador. Especialmente intensivos han sido las operaciones necesarias para simulación, para generación de muestras y para el entrenamiento de las redes neuronales. No obstante, se ha intentado optimizar este consumo, aprovechando horas de baja demanda (principalmente, por la noche) para los entrenamientos y los muestreos largos.

9.3. Líneas futuras de trabajo

El trabajo realizado permite proponer algunas líneas de investigación futuras que permitan construir nuevos proyectos y avanzar en el conocimiento científico a partir de lo reflejado en esta memoria. Entre otras opciones, se destacan las siguientes posibles líneas futuras:

- En lo relativo al modelado de robots hiperredundantes, sería interesante buscar una frontera numérica de admisibilidad de la hipótesis PCC: a partir de los indicios recogidos en la bibliografía (tales como peso específico o fuerzas externas), sería interesante buscar si puede encontrarse una relación numérica que ayude como un discriminante adicional.
- Otra consecuencia derivada de este trabajo podría ser la elaboración de un simulador para RHACs que incorpore los modelos aquí desarrollados. De esta forma, dado que el grupo de investigación trabaja activamente con distintas morfologías de robots hiperredundantes, se pueda obtener a partir de unos parámetros de entrada un modelo virtual de un nuevo robot, sin necesidad de programar algo particularizado, al menos para pruebas preliminares.
- Centrándose en los robots del CAR, sería interesante revisar los modelos cinemáticos de los otros RHAC según los parámetros de esta guía. En concreto, sobre Ruan no se han podido realizar pruebas adicionales por requerir una reconstrucción integral. Así, plantear la construcción de un nuevo robot teniendo en cuenta desde el origen la guía tutorial propuesta puede ser de gran interés, validando si es de utilidad para estudiantes que se introducen en la cinemática hiperredundante.
- Por otra parte, el trabajo reafirma la importancia que tendría tratar de establecer un mecanismo de calibración automática para los robots hiperredundantes, dada la complejidad actual de tener que realizar una calibración manual. Para ello sería necesario abordar alguna modificación hardware para medir las tensiones de los cables, y posiblemente adoptar algún modelo dinámico que permita verificar si coincide con la esperada.

10 | PLANIFICACIÓN DEL TRABAJO

10.1. Presupuesto del trabajo

En la ingeniería es fundamental ajustar los costes reales del proyecto al presupuesto disponible para el mismo. Por ello, los presupuestos son una herramienta esencial que deben incorporarse a cualquier proyecto. La estimación económica para este trabajo se presenta a continuación, con tres partidas fundamentales: costes de personal, costes materiales y amortización.

Los costes de personal son aquellos debidos al empleo de recursos humanos para realizar el trabajo. En concreto, esta partida tendrá dos conceptos: el coste relativo al trabajo del alumno y el coste relativo al trabajo del tutor. Para ello, el primer paso es determinar el tiempo que ha dedicado cada una de las partes al proyecto y, conocido el coste horario de su trabajo, determinar el coste total.

En el caso del alumno, la estimación temporal debería coincidir con una ventana horaria de entre 300 y 360 horas, por tratarse de un trabajo fin de máster para el máster en automática y robótica. En concreto, se ha estimado para este trabajo un total de 400 horas, lo que supone un exceso de 40 horas respecto a lo establecido. En el BOE se puede encontrar el convenio de empresas de ingeniería que, para un titulado universitario, fija un salario anual de 26.065,07€, siendo la jornada laboral de 1792 horas anuales. Calculando el coste horario se tiene un valor de 14,55€/h, que multiplicado por las horas trabajadas devuelve un valor final de 5.820€.

En cuanto al profesor, se ha estimado un total de dedicación de 25 horas entre reuniones y tiempo dedicado a realizar correcciones del trabajo. El tutor es Catedrático en la Universidad Politécnica de Madrid, que en su portal de transparencia fija un salario anual de 48.072,04€. Dividiendo por la jornada laboral completa, se obtiene un coste de 26,82€/h, que para el total de horas suponen 670,65€.

El único elemento a amortizar es un ordenador portátil con valor de 1.500€, para el que se estima una vida útil alrededor de 5 años. Asumiendo la hipótesis de amortización lineal da lugar a una amortización de 300€. Finalmente, el único coste de material que se ha tenido es el de sustitución del cable roto del robot. Esto ha supuesto un coste de 23,47€, a partir de la información del trabajo original. [37] El total de los costes asciende a 6.814,12€.

Partida	Concepto	Coste
Personal	Coste de trabajo del estudiante	5.820€
	Coste del trabajo del profesor	670,65€
	Total partida personal	6.490,65€
Amortización	Amortización anual del ordenador portátil	300,00€
Material	Coste de sustitución del cable	23,47€
Coste total	Coste total del proyecto	6.814,12€

Cuadro 10.1: Presupuesto del proyecto

10.2. Estructura de Descomposición del Proyecto (EDP)

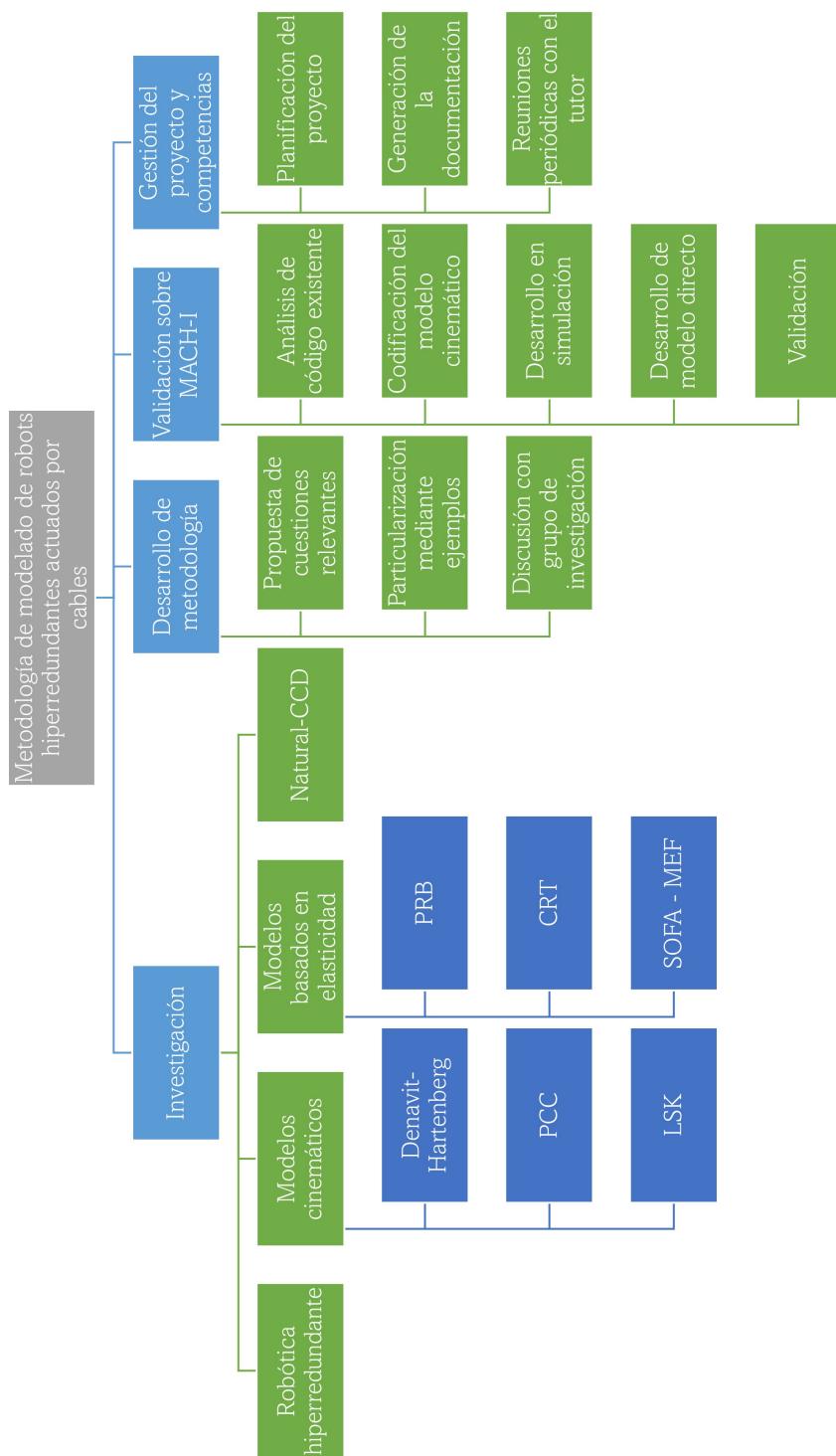


Figura 10.1: Estructura de Descomposición de Trabajo de este proyecto

10.3. Planificación temporal del trabajo

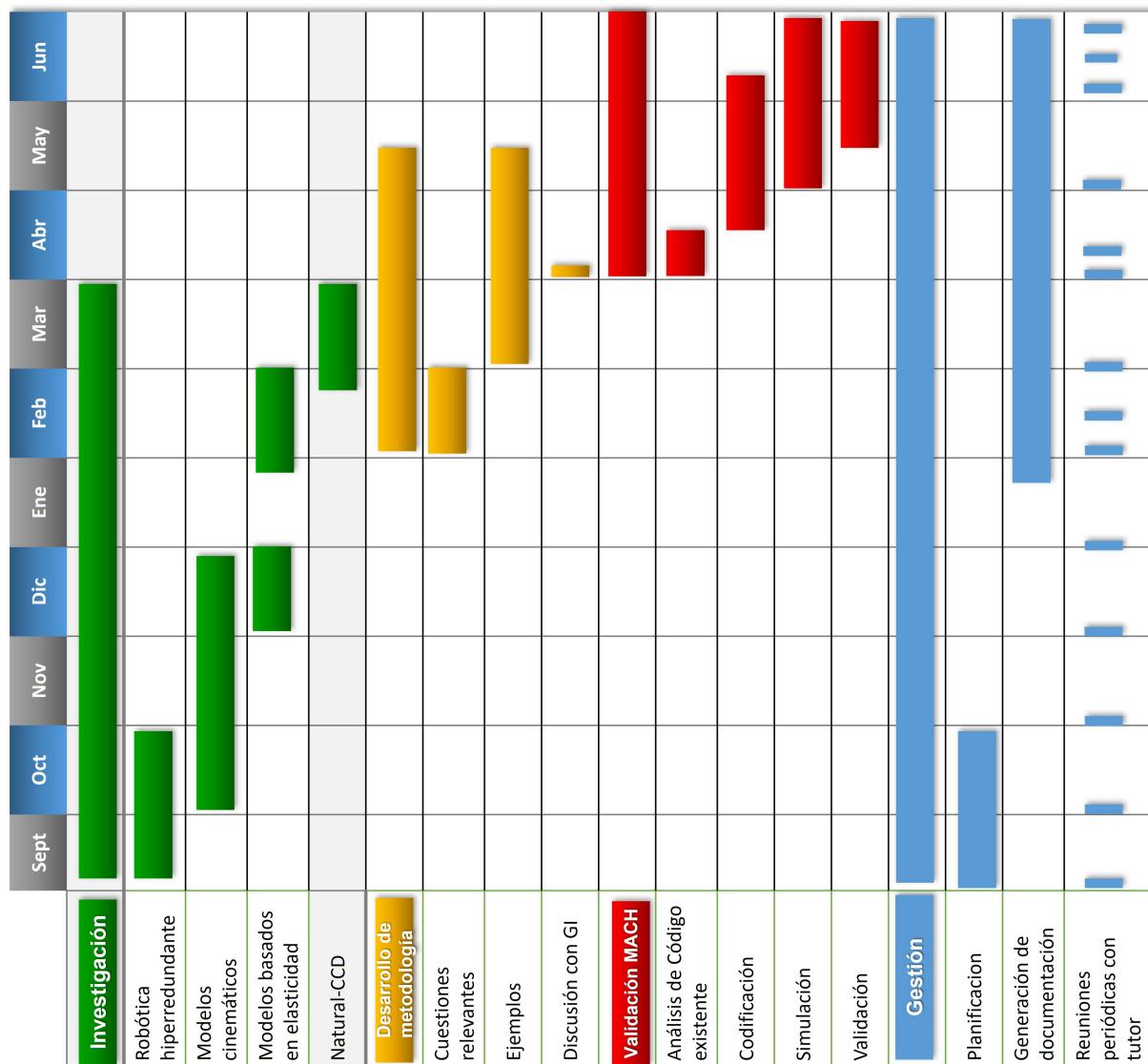


Figura 10.2: Diagrama de Gantt de este proyecto

BIBLIOGRAFÍA

- [1] David Márquez Alcolea. «CONTROL Y TELEOPERACIÓN DE MACH I , UN ROBOT MANIPULADOR HÍPER-REDUNDANTE PARA TAREAS». En: (2021).
- [2] Holm Altenbach, Victor A. Eremeyev y Nikita F. Morozov. «On the Influence of Residual Surface Stresses on the Properties of Structures at the Nanoscale». En: January (2013), págs. 21-32. DOI: 10.1007/978-3-642-35783-1_2.
- [3] Antonio Barrientos Cruz, Luis Felipe Peñín, Carlos Balaguer y Rafael Aracil. *Fundamentos de Robótica*. McGraw-Hill, 2007, pág. 283. ISBN: 9788448156367.
- [4] Jorge Barrientos-Diez, Xin Dong, Dragos Axinte y James Kell. «Real-Time Kinematics of Continuum Robots: Modelling and Validation». En: *Robotics and Computer-Integrated Manufacturing* 67.June (2021). ISSN: 07365845. DOI: 10.1016/j.rcim.2020.102019.
- [5] Joel W. Burdick. «A Modal Approach to Hyper-Redundant Manipulator Kinematics». En: *IEEE Transactions on Robotics and Automation* 10.3 (1994), págs. 343-354. ISSN: 1042296X. DOI: 10.1109/70.294209.
- [6] Pramod Chembrammel y Thenkurussi Kesavadas. «A new implementation for online calculation of manipulator Jacobian». En: *PLoS ONE* 14.2 (2019), págs. 1-16. ISSN: 19326203. DOI: 10.1371/journal.pone.0212018. URL: <http://dx.doi.org/10.1371/journal.pone.0212018>.
- [7] Guimin Chen, Botao Xiong y Xinbo Huang. «Finding the optimal characteristic parameters for 3R pseudo-rigid-body model using an improved particle swarm optimizer». En: *Precision Engineering* 35.3 (jul. de 2011), págs. 505-511. ISSN: 01416359. DOI: 10.1016/j.precisioneng.2011.02.006. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0141635911000353>.
- [8] Gregory S. Chirikjian. «Theory and Applications of Hyper-Redundant Robotic Manipulators». Tesis doct. California Institute of Technology, 1992, pág. 199.
- [9] E. Coevoet y col. «Software toolkit for modeling, simulation, and control of soft robots». En: *Advanced Robotics* 31.22 (2017), págs. 1208-1224. ISSN: 15685535. DOI: 10.1080/01691864.2017.1395362.
- [10] Mohammad Dehghani y S. Ali A. Moosavian. «Modeling and control of a planar continuum robot». En: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM* July (2011), págs. 966-971. DOI: 10.1109/AIM.2011.6027137.
- [11] Christian Duriez. «Control of elastic soft robots based on real-time finite element method». En: *Proceedings - IEEE International Conference on Robotics and Automation* (2013), págs. 3982-3987. ISSN: 10504729. DOI: 10.1109/ICRA.2013.6631138.
- [12] I.D. Erhunmwun y U.B. Ikponmwosa. «Review on finite element method». En: *Journal of Applied Sciences and Environmental Management* 21.5 (2017), pág. 999. DOI: 10.4314/jasem.v21i5.30.
- [13] Jorge Francisco García Samartín. «Modelado cinemático de un robot planar 3RRR mediante técnicas de inteligencia artificial.» En: (2020). URL: <http://oa.upm.es/view/institution/Industriales/>.
- [14] M. W. Hannan. «Theory and Experimentation with an 'Elephant's trunk' robotic manipulator». En: *Oxford University Press* (2002), pág. 649.

- [15] Michael W. Hannan y Ian D. Walker. «Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots». En: *Journal of Robotic Systems* 20.2 (2003), págs. 45-63. ISSN: 07412223. DOI: 10.1002/rob.10070.
- [16] Shaoping Huang, Deshan Meng, Xueqian Wang, Bin Liang y Weining Lu. «A 3D Static Modeling Method and Experimental Verification of Continuum Robots Based on Pseudo-Rigid Body Theory». En: *IEEE International Conference on Intelligent Robots and Systems* (2019), págs. 4672-4677. ISSN: 21530866. DOI: 10.1109/IROS40897.2019.8968526.
- [17] Bryan A. Jones, Ricky L. Gray y Krishna Turlapati. «Three dimensional statics for continuum robotics». En: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009* (2009), págs. 2659-2664. DOI: 10.1109/IROS.2009.5354199.
- [18] Bryan A. Jones, William McMahan y Ian D. Walker. «Practical kinematics for real-time implementation of continuum robots». En: *Proceedings - IEEE International Conference on Robotics and Automation 2006.6* (2006), págs. 1840-1847. ISSN: 10504729. DOI: 10.1109/ROBOT.2006.1641974.
- [19] Bryan A. Jones y Ian D. Walker. «Kinematics for multisection continuum robots». En: *IEEE Transactions on Robotics* 22.1 (2006), págs. 43-55. ISSN: 15523098. DOI: 10.1109/TRO.2005.861458.
- [20] Sehun Kim, Wenjun Xu y Hongliang Ren. «Inverse kinematics with a geometrical approximation for multi-segment flexible curvilinear robots». En: *Robotics* 8.2 (2019). ISSN: 22186581. DOI: 10.3390/ROBOTICS8020048.
- [21] Carmel Majidi. «Soft Robotics: A Perspective - Current Trends and Prospects for the Future». En: *Soft Robotics* 1.1 (2014), págs. 5-11. ISSN: 21695180. DOI: 10.1089/soro.2013.0001.
- [22] Andrés Martín, Antonio Barrientos y Jaime Del Cerro. «The natural-CCD algorithm, a novel method to solve the inverse kinematics of hyper-redundant and soft robots». En: *Soft Robotics* 5.3 (2018), págs. 242-257. ISSN: 21695180. DOI: 10.1089/soro.2017.0009.
- [23] Andrés Martín Barrio. «Design, modelling, control and teleoperation of hyper-redundant robots. 2020». Tesis doct. Universidad Politécnica de Madrid, 2020.
- [24] A. Martin-Barrio, S. Terrile, M. Diaz-Carrasco, J. del Cerro y A. Barrientos. «Modelling the Soft Robot Kyma Based on Real-Time Finite Element Method». En: *Computer Graphics Forum* 39.6 (2020), págs. 289-302. ISSN: 14678659. DOI: 10.1111/cgf.14026.
- [25] Andrés Martín-Barrio, Jaime del Cerro, Antonio Barrientos y Helmut Hauser. «Emerging behaviours from cyclical, incremental and uniform movements of hyper-redundant and growing robots». En: *Mechanism and Machine Theory* 158 (2021). ISSN: 0094114X. DOI: 10.1016/j.mechmachtheory.2020.104198.
- [26] Andrés Martín-Barrio, Juan Jesús Roldán-Gómez, Iván Rodríguez, Jaime Del Cerro y Antonio Barrientos. «Design of a hyper-redundant robot and teleoperation using mixed reality for inspection tasks». En: *Sensors (Switzerland)* 20.8 (2020). ISSN: 14248220. DOI: 10.3390/s20082181.
- [27] Mathworks. *Deep Learning Toolbox*. 2022. URL: <https://es.mathworks.com/help/deeplearning/index.html>.
- [28] J. P. Merlet. *Parallel robots*. Vol. 128. 2006, págs. 1-413. ISBN: 9781402041327. DOI: 10.25103/jestr.146.02.
- [29] Elena Muñoz Sánchez. «Construcción y control de un robot continuo de cables con discos pivotantes». Tesis doct. Madrid: Universidad Politécnica de Madrid, 2022.
- [30] Srinivas Neppalli, Matthew A. Csencsits, Bryan A. Jones y Ian D. Walker. «Closed-form inverse kinematics for continuum manipulators». En: *Advanced Robotics* 23.15 (2009), págs. 2077-2091. ISSN: 01691864. DOI: 10.1163/016918609X12529299964101.

- [31] OC Robotics. *Robot articulado Series II, X125*. URL: <https://www.aeroexpo.online/es/prod/oc-robotics/product-172099-26936.html>.
- [32] David E. Orin y William W. Schrader. «Efficient Computation of the Jacobian for Robot Manipulators». En: *The International Journal of Robotics Research* 3.4 (1984), págs. 66-75. ISSN: 17413176. DOI: 10.1177/027836498400300404.
- [33] Diego Paez-Grandos, Oscar Eduardo Gualdrón, Jairo Lenin y Ramón Valencia. «Aprendizaje de la cinemática en robots redundantes utilizando mapas de b茅zier Learning kinematics redundant robots using maps b茅zier». En: *Colombian Journal of Technology* 14.1 (2015), págs. 23-32. URL: <http://revistas.unbosque.edu.co/index.php/RevTec/article/view/1844>.
- [34] Priyanka Rao, Quentin Peyron, Sven Lilge y Jessica Burgner-Kahrs. «How to Model Tendon-Driven Continuum Robots and Benchmark Modelling Performance». En: *Frontiers in Robotics and AI* 7.February (2021), págs. 1-20. ISSN: 22969144. DOI: 10.3389/frobt.2020.630245.
- [35] Federico Renda y Cecilia Laschi. «A general mechanical model for tendon-driven continuum manipulators». En: *Proceedings - IEEE International Conference on Robotics and Automation* May (2012), págs. 3813-3818. ISSN: 10504729. DOI: 10.1109/ICRA.2012.6224703.
- [36] OC Robotics. *Series II, X125 System*. 2015. URL: <https://www.ocrobotics.com/technology-/series-ii-x125-system/> (visitado 23-10-2021).
- [37] Ivan Rodriguez. «Mach I. Un manipulador actuado por cables hiperredundante». En: (2019).
- [38] Nabil Simaan, Russell Taylor y Paul Flint. «A dexterous system for laryngeal surgery». En: *Proceedings - IEEE International Conference on Robotics and Automation* 2004.1 (2004), págs. 351-357. ISSN: 10504729. DOI: 10.1109/robot.2004.1307175.
- [39] Hai Jun Su. «A pseudorigid-body 3r model for determining large deflection of cantilever beams subject to tip loads». En: *Journal of Mechanisms and Robotics* 1.2 (2009), págs. 1-9. ISSN: 19424302. DOI: 10.1115/1.3046148.
- [40] Silvia Terrile. «Soft Robotics : Applications, Design and Control». Tesis doct. Universidad Politecnica de Madrid, 2021.
- [41] Li-Chun Tommy Wang y Chih Cheng Chen. «A Combined Optimization Method for Solving the Inverse Kinematics Problem of Mechanical Manipulators». En: *IEEE Transactions on Robotics* 7.4 (1991).
- [42] Robert J. Webster y Bryan A. Jones. «Design and kinematic modeling of constant curvature continuum robots: A review». En: *International Journal of Robotics Research* 29.13 (2010), págs. 1661-1683. ISSN: 02783649. DOI: 10.1177/0278364910368147.
- [43] Robert J. Webster, John P. Swensen, Joseph M. Romano y Noah J. Cowan. «Closed-Form Differential Kinematics for Concentric-Tube Continuum Robots with Application to Visual Servoing». En: *Springer Tracts in Advanced Robotics* 54 (2009), págs. 485-494. ISSN: 16107438. DOI: 10.1007/978-3-642-00196-3_56.
- [44] Han Yuan, Lili Zhou y Wenfu Xu. «A comprehensive static model of cable-driven multi-section continuum robots considering friction effect». En: *Mechanism and Machine Theory* 135 (2019), págs. 130-149. ISSN: 0094114X. DOI: 10.1016/j.mechmachtheory.2019.02.005. URL: <https://doi.org/10.1016/j.mechmachtheory.2019.02.005>.

ÍNDICE DE FIGURAS

2.1. Gráfico resumen de clasificación de robots hiperredundantes. [23]	10
2.2. Ejemplos de robots discretos	11
2.3. Ejemplos de robots continuos no blandos	12
2.4. Ejemplos de robots continuos blandos	13
3.1. Esquema sobre las definiciones geométricas en RHAC.	16
3.2. Resumen de las distintas etapas para el modelo cinemático directo. Elaboración propia	17
3.3. Resumen de las distintas etapas para el modelado cinemático inverso. Elaboración propia	19
4.1. Aplicación del marco teórico directo a la hipótesis discreta. Elaboración propia	21
4.2. Aplicación del marco teórico inverso a la hipótesis discreta. Elaboración propia	22
4.3. Aplicación del método de Denavit-Hartenberg a un caso de ejemplo	24
4.4. Aplicación del marco teórico directo a la hipótesis PCC. Elaboración propia	25
4.5. Esquema auxiliar para el modelo PCC. Elaboración propia	26
4.6. Figura auxiliar: plano que contiene al arco. Elaboración propia	28
4.7. Aplicación del Algoritmo de D-H a un robot, aplicando el modelo PCC. Elaboración propia	29
4.8. Figuras auxiliares para las relaciones geométricas de los cables	32
4.9. Aplicación del marco teórico inverso a la hipótesis PCC. Elaboración propia	35
4.10. Longitudes de los cables para robots multisección. Elaboración propia	37
4.11. Aplicación del marco teórico inverso al modelo LSK. Elaboración propia	41
5.1. Aplicación del marco teórico directo al modelo PRB. Elaboración propia	46
5.2. Diagrama conceptual para el modelo PRB. Elaboración propia	47
5.3. Sección teórica de una viga (o núcleo del robot) para explicar la CRT. Elaboración propia	50
6.1. Definición de las articulaciones virtuales para cada sección	56
6.2. Ejemplo de definición de referencias para el robot discreto	57
6.3. Inverse Kinematics using CCD	58
6.4. Definición de los vectores necesarios para el algoritmo CCD	59
6.5. Resumen gráfico del flujo de datos del Natural-CCD. Elaboración propia a partir de [22]	62
7.1. Resumen de la propuesta para el modelado de RHACs. Elaboración propia	66
7.2. Diferencias entre los planteamientos de las longitudes de cables para RHAC discretos y RHAC continuos. Elaboración propia	67
7.3. Detalles del MACH-I que apoyan su clasificación como robot discreto	71
7.4. Aplicación del método Denavit-Hartenberg al MACH-I	72
7.5. Descomposición del giro en los parámetros q_1 y q_2 . Elaboración propia	74
7.6. Razonamiento geométrico para las longitudes de los cables. Elaboración propia	75
7.7. Ruan, un robot blando	77
7.8. Modelo cinemático para el robot Ruan	78

7.9. Pilory, un RHAC continuo	79
7.10. Ecuaciones con el modelo cinemático completo para el robot Pilory	81
8.1. Interfaz gráfica para el manejo del MACH-I	84
8.2. Definición de α respecto al sistema de referencia de la sección. [1]	84
8.3. Interfaz integrada para usar Natural-CCD con MACH-I. [1]	85
8.4. Arquitectura de código generada para la validación del modelo cinemático de MACH-I. Elaboración propia	86
8.5. Ejemplo de ejecución del simulador del MACH-I para el punto [400 100 400] Elaboración propia	89
8.6. Relación entre parámetros de ambos modelos cinemáticos. Elaboración propia . .	90
8.7. Diagrama de caja y bigotes del error cometido en porcentaje para la muestra analizada. Elaboración propia	91
8.8. Arquitectura para las redes neuronales del modelo directo.	93
8.9. Distribución de los puntos para la muestra aleatoria de la red neuronal.	94
8.10. Entrenamiento para la red neuronal empleando el subconjunto de 50000 puntos .	95
8.11. Variación del error en función del tamaño del conjunto de entrenamiento	95
10.1. Estructura de Descomposición de Trabajo de este proyecto	100
10.2. Diagrama de Gantt de este proyecto	101

GLOSARIO

- **GdL:** Grados de libertad
- **RHAC:** Robot hiperredundante actuado por cable. Es un robot con un número de grados de libertad mayor que el doble de los necesarios para determinar la posición y orientación de un sólido rígido (en el espacio son 6 GdL), y que sus actuadores son cables o tendones.
- **Denavit-Hartenberg:** método basado en cálculo matricial para obtener modelos cinemáticos de series de articulaciones y barras.
- **PCC:** Piecewise Constant Curvature. Modelo para robots hiperredundantes que asume que la forma del núcleo es de un arco de circunferencia con curvatura constante.
- **LSK:** Linearized Segments Kinematics. Modificación del modelo PCC que linealiza las ecuaciones para reducir el coste computacional del cálculo de funciones trigonométricas.
- **PRB:** Pseudo-Rigid Model. Modelo basado en parámetros elásticos del material del núcleo que divide cada sección en cuatro segmentos rígidos unidos por tres articulaciones.
- **CRT:** Cosserat Rod Theory, teoría elástica que aplicada al núcleo de un RHAC permite obtener un modelo dependiente de las características de su material.
- **SOFA:** software de simulación basado en el método de elementos finitos que se usa en la bibliografía para modelar robots blandos.
- **Natural-CCD:** algoritmo de optimización para funciones en coordenadas articulares que modifica al CCD. Es un algoritmo de alta eficiencia computacional.



POLITÉCNICA

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES
UNIVERSIDAD POLITÉCNICA DE MADRID**

José Gutiérrez Abascal, 2. 28006 Madrid

Tel.: 91 336 3060

info.industriales@upm.es

www.industriales.upm.es