

Robótica

Apuntes para la asignatura

Módulo optativo recomendado para 3^o de Ingeniería Informática

Módulo optativo para Ingeniería Electrónica

Juan Domingo

3 de febrero de 2001

ADVERTENCIA IMPORTANTE

Según se indica en la introducción que sigue inmediatamente al índice, estas notas no son un libro de Robótica. Fueron escritas en principio sólo para facilitar el trabajo a mis alumnos, todos ellos de las titulaciones de Ingeniería Informática e Ingeniería Electrónica de la *Universitat de Valencia*, y sólo a ellos iban destinadas cuando se incluyeron en la página web de la asignatura. No obstante, atrajeron la atención de los responsables de la excelente web <http://www.ingenieroseninformatica.org> que es, más que una simple página, un foro sobre variados aspectos científicos, tecnológicos y profesionales de la ingeniería informática. Con la intención de contribuir en lo posible a ella he cedido gustoso estas notas, con la única condición de que la presente advertencia figure siempre como cabecera del texto, y sea difundida junto con el resto del mismo, o incluso con partes separadas de él, si éste se fragmentase. En cualquier caso debo manifestar que

- Gran parte del material, y en particular muchas de las figuras, han sido extraídas de los libros que se citan en la bibliografía (aun cuando ninguna ha sido directamente reproducida por medios electrónicos). Por tanto, la bibliografía debiera ser también difundida con el texto o cualesquiera partes de él, de modo que el reconocimiento a los autores originales se preserve.
- Ocasionalmente se citan en el texto marcas comerciales o empresas. El autor no tiene relación contractual ni de ningún otro tipo con ninguna de ellas, y no promueve ni tampoco desaconseja el uso de sus productos.
- El texto puede contener errores, tanto de tipografía como conceptuales. El autor no se hace responsable de las consecuencias que el uso de la información pueda provocar.
- Las posibles opiniones vertidas en el texto son propias del autor, y no reflejan de modo implícito ni explícito ninguna postura oficial de su institución (la *Universitat de Valencia*).
- Del mismo modo, las posibles responsabilidades legales de su difusión si las hubiere corresponderán sólo al autor y al responsable legal de la página web en que se inserten. La inserción de todo o parte del texto en una página web sin el acompañamiento de esta advertencia exime al autor de toda responsabilidad.

Burjasot (Valencia) a 3 de Febrero de 2001

Juan Domingo Esteve (Juan.Domingo@uv.es)

Índice general

Prefacio	P/1
1. Introducción	1/1
1.1. Historia. Orígenes y concepto de robot	1/1
1.2. Tipos de robots	1/3
1.3. Componentes mecánicos de un robot	1/5
1.3.1. Definiciones, componentes y sus tipos	1/5
1.3.2. Accesibilidad	1/5
2. Mecánica de robots	2/1
2.1. Conceptos básicos de geometría espacial	2/2
2.1.1. Sistemas de coordenadas	2/2
2.1.2. Traslaciones y rotaciones	2/2
2.1.3. Descripción de la orientación	2/6
2.1.4. Transformaciones entre sistemas de coordenadas	2/7
2.1.5. El sistema de la mano	2/9
2.1.6. Inversa de una transformación homogénea	2/9
2.2. Cinemática directa del manipulador	2/10
2.2.1. El formalismo de Denavit-Hartenberg	2/10
2.2.2. Estudios de caso	2/15
2.3. Cinemática inversa del manipulador	2/24
2.3.1. Aproximación directa	2/25
2.3.2. Aproximación geométrica	2/25
2.3.3. Manipulación de matrices simbólicas	2/28
2.3.4. Transformaciones de velocidad: el Jacobiano, singularidades y manip- ulabilidad	2/30
2.4. Dinámica del manipulador	2/33
2.5. Generación de trayectorias	2/41
2.5.1. Trayectorias en el espacio de articulaciones	2/41
2.5.2. Trayectorias en el espacio cartesiano	2/47
3. Sensorización	3/1
3.1. Necesidad e importancia. Tipos	3/1
3.2. Sensores internos	3/3
3.2.1. Sensores de posición	3/3
3.2.2. Sensores de velocidad	3/8
3.2.3. Acelerómetros	3/9
3.3. Sensores externos	3/10
3.3.1. Sensores de proximidad	3/11
3.3.2. Sensores de tacto	3/15
3.3.3. Sensores de fuerza	3/16
3.3.4. Sensores de visión	3/18

4. Tecnología de actuadores robóticos	4/ 1
4.1. Actuadores hidráulicos	4/ 1
4.2. Actuadores neumáticos	4/ 2
4.3. Actuadores eléctricos	4/ 3
4.3.1. Motores de corriente continua (CC)	4/ 4
4.3.2. Motores paso a paso	4/ 7
4.4. Servo-amplificadores	4/ 12
4.5. Transmisiones mecánicas y dispositivos de conversión	4/ 13
4.6. Precisión, repetibilidad y resolución	4/ 18
5. Introducción al control de robots	5/ 1
5.1. Técnicas de control clásico sobre motores CC	5/ 1
5.2. Control de una articulación	5/ 7
5.3. Nota al control adaptativo	5/ 9
6. Programación de robots	6/ 1
6.1. Introducción	6/ 1
6.2. Requerimientos de los lenguajes de programación de robots	6/ 2
6.3. Sistemas operativos	6/ 3
6.4. Clasificación de los lenguajes de prog. de robots	6/ 4
6.5. Niveles de programación. Cuadro resumen	6/ 5
6.6. Lenguajes orientados al robot	6/ 6
6.6.1. Movimientos del robot	6/ 6
6.6.2. Evolución y características	6/ 7
6.6.3. Estudios de caso	6/ 8
6.7. Lenguajes orientados a la tarea	6/ 14
6.7.1. Características	6/ 14
6.7.2. Estudios de caso	6/ 17
6.8. Planificación global de tareas. Nota sobre sistemas CIM	6/ 20
7. Robots móviles	7/ 1
7.1. Introducción	7/ 1
7.2. Cinemática de robots móviles	7/ 1
7.3. Navegación	7/ 12
7.3.1. Mapas del entorno	7/ 13
7.3.2. Autolocalización	7/ 15
7.3.3. Planificación y seguimiento de caminos	7/ 16
7.4. Tecnologías de construcción y conexión senso-motora	7/ 19
8. Inteligencia en robots	8/ 1
8.1. Introducción	8/ 1
8.2. La noción de inteligencia y su aplicación en Robótica	8/ 2
8.3. La relación Inteligencia Artificial-Robótica	8/ 3
8.3.1. Robótica clásica	8/ 3
8.3.2. Robótica comportamental	8/ 5
8.4. Estudios de caso	8/ 6
8.4.1. HANDEY: un sistema clásico de ensamblado	8/ 7
8.4.2. Ejemplo comportamental: Herbert	8/ 11

Apéndices

	ApA/ 1
A. Algunas identidades trigonométricas útiles	ApA/ 1
B. Bibliografía comentada y tabla por temas	ApB/ 1

Índice de figuras

1.1. Clasificación de los robots por la geometría	1/ 4
1.2. Tipos de articulaciones y su representación	1/ 6
2.1. Diversos sistema de referencia	2/ 2
2.2. Rotación alrededor de un eje	2/ 3
2.3. Rotación alrededor de un eje arbitrario	2/ 5
2.4. Sistemas de ángulos de Euler e $y/p/r$	2/ 6
2.5. Transformaciones entre sistemas	2/ 8
2.6. Esquema del sistema de la pinza (punto terminal)	2/ 9
2.7. Cadena cinemática abierta y su numeración	2/ 11
2.8. Parámetros DH de un enlace genérico	2/ 12
2.9. Sistemas de coordenadas fijos a una articulación	2/ 13
2.10. Esquema del brazo de Stanford	2/ 16
2.11. Estructura del RT100	2/ 19
2.12. Esquema y cotas del robot RT100	2/ 20
2.13. Colocación de los sistemas en el RT100	2/ 20
2.14. Manipulador con los 3 últimos g.d.l. rotacionales	2/ 26
2.15. Aproximación lineal de una función	2/ 31
2.16. Giro de un sistema de referencia centrado en C	2/ 34
2.17. Posicion de los sistemas O, i e $i+1$	2/ 35
2.18. La velocidad del CM conocida la del origen	2/ 38
2.19. Cuadro resumen del uso de las ecuaciones dinámicas	2/ 40
2.20. Elección de la velocidad intermedia	2/ 43
2.21. Interpolación lineal con extremos parabólicos	2/ 45
2.22. Graficas de la variable de articulación, su velocidad y aceleración	2/ 46
2.23. Aproximación lineal-parabólica para dos puntos	2/ 46
2.24. Brazo con 3 g.d.l.	2/ 48
3.1. Esquemas del sensor potenciométrico	3/ 3
3.2. Tensión obtenida y esperada	3/ 3
3.3. Esquema del sincro	3/ 4
3.4. Esquema del Inductosyn TM	3/ 5
3.5. Discos con una o varias muescas	3/ 5
3.6. Disco de un codificador óptico absoluto de 16 sectores, 4 pistas	3/ 6
3.7. Ondas en los dos canales de un encoder incremental	3/ 6
3.8. Estados/salidas para el circuito de incremento de la resolución	3/ 7
3.9. Esquema de la dinamo tacométrica	3/ 8
3.10. Esquema de un acelerómetro inercial	3/ 10
3.11. Acelerómetro de estado sólido	3/ 10
3.12. Sensor de contacto	3/ 11
3.13. Sensor de reflexión luminosa	3/ 11
3.14. Intensidad emitida y reflejada en modo pulsante	3/ 12
3.15. Reflexión total y sensores con fibra óptica	3/ 13
3.16. Dos tipos de sensores laser de distancia	3/ 13

3.17. Esquema del sensor Polaroid TM	3/14
3.18. Sensor táctil de varillas	3/15
3.19. Sensor de elastómeros	3/16
3.20. Sensor de presión neumática	3/16
3.21. Galga extensiométrica	3/17
3.22. Sensor de fuerza-momento	3/18
4.1. Esquema de la servoválvula hidráulica	4/2
4.2. Cilindro y pistón hidráulicos lineales	4/2
4.3. Control de un sistema hidráulico	4/3
4.4. Controlador proporcional neumático	4/4
4.5. Construcción de un motor eléctrico de CC	4/4
4.6. Curvas V/T e I/T para un motor de CC	4/5
4.7. Modelo para un motor de CC	4/5
4.8. Diagrama de bloques para el motor CC	4/6
4.9. Funcionamiento del motor paso a paso VR	4/8
4.10. Esquema de un motor paso a paso de imán permanente	4/10
4.11. Esquema de un motor paso a paso lineal	4/10
4.12. Esquema del motor brushless	4/11
4.13. Señales para control continuo de motores paso a paso	4/11
4.14. Esquema de los servo-amplificadores lineales	4/12
4.15. Tensión en bornes de un motor CC con control PWM	4/13
4.16. Par de engranajes	4/14
4.17. Tornillo fijo	4/15
4.18. Dispositivos de conversión rotacional-lineal	4/16
4.19. Transmisión universal	4/17
4.20. Alinealidad en engranajes	4/18
4.21. Resolución y medidas de repetibilidad	4/19
5.1. Control de un motor CC: figura principal	5/2
5.2. Lugar de las raíces con $K_g = 0$	5/2
5.3. Control de un motor CC: realim. en velocidad	5/3
5.4. Lugar de las raíces con $K_g \neq 0$	5/4
5.5. Diagrama del motor CC no simplificado	5/5
5.6. Diagrama completo	5/5
5.7. Control PID de un motor CC	5/6
5.8. Respuesta a escalón de un motor CC	5/7
5.9. Control de una articulación	5/8
5.10. Oscilación digital provocada por la cuantización	5/8
5.11. Esquema de control adaptativo	5/9
6.1. Cuadro resumen del software de un sistema robotizado	6/5
6.2. Ejemplo de tarea para resolver en AL	6/10
6.3. Ejemplo de tarea para resolver en VAL	6/13
6.4. Acción de un lenguaje a nivel tarea	6/14
6.5. Ejemplos de cilindros generalizados	6/16
6.6. Ejemplo de pieza en AUTOPASS	6/17
6.7. Ejemplo de tarea para resolver en RAPT	6/18

7.1. Esquema de la rueda de un robot móvil	7/ 3
7.2. Sistemas de referencia para un robot móvil	7/ 4
7.3. Coordenadas de posición-orientación de un robot móvil	7/ 4
7.4. Grafo de transformaciones entre sistemas	7/ 5
7.5. Angulos de giro de la rueda	7/ 7
7.6. Ejemplo de robot móvil	7/ 10
7.7. Construcción visual del mapa de ocupación	7/ 14
7.8. Mapa de espacio libre, y su dual	7/ 14
7.9. Odometría afectada por el terreno	7/ 16
7.10. Autocalización por balizas	7/ 16
7.11. Caminos en espacio libre y negociación de esquinas	7/ 18
7.12. Ejemplo de campo potencial	7/ 18
7.13. Esquema del robot Tut-Bot	7/ 19
7.14. Movimiento del Tut-Bot	7/ 20
8.1. Sistemas de símbolos y su relación con el mundo	8/ 4
8.2. Descomposición funcional y vertical	8/ 5
8.3. Descomposición horizontal	8/ 7
8.4. Parámetros del espacio de configuraciones	8/ 10
8.5. Un ejemplo de espacio de configuraciones	8/ 10
8.6. Otro ejemplo de espacio de configuraciones	8/ 11
8.7. Esquema de Herbert	8/ 12
8.8. La mano de Herbert	8/ 12
8.9. Sistema de visión de Herbert	8/ 13

Prefacio

El volumen que el lector tiene ahora en su mano es un compendio pormenorizado de contenidos que se integran en el módulo "Robótica". A partir de la introducción de los nuevos planes de estudios, esta disciplina, presente en los planes antiguos a veces como una asignatura autónoma, o generalmente como parte de una asignatura mayor, ha pasado a ser un módulo integrante de los planes de la mayoría de las ingenierías en Informática, Ingenierías Industriales, y similares de casi todas las universidades de España. Partiendo de las orientaciones generales del MEC, que aparecen como descriptores genéricos que indican contenidos básicos que deben ser explicados, cada licenciatura o ingeniería estructura su plan de estudios, el cual especifica en qué módulos (y en su caso, en que cursos) deben ser impartidos estos conocimientos.

Las notas que siguen han sido redactadas como guía del módulo impartido como optativo en las titulaciones de Ingeniería Informática e Ingeniería Electrónica de la Univ. de Valencia. Como es preceptivo, se siguen las indicaciones generales del MEC explícitas en forma de descriptores, pero adaptándolas a los conocimientos que se juzgan apropiados para un alumno de esta licenciatura en esta universidad, dado que por supuesto el contenido hace referencia a otros módulos y se conecta con ellos. Por tanto, aun cuando este documento ha sido escrito con intención de generalidad, y tratando de introducir debidamente los conocimientos previos, no pretende ser un texto completo de Robótica, y no debiera ser tomado como tal. A este respecto, se recuerda encarecidamente a los alumnos que los apuntes **no** sustituyen a un buen texto, y mucho menos al seguimiento puntual y atento de las explicaciones dadas en el aula. Quieren ser, más bien, una ayuda, especialmente en aquellos temas en que la abundancia de fórmulas dificulta la comprensión, porque el alumno obligado a transcribirlas se fija más en la letra que en el contenido y significado físico de éstas. Se verá, sin embargo, que estos apuntes no son una mera sucesión de fórmulas, y que no se ha escatimado texto en ellos; esto responde, precisamente, al deseo de explicar y comentar debidamente los aspectos físicos o computacionales, de los que las matemáticas son aquí mera expresión.

En Robótica, este último comentario es especialmente importante. En nuestra opinión, la Robótica no es software, ni hardware, ni siquiera una suma de éstos. Una de las ideas más importantes que se intenta que un alumno adquiera en este módulo es que la Robótica intenta establecer una conexión entre las ideas que tenemos sobre el mundo físico y el flujo de información en él, las cuales plasmamos mediante programas, y el mundo físico mismo. Los problemas de la programación cuando se enfrenta a sistemas reales son, en general, bastante diferentes de los que se plantean en un ordenador aislado; esto se manifiesta patente a los alumnos en las prácticas que constituyen el módulo asociado a éste.

Como se verá en el índice, el curso se estructura en ocho temas. En el primero se introduce el concepto de Robótica (el cual será discutido y modificado posteriormente en el tema 8), una reseña histórica y algo de la terminología básica.

El segundo tema explica la relación entre las variables propias del robot (variables de articulación) y las coordenadas cartesianas usadas normalmente por humanos, así como entre sus velocidades, aceleraciones, y fuerzas aplicadas; para ello se introduce la formulación de Denavit-Hartenberg, comunmente aceptada.

Los temas tercero y cuarto hablan respectivamente de sensores y actuadores, desde un punto de vista principalmente tecnológico, pero haciendo hincapié en su papel de recolec-

tores de información, y alteradores del mundo, respectivamente, lo cual permitirá después hablar de Robótica como conexión entre percepción y acción.

El tema quinto explica los modelos usados para los actuadores más comunes hoy día (motores de CC) así como algunas técnicas básicas para su control. La correcta comprensión de este tema requiere conocimientos previos sobre control, al menos de sistemas continuos, de carácter elemental. Esto es cierto en el caso de nuestros alumnos que cursan al menos una asignatura obligatoria sobre el tema anteriormente; pero es necesario reseñar que otras facultades de Informática podrían no encontrarse en el mismo caso, y, por contra, algunas de Ingeniería Industrial encontrarían este tema parco. Se ha buscado el compromiso más adaptado a nuestro caso.

El tema sexto, programación, es uno de los más importantes para un alumno de informática, al que se le supone fluidez en ella, y ha sido planteado con el objetivo de intentar mostrar las diferencias esenciales entre programar un dispositivo físico que interactúa con el mundo real y programar un ordenador; en los ejemplos de lenguajes dados se incide en ello.

El tema séptimo, sobre robots móviles, ha sido introducido para dar cuenta de una tecnología aún en desarrollo, pero que ya comienza a tener algunas bases, a nuestro entender, suficientemente firmes como para incluirla en un temario académico, y que creemos que tendrá un impacto industrial y científico importante en un futuro próximo.

Finalmente, el tema octavo explica la conexión entre Inteligencia Artificial y Robótica, haciendo hincapié en la fructífera relación entre ambas disciplinas, no exenta sin embargo de tensiones, que se estableció en su día, y que hoy se ha revitalizado. Pensamos que una visión genérica de este tipo de cuestiones es apropiada para un estudiante de Ingeniería Informática.

Finalmente, hay que indicar que estas notas son preliminares, y sometidas constantemente a revisión. Pese al trabajo de mecanografía lo más cuidadoso posible, y a una organización estricta y presentación cuidada, que son más mérito de L^AT_EX que del autor, multitud de erratas pueden haber subsistido. Se pide disculpas de antemano por ellas, y se ruega al lector interesado que las haga notar, para su corrección en futuras versiones.

Tema 1: Introducción

1.1. Historia. Orígenes y concepto de robot

El primer punto que debemos considerar en un curso de Robótica debería ser, obviamente, la definición. Se admite que Robótica es la ciencia de los robots, con lo que se traslada el problema a la definición de robot. Sin embargo, y como luego veremos, darla no es tan simple. Antes de hacerlo, veamos mejor algunas ideas informales sobre qué es un robot, y qué concepto se ha ido formando históricamente de ellos.

Una noción usual es que un robot es un dispositivo humanoide, más o menos inteligente, que reemplaza a los humanos en la realización de tareas útiles. Generalmente, la imaginación popular no considera robots a las máquinas que no tienen una forma al menos vagamente humana. Desde el punto de vista literario o cinematográfico, que ha ido creando esta visión popular, el origen se suele poner en 1921, con la obra de teatro "RUR" (*Rossum Universal Robota*), del checo Karel Capek, de donde surge la palabra robot (*robota* en checo significa esclavo, o trabajador obligado). Más tarde esta visión fue reforzada por Fritz Lang en su película "Metrópolis", de 1926, y bastante después, una versión más humanizada es la que presenta Isaac Asimov en su archiconocida serie de relatos, escritos a partir de 1942 (en los que, por cierto, se introduce por primera vez el término Robótica con el sentido de disciplina científica encargada de construir y programar robots). Entre los últimos ejemplos de esta forma ingénuo de ver los robots se encuentran los andróides R2D2 y C3PO de la serie de películas de George Lucas "La Guerra de las Galaxias".

Por otra parte, desde la generalización del uso de la tecnología en procesos de producción con la Revolución Industrial (y aun antes) se intentó la construcción de dispositivos automáticos que ayudasen o sustituyesen al hombre, o simplemente fuesen en sí mismos objetos curiosos. Entre ellos destacaron:

-Los Jaquemarts, muñecos de dos o más posiciones que golpean campanas accionados por mecanismos de relojería. Entre los más famosos están el "papamoscas", de la Catedral de Burgos, o los de la Catedral de San Marcos, de Venecia. Su importancia radica en el hecho de transferir la tecnología usada en relojería al movimiento preciso de piezas pesadas.

-En la misma línea, pero mucho más sofisticados, son los autómatas de engranajes y mecanismos de relojería que surgen sobre todo en Francia y Alemania entre los siglos XVII y XVIII. El más famoso fué el llamado "concertista de tímpano", construido por Kintzing y Roentgen y regalado a María Antonieta en 1784. Funcionaba mediante un mecanismo de cuerda que activaba una compleja sucesión de palancas y árboles de levas, y tiene importancia por ser uno de los primeros dispositivos "programables" (tocaba cualquier melodía cambiando el tambor metálico con dientes que la codificaba).

-En 1801, C. Spencer inventa una máquina, antecesora de las modernas máquinas-herramienta, que producía tornillos, tuercas y arandelas. El tipo de pieza, tamaño y paso de la misma era variable, mediante la sustitución de unas guías intercambiables, que actuaban a modo de "programa".

-En 1938-39, el americano Pollard inventa el primer brazo mecánico articulado para ser usado en aplicaciones de pintura al spray. Ciertamente, uno de los campos de aplicación más extendidos de los robots son las cadenas de montaje en general, y las de automóviles en particular. Recordemos que hacía ya tiempo, desde 1906, que Henry Ford había instalado la primera cadena de montaje para su mítico modelo T, que significó una revolución en el

campo de la producción industrial en general.

De todos modos, ninguno de los ingenios citados hasta ahora puede realmente considerarse un robot. Algunos se pueden programar, pero todos carecen de la capacidad de ejecutar tareas diversas, puesto que están por su propia estructura mecánica, concebidos para una sólo aplicación. Inmediatamente comenzaron a aparecer otros dispositivos no sujetos a esta limitación:

-En 1947, Goetz construye el primer manipulador eléctrico servocontrolado, es decir, capaz de conocer su propio estado (posición), y modificarlo para llevarlo al estado deseado por el programador.

-En 1952 aparece la primera máquina de control numérico, o sea, que se programa no de modo mecánico o eléctrico, sino mediante intrucciones simbólicas (en cierto modo, un lenguaje) codificadas apropiadamente. En principio, se trataba sólo de tornos programables, pero su importancia está en que introducen el concepto de "software" de un robot.

-En 1954 Devol desarrolla el que puede considerarse el primer robot: un manipulador con memoria legible y escribible (o sea, programable) que almacena trayectorias punto a punto. Devol fundó entonces Unimation, la que sería la mayor compañía productora de robots durante los años siguientes.

-Al mismo tiempo que los robots industriales se hacían más complejos, durante las décadas de los 60 y 70, la Robótica comienza a introducirse como disciplina académica y a formar parte de la investigación de algunos centros universitarios específicos, que aún hoy mantienen su preeminencia; cabe citar al Massachusetts Institute of Technology (MIT), las Universidades de Carnegie Mellon (Illinois) y Stanford (California), el Technological Institute of California (CALTECH) y la Universidad de Edimburgo (Escocia). De hecho, en 1969 los investigadores de Stanford contruyen el primer prototipo de Shakey, un robot cilíndrico móvil dotado de sensores de ultrasonidos y una visión rudimentaria, y en 1970 comienza a funcionar el brazo de Stanford, uno de los robots académicos más extendidos; tiene 6 grados de libertad, es eléctrico y servocontrolado.

-La investigación espacial en la década de los 70 también impulsó la tecnología robótica, especialmente los proyectos VIKING-I y II, para un descenso no tripulado en Marte. El brazo de la sonda del VIKING-I requirió la invención de la mano con *remote compliance center* (centro de giro remoto, RCC), especialmente usada luego en tareas de ensamblado.

-En 1978 Unimation lanza el PUMA, probablemente el robot comercial más difundido, y en torno a la misma fecha se dan en las universidades algunos avances en visión que todavía están por incorporarse a los robots industriales.

-Durante los años 80 y 90, la investigación continúa, aunque no al ritmo que las compañías desearían para una mayor aplicabilidad industrial de los robots, y la diversificación es enorme. Existen multitud de modelos de brazo articulado, varios tipos de robots móviles, e incluso robots que andan (como el ODEX, desarrollado por la firma Odetics Inc. para manejo de material radiactivo).

De todo lo dicho se deduce que debe haber un nivel de complejidad a partir del cual un dispositivo puede considerarse robot. Esto será aclarado en la definición, pero en cualquier caso, los tres niveles de complejidad de un dispositivo de manipulación puede considerarse que son:

- Teleoperación: Es la manipulación a distancia por un operador humano. En este nivel están, p. ej., los brazos desarrollados a partir del final de la segunda guerra mundial para la manipulación de material radiactivo.

- Telepresencia: Una teleoperación a la que además se añaden sensores que informan al operador del estado de la tarea, idealmente, como si éste se encontrase en el lugar de ejecución de la misma. Los dispositivos móviles para desactivación de explosivos dotados de cámaras de TV y sensores de contacto o fuerza son ejemplos de esto.

- Autonomía: El robot hace su trabajo tomando las decisiones oportunas a partir del programa que almacena, y de las señales recibidas de los sensores en cada instante, sin necesidad de intervención del operador durante la ejecución de la tarea. Lo idóneo sería que todo robot funcionase de este modo, e incluso con capacidad para variar su programa, y aprender. Si esto puede o no ser posible en el futuro, es objeto de investigación.

Ahora estamos por fin en condiciones de dar una definición de robot. Según el *Robot Institute of America*, que, significativamente, se transformó después en la *Robot Industries Association (RIA)*, un robot es

"Un manipulador reprogramable multifuncional, diseñado para mover material, partes, herramientas o dispositivos especializados mediante movimientos programados variables para la ejecución de tareas diversas"

Las características importantes que aparecen en esta definición y que distinguen a un robot de otros dispositivos manipuladores, como máquinas-herramienta o similares, son la multifuncionalidad, es decir, el robot debe ser lo suficientemente versátil como para ejecutar tareas diversas, no previstas a priori por su diseñador, y la programabilidad, o sea, la capacidad de cambiar de una tarea a otra sin más que cambiar el programa (la secuencia de instrucciones) que debe ejecutar.

Esta definición es correcta y suficientemente general, en el sentido de que no restringe la tecnología usada para la construcción del robot, ni el método por el que se le programe. Pero peca de excesivamente ingenieril, en cuanto que olvida el aspecto estrictamente científico de los robots: su uso como herramientas para entender los procesos de la percepción y la acción en entornos reales. Se incidirá más sobre esto en el capítulo 8. Por el momento, se puede dar una definición alternativa de Robótica como

"La ciencia que estudia los robots como sistemas que operan en algún entorno real, estableciendo algún tipo de conexión inteligente entre percepción y acción"

Aquí no nos referimos a inteligencia general, en el sentido humano del término, sino sólo a la capacidad de realizar bien la tarea encomendada, aun ante cualesquiera cambios razonables en el entorno.

1.2. Tipos de robots

Existen hoy día multitud de robots, algunos de ellos muy específicos, por lo que es difícil establecer una única clasificación válida. No obstante, hay dos criterios que ayudan:

+Clasificación por la geometría (llamada también por las coordenadas):

- Cilíndricos: cada eje es de revolución total (o casi) y está encajado en el anterior
- Esféricos: hay ejes de rotación que hacen pivotar una pieza sobre otra.
- De paralelogramo: La articulación tiene una doble barra de sujección.
- Mixtos: poseen varios tipos de articulación, como los SCARA.
- Cartesianos: las articulaciones hacen desplazar linealmente una pieza sobre otra.

Pueden ser de tipo cantilever, o pórtico (ver figura 1.1)

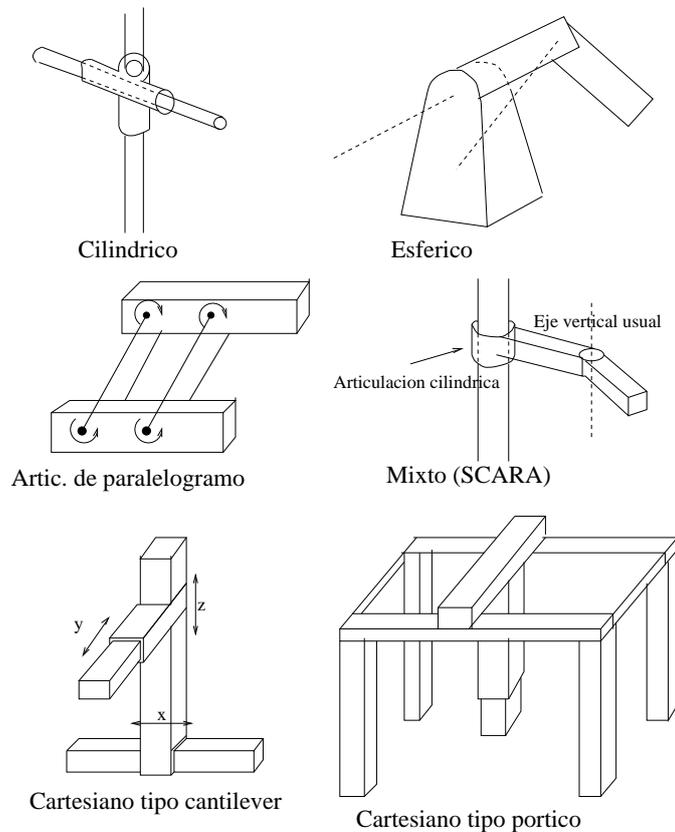


Figura 1.1: Clasificación de los robots por la geometría

+Clasificación por el método de control:

-No servo-controlados: son aquellos en los que cada articulación tiene un número fijo (normalmente, dos) posiciones con topes y sólo se desplazan para fijarse en ellas. Suelen ser neumáticos, bastante rápidos y precisos.

-Servo-controlados: en ellos cada articulación lleva un sensor de posición (lineal o angular) que es leído, y enviado al sistema de control que genera la potencia para el motor. Se pueden así parar en cualquier punto deseado.

-Servo-controlados punto a punto: Para controlarlos sólo se les indican los puntos iniciales y finales de la trayectoria; el ordenador calcula el resto siguiendo ciertos algoritmos que se verán en el tema 2. Normalmente pueden memorizar posiciones.

+Clasificación por la función:

-De producción, usados para la manufactura de bienes, pueden a su vez ser de manipulación, de fabricación, de ensamblado y de test.

-De exploración, usados para obtener datos acerca de terreno desconocido, pueden ser de

exploración terrestre, minera, oceánica, espacial, etc.

-De rehabilitación: usados para ayudar a discapacitados, pueden ser una prolongación de la anatomía, o sustituir completamente la función del órgano perdido.

1.3. Componentes mecánicos de un robot

1.3.1. Definiciones, componentes y sus tipos

Daremos ahora unas definiciones que serán útiles para los temas posteriores.

Un **manipulador** robótico consiste en una secuencia de cuerpos rígidos, llamados enlaces (links) que se conectan unos a otros mediante articulaciones (joints). Todos juntos forman una **cadena cinemática**. Se dice que una cadena cinemática es abierta si, numerando secuencialmente los enlaces desde el primero, cada enlace está conectado mediante articulaciones exclusivamente al enlace anterior, y al siguiente, excepto el primero, que se suele fijar al suelo, y el último, uno de cuyos extremos queda libre.

Cada articulación puede ser **rotacional** o **traslacional**, según el enlace dado gire alrededor de un eje fijo al enlace anterior, o se deslice sobre él en línea recta. Estos dos tipos de articulaciones son las principales, aunque hay más. Véase la figura 1.2, al final del tema.

Se define **grado de libertad** como cada una de las coordenadas independientes necesarias para describir el estado de un sistema mecánico. Normalmente, en cadenas cinemáticas abiertas, cada par enlace-articulación tiene un sólo grado de libertad, bien rotacional o traslacional, pero no necesariamente. Puede haber enlaces de longitud 0 (inexistentes), p. ej., cuando una articulación tiene dos o más grados de libertad que operan sobre ejes que se cortan.

1.3.2. Accesibilidad

Lo que debemos hacer con un manipulador es normalmente situar su punto terminal en el punto del espacio pedido por el usuario, haciendo además que la dirección por la que se aproxima a ese punto sea también una dada. En algunos manipuladores no todas las direcciones de aproximación son posibles.

Usando esta noción se dice que un punto del espacio es **totalmente accesible** si el punto terminal del manipulador puede situarse en él, en todas las orientaciones que su construcción mecánica en principio le permita. Es **parcialmente accesible** si el punto terminal puede alcanzarlos, pero no en todas las orientaciones con las que puede alcanzar otros puntos. De modo obvio, se define el **espacio de accesibilidad total** (en su caso, **parcial**) de un robot como el conjunto de todos los puntos del espacio totalmente (parcialmente) accesibles para ese robot.

La accesibilidad viene limitada por varios factores: de tipo geométrico (el punto está demasiado alejado para alcanzarlo, aun con el manipulador totalmente extendido), de tipo mecánico (unos enlaces del robot chocan con otros, impidiendo así el acceso a ciertas áreas) o de tipo constructivo (las articulaciones tienen límites a su movimiento, angular o lineal).

Muy relacionado con el concepto de accesibilidad está el de **manipulabilidad**, que tiene que ver con la dificultad de controlar el robot para que acceda a ciertos puntos. Su definición formal no puede darse hasta el tema 2, pero hay que señalar que existen puntos llamados singularidades que, aun siendo accesibles, requieren ciertas precauciones para llevar el extremo del manipulador hasta ellos.

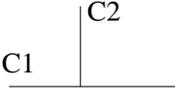
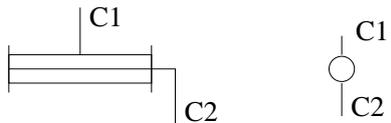
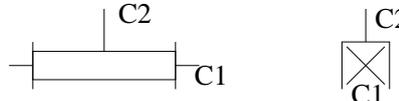
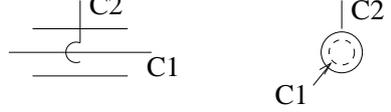
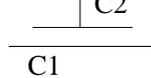
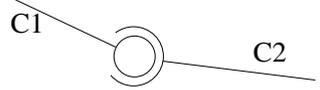
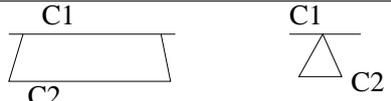
Nombre de la union	Movimien. relativos	Num. de g.l.	Simbolos (vistas lateral y frontal)
Union empotramiento	0 traslac. 0 rotac.	0	 <p>C1: cuerpo 1 C2: cuerpo 2</p>
Union pivote (rotacional)	1 rotac. 0 traslac.	1	
Union deslizante (traslacional)	0 rotac. 1 traslac.	1	
Union deslizante helicoidal	1 rotac. y 1 traslac. conjugadas	1	
Union pivote deslizante	1 rotac. 1 traslac.	2	
Union apoyo plano	1 rotac. 2 traslac.	3	
Union rotula	3 rotac. 0 traslac.	3	
Union lineal deslizante	2 rotac. 2 traslac.	4	
Union lineal anular	3 rotac. 1 traslac.	4	
Union puntual	3 rotac. 2 traslac.	5	
Union libre	3 rotac. 3 traslac.	6	No hay simbolo No hay contacto entre los dos cuerpos

Figura 1.2: Tipos de articulaciones y su representación

Tema 2: Mecánica de robots

En este tema se desarrollarán la formulación y las técnicas apropiadas para caracterizar mecánicamente el comportamiento de un brazo robot. El brazo, como cualquier otro cuerpo, está sometido a las leyes usuales de la mecánica, las cuales, expresadas en alguna formulación apropiada (Newton, Lagrange, etc.) deberán aplicársele para conocer cuál es su movimiento, o sus condiciones de reposo. Hay dos objetivos últimos, resultado de dicha aplicación: conocer la posición del punto terminal (o de cualquier otro punto) de un brazo robot respecto a un sistema de coordenadas externo y fijo (el sistema del mundo), y conocer cuál será el movimiento del brazo cuando los actuadores que lo controlan le apliquen determinadas fuerzas y momentos.

El análisis mecánico de un robot puede hacerse bien atendiendo exclusivamente a sus movimientos, o bien a éstos y también a las fuerzas que actúan sobre él.

Cuando se estudian exclusivamente los movimientos (posición y velocidad de cada articulación o del punto terminal) se dice que hacemos un estudio **cinemático**. Podemos pasar:

- Bien desde las coordenadas propias del robot (ángulos o longitudes de cada articulación) hasta las coordenadas cartesianas de posición y orientación del punto terminal (usualmente, (x, y, z) y tres ángulos. Esto se llama construir la cinemática directa, y existe un método sistemático para hacerlo, basado en la llamada formulación de Denavit-Hartenberg, que va a ser explicada con detalle inmediatamente.
- Bien desde las coordenadas cartesianas referidas a algún sistema externo fijo a las coordenadas propias del robot (ángulos o longitudes). En esto consiste hallar la cinemática inversa. No en todos los manipuladores existe una solución expresable analíticamente para este problema, y en la mayoría de los casos la solución no es única; también se verán a continuación métodos y ejemplos.

Por otra parte, cuando se estudian las fuerzas y momentos que ejerce la carga transportada sobre la última articulación, así como las que ejercen los actuadores, y cada articulación sobre las contiguas, es posible determinar el movimiento, aplicando las leyes de la mecánica en cualquiera de sus formulaciones (Newton, Lagrange, D' Alembert, ...). En esto consiste hacer un estudio dinámico; nosotros lo haremos usando la formulación de Newton.

Para poder expresar de forma apropiada las ecuaciones que caracterizan estos fenómenos necesitamos conocer conceptos básicos de geometría que permitan expresar las transformaciones entre sistemas de coordenadas. De lo dicho antes, es también obvio que necesitaremos conceptos básicos de mecánica (las leyes de Newton). Cada uno de éstos sirve respectivamente para describir el estado del robot únicamente en términos de su movimiento (estudio cinemático), y en términos de las fuerzas y momentos que actúan sobre él (estudio dinámico). Finalmente, y si somos capaces de describir la posición y velocidad del brazo en cada instante, lo seremos igualmente de generar para él una trayectoria que cumpla requerimientos apropiados.

Toda esta secuencia de objetivos estructura este tema en las secciones que siguen.

Esto tiene la ventaja de que, si $\vec{x}' = H \cdot \vec{x}$, entonces $\vec{x} = (H)^{-1} \cdot \vec{x}'$, donde se puede calcular la inversa, que resulta ser

$$(H)^{-1} = \begin{pmatrix} 1 & 0 & 0 & -v_x \\ 0 & 1 & 0 & -v_y \\ 0 & 0 & 1 & -v_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

lo cual es consistente con el hecho de que, respecto a \vec{x}' , \vec{x} está trasladado por un vector $-\vec{v}$.

Respecto a la rotación alrededor de un eje, en el caso bidimensional, véase la figura 2.2, en la cual se gira alrededor de un eje perpendicular al plano de la figura.

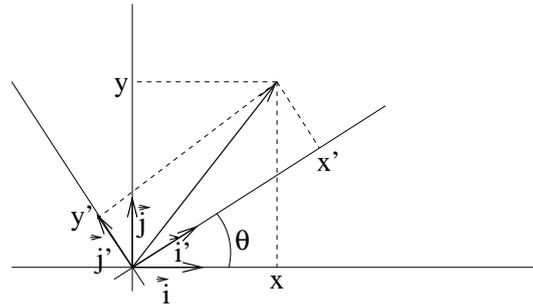


Figura 2.2: Rotación alrededor de un eje

Llamando \vec{i} y \vec{j} a los vectores básicos del sistema original, e \vec{i}' y \vec{j}' a los del sistema girado, se tiene que

$$\vec{x} = x\vec{i} + y\vec{j} = x'\vec{i}' + y'\vec{j}' \quad \text{y también} \quad \begin{aligned} \vec{i}' &= \cos\theta\vec{i} + \sin\theta\vec{j} \\ \vec{j}' &= -\sin\theta\vec{i} + \cos\theta\vec{j} \end{aligned}$$

es decir que $x\vec{i} + y\vec{j} = x'(\cos\theta\vec{i} + \sin\theta\vec{j}) + y'(-\sin\theta\vec{i} + \cos\theta\vec{j})$

Igualando componente a componente, y escribiendo como matriz,

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix}$$

Si generalizamos a tres dimensiones, la coordenada z no varía, y la cuarta coordenada homogénea sigue siendo 1, luego

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix}$$

Para hallar la transformación inversa, basta ver que desde el punto de vista de R' , R está rotado un ángulo $-\theta$, luego

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(-\theta) & -\sin(-\theta) & 0 & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix}$$

que es más simple que invertir la matriz, aunque por supuesto, equivalente.

En general, si hubiéramos rotado alrededor de otro de los ejes básicos, se puede ver que

$$Rot(x, \theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Rot(y, \theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Rot(z, \theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Nótese el cambio de signo en la rotación alrededor del eje y , debido a que, si el eje alrededor del cual rotamos nos apunta, los otros dos forman ángulo de 90° en el caso de x y z , pero de -90° en el caso de y .

Es claro que se pueden aplicar a un punto tantas transformaciones sucesivas (rotaciones y traslaciones) como se quiera. La operación resultante vendría expresada por una matriz que sería producto de las matrices de cada operación, aplicadas en el orden debido, dado que el producto de matrices no es conmutativo. Se pone más a la derecha la primera transformación que se aplique. Ej.: $Y = T_2 R_3 T_1 R_2 R_1 X$ significa que se aplica al punto X la rotación R_1 , seguida de la R_2 , seguida de la traslación T_1 , luego la rotación R_3 y por último la traslación T_2 .

Veamos ahora cuál sería la matriz de rotación respecto a un eje cualquiera. Sea un eje que pasa por el origen definido por un vector unitario $\vec{r} = (r_x, r_y, r_z)$, alrededor del cual giraremos un ángulo θ . La matriz será $R_{\vec{r}, \theta}$. La rotación se descompondrá en otras tres alrededor de los ejes básicos: (ver figura 2.3):

Por este orden:

- Rotar un ángulo α alrededor de x , con lo que $P \rightarrow P'$
- Rotar un ángulo $-\beta$ alrededor de y , con lo que $P' \rightarrow P''$
- Rotar un ángulo θ alrededor de z , que es la rotación pedida.
- Rotar un ángulo β alrededor de y , lo que deshace la segunda rotación.
- Rotar un ángulo $-\alpha$ alrededor de x , lo que deshace la primera rotación.

Entonces, $R_{\vec{r}, \theta} = R_{x, -\alpha} R_{y, \beta} R_{z, \theta} R_{y, -\beta} R_{x, \alpha}$ o sea,

$$R_{\vec{r}, \theta} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha & s\alpha & 0 \\ 0 & -s\alpha & c\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c\beta & 0 & s\beta & 0 \\ 0 & 1 & 0 & 0 \\ -s\beta & 0 & c\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c\beta & 0 & -s\beta & 0 \\ 0 & 1 & 0 & 0 \\ s\beta & 0 & c\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha & -s\alpha & 0 \\ 0 & s\alpha & c\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

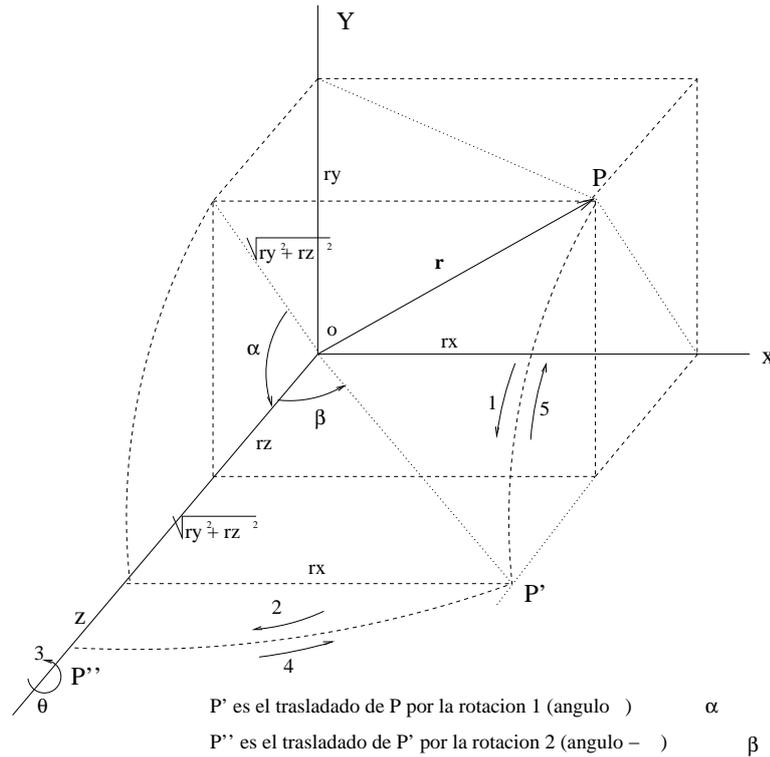


Figura 2.3: Rotación alrededor de un eje arbitrario

Ahora bien, de la figura se ve que

$$\begin{aligned}
 s\alpha &= \frac{r_y}{\sqrt{r_y^2 + r_z^2}} \\
 c\alpha &= \frac{r_z}{\sqrt{r_y^2 + r_z^2}} \\
 s\beta &= \frac{r_x}{|r|} = \frac{r_x}{\sqrt{r_x^2 + r_y^2 + r_z^2}} \\
 c\beta &= \frac{\sqrt{r_y^2 + r_z^2}}{|r|} = \frac{\sqrt{r_y^2 + r_z^2}}{\sqrt{r_x^2 + r_y^2 + r_z^2}}
 \end{aligned}$$

Al multiplicar todo queda:

$$R_{\vec{r},\theta} = \begin{pmatrix} r_x^2 v\theta + c\theta & r_x r_y v\theta - r_z s\theta & r_x r_z v\theta + r_y c\theta & 0 \\ r_x r_y v\theta + r_z s\theta & r_y^2 v\theta + c\theta & r_y r_z v\theta - r_x s\theta & 0 \\ r_x r_z v\theta - r_y s\theta & r_y r_z v\theta + r_x s\theta & r_x^2 v\theta + c\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

siendo $s\theta = \sin \theta$, $c\theta = \cos \theta$ y $v\theta = 1 - \cos \theta$.

Naturalmente, si se hace $r_x = r_y = 0$ se obtiene la rotación alrededor del eje z, y similarmente con los otros ejes.

Debemos hacer hincapié en un detalle importante: cualquier secuencia consecutiva de transformaciones (rotaciones, traslaciones, o la transformación homogénea conjunta) se puede especificar de dos maneras:

- Realizando la rotación que lleva un sistema al otro alrededor de uno de los ejes iniciales.

En este caso, la matriz que describe esta transformación deberá premultiplicarse por la que

describía las transformaciones efectuadas hasta ese momento, obteniendo la transformación total.

- Realizando la rotación que lleva un sistema al otro alrededor de uno de los ejes girados, es decir, los que resultaron de la última transformación. En este caso, la matriz que describe esta transformación deberá postmultiplicarse por la que describía las transformaciones efectuadas hasta ese momento, obteniendo la transformación total.

2.1.3. Descripción de la orientación

En vez de describir la orientación en función de las componentes (r_x, r_y, r_z) de un vector unitario, es apropiado para muchos propósitos describirlas en términos de tres ángulos. Hay dos representaciones usuales: los llamados ángulos de Euler, y el sistema yaw/pitch/roll (desviación/elevación/giro, o, en términos aeronáuticos, deriva/cabeceo/balanceo). Véase la figura 2.4.

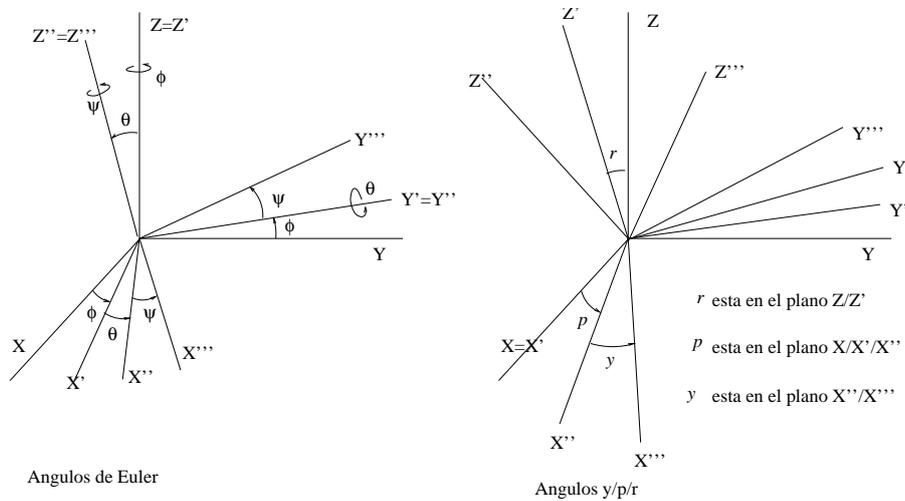


Figura 2.4: Sistemas de ángulos de Euler e y/p/r

Los ángulos de Euler permiten fijar una orientación cualquiera en el espacio. Es fácil ver que, dados dos sistemas cualesquiera cuyo origen coincida, uno puede superponerse al otro mediante la aplicación de tres rotaciones consecutivas, del siguiente modo:

- Una rotación de ángulo ϕ respecto al eje z
- Una rotación de ángulo θ respecto al eje y' (resultante de y por la rotación anterior)
- Una rotación de ángulo ψ respecto al eje z'' (resultante de z' por la rotación anterior)

Entonces,

$$R_{\phi,\theta,\psi} = \begin{pmatrix} c\phi & -s\phi & 0 & 0 \\ s\phi & c\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c\theta & 0 & s\theta & 0 \\ 0 & 1 & 0 & 0 \\ -s\theta & 0 & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c\psi & -s\psi & 0 & 0 \\ s\psi & c\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\
 \begin{pmatrix} c\phi c\theta c\psi - s\phi s\psi & -c\phi c\theta s\psi - s\phi c\psi & c\phi c\theta & 0 \\ s\phi c\theta c\psi + c\phi s\psi & -s\phi c\theta s\psi + c\phi c\psi & s\phi c\theta & 0 \\ -s\phi c\psi & s\theta s\psi & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Nótese la postmultiplicación, debido a que las rotaciones segunda y tercera se hacen alrededor de los ejes girados.

Respecto a la segunda especificación de orientación (yaw/pitch/roll, $y/p/r$), puede expresarse como:

- Una rotación de ángulo r respecto al eje x
- Una rotación de ángulo p respecto al eje y
- Una rotación de ángulo y respecto al eje z

De este modo,

$$R_{y,p,r} = \begin{pmatrix} cy & -sy & 0 & 0 \\ sy & cy & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} cp & 0 & sp & 0 \\ 0 & 1 & 0 & 0 \\ -sp & 0 & cp & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & cr & -sr & 0 \\ 0 & sr & cr & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} cypc & cyspsr - syrc & cyspcr + sysr & 0 \\ sysc & syspsr + cyrc & syspcr - cysr & 0 \\ -sp & cpsr & cpcr & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.1)$$

Nótese la premultiplicación, por darse todas las rotaciones respecto a los ejes iniciales. Esta matriz deberá, sin embargo, modificarse debido a ciertas asignaciones convencionales de direcciones. Véase la subsección 2.1.5

2.1.4. Transformaciones entre sistemas de coordenadas

Como dijimos anteriormente, un sistema de coordenadas es para nosotros un conjunto de tres ejes rectilíneos y ortogonales, a cada uno de los cuales se asocia un vector unitario. Esto son sistemas ortonormales, pero además nos restringimos a sistemas dextrógiros,¹ en los que $\vec{x} \wedge \vec{y} = \vec{z}$. Dados dos sistemas ortonormales dextrógiros (desde ahora, simplemente sistemas) cualesquiera, siempre es posible encontrar la transformación que liga las coordenadas de un punto cualquiera respecto al segundo con las coordenadas de ese mismo punto respecto al primero. Esta transformación sería la que rotase y trasladase apropiadamente el segundo sistema de modo que sus ejes se superpongan a los del primero, y debería ser usada para pre- o postmultiplicar, según a qué coordenadas queramos referirnos.

Para deducir las ecuaciones de transformación genérica que producirán las coordenadas de un punto cualquiera expresadas en un sistema de referencia a partir de las expresadas en otro, partamos de la figura 2.5.

De ella, es claro que, dado un punto arbitrario X , se cumple que $\vec{OX} = \vec{OP} + \vec{PX}$. Expresando esto en su descomposición de coordenadas respecto a cada sistema,

$$x\vec{i} + y\vec{j} + z\vec{k} = p_x\vec{i} + p_y\vec{j} + p_z\vec{k} + x'\vec{i}' + y'\vec{j}' + z'\vec{k}'$$

Ahora bien, cada vector básico del nuevo sistema puede a su vez ser descompuesto en una combinación lineal de los vectores básicos del sistema antiguo, en la forma:

$$\begin{aligned} \vec{i}' &= \cos(\alpha_{ix})\vec{i} + \cos(\alpha_{iy})\vec{j} + \cos(\alpha_{iz})\vec{k} \\ \vec{j}' &= \cos(\alpha_{jx})\vec{i} + \cos(\alpha_{jy})\vec{j} + \cos(\alpha_{jz})\vec{k} \\ \vec{k}' &= \cos(\alpha_{kx})\vec{i} + \cos(\alpha_{ky})\vec{j} + \cos(\alpha_{kz})\vec{k} \end{aligned}$$

¹Los levógiros, en los que $\vec{x} \wedge \vec{y} = -\vec{z}$, son imagen especular de éstos.

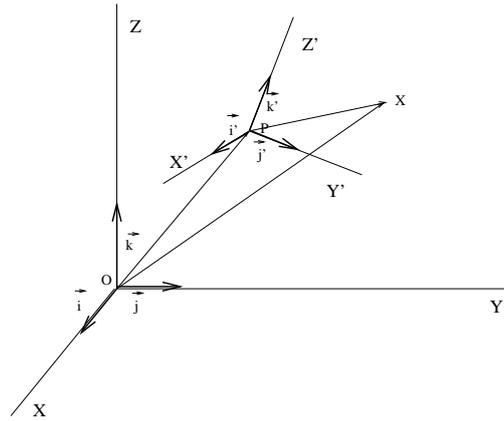


Figura 2.5: Transformaciones entre sistemas

Por tanto, e igualando componente a componente,

$$\begin{aligned}x &= p_x + x' \cos(\alpha_{ix}) + y' \cos(\alpha_{jx}) + z' \cos(\alpha_{kx}) \\y &= p_y + x' \cos(\alpha_{iy}) + y' \cos(\alpha_{jy}) + z' \cos(\alpha_{ky}) \\z &= p_z + x' \cos(\alpha_{iz}) + y' \cos(\alpha_{jz}) + z' \cos(\alpha_{kz})\end{aligned}$$

o expresado como matriz,

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\alpha_{ix}) & \cos(\alpha_{jx}) & \cos(\alpha_{kx}) & p_x \\ \cos(\alpha_{iy}) & \cos(\alpha_{jy}) & \cos(\alpha_{ky}) & p_y \\ \cos(\alpha_{iz}) & \cos(\alpha_{jz}) & \cos(\alpha_{kz}) & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix}$$

Nótese que la matriz homogénea ha subsumido rotación y traslación. Además, las tres primeras columnas son las componentes de los vectores básicos del nuevo sistema referidos al antiguo que, puesto que se trata de vectores unitarios, son los cosenos de los ángulos que cada nuevo vector forma con cada uno de los antiguos.

Podemos considerar la matriz de transformación homogénea como descompuesta del siguiente modo:

$$T = \left(\begin{array}{c|c} R_{3 \times 3} & p_{3 \times 1} \\ \hline f_{1 \times 3} & 1 \times 1 \end{array} \right) = \left(\begin{array}{c|c} Rot. & Tras. \\ \hline Persp. & Esc.global \end{array} \right)$$

La submatriz de rotación 3×3 cumple que:

- $\det(R) = 1$, y si no hay perspectiva ni escalado, $\det(T) = 1$
- El producto escalar de cualesquiera dos columnas es 0
- $R^{-1} = R^T$, o sea, una matriz de giro es ortogonal

El vector f de la cuarta fila se usa para dar cuenta de la transformación perspectiva introducida por cámaras, y la componente de la (cuarta fila, cuarta columna) representa un escalado global. Durante este tema no se considerarán tales aplicaciones, con lo que $f = (0, 0, 0)$ y el escalado global será 1.

2.1.5. El sistema de la mano

Se suele emplear para las componentes de la matriz de transformación homogénea la notación

$$T = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \equiv \begin{pmatrix} \vec{n} & \vec{o} & \vec{a} & \vec{p} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Esto se hace porque la idea final es realizar una cadena de transformaciones entre sistemas de coordenadas que relacione las coordenadas de los objetos medidas respecto a un último sistema solidario al punto terminal (la pinza) del brazo con las coordenadas respecto a un sistema fijo. El último sistema se pondrá de tal modo que su origen coincida con el punto terminal (el punto al centro de las dos garras de la pinza), y además:

- Su eje z se sitúe en la dirección de avance (aproximación, \vec{a})
- Su eje y se sitúe en la dirección de orientación (de garra a garra de la pinza, \vec{o})
- Su eje x resultará normal a los dos anteriores (\vec{n})

Veáse la figura 2.6

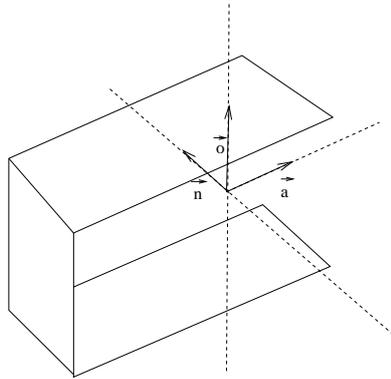


Figura 2.6: Esquema del sistema de la pinza (punto terminal)

Esta asignación tiene consecuencias sobre la elección de los ángulos de orientación. En concreto: al haberse deducido estos ángulos respecto a cualquier sistema de coordenadas fijado según las reglas, en particular, el del punto terminal del robot, no coinciden con la intuición, que los supone fijados respecto a los ejes del mundo. Cuando la mano del robot apunta al frente, entendemos como ángulo de *yaw* su movimiento horizontal, o sea, alrededor del eje z del mundo, no alrededor del z de la mano (vector de aproximación) que nos apunta frontalmente. Para ser consistentes con la forma en la que se suelen dar (y, p, r) deberemos, pues, permutar los ejes en los cuales estaban deducidos, del modo: $x \rightarrow z, y \rightarrow x$ y $z \rightarrow y$, con lo que la matriz $R_{y,p,r}$ dada por la ecuación 2.1 quedará:

$$R_{y,p,r} = \begin{pmatrix} cyspsr - syncr & cyspcr + sysr & cycp & 0 \\ syspsr + cycr & syspcr - cysr & syncp & 0 \\ cpsr & cpcr & -sp & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2.1.6. Inversa de una transformación homogénea

En varias ocasiones será necesario calcular la inversa de una transformación. Para ello, veamos la forma general de la inversa de una matriz de transformación homogénea. Dada

una matriz M de la forma

$$M = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

hallaremos explícitamente cada elemento $m^{-1}(i, j)$ de la inversa de acuerdo a la fórmula usual, $(-1)^{i+j} \det(A(j, i))$, donde $A(j, i)$ es la matriz adjunta al elemento (i, j) , obtenida al suprimir de M la fila i , columna j . Hallemos dos de ellos:

$$m_{11}^{-1} = \det(A(n_x)) = \begin{vmatrix} o_y & a_y & p_y \\ o_z & a_z & p_z \\ 0 & 0 & 1 \end{vmatrix} = o_y a_z - o_z a_y = (\vec{o} \wedge \vec{a})_x = n_x$$

$$m_{14}^{-1} = -\det(A(o_{41})) = - \begin{vmatrix} o_x & a_x & p_x \\ o_y & a_y & p_y \\ o_z & a_z & p_z \end{vmatrix} = - \begin{vmatrix} o_x & o_y & o_z \\ a_x & a_y & a_z \\ p_x & p_y & p_z \end{vmatrix} = -(\vec{o} \wedge \vec{a}) \cdot \vec{p} = -\vec{n} \cdot \vec{p}$$

donde se ha usado que el producto vectorial de dos de los vectores básicos es el otro. Igualmente se efectúan las operaciones para el resto de elementos, obteniendo finalmente que

$$M^{-1} = \begin{pmatrix} n_x & n_y & n_z & -\vec{n} \cdot \vec{p} \\ o_x & o_y & o_z & -\vec{o} \cdot \vec{p} \\ a_x & a_y & a_z & -\vec{a} \cdot \vec{p} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2.2. Cinemática directa del manipulador

2.2.1. El formalismo de Denavit-Hartenberg

Después de todo lo visto, estamos ya en condiciones de abordar el problema de la cinemática directa, el cual trata de encontrar la forma explícita de la función que relaciona el espacio de articulaciones con el espacio cartesiano de posiciones/orientaciones. Esta función

$$f : J \longrightarrow \mathbb{R}^6$$

toma como argumento un vector en el espacio de articulaciones, que tiene tantas componentes como grados de libertad tenga la cadena cinemática que se considere, y devuelve un vector de 6 componentes; las tres primeras serán la posición en el espacio del punto terminal de la cadena, expresada en un sistema de referencia externo, y las tres últimas, la orientación, expresada bien como las componentes (a_x, a_y, a_z) del vector de aproximación, bien como ángulos de orientación (Euler o $y/p/r$).

La forma en que conseguiremos conocer este vector de \mathbb{R}^6 será mediante la construcción de la matriz de transformación homogénea T que relaciona el sistema solidario al punto terminal con un sistema de referencia fijo arbitrariamente escogido, que llamaremos sistema del mundo. En principio, cada uno de los elementos $(a_x, a_y, a_z, o_x, o_y, o_z, n_x, n_y, n_z, p_x, p_y, p_z)$ de la matriz T será función de algunas o todas las variables de articulación, y de las constantes geométricas del manipulador. Según se vió al detallar la forma explícita de la matriz T , multiplicándola por el vector $(0, 0, 0, 1)$, que expresa las coordenadas homogéneas del punto terminal respecto a su propio sistema, obtendremos éstas respecto al sistema del mundo. Y, por otra parte, según se vió en la sección anterior, existen fórmulas que

relacionan los ángulos de orientación en cualquiera de sus expresiones con los elementos de la submatriz de rotación de T .

Seguidamente se describe con detalle el proceso de construcción de T para cadenas cinemáticas abiertas, en las que cada articulación tenga un sólo grado de libertad. El caso de las cadenas cinemáticas cerradas es más complejo, y queda fuera del alcance de este curso. En cambio, el caso de articulaciones con más de un grado de libertad no presenta ningún problema: bastaría con considerar la articulación como si se tratase de dos, unidas por un enlace ficticio de longitud 0.

Sustancialmente, el proceso consiste en fijar un sistema de coordenadas a cada enlace, que se moverá con él, de acuerdo a un conjunto de normas fijas. A continuación, identificar ciertos parámetros geométricos que lo relacionan con el sistema fijo al siguiente enlace, y usarlos para escribir la matriz de transformación homogénea entre cada par de sistemas. Finalmente, el producto de todas las matrices de transformación generará la matriz T . El conjunto de normas que establece cómo deben fijarse los sistemas de coordenadas se conoce como convenio de Denavit-Hartenberg, descrito por éstos en 1955, y a los parámetros geométricos que relacionan los sistemas, parámetros de Denavit-Hartenberg (desde ahora, DH).

Comenzaremos por establecer convenciones para la nomenclatura:

Según se vió en el tema 1, en cadenas cinemáticas abiertas cada par enlace-articulación (link-joint) era un grado de libertad. Numeraremos los links y joints secuencialmente, desde el inicio de la cadena. La base de ésta, fija normalmente al suelo, será el enlace (link) 0, y no se cuenta como grado de libertad. La articulación (joint) 1 será la que conecte la base al primer enlace móvil; los joints comienzan, pues, a numerarse desde 1, y no existe joint al final del último enlace.² Véase la figura 2.7.

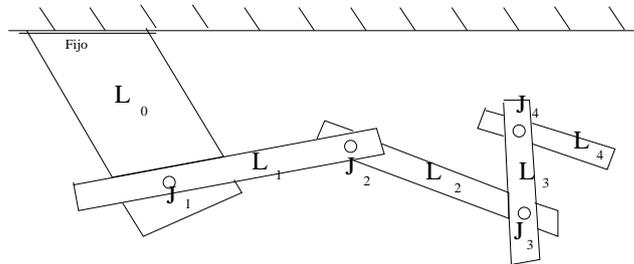


Figura 2.7: Cadena cinemática abierta y su numeración

Eje de una articulación es la recta definida como:

- La dirección de desplazamiento, en articulaciones traslacionales
- El eje de giro, en articulaciones rotacionales

A continuación los parámetros DH de cada enlace son cuatro números reales, dos de ellos representando ángulos, y los otros dos, distancias, definidos del siguiente modo:

- a_i es la mínima distancia (distancia perpendicular) entre el eje de la articulación i y el eje de la $i + 1$. Por extensión, también denotaremos por a_i al segmento de recta a lo largo de cual se da precisamente esa mínima distancia entre ejes.

- α_i es el ángulo que forman el eje i y el $i + 1$, medido en un plano perpendicular al segmento a_i

- d_i es la distancia entre los puntos de intersección de la normal a los ejes $i/i + 1$ con el eje i , y la normal a los ejes $i - 1/i$ también con el eje i , medida a lo largo del este eje.

²Esta es la notación original de Denavit y Hartenberg, y es la más usual. Pero existe la notación modificada, que comienza a numerar los enlaces en el 1, usada, p.ej., por Craig

• θ_i es el ángulo entre la normal a los ejes $i-1/i$ y la normal a los ejes $i/i+1$, medido en un plano perpendicular al eje i .

Véase la representación de estos parámetros en la figura 2.8. Obsérvese que basta el conocimiento de los dos ejes en los extremos de un enlace (digamos, el i) para conocer α_i y a_i ; sin embargo, es necesario conocer los ejes anterior y siguiente para determinar θ_i y d_i . Daremos para estos parámetros una definición alternativa cuando hayamos fijado los respectivos sistemas de coordenadas.

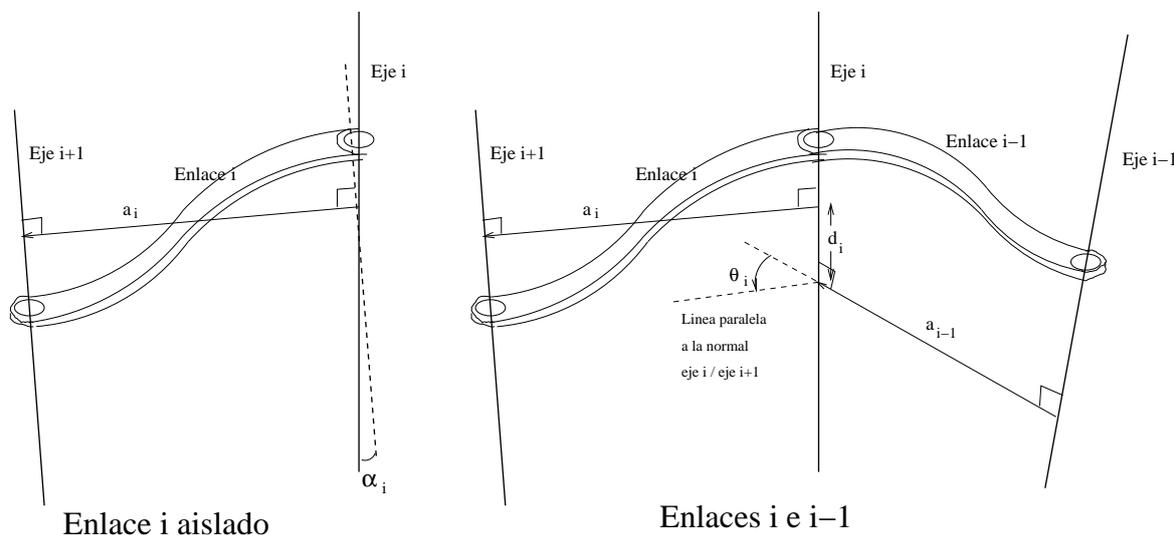


Figura 2.8: Parámetros DH de un enlace genérico

Ahora, las normas para determinar los ejes de cada sistema ortonormal asociado a un link son las que siguen:

- El eje z_{i-1} es el eje de la articulación i . (Con lo que z_i es el de la $i+1$). No importa el sentido a lo largo de la recta en que se oriente. Se recomienda orientar todos los z que sigan la misma dirección en el mismo sentido.
- El eje x_i debe escogerse perpendicular a su propio z (z_i) y también al z anterior (z_{i-1}) **y a lo largo de la perpendicular común** (el segmento a_i).
- El eje y_i se determina de tal modo que el sistema forme un triedro directo (es decir, que $x_i \wedge y_i = z_i$).
- El origen se fija en la intersección de la normal eje $i-1$ /eje i (es decir, el segmento a_{i-1}) con el eje i .

Véanse en la figura 2.9 los sistemas de coordenadas $i-1$ e i situados sobre los enlaces del ejemplo anterior.

Hay algunas excepciones y casos particulares que deben tenerse en cuenta:

- El eje x_0 debe fijarse arbitrariamente como cualquier perpendicular a z_0 , dado que no existe ningún z_{-1} al cual hacerle ser también perpendicular. Análogamente, el origen del sistema 0 es un punto arbitrario sobre el eje z_0 .
- Debe existir un último sistema de coordenadas, fijo a la mano o herramienta, que no sigue las mismas reglas que los anteriores. Debe situarse de modo que su eje z esté en la dirección de aproximación (el avance natural de la mano) y su eje y (vector de orientación) debe apuntar de garra a garra de la pinza (en caso de pinzas con simetría cilíndrica, es arbitrario). Este sistema "especial" sólo debe añadirse si el último sistema obtenido por las reglas usuales no cumpliera estas condiciones. En ese caso, la transformación entre el último sistema natural y este sistema especial es fija, y se halla por observación directa.

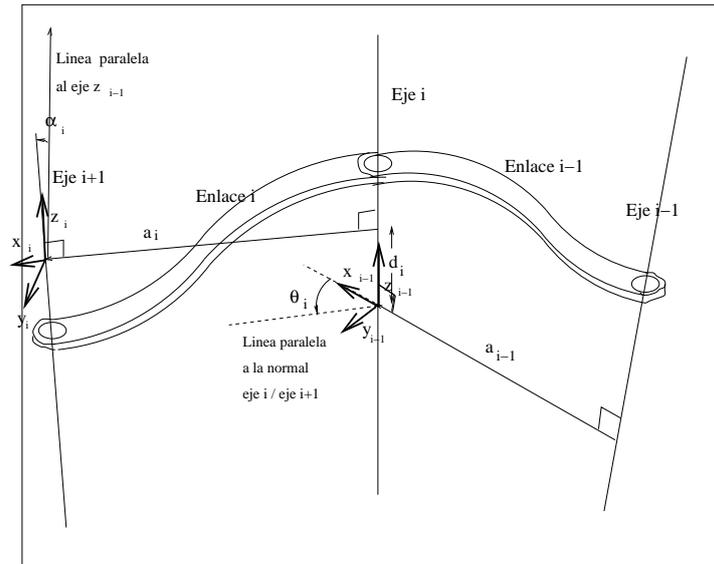


Figura 2.9: Sistemas de coordenadas fijos a una articulación

- Cuando dos ejes z consecutivos son paralelos, hay infinitas perpendiculares comunes. En ese caso, lo normal es tomar el origen a la altura del centro de la articulación, y el eje x a lo largo de la normal común que pasa por ese centro.
- Cuando dos ejes z consecutivos se intersectan, determinan un plano. La normal común es la normal al plano, pero el segmento a_i tiene longitud nula. En ese caso el eje x se escoge normal al plano que determinan los dos z , en cualquiera de los dos sentidos. El origen se toma en el punto de intersección de los dos ejes z .
- Cuando dos ejes z consecutivos son colineales (están superpuestos), el origen se fija arbitrariamente, así como la dirección de x , a lo largo de cualquier perpendicular a los z (que son la misma recta). Se aconseja en ese caso tomar la dirección de x lo más parecida posible a la dirección del x anterior. a_i , que no está definido, se toma como 0.

Nótese que:

- En una articulación rotacional, θ_i es el ángulo de rotación, y d_i , a_i y α_i son constantes.
- En una articulación traslacional, d_i es distancia de traslación, y θ_i , y α_i son constantes. a_i es también constante, y usualmente 0 (suele corresponder al último de los casos particulares anteriores).

Todo parámetro constante en un robot lo es por construcción del mismo, y permanece constante en toda circunstancia, salvo que se altere mecánicamente al robot. La acción de los actuadores que provocarán el movimiento sólo cambia el parámetro variable de cada articulación, θ_i para rotacionales, y d_i para traslacionales. Por otra parte, lo normal es construir robots en los que los ejes de las articulaciones sean o bien paralelos, o bien perpendiculares entre sí; esto hace que los valores para α_i suelen ser bien 0^0 , 90^0 , 180^0 ó 270^0 .

El siguiente paso es determinar la transformación que llevaría el sistema $i - 1$ hasta el i . Nótese que podemos hacerlos coincidir aplicando sucesivamente las siguientes transformaciones:

- Rotar alrededor de z_{i-1} un ángulo θ_i . Esto deja los ejes x_{i-1} y x_i paralelos, pues deja a las rectas a_{i-1} y a_i en el mismo plano, y siendo ambas perpendiculares a z_{i-1} .
- Trasladar a lo largo del eje z'_{i-1} (aunque es el mismo que z_i) el origen una distancia d_i . Esto deja los ejes x_{i-1} y x_i colineales. Nótese que no por ello los ejes z_{i-1} y z_i son aún

coincidentes

- Trasladar a lo largo del eje x_i'' una distancia a_i . Esto hace coincidir los orígenes, y superpone los vectores básicos en la dirección \vec{i} (x_{i-1}'' y x_i'').

- Rotar alrededor de x_{i-1}'' (o de x_i , ahora coinciden) un ángulo α_i . Esto hará coincidir z_{i-1}'' con z_i (y, por tanto, y_{i-1}'' con y_i) y está concluido.

Así pues, podemos escribir la transformación desde el sistema $i - 1$ hasta el i como

$${}^{i-1}A_i = Rot_{z_{i-1}, \theta_i} \cdot Tr_{(0,0,d_i)} \cdot Tr_{(a_i,0,0)} \cdot Rot_{x_i'', \alpha_i}$$

donde las matrices se han postmutiplicado, pues las transformaciones se efectúan siempre respecto a los nuevos ejes que van resultando de la transformación anterior. En forma explícita,

$$\begin{aligned} {}^{i-1}A_i &= \begin{pmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\ &= \begin{pmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

El resultado final es la matriz DH para el enlace i , la cual, conociendo las características geométricas de dicha articulación y de su enlace, da la transformación que lleva de coordenadas expresadas en el sistema i a coordenadas expresadas en el sistema $i - 1$. Igualmente, da la posición del origen del sistema i respecto al $i - 1$ (viendo la 4ª columna), así como su orientación (viendo la submatriz de rotación).

Ahora estamos en condiciones de dar las definiciones alternativas para los parámetros de DH, que son:

- a_i es la distancia perpendicular (distancia mínima) entre los ejes z_{i-1} y z_i , medida en la dirección positiva de x_i .

- α_i es el ángulo que forma el eje z_{i-1} con el eje z_i , girando alrededor de x_i , con x_i apuntando hacia el observador.

- θ_i es el ángulo que forma el eje x_{i-1} con el eje x_i , girando alrededor de z_{i-1} , con z_{i-1} apuntando hacia el observador.

- d_i es la distancia que queda entre los orígenes de los sistemas $i - 1$ y i después de haber trasladado el origen i a lo largo de la perpendicular común a_i hasta situarlo sobre el eje z_{i-1} . Su signo viene dado por el eje z_{i-1} .

Una vez todos los parámetros estén identificados, y las matrices DH escritas, recordemos que lo que se pretende es encontrar la transformación entre el sistema del mundo (sistema 0) y el último (sistema n). Es obvio que

$${}^0A_n = {}^0A_1 \cdot {}^1A_2 \cdot \dots \cdot {}^{n-1}A_n$$

Cada elemento de la matriz ${}^{i-1}A_i$ es función de α_i, a_i, θ_i y d_i , (α_i y a_i constantes para cada robot, y d_i o bien θ_i variables para cada tipo de articulación), y por tanto cada elemento de 0A_n es, en principio, función de todos los ($\alpha_i, a_i, \theta_i, d_i$) con $i = 1 \dots n$.

Resumiendo: los pasos que deben seguirse para la construcción de la cinemática directa son:

1. Identificar cuántos grados de libertad tiene el robot, y cuántas articulaciones; si tuviera alguna articulación con más de un grado de libertad (digamos, n) habrá n sistemas superpuestos en un punto.
2. Asignar los ejes z , sabiendo que z_{i-1} es el eje de la articulación i .
3. Asignar todos los ejes x , sabiendo que x_i es perpendicular a z_i , y a z_{i-1} **y va en la dirección de la perpendicular común**.
4. Asignar los ejes y , de modo que se cumpla que $x_i \wedge y_i = z_i$
5. Determinar los parámetros $(\alpha_i, a_i, \theta_i, d_i)$ de cada enlace por inspección visual.
6. Construir la tabla de parámetros, y a partir de cada una de sus filas, usando la fórmula general de la matriz DH, escribir cada una de las ${}^{i-1}A_i$.
7. Multiplicar todas ellas para generar la 0A_n .

2.2.2. Estudios de caso

Veremos a continuación dos ejemplos interesantes de robots de 6 grados de libertad a los que se aplicarán las normas anteriores con objeto de construir la matriz de transformación homogénea final entre el sistema del mundo y el punto terminal.

El brazo de Stanford

Este manipulador, construido en la Universidad de Stanford con propósitos principalmente académicos, tiene 5 grados de libertad rotacionales, y uno traslacional, en el orden R^2TR^3 , comenzando desde la base. Su estructura puede verse en la figura 2.10.

Nótese que el robot tiene 5 enlaces, que numeraremos de 1 a 5, sin contar el número 0, que como dijimos debe estar fijo al suelo, y en este caso es el propio suelo. El enlace (link) L_1 es la columna vertical; el enlace L_2 es el hombro, que gira y lleva fijada una caja hueca, dentro de la cual se desliza la pieza prismática que es el enlace 3. El enlace 4 es la pequeña pieza prismática que gira alrededor de un eje longitudinal al prisma, y el L_5 es la pieza con dos caras paralelas que gira sobre un eje fijo a la anterior. No existe enlace físico L_6 , pero debemos considerar uno ficticio en el extremo del brazo que dará cuenta del último grado de libertad. Nótese que éste **no** es el correspondiente al sistema especial fijo al punto terminal, que sólo cuando hayamos terminado la asignación sabremos si debemos o no añadir.

Identifiquemos los ejes z :

z_0 es vertical, es el eje del sistema del mundo, de modo que es absolutamente fijo, y se le elige de modo que coincida con el eje de la primera rotación.

z_1 es horizontal; es el eje de la segunda rotación, y está fijo al hombro, de modo que el resto del brazo gira alrededor de él, mientras él mismo gira con el hombro alrededor de z_1

z_2 es un eje de traslación, y sigue la dirección del enlace L_3 . En la posición concreta en la que se ha dibujado el robot, es horizontal, pero en general está inclinado al rotar el hombro.

z_3 sigue también la dirección de z_2 , y es el primer eje de rotación de la mano. En la posición dibujada, es horizontal, pero se le aplica en esto la misma consideración que a z_2

z_4 es el segundo eje de rotación de la mano, y en esta posición aparece vertical, aunque en general se encuentra sobre un plano vertical perpendicular al eje anterior.

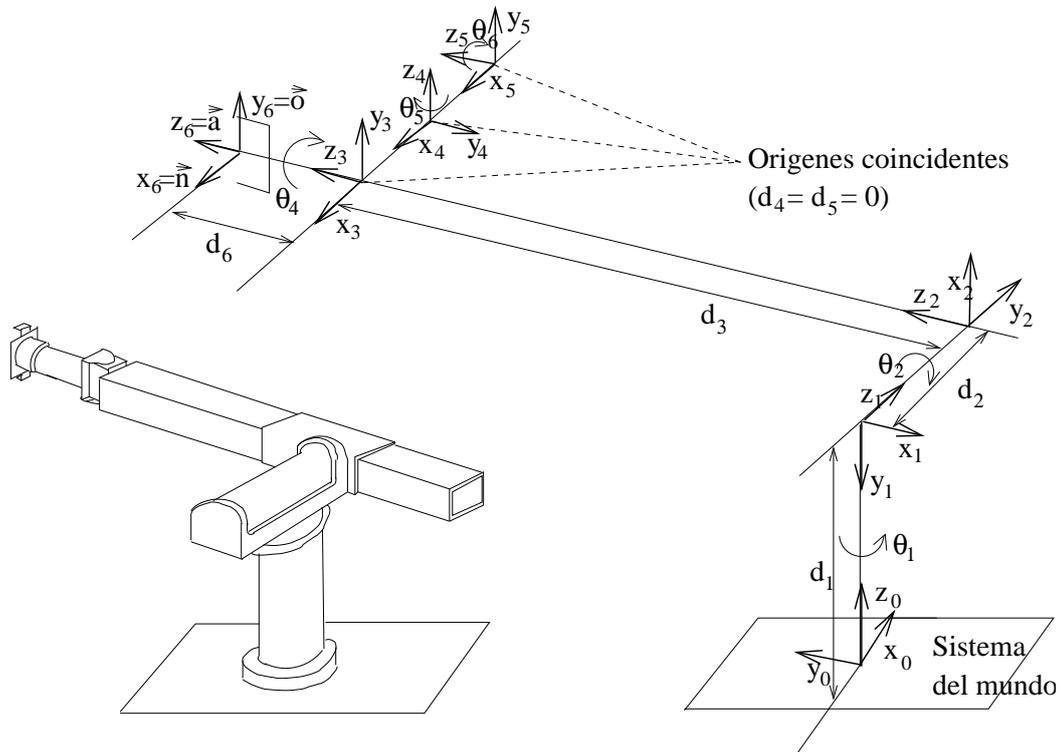


Figura 2.10: Esquema del brazo de Stanford

z_5 es el tercer y último eje de rotación de la mano, y apunta en su propia dirección. En este caso también es horizontal.

z_6 se sitúa colineal (superpuesto) a z_5 . Aun no habiendo un enlace físico al que vaya unido, es necesario incluirlo para que de cuenta de la última rotación. Nótese que cuando las pinzas cambian su orientación, x_6 e y_6 girarán con ellas, pero no el sistema 5. Por otra parte, el robot tiene 6 grados de libertad; por ello, debe haber 6 sistemas móviles (los números 1 a 6, recordemos que el 0 era fijo).

Ahora fijaremos los ejes x :

x_0 será perpendicular a z_0 , y por lo tanto horizontal, pero al no haber un eje z anterior, su dirección se puede elegir. Se elige hacia la derecha, como se ve en la figura.

x_1 debe ser perpendicular a su propio z (z_1) y al z anterior (z_0). Estos dos ejes se cortan, y determinan un plano vertical. Así pues, deberá x_1 ser perpendicular a un plano vertical, y en la posición dibujada, hacia adelante o hacia atrás. Se elige hacia atrás.

x_2 es perpendicular a z_1 y a z_2 , que también se cortan, determinando un plano horizontal, por lo que x_2 deberá ir hacia arriba o hacia abajo. Se elige hacia arriba.

x_3 es un caso especial. Como z_2 y z_3 son coincidentes, hay infinitas direcciones perpendiculares. Se escoge una, en este caso, hacia la izquierda.

x_4 será perpendicular a z_3 y a z_4 , y como éstos se cortan determinando un plano que contendría al eje z_2 , x_4 estaría en dicho plano, y en este caso va hacia la izquierda, aunque girará estando siempre en el plano vertical antedicho.

x_5 está en el mismo caso, y en el mismo plano que el anterior. La posición concreta que se ha escogido para dibujar no es particularmente indicativa. Parece dar a entender que z_5 y z_3 siempre apuntan al frente (y, por tanto, que x_5 y x_3 coinciden). Esto no es cierto; al girar un ángulo arbitrario alrededor de z_3 y otro ángulo cualquiera alrededor de z_4 , z_3 seguirá apuntando al frente, pero no así z_5 .

x_6 se escoge arbitrariamente, por ser z_5 y z_6 colineales. En este caso, hacia la izquierda.

Finalmente, los ejes y se escogen de modo que $x_i \wedge y_i = z_i$.

Respecto a los orígenes, el del sistema 0 se fija arbitrariamente al suelo. Los de los sistemas 1 y 2 vienen determinados sin posibilidad de elección, ya que hay un único punto donde z_0 corta a z_1 , y otro punto único donde z_1 corta a z_2 . El del sistema 3 puede ponerse en cualquier punto a lo largo de la línea, pero sabiendo que se desplazará con el prisma móvil. Se escoge ponerlo en el punto de inicio de la mano. Los orígenes 4 y 5 también son unívocos, dado que sus correspondientes ejes z se cortan. El origen del sistema 6 se escoge en el centro de la pinza. De este modo, si la orientación de este sistema es adecuada, podría servir como sistema final.

Con respecto a esto, nos preguntamos si necesitamos poner otro sistema extra en la mano. Obsérvese que el último sistema que nos resultó por aplicación de las reglas era el 6, y que éste tiene:

Su eje z en la dirección de aproximación de la mano

Su eje y en la dirección de orientación (de garra a garra de la pinza)

Su eje x normal a éstos dos

Por tanto, la convención para los vectores \vec{n} , \vec{o} y \vec{a} se cumple, y no es necesario añadir ningún otro sistema.

Ahora, llenaremos la tabla con los parámetros DH de cada articulación. De acuerdo a las definiciones dadas anteriormente, tendremos:

α_1 es el ángulo entre z_0 y z_1 , girando alrededor de x_1 , con x_1 apuntando hacia el observador. En este caso, el observador miraría desde atrás. Vería a z_0 apuntar hacia arriba, y a z_1 hacia la derecha; el ángulo sería, pues, $-\frac{\pi}{2}$.

a_1 es nula, pues es la distancia entre z_0 y z_1 (a lo largo de la perpendicular común) y éstos se cortan.

d_1 es la distancia entre los orígenes, medida a lo largo de z_0 , después de haber trasladado el origen 0 a lo largo de la perpendicular común. Al no haber perpendicular común, no es necesario trasladar, y d_1 es simplemente distancia entre orígenes. Es constante, y físicamente es la altura de la columna.

θ_1 es el ángulo entre x_0 y x_1 girando alrededor de z_0 , con z_0 apuntando al observador. En este caso, y tal como se ha dibujado, el observador miraría desde arriba, vería x_0 hacia el oeste, y x_1 hacia el norte, de modo que el ángulo sería de $-\frac{\pi}{2}$. Pero no es constante, se trata precisamente de la variable de articulación que representa el giro alrededor de la columna vertical.

Análogamente se obtienen los parámetros para las demás articulaciones, obteniendo al final la tabla siguiente, en la que se indican entre paréntesis los valores que son variables, y cuyo valor es el correspondiente a la posición concreta en que se ha dibujado la figura.

Articulacion	θ_i	α_i	a_i	d_i
1	$(-\frac{\pi}{2})$	$-\frac{\pi}{2}$	0	$d_1 = 412mm.$
2	$(-\frac{\pi}{2})$	$\frac{\pi}{2}$	0	$d_2 = 154mm.$
3	$-\frac{\pi}{2}$	0	0	(d_3)
4	(0)	$-\frac{\pi}{2}$	0	0
5	(0)	$\frac{\pi}{2}$	0	0
6	(0)	0	0	$d_6 = 263mm.$

El vector de articulaciones es en este caso

$$J = \begin{pmatrix} \theta_1 \\ \theta_2 \\ d_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{pmatrix}$$

Ahora se pueden escribir las matrices de transformación ${}^{i-1}A_i$ para cada articulación i , sustituyendo en la fórmula general:

$$\begin{aligned} {}^0A_1 &= \begin{pmatrix} c\theta_1 & 0 & -s\theta_1 & 0 \\ s\theta_1 & 0 & c\theta_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^1A_2 &= \begin{pmatrix} c\theta_2 & 0 & s\theta_2 & 0 \\ s\theta_2 & 0 & -c\theta_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ {}^2A_3 &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^3A_4 &= \begin{pmatrix} c\theta_4 & 0 & -s\theta_4 & 0 \\ s\theta_4 & 0 & c\theta_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ {}^4A_5 &= \begin{pmatrix} c\theta_5 & 0 & s\theta_5 & 0 \\ s\theta_5 & 0 & -c\theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^5A_6 &= \begin{pmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ s\theta_6 & c\theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

Finalmente, deberán multiplicarse todas las matrices. Dos de los productos parciales son:

$$\begin{aligned} {}^0A_3 = {}^0A_1 {}^1A_2 {}^2A_3 &= \begin{pmatrix} c\theta_1 & 0 & -s\theta_1 & 0 \\ s\theta_1 & 0 & c\theta_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c\theta_2 & 0 & s\theta_2 & 0 \\ s\theta_2 & 0 & -c\theta_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\ &= \begin{pmatrix} c\theta_1 & 0 & -s\theta_1 & 0 \\ s\theta_1 & 0 & c\theta_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & c\theta_2 & s\theta_2 & d_3s\theta_2 \\ 0 & s\theta_2 & -c\theta_2 & -d_3c\theta_2 \\ -1 & 0 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s\theta_1 & c\theta_1c\theta_2 & c\theta_1s\theta_2 & d_3c\theta_1s\theta_2 - d_2s\theta_1 \\ -c\theta_1 & s\theta_1c\theta_2 & s\theta_1s\theta_2 & d_3s\theta_1s\theta_2 + d_2c\theta_1 \\ 0 & -s\theta_2 & c\theta_2 & d_3c\theta_2 + d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

Como comprobación, sea la posición dibujada en la que $\theta_1 = -\frac{\pi}{2}$, $\theta_2 = -\frac{\pi}{2}$ y $d_3 = l$, siendo l la longitud a la que el prisma está extendido. En ese caso,

$${}^0A_3 = \begin{pmatrix} -1 & 0 & 0 & d_2 \\ 0 & 0 & 1 & l \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

que es lo esperado, puesto que, de la figura, nótese que, respecto al sistema 0 (sistema del mundo) el origen del sistema 3 está en las coordenadas (d_2, l, d_1) , y además, que el eje x_3 , referido al sistema del mundo, tiene por componentes $(-1, 0, 0)$ (va en la dirección de $-x_0$), así como los ejes y_3 y z_3 van respectivamente en las direcciones de z_0 e y_0 , y por tanto sus componentes son $(0, 0, 1)$ y $(0, 1, 0)$.

Por otra parte, la transformación entre el sistema del inicio de la mano y el punto terminal se obtendría como:

$$\begin{aligned}
 {}^3A_6 &= {}^3A_4 {}^4A_5 {}^5A_6 = \begin{pmatrix} c\theta_4 & 0 & -s\theta_4 & 0 \\ s\theta_4 & 0 & c\theta_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c\theta_5 & 0 & s\theta_5 & 0 \\ s\theta_5 & 0 & -c\theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ s\theta_6 & c\theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\
 &\begin{pmatrix} c\theta_4 & 0 & -s\theta_4 & 0 \\ s\theta_4 & 0 & c\theta_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c\theta_5 c\theta_6 & -c\theta_5 s\theta_6 & s\theta_5 & d_6 s\theta_5 \\ s\theta_5 c\theta_6 & -s\theta_5 s\theta_6 & -c\theta_5 & -d_6 c\theta_5 \\ s\theta_6 & c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\
 &\begin{pmatrix} c\theta_4 c\theta_5 c\theta_6 - s\theta_4 s\theta_6 & -c\theta_4 c\theta_5 s\theta_6 - s\theta_4 s\theta_6 & c\theta_4 s\theta_5 & d_6 c\theta_4 s\theta_5 \\ s\theta_4 c\theta_5 c\theta_6 + c\theta_4 s\theta_6 & -s\theta_4 c\theta_5 s\theta_6 + c\theta_4 s\theta_6 & s\theta_4 s\theta_5 & -d_6 s\theta_4 s\theta_5 \\ -s\theta_5 c\theta_6 & s\theta_5 s\theta_6 & c\theta_5 & d_6 c\theta_5 \\ 0 & 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

La transformación final se obtendría multiplicando la 0A_3 por la 3A_6 .

El RTX/RT100 ©

Los robots RTX y RT100³ son manipuladores, construido por una firma inglesa con propósitos de uso científico, o para movimiento de objetos ligeros, tienen, igual que el brazo de Stanford, 5 grados de libertad rotacionales, y uno traslacional, pero esta vez el orden es TR^5 , comenzando desde la base. Su estructura puede verse en la figura 2.11.

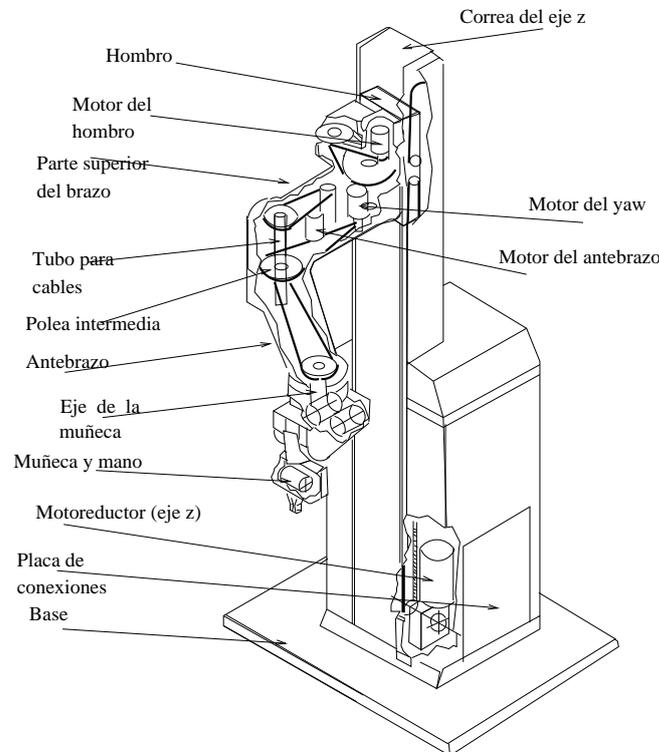


Figura 2.11: Estructura del RT100

En cuanto a sus dimensiones, y vistas lateral y superior, véase la figura 2.12.

³RTX y RT100 son marcas registradas de Oxford Intelligent Machines, Ltd.

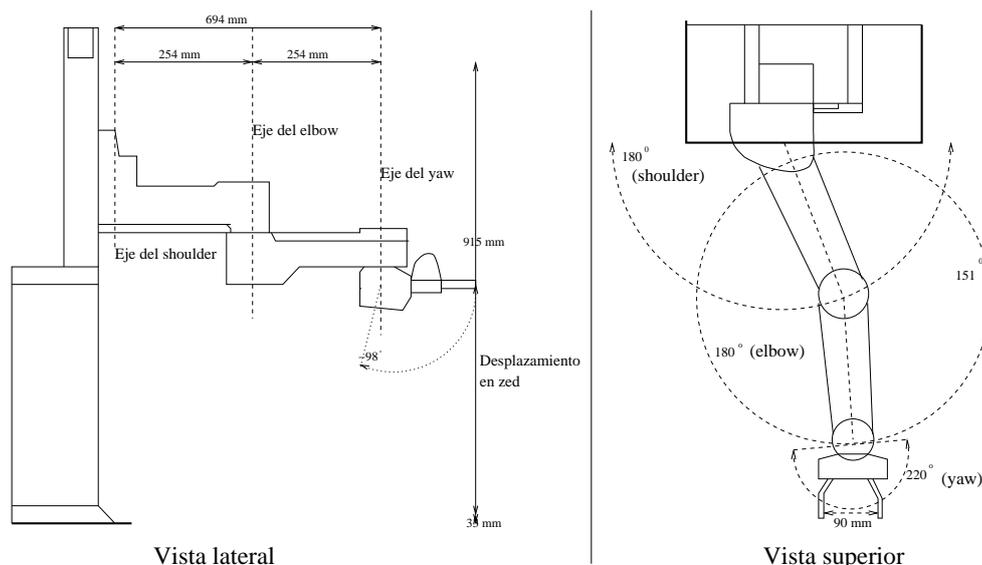


Figura 2.12: Esquema y cotas del robot RT100

La colocación de los ejes se hará según las reglas; véase la figura 2.13

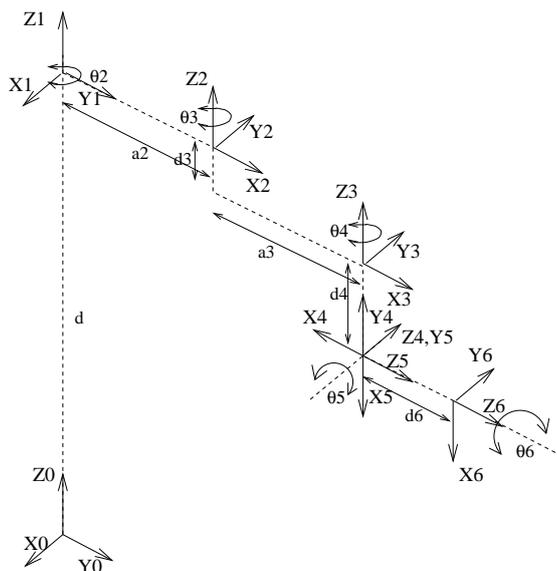


Figura 2.13: Colocación de los sistemas en el RT100

Identifiquemos los ejes z :

z_0 es vertical, es el eje del sistema del mundo, de modo que es absolutamente fijo, y se le elige de modo que coincida con el eje de la primera traslación.

z_1 es también vertical; es el eje de la primera rotación, (segundo grado de libertad) y en este caso, coincide con z_0

z_2 es otro eje de rotación, también vertical (por tanto, paralelo a z_1 , y fijo al enlace 2 (el primer elemento del brazo).

z_3 sigue también la dirección de z_2 , y es el primer eje de rotación de la mano (el que dará el ángulo de yaw). Es también vertical, y se mueve con el enlace 3 (el segundo elemento del brazo).

z_4 es horizontal, y por tanto perpendicular a z_3 , y dará el ángulo de inclinación de la mano (pitch).

z_5 va hacia el exterior, y en la posición dibujada es horizontal, aunque se inclinará con la mano.

z_6 se sitúa colineal (superpuesto) a z_5 . Aun no habiendo un enlace físico al que vaya unido, es necesario incluirlo para que de cuenta de la última rotación (el ángulo de roll), Nótese que cuando las pinzas cambian su orientación, x_6 e y_6 girarán con ellas, pero no el sistema 5. Por otra parte, el robot tiene 6 grados de libertad; por ello, debe haber 6 sistemas móviles (los números 1 a 6, recordemos que el 0 era fijo).

Ahora fijaremos los ejes x :

x_0 será perpendicular a z_0 , y por lo tanto horizontal, pero al no haber un eje z anterior, su dirección se puede elegir. Se elige hacia la izquierda, (mirando el robot de frente) como se ve en la figura.

x_1 debe ser perpendicular a su propio z (z_1) y al z anterior (z_0). Estos dos ejes son colineales, por tanto cualquier perpendicular común será válida. Se escoge la que sigue la misma dirección que el x anterior.

x_2 es perpendicular a z_1 y a z_2 , y sigue la dirección de la perpendicular común, que va a lo largo del primer elemento del brazo. Tal como está dibujado, apunta hacia afuera.

x_3 es exactamente el mismo caso que x_2 , y se le aplican, por tanto, las mismas consideraciones.

x_4 será perpendicular a z_3 y a z_4 , y como éstos se cortan determinando un plano vertical cuyo vector normal sería (en la posición del dibujo) y_0 . x_4 sería perpendicular a dicho plano, y en este caso va hacia atrás, aunque girará con el yaw.

x_5 es perpendicular a z_4 y a z_5 , que determinan un plano que en el dibujo es horizontal, luego x_5 sería así vertical, y se escoge hacia abajo. No obstante, nótese que se inclinará con la mano. Cuando ésta estuviese vertical (pitch de -90°) x_5 se habría puesto horizontal.

x_6 es perpendicular a los ejes colineales z_5 y z_6 , luego hay infinitas elecciones; se escoge una por semejanza con x_5 .

Finalmente, los ejes y se escogen de modo que $x_i \wedge y_i = z_i$.

Respecto a los orígenes, el del sistema 0 se fija arbitrariamente al suelo. El del sistema 1 sube y baja con todo el brazo. El del sistema 2 se podría poner a cualquier altura, se escoge el centro de rotación de su articulación. Lo mismo se puede decir del origen 3. El 4, sin embargo, está unívocamente determinado por la intersección de z_3 y z_4 . Lo mismo ocurre con el 5, que además resulta estar superpuesto al 4. El origen 6 se podría escoger a lo largo de la recta que sigue la dirección de la mano; se toma en el centro de la pinza (punto terminal), por si de este modo el sistema 6 pudiera servir como último sistema.

Nos preguntamos si necesitamos poner otro sistema extra en la mano. Obsérvese que el último sistema que nos resultó por aplicación de las reglas era el 6, y que éste tiene:

Su eje z en la dirección de aproximación de la mano

Su eje y en la dirección de orientación (de garra a garra de la pinza)

Su eje x normal a éstos dos

Por tanto, la convención para los vectores \vec{n} , \vec{o} y \vec{a} se cumple, y, como en el caso del brazo de Stanford, no es necesario añadir ningún otro sistema.

Ahora, llenaremos la tabla con los parámetros DH de cada articulación. De acuerdo a las definiciones dadas anteriormente, tendremos:

α_1 es el ángulo entre z_0 y z_1 . Por ser colineales, es 0.

a_1 es nula, pues no hay distancia entre z_0 y z_1 .

d_1 es la distancia entre los orígenes, medida a lo largo de z_0 , después de haber trasladado

el origen 0 a lo largo de la perpendicular común. Al no haber perpendicular común, no es necesario trasladar, y d_1 es simplemente distancia entre orígenes. Es precisamente el primer grado de libertad, d .

θ_1 es el ángulo entre x_0 y x_1 girando alrededor de z_0 , con z_0 apuntando al observador. En este caso, y tal como se ha dibujado, el observador miraría desde arriba, y vería x_0 y x_1 superpuestos, luego θ_1 es constante, e igual a 0.

α_2 es el ángulo entre z_1 y z_2 . Por ser paralelos, es 0.

a_2 es la distancia entre z_1 y z_2 , que es constante e igual a la longitud del primer elemento del brazo.

d_2 es la distancia entre los orígenes, después de haber trasladado el origen 2 a lo largo de la recta a_2 . Al hacer esto, quedarían superpuestos, pues el origen 2 está a la misma altura que el 1, y por tanto d_2 es 0.

θ_2 es el ángulo entre x_1 y x_2 girando alrededor de z_1 , con z_1 apuntando al observador. En este caso, y tal como se ha dibujado, el observador miraría desde arriba, y vería x_1 apuntando hacia el oeste, y x_2 hacia el sur, luego el ángulo sería de $\frac{\pi}{2}$, aunque cambia con el giro, por ser precisamente la variable de articulación.

Sobre el sistema 3, vale todo lo dicho para el dos, excepto el hecho de que en la posición de dibujo, x_2 y x_3 son paralelos, y por tanto θ_3 vale 0, y además que una vez trasladado el origen 3 a lo largo de a_3 , no se superpondría con el dos, sino que quedaría por debajo. Por tanto, hay una distancia d_3 , que es constante y negativa.

α_4 es el ángulo entre z_3 y z_4 girando alrededor de x_4 con x_4 mirando al observador. El observador miraría, pues, desde atrás, y vería z_3 hacia el norte, y z_4 hacia el oeste, luego α_4 valdrá $\frac{\pi}{2}$.

a_4 valdrá 0, pues z_3 y z_4 se cortan.

θ_4 es el ángulo entre x_3 y x_4 girando alrededor de z_3 , con z_3 apuntando al observador. En este caso, y tal como se ha dibujado, el observador miraría desde arriba, y vería x_3 apuntando hacia el norte, y x_4 hacia el sur, luego el ángulo sería de π , aunque cambia con el giro, por ser precisamente la variable de articulación.

d_4 será simplemente la distancia entre orígenes, pues aquí no hay perpendicular común a lo largo de la cual trasladar. Será constante, y negativa, puesto que se mide a lo largo del eje z_3 , y en este caso va en su sentido negativo.

α_5 es el ángulo entre z_4 y z_5 girando alrededor de x_5 con x_5 mirando al observador. El observador miraría, pues, desde abajo, supongamos que tumbado con los pies hacia el sistema del mundo (el 0), con lo que vería z_4 hacia el este, y z_5 hacia el norte, luego α_5 valdrá $\frac{\pi}{2}$, en la posición dibujada (aunque variará al girar).

a_5 es nula, por cortarse z_4 y z_5 .

d_5 es nula, por coincidir los orígenes 4 y 5.

α_6 es el ángulo entre z_5 y z_6 , que son colineales, por tanto vale 0.

a_6 es la distancia entre estos mismo ejes, y por tanto es igualmente 0.

θ_6 es el ángulo entre x_5 y x_6 girando alrededor de z_5 , con z_5 apuntando al observador. En la posición dibujada, ambos x apuntan hacia abajo, luego θ_6 valdrá 0.

d_6 es la distancia entre los orígenes 5 y 6 (no hay perpendicular común a lo largo de la cual trasladar), y es constante e igual a la longitud de la mano. En este caso es positiva,

pues se mide a lo largo de z_5 , cuyo sentido positivo apunta hacia el origen 6.

Se obtiene al final la tabla siguiente, en la que se indican entre paréntesis los valores que son variables, y cuyo valor es el correspondiente a la posición concreta en que se ha dibujado la figura.

Articulacion	θ_i	α_i	a_i	d_i
1	0	0	0	(d)
2	$(\frac{\pi}{2})$	0	$a_2 = 254mm.$	0
3	(0)	0	$a_3 = 254mm.$	$d_3 = -75mm.$
4	(π)	$\frac{\pi}{2}$	0	$d_4 = -85mm.$
5	$(\frac{\pi}{2})$	$\frac{\pi}{2}$	0	0
6	(0)	0	0	$d_6 = 176mm.$

El vector de articulaciones es en este caso

$$J = \begin{pmatrix} d \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{pmatrix}$$

Ahora se pueden escribir las matrices de transformación ${}^{i-1}A_i$ para cada articulación i , sustituyendo en la fórmula general:

$$\begin{aligned} {}^0A_1 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^1A_2 &= \begin{pmatrix} c\theta_2 & -s\theta_2 & 0 & a_2c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & a_2s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ {}^2A_3 &= \begin{pmatrix} c\theta_3 & -s\theta_3 & 0 & a_3c\theta_3 \\ s\theta_3 & c\theta_3 & 0 & a_3s\theta_3 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^3A_4 &= \begin{pmatrix} c\theta_4 & 0 & s\theta_4 & 0 \\ s\theta_4 & 0 & -c\theta_4 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ {}^4A_5 &= \begin{pmatrix} c\theta_5 & 0 & s\theta_5 & 0 \\ s\theta_5 & 0 & -c\theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^5A_6 &= \begin{pmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ s\theta_6 & c\theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

Finalmente, deberán multiplicarse todas las matrices. Dos de los productos parciales son:

$$\begin{aligned} {}^0A_3 &= {}^0A_1 {}^1A_2 {}^2A_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c\theta_2 & -s\theta_2 & 0 & a_2c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & a_2s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c\theta_3 & -s\theta_3 & 0 & a_3c\theta_3 \\ s\theta_3 & c\theta_3 & 0 & a_3s\theta_3 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\ & \begin{pmatrix} \cos(\theta_2 + \theta_3) & -\sin(\theta_2 + \theta_3) & 0 & a_3 \cos(\theta_2 + \theta_3) + a_2 \cos(\theta_2) \\ \sin(\theta_2 + \theta_3) & \cos(\theta_2 + \theta_3) & 0 & a_3 \sin(\theta_2 + \theta_3) + a_2 \sin(\theta_2) \\ 0 & 0 & 1 & d_2 + d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ {}^3A_6 &= {}^3A_4 {}^4A_5 {}^5A_6 = \begin{pmatrix} c\theta_4c\theta_5c\theta_6 + s\theta_4s\theta_6 & -c\theta_4c\theta_5s\theta_6 + s\theta_4c\theta_6 & c\theta_4s\theta_5 & c\theta_4s\theta_5d_6 \\ s\theta_4c\theta_5c\theta_6 - c\theta_4s\theta_6 & -s\theta_4c\theta_5s\theta_6 - c\theta_4c\theta_6 & s\theta_4s\theta_5 & s\theta_4s\theta_5d_6 \\ s\theta_5c\theta_6 & -s\theta_5s\theta_6 & -c\theta_5 & -c\theta_6d_6 - d_4 \end{pmatrix} \end{aligned}$$

Los elementos de la matriz 0A_6 resultan ser:

$$A_{11} = c(\theta_2 + \theta_3 + \theta_4)c\theta_5c\theta_6 + s(\theta_2 + \theta_3 + \theta_4)s\theta_6$$

$$A_{12} = s(\theta_2 + \theta_3 + \theta_4)c\theta_6 - c(\theta_2 + \theta_3 + \theta_4)c\theta_5s\theta_6$$

$$A_{13} = c(\theta_2 + \theta_3 + \theta_4)s\theta_5$$

$$A_{14} = d_6s\theta_5c(\theta_2 + \theta_3 + \theta_4) + a_3c(\theta_2 + \theta_3) + a_2c(\theta_2)$$

$$A_{21} = s(\theta_2 + \theta_3 + \theta_4)c\theta_5c\theta_6 - c(\theta_2 + \theta_3 + \theta_4)s\theta_6$$

$$A_{22} = c(\theta_2 + \theta_3 + \theta_4)c\theta_6 - s(\theta_2 + \theta_3 + \theta_4)c\theta_5s\theta_6$$

$$A_{23} = s(\theta_2 + \theta_3 + \theta_4)s\theta_5$$

$$A_{24} = d_6s\theta_5s(\theta_2 + \theta_3 + \theta_4) + a_3s(\theta_2 + \theta_3) + a_2s(\theta_2)$$

$$A_{31} = s\theta_5c\theta_6$$

$$A_{32} = -s\theta_5s\theta_6$$

$$A_{33} = -c\theta_5$$

$$A_{34} = d_1 - d_3 - d_4 - d_6c\theta_5$$

$$A_{41} = 0$$

$$A_{42} = 0$$

$$A_{43} = 0$$

$$A_{44} = 1$$

2.3. Cinemática inversa del manipulador

Hasta ahora, habíamos obtenido las coordenadas para la posición del punto terminal a partir de los valores de las variables de articulación. Ahora nos planteamos el problema inverso: obtener los valores de los joints, que son realmente lo que se envía al sistema de control del manipulador, a partir de la posición y orientación deseadas para el punto terminal. Esto es lo más usado, y absolutamente necesario, dado que las tareas a realizar o trayectorias a recorrer por un manipulador se dan casi siempre en coordenadas cartesianas referidas a algún sistema fijo.

La cinemática inversa es, en general, mucho más problemática que la directa. Los problemas pueden ser:

- El punto pedido está, por su lejanía, fuera del alcance del robot.
- El punto pedido está fuera del alcance, pero debido a problemas geométricos (unos enlaces del robot chocan con otros).

- El punto pedido puede alcanzarse mediante dos o más combinaciones (valores) del vector de articulación. Cada una de estas combinaciones se llama configuración; es típico el caso de los brazos SCARA, como el del robot RT100, donde cualquier punto alcanzable puede alcanzarse con el codo doblado hacia la derecha, o hacia la izquierda (lo cual también modifica, naturalmente, los valores de las demás articulaciones).

- Las ecuaciones que relacionan las variables de articulación con las variables cartesianas no son solubles, en el sentido de que no se pueden encontrar soluciones separables, llamadas forma cerrada, que den el valor de cada variable cartesiana como función exclusivamente de las variables de articulación. No obstante, se puede probar que para un manipulador de 6 grados de libertad tal que los tres últimos sean rotacionales y sus ejes se intersecten en un punto común siempre existe solución cerrada.

A diferencia de la cinemática directa, no existía hasta ahora ningún método sistemático que permita resolver la cinemática inversa de cualquier manipulador por aplicación mecánica de una reglas fijas. Existían, más bien, técnicas que ayudan, debiendo elegir la más adaptada al caso particular. En 1998 se publicó el primer libro que contiene una exposición

sistemática del problema de la cinemática inversa. Sus autores, C. D. Crane y J. Duffy lo resuelven para el caso de cualquier robot de hasta 6 grados de libertad, dando complejas reglas de aplicación de fórmulas predeterminadas. El carácter introductorio de este curso y la extensión limitada de estos apuntes no permiten la exposición de este método. Por ello, nos limitaremos a ver ahora algunas de las técnicas simples más usadas para resolver la cinemática inversa de robots.

2.3.1. Aproximación directa

Se trata de usar las normas de construcción de la matriz de transformación T entre el mundo y el punto terminal (cinemática directa), escribiendo dicha matriz de modo simbólico, es decir, dejando cada elemento en función de las variables de articulación J_i de las que dependa. A continuación, y usando la interpretación dada para la matriz T , podemos escribir que

$$T = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

y, dados (X, Y, Z, y, p, r) que el usuario debe proveer como especificación de la posición y orientación, se obtienen las ecuaciones

$$\begin{aligned} X &= p_x(J_1 \dots J_6) & y &= \arctan\left(\frac{a_y}{a_x}\right) \\ Y &= p_y(J_1 \dots J_6) & p &= -\arctan\left(\frac{-a_z}{a_x/\cos(y)}\right) \\ Z &= p_z(J_1 \dots J_6) & r &= \arctan\left(\frac{n_z}{o_z}\right) \end{aligned}$$

Se pueden dar diferentes ecuaciones para los ángulos de orientación, usando otros elementos de la matriz. Además, las ecuaciones serían por supuesto diferentes en caso de haber optado por otra representación para la orientación en lugar de la $y/p/r$ (p.ej., los ángulos de Euler). En cualquier caso, la idea es simplemente plantear directamente tantas ecuaciones como grados de libertad haya, y tratar de resolverlas como se pueda. Si hubiera menos de 6 grados de libertad, habría, por supuesto menos ecuaciones. Generalmente, suelen ser ecuaciones no lineales bastante complejas, y es difícil encontrar de modo simple una solución en forma cerrada.

2.3.2. Aproximación geométrica

Normalmente, las cadenas cinemáticas constan de 3 grados de libertad para la posición gruesa (enlaces mayores) más otros 3 para la posición fina y orientación (enlaces menores). Los últimos suelen estar justo antes de la mano; unos y otros no son absolutamente independientes. No obstante, se puede probar que si los tres enlaces menores son rotacionales, y sus ejes se intersectan en un punto común (caso de la mayoría de los robots industriales con manos intercambiables), entonces las 3 últimas articulaciones pueden ser sustituidas por una articulación esférica con tres grados de libertad, y un enlace cuya longitud, fija, es la distancia entre el extremo de los enlaces mayores y el extremo de la mano. Esta distancia es fija precisamente porque ninguno de los 3 últimos enlaces es traslacional, y por tanto el punto terminal está restringido a moverse en la superficie de una esfera centrada en el final de los enlaces mayores, cuyo radio sería la longitud de la mano. Véase la figura 2.14

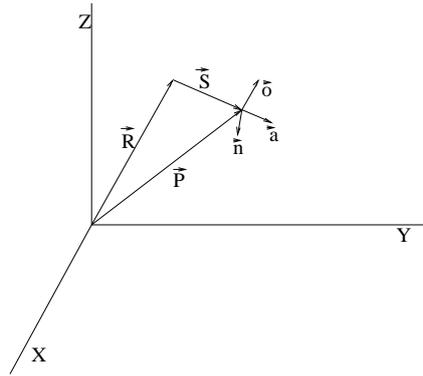


Figura 2.14: Manipulador con los 3 últimos g.d.l. rotacionales

En esta figura, el vector \vec{R} une el origen con el final de los tres enlaces mayores (el punto de inicio de la muñeca). El vector \vec{S} une este punto con el punto terminal, apuntando, por tanto, en la dirección de la mano, y el vector \vec{P} une el origen con el punto terminal, y sus componentes serán las que el usuario pedirá como coordenadas (X, Y, Z) de este punto respecto al sistema del mundo. El sistema situado sobre el punto terminal es el que determina la orientación de la mano, con su vector \vec{a} como aproximación, $\vec{\sigma}$ como orientación (de dedo a dedo de la garra) y \vec{n} normal a éstos.

Son conocidos los vectores \vec{P} y \vec{a} , puesto que el usuario da la posición y orientación. Además, de la figura se observa que \vec{S} y \vec{a} van en la misma dirección; por lo tanto, y como \vec{a} es unitario, podemos escribir que $\vec{S} = |\vec{S}| \vec{a}$. Por otra parte, $|\vec{S}|$ es la longitud de la mano, y es también conocida. Por lo tanto, el vector \vec{S} es completamente conocido. Entoces, podemos conocer \vec{R} ya que $\vec{R} = \vec{P} - \vec{S}$. La simplificación que este método introduce consiste en que \vec{R} , ya conocido, representa el final de los tres enlaces mayores; por tanto, cada componente de \vec{R} deberá igualarse respectivamente a (p_x, p_y, p_z) pero de la transformación 0A_3 , no de la 0A_6 . Esto hace que se tengan que resolver sólo tres ecuaciones con tres incógnitas, que son (J_1, J_2, J_3) . Cada J_i es bien un ángulo θ_i , bien una distancia d_i , según la articulación sea rotacional o traslacional.

Respecto al planteamiento de estas ecuaciones, hay que hacer notar que los tres enlaces mayores pueden responder sólo a una de las siguientes ocho configuraciones: $R^3, R^2T, RTR, TR^2, RT^2, TRT, T^2R, T^3$. Además, en los casos usuales, ejes consecutivos suelen ser o bien paralelos, o bien perpendiculares, con lo que las α_i son $0, \frac{\pi}{2}$ ó $-\frac{\pi}{2}$. Planteando todas las posibles combinaciones de éstos valores α para cada $i = 1 \dots 3$, obtenemos 27 posibilidades por cada configuración, lo que da un total de $8 * 27 = 216$ diferentes matrices 0A_3 . Existen tablas que listan todas estas matrices, y así, para cada robot con una configuración dada y una terna dada de valores $(\alpha_1, \alpha_2, \alpha_3)$ se consulta la tabla para obtener la matriz (y por tanto, las ecuaciones) correspondientes.

Ejemplo: Sea el caso de un robot cuyas tres primeras articulaciones sigan la configuración TR^2 , como el RT100, y en el que $\theta_1 = 0$ (esto siempre es posible, escogiendo el sistema del mundo de modo que los ejes x_0 y x_1 coincidan, recuérdese que la elección de la dirección de x_0 era arbitraria, siempre que fuese perpendicular a z_0). En este caso, la tabla para cada

posible valor de α_i quedaría:

	0A_1	1A_2	2A_3
$\alpha_i = 0$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} c\theta_2 & -s\theta_2 & 0 & a_2c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & a_2s\theta_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} c\theta_3 & -s\theta_3 & 0 & a_3c\theta_3 \\ s\theta_3 & c\theta_3 & 0 & a_3s\theta_3 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
$\alpha_i = \frac{\pi}{2}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} c\theta_2 & 0 & s\theta_2 & a_2c\theta_2 \\ s\theta_2 & 0 & -c\theta_2 & a_2s\theta_2 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} c\theta_3 & 0 & s\theta_3 & a_3c\theta_3 \\ s\theta_3 & 0 & -c\theta_3 & a_3s\theta_3 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
$\alpha_i = -\frac{\pi}{2}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} c\theta_2 & 0 & -s\theta_2 & a_2c\theta_2 \\ s\theta_2 & 0 & c\theta_2 & a_2s\theta_2 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} c\theta_3 & 0 & -s\theta_3 & a_3c\theta_3 \\ s\theta_3 & 0 & c\theta_3 & a_3s\theta_3 \\ 0 & -1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

En el caso del RT100, se tiene que $\alpha_1 = \alpha_2 = \alpha_3 = 0$, con lo que

$${}^0A_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c\theta_2 & -s\theta_2 & 0 & a_2c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & a_2s\theta_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c\theta_3 & -s\theta_3 & 0 & a_3c\theta_3 \\ s\theta_3 & c\theta_3 & 0 & a_3s\theta_3 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} c(\theta_2 + \theta_3) & -s(\theta_2 + \theta_3) & 0 & a_3c(\theta_2 + \theta_3) + a_2c\theta_2 \\ s(\theta_2 + \theta_3) & c(\theta_2 + \theta_3) & 0 & a_3s(\theta_2 + \theta_3) + a_2s\theta_2 \\ 0 & 0 & 1 & d_2 + d_3 + d \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Suponiendo conocidas (por el procedimiento detallado al inicio de esta sección) las componentes del vector \vec{R} , el sistema de tres ecuaciones cuyas tres incógnitas son d , θ_2 y θ_3 queda:

$$\begin{aligned} R_x &= a_3c(\theta_2 + \theta_3) + a_2c\theta_2 \\ R_y &= a_3s(\theta_2 + \theta_3) + a_2s\theta_2 \\ R_z &= d_2 + d_3 + d \end{aligned}$$

De la tercera de estas ecuaciones se obtiene directamente d , como $d = R_z - d_2 - d_3$. Las otras dos, usando el hecho de que en este robot a_2 y a_3 son iguales, y se denotarán sólo como a , se reescriben así:

$$\begin{aligned} R_x &= a(c(\theta_2 + \theta_3) + c\theta_2) \\ R_y &= a(s(\theta_2 + \theta_3) + s\theta_2) \end{aligned}$$

Por manipulación de estas ecuaciones, usando las fórmulas trigonométricas para la suma de cosenos y la suma de senos se llega a

$$\begin{aligned} R_x &= 2a \cos\left(\theta_2 + \frac{\theta_3}{2}\right) \cos \frac{\theta_3}{2} \\ R_y &= 2a \sin\left(\theta_2 + \frac{\theta_3}{2}\right) \cos \frac{\theta_3}{2} \end{aligned}$$

Por división de una ecuación entre otra, y por suma de ellas, ayudándose de la variable auxiliar $\theta_g = \theta_2 + \theta_3/2$, se llega al final a obtener

$$\theta_3 = 2 \arccos \frac{\sqrt{R_x^2 + R_y^2}}{2a}$$

$$\theta_2 = \operatorname{atan2}(R_y, R_x) - \arccos \frac{\sqrt{R_x^2 + R_y^2}}{2a}$$

Ahora, se halla $\vec{R} = \vec{P} - \vec{S}^0$, y de ahí, θ_1, θ_2 y θ_3 , usando las ecuaciones recién deducidas. Con ellas se puede escribir la matriz 0A_3 , que necesitaremos inmediatamente.

Respecto a los otros tres ángulos, se escribe la matriz 3A_6 , que resulta ser

$${}^3A_6 = \begin{pmatrix} c\theta_4 c\theta_5 c\theta_6 + s\theta_4 s\theta_6 & -c\theta_4 c\theta_5 s\theta_6 + s\theta_4 s\theta_6 & c\theta_4 s\theta_5 & c\theta_4 s\theta_5 d_6 \\ s\theta_4 c\theta_5 c\theta_6 - c\theta_4 s\theta_6 & -s\theta_4 c\theta_5 s\theta_6 - c\theta_4 c\theta_6 & s\theta_4 s\theta_5 & s\theta_4 s\theta_5 d_6 \\ s\theta_4 c\theta_6 & -s\theta_4 s\theta_6 & -c\theta_5 & -c\theta_5 d_6 - d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Como el vector \vec{S} , que es conocido, une el origen del sistema 3 con el del 6, podemos escribir

$$S_x^3 = c\theta_4 s\theta_5 d_6$$

$$S_y^3 = s\theta_4 s\theta_5 d_6$$

$$S_z^3 = -c\theta_5 d_6 - d_4$$

en las que \vec{S}^3 es el vector \vec{S} , pero referido al sistema 3. Podemos hallarlo a partir del \vec{S}^0 , que conocemos, y de la matriz de rotación entre los sistemas 0 y 3, la cual obtendremos a partir de la 0A_3 . Es decir:

$$\vec{S}^3 = {}^3R_0 \cdot \vec{S}^0$$

$${}^3R_0 = ({}^0R_3)^T$$

A partir de estas se puede por fin despejar,

$$\theta_4 = \operatorname{atan2}(S_y^3, S_x^3)$$

$$\theta_5 = \arccos\left(-\frac{S_z^3 + d_4}{d_6}\right)$$

Nótese que la posición del punto terminal no depende de θ_6 , y por tanto no encontramos aquí ninguna ecuación para despejarla. La única manera sería relacionarla con el vector de orientación \vec{o} , o con la orientación especificada por el usuario de algún otro modo.

2.3.3. Manipulación de matrices simbólicas

A diferencia del anterior, este método, en principio, puede valer para cualquier manipulador. Usa como base las expresiones simbólicas para la cinemática directa. Se trata de igualar una matriz T de transformación homogénea expresada en función de sus variables simbólicas ($n_x, n_y, n_z, o_x, \dots etc$) al producto de las matrices de la cinemática directa, y pre-multiplicar ambos lados de la igualdad por la inversa de cada una, después de haber hallado sustituciones simbólicas para la variable de articulación de la cual dependa la que en este

momento sea primera matriz del producto. Veamos el mismo caso del apartado anterior, los tres últimos grados de libertad del RT100. Se igualarían

$${}^3A_6 = \begin{pmatrix} c\theta_4 & 0 & s\theta_4 & 0 \\ s\theta_4 & 0 & -c\theta_4 & 0 \\ 0 & 1 & 0 & -d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c\theta_5 & 0 & s\theta_5 & 0 \\ s\theta_5 & 0 & -c\theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ s\theta_6 & c\theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.2)$$

o sea,

$${}^3A_6 = \begin{pmatrix} c\theta_4 c\theta_5 c\theta_6 + s\theta_4 s\theta_6 & -c\theta_4 c\theta_5 s\theta_6 + s\theta_4 s\theta_6 & c\theta_4 s\theta_5 & c\theta_4 s\theta_5 d_6 \\ s\theta_4 c\theta_5 c\theta_6 - c\theta_4 s\theta_6 & -s\theta_4 c\theta_5 s\theta_6 - c\theta_4 c\theta_6 & s\theta_4 s\theta_5 & s\theta_4 s\theta_5 d_6 \\ s\theta_4 c\theta_6 & -s\theta_4 s\theta_6 & -c\theta_5 & -c\theta_5 d_6 - d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.3)$$

Entonces, habrá que tratar de usar la ecuación 2.3 para encontrar una sustitución que nos permita despejar θ_4 en función de las variables simbólicas de T . Entonces, se hallará la inversa de 3A_4 bajo esta sustitución, y se multiplicarán ambas partes de la ecuación 2.2 por $({}^3A_4)^{-1}$. De este modo se conseguirá tener en una parte de la igualdad cantidades conocidas, y en la otra, funciones, no ya de los tres ángulos iniciales, sino sólo de dos (θ_5 y θ_6). Repitiendo el proceso con las inversas de cada matriz, una en cada paso, se llegan a despejar todas. En concreto, de la ecuación 2.3 podemos extraer que

$$\begin{aligned} c\theta_4 s\theta_5 &= a_x \\ s\theta_4 s\theta_5 &= a_y \end{aligned}$$

de donde $\theta_4 = \text{atan2}(a_y, a_x)$, y así

$$\begin{aligned} \cos \theta_4 &= \frac{1}{\sqrt{1 + \tan^2 \theta_4}} = \frac{a_x}{\sqrt{a_x^2 + a_y^2}} \\ \sin \theta_4 &= \frac{1}{\sqrt{1 + \cotan^2 \theta_4}} = \frac{a_y}{\sqrt{a_x^2 + a_y^2}} \end{aligned}$$

Además, escribiendo la inversa de 3A_4 , tenemos que

$$({}^3A_4)^{-1} = \begin{pmatrix} c\theta_4 & s\theta_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ s\theta_4 & -c\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{a_x}{\sqrt{a_x^2 + a_y^2}} & \frac{a_y}{\sqrt{a_x^2 + a_y^2}} & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ \frac{a_y}{\sqrt{a_x^2 + a_y^2}} & -\frac{a_x}{\sqrt{a_x^2 + a_y^2}} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Ahora, multiplicaremos por esta matriz ambas partes de la ecuación 2.2. Se usa que

$$T = {}^3A_4 \cdot {}^4A_5 \cdot {}^5A_6 \implies ({}^3A_4)^{-1} \cdot T = {}^4A_5 \cdot {}^5A_6$$

o sea,

$$\begin{pmatrix} \frac{a_x}{\sqrt{a_x^2 + a_y^2}} & \frac{a_y}{\sqrt{a_x^2 + a_y^2}} & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ \frac{a_y}{\sqrt{a_x^2 + a_y^2}} & -\frac{a_x}{\sqrt{a_x^2 + a_y^2}} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c\theta_5 & 0 & s\theta_5 & 0 \\ s\theta_5 & 0 & -c\theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ s\theta_6 & c\theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Multiplicando, queda

$$\begin{pmatrix} \frac{a_x n_x + a_y n_y}{\sqrt{a_x^2 + a_y^2}} & \frac{a_x o_x + a_y o_y}{\sqrt{a_x^2 + a_y^2}} & \sqrt{a_x^2 + a_y^2} & \frac{a_x p_x + a_y p_y}{\sqrt{a_x^2 + a_y^2}} \\ n_z & o_z & a_z & p_z + d_4 \\ \frac{a_y n_x - a_x n_y}{\sqrt{a_x^2 + a_y^2}} & \frac{a_y o_x - a_x o_y}{\sqrt{a_x^2 + a_y^2}} & 0 & \frac{a_y p_x - a_x p_y}{\sqrt{a_x^2 + a_y^2}} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c\theta_5 c\theta_6 & -c\theta_5 s\theta_6 & s\theta_5 & d_6 s\theta_5 \\ s\theta_5 c\theta_6 & -s\theta_5 s\theta_6 & -c\theta_5 & -d_6 c\theta_5 \\ s\theta_6 & c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Ahora aplicaremos nuevamente alguna sustitución que nos permita despejar θ_5 . En particular, usando los elementos (1, 3) y (2, 3) de las matrices anteriores, al dividirlos se obtiene

$$-\tan \theta_5 = \frac{\sqrt{a_x^2 + a_y^2}}{a_z} \implies \theta_5 = \text{atan2}(\sqrt{a_x^2 + a_y^2}, -a_z)$$

De este modo, y como

$$({}^4A_5)^{-1} = \begin{pmatrix} c\theta_5 & s\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ s\theta_5 & -c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} -a_z & \sqrt{a_x^2 + a_y^2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \sqrt{a_x^2 + a_y^2} & a_z & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Se usará seguidamente que

$$({}^4A_5)^{-1} \cdot ({}^3A_4)^{-1} \cdot T = {}^5A_6$$

de donde se puede despejar θ_6 , que resulta ser

$$\theta_6 = \arccos \left(-\frac{(a_x n_x - a_y n_y) a_z}{\sqrt{a_x^2 + a_y^2}} + n_z \sqrt{1 - a_z^2} \right)$$

(equivalentemente, se podría haber despejado a partir de la penúltima multiplicación, en particular, de los términos A_{31} y A_{32} , aunque se ha seguido el método sistemático hasta el final, como ilustración).

Como se ve, este método es muy general, pero tiene el inconveniente de los signos para los ángulos, que debería eliminarse usando la función atan2 siempre que sea posible. En cualquier caso, el mayor inconveniente es que no siempre tenemos por qué encontrar ecuaciones de las cuales podamos en cada paso despejar la variable que nos interesa para poder continuar. Finalmente, al programar las ecuaciones obtenidas (y esto vale para todos los métodos), es necesario comprobar cuándo aparecen en los denominadores cantidades que puedan anularse (en el ejemplo anterior, $\sqrt{a_x^2 + a_y^2}$, que resultaría nula si el vector de aproximación coincidiera con el eje z del final de los tres primeros enlaces). En ese caso, se debería buscar la fórmula para tratar ese caso particular de un modo geométrico.

2.3.4. Transformaciones de velocidad: el Jacobiano, singularidades y manipulabilidad

Como hemos visto anteriormente, en la inmensa mayoría de los robots la relación entre coordenadas Cartesianas y coordenadas del robot (variables de articulación) no es lineal; no obstante, sabemos que cualquier función f no lineal, pero continua y derivable (de clase C^1) puede ser aproximada en un entorno suficientemente pequeño de cualquier punto x_0 por un desarrollo en serie de Taylor en torno a x_0 que conserve sólo los términos de orden

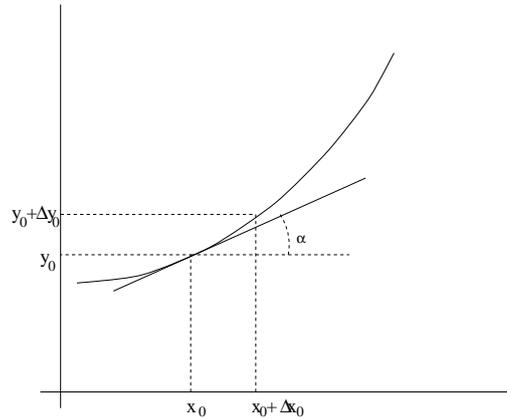


Figura 2.15: Aproximación lineal de una función

0 y 1. En este caso, decimos que hemos aproximado linealmente f en un entorno de x_0 . Intuitivamente, véase la figura 2.15 que se refiere a una función de \mathbb{R} en \mathbb{R} .

En el punto x_0 , la función f vale $y_0 = f(x_0)$. En un punto no muy alejado, podemos aproximarla por

$$f(x_0 + \Delta x_0) \simeq f(x_0) + (\tan(\alpha))\Delta x_0 = y_0 + \left. \frac{df}{dx} \right|_{x=x_0} \cdot \Delta x_0$$

De análoga manera esto se puede aplicar a una función de varias variables, y en concreto lo aplicaremos a la cinemática directa, que como dijimos, va de \mathbb{R}^{n_j} en \mathbb{R}^6 , siendo n_j el número de variables de articulación (joints) del robot. Denotando genéricamente por θ_i a las variables de articulación, (incluso traslacionales), y Θ al vector formado por todas ellas, y denotando por X al vector de variables cartesianas, se tiene que $X = F(\Theta)$ y, para un punto dado, $X_0 + \Delta X$, próximo a X_0 , se puede escribir

$$F(X_0 + \Delta X) \simeq F(X_0) + J_{f_{X_0}} \cdot \delta \Theta$$

siendo $J_{f_{X_0}}$ el Jacobiano de f evaluado en $X_0 = (x, y, z, y, p, r)$, definido como

$$J_{f_{X_0}} = \begin{pmatrix} \frac{\partial x}{\partial \theta_1} & \cdots & \frac{\partial x}{\partial \theta_{n_j}} \\ \frac{\partial y}{\partial \theta_1} & \cdots & \frac{\partial y}{\partial \theta_{n_j}} \\ \frac{\partial z}{\partial \theta_1} & \cdots & \frac{\partial z}{\partial \theta_{n_j}} \\ \frac{\partial y}{\partial \theta_1} & \cdots & \frac{\partial y}{\partial \theta_{n_j}} \\ \frac{\partial p}{\partial \theta_1} & \cdots & \frac{\partial p}{\partial \theta_{n_j}} \\ \frac{\partial r}{\partial \theta_1} & \cdots & \frac{\partial r}{\partial \theta_{n_j}} \end{pmatrix} \quad (2.4)$$

Esto vale para los robots de 6 grados de libertad. La mayoría de éstos, que pueden alcanzar todas las posiciones y orientaciones, tienen 6 joints, al igual que 6 son las componentes del vector X , con lo que el Jacobiano resulta cuadrado, pero no necesariamente tiene que ser así en todos los robots.

Nótese que el Jacobiano varía con el tiempo (en realidad, con la posición, que se supone está a su vez variando). La utilidad que tiene es relacionar velocidades del espacio de articulaciones con velocidades cartesianas; si a partir de la ecuación de incrementos

$$\delta X = J_{f_X} \cdot \delta \Theta$$

derivamos respecto al tiempo, obtenemos para un punto dado X

$$\delta \dot{X} = J_{f_X} \cdot \delta \dot{\Theta}$$

o explícitamente,

$$\begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_y \\ \omega_p \\ \omega_r \end{pmatrix} = \begin{pmatrix} & & & & & \\ & & & & & \\ & & J_f & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \end{pmatrix}_X \cdot \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{pmatrix}$$

Así pues, podemos conocer punto a punto la velocidad cartesiana para un movimiento para el cual habíamos planeado una velocidad de articulación. Sobre la recíproca, dada una velocidad cartesiana de paso por un punto, nos preguntamos si es posible averiguar la velocidad que habría de imprimirse a cada articulación. Despejando tenemos

$$\begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{pmatrix} = \begin{pmatrix} & & & & & \\ & & & & & \\ & & J_f & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \end{pmatrix}_X^{-1} \cdot \begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_y \\ \omega_p \\ \omega_r \end{pmatrix}$$

supuesto que el Jacobiano evaluado en el punto X tenga inversa. Los puntos en los que J no es invertible (o sea, aquellos en los que su determinante es nulo) se llaman **singularidades**. La influencia de estos puntos en el movimiento es muy seria: para una velocidad cartesiana \vec{v} constante, la velocidad de articulación $\dot{\Theta}$ sería infinita. Y aun sin llegar a ese extremo,

$$J_X^{-1} = \frac{1}{\det(J_X)} Ad(J_X)'$$

siendo $Ad(J_X)'$ la matriz adjunta a J_X , traspuesta. Si $\det(J_X)$ es muy pequeño, los elementos de J_X^{-1} serían enormes, y así velocidades cartesianas constantes y de valor razonable para casi todos los puntos generarían sin embargo velocidades de articulación enormes (y, por tanto, inmantenibles por los motores) en las cercanías de una singularidad. La forma de evitar esto es moverse lejos de las singularidades, o bien usar reglas heurísticas (reglas fijas codificadas a mano) para determinar el valor de las velocidades de articulación en estos puntos.

Para conocer cómo afecta este hecho a cada punto del espacio de trabajo, se define el índice de manipulabilidad de Ashada del robot dado en un punto X de su espacio de trabajo como:

$$m(X) = \sqrt{\det(J_X \cdot J_X^T)}$$

siendo \det el determinante, y J_X^T la traspuesta de la matriz Jacobiana en X . Si el Jacobiano es una matriz cuadrada, entonces es claro que

$$m(X) = \det(J)$$

Cuanto mayor sea el valor de este índice, más pequeñas serán las velocidades de articulación que los motores deben procurar, y por tanto, más fácil será controlar el robot.

Veamos ahora brevemente sin demostración unas fórmulas para el cálculo del Jacobiano en robots de 6 grados de libertad (en otros, hay que usar una fórmula genérica del tipo de la definida en la ecuación 2.4, hallada a partir de las fórmulas de la cinemática directa, derivando cada variable cartesiana existente respecto a cada una de las variables de articulación). En el caso de robots de 6 g.d.l. se puede calcular la matriz T que da la cinemática directa como:

$$T = {}^0A_1 \cdot {}^1A_2 \cdot \dots \cdot {}^5A_6$$

Entonces, para cada $i = 1 \dots 6$ se definen las matrices U_i como

$$U_i = {}^{i-1}A_i \cdot {}^iA_{i+1} \cdot \dots \cdot {}^5A_6$$

es decir, el producto de todas las matrices DH de los enlaces que van desde el propio enlace i (incluido) hasta el final de la cadena. En cada caso se tendrá una matriz de la forma

$$U_i = \begin{pmatrix} n_{xi} & o_{xi} & a_{xi} & p_{xi} \\ n_{yi} & o_{yi} & a_{yi} & p_{yi} \\ n_{zi} & o_{zi} & a_{zi} & p_{zi} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Entonces, la columna i -ésima del Jacobiano es:

$$\begin{pmatrix} p_{xi}n_{yi} - p_{yi}n_{xi} \\ p_{xi}o_{yi} - p_{yi}o_{xi} \\ p_{xi}a_{yi} - p_{yi}a_{xi} \\ n_{xi} \\ o_{xi} \\ a_{xi} \end{pmatrix} \qquad \begin{pmatrix} n_{zi} \\ o_{zi} \\ a_{zi} \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Si la art. i es rotacional

Si la art. i es traslacional

De este modo, el procedimiento sería construir cada U_i según la fórmula anterior, y a partir de sus elementos, construir cada columna del Jacobiano usando bien la parte izquierda, o bien la derecha de la ecuación anterior, según el tipo de articulación de que se trate.

2.4. Dinámica del manipulador

En el siguiente desarrollo trataremos de establecer la relación entre las velocidades a las que se mueve cada articulación y la velocidad a la que lo hace el punto terminal del brazo. Esto es importante porque no sólo queremos que el manipulador pase por ciertos puntos del espacio, sino además que lo haga cuando nosotros lo deseamos. Con este objetivo se definen:

Camino: Una sucesión de puntos en el espacio.

Trayectoria: Un camino con restricciones temporales.

Una estrategia muy común para construir caminos es interpolar en el espacio de articulaciones. Esto trae varios problemas:

- No se sabe cuál será el camino en el espacio cartesiano (real). Por ello, si se desea alguno en concreto para evitar choques o mantener una orientación (p. ej., si se transporta un contenedor de líquido), este tipo de interpolación no funciona.

- No se conoce en principio la aceleración cartesiana. Esto puede generar aceleraciones excesivas, que llevarían a grandes fuerzas de inercia, inaceptables para los motores.

Por ello, es generalmente necesario diseñar y programar trayectorias en el espacio cartesiano.

Por otra parte, es importante conocer la relación entre trayectoria cartesiana y de articulaciones para pequeños desplazamientos a partir de una posición dada; esta relación, que es la misma que se da entre velocidades cartesianas y de articulación en cada instante, se puede considerar lineal para pequeños desplazamientos, y, como se vió anteriormente, viene dada por la llamada matriz Jacobiana. Se usa especialmente en los movimientos finos al final de un ensamblado.

Para determinar velocidad y aceleración del punto terminal necesitamos saber cómo se transforman estas magnitudes, o cualesquiera otras, cuando cambiamos de un sistema de coordenadas a otro que se mueve respecto al primero. Consideremos el caso de un giro.

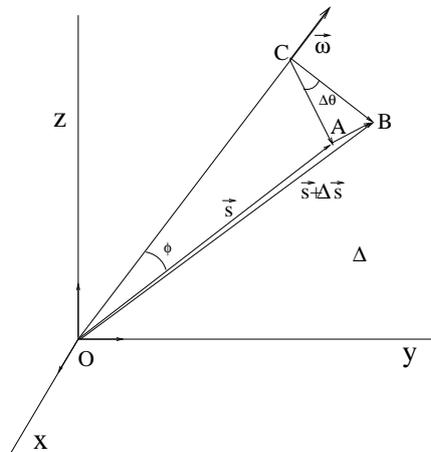


Figura 2.16: Giro de un sistema de referencia centrado en C

Supongamos unos ejes que giran con cierta velocidad angular, $\vec{\omega}$. Queremos conocer $\frac{d\vec{s}}{dt}$ debida a un giro pequeño, de ángulo $d\theta$. Las coordenadas de \vec{s} referidas al sistema rotante cambian, aun cuando \vec{s} no cambie en sí. Nótese que

$$|\Delta \vec{s}| = |\vec{AB}| = |\vec{CA}| d\theta = |\vec{OA}| \sin(\phi) |\vec{\omega}| dt$$

$$\frac{|\Delta \vec{s}|}{dt} = |\vec{OA}| \sin(\phi) |\vec{\omega}|$$

Nótese que $\Delta \vec{s}$ como vector es perpendicular al plano que forman \vec{s} y \vec{OC} (ó $\vec{\omega}$) y que su módulo es precisamente el del producto vectorial de \vec{s} y $\vec{\omega}$, luego

$$\frac{d\vec{s}}{dt} = \vec{\omega} \wedge \vec{s}$$

(en ese orden, regla de la mano derecha).

En particular, para cualquier magnitud vectorial, esta ecuación vale en cada sistema de referencia. En concreto, si queremos calcular la derivada temporal respecto al sistema de referencia del mundo, al cual representaremos con el superíndice R, sabiendo la derivada en un sistema cualquiera, digamos el N,

$$R\left(\frac{ds}{dt}\right) = {}^R T_N {}^N\left(\frac{ds}{dt}\right) + \vec{\omega} \wedge {}^R T_N \vec{s}$$

El primer término representa la variación, por así decirlo, propia del vector, mientras que el segundo representa la variación debida al giro del sistema N respecto al R. En particular, si tomamos dos sistemas, el i y el i+1,

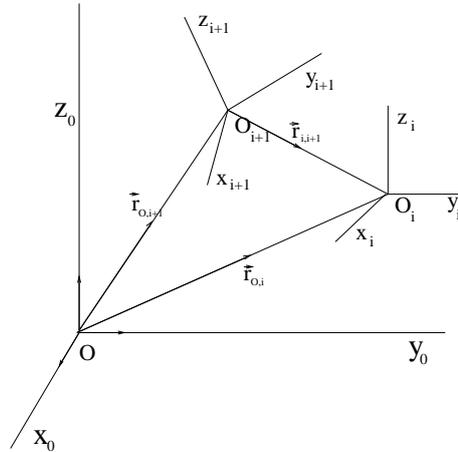


Figura 2.17: Posición de los sistemas O, i e i+1

$$\vec{r}_{o,i+1} = \vec{r}_{o,i} + \vec{r}_{i,i+1}$$

$$\left. \frac{d\vec{r}_{o,i+1}}{dt} \right|_R = \left. \frac{d\vec{r}_{o,i}}{dt} \right|_R + \left. \frac{d\vec{r}_{i,i+1}}{dt} \right|_R$$

Entonces las velocidades son

$${}^R\vec{v}_{i+1} = {}^R\vec{v}_i + \left. \frac{d\vec{r}_{i,i+1}}{dt} \right|_{R_i} + {}^R\vec{\omega}_{i+1} \wedge {}^R\vec{r}_{i,i+1} \tag{2.5}$$

Esto da la velocidad de un elemento (el i+1) conocidas la del anterior (el i), y el movimiento de un elemento respecto a otro.

Nótese que si la articulación es **traslacional**, $\vec{r}_{i,i+1} = d_{i+1}\vec{k}_i$, (siempre se colocaba el eje de traslación en la dirección z) con lo que

$$\left. \frac{d\vec{r}_{i,i+1}}{dt} \right|_{R_i} = \dot{d}_{i+1}\vec{k}_i$$

Por otra parte, la velocidad angular del siguiente elemento respecto a la referencia será la misma que el elemento i lleve, dado que no gira respecto a él. Por ello,

$$\vec{\omega}_{i+1} = \vec{\omega}_i$$

Por contra, si la articulación i es **rotacional**, $\vec{r}_{i,i+1}$ va a lo largo del eje z, que es precisamente el eje de giro, por lo cual no cambia al girar el segundo sistema, y por ello su derivada es nula, es decir,

$$\left. \frac{d\vec{r}_{i,i+1}}{dt} \right|_{R_i} = 0$$

Sin embargo, las velocidades angulares para el último elemento se suman: se tiene la que ya llevaba el elemento anterior, que es con la que gira también el origen de i+1, más la debida al propio giro de la articulación, es decir,

$$\vec{\omega}_{i+1} = \vec{\omega}_i + \dot{\theta}_{i+1}\vec{k}_i$$

Con respecto a la aceleración: derivamos la ecuación 2.5 anterior respecto al tiempo de nuevo y obtenemos

$${}^R\vec{a}_{i+1} = {}^R\vec{a}_i + \left. \frac{d^2\vec{r}_{i,i+1}}{dt^2} \right|_{R_i} + {}^R\vec{\omega}_{i+1} \wedge \left. \frac{d\vec{r}_{i,i+1}}{dt} \right|_{R_i} + {}^R\dot{\vec{\omega}}_{i+1} \wedge \vec{r}_{i,i+1} + {}^R\vec{\omega}_{i+1} \wedge \left[\left. \frac{d\vec{r}_{i,i+1}}{dt} \right|_{R_i} + {}^R\vec{\omega}_{i+1} \wedge \vec{r}_{i,i+1} \right]$$

que es

$${}^R\vec{a}_{i+1} = {}^R\vec{a}_i + \left. \frac{d^2\vec{r}_{i,i+1}}{dt^2} \right|_{R_i} + 2 {}^R\vec{\omega}_{i+1} \wedge \left. \frac{d\vec{r}_{i,i+1}}{dt} \right|_{R_i} + {}^R\dot{\vec{\omega}}_{i+1} \wedge \vec{r}_{i,i+1} + {}^R\vec{\omega}_{i+1} \wedge ({}^R\vec{\omega}_{i+1} \wedge \vec{r}_{i,i+1})$$

Los términos ${}^R\vec{\omega}_{i+1}$ y $\left. \frac{d\vec{r}_{i,i+1}}{dt} \right|_{R_i}$ que intervienen en esta ecuación se han calculado ya para la velocidad; los que aparecen nuevos son:

$$\left. \frac{d^2\vec{r}_{i,i+1}}{dt^2} \right|_{R_i} = \begin{cases} \ddot{d}_{i+1}\vec{k}_i & \text{si } i \text{ es traslacional} \\ 0 & \text{si } i \text{ es rotacional} \end{cases}$$

$$\dot{\vec{\omega}}_{i+1} = \frac{d\vec{\omega}_{i+1}}{dt} = \begin{cases} \frac{d\vec{\omega}_i}{dt} = \dot{\vec{\omega}}_i & \text{si } i \text{ es traslacional} \\ \frac{d(\vec{\omega}_i + \dot{\theta}_{i+1}\vec{k}_i)}{dt} = \dot{\vec{\omega}}_i + \ddot{\theta}_{i+1}\vec{k}_i + \vec{\omega}_i \wedge \dot{\theta}_{i+1}\vec{k}_i & \text{si } i \text{ es rotacional} \end{cases}$$

El resumen de las ecuaciones es (omitiendo en todos los casos el superíndice R , pues todo se mide respecto al sistema de referencia de la base, sistema del mundo, salvo que se indique lo contrario):

<i>Articulacion i es traslacional</i>	<i>Articulacion i es rotacional</i>
$\vec{v}_{i+1} = \vec{v}_i + \dot{d}_{i+1}\vec{k}_i + \vec{\omega}_{i+1} \wedge d_{i+1}\vec{k}_i$ (T1a)	$\vec{v}_{i+1} = \vec{v}_i + \vec{\omega}_{i+1} \wedge \vec{r}_{i,i+1}$ (T1b)
$\vec{a}_{i+1} = \vec{a}_i + \ddot{d}_{i+1}\vec{k}_i + 2\vec{\omega}_{i+1} \wedge \dot{d}_{i+1}\vec{k}_i + \dot{\vec{\omega}}_{i+1} \wedge d_{i+1}\vec{k}_i + \vec{\omega}_{i+1} \wedge (\vec{\omega}_{i+1} \wedge d_{i+1}\vec{k}_i)$ (T2a)	$\vec{a}_{i+1} = \vec{a}_i + \dot{\vec{\omega}}_{i+1} \wedge \vec{r}_{i,i+1} + \vec{\omega}_{i+1} \wedge (\vec{\omega}_{i+1} \wedge \vec{r}_{i,i+1})$ (T2b)
$\vec{\omega}_{i+1} = \vec{\omega}_i$ (T3a)	$\vec{\omega}_{i+1} = \vec{\omega}_i + \dot{\theta}_{i+1}\vec{k}_i$ (T3b)
$\dot{\vec{\omega}}_{i+1} = \dot{\vec{\omega}}_i$ (T4a)	$\dot{\vec{\omega}}_{i+1} = \dot{\vec{\omega}}_i + \ddot{\theta}_{i+1}\vec{k}_i + \vec{\omega}_i \wedge \dot{\theta}_{i+1}\vec{k}_i$ (T4b)

La idea es que, conociendo las posiciones y velocidades lineales y angulares del elemento 0, se calcularán las del 1, y así sucesivamente, hasta llegar al punto terminal. Por supuesto, se deben conocer también las $\theta_i, \dot{\theta}_i, \ddot{\theta}_i$, o en su caso, $d_i, \dot{d}_i, \ddot{d}_i$ que los motores están imprimiendo a cada articulación.

Como el enlace 0 (L_0) está fijo al suelo, los valores usuales son $\vec{v}_0 = 0, \vec{a}_0 = -\vec{g}, \vec{\omega}_0 = 0, \dot{\vec{\omega}}_0 = 0$ siendo \vec{g} el vector aceleración de la gravedad. A partir de ellos se usan las fórmulas anteriores de la columna apropiada para calcular $\vec{\omega}_1, \dot{\vec{\omega}}_1, \vec{v}_1$ y \vec{a}_1 , conociendo $(\theta_1, \dot{\theta}_1, \ddot{\theta}_1)$ o bien $d_1, \dot{d}_1, \ddot{d}_1$, según sea la articulación. Usando estos valores y los $(\theta_2, \dot{\theta}_2, \ddot{\theta}_2)$ se calculan $\vec{\omega}_2, \dot{\vec{\omega}}_2, \vec{v}_2$ y \vec{a}_2 , y así sucesivamente.

Las expresiones usadas representan la formulación de Newton-Euler, y su uso en la forma indicada es el algoritmo de Luh-Walter-Paul. Las ecuaciones de la tabla anterior se llaman ecuaciones hacia adelante, porque van del origen al punto terminal; serían una especie de "cinética directa", en el sentido de que calculan la velocidad cartesiana del punto terminal a partir de las "velocidades de articulación".

El problema más interesante sería el inverso: el que devolviese velocidades y aceleraciones para las variables de articulación a partir de la velocidad y aceleración deseadas para el punto terminal. Esto se puede hacer de dos maneras: numéricamente y analíticamente. En el

primer caso, un número suficiente (idealmente, todos) los puntos de la trayectoria cartesiana, unido cada uno a su respectivo instante deseado de paso por ellos, son convertidos al espacio de articulaciones usando la cinemática inversa. A partir de ahí se interpola linealmente entre cada punto (deben estar muy juntos) cuidando de que la velocidad y aceleración de las articulaciones sean tolerables. Conseguir que cada articulación tome el valor dado por la interpolación en todo instante es tarea del sistema de control, que debe, por tanto, conocer el valor de cada variable de articulación en todo momento, lo que se consigue mediante sensores.

Por contra, en la aproximación analítica es necesario conocer las fuerzas y momentos que van a actuar sobre el punto terminal, y a partir de ellos, calcular los que deberá ejercer cada motor (o actuador) sabiendo, además, las velocidades y aceleraciones que deseamos que lleve.

Para ello, debemos partir de dos ecuaciones fundamentales de la dinámica: la primera ley de Newton aplicada a la mecánica de traslación, y a la de rotación. Estas ecuaciones dicen que si existe un cuerpo de masa m_i sobre el que actúan unas fuerzas, \vec{F}_i y unos momentos, $\vec{\tau}_i$, se cumple

$$\sum_i \vec{F}_i = m_i \vec{a} \quad \text{y tambien} \quad \sum_i \vec{\tau}_i = \frac{d\vec{L}}{dt}$$

donde el torque o momento de una fuerza \vec{F} respecto a un punto O se define como $\vec{\tau} = \vec{r} \wedge \vec{F}$ siendo \vec{r} el vector que une el punto O con el punto de aplicación de la fuerza.

Por otra parte, \vec{L} es el momento angular total del cuerpo, definido como

$$\vec{L} = \int_V \vec{r}_i \wedge \vec{p}_i dV_i$$

donde la integral se extiende a todo el volumen del cuerpo, siendo \vec{r}_i el vector de posición de cada partícula del cuerpo, y \vec{p}_i su momento lineal, o sea, $\vec{p}_i = m_i \vec{v}_i$

Por consideraciones de mecánica elemental que no entran dentro del contenido de este curso, se puede probar que, para un sólido rígido que en un instante dado esté girando con velocidad angular instantánea $\vec{\omega}$ respecto a un cierto eje de giro, el momento angular se puede escribir como

$$\vec{L} = I \vec{\omega}$$

siendo I el llamado tensor de inercia, cuya expresión respecto a algún sistema de coordenadas dado es

$$I = \begin{pmatrix} \int m_i (r_{iy}^2 + r_{iz}^2) dV_i & - \int m_i r_{ix} r_{iy} dV_i & - \int m_i r_{ix} r_{iz} dV_i \\ - \int m_i r_{ix} r_{iy} dV_i & \int m_i (r_{ix}^2 + r_{iz}^2) dV_i & - \int m_i r_{iy} r_{iz} dV_i \\ - \int m_i r_{ix} r_{iy} dV_i & - \int m_i r_{iy} r_{iz} dV_i & \int m_i (r_{ix}^2 + r_{iy}^2) dV_i \end{pmatrix}$$

siendo $\vec{r}_i = (r_{ix}, r_{iy}, r_{iz})$ el vector de posición del punto con masa m_i respecto al sistema de referencia escogido.

Al derivar,

$$\left. \frac{d\vec{L}}{dt} \right|_R = I \dot{\vec{\omega}} + \vec{\omega} \wedge I \vec{\omega}$$

donde tanto el tensor de inercia I como $\vec{\omega}$ y $\dot{\vec{\omega}}$ se miden respecto a la referencia.⁴

⁴La transformación del tensor de inercia desde el sistema de coordenadas de la articulación, respecto al cual se le suele calcular, al sistema del mundo requiere el uso de consideraciones de álgebra tensorial fuera del alcance de este curso. De todos modos, la fórmula es ${}^R(I) = {}^R \mathbf{Rot}_N^T \cdot {}^N(I) \cdot {}^R \mathbf{Rot}_N$ siendo ${}^R \mathbf{Rot}_N$ la matriz de rotación entre el sistema del mundo y el sistema N (T significa traspuesta)

Queda un último detalle antes de poder plantear las ecuaciones dinámicas: necesitaremos usar las ecuaciones hacia adelante dadas antes, pero con las velocidades y aceleraciones del centro de masas de cada articulación, en lugar de las del origen del sistema de referencia asociado a ella. Esto es porque las ecuaciones de la ley de Newton las escribiremos referidas al centro de masas de cada enlace. La relación entre unas y otras es sencilla. Dado un enlace cualquiera

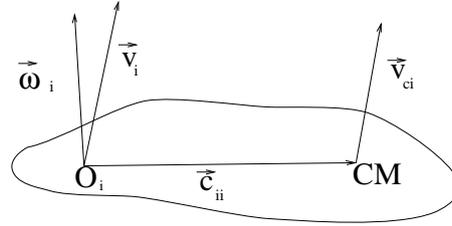


Figura 2.18: La velocidad del CM conocida la del origen

$$\vec{v}_{ci} = \vec{v}_i + \vec{\omega}_i \wedge \vec{c}_{ii} \quad (2.6)$$

siendo \$\vec{c}_{ii}\$ el vector de posición del CM referido a su propio sistema. Es claro que \$\frac{d\vec{c}_{ii}}{dt} = 0\$. Derivando la ecuación anterior obtenemos

$$\vec{a}_{ci} = \vec{a}_i + \dot{\vec{\omega}}_i \wedge \vec{c}_{ii} + \vec{\omega}_i \wedge (\vec{\omega}_i \wedge \vec{c}_{ii}) \quad (2.7)$$

de donde, en cada paso de la iteración se pueden obtener las \$\vec{v}_{ci}\$ y \$\vec{a}_{ci}\$ de las del origen de su enlace, recién obtenidas. Las necesitamos porque ahora aparecen en las ecuaciones dinámicas, que son:

$$\sum_i F_i = m_i \vec{a}_{ci}$$

que queda como

$$\vec{f}_{i-1,i} - \vec{f}_{i,i+1} + m_i \vec{g} = m_i \vec{a}_{ci}$$

siendo \$\vec{f}_{kl}\$ en general la fuerza que un elemento \$k\$ ejerce sobre otro elemento \$l\$, y donde la fuerza del siguiente elemento sobre éste, \$\vec{f}_{i+1,i}\$, se ha escrito como \$-\vec{f}_{i,i+1}\$ usando el principio de acción y reacción. Como anteriormente, todo está referido al sistema del mundo. Por otra parte,

$$\sum_i \tau_i = \frac{d\vec{L}}{dt}$$

queda

$$\vec{\tau}_{i-1,i} - \vec{\tau}_{i,i+1} - \vec{c}_{i-1,i} \wedge \vec{f}_{i-1,i} + \vec{c}_{i,i} \wedge \vec{f}_{i,i+1} = I_i \dot{\vec{\omega}}_i + \vec{\omega}_i \wedge I_i \vec{\omega}_i$$

siendo \$\vec{c}_{i-1,i} = \vec{r}_{i-1,i} + \vec{c}_{ii}\$.

Reordenando estas ecuaciones, lo que tenemos es

$$\vec{f}_{i-1,i} = \vec{f}_{i,i+1} - m_i \vec{g} + m_i \vec{a}_{ci} \quad (2.8)$$

$$\vec{\tau}_{i-1,i} = \vec{\tau}_{i,i+1} + (\vec{r}_{i-1,i} - \vec{c}_{ii}) \wedge \vec{f}_{i-1,i} + \vec{c}_{ii} \wedge \vec{f}_{i,i+1} + I_i \dot{\vec{\omega}}_i + \vec{\omega}_i \wedge I_i \vec{\omega}_i \quad (2.9)$$

Nótese que ahora es necesario ir de adelante hacia atrás: conociendo la fuerza y el momento que ejerce la articulación siguiente en la cadena sobre la dada, se calcula la que ejerce ella sobre la anterior, y así hasta el principio.

La idea es que si, p. ej., queremos levantar un objeto de masa M , situado con su CM en una posición \vec{R} respecto al origen, y el enlace n es el último,

$$\vec{f}_{n,n+1} = -M\vec{g}$$

es decir, el peso del cuerpo actúa como un enlace suplementario ficticio. El signo menos es debido al principio de acción y reacción: esta es en realidad la fuerza con la que el último enlace sostiene al cuerpo, y se opone a su peso. Además,

$$\vec{\tau}_{n,n+1} = \vec{R} \wedge (-M\vec{g})$$

Esto significa que la fuerza y el momento que deberá ejercer en ese instante el motor de la última articulación sería

$$\vec{f}_{n-1,n} = -M\vec{g} - m_n\vec{g} + m_n\vec{a}_{cn}$$

que una vez calculada puede usarse en

$$\vec{\tau}_{n-1,n} = \vec{R} \wedge (-M\vec{g}) + (\vec{r}_{n-1} - \vec{c}_{nn}) \wedge \vec{f}_{n-1,n} + \vec{c}_{nn} \wedge (-M\vec{g}) + I_n\dot{\vec{\omega}}_n + \vec{\omega}_n \wedge I_n\vec{\omega}_n$$

donde las \vec{a}_{cn} y \vec{r}_{n-1} se han calculado en ese mismo instante a partir de la tabla de ecuaciones hacia adelante dadas en la primera parte de esta sección. (recuérdese que \vec{a}_{cn} sería $\vec{a}_{cn} = \vec{a}_n + \dot{\vec{\omega}}_n \wedge \vec{c}_{nn} + \vec{\omega}_n \wedge (\vec{\omega}_n \wedge \vec{c}_{nn})$)

El caso en que el robot no trata de levantar nada, sino sólo moverse (levantarse a sí mismo), es más fácil: $\vec{f}_{n,n+1} = 0$ y $\vec{\tau}_{n,n+1} = 0$. En realidad, las fuerzas no son cantidades que nos interese especialmente conocer (salvo la última) pero son un paso previo al cálculo de los momentos; éstos sí nos interesan, porque, como se verá en el tema 5, en un motor eléctrico el momento (torque) es proporcional a la intensidad de corriente que se le suministra.

Al final, el cuadro resumen de la utilización de todas las ecuaciones como algoritmo sería: (como notación: supondremos que J_i es d_i ó θ_i según proceda)

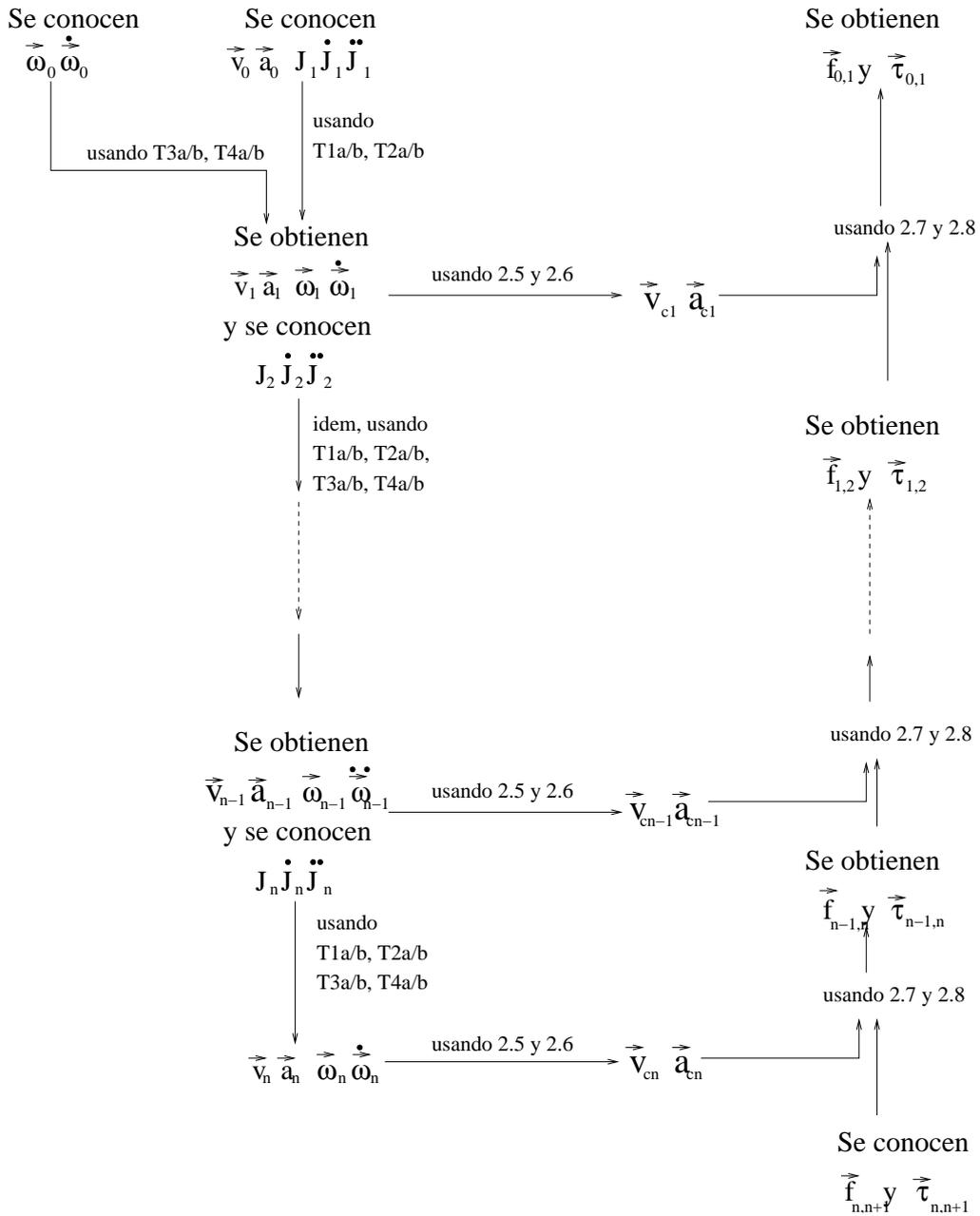


Figura 2.19: Cuadro resumen del uso de las ecuaciones dinámicas

2.5. Generación de trayectorias

2.5.1. Trayectorias en el espacio de articulaciones

Hasta ahora hemos visto cuáles son las coordenadas de articulación que llevarían el punto terminal de un brazo robot a la posición deseada. En esta sección trataremos de ver cómo debería moverse para trasladarse de un punto a otro. Esto significa conocer cuál es la posición, velocidad y aceleración del brazo en función del tiempo.

Es importante decir cómo vamos a especificar la trayectoria. El usuario no debería tener que dar especificaciones detalladas de posición, velocidad y aceleración para cada articulación en cada punto, sino tan sólo valores iniciales y finales; el robot debería saber cómo interpolar para hallar los puntos intermedios. La posición, velocidad y aceleración deben computarse en tiempo real, a una tasa que suele oscilar entre las 20 y 200 veces por segundo.

La trayectoria que normalmente especificamos es la del efector final, referida al sistema de coordenadas de la base. Además de el punto inicial y final, es usual dar algunos puntos intermedios para evitar choques con obstáculos, e incluso el tiempo de paso por cada uno de esos puntos.

La función que describirá la trayectoria va de \mathbb{R} en \mathbb{R}^6 o en \mathbb{R}^{n_j} , puesto que a cada instante de tiempo (un número real) le asignará un vector de valores representando la posición y orientación, bien en el espacio cartesiano, bien en el de articulación.

$$f : \quad \mathbb{R} \longrightarrow \mathbb{R}^6 \\ t \in [0 \dots t_0] \longrightarrow (x, y, z, \phi, p, r)$$

Se puede (y se suele) exigir a esta función que sea suave, entendiendo por tal que sea de clase C^1 (continua y con derivada continua). Hay muchas formas de escoger la función f . Comentaremos las más sencillas.

En primer lugar, especificaremos la trayectoria en el espacio de articulaciones para cada una de ellas por separado. Esto no dará lugar a trayectorias simples para el punto terminal, pero le dejará al final en el lugar requerido, y la trayectoria que éste ejecute será suave (de clase C^1) puesto que sería la composición de la función que elijamos para la variable de articulación (que será C^1) con la cinemática inversa, que, salvo en las singularidades, también lo es.

Para cada articulación, θ , (puede ser rotacional o traslacional) definiremos una función

$$\theta : \quad \mathbb{R} \longrightarrow \mathbb{R} \\ t \longrightarrow \theta(t) \quad \text{con } t \in [0 \dots t_0]$$

y es obvio que la función θ deberá satisfacer ciertos requerimientos:

- Ser continua. Esto es obvio: un objeto no puede encontrarse en una posición en el espacio, y aparecer instantáneamente en otra

- Ser derivable, y con derivada continua; de otro modo, la derivada (que representa la velocidad lineal o angular) de esa articulación estaría en cierto instante no definida, o a su vez, su derivada (la aceleración) sería infinita.

- Debe cumplir que:

$$\theta(0) = \theta_0 \qquad \dot{\theta}(0) = 0 \qquad (2.10)$$

$$\theta(t_f) = \theta_f \qquad \dot{\theta}(t_f) = 0 \qquad (2.11)$$

siendo respectivamente θ_0 y θ_f los valores iniciales y finales de la articulación, y sus derivadas son nulas, porque se supone que parte del reposo, y se llega al reposo.

La elección más sencilla para θ es un polinomio en t , y deberá ser al menos de tercer grado, dado que uno de grado 0 es una constante, lo cual no movería la articulación. Uno de grado 1 es lineal, con lo cual su derivada no podría ser nula en algún punto; y uno de grado dos sólo tiene un máximo o un mínimo, con lo cual su derivada no podría anularse en dos puntos, como se pide.

Así pues, θ será de la forma:

$$\theta(t) = a_0 + a_1t + a_2t^2 + a_3t^3$$

Para determinar los coeficientes a_i aplicaremos las condiciones dadas en las ecuaciones 2.10 y 2.11, lo cual resulta en el sistema:

$$\begin{aligned} a_0 &= \theta_0 \\ a_0 + a_1t_f + a_2t_f^2 + a_3t_f^3 &= \theta_f \\ a_1 &= 0 \\ a_1 + 2a_2t_f + 3a_3t_f^2 &= 0 \end{aligned}$$

que es lineal en las a_i (t_f es conocido), y por tanto, se puede resolver directamente, siendo la solución:

$$\begin{aligned} a_0 &= \theta_0 \\ a_1 &= 0 \\ a_2 &= \frac{3}{t_f^2}(\theta_f - \theta_0) \\ a_3 &= -\frac{2}{t_f^3}(\theta_f - \theta_0) \end{aligned}$$

Si el movimiento no partiese del reposo, o no parase al final, se hace de modo similar, con condiciones ligeramente diferentes.

Ahora, supongamos que queremos llevar la articulación desde un valor inicial, θ_0 hasta un valor final θ_f , pero pasando por una serie de puntos intermedios, θ_i , con $i = 1 \dots n$. Si queremos que la articulación se detenga en cada uno de estos puntos intermedios, no hay problema: se usa el método anterior aplicándolo a cada tramo; pero si, como es usual, no deseamos esto, podemos fijar la velocidad angular o lineal a la que queremos que la articulación pase por cada punto, y aplicar el método a cada segmento. En este caso se tiene para cada tramo:

$$\begin{aligned} \theta(t_{i-1}) &= \theta_{i-1} & a_0 + a_1t_{i-1} + a_2t_{i-1}^2 + a_3t_{i-1}^3 &= \theta_{i-1} \\ \theta(t_i) &= \theta_i & a_0 + a_1t_i + a_2t_i^2 + a_3t_i^3 &= \theta_i \\ \dot{\theta}(t_{i-1}) &= \dot{\theta}_{i-1} & a_1 + 2a_2t_{i-1} + 3a_3t_{i-1}^2 &= \dot{\theta}_{i-1} \\ \dot{\theta}(t_i) &= \dot{\theta}_i & a_1 + 2a_2t_i + 3a_3t_i^2 &= \dot{\theta}_i \end{aligned}$$

Este sistema también se puede resolver sin problemas para cada tramo, pues es lineal (los tiempos de paso, t_i , los fija el usuario, y por tanto en estas ecuaciones son contantes).

Ahora bien, hay aún que buscar alguna manera razonable de fijar las velocidades de articulación, $\dot{\theta}(t_i)$ en cada punto intermedio. Hay tres maneras:

- Especificar en cada punto intermedio velocidad cartesiana (lineal y angular) del punto terminal, y convertirla mediante la cinemática inversa (o usando el Jacobiano para ese punto) en velocidades de articulación.

- Dejar que el sistema escoja la velocidad de articulación en cada punto intermedio usando alguna regla heurística (codificada explícitamente). Una regla razonable sería escoger la velocidad (pendiente de la curva) como la pendiente media entre segmentos rectilíneos contiguos, en el caso de que la velocidad no cambie de signo entre ese tramo y el siguiente, y cero, si dicha velocidad cambia de signo. Véase la figura 2.20

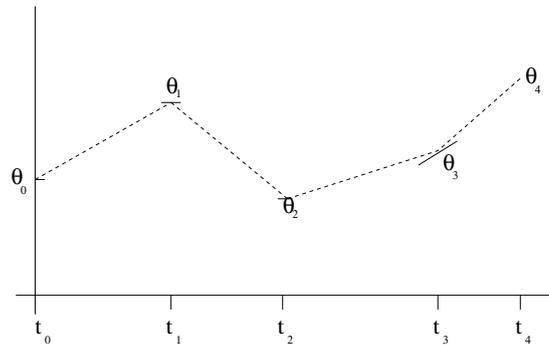


Figura 2.20: Elección de la velocidad intermedia

- Hacer que la aceleración en los puntos intermedios sea continua. Para escribir las ecuaciones de este caso, llamaremos $\theta_a(t)$ a la función (polinomio de tercer grado) que da el valor de la articulación en el primer tramo, y $\theta_b(t)$ al que lo da para el segundo tramo. Su forma explícita, junto con sus correspondientes derivadas, será:

$$\begin{aligned} \theta_a(t) &= a_{10} + a_{11}t + a_{12}t^2 + a_{13}t^3 & \dot{\theta}_a(t) &= a_{11} + 2a_{12}t + 3a_{13}t^2 & \ddot{\theta}_a(t) &= 2a_{12} + 6a_{13}t \\ \theta_b(t) &= a_{20} + a_{21}t + a_{22}t^2 + a_{23}t^3 & \dot{\theta}_b(t) &= a_{21} + 2a_{22}t + 3a_{23}t^2 & \ddot{\theta}_b(t) &= 2a_{22} + 6a_{23}t \end{aligned}$$

Entonces, las ecuaciones para este caso son las siguientes (suponiendo origen de tiempos en el inicio de cada tramo, lo que significa simplemente empezar a contar el tiempo en cada punto intermedio de paso. Se denota así por t_{f1} al tiempo que le cuesta a la articulación recorrer el primer tramo, y t_{f2} al que le cuesta recorrer el segundo):

$$\theta_0 = \theta_a(0) = a_{10} \quad (1)$$

$$\theta_v = \theta_a(t_{f1}) = a_{10} + a_{11}t_{f1} + a_{12}t_{f1}^2 + a_{13}t_{f1}^3 \quad (2)$$

$$\theta_v = \theta_b(0) = a_{20} \quad (3)$$

$$\theta_g = \theta_b(t_{f2}) = a_{20} + a_{21}t_{f2} + a_{22}t_{f2}^2 + a_{23}t_{f2}^3 \quad (4)$$

$$\dot{\theta}_a(t_{f1}) = \dot{\theta}_b(0) \implies a_{11} + 2a_{12}t_{f1} + 3a_{13}t_{f1}^2 = a_{21} \quad (5)$$

$$\ddot{\theta}_a(t_{f1}) = \ddot{\theta}_b(0) \implies 2a_{12} + 6a_{13}t_{f1} = 2a_{22} \quad (6)$$

$$\dot{\theta}_a(0) = 0 \implies a_{11} = 0 \quad (7)$$

$$\dot{\theta}_b(t_{f2}) = 0 \implies a_{21} + 2a_{22}t_{f2} + 3a_{23}t_{f2}^2 = 0 \quad (8)$$

En este sistema, la ecuación (1) expresa el valor inicial, la (2), el valor al final del primer tramo, que hemos llamado θ_v , y que es igual al valor al principio del segundo tramo (ecuación (3)) por continuidad. La ecuación (4) es el valor final que ha de tener θ , al cual hemos llamado θ_g . Por otra parte, la velocidad debe ser continua, por lo que su valor al final del primer tramo igualará al del inicio del segundo, lo cual expresa la ecuación (5). La ecuación (6) expresa la continuidad de las aceleraciones, y es la elección de la que partíamos

en este caso. Las ecuaciones (7) y (8) expresan que se parte del reposo al inicio del primer tramo, y que la articulación se detiene al final del segundo.

Como se ve, ha resultado un sistema de 8 ecuaciones con ocho incógnitas, pero tres de ellas (las numeradas (1),(3) y (7)) son triviales. De este modo, las soluciones para tres de las incógnitas son:

$$a_{10} = \theta_0 \quad a_{20} = \theta_v \quad a_{11} = 0$$

y las cinco ecuaciones resultantes, sustituyendo estas soluciones, son:

$$a_{12}t_{f1}^2 + a_{13}t_{f1}^3 = \theta_v - \theta_0 \quad (2)$$

$$a_{21}t_{f2} + a_{22}t_{f2}^2 + a_{23}t_{f2}^3 = \theta_g - \theta_v \quad (4)$$

$$2a_{12}t_{f1} + 3a_{13}t_{f1}^2 - a_{21}t_{f1}^3 = 0 \quad (5)$$

$$2a_{12} + 6a_{13}t_{f1} - 2a_{22} = 0 \quad (6)$$

$$a_{21} + 2a_{22}t_{f2} + 3a_{23}t_{f2}^2 = 0 \quad (8)$$

sistema que escrito en forma matricial queda:

$$\begin{pmatrix} t_{f1}^2 & t_{f1}^3 & 0 & 0 & 0 \\ 0 & 0 & t_{f2} & t_{f2}^2 & t_{f2}^3 \\ 2t_{f1} & 3t_{f1}^2 & -1 & 0 & 0 \\ 2 & 6t_{f1} & 0 & -2 & 0 \\ 0 & 0 & 1 & 2t_{f2} & 3t_{f2}^2 \end{pmatrix} \begin{pmatrix} a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \end{pmatrix} = \begin{pmatrix} \theta_v - \theta_0 \\ \theta_g - \theta_v \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Hay que hacer dos consideraciones: una alternativa a la ecuación (6) consistiría en que, en lugar de exigir continuidad de la aceleración, en el cambio de tramo, se exija que la velocidad, además de ser continua (ec. (5)) tuviera un valor dado, sea $\dot{\theta}_c$. En ese caso, la nueva ecuación (6) sería $\dot{\theta}_a(t_{f1}) = \dot{\theta}_c$, o sea,

$$a_{11} + 2a_{12}t_{f1} + 3a_{13}t_{f1}^2 = \dot{\theta}_c \quad (\text{o bien, } a_{21} = \dot{\theta}_c, \text{ lo cual, por la ec. (5), es equivalente}).$$

Además de esto: al plantear para dos tramos, hemos obtenido 8 ecuaciones con ocho incógnitas. Análogamente, se podría plantear el problema para n tramos, obteniendo en ese caso $4n$ ecuaciones con $4n$ incógnitas.

Hasta aquí se ha descrito el método que usa como funciones de interpolación polinomios cúbicos. Otro método posible es usar en el centro de cada tramo una interpolación lineal, y en los extremos, parábolas. Se escoge un tiempo t_b , menor que el tiempo total del tramo, t_f ; se trata de interpolar con una parábola desde 0 hasta t_b , con una recta desde t_b hasta $t_f - t_b$, y nuevamente con otra parábola entre $t_f - t_b$ y t_f . Nótese que si el valor de la variable de articulación es cuadrático con el tiempo, su derivada (la velocidad) es una función lineal del tiempo, y a su vez la derivada de ésta (la aceleración) es constante. Esto es bueno para un control sencillo de los motores, ya que, como veremos en el tema 5, la fuerza (o momento) y la aceleración lineal (o angular) se relacionan linealmente.

La gráfica para la variación de la variable de articulación en función del tiempo en cada tramo sería la mostrada en la figura 2.21, la cual es antisimétrica respecto al punto medio.

Entre los instantes 0 y t_b se tendrá:

$$\theta(t) = \theta_0 + \frac{1}{2}\ddot{\theta}t^2$$

por lo que

$$\theta_b = \theta_0 + \frac{1}{2}\ddot{\theta}t_b^2$$

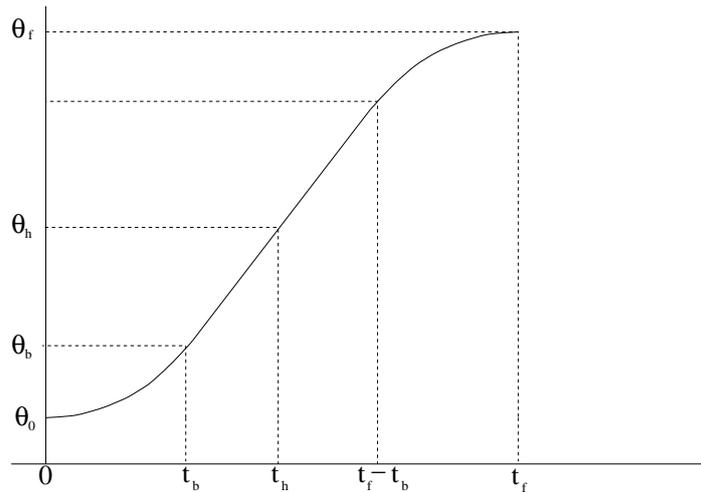


Figura 2.21: Interpolación lineal con extremos parabólicos

y la velocidad en cualquier punto de ese tramo será

$$\dot{\theta}(t) = \ddot{\theta}t \quad \text{en particular, en } t_b, \quad \dot{\theta}(t) = \ddot{\theta}t_b$$

Ahora bien, ésta debe ser igual a la velocidad en el tramo lineal, pues se exige, como siempre, continuidad de la velocidad. De ello se obtiene

$$\ddot{\theta}t_b = \frac{\theta_h - \theta_b}{t_h - t_b} \quad (2.12)$$

Además, como $t_h = t_f/2$, y el valor de la variable en este punto medio es, por simetría de la curva, la media de los valores límite (o sea, $\theta_h = (\theta_0 + \theta_f)/2$), la ecuación 2.12 queda

$$\ddot{\theta}t_b = \frac{\frac{\theta_0 + \theta_f}{2} - (\theta_0 + \frac{1}{2}\ddot{\theta}t_b^2)}{\frac{t_f}{2} - t_b}$$

de la que se obtiene operando

$$\ddot{\theta}t_b^2 - \ddot{\theta}t_f t_b + (\theta_f - \theta_0) = 0$$

que es una ecuación de segundo grado con variable t_b (el resto de factores que aparecen en la ecuación son conocidos, excepto $\ddot{\theta}$, la aceleración, que habremos de elegir. Al resolver en t_b obtenemos

$$t_b = \frac{t_f}{2} - \frac{\sqrt{\ddot{\theta}^2 t_f^2 - 4\ddot{\theta}(\theta_f - \theta_0)}}{2\ddot{\theta}}$$

Para que haya solución, el discriminante de esta ecuación de segundo grado debe ser positivo, o sea,

$$\ddot{\theta}^2 t_f^2 - 4\ddot{\theta}(\theta_f - \theta_0) > 0$$

para lo cual habrá de escogerse $\ddot{\theta}$ de modo que

$$\ddot{\theta} \geq \frac{4(\theta_f - \theta_0)}{t_f^2}$$

En el caso de escogerlo exactamente igual (no mayor), no habría segmento recto central, los dos tramos parabólicos enlazarían directamente.

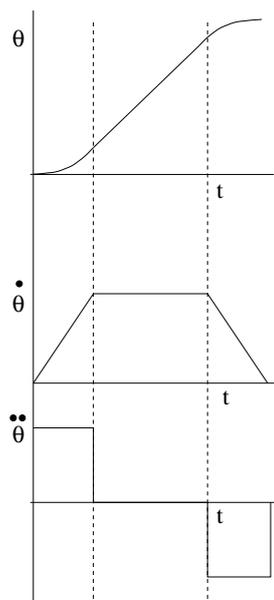


Figura 2.22: Graficas de la variable de articulación, su velocidad y aceleración

En el caso usual las gráficas son las mostradas en la figura 2.22

El método descrito también puede aplicarse al caso de una trayectoria que forzamos a pasar por una serie de puntos dados. En realidad, no pasará exactamente sobre ellos, a menos que nos detengamos. No obstante, será suave. Dados dos puntos, cerca de los cuales se debe pasar en los instantes t_j y t_k , según se aprecia en la figura 2.23, sean los respectivos valores de la variable en ellos θ_j y θ_k , el tiempo entre ambos instantes, $t_{dj k}$, y el tiempo ocupado por el tramo lineal, $t_{j k}$.

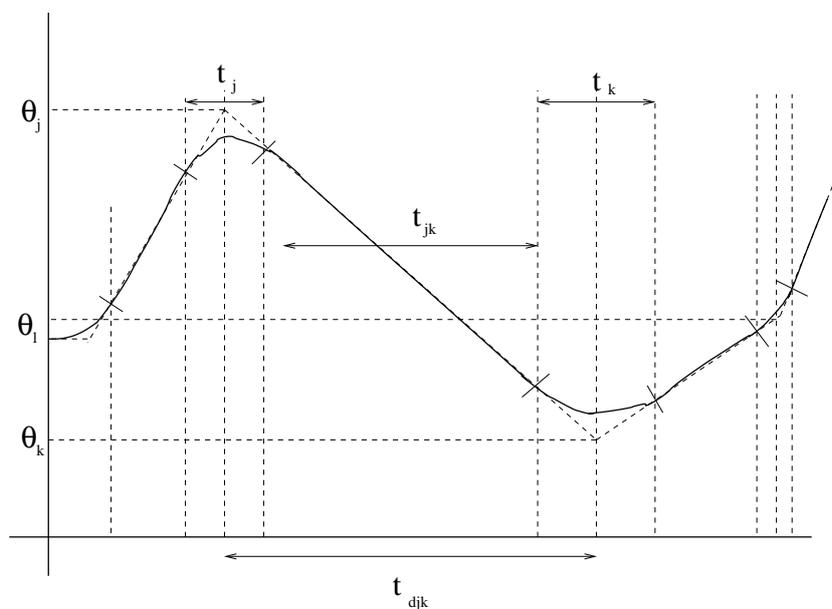


Figura 2.23: Aproximación lineal-parabólica para dos puntos

Trazaremos el tramo lineal de modo que, si se prolongase, pasaría por ambos puntos; en ese caso la velocidad durante él será la pendiente de la recta que pasa por los puntos j

y k , que es

$$\dot{\theta}_{jk} = \frac{\theta_k - \theta_j}{t_{dj k}}$$

donde todos los datos que aparecen son conocidos. Por otra parte, la aceleración para el tramo parabólico, $\ddot{\theta}_k$, debía fijarse a un valor razonable. Se hará esto, teniendo en cuenta que el signo de la aceleración vendrá dado por la diferencia de velocidades. o sea,

$$\ddot{\theta}_k = \text{sig}(\dot{\theta}_{kl} - \dot{\theta}_{jk}) | \ddot{\theta}_k |$$

El tiempo que transcurre durante el tramo parabólico será

$$t_k = \frac{\dot{\theta}_{kl} - \dot{\theta}_{jk}}{\ddot{\theta}_k}$$

(puesto que $\dot{\theta}_{kl} = \dot{\theta}_{jk} + \ddot{\theta}_k t_k$) y por último, el tiempo de paso por el tramo recto sería

$$t_{jk} = t_{dj k} - \frac{1}{2}t_j - \frac{1}{2}t_k$$

Todo esto se aplica a segmentos intermedios. Para los tramos inicial y final existen fórmulas análogas.

Como hemos visto, este método pasa de un modo suave cerca de los puntos intermedios, pero no estrictamente sobre ellos. Para forzar el paso por un punto concreto, i , una solución es situar otros dos puntos intermedios antes y después del mismo ($i - 1$ e $i + 1$), de modo que los tres sean colineales, y que el i esté al centro de los otros. De esta forma, las rectas que unen $i - 1$ con i e i con $i + 1$ son la misma, y cruzarán sobre el punto i , degenerando el tramo parabólico t_i en una recta.

2.5.2. Trayectorias en el espacio cartesiano

Este caso es similar al anterior, pero ahora lo que se pretende es que el punto terminal describa una recta, o alguna curva arbitraria, dada en coordenadas cartesianas. Idealmente, debería calcularse la trayectoria que pase por los puntos indicados siguiendo para cada coordenada cartesiana los métodos de interpolación ya vistos para el espacio de articulaciones. Para interpolar la posición esto no crea problemas, pero para interpolar las orientaciones no podemos usar las componentes de los vectores $\vec{n}, \vec{\sigma}$ y \vec{a} , puesto que perderíamos la ortogonalidad. Habrá que interpolar sobre alguno de los sistemas de ángulos que se describieron anteriormente, los de Euler, o los $y/p/r$.

Aun así, no se puede interpolar usando todos los puntos de la trayectoria (son infinitos). En las situaciones prácticas, se escoge un conjunto discreto, se calcula la cinemática inversa en ellos, y se interpola sobre cada componente del vector de articulación. Esto hará que en el espacio cartesiano la trayectoria no sea exactamente recta, sino que describa pequeños arcos entre cada dos puntos de interpolación; no obstante, si éstos están suficientemente próximos, el efecto es el de un recorrido recto.

Finalmente, hacer notar dos problemas adicionales: el primero es que parte del camino que se desea recorrer esté fuera del espacio de accesibilidad; el programa de generación de trayectorias debería darse cuenta de ello antes de iniciar el movimiento, y por tanto, no hacerlo, avisando al usuario, o incluso generando un camino alternativo.

El segundo problema es que la trayectoria pase sobre o cerca de singularidades. Veámoslo con un ejemplo:

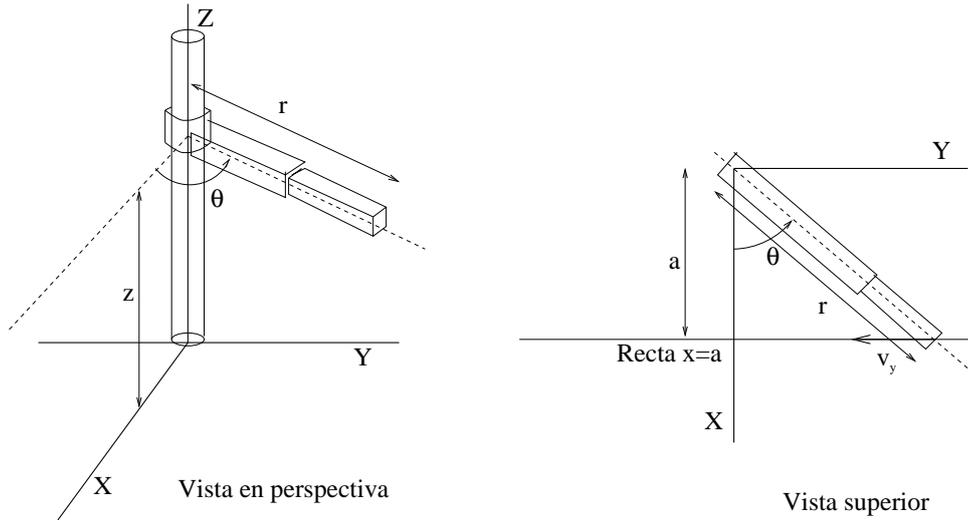


Figura 2.24: Brazo con 3 g.d.l.

Sea el brazo con tres g.d.l. que se muestra en la figura 2.24. del tipo TRT

Fácilmente se puede comprobar que las ecuaciones de la cinemática directa para él resultan ser

$$\begin{aligned}x &= r \cos(\theta) \\y &= r \sin(\theta) \\z &= z\end{aligned}$$

y las de la inversa,

$$\begin{aligned}\theta &= \text{atan2}(y, x) \\r &= \sqrt{x^2 + y^2} \\z &= z\end{aligned}$$

El Jacobiano se calcula de la ecuación 2.4, siendo

$$J = \begin{pmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} & \frac{\partial x}{\partial z} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} & \frac{\partial y}{\partial z} \\ \frac{\partial z}{\partial r} & \frac{\partial z}{\partial \theta} & \frac{\partial z}{\partial z} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -r \sin(\theta) & 0 \\ \sin(\theta) & r \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

De aquí podemos concluir que la relación entre velocidades cartesianas y de articulación será

$$\begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -r \sin(\theta) & 0 \\ \sin(\theta) & r \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{r} \\ \dot{\theta} \\ \dot{z} \end{pmatrix}$$

lo que, despejando las velocidades de articulación por inversión de la matriz, y usando las coordenadas cartesianas, resulta en:

$$\begin{pmatrix} \dot{r} \\ \dot{\theta} \\ \dot{z} \end{pmatrix} = \frac{1}{r} \begin{pmatrix} r \cos(\theta) & r \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & r \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} \frac{x}{\sqrt{x^2+y^2}} & \frac{y}{\sqrt{x^2+y^2}} & 0 \\ \frac{-y}{x^2+y^2} & \frac{x}{x^2+y^2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}$$

por tanto, tendremos

$$\begin{aligned}\dot{r} &= \frac{1}{\sqrt{x^2 + y^2}}(xv_x + yv_y) \\ \dot{\theta} &= \frac{1}{x^2 + y^2}(-yv_x + xv_y) \\ \dot{z} &= \dot{z}\end{aligned}$$

Supongamos que queremos recorrer la trayectoria rectilínea $x = a$, siendo a una constante, con velocidad $v_x = 0$ y $v_y = cte$. En ese caso,

$$\begin{aligned}\dot{r} &= \frac{y}{\sqrt{a^2 + y^2}}v_y \\ \dot{\theta} &= \frac{a}{a^2 + y^2}v_y\end{aligned}$$

En particular, para el punto $x = a$, $y = 0$, tendríamos que imprimir a las articulaciones unas velocidades $\dot{r} = 0$ y $\dot{\theta} = \frac{v_y}{a}$. Esta última sería enorme, si a es muy pequeña, es decir, si pretendemos pasar cerca del origen, y es posible que el motor que controla la articulación rotacional θ no pudiera imprimírsela. Esto significa que la trayectoria rectilínea que habíamos planeado no puede recorrerse a velocidad constante, aun cuando no pasa sobre la singularidad (que está en $(0, 0)$) sino simplemente cerca de ella.

Tema 3: Sensorización

3.1. Necesidad e importancia. Tipos

El desarrollo con éxito de la tarea de un robot depende absolutamente de que éste tenga información correcta y actualizada a un ritmo suficientemente rápido, de su propio estado y de la situación del entorno. En particular, deben conocerse posición, velocidad y aceleración de las articulaciones (al menos, una representación digital de estas magnitudes) para estar seguros de que el robot sigue una determinada trayectoria (según ésta fue definida en el tema 2) y también de que alcanza la posición final deseada en el instante requerido, y con la mínima o ninguna sobreoscilación. Los sensores que permitirán este conocimiento, así como en general todos aquellos que produzcan información sobre el estado del propio robot, serán llamados sensores **internos**.

Por otra parte, en la mayoría de las tareas es necesario conocer datos del mundo que rodea al robot, como distancias a objetos (o contacto con ellos), fuerza ejercida por la mano en las operaciones de prensión, o ejercida por objetos externos (su peso), etc. Este tipo de conocimiento se puede adquirir con dispositivos muy diferentes, desde los más simples (microinterruptores) a los más complejos (cámaras de TV). Todos estos sensores que dan información acerca de lo que rodea al robot serán llamados **externos**.

La importancia de los procesos de sensorización en Robótica debiera ser obvia sin más que examinar el desarrollo de cualquier tarea mínimamente compleja. Sin sensores internos sería imposible establecer los lazos de realimentación (normalmente negativa) que se estudiarán en el tema 5, y que hacen posible el posicionado correcto. Sin sensores externos, cualquier evento inesperado bloquearía el robot, pudiendo dañarlo, y la imprecisión, siempre presente en las magnitudes que definen cualquier tarea (p. ej., las posiciones de las piezas) abortaría cualquier intento de ejecución fiable.

A continuación se detallan las clases de sensores correspondientes a cada tipo que se describirán en el resto del tema.

- Sensores internos
 - De posición
 - Eléctricos
 - Potenciómetros
 - Sincros y resolvers
 - El Inductosyn
 - Opticos
 - Optointerruptores
 - Codificadores absolutos e incrementales
 - Sensores de velocidad
 - Electricos: Dinamos tacométricas
 - Opticos: medición de la velocidad con un encoder
 - Acelerómetros
- Sensores externos
 - De proximidad
 - De contacto: microinterruptores
 - Sin contacto físico
 - De reflexión lumínica (incluyendo infrarrojos)
 - De fibra óptica
 - Scanners laser
 - De ultrasonidos
 - De corriente inducida
 - Resistivos
 - De efecto Hall
 - De tacto
 - De varillas
 - De fotodetectores
 - De elastómeros de conductividad
 - De presión neumática
 - De polímeros (piel artificial)
 - De transferencia de carga
 - De fuerza
 - Por corriente en el motor
 - Por deflexión de los dedos
 - De visión
 - Cámaras de tubo
 - Cámaras lineales CCD
 - Cámaras usuales CCD

3.2. Sensores internos

3.2.1. Sensores de posición

Como su nombre indica, son los que dicen en qué posición, o, más exactamente, en qué punto de su recorrido permitido se encuentra una articulación. Según ésta sea rotacional o traslacional, el sensor deberá tener una estructura mecánica adaptada a la medición de ángulos o de distancias. Existen dos tipos fundamentales: eléctricos y ópticos. Entre los primeros destacan:

Potenciómetros: Consisten en un contacto que se mueve sobre un hilo de material resistivo (p. ej. constantán) arrollado en espiral. La resistencia es proporcional a la cantidad de hilo desde el inicio hasta la posición del contacto móvil. El esquema es:

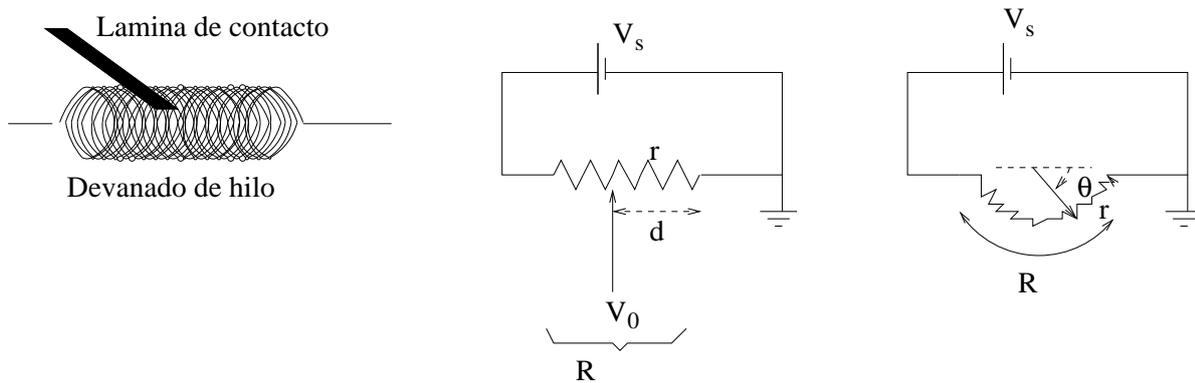


Figura 3.1: Esquemas del sensor potenciométrico

Como se ve, pueden construirse lineales, o circulares, según el tipo de articulación que los requiera. La tensión en el punto intermedio es

$$V_0 = \frac{r}{R} V_s$$

Por supuesto, debe procurarse que la construcción haga que r sea lineal con la distancia d (o con el ángulo θ). El porcentaje de no linealidad viene dado por $100 \frac{\epsilon}{V_s}$ donde ϵ es la máxima desviación de la linealidad.

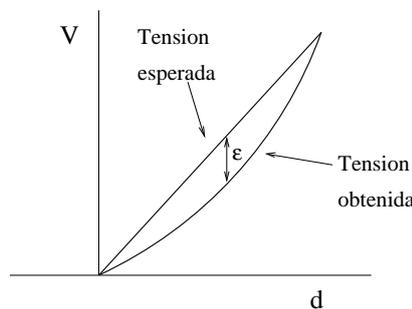


Figura 3.2: Tensión obtenida y esperada

Por otra parte, si la bobina es de hilo, la máxima resolución viene dada por la anchura de una espira, con lo que el porcentaje de resolución es

$$\%(res) = 100 \frac{V_s/N}{V_s} = \frac{100}{N}$$

La resolución no suele ser muy buena. Para evitarlo se pueden emplear potenciómetros de lámina resistiva, donde el hilo devanado se sustituye por una superficie resistiva. Tienen, además, menos ruido, (no hay cambio de espira a espira) pero la pletina roza lo cual provoca deterioro.

Existen también versiones bipolares de los potenciómetros, que dan señal tanto positiva como negativa, lo cual podría usarse como entrada a un sistema de control analógico, aunque hoy raramente se haría.

Sincros y resolvers: Son sensores de posición exclusivamente angulares, que no requieren contacto físico entre las piezas, por lo que se deterioran menos. Hoy día no se suelen emplear porque dan señal analógica, y además por su peso y coste. Se basan en un montaje similar a un transformador, en el que tres bobinas dispuestas en estrella rodean a un primario, que está mecánicamente fijo, pero al que se aplica una tensión variable como $V(t) = V_0 \sin(\omega t)$. El secundario gira alrededor de un eje perpendicular al plano de la figura. Con esta disposición, se puede probar que la tensión inducida en él tiene la forma $V(t) = V_0 * \cos \theta * \sin(\omega t)$ cuyo valor eficaz, extraído mediante un circuito al efecto, es proporcional al coseno del ángulo θ que el secundario está girado respecto al primario del transformador.

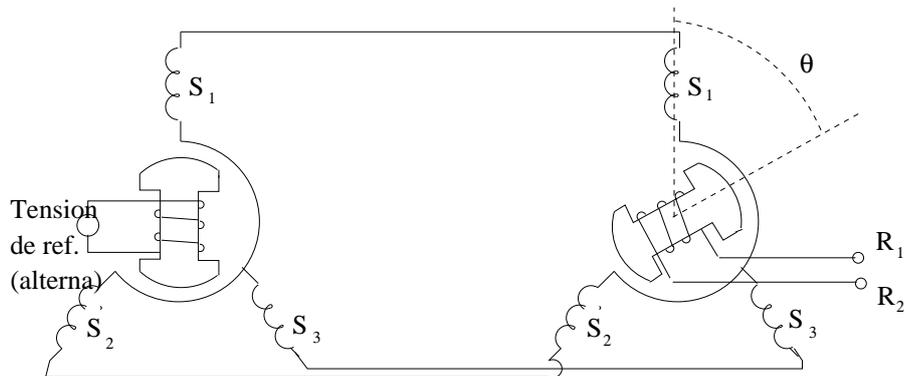


Figura 3.3: Esquema del sincro

Los resolvers son similares a los sincros, pero con las bobinas colocadas en posición diferente (separadas 120°)

El InductosynTM En su variante lineal, este sensor está formado por dos reglas que se desplazan una sobre otra, con circuitos (pistas) dibujados sobre cada una en forma de líneas almenadas, como se muestra en la figura.

En la regla fija las líneas ocupan toda su longitud. En la regla móvil, que se desplaza sobre ésta con sus líneas superpuestas, pero sin contacto eléctrico con las líneas de la regla fija, sólo hay dibujados dos grupos de almenas, desfasadas un cuarto de periodo una con respecto a otra (y por tanto, con respecto a las fijas). Al aplicar una tensión alterna a la línea fija de valor $V(t) = V \sin(\omega t)$ en los grupos de líneas de la regla móvil aparecen sendas tensiones inducidas de valor

$$V_1 = V \sin\left(\frac{2\pi X}{S}\right) \sin(\omega t) \text{ y}$$

$$V_2 = V \cos\left(\frac{2\pi X}{S}\right) \sin(\omega t)$$

donde X es el desplazamiento lineal de la regla móvil desde el inicio de su recorrido, y S la longitud de una onda (almena) de la escala. A partir del valor eficaz de estas dos tensiones es posible extraer el valor de X . La señal es similar a la de los sincros y resolvers, pero mucho más exacta, porque resulta de un promedio sobre muchas espiras (almenas). Existe una versión rotacional que alcanza precisiones de hasta 5 milésimas de grado.

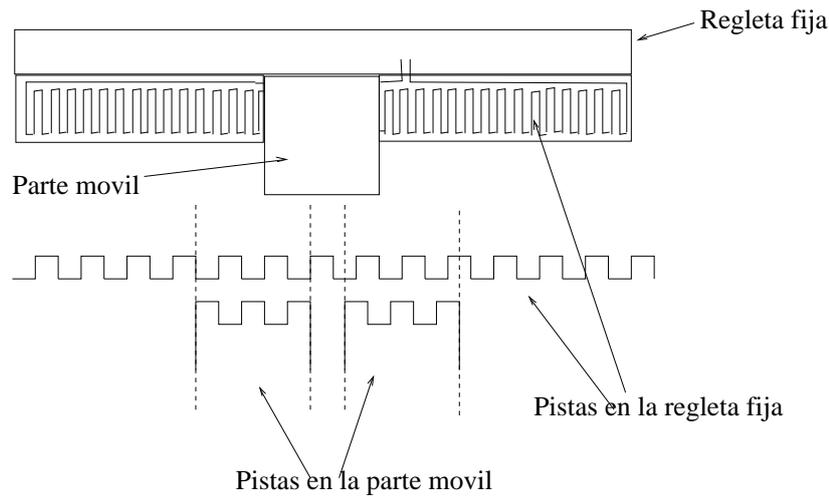


Figura 3.4: Esquema del InductosynTM

El siguiente apartado se refiere a sensores de posición de tipo óptico. Estos son especialmente importantes por ser los más usados, y entre ellos el ejemplo casi exclusivo son los codificadores (o encoders) ópticos de posición, que se basan en el principio del optointerruptor. Veámoslos en detalle.

Optointerruptores: Son interruptores de final de carrera (es decir, no detectan cuál es la posición de la articulación, sino sólo si ésta ha llegado o no a un punto determinado de su recorrido, usualmente el tope). No usan contactos mecánicos, sino un fotodiodo (o fotoreistencia) y un LED (diodo emisor de luz) que emite frente a él. Al moverse la articulación un disco o tope acoplado con ella (o, más usualmente, con el motor) interrumpe la luz del LED, dando en el fotodiodo un flanco negativo que es detectado por la circuitería apropiada. Hay versión tanto lineal como rotacional. En éste último caso tiene el problema de que no se conoce el sentido en que giraba el motor antes de llegar al punto de interrupción; esto puede ser un problema porque una articulación típica suele necesitar varias vueltas de motor para completar su recorrido. El disco puede tener una sólo marca, o varias, y en este caso un circuito apropiado detendría el motor a cada una de ellas, llevando el robot a través de una secuencia de movimientos con detenciones en puntos establecidos. "Programar" este tipo de robots significa cambiar los discos por otros con las muescas apropiadas.

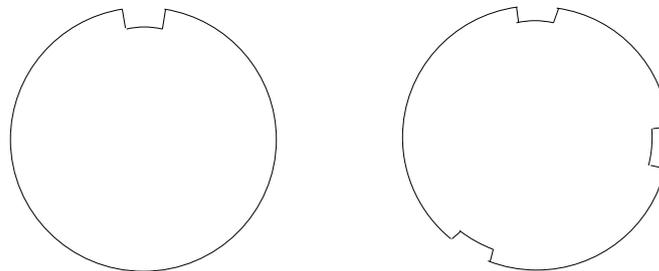


Figura 3.5: Discos con una o varias muescas

Codificadores ópticos: Se construyen como los microinterruptores, pero con numerosas muescas apropiadamente distribuidas. Hay dos tipos fundamentales:

+Absolutos: El disco que gira está impreso de tal modo que resulta opaco en ciertas áreas, y éstas están dispuestas como sectores, de modo que para cada sector radial la alternancia de zonas claras y oscuras corresponda a un código binario asignado de modo

único al sector. Cada uno de los "bits" de ese código es leído por un fotodiodo diferente que se encuentra cada vez más lejos del eje. La secuencia de asignación para sectores contiguos puede ser correlativa, o dar, p. ej., el número de sector en el código BCD.

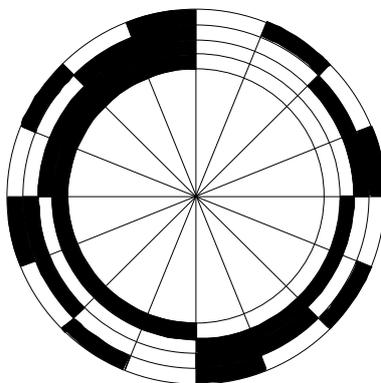


Figura 3.6: Disco de un codificador óptico absoluto de 16 sectores, 4 pistas

Tiene la ventaja de que "conserva" la lectura incluso sin corriente, de modo que al conectar el sensor se puede saber por lectura directa el sector en que se está. La resolución máxima en grados es, por supuesto, $360/N^{\circ} \text{ sectores}$, y el número de pistas (por tanto: de partes LED-fotodiodo) debe ser tal que $2^{N^{\circ} \text{ pistas}} = N^{\circ} \text{ sectores}$. Esto los hace caros y complejos, y por ello, no muy usados.

+Incrementales: De construcción similar a los anteriores, pero con sólo una pista que contiene muescas (o marcas opacas regulares) y dos pares LED-fotodiodo, colocados con una separación angular tal que las ondas cuadradas que cada uno genera cuando el eje se mueve estén desfasadas un cuarto de periodo. En realidad, la señal generada por el fotodiodo no es cuadrada, debido a que la transición no es lo abrupta que debiera, y a que puede recoger luz de estrías vecinas, si éstas son finas (de hecho, resoluciones comunes están en el rango de 200 a 1000 líneas/vuelta). Esto se soluciona pasando la señal por circuitos comparadores que generan un 1 lógico para tensiones mayores que un umbral, y 0 para las menores. Aparte, todo el dispositivo se encapsula herméticamente para evitar el polvo, suciedad e influencia de la luz ambiente. Las ondas generadas son como se muestra en la figura 3.7

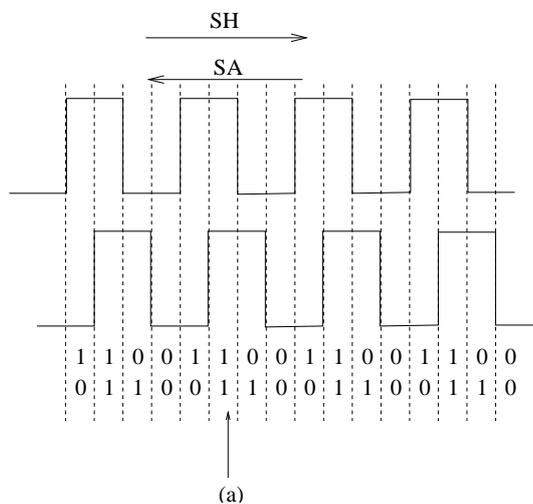
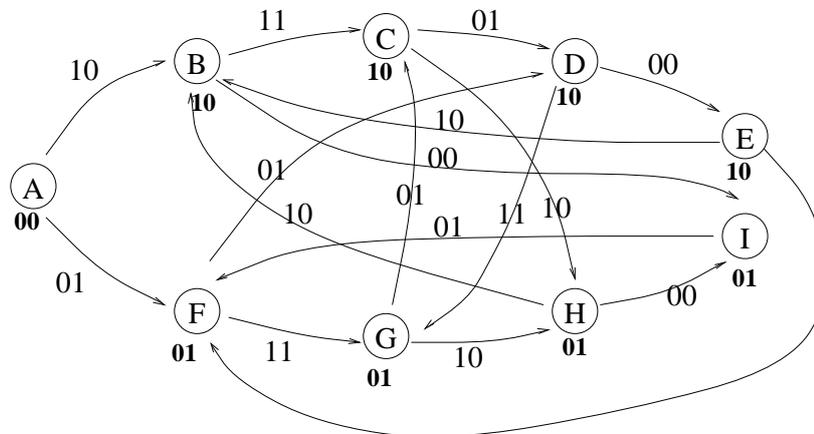


Figura 3.7: Ondas en los dos canales de un encoder incremental

Dadas estas señales, es imposible conocer la posición angular absoluta, pero el hecho de usar dos canales permite conocer el sentido de giro. Como ejemplo, supongamos que estamos en la situación (a) de la figura 3.7, leyendo por tanto $\{1,1\}$ en cada canal, y las entradas siguientes son respectivamente $\{1,0\}$. En ese caso, es claro de que estamos girando en sentido antihorario (SA). De otro modo, las siguientes entradas habrían sido $\{0,1\}$.

En principio, cada muesca genera una onda cuadrada completa, de modo que la resolución crece con el número de muescas. Pero es posible aumentarla electrónicamente, dándose cuenta de que en cada onda completa hay, contando los dos canales, 4 transiciones, de tal modo que contando transiciones la resolución queda multiplicada por 4. Un posible circuito apropiado para procesar esta señal sería un combinacional asíncrono con dos entradas (los dos canales) y dos salidas (U,D) que vayan a las entradas de un contador UP/DOWN, producido por el autómata de la figura 3.8 (sin simplificar):



Entradas sobre las flechas de transición: canal1/canal2

Salida (en negrilla) bajo los estados: (UP counter,DOWN counter)

Figura 3.8: Estados/salidas para el circuito de incremento de la resolución

Hay otras posibilidades, como el uso de un combinacional asíncrono implementado en ROM, o contadores en chip. Y aún más, es posible sustituir las señales cuadradas por señales sinusoidales o triangulares usando un disco de transmitancia óptica variable, lo cual genera una pareja de valores analógicos para cada posición concreta.

Codificadores ópticos incrementales como los descritos son el medio más habitual de conocer la posición de una articulación robótica. Pero como ésta debe ser conocida de modo absoluto, es preciso al conectar el robot proceder al proceso llamado calibrado, que consiste en poner cada articulación en su posición inicial, lo cual se hace mandando corriente a cada motor, hasta que se observe que no se mueve más (la lectura del codificador deja de cambiar); esto debe hacerse durante breve tiempo, para no quemar el motor. A continuación se sitúa el valor elegido como referencia en cada registro contador de pulsos.

Una forma posible de convertir un codificador incremental en absoluto es usar juntos el codificador y un potenciómetro, éste último unido a un engranaje desmultiplicador de modo que no de más de una vuelta en todo el recorrido. Así, el potenciómetro da la posición gruesa (con poca precisión) y el codificador la posición fina. La calibración se haría aquí de modo similar, pero sin necesidad de ir al tope, sólo hasta la siguiente salto discreto del potenciómetro.

Antes de concluir es necesario apuntar dos problemas que se presentan en el uso de codificadores ópticos:

±Oscilación: Cuando se levanta una carga a una cierta altura, el motor recibe corriente hasta que el codificador marque la posición deseada; pero al estar en ella, el sistema de control no debe mover más el motor, y por tanto no le envía corriente, con lo cual, por el propio peso de la carga, la articulación cae hasta la posición inferior del encoder (recuérdese que hay un error de $360/N$ grados) lo que da un señal de error de 1 pulso, que activa el motor de nuevo, y así sucesivamente. Esta oscilación se puede reducir aumentando la resolución del encoder, y se elimina si hay una cierta fricción, que amortigue o elimine las oscilaciones (sistema sobreamortiguado). La eliminación total sólo se conseguiría haciendo control analógico sobre la "raw-signal" (la señal original de los encoders, antes de pasar por los comparadores).

Otro problema está relacionado con la velocidad de cambio de la señal y el periodo de muestreo, y se estudiará en el tema 5.

3.2.2. Sensores de velocidad

Como su nombre indica, miden la velocidad (normalmente, angular, puesto que suelen ser rotacionales) a la que gira la articulación a que se conectan. Existen dos tipos: eléctricos y ópticos.

Eléctricos: Sólo veremos un caso: el tacómetro o dinamo tacométrica, que es un dispositivo similar a un motor, que genera una tensión alterna de amplitud proporcional a la velocidad angular de giro. Se diseñan de tal modo que esta amplitud sea lo más lineal posible con la vel. angular en el rango de uso. Suelen dar muy poca corriente (no es necesario que den más) dado que se usan pocas espiras en su bobinado, para que sean ligeros. Consisten en un devanado que gira perpendicularmente a un campo magnético creado (normalmente) por un imán permanente. Véase la figura

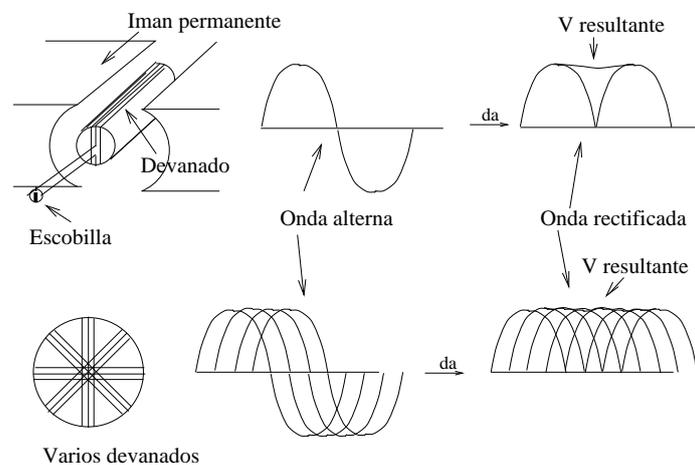


Figura 3.9: Esquema de la dinamo tacométrica

Recuérdese que la tensión que aparece en un conductor que se mueve dentro de un campo magnético es

$$V = -\frac{d\Phi}{dt} = -\frac{d(BS)}{dt} = -B\frac{dS_0 \cos \alpha}{dt} = BS_0 \sin \alpha \frac{d\alpha}{dt} = BS_0\omega \sin(\omega t)$$

Como se ve, esta es una tensión alterna cuya amplitud ($BS_0\omega$) es proporcional a ω . Las escobillas hacen que, al producirse un cambio de sentido de la corriente a cada media vuelta (semiperiodo) la tensión salga rectificadora, pero sigue sin ser continua. Para paliar esto se

pueden usar n devanados, arrollados en planos regularmente espaciados angularmente, y escobillas de $2n$ contactos. La onda resultante sigue sin ser estrictamente constante, aunque se parece más. La pequeña oscilación, reducible todavía más con un condensador, se llama rizado, y puede ir del 3 al 5 % del valor medio.

Otra alternativa es usar una armadura móvil (hacer girar el imán), y poner en el centro, que ya no gira, más bobinas (de 19 a 23); además, se puede eliminar el núcleo de hierro sobre el que iban bobinadas. Estos tacómetros son más caros, pero dan un rizado menor (del orden del 1 %).

+Ópticos: Tratan de usar los codificadores ópticos antes explicados para medir la velocidad, realizando una aproximación discreta, pues como se vio, estos sensores generaban señal digital. Hay dos métodos para obtener el valor de la velocidad. Estos son:

- La conversión frecuencia-voltaje: consiste en usar circuitos que realizan una aproximación discreta a la derivada en la forma

$$v = \frac{dx}{dt} \simeq \frac{\Delta x}{\Delta t}$$

siendo Δx el número de pulsos contados en el tiempo Δt . Constan de un contador, un reloj interno y normalmente un conversor D/A que convierte el número de cuentas a voltaje. Esto último sólo es necesario si estamos haciendo un control analógico en velocidad del motor. Estos dispositivos tienen dos problemas: un pequeño retraso (que habría que modelizar en continuo como un retraso de transporte, $e^{\Delta ts}$) y el hecho de que el conversor D/A genera una señal continua a trozos (aunque esto siempre es mejor que el rizado de los tacómetros).
- El otro método es efectuar la aproximación por software mediante un programa que implementase un algoritmo similar al siguiente:

Hacer

```

posición_presente ← P(kT)
v(k) ← (posición_presente - posición_anterior)/T
posición_anterior ← posición_presente
(ejecución del algoritmo de control)
k ← k + 1
espera(T - δt)

```

hasta (desconexión del dispositivo)

donde T es el periodo de muestreo, la función P lee el hardware para dar la lectura del encoder en ese momento, v es la velocidad en el instante discreto k (continuo: kT) y δt es el tiempo que cuesta la ejecución del cálculo en cada pasada del bucle. Por supuesto, debe ser menor que T , en caso contrario, habría que buscar un ordenador más rápido. El cálculo de $v(k)$ puede realizarse con un algoritmo más complejo que tome en cuenta el valor de la posición en varios instantes anteriores (recuérdense los bloqueadores de orden superior).

La elección del periodo de muestreo T es importante. Si se toma demasiado grande, podría provocar inestabilidad, porque la lectura de los encoders habría cambiado mucho entre cada dos periodos, y el comportamiento del sistema durante ese tiempo sería incontrolable. Pero si escoge demasiado pequeño también tendremos problemas debidos al carácter cuantizado de la señal generada por los encoders. Se incidirá en este punto en el tema 5

3.2.3. Acelerómetros

Miden la aceleración del dispositivo al que van físicamente unidos, y se basan en la ley de Newton. Al mover el cuerpo con cierta aceleración a aparece sobre él una fuerza de inercia, $F = ma$, que puede ser medida con un resorte, usando la ley de Hooke, $F = kx$,

siendo x el alargamiento del resorte y k su constante elástica. Para medir el alargamiento se puede a su vez usar un encoder lineal.

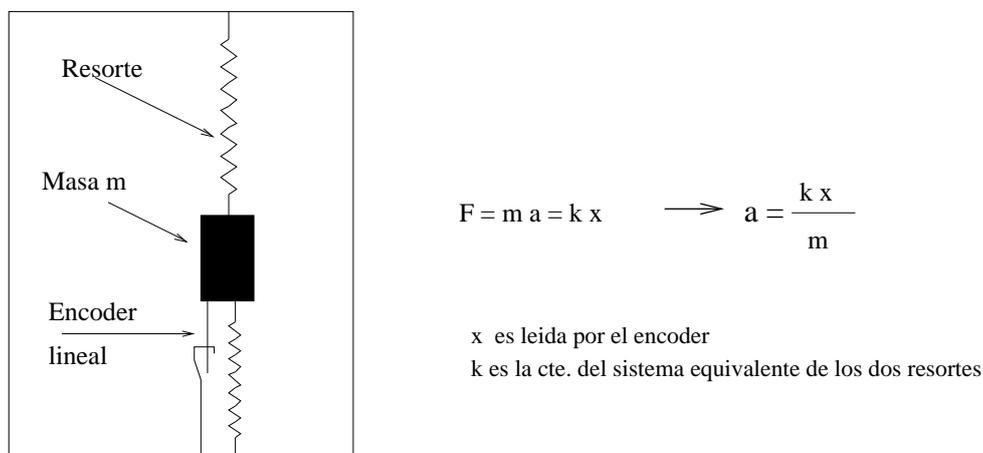


Figura 3.10: Esquema de un acelerómetro inercial

Este tipo de sensores se usa poco; generalmente, suelen ser para monitorizar problemas debidos a la falta de rigidez de los brazos. Deben tener cierto amortiguamiento para llegar pronto a la posición de equilibrio y no oscilar.

Recientemente han aparecido también acelerómetros de estado sólido, que constan de una base de silicio con estrías y un material conductor sobre él formando un condensador cuya capacidad varía de acuerdo a la distancia entre las placas, que cambia ligeramente cuando éstas se curvan debido a la acción de la fuerza de inercia.

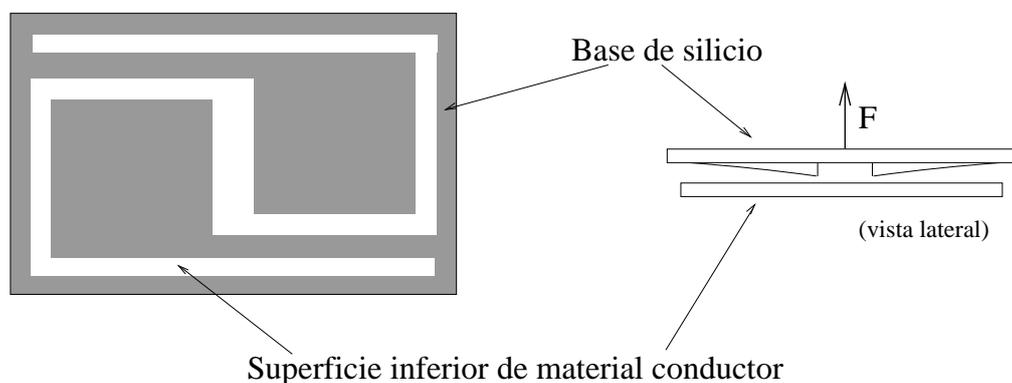


Figura 3.11: Acelerómetro de estado sólido

3.3. Sensores externos

Como se dijo, estos sensores dan información acerca de sucesos y estado del mundo que rodea al robot, es decir, monitorizan dinámicamente la relación de un robot con su entorno, y el desarrollo de la ejecución de una tarea. Idealmente, deben alterar lo menos posible el entorno que monitoricen.

3.3.1. Sensores de proximidad

Señalan la distancia entre el punto terminal (u otro punto) del robot, y otros objetos. Pueden ser de contacto, o sin contacto físico.

+De contacto: Son simples microinterruptores colocados en cabeza del brazo, o en algún punto que se piense que puede chocar. Detienen o hacen retroceder el elemento cuando se activan. Pueden usarse para controlar cuándo una articulación llega a su límite, o a una posición dada. En este caso se llaman de fin de carrera. Otro tipo de sensores con contacto físico son codificadores lineales acoplados a un vástago que se desliza sobre la superficie del objeto; si lo hace a velocidad constante, sirve para conocer el perfil del objeto por lectura sucesiva de su valor.

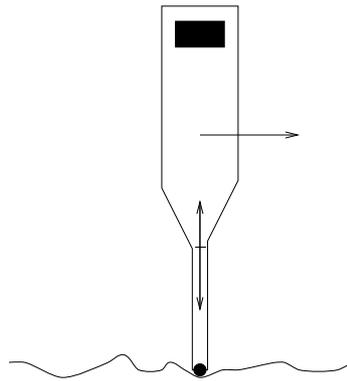


Figura 3.12: Sensor de contacto

+Sin contacto: Hay tres tipos, cada uno con varios ejemplos:

- De reflexión luminosa: constan de una fuente de luz, una lente para focalizar la luz aproximadamente sobre el objeto, otra lente para concentrar la luz reflejada, y una fotoreistencia, que medirá la intensidad de luz recibida. Esta está relacionada con la distancia al objeto, pero también con la intensidad a través de la fotoreistencia (a tensión constante), lo cual relaciona indirectamente a ésta con la distancia según una gráfica como la que se muestra en la figura 3.13.

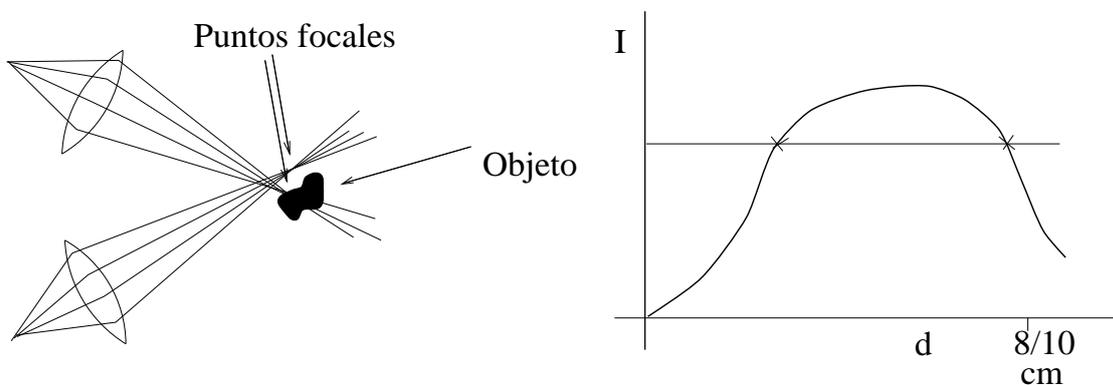


Figura 3.13: Sensor de reflexión luminosa

Aparecen tres problemas con este dispositivo: el primero es que hay dos puntos a diferente distancia que dan la misma intensidad (los que se encuentran a un lado y al otro del punto focal de la lente). Esto se resuelve usando otro sensor, o viendo si la señal crece o

decrece al avanzar. El segundo problema es que son sensibles a las variaciones de la luz ambiente y de la temperatura. Esto se resuelve mandando la luz no de modo continuo, sino pulsante, a una frecuencia de unos 6KHz. En este caso de la señal recibida se puede filtrar la componente interesante y ver su amplitud (véase la figura).

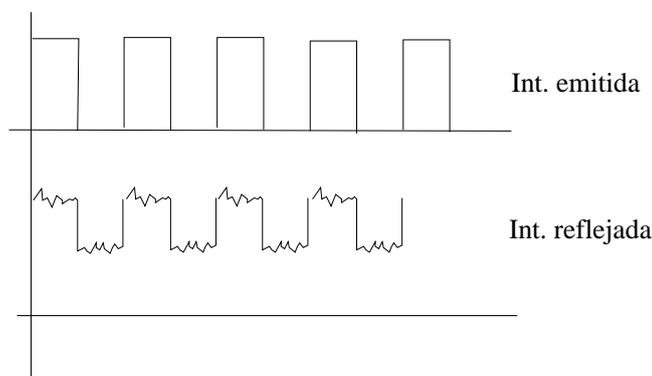


Figura 3.14: Intensidad emitida y reflejada en modo pulsante

El último problema es que la intensidad reflejada depende de la naturaleza del material. Esto no se puede evitar del todo; se intenta paliar usando luz infrarroja. Esto hace que se usen más como aviso (de un modo similar a un microinterruptor, pero sin contacto) que como medidores de distancia absoluta.

- De fibra óptica: La fibra óptica es fibra de vidrio que conduce la luz basándose en el fenómeno de la refracción y el ángulo límite. Cuando la luz incide en la superficie de separación de dos medios viniendo desde el que tiene mayor índice de refracción hacia el que lo tiene menor, si incide con un ángulo más pequeño que cierto valor (el llamado ángulo límite) pasa al segundo medio, refractándose. Pero si lo hace con ángulo mayor, se refleja de nuevo hacia el interior del primer medio. Esto se observa en la parte superior izquierda de la figura 3.15, donde el rayo 1 incide con ángulo menor que el ángulo límite, el 2 exactamente con el ángulo límite, y el 3, con un ángulo mayor. Un conjunto de reflexiones sucesivas pueden "conducir" un rayo de luz por el interior de un tubo de vidrio de geometría apropiada. (figura 3.15, parte superior derecha).

A partir de este principio se pueden construir sensores de distancia (o de presencia de objeto) de tres tipos: de corte del haz, en el que el objeto intersecta el haz entre dos cabos de la fibra óptica, si está allí; de retroreflector, en el que el mismo cabo de fibra óptica emite y recibe el rayo de luz reflejado por un catadióptrico, y de reflexión difusa, igual que el anterior, pero en el que la reflexión la realiza la propia superficie del objeto. Todos ellos son sensibles a los mismos problemas que el tipo anterior, y por ello también se usan más como detectores de presencia que para medir distancias.

- Sensores laser: Se basan en dos espejos perpendiculares acoplados a motores eléctricos que permiten deflejar un laser de modo que apunte en cualquier dirección deseada del espacio. Para usarlos hay que mover el laser barriendo la superficie con velocidad angular constante. Además, se sitúa un dispositivo colimador apuntando en una dirección conocida, y se observa cuándo el punto brillante que el laser marca en la superficie del objeto se observa precisamente en esa dirección. De acuerdo al tiempo que el punto laser ha tardado en pasar por ella se determina la distancia de la superficie al colimador. Véase la figura, donde la misma superficie S está puesta más adelante (SA) o más atrás (SB). Si el rayo se desplaza sobre el plano del papel en la dirección de la flecha, el colimador lo verá antes si incide en A (lo que indicará que la superficie era la SA) que si incide en B (la superficie era

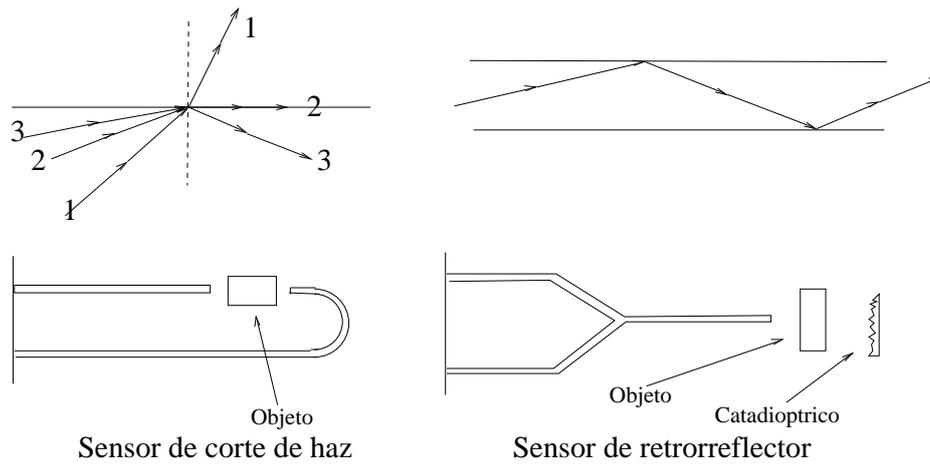


Figura 3.15: Reflexión total y sensores con fibra óptica

SB), dado que el ángulo α_A es menor que α_B , y el rayo se desplazará antes con este ángulo.

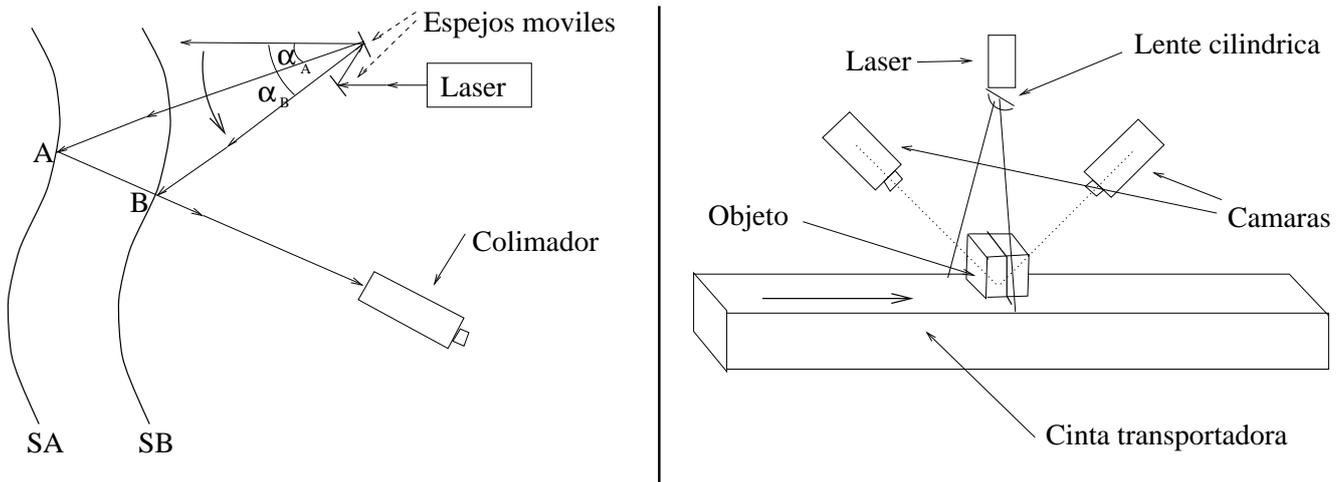


Figura 3.16: Dos tipos de sensores laser de distancia

Otro montaje alternativo usa una lente cilíndrica, que abre el rayo dando una "cortina" lámina de luz, que incide perpendicularmente a una cinta transportadora que desplaza al objeto sobre ella. Dos cámaras colocadas con sus ejes ópticos apuntando al punto central y formando ángulos de 45° con el plano horizontal recogen imágenes donde aparecen líneas brillantes sobre el perfil del objeto, a distintas posiciones dependiendo de su altura.

+Sensores de ultrasonidos: Estos son uno de los tipos más usados de sensores de distancia sin contacto físico. Se basan en emitir pulsos de ultrasonidos, y medir el tiempo de vuelo entre la emisión y la recepción, conociendo la velocidad del sonido (340 m/s, en aire seco a $20^\circ C$, y varía con la temperatura). La frecuencia de emisión es fija, normalmente 40 KHz.

Se suelen emitir pulsos de aproximadamente 1 ms. (40 ondas completas). El receptor tiene un filtro pasa-banda fino sintonizado a los 40KHz, o bien es un dispositivo físico, cristal u otro, que oscila sólo a esa frecuencia. Entre los más populares se encuentran los sensores PolaroidTM para cámara fotográfica. Emiten pulsos a varias frecuencias para evitar que alguna frecuencia desaparezca debido a la forma o características de reflexión del objeto.

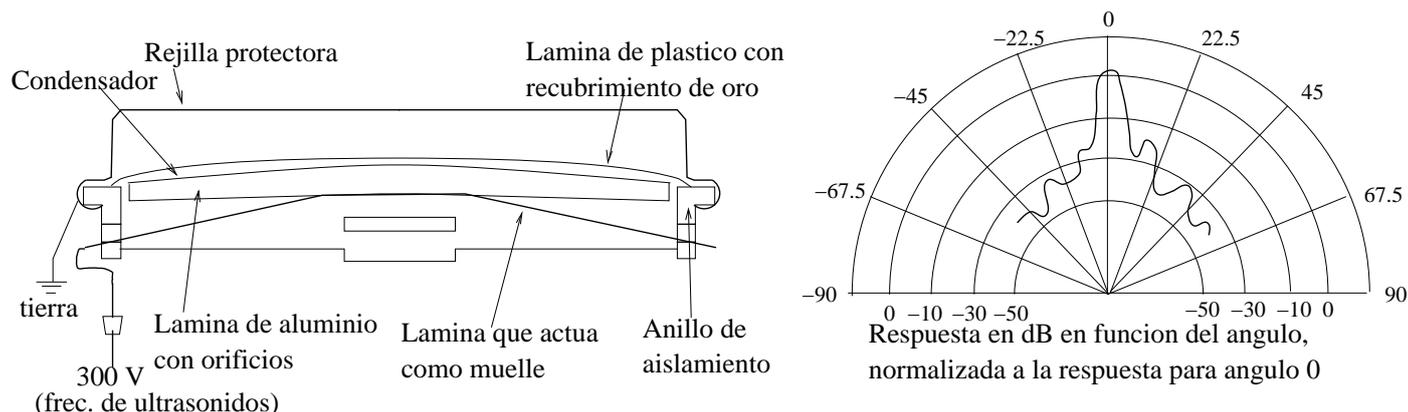


Figura 3.17: Esquema del sensor Polaroid TM

Se usan normalmente varios de éstos, orientándolos en diferentes direcciones. La precisión puede llegar a ser de unos 0.5 cm en 2 m., pero esto es en condiciones óptimas. En general, tienen mucho ruido y se ven sometidos a reflexiones espurias.

Una forma alternativa de medir la distancia usando sensores de ultrasonidos es la medición de la amplitud de la onda reflejada. Se observa una atenuación aproximadamente cuadrática de ésta con la distancia al objeto, pero en general también son poco precisos y dependen del material que refleja.

+Sensores de corriente inducida: Se basan en usar una bobina por la que circula una corriente alterna que genera un campo magnético variable. Cuando esta bobina se acerca a un objeto de material ferromagnético (Fe, acero o Al) se generan en él corrientes parásitas, las cuales a su vez generan otro campo que tiende a anular al primero, con lo que la intensidad que circula por el solenoide varía, siempre que el voltaje se mantenga constante. Esta variación no es lineal con la distancia, y depende de la forma del objeto, del material, y del ángulo de aproximación del sensor; por ello, es necesario un calibrado para cada uso concreto. No obstante, son robustos y apropiados para ser usados en ambientes hostiles (polvo, grasa, etc.).

+Sensores resistivos: Se usan en aplicaciones de soldadura por arco voltaico, donde hay que mantener constante la altura sobre el material (normalmente, dos planchas metálicas) que está siendo unido. Se basan en el hecho de que la resistencia del arco voltaico (más exactamente: la intensidad que circula a voltaje constante) es proporcional a la longitud del arco, que es precisamente la distancia entre el electrodo colocado en la punta del brazo y la superficie. Esta intensidad está entre 100 y 200 amperios.

+Otros tipos de sensores de distancia sin contacto: Se mencionarán simplemente las sondas capacitivas, que usan el objeto a detectar (que debe ser conductor) como una de las placas de un supuesto condensador, y la punta del brazo, o una pieza metálica adosada a ella, como la otra placa. La capacidad varía en función de la distancia. Si las placas fueran planas y paralelas, lo cual es sólo una aproximación, la capacidad sería $C = \epsilon \frac{S}{d}$, siendo ϵ la cte. dieléctrica del aire, S la superficie de las placas, y d la distancia entre ellas.

Otro sensor interesante de este tipo es el de efecto Hall. Se basan en que algunos materiales semiconductores varían su conductividad en cierta dirección cuando están sometidos a la acción de un campo magnético. Por ello, es necesario colocar un imán, aun pequeño, fijo al objeto cuya distancia (o, más habitualmente, presencia o ausencia) queremos detectar.

3.3.2. Sensores de tacto

No siempre es posible usarlos, pero cuando lo es son muy útiles; van desde los que sólo dan señal ON/OFF en puntos seleccionados, hasta los que dan una medida de la presión en cada punto. Por ahora, la mayoría son experimentales. Entre ellos cabe citar:

+De varillas: Son simplemente una matriz de varillas que se coloca horizontalmente y descende hasta hacer contacto con el objeto. Si es de tipo ON/OFF, hay que bajarlo hasta que todos los sensores se activan, y entonces ir subiendo lentamente y tomar nota del instante en que cada uno se desactiva. Otro tipo más evolucionado consiste en varillas de material ferromagnético que se introducen más o menos en bobinas, variando su inductancia, que se mide, y resulta ser proporcional a (o relacionada con) la longitud de varilla que quede dentro de la bobina.

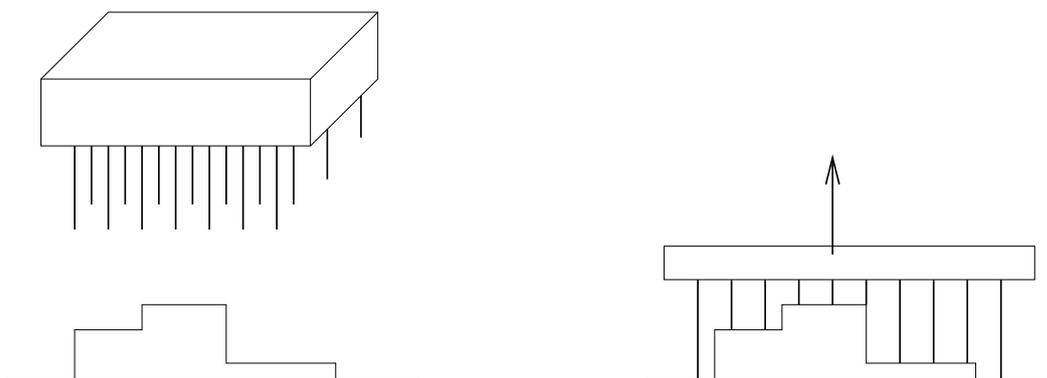


Figura 3.18: Sensor táctil de varillas

+De fotodetectores: Se basan en un principio idéntico a los anteriores, pero aquí la varilla corta el rayo de luz que va de un LED a un fotodiodo. Todas las varillas están cubiertas por una capa elástica. Son siempre de tipo ON/OFF, y presentan los problemas de que la capa elástica tiene cierta histéresis, y además se desgasta y hay que reemplazarla.

+De elastómeros de conductividad: Se basan en que ciertos materiales elásticos (algunos plásticos, normalmente) pueden hacerse más o menos conductores impregnándolos con polvo de hierro o similar. De este modo pueden poner en contacto dos electrodos con una resistencia mayor o menor, según sea la superficie de contacto. Tienen un problema esencial: el elastómero tiene una vida limitada; tras unos cientos de operaciones, su resistencia no vuelve al valor original después de decomprimir, o, en otros tipos, lo hacen al cabo de un tiempo excesivo.

+De presión neumática: Se basan en unos contactos regularmente distribuidos, y una lámina metálica que se sitúa sobre ellos, pero no los toca, porque entre ambos queda unas cavidades que se llenan con aire comprimido. Sólo cuando se presiona por la parte exterior la lámina vence la presión del aire y toca el contacto.

Estrictamente, son binarios, pero cambiando la presión del aire que circula pueden dar una idea de la fuerza con que se presiona: oprimiendo exactamente hasta que haga contacto, la fuerza es entonces justo la necesaria para vencer la presión del aire. Se usó para operaciones de inserción; sólo daba lecturas de presión de 0 a 50 gramos.

+De polímeros (piel artificial): Se basan en que ciertos materiales, como el cuarzo o algunos polímeros sintéticos, presentan el fenómeno conocido como piezoelectricidad, que consiste en que generan una pequeña corriente eléctrica cuando se les presiona mecánicamente. Los sensores de polímeros se construyen con una capa protectora, una del material

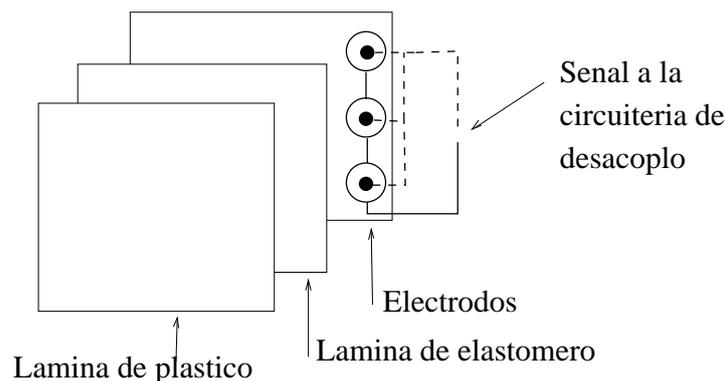


Figura 3.19: Sensor de elastómeros

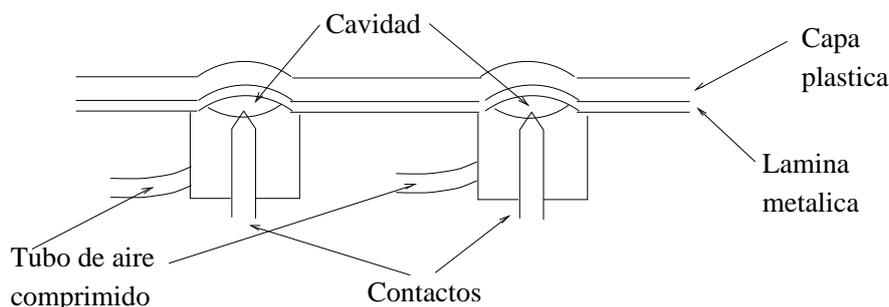


Figura 3.20: Sensor de presión neumática

piezoeléctrico (fluoruro de polivinilo, PVDF) y otra con electrodos en filas y columnas que se sitúa debajo. Al presionar en cierta área, el PVDF genera una carga local que es recogida por los electrodos; este fenómeno dura unos pocos segundos, lo cual es suficiente para medir el máximo de carga generada, que está relacionada con la presión. Este sensor puede usarse en modo binario, o analógico. Tiene el problema del acoplo entre unos circuitos y otros, correspondientes a electrodos vecinos, que se debe solucionar con circuitos de desacoplo de bajísima capacidad (menor de 5 pF). El PVDF es también piroeléctrico (genera carga al variar la temperatura), lo cual puede ser ventajoso en ciertas aplicaciones, pero en general es un inconveniente, ya que obliga a corregir las medidas con la temperatura ambiente.

3.3.3. Sensores de fuerza

Son necesarios para ajustar correctamente la presión que ejercen los motores de la pinza de un brazo robot en operaciones de presión, particularmente, en ensamblado de piezas, para así estar seguros de no romper éstas. También son prácticamente imprescindibles en operaciones de inserción en las que otros sensores no pueden actuar, por no tener acceso al lugar físico de la inserción. Es importante medir tanto la fuerza ejercida, como el momento o torque respecto a algún punto, normalmente el eje de rotación. Existen dos casos importantes de sensores de fuerza:

+Por variación de la corriente del motor: Como se verá en el tema siguiente, para servomotores eléctricos de corriente continua el momento o torque T ejercido por el motor es directamente proporcional a la intensidad que circula por su devanado (corriente de armadura, I_a). Si $T = K_T I_a$, entonces midiendo I_a con un amperímetro podremos conocer el momento ejercido por el motor. La transformación de esto en fuerza depende del dispositivo de conversión de movimiento rotacional-lineal que se use, p. ej., si se usa un piñón de radio

R conectado a un tornillo sin fin, entonces

$$T = \frac{RF}{\eta} \implies F = \frac{KI_a\eta}{R}$$

siendo η la eficiencia de la transmisión, una constante del orden de 0,9. De este modo pueden darse valores máximos y mínimos de I_a para cada pieza que haya que ensamblar. Nótese que esto no mide realmente la fuerza, sino una magnitud relacionada, I_a , y que por tanto si la relación entre ambas varía (p.ej., por variar K_T o η) podría ejercerse con la misma corriente una fuerza mayor o menor, que podría respectivamente romper o dejar caer el objeto.

+Por deflexión de los dedos (galgas extensiométricas): Se basan en la variación de resistencia eléctrica de cualquier material en función de su longitud y sección. En particular, sabemos que para una amplia gama de materiales, su resistencia es

$$R = \rho \frac{L}{S}$$

siendo L la longitud y S la sección de la porción de material. Dada una fina lámina de material fijada sobre una base flexible, si se dobla como se muestra en la figura, la variación de sección es despreciable; sin embargo, la variación de longitud es significativa, y hace cambiar R , que se puede medir. Normalmente se usan dos láminas unidas de diferentes materiales, como platino-tungsteno o cobre-aluminio, lo cual se llama galga extensiométrica, o bien con un semiconductor sobre una base de silicio. Las dos láminas forman las resistencias opuestas de un puente de Weathstone, y la variación de tensión se mide con un amplificador operacional.

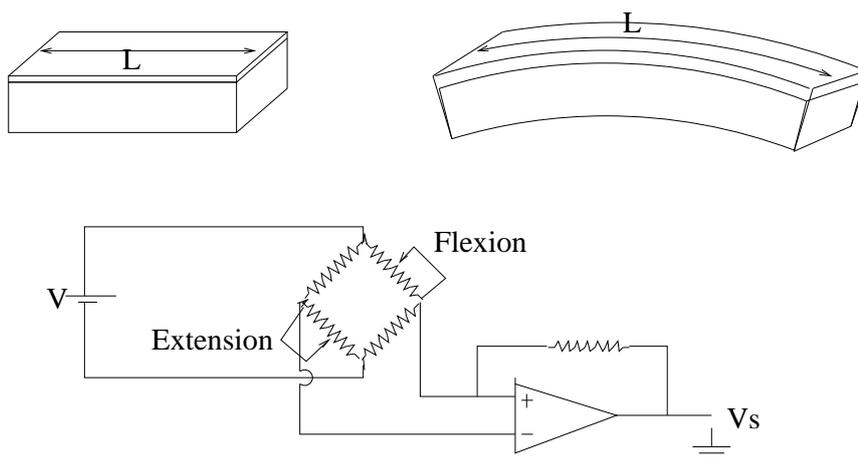


Figura 3.21: Galga extensiométrica

Pueden usarse simplemente unidas a cada una de las caras de la pinza de agarre, pero la configuración más inteligente, que permite medir fuerza y momento, es la disposición en los brazos de una cruz metálica como se muestra en la figura 3.22

En este caso, la fuerza $\vec{F} = (F_x, F_y, F_z)$ y el momento $\vec{M} = (M_x, M_y, M_z)$ ejercidos

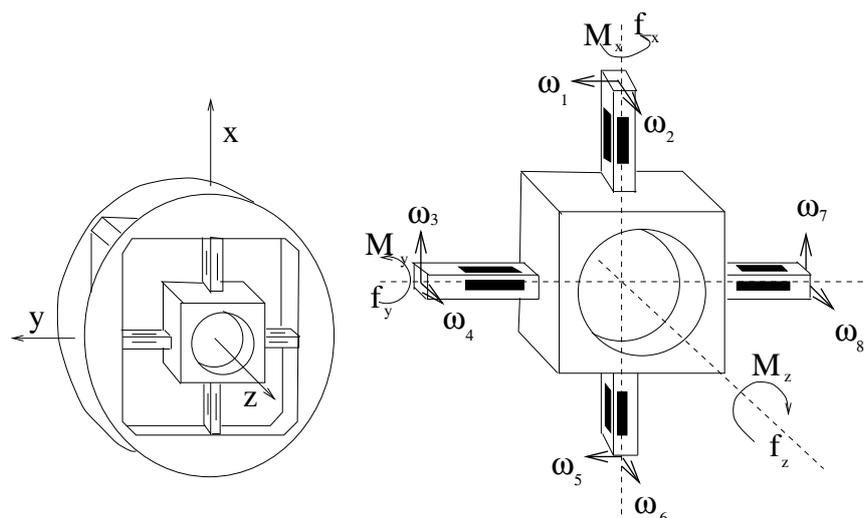


Figura 3.22: Sensor de fuerza-momento

pueden relacionarse con las lecturas de cada sensor, (w_1, \dots, w_8) como

$$\begin{pmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{pmatrix} = \begin{pmatrix} 0 & 0 & K_{13} & 0 & 0 & 0 & K_{17} & 0 \\ K_{21} & 0 & 0 & 0 & K_{25} & 0 & 0 & 0 \\ 0 & K_{32} & 0 & K_{34} & 0 & K_{36} & 0 & K_{38} \\ 0 & 0 & 0 & K_{44} & 0 & 0 & 0 & K_{48} \\ 0 & K_{52} & 0 & 0 & 0 & K_{56} & 0 & 0 \\ K_{61} & 0 & K_{63} & 0 & K_{65} & 0 & K_{67} & 0 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \end{pmatrix}$$

3.3.4. Sensores de visión

Como su nombre indica, envían al software de control y programación del robot una imagen de la escena o área de trabajo, que programas adecuados deben encargarse de interpretar para extraer la información útil sobre posiciones y orientaciones de los objetos presentes (o simplemente, su presencia o ausencia). Este objetivo entra dentro del campo de la visión por computador, y por tanto no se tratará aquí. Por otra parte, los dispositivos de captura de la imagen son cámaras de televisión, bien de tubo de rayos catódicos, bien de tipo CCD, las más usadas hoy día en aplicaciones robóticas. Su tecnología tampoco se cubrirá aquí, puesto que ya fue expuesta en el módulo "Sistemas de percepción", o en los módulos apropiados de Ingeniería electrónica.

Tema 4: Tecnología de actuadores robóticos

Actuador es todo dispositivo que ejerce fuerzas o momentos sobre las partes de un robot haciendo que éstas se muevan. Transforman algún tipo de energía en energía mecánica, y para que sean útiles en Robótica deben poder ser controlados con rapidez y precisión. Las tecnologías fundamentales que se usan hoy en robots son hidráulica, neumática y eléctrica. Estas emplean respectivamente un fluido en circulación, aire comprimido y electricidad. Los motores de combustión interna o de vapor no se usan por la dificultad de controlarlos con precisión y su largo tiempo de respuesta.

4.1. Actuadores hidráulicos

Se usan para levantar cargas mayores de 6 o 7 Kg., o para potencias aproximadas de 5 a 7 HP. El fluido que transmite la potencia, normalmente aceite especial, circula por tuberías a presión de unas 200 atmósferas y un caudal de unos 0.25 l/seg. Ejercen presiones aplicando el principio de la prensa hidráulica de Pascal para aumentar la fuerza al disminuir la superficie sobre la que se aplica, y para su control se usan las llamadas servoválvulas, que son dispositivos que controlan el flujo de fluido que las atraviesa de acuerdo a la corriente eléctrica que se les suministra. El flujo, mayor o menor, que aparece tras la servoválvula hace que un cilindro o pistón se mueva, provocando desplazamiento lineal, que puede ser convertido en rotacional mediante un sistema biela/manivela, como se verá al final de este tema.

Una servoválvula es esencialmente un motor eléctrico de baja velocidad y alto torque, que no gira vueltas enteras, sino fracciones de vuelta en contra de una resistencia mecánica; este motor tira de un tubo flexible que sujeta una pieza que hace que el flujo de entrada se reparta desigualmente entre cada uno de los tubos de salida, modificando así el flujo que sale por éstos. En la figura 4.1, parte (b), se observa la pieza centrada, lo que da un caudal igual por ambas salidas, y en (c) descentrada, lo que da mayor caudal por la salida derecha que por la izquierda.

El dispositivo completo se puede ver en la misma figura, parte (a). Su objetivo final es conseguir que la presión en las salidas C_1 y C_2 , que forman otro circuito hidráulico con el canal central como drenaje sea bien igual, o bien mayor en uno que en el otro. Ello es porque estos dos canales se conectan al actuador final, según se muestra en la figura 4.2, el cual mueve la masa M . La velocidad con que se mueve es aproximadamente proporcional a la diferencia de caudales. Se pueden obtener fuerzas mayores con la misma señal para la electroválvula simplemente aumentando el caudal que circula por el segundo circuito hidráulico.

Para cerrar el lazo de realimentación, la posición de la masa se puede medir con un encoder lineal cuya señal servirá al sistema de control para generar la señal de control de la electroválvula. Si hay una fricción externa o resistencia al movimiento de la masa, F_d , habrá al final un error de posición que se puede solucionar mediante el control integral apropiado. El esquema del sistema completo puede verse en la figura 4.3.

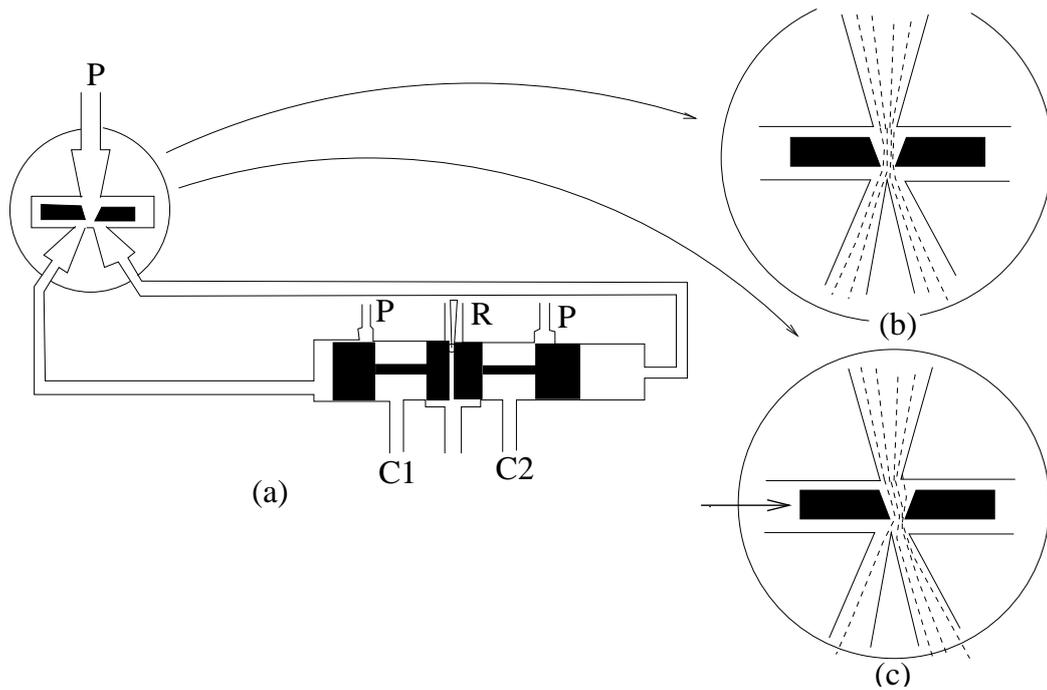


Figura 4.1: Esquema de la servoválvula hidráulica

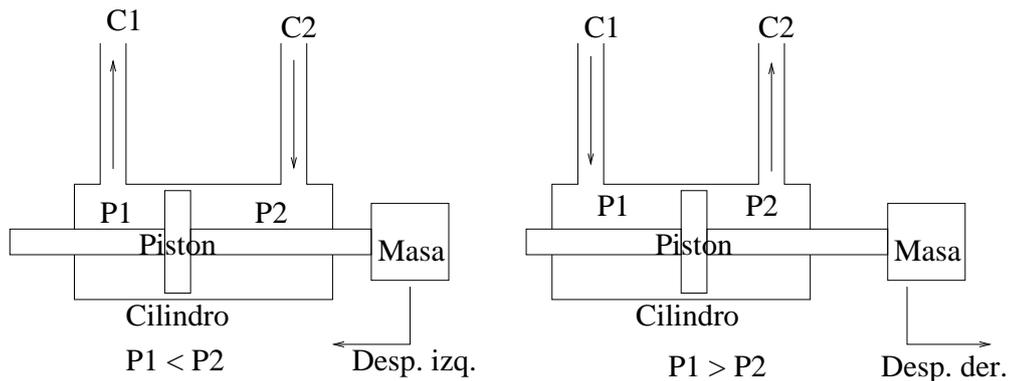


Figura 4.2: Cilindro y pistón hidráulicos lineales

4.2. Actuadores neumáticos

Su principio de funcionamiento es similar al de los actuadores hidráulicos, pero a diferencia de aquellos, que empleaban un fluido incompresible, éstos emplean aire, altamente compresible. El no llevar fluidos potencialmente inflamables los hace más seguros, y además no hay que reemplazar periódicamente el fluido. Pero al ser el aire tan compresible suelen ser subamortiguados, lo cual es malo, y una vez alcanzada su posición final presentan poca rigidez. Se suelen usar para mover pistones lineales punto a punto, usando topes, pero existe la posibilidad de usar control neumático. Una especie de controlador proporcional sería el mostrado en la figura 4.4

Se supone que P_1 es la presión producida por un compresor que es aproximadamente (aunque no exactamente) constante. Se trata de que P_0 sí lo sea, independientemente de variaciones en la resistencia mecánica o masa de las piezas que el pistón conectado a P_0 mueva. Este movimiento deberá ser transmitido al dispositivo a través de la palanca e. El funcionamiento del controlador es como sigue: si la barra e se mueve hacia la derecha,

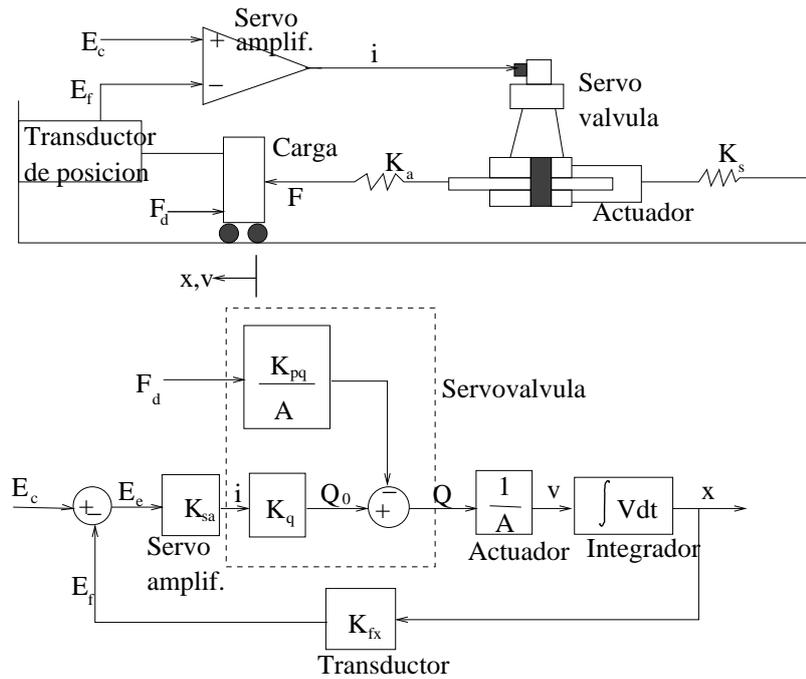


Figura 4.3: Control de un sistema hidráulico

la espita x se abre con lo cual P_2 decrece. Esto hace que el sensor A_2 , por la acción de los muelles inferiores, se mueva hacia arriba. Con ello la pieza que bloquea la salida a la atmósfera sube, y esto hace que P_0 aumente (es decir, se aproxime más a P_1). Esto a su vez provoca que la válvula A_f (sensor) se mueva hacia la izquierda, tendiendo a cerrar de nuevo la espita x . Cambiando las constantes elásticas de los muelles de los sensores se consigue equilibrar el sistema para mantener una cierta presión de salida deseada, P_0 , constante.

Recíprocamente, si e se mueve hacia la izquierda, la espita x se cierra, P_2 aumenta, A_2 baja y deja escapar más aire a la atmósfera, con lo que P_0 disminuye, alejándose de P_1 .

Finalmente, P_0 se conecta a un dispositivo de émbolo-pistón semejante al visto en la figura 4.2 para actuadores hidráulicos.

Para concluir, comentar que existen actualmente combinaciones de sistemas neumáticos y eléctricos; la parte neumática genera el movimiento grueso con bastante fuerza, mientras que la parte eléctrica controla el posicionado fino. Un uso muy apropiado de éstos es en plataformas para simulación, en la configuración llamada plataforma Stewart, con 3 o 6 actuadores neumáticos lineales unidos a los vértices de un triángulo equilátero sobre el que se fija una plancha, y sobre ella la cabina del simulador propiamente dicho.

4.3. Actuadores eléctricos

Son, con gran diferencia, los más usados actualmente en robots comerciales y experimentales. Los habituales son los motores de corriente continua (CC) y, en menor medida, los motores paso a paso. Los de corriente alterna se usan raramente por no poderlos controlar con precisión, y por depender su velocidad de giro de la frecuencia de la corriente alterna que los alimenta, la cual no se puede variar más que con dispositivos electrónicos caros y no extraordinariamente precisos. Es realmente la facilidad de control lo que hace de los motores CC los más usados.

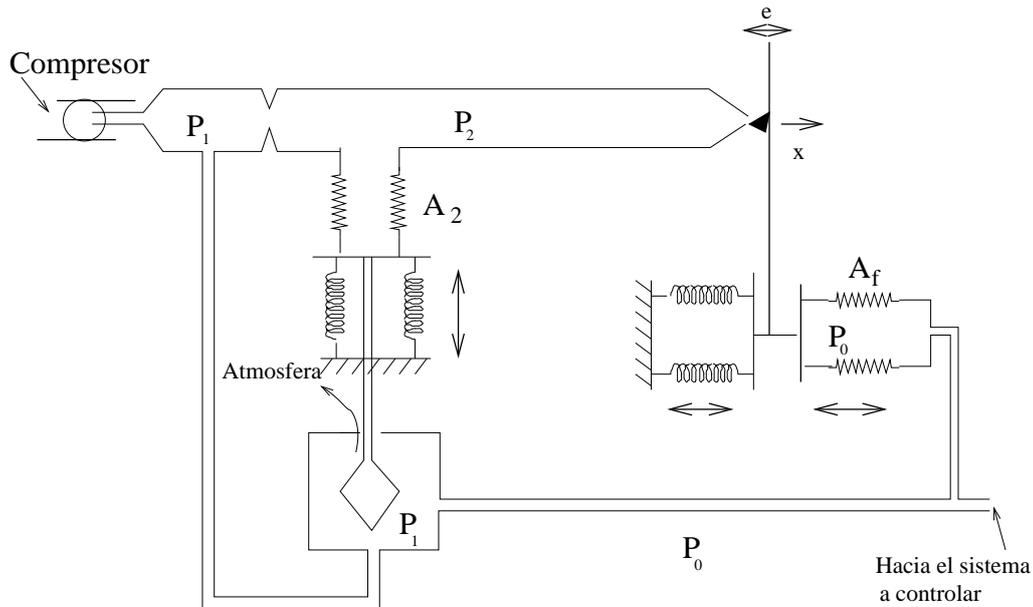


Figura 4.4: Controlador proporcional neumático

4.3.1. Motores de corriente continua (CC)

Se basan en la fuerza de Lorentz, que aparece sobre una carga que se mueve (es decir, una corriente) cuando lo hace en el interior de un campo magnético. La corriente antedicha circula por un devanado de hilo de cobre, y el campo magnético externo puede estar creado bien por un imán permanente, bien por otra bobina.

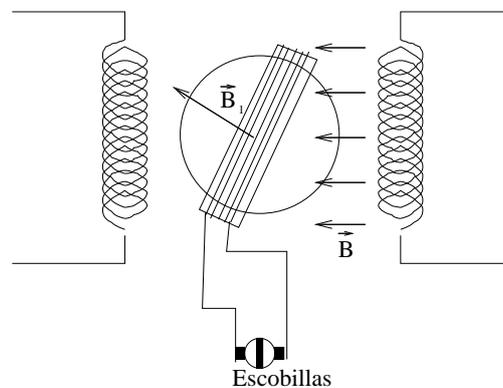


Figura 4.5: Construcción de un motor eléctrico de CC

El campo interno o inducido (\vec{B}_1) tiende a oponerse al campo externo o inductor (\vec{B}).

Los motores de corriente continua se pueden clasificar en dos tipos: controlados por armadura, y por campo. En los primeros, el campo externo es constante, y en los segundos, ambos campos pueden variar. Es claro que en los motores de imán permanente el campo externo tiene que ser constante, pues es el que crea el imán, y por tanto todos los de este tipo han de ser controlados por armadura. De hecho, es el tipo más usado, y será el que estudiaremos con detalle. Presenta algunas ventajas:

- Genera un alto torque a bajas velocidades. Esto es bueno para acelerar partiendo del reposo y para levantar cargas.

- Tienen un menor tamaño y peso que los controlados por armadura. Esto es especialmente importante si, como suele ser el caso, hay que mover el motor junto con la articulación.
- La curva velocidad/torque es lineal, lo cual simplifica el control.

Las curvas velocidad/torque e intensidad/torque son:

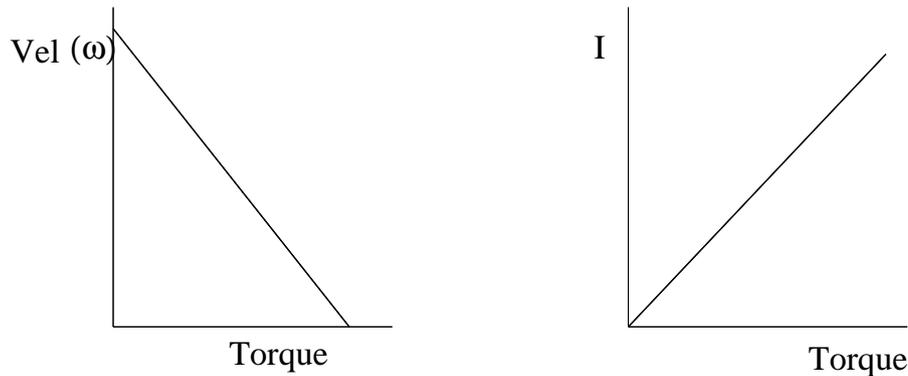


Figura 4.6: Curvas V/T e I/T para un motor de CC

Si se combinan podemos saber la dependencia velocidad/intensidad para el motor sin carga.

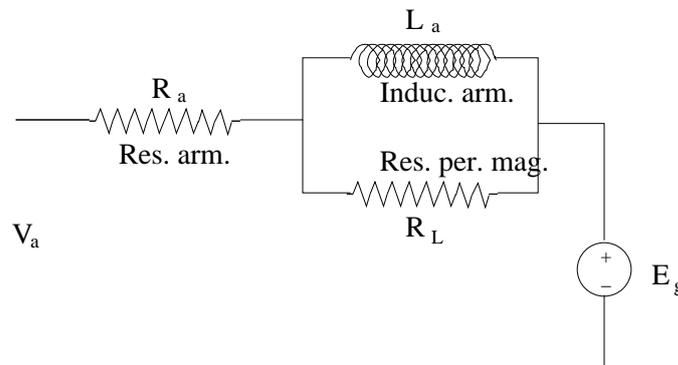


Figura 4.7: Modelo para un motor de CC

Un motor de CC controlado por armadura se puede modelizar según se muestra en la figura 4.7, donde R_a es la resistencia eléctrica de la armadura, L_a la inductancia de la armadura, R_L una resistencia equivalente que da cuenta de las pérdidas de energía en el campo magnético, y E_g la fuerza electromotriz. V_a es el voltaje que se aplica a la armadura mediante una fuente externa, y ω la velocidad angular, en principio variable con el tiempo, con la que gira el motor. La ecuación eléctrica del circuito, unida a la ley de Faraday y a la ecuación que da la dependencia lineal entre el torque y la intensidad, serán el punto de partida para el análisis del motor. R_L suele ser despreciable a las frecuencias normales de funcionamiento. Así, se cumple:

$$V_a(t) = R_a I_a(t) + L_a \frac{dI_a(t)}{dt} + E_g(t) \quad (4.1)$$

$$E_g(t) = K_E \omega(t) \quad (4.2)$$

$$T(t) = K_T I_a(t) \quad (4.3)$$

Por otra parte, el motor se usa para hacer girar a su propio eje y devanado, y a cargas externas mecánicamente unidas a él. Si llamamos J_M al momento de inercia del motor (para no confundir con la I de intensidad), J_L al momento de inercia de la carga, B al coeficiente de rozamiento viscoso, T_f al torque necesario para vencer la fricción, y T_g al torque ejercido por la fuerza de gravedad, si es que se está actuando contra ella, podemos escribir la ecuación de Newton para la dinámica de rotación

($\sum T = J\alpha = J\dot{\omega}$) como:

$$T(t) = (J_M + J_L)\dot{\omega}(t) + B\omega(t) + T_f(t) + T_g(t) \tag{4.4}$$

Tomaremos ahora transformadas de Laplace de todas estas ecuaciones, sustituyendo E_g para eliminarlo, e igualando T de las dos ecuaciones en que aparece. Por $\Omega(s)$ denotaremos la transformada de $\omega(t)$. Además, el momento de inercia total será denotado por $J_T = J_M + J_L$

$$V_a(s) = R_a I_a(s) + L_a s I_a(s) + K_E \Omega(s) \tag{4.5}$$

$$T(s) = K_T I_a(s) = J_T s \Omega(s) + B \Omega(s) + T_f(s) + T_g(s) \tag{4.6}$$

Obsérvese que podemos escribir

$$I_a(s) = \frac{V_a(s) - K_E \Omega(s)}{R_a + L_a s} \tag{4.7}$$

$$\Omega(s) = \frac{T(s) - (T_f(s) + T_g(s))}{J_T s + B} \tag{4.8}$$

y así puede verse que el diagrama de bloques del sistema descrito por estas ecuaciones sería el mostrado en la figura 4.8

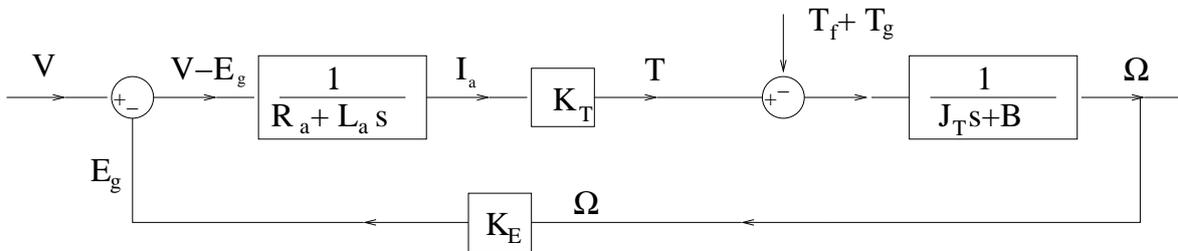


Figura 4.8: Diagrama de bloques para el motor CC

Supongamos ahora que el motor hace girar una carga externa, pero no en contra de la gravedad (p. ej., en un plano horizontal) con lo que $T_g = 0$, y que la fricción es despreciable, $T_f = 0$. En este caso, la función de transferencia total para el motor, definida como el cociente de las transformadas de Laplace de la salida, que es la velocidad angular Ω a la que gira el motor, y la entrada, que es la tensión que aplicamos a la armadura, V_a , sería:

$$G_m(s) = \frac{\Omega(s)}{V_a(s)} = \frac{\frac{K_T}{(R_a + L_a s)(J_T s + B)}}{1 + K_E \frac{K_T}{(R_a + L_a s)(J_T s + B)}} = \frac{\frac{K_T}{L_a J_T}}{s^2 + \left(\frac{L_a}{R_a} + \frac{B}{J_T}\right)s + \left(\frac{B R_a}{L_a J_T} + \frac{K_T K_E}{L_a J_T}\right)} \tag{4.9}$$

función de transferencia de segundo orden, que tiene dos polos cuyas posiciones determinarán el comportamiento transitorio del sistema. Estas posiciones dependen de los valores de las constantes físicas que intervienen en las ecuaciones. No obstante, con los valores

habituales en motores reales los polos resultan siempre reales y negativos, lo que implica que la respuesta sin carga es de tipo sobreamortiguado; en ciertos casos, uno de los polos está mucho más cercano al origen que el otro, y puede por tanto considerársele como dominante, con lo que se puede adoptar para el motor una función de transferencia aproximada de primer orden, es decir,

$$G_m(s) = \frac{\Omega(s)}{V_a(s)} \simeq \frac{K_m}{1 + \tau_m s} \quad (4.10)$$

lo cual es bastante común.

Observando de nuevo el diagrama de bloques, notamos que se trata de un sistema realimentado. Esto puede parecer extraño, puesto que aparentemente no hemos puesto en el motor ningún tipo de sensor que mida la salida, ni restador alguno que la sustraiga de la entrada. En realidad, sí los hay. El propio motor es, de por sí, un sistema realimentado, y a ésta se la llama realimentación interna. Su manifestación física consiste en que, si aumentamos el torque que el motor debe ejercer, por ejemplo, aumentando de repente la carga, en principio, la velocidad disminuye. Pero recordando las gráficas de la figura 4.6, el torque deberá aumentar, lo que hará a su vez que la intensidad que circula por la armadura aumente, haciendo crecer la velocidad, y oponiéndose de este modo en parte a la disminución inicial. Es decir, se trata claramente de realimentación negativa.

Finalmente, comentar que la función de transferencia ha sido dada dependiendo de la velocidad angular. Si se necesita relacionarla con el ángulo, recordemos que $\omega(t) = \frac{d\theta(t)}{dt}$ y por tanto, $\Omega(s) = s\Theta(s)$

Esta función de transferencia, tanto en su versión completa como simplificada, se usará como punto de partida en el tema 5 para el diseño de controladores apropiados.

4.3.2. Motores paso a paso

Igual que los motores de CC, éstos van también a convertir energía eléctrica en movimiento rotacional. A diferencia de los motores CC, aquí el devanado no está en el rotor (parte móvil) sino en el estátor (parte fija). El rotor es, o bien una sustancia paramagnética, o bien un imán permanente. Estos motores tienen varias ventajas:

- + Operan en lazo abierto, con una precisión de $\pm 1,8^0$ (o mejor, si se usa un control apropiado). En principio, el usuario puede razonablemente suponer que la señal de control que envió se ejecutará, aun sin comprobarlo con un sensor.

- + Tienen alto torque a baja velocidad angular, lo cual es bueno para levantar cargas.

- + Tienen alto torque de sostenimiento (se quedan bloqueados al mandarles una corriente constante).

- + Son directamente compatibles con señales digitales de control, pues se activan y mueven cambiando el estado de sus bobinas fijas de energizada a no energizada.

- + Los errores en la posición no son acumulativos de un movimiento a otro, en realidad, el error cometido no depende del punto en que se inició el movimiento, pues se mueven por pasos discretos.

- + No necesitan encoders ni tacómetros (si no hay lazo cerrado, no hacen falta sensores internos). Esto, sin embargo, es en cierto sentido un inconveniente, porque el controlador no sabe si algo extraño está ocurriéndole al motor (obstáculo, etc.) Se puede, en todo caso, usar algún tipo de sensor barato y poco preciso.

- + Si se quedan detenidos o agarrotados no sufren ni se queman

- + Si falla la fuente de potencia, por cortocircuito del emisor y el colector de uno de los transistores de potencia (fallo bastante común), un motor CC se mueve a máxima velocidad,

saliéndose de recorrido. Estos simplemente se paran.

Con todas estas ventajas el uso de este tipo de motores resultaría la elección obvia, pero desgraciadamente presentan también algunos graves inconvenientes:

+ La potencia mecánica que pueden suministrar es, por el momento, demasiado pequeña para su uso en la mayoría de las aplicaciones robóticas.

+ Tienen un rendimiento muy bajo, pues buena parte de la energía suministrada se disipa en forma de calor en las bobinas del estátor. Cuando más disipan es cuando están parados, a diferencia de los de CC.

+ Pueden producirse errores de posición si se manda una secuencia de pasos a velocidad excesiva y al motor no le da tiempo a seguirlos, saltándose uno o más pasos de la secuencia.

+ Tienden a ser subamortiguados, es decir, oscilan alrededor de la posición final durante cierto tiempo.

Hay dos tipos principales de motores paso a paso, cuyo esquema y funcionamiento describiremos a continuación.

+ **Motores de reluctancia variable (VR)**: Están formados por un estátor y un rotor, el primero, torneado sobre una pieza cilíndrica con sección estrellada de cierto número de dientes, N_r . El estátor está vaciado en otro cilindro, mostrando también un número diferente de dientes (N_s) hacia el interior, alrededor de cada uno de los cuales hay arrollada una bobina. De hecho, el estátor puede ser simplemente un conjunto de bobinas apropiadamente orientadas. El rotor se debe construir en material paramagnético. Para entender el funcionamiento véase la figura 4.9

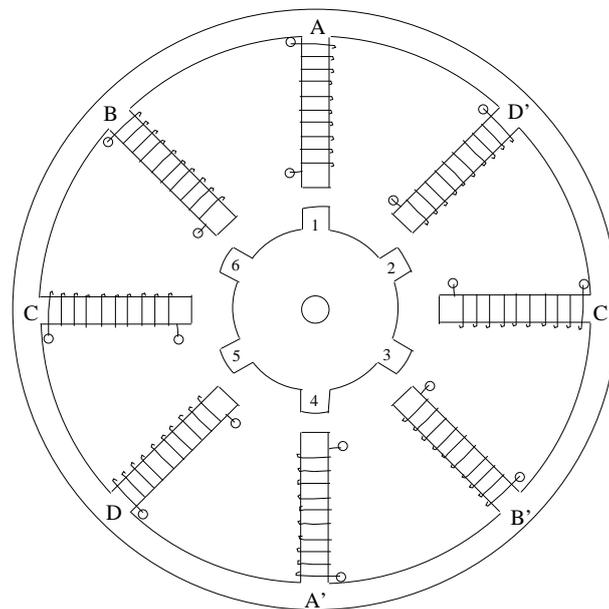


Figura 4.9: Funcionamiento del motor paso a paso VR

El mostrado en la figura es un motor de cuatro fases, que llamaremos A,B,C y D. Las bobinas de cada fase (p.ej., A y A') están colocadas en posiciones opuestas, y están en serie, pero devanadas al revés, de modo que se energizan simultáneamente al darles corriente, mostrando una su polo magnético N hacia el interior, y otra el polo S. De este modo las líneas de campo magnético fluyen de la una a la otra por el interior del rotor. Pero como éste era paramagnético, tiende a alinearse de modo que el máximo número de líneas de fuerza lo atraviesen. En la figura, esto está ocurriendo al alinearse el diente 1 del rotor

con A, y el diente 4 con A'. Mientras se siga enviando corriente a A-A', el rotor tenderá a mantenerse así.

Supongamos que ahora se desenergiza el par A-A', y se energiza B-B'. El rotor se mueve de modo que los dientes 6 y 3 se alineen respectivamente con B y B'. El resto de la secuencia es análogo, y se ve en la figura. Después de un ciclo de energización A-B-C-D-A..., el rotor ha girado 60° . En este caso, el ciclo completo consta de 4 pasos. Para calcular el ángulo girado en cada paso, démonos cuenta de que es la resta del ángulo entre cada par de dientes del rotor menos el ángulo entre cada par de bobinas. Así pues

$$\text{rad/paso} = \left(\frac{2\pi}{N_r} - \frac{2\pi}{N_s} \right) = 2\pi \frac{N_s - N_r}{N_s N_r}$$

siendo N_s el número de bobinas del estátor y N_r el número de dientes del rotor. Esta sería también la máxima precisión angular obtenible. Por otra parte, el número de pasos por revolución es

$$\text{pasos/rev} = \frac{2\pi \text{ rad/rev}}{2\pi \frac{N_s - N_r}{N_s N_r} \text{ rad/paso}} = \frac{N_s N_r}{N_s - N_r} \text{ pasos/rev}$$

En el ejemplo de la figura, el ángulo por paso sería $2\pi \frac{8-6}{8 \cdot 6} = \frac{\pi}{12} = 15^\circ$, y el número de pasos por revolución, $\frac{8 \cdot 6}{8-6} = 24$

La circuitería lógica iría generando impulsos para cada bobina, en la forma

... 1000 \longleftrightarrow 0100 \longleftrightarrow 0010 \longleftrightarrow 0001 \longleftrightarrow 1000 ...

en un sentido o en otro, según se desee el movimiento.

Aparece un problema: cada impulso es como una entrada escalón, y la respuesta suele ser subamortiguada, de modo que a cada paso el eje oscilaría. La amplitud de estas oscilaciones depende del ángulo del paso; por ello, una buena solución sería reducirlo. Usualmente se puede llegar a 100 pasos por revolución, que son $3,6^\circ$ por paso, pero se puede usar el control llamado de semipaso. Se consigue energizando con la misma intensidad bobinas contiguas. La secuencia, en vez de ser A-B-C-D-A sería A-AB-B-BC-C-CD-D-DA. El rotor quedará en los pasos de dos bobinas en medio de ambas, siempre que ambas sean iguales y se energizen por igual. Habrá entonces doble número de puntos de equilibrio. La circuitería de conmutación es sólo un poco más complicada, generando la secuencia

... 1000 \longleftrightarrow 1100 \longleftrightarrow 0100 \longleftrightarrow 0110 \longleftrightarrow 0010 \longleftrightarrow 0011 \longleftrightarrow 0001 \longleftrightarrow 1001 ...

Otra solución al problema de la sobreoscilación consiste en cambiar a la secuencia de energización anterior justo antes de que el rotor llegue al paso dado, e inmediatamente volver a cambiarla. El instante de cambio es crítico, y puede depender de la carga externa.

Otro tipo interesante de motores paso a paso son los de imán permanente (PM). El rotor es un imán cilíndrico, tallado en los extremos con varios dientes que están desfasados entre la parte superior y la inferior (es decir, un diente de la parte superior del cilindro, polo N, se encuentra en la vertical de un vano del polo S, y viceversa). Véase la figura 4.10. La estructura del estátor es prácticamente idéntica a los motores VR. Para controlarlos se requiere una fuente bipolar; las dos bobinas de cada fase (en este caso, dos fases, A-A' y B-B') se energizan al revés, de modo que la superior, creando un polo N, atraería al S directamente enfrente a ella, y la inferior, creando un S, al N igualmente enfrente. El siguiente paso sería energizar B de modo que crease un S, y B' de modo que crease un N, con lo cual el imán gira en sentido antihorario.

Existen también motores paso a paso lineales. Funcionan mediante un principio similar, aunque un poco más complicado que los rotacionales. Están constituidos por una regleta dentada de material paramagnético, con los huecos de los dientes rellenos de material diamagnético, que tiene encima un imán permanente en cuyos extremos hay dos electroimanes.

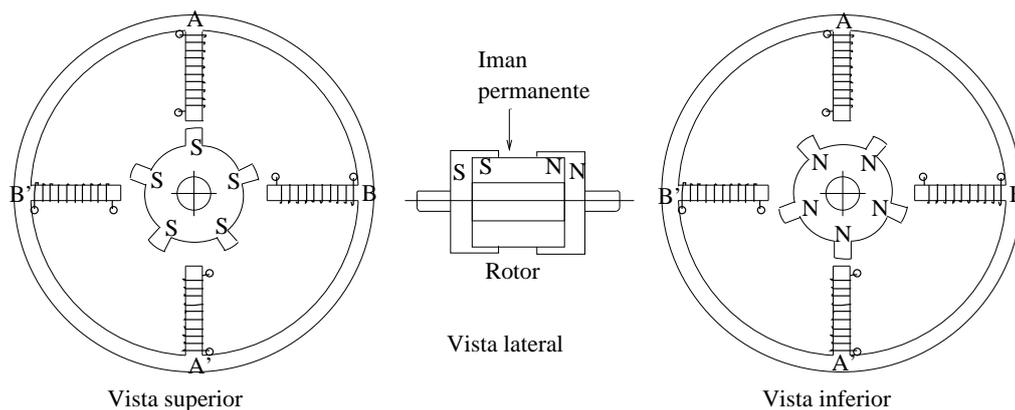


Figura 4.10: Esquema de un motor paso a paso de imán permanente

Esta pieza superior se sustenta en equilibrio sobre la inferior gracias a un colchón de aire a presión inyectado por pequeños orificios.

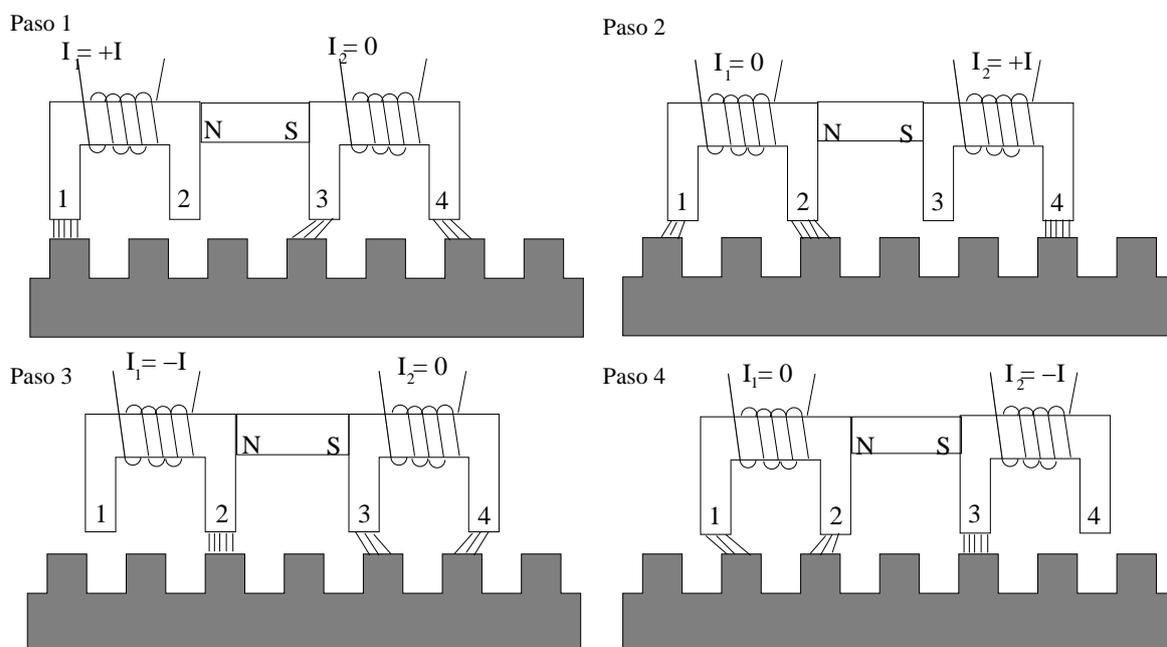


Figura 4.11: Esquema de un motor paso a paso lineal

El funcionamiento sigue el esquema que se muestra en la figura 4.11. Cuando hay intensidad de corriente en un sentido dado, el campo magnético creado por la bobina anula el flujo magnético del imán por uno de sus polos, lo que hace que todo el sistema avance hasta situar el otro polo sobre el diente de material paramagnético. La secuencia es $I/0$, $0/I$, $-I/0$, $0/-I$, ... que hace avanzar el bloque completo la longitud de medio diente. Existen también versiones bidimensionales con dos imanes perpendiculares.

El último tipo reseñable de motores paso a paso son los llamados "brushless" (literalmente, sin escobillas, aunque en realidad ningún motor paso a paso las tiene). Están formados por varias bobinas e imanes permanentes que giran al dar la secuencia de energización apropiada. Para la conmutación de las bobinas se pueden usar sensores de efecto Hall colocados en un extremo; éstos se activan cuando el campo magnético de los imanes pasa bajo ellos.

Finalmente, comentar que todos los tipos de motores paso a paso se pueden controlar

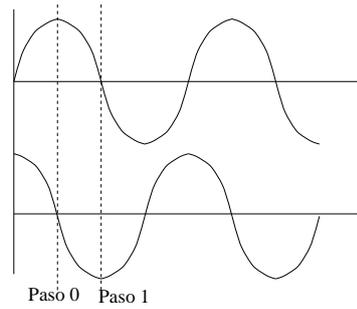


Figura 4.12: Esquema del motor brushless

por pasos, por semipasos, o en modo continuo. Para este último caso, imaginemos p. ej. un motor de dos fases de imán permanente, y que a cada fase se le manda una señal de la forma que se muestra en la figura 4.13.

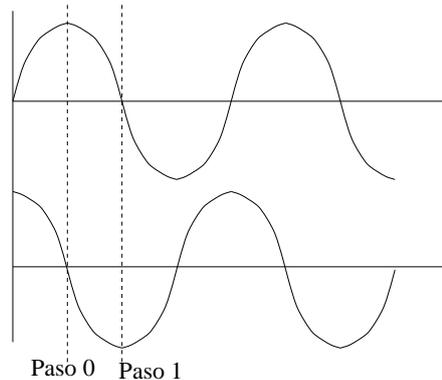


Figura 4.13: Señales para control continuo de motores paso a paso

Se puede parar el envío de estas señales de tensión, no necesariamente en 'Paso 0' o en 'Paso 1', sino en cualquier punto intermedio; esto daría una energización diferente a ambas bobinas, y situaría el rotor en un punto intermedio arbitrario entre una y otra. A efectos de cálculo, el seno y el coseno no tienen por qué ser calculados llamando a las funciones trigonométricas en cada instante; pueden ser obtenidos a partir del valor en el instante anterior, ya que, si δ es el incremento angular que deseamos para cada periodo (dependiente de la velocidad angular pedida), se tiene que

$$\begin{aligned} \sin(x + \delta) &= \sin x \cos \delta + \cos x \sin \delta \\ \cos(x + \delta) &= \cos x \cos \delta - \sin x \sin \delta \\ s_{i+1} &= s_i \cos \delta + c_i \sin \delta \\ c_{i+1} &= c_i \cos \delta - s_i \sin \delta \\ \begin{pmatrix} s_{i+1} \\ c_{i+1} \end{pmatrix} &= \begin{pmatrix} \cos \delta & \sin \delta \\ -\sin \delta & \cos \delta \end{pmatrix} \begin{pmatrix} s_i \\ c_i \end{pmatrix} \end{aligned}$$

lo cual sólo requiere unas pocas multiplicaciones y sumas, pues δ es constante, y por tanto lo son su seno y coseno. Para girar al revés la matriz es análoga, cambiando δ por $-\delta$.

4.4. Servo-amplificadores

Como complemento a este tema, mencionemos estos dispositivos que, aun cuando no son propiamente robóticos, son necesarios para el funcionamiento de los actuadores eléctricos.

Los servo-amplificadores convierten la señal de baja tensión generada por el sistema de control en una de más alta tensión y la suficiente intensidad como para hacer girar el motor y la carga a él unida. Hay dos tipos esenciales: lineales, y de modulación PWM.

+Servo-amplificadores lineales: Están basados en el uso de transistores de potencia que funcionan en la zona de característica lineal, es decir, en clase A, con lo que la intensidad que circula entre colector y emisor es proporcional a la intensidad de base, o sea, $I_c = \beta I_B$, donde la constante β está entre 10 y 200, dependiendo del transistor. Existen dos variantes, el tipo H y el T. Sus esquemas pueden verse en la figura 4.14

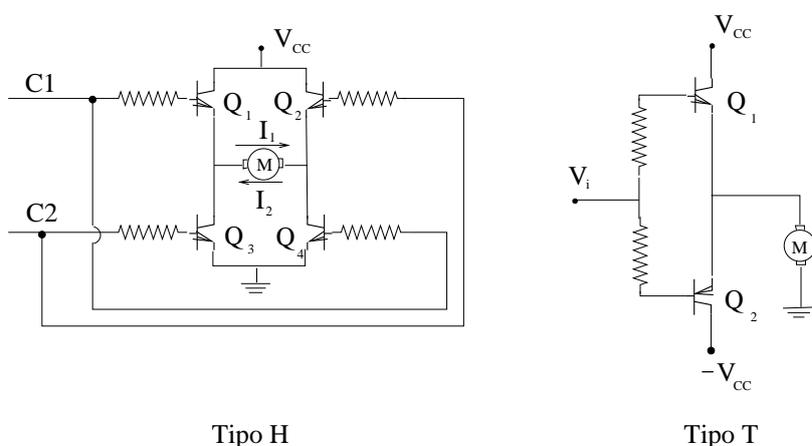


Figura 4.14: Esquema de los servo-amplificadores lineales

Respecto al tipo H, si $C1$ es un voltaje positivo, y $C2$ se conecta a tierra, Q_1 y Q_4 conducen, y la corriente es I_1 , con lo que el motor gira en cierto sentido. Si, por contra, $C1$ se conecta a tierra y $C2$ es positivo, son Q_2 y Q_3 los que conducen, y entonces la intensidad es I_2 , con lo que el motor gira en sentido opuesto. La velocidad se controla cambiando la intensidad de base, I_B , la cual es proporcional a la tensión de entrada. En esta configuración el motor está "flotante" (es decir, no conectado a tierra). En el tipo T no ocurre esto, pero se requiere alimentación bipolar; este caso usa sólo dos transistores complementarios. La V_i debe ponerse positiva o negativa, según el sentido de giro deseado. En ambos tipos es necesario poner diodos de protección (el motor tiene alta inductancia y provoca extracorrientes de apertura y cierre), y también disipadores de calor para los transistores, pasivos, o incluso activos (ventiladores).

+Amplificadores tipo PWM: Los amplificadores controlados por anchura de pulso (Pulse Width Modulation, o PWM) se basan en el hecho de que la función de transferencia del motor puede verse como la de un filtro pasa-baja, y por tanto si se le envía una onda, digamos cuadrada, de frecuencia mayor que la frecuencia de corte, el motor se comporta aproximadamente como un integrador, y gira con una velocidad proporcional al valor de la integral de la corriente (a impedancia fija, de la tensión) que lo atraviesa. Son iguales a los anteriores, pero la tensión de entrada es un onda cuadrada, de amplitud tal que haga trabajar a los transistores en saturación cuando da el valor máximo, y en corte cuando da el mínimo. De este modo los transistores trabajan en clase C, y esto hace que apenas disipen calor. Ello es porque la potencia disipada en un transistor es $W_{dis} = V_{CE}I_C$, siendo V_{CE} la

tensión colector-emisor, e I_C la intensidad de colector. Con el transistor en corte, la V_{CE} vale V_{CC} , pero la I_C es nula; en saturación, la I_C es la I que circula por el motor, pero la V_{CE} es nula. La onda enviada a la entrada hará que la tensión en bornes del motor sea como se muestra en la figura 4.15.

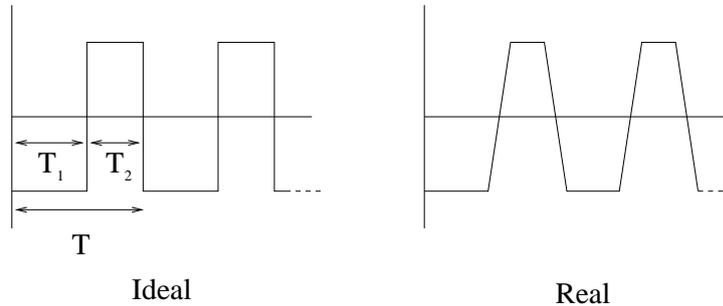


Figura 4.15: Tensión en bornes de un motor CC con control PWM

El periodo T , suma de T_1 y T_2 , es fijo, y suele ser tal que la frecuencia va de 1 a 10 KHz. Sin embargo, T_1 es variable (y con él, por supuesto, T_2), lo cual da una tensión eficaz diferente. Esta es:

$$V_{ef} = \frac{1}{T} \int_0^T V(t) dt = \frac{1}{T} \left[\int_0^{T_1} (-V_{CC}) dt + \int_{T_1}^T V_{CC} dt \right] = \frac{1}{T} [-V_{CC}T_1 + V_{CC}(T - T_1)] = \frac{V_{CC}}{T} [T_2 - T_1]$$

Es claro que, si $T_1 = T_2$, la tensión eficaz es cero, con lo que el motor está parado, pero frenado. En los casos extremos ($T_1 = T$ y $T_2 = 0$, o bien $T_1 = 0$ y $T_2 = T$) el motor gira a máxima velocidad en uno u otro sentido.

Las ventajas principales de este método, que lo convierten en el más usado, son la precisión que se puede obtener en el control, y la idealmente nula pérdida de potencia en los transistores. En realidad, la conmutación del transistor no es instantánea, con lo cual la tensión real en bornes del motor es la mostrada en la parte derecha de la figura 4.15, de modo que sí se da cierta disipación de calor.

4.5. Transmisiones mecánicas y dispositivos de conversión

Para concluir este tema veremos los elementos que ligán al actuador con el eje físico al que hacen girar o desplazan, el efector. A este eje, y a todo lo que esté unido a él, lo llamaremos la carga.

La misión de las transmisiones es cambiar la dirección de rotación (caso de los piñones acodados), cambiar el eje de rotación (correas), multiplicar el torque y reducir (usualmente) la velocidad de rotación. También pueden convertir movimiento rotacional en lineal, o viceversa. En cualquier caso, siempre debe procurarse que se transmita la máxima energía desde el actuador a la carga. Un problema es que en muchos actuadores la relación movimiento del actuador/movimiento del efector es absolutamente no lineal, lo cual hace difícil el control.

Desde luego, sería deseable no tener que usar transmisiones y mover los elementos directamente con el actuador, pero esto no suele ser posible por razones de coste o de tamaño. Una notable excepción es el actuador tipo MegatorqueTM que equipan los robots ADEPTTM, cuyo eje es el propio eje de rotación.

Pasemos ahora a analizar cada uno de los posibles tipos de conversión de movimiento.

± Conversión de movimiento rotatorio a rotatorio: Se usa cuando se desea variar la velocidad angular (y, como veremos, el torque ejercido) por un motor. El dispositivo

usual para esto es un par o un tren de engranajes. Idealmente, los engranajes debieran ser perfectamente circulares, y rotar exactamente sobre su centro, sin inercia, y sin rozamiento entre sus superficies. Uno de los engranajes será la entrada, y el otro la salida. Véase la figura 4.16

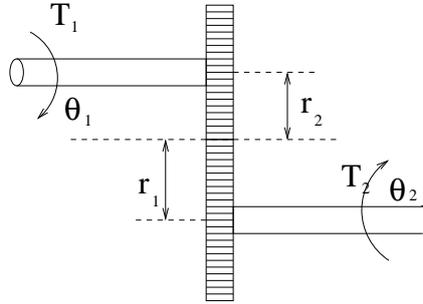


Figura 4.16: Par de engranajes

Supongamos que la entrada tiene un radio r_1 y N_1 dientes, y la salida tiene radio r_2 y N_2 dientes. Ambos están unidos a ejes que ejercen torques respectivos T_1 y T_2 , y que giran ángulos θ_1 y θ_2 . Véase la figura 4.16. Para que puedan encajar, deben construirse de modo que el número de dientes sea proporcional a la longitud de su circunferencia, y por tanto, proporcional al radio, pues la medida de cada diente debe ser igual en uno y otro engranaje. Así, se tiene

$$\frac{N_1}{r_1} = \frac{N_2}{r_2}$$

Además, N_1 y N_2 son enteros. Al cociente N_1/N_2 , usualmente denotado $N_1 : N_2$, se le llama relación de acoplo. Si no hubiese pérdidas por rozamiento, la energía, aquí obtenida como trabajo de rotación, debe conservarse, y la longitud de arco recorrida también debe ser la misma, pues encajan, por lo que

$$r_1\theta_1 = r_2\theta_2 \quad (4.11)$$

$$T_1\theta_1 = T_2\theta_2 \quad (4.12)$$

De aquí se obtiene

$$\frac{N_1}{N_2} = \frac{r_1}{r_2} = \frac{\theta_2}{\theta_1} = \frac{T_1}{T_2} = \frac{\omega_2}{\omega_1} = \frac{\alpha_2}{\alpha_1}$$

donde se ha derivado respecto al tiempo la ecuación 4.11 para obtener a partir del ángulo las velocidades angulares ω y las aceleraciones angulares α . Estas igualdades sirven para calcular cualquier característica de la salida, conocidas las de la entrada.

Un par de engranajes actúa como un transformador de torques, y por tanto de momentos de inercia. Si a la salida hay una carga con momento de inercia J_2 que se mueve con una aceleración angular α_2 , se tiene que

$$T_2 = J_2\alpha_2 = T_1 \frac{N_2}{N_1}$$

luego

$$T_1 = \frac{N_1}{N_2} J_2\alpha_2$$

Nótese que el eje 1 gira con una aceleración angular $\alpha_1 = \frac{N_2}{N_1}\alpha_2$. Ahora, nos preguntamos qué momento de inercia debiera tener una carga simple (es decir, fija al eje 1) para que

éste girase con la misma aceleración angular con que lo hace (α_1) al serle aplicado el mismo torque, T_1 . Lo llamaremos momento de inercia equivalente, J_{eq} , y por definición,

$$T_1 = J_{eq}\alpha_1$$

Es claro que J_{eq} es el que ve el eje 1 al conectársele el engranaje y la carga 2. Entonces,

$$J_{eq}\alpha_1 = J_{eq}\frac{N_2}{N_1}\alpha_2 = \frac{N_1}{N_2}J_2\alpha_2$$

por tanto

$$J_{eq} = \left(\frac{N_1}{N_2}\right)^2 J_2$$

En el caso de que la salida tenga más dientes (más radio) que la entrada, la inercia reflejada es menor que la J_2 real; en ese caso, $\omega_2 < \omega_1$ (se reduce la velocidad) y $T_2 > T_1$ (se aumenta el torque ejercido). Este es el caso usual en robots, donde pequeños motores pueden así mover cargas grandes. Hay un límite: si se transmite demasiado torque, las fuerzas tangenciales sobre cada diente aumentan, y pueden llegar a romperlos. Aparte, hay un límite de tamaño; si quisiéramos una relación, digamos, de 100 : 1, un piñón de entrada con radio pequeño, p. ej., 1 cm, necesitaría uno de salida de radio 1 metro. Esto se puede solucionar usando trenes de n engranajes, cada uno de los cuales multiplicase el número de dientes por un factor p , con lo cual la relación de acoplo total sería de $n^p : 1$, pero situándolos engranzados, para reducir espacio. Si, por contra, el actuador y el eje están muy separados, se pueden usar engranajes grandes, pero suelen ser pesados, e incrementar demasiado la inercia.

Una solución es la llamada transmisión armónica, que usa un soporte cilíndrico externo, fijo o no, con la cara interna dentada.

Como último dispositivo de conversión de movimiento rotatorio a rotatorio hay que mencionar las correas y cadenas, que se usan uniendo dos poleas o ruedas dentadas cuando hay que transmitir el movimiento a distancias mayores. A veces, las correas pueden tener muescas que encajan en engranajes anchos para evitar el deslizamiento. Un problema con estos dispositivos es una cierta holgura provocada por la deformación de la correa o cadena que hace que un bloqueo en una de las partes conectadas no sea inmediatamente percibido por la otra. Además, el desgaste obliga a reemplazamientos periódicos.

Conversión de movimiento rotacional a lineal: Se usan sobre todo cuando motores usuales deben mover articulaciones traslacionales. Hay varios dispositivos apropiados para realizar esta conversión:

+Tornillo fijo: Se basan en una tuerca ancha enroscada al centro de un tornillo que puede girar libremente, y la carga se fija a esta tuerca. Véase la figura 4.17.

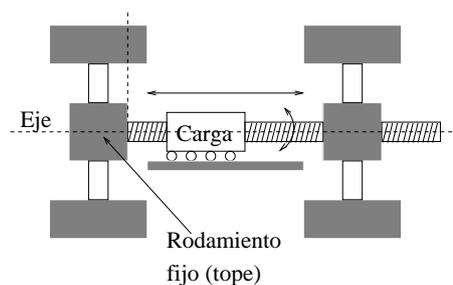


Figura 4.17: Tornillo fijo

Aquí la razón de acoplo, P , se define como el ángulo que hay que girar la entrada para obtener en la salida un desplazamiento lineal unitario. Un valor típico puede ser $P = 4$ vueltas/cm. A veces se usa su inversa, $L = 1/P$.

Siendo x el desplazamiento lineal, θ el ángulo, ω la velocidad angular, α la aceleración angular, y v y a la velocidad y aceleración lineales de la salida, las ecuaciones son:

$$\theta = Px \quad (4.13)$$

$$\omega = Pv \quad (4.14)$$

$$\alpha = Pa \quad (4.15)$$

Respecto al momento de inercia equivalente que ve el motor que hace girar el tornillo, se puede calcular por igualación de energías: la cinética de rotación del tornillo, que se convierte en energía cinética de traslación de la carga (suponiendo que no hay pérdidas de energía por rozamiento, lo cual no es en realidad nada correcto). En este caso

$$\frac{1}{2}Mv^2 = \frac{1}{2}J_{eq}\omega^2 = \frac{1}{2}J_{eq}P^2v^2$$

de donde $J_{eq} = M/P^2$.

Otros posibles dispositivos de conversión rotacional-lineal son el sistema piñón-cremallera, la correa con polea doble, el sistema biela-manivela y las transmisiones variables con levas. Estas últimas se usan para movimientos irregulares. Todos estos se muestran en la figura 4.18.

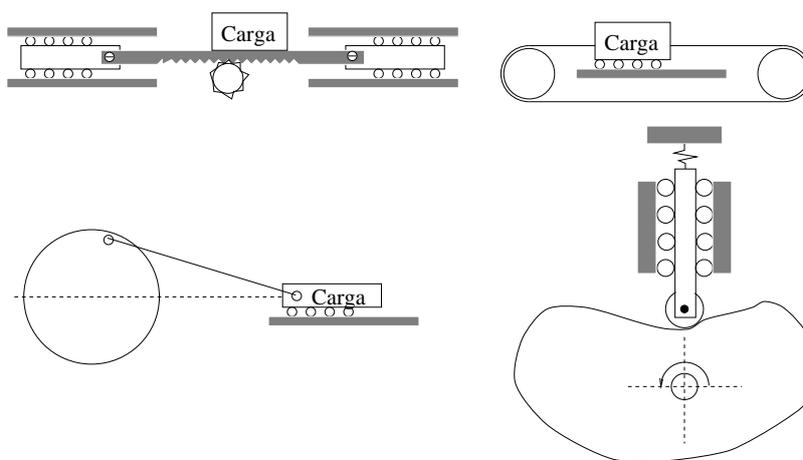


Figura 4.18: Dispositivos de conversión rotacional-lineal

+Acopladores: Son dispositivos mecánicos usados para conectar dos ejes que no sean exactamente paralelos, o estén ligeramente desalineados. Suelen hacerse con materiales flexibles, pero resistentes, como un trozo de espiral de nylon, o muelle de acero. También existe la llamada transmisión universal, formada por dos tubos conectados por una cruz de ejes perpendiculares, como se muestra en la figura 4.19.

+Transferencia de potencia: Es necesario mencionar que en cualquiera de estos dispositivos la potencia se transmite desde el motor hasta la carga de tal manera que la máxima transmisión se da cuando la inercia equivalente (o reflejada) que ve el eje de entrada es igual a la inercia de él mismo junto con el motor que lo mueve. Se puede conseguir que esto se cumpla eligiendo la relación de acoplo apropiada; p.ej., en un par de engranajes, se vio que

$$J_{eq} = \left(\frac{N_1}{N_2}\right)^2 J_2$$

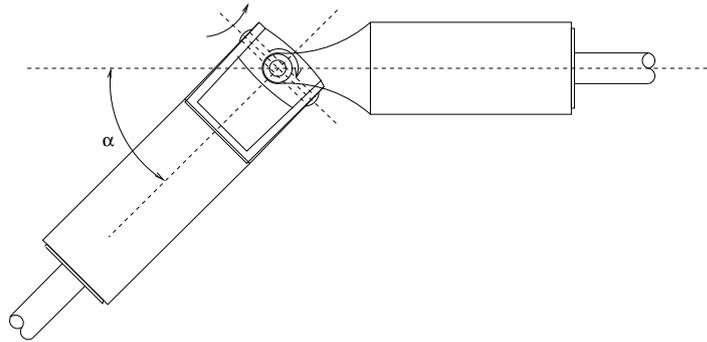


Figura 4.19: Transmisión universal

Si elegimos $\frac{N_1}{N_2} = \sqrt{\frac{J_1}{J_2}}$, se tiene que

$$J_{eq} = \left(\sqrt{\frac{J_1}{J_2}} \right)^2 J_2 = J_1$$

+Problemas con componentes reales: Todo lo indicado hasta ahora supone dispositivos mecánicos perfectos. Desde luego, esto no es así, y se presentan siempre varios problemas que se deben comentar:

+Rozamiento: Provoca invariablemente pérdidas, a veces sustanciales, de energía. Para evaluarlas se define el rendimiento η como:

$$\eta = \frac{\text{Pot. suministrada}}{\text{Pot. consumida}} = \frac{\text{Trab. suministrado}}{\text{Trab. realizado}}$$

P. ej., en un tren de engranajes, si T es torque, η sería

$$\eta = \frac{T_{sal}(real) \cdot \theta_s}{T_{ent} \cdot \theta_e} = \frac{T_{sal}(real) \cdot \frac{N_1}{N_2}}{T_{ent}} = \frac{T_{sal}(real)}{T_{ent} \cdot rel_acoplo}$$

Cuanto menor sea el rozamiento del sistema, más se acercará a uno (100%) el rendimiento. En la práctica, puede tomar valores tan bajos como el 20%.

Rendimientos menores que 1 no afectan a las relaciones especificadas entre ángulos, velocidades y aceleraciones, pero sí a la relación entre torques.

+Excentricidad: Es el problema que se presenta por el hecho de que las ruedas o engranajes no giren alrededor de su auténtico centro, sino de un punto separado décimas o centésimas de mm. de él. Esto provoca fuerzas (y por tanto, torques) no constantes sobre el otro engranaje, además de rozamientos altos en determinados puntos, y por tanto, desgaste.

+Holgura: Es la falta de encaje perfecto de ciertos dientes de un engranaje con los de su par cuando existe excentricidad. Se trata de los dientes situados a menor distancia (radio) del centro. La holgura provoca que el movimiento no se transmita inmediatamente de un engranaje al otro al arrancar desde la situación estática; el problema se agudiza en la inversión del sentido de movimiento.

+Errores diente a diente: No todos los dientes de un engranaje son exactamente iguales, y esto provoca irregularidad en el movimiento de la carga, y alinealidades. Una gráfica que representa el error en el ángulo (es decir, valor esperado menos valor obtenido) en función del propio ángulo de giro, suponiendo relación de acoplo de 1:1, podría ser la mostrada en la figura 4.20. En ella, la sinusoide mayor es provocada por la excentricidad, y las menores superpuestas a ella, por los errores diente a diente.

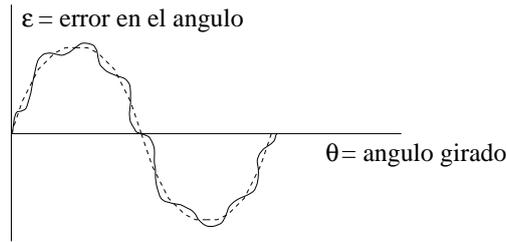


Figura 4.20: Alinealidad en engranajes

+**Vibraciones**: Estrictamente, no son un problema propio sólo de las transmisiones, sino de toda parte mecánica del robot, pero se incluyen aquí por similitud. La aparición de vibraciones en un robot es debida a resonancias mecánicas de piezas o conjuntos de ellas cuya frecuencia natural de oscilación es momentáneamente excitada por golpes o por el propio movimiento de los motores. Suelen ser transitorias, excepto si la perturbación que las provoca es periódica, en cuyo caso debieran evitarse completamente por la posibilidad de provocar resonancias que causan fatiga en el metal, y a la larga, rotura. Se debe procurar diseñar los sistemas mecánicos bien rígidos, y en cualquier caso, de modo que su frecuencia natural no coincida con ninguna perturbación periódica presente en el entorno. La determinación de la frecuencia natural de oscilación de un sistema mecánico complejo, como un robot, es difícil, y se puede realizar bien por simulación mediante técnicas de análisis de elementos finitos, bien mediante respuesta experimental al impulso, o respuesta en frecuencias.

4.6. Precisión, repetibilidad y resolución

Estos son tres conceptos importantísimos en un robot, y están relacionados con la capacidad de posicionarse en el punto deseado. Dependen de muchos factores externos, tales como carga, temperatura, velocidad, dirección del movimiento, punto del espacio de trabajo en que se opera, etc. Los definiremos uno por uno a continuación.

+**Resolución**: es la mínima diferencia entre dos puntos consecutivos en que el robot puede situarse, siendo capaz de distinguir uno de otro. Hay dos tipos de resolución:

+**En el control**: es la variación más pequeña entre puntos consecutivos que el sistema de control es capaz de detectar.

+**En la posición**: Es la máxima distancia entre dos puntos consecutivos que la articulación puede adoptar, debida a imprecisiones mecánicas. Máxima distancia significa en el peor de los casos, es decir, cuando el error cometido en un punto y en el consecutivo se oponen. Estos dos tipos de error se aprecian en la figura 4.21, suponiendo, p. ej., desplazamiento lineal.

+**Precisión**: es el máximo error con el que se alcanza un punto del espacio no visitado previamente. Aparentemente, sería la mitad de la resolución en la posición, pero si el punto viene dado en coordenadas cartesianas, suele ser mucho mayor, porque a la imprecisión mecánica antedicha se suman la imprecisión numérica en el cálculo de la cinemática inversa, más la debida a las diferencias provocadas por el uso de los parámetros teóricos para el robot en lugar de los reales (los valores imprecisos de d_i , a_i y α_i en las matrices DH). Esto puede provocar grandes diferencias (de hasta varios milímetros) entre las posiciones predicha y real a las que se llega. Según la zona del espacio de trabajo que se considere, la precisión varía, por lo que se suele distinguir entre precisión global (la media) y local para una determinada zona.

+Repetibilidad: Es la capacidad de un manipulador robótico de posicionarse en el mismo punto al que fue mandado anteriormente. Idealmente, las medidas debieran realizarse en idénticas condiciones de temperatura, dirección y velocidad de aproximación. La forma correcta de medir la repetibilidad es usar métodos estadísticos, enviando muchas veces el brazo a la misma posición. Las gráficas de posición alcanzada podrían ser similares a las mostradas en la figura 4.21, donde la derecha muestra la diferencia que puede darse según cuál sea la dirección de aproximación.

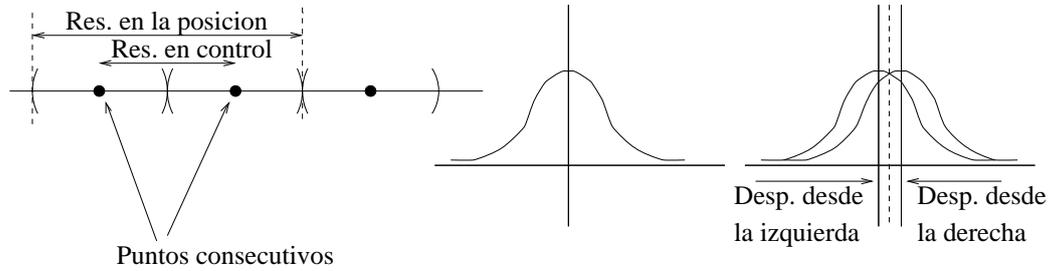


Figura 4.21: Resolución y medidas de repetibilidad

Tema 5: Introducción al control de robots

En este tema se estudiarán las diversas formas de control de actuadores robóticos a partir de las técnicas y métodos de modelización vistos en módulos anteriores, en concreto, usando la formulación de Laplace aplicada a sistemas lineales. Aun cuando, según se vio en el tema 4, es posible controlar los actuadores hidráulicos y neumáticos, la absoluta preponderancia de los motores eléctricos de CC hará que nos centremos particularmente en ellos. Más restringidamente, se estudiará el control de un motor, que se supone actúa sobre una articulación aislada. Esto es sólo una aproximación, dado que cuando el robot se mueve, la inercia que ve cada articulación depende de la posición, y por tanto, no se puede considerar que la función de transferencia admitida para motor más carga sea constante, lo que excluiría en principio las técnicas básicas de control de sistemas lineales. No obstante, la inmensa mayoría de los robots comerciales hoy en uso admiten esta aproximación, que sigue dando resultados, al menos, tolerables usada con las debidas precauciones. Se incidirá en ello al hablar de diseño de controladores.

5.1. Técnicas de control clásico sobre motores CC

Como se vio en el tema anterior, un motor de CC puede ser modelizado mediante una función de transferencia de segundo orden, que en muchos casos (uno de los polos suele ser dominante) es incluso aproximable mediante una de primer orden. Esto simplificará el análisis inicial, y permitirá tomar en cuenta un aspecto más: el amplificador de potencia que alimenta al motor, y que también se describió en el tema anterior. Un amplificador ideal debería ser absolutamente lineal, es decir, tener como función de transferencia una constante A ; pero en la práctica el modelo correcto resulta ser

$$G_{amp}(s) = \frac{A}{1 + \tau_a s}$$

Para ejercer control realimentado deberemos medir posición y/o velocidad. Si disponemos de un sensor analógico de velocidad (un tacómetro) la posición se puede obtener por integración con el circuito apropiado. Si, por contra, tenemos un sensor digital de posición (un codificador óptico) la velocidad se obtiene por derivación numérica. Suponiendo el motor aproximado por la f.d.t. de primer orden

$$G_m(s) = \frac{K_m}{1 + \tau_m s}$$

el lazo de realimentación típico para el control del motor sería el mostrado en la figura 5.1

En esta figura, K_g es la constante del tacómetro que relaciona la velocidad angular medida con la señal que éste genera. La K_p es la constante de realimentación de posición, que el usuario deberá fijar. El bloque $\frac{1}{s}$ situado a la salida del motor representa en cierto sentido al codificador óptico, y da cuenta del hecho de que la f.d.t. del motor relaciona tensión V con velocidad angular, $\omega = \dot{\theta}$, mientras que lo que se lee usando el codificador es la posición angular θ . La entrada al bucle más interno es el error en la velocidad en cada momento, y puede ser visto como un comando de velocidad deseada.

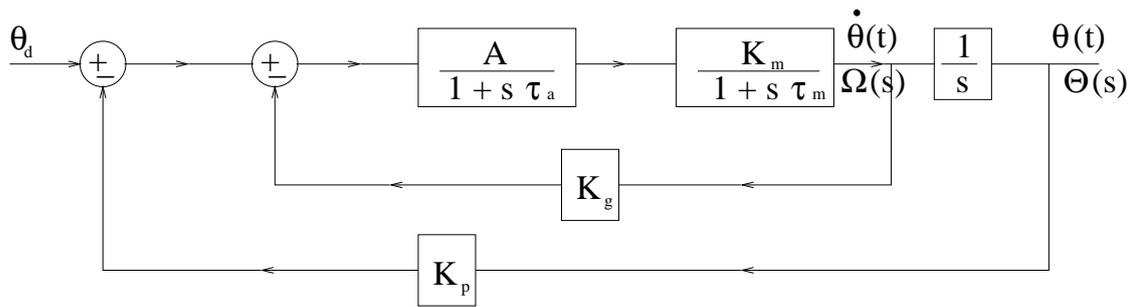


Figura 5.1: Control de un motor CC: figura principal

Analizaremos este esquema según los diferentes casos que pueden darse:

Si no hay realimentación de velocidad (no hay tacómetro) podemos considerar la $K_g = 0$, y en ese caso la f.d.t. en lazo abierto sería

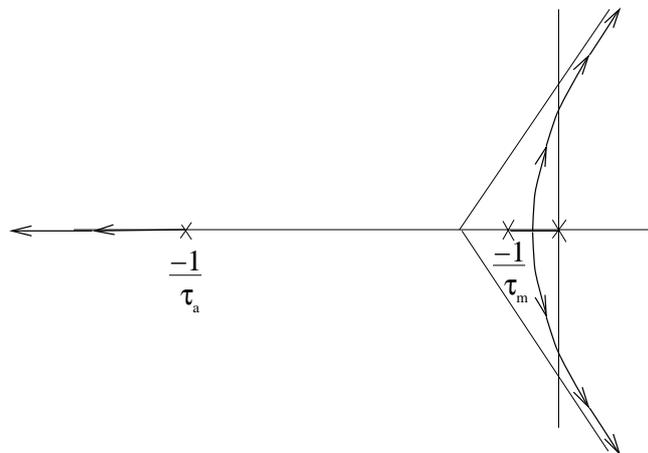
$$G(s) = \frac{AK_m}{s(1+s\tau_m)(1+s\tau_a)}$$

siendo el bloque de realimentación $H = K_p$, y por tanto la ecuación característica para trazar el lugar de las raíces es

$$1 + GH = 0 \quad \Rightarrow \quad 1 + \frac{K}{s(1+s\tau_m)(1+s\tau_a)} = 0$$

siendo $K = AK_mK_p$ la constante que varía para generar los polos en lazo cerrado según sus distintos valores.

Los polos en lazo abierto están en $s = 0$, $s = -1/\tau_m$ y $s = -1/\tau_a$. Por ser la constante de tiempo del amplificador, τ_a , de tipo eléctrico, menor que la constante τ_m relacionada con aspectos mecánicos, se cumple que $|\frac{1}{\tau_a}| > |\frac{1}{\tau_m}|$ con lo que el polo del amplificador está más lejos del origen. El lugar de las raíces tendrá tres ramas, que comenzarán en cada polo y acabarán en el infinito, pues no hay ceros. Se puede comprobar que hay tres asíntotas, que forman con el eje real ángulos de 0 , $\frac{\pi}{3}$ y $-\frac{\pi}{3}$, y que cortan a este eje en el punto $\sigma = -\frac{\tau_a + \tau_m}{3\tau_a\tau_m}$. El lugar de las raíces se puede ver en la figura 5.2

Figura 5.2: Lugar de las raíces con $K_g = 0$

Para unos valores dados de τ_a y τ_m , deberíamos escoger la K (modificable cambiando la ganancia A del amplificador, y la constante de realimentación K_p) de modo que los

polos sean reales, es decir, de modo que el sistema en lazo cerrado sea sobreamortiguado. Lo idóneo sería tener los polos muy cerca (o incluso sobre) el punto de ruptura (sistema críticamente amortiguado), pues de este modo se llegaría al estado estacionario lo más rápidamente posible, pero sin oscilar. Hay, en cualquier caso, un valor de K que no debe excederse, pues el sistema se volvería inestable (las ramas del lugar de las raíces entran en el semiplano derecho).

La elección de la K (via A o K_p) plantea un problema importante: depende de la K_m , la cual no es realmente constante (véanse las ecuaciones para el motor eléctrico en el tema 4) sino que depende del momento de inercia de la carga que mueve el motor, y por tanto, de en qué ángulo se encuentre éste (no es lo mismo mover un brazo robot contraído que extendido). Una solución sería buscar una K_p o A diferentes para cada posición angular.

Veamos ahora el caso en que se aplica realimentación en velocidad, y por tanto, K_g no es nula. El diagrama de bloques de la figura principal se puede transformar según se muestra en la figura 5.3

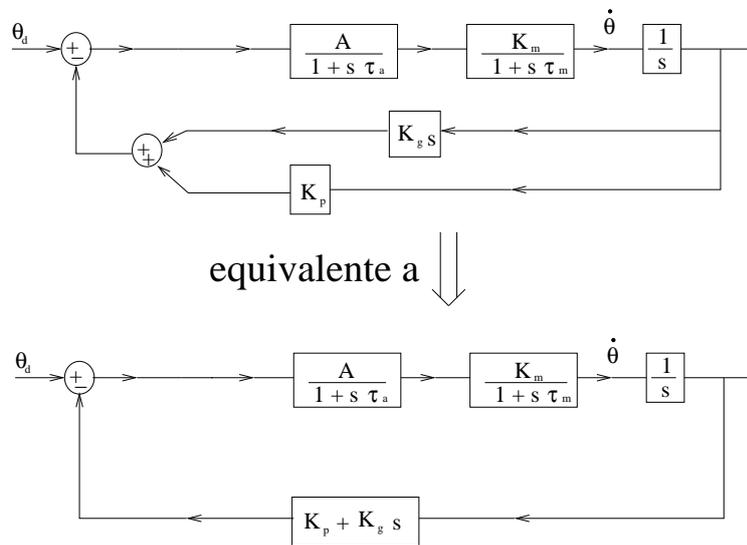


Figura 5.3: Control de un motor CC: realim. en velocidad

En este caso la ecuación característica que servirá para dibujar el lugar de las raíces queda:

$$1 + GH = 0 \implies 1 + \frac{AK_m \cdot (K_p + K_g s)}{s(1 + s\tau_m)(1 + s\tau_a)}$$

GH tiene un cero en $s = -\frac{K_p}{K_g}$, y los mismos polos que el caso anterior. En este caso hay dos asíntotas, que forman con el eje real ángulos de $\pm \frac{\pi}{2}$ y lo cortan en el punto

$$\sigma = \frac{1}{2} \left(\frac{K_p}{K_g} - \frac{1}{\tau_m} - \frac{1}{\tau_a} \right)$$

Los puntos de ruptura resultan de la resolución de una ecuación de tercer grado, y de los tres, sólo uno es real y pertenece al L.R. En este caso, hay dos posibles lugares de las raíces según cuáles sean los valores de las constantes que intervienen en el sistema. Un caso se da cuando el cero está más cerca del origen que el primer polo, y el otro cuando está entre los dos polos. Ambos se muestran en la figura 5.4

A la vista de los lugares de las raíces, una ventaja obvia es que ya no hay inestabilidad, independientemente del valor de las constantes K_p , K_g u otras. Además, si se elige la segunda posibilidad (buscando los valores de K_p y K_g de modo que $\frac{K_p}{K_g} > \frac{1}{\tau_m}$) puede hacerse que los polos en 0 y $-\frac{1}{\tau_m}$ sean dominantes frente al del amplificador, que aparte de estar más lejos

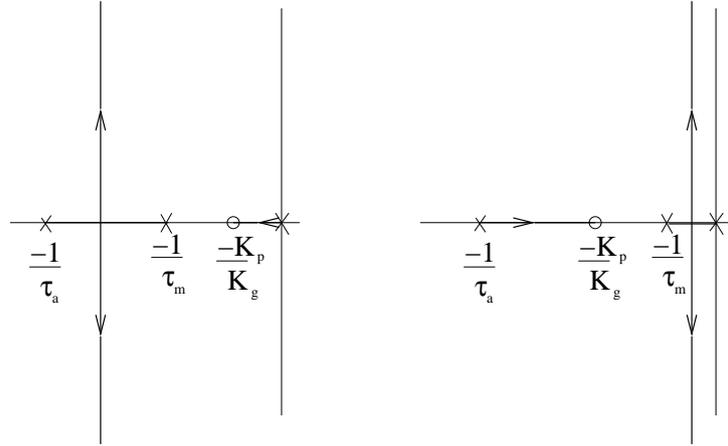


Figura 5.4: Lugar de las raíces con $K_g \neq 0$

del origen, queda parcialmente compensado por la proximidad del cero. De este modo el sistema total, incluyendo al amplificador, se puede aproximar como un sistema de segundo orden.

Otra forma también usada habitualmente de aproximar el motor realimentado en velocidad como un sistema de segundo orden es la que se aplica cuando se considera que el amplificador tiene una respuesta mucho más rápida que el motor, su polo, $-1/\tau_a$, está tan a la izquierda que puede ser despreciado, con lo que su función de transferencia queda reducida simplemente a la constante A . En ese caso, y volviendo a la figura principal (figura 5.1), la f.d.t. del lazo más interno (ya cerrado) quedaría

$$G_{in}(s) = \frac{\frac{AK_m}{1+s\tau_m}}{1 + K_g \frac{AK_m}{1+s\tau_m}} = \frac{AK_m}{\tau_m s + AK_g K_m + 1}$$

y la f.d.t. total resultaría

$$G(s) = \frac{\frac{AK_m}{\tau_m s + AK_g K_m + 1} \frac{1}{s}}{1 + \frac{AK_m}{\tau_m s + AK_g K_m + 1} \frac{1}{s} K_p} = \frac{1}{K_p} \frac{\frac{AK_m K_p}{\tau_m}}{s^2 + \frac{AK_g K_m + 1}{\tau_m} s + \frac{AK_p K_m}{\tau_m}} = \frac{1}{K_p} \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

donde se observa que este sistema puede escribirse en la forma del sistema standard de segundo orden, donde

$$\omega_n = \sqrt{\frac{AK_m K_p}{\tau_m}} \quad \zeta = \frac{1}{2} \frac{1 + AK_m K_g}{\sqrt{\tau_m AK_p K_m}} \tag{5.1}$$

Podemos variar K_p y K_g , y variar con ello ζ , dando al sistema el carácter sobreamortiguado que deseamos, pues el amortiguamiento es directamente proporcional a K_g e inversamente proporcional a la raíz de K_p .

Para finalizar, veamos el caso más general no simplificado de control de un motor de CC, en el que no se considera que los torques gravitacionales ni de las fuerzas externas sean nulos. Aquí ya no se puede simplificar el motor considerándolo un sistema de primer orden, sino que debe ser tomado como un sistema con dos entradas, según se vio en el tema 5. Recuérdese que el diagrama de bloques era (figura 5.5).

Si introducimos este diagrama dentro del diagrama de bloques principal del control realimentado tenemos el resultado que muestra la figura 5.6

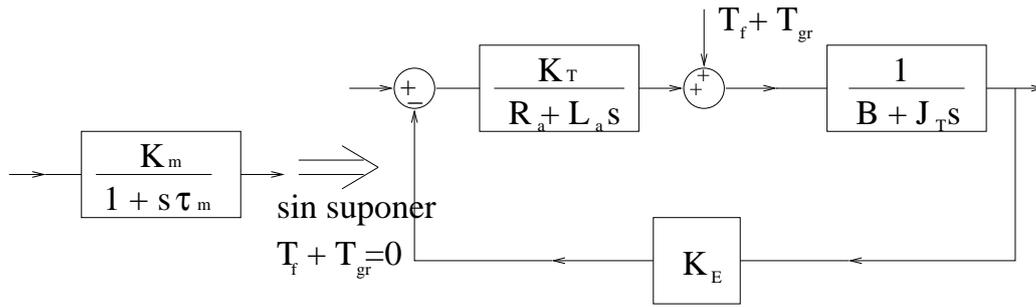


Figura 5.5: Diagrama del motor CC no simplificado

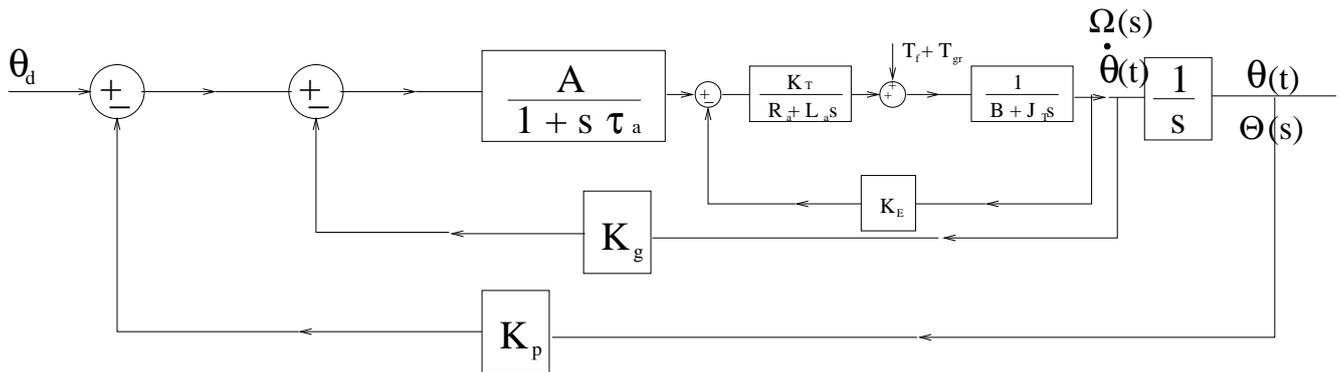


Figura 5.6: Diagrama completo

Del diagrama de bloques se ve que

$$\Theta(s) = \frac{1}{s}\Omega(s)$$

siendo $\Theta(s) = \mathcal{L}\{\theta(t)\}$ y $\Omega(s) = \mathcal{L}\{\omega(t)\} = \mathcal{L}\left\{\frac{d\theta(t)}{dt}\right\}$; además,

$$\Omega(s) = \frac{1}{B + J_T s} \left[\frac{K_T}{R_a + sL_a} \left\{ -K_E \Omega(s) + \frac{A}{s\tau_a + 1} (-K_g \Omega(s) - K_p \Theta(s) + \theta_d(s)) \right\} - (T_f(s) + T_{gr}(s)) \right]$$

de donde se obtiene

$$\Theta(s) = G_{eq}(s) \left\{ \theta_d(s) - \frac{[T_f(s) + T_{gr}(s)](s\tau_a + 1)(R_a + sL_a)}{AK_T} \right\} \quad (5.2)$$

donde se denota

$$G_{eq}(s) = \frac{AK_T}{s(s\tau_a + 1) [K_E K_T + (sJ_T + B)(R_a + sL_a)] + AK_T(sK_g + K_p)}$$

Para estudiar este sistema, que tiene dos entradas, aplicaremos el principio de superposición, que sabemos se cumple en todo sistema lineal. La forma de hacerlo es suponer sucesivamente nulas cada una de las entradas, y obtener en cada caso la salida ante la otra entrada.

Supongamos primero que $T_f + T_{gr}$ es nula, y apliquemos a la entrada θ_d un escalón unitario de valor θ_{d0} . Entonces, aplicando la ecuación 5.2, se tiene que

$$\Theta(s) = G_{eq}(s) \cdot \frac{\theta_{d0}}{s}$$

Para obtener su valor final, aplicamos el teorema del mismo nombre:

$$\lim_{t \rightarrow \infty} \theta(t) = \lim_{s \rightarrow 0} s\Theta(s) = \lim_{s \rightarrow 0} sG_{eq}(s) \frac{\theta_{d0}}{s} = \frac{AK_T}{AK_T K_p} \theta_{d0} = \frac{\theta_{d0}}{K_p}$$

Esto significa que si queremos alcanzar un cierto valor final, el comando de control que en realidad hay que enviar es el valor deseado, dividido entre K_p (pero recuérdese que K_p era variable a voluntad).

Ahora, supongamos que θ_d es nula (lo cual no resta generalidad, pues el origen de ángulos puede fijarse arbitrariamente), pero que hay unos ciertos torques externos, cuya transformada de Laplace denotaremos como $T_L(s) = T_f(s) + T_{gr}(s)$, y introduzcamos al sistema una entrada escalón ($T_L(s) = T_{L0}/s$, una carga asida de repente). En este caso, la ecuación 5.2 nos sirve de nuevo para calcular el valor final de la posición angular:

$$\lim_{t \rightarrow \infty} \theta(t) = \lim_{s \rightarrow 0} s\Theta(s) = \lim_{s \rightarrow 0} sG_{eq}(s) \frac{(-T_{L0})(1 + s\tau_a)(R_a + sL_a)}{AK_T s} = -\frac{R_a T_{L0}}{AK_T K_p}$$

Hay, pues, error de posición (la posición debía haber sido 0, según pedíamos). A este error se le suele llamar histéresis. Se reduce aumentando la K_p (dado que el resto son constantes que no controlamos, excepto quizá la A del amplificador). No obstante, aumentar K_p tiene el problema de hacer el sistema subamortiguado (recuérdese el lugar de las raíces). Hay que evitar esto usando un valor alto para K_g .

Una forma alternativa, y ampliamente usada, de conseguir el comportamiento deseado es anteponer un controlador PID al motor. Un posible esquema sería el mostrado en la figura 5.7

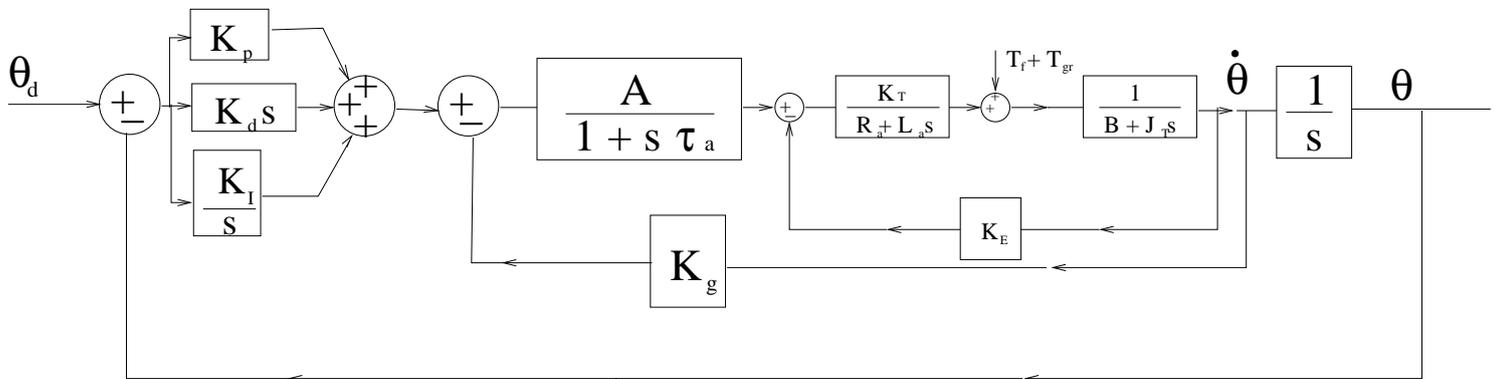


Figura 5.7: Control PID de un motor CC

Posibles respuestas del motor ante entrada escalón para varios tipos de control derivados del PID pueden verse en la figura 5.8.

En la gráfica (a) (control proporcional) se observa como se puede eliminar la sobreoscilación disminuyendo K_p , pero a costa de aumentar el error de posición. En la (b) (control PD) sigue habiendo error de posición, pero, a igualdad de dicho error (es decir, para la misma K_p) el sistema oscila menos. En el caso mostrado en (c) (control PI) el integrador libre del controlador anula el error de posición, pero no evita la sobreoscilación. La mejor alternativa (caso (d)) es el uso de un controlador PID, con los parámetros apropiadamente ajustados (respuesta en línea continua) para la cual se alcanza la posición deseada sin sobreoscilación, y en el mínimo tiempo posible. Aun cuando existen métodos apropiados para el diseño de PIDs, obtener esta combinación no es sencillo, y en realidad, no existe un único

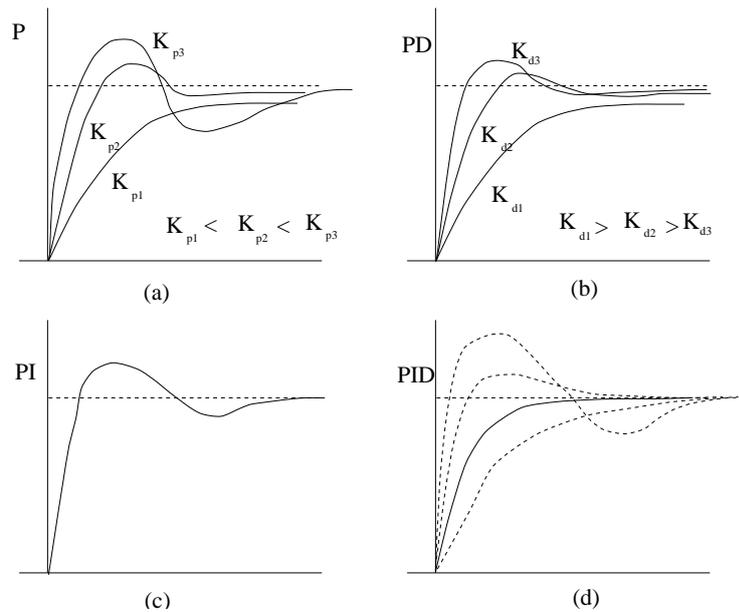


Figura 5.8: Respuesta a escalón de un motor CC

conjunto de valores válido, debido a que, como se comentó antes, la inercia de la carga fija al motor varía, por lo que sería necesario diseñar un controlador diferente para cada posición de partida y llegada. Esto hace que se suelen escoger parámetros "conservadores" (es decir, que tiendan más hacia el sobreamortiguamiento) aun cuando esto haga el funcionamiento del sistema más lento de lo que sería posible.

5.2. Control de una articulación

Supongamos que las estrategias de control descritas deben aplicarse a una articulación concreta de un brazo robot. En este caso, la señal de referencia, θ_d , que indica en qué posición debe colocarse el motor es siempre digital, pues viene del ordenador de control (en concreto, del programa generador de trayectorias). Las señales de los sensores, sin embargo, pueden ser analógicas o digitales. Lo normal es que el ordenador de control compute lo más rápido posible las señales de referencia para las articulaciones, y las envíe en secuencia a cada una de ellas. En cada articulación hay un pequeño microprocesador o microcontrolador que ejecuta el algoritmo de control (el PID, u otro, que se la haya programado, normalmente en firmware); en ocasiones un mismo microcontrolador se ocupa de dos o más articulaciones.

Los datos de posición y velocidad presente se pueden obtener de varios modos. Veamos los posibles casos:

+Posición digital y velocidad analógica con sensores diferentes: se usan un codificador óptico para medir la posición y un tacómetro para medir la velocidad. (figura 5.9. Como la señal de control es en ocasiones proporcional al error en la velocidad (si se usa, p. ej., un PD) se podría evitar el uso del conversor A/D del tacómetro, según se muestra en la parte inferior de la figura 5.9. Esto reduce el coste, porque el restador analógico, hecho con un operacional, es más barato que el conversor A/D.

+Posición digital y velocidad digital: Aquí sólo se usaría un codificador óptico que generaría una señal digital para la posición, y la velocidad se obtendría numéricamente a partir de ésta. Es la solución más comunmente aceptada hoy, pues los tacómetros son caros y pesados (a veces, tanto como el propio motor). El circuito digital apropiado procesa las

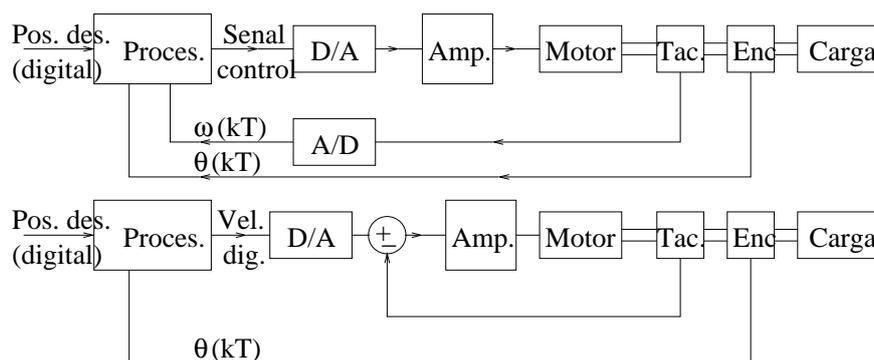


Figura 5.9: Control de una articulación

señales del encoder y la velocidad se obtiene midiendo a intervalos regulares de tiempo como

$$\omega(kT) \simeq \frac{\Delta\theta}{\Delta T}$$

Si, como suele ser habitual, ΔT es constante, no hace falta dividir. No obstante, esto presenta un grave problema: la velocidad así obtenida está cuantizada, por ser diferencia de valores enteros (el número de pulsos en instantes de medición consecutivos), y debe ser almacenada en un registro de longitud finita. Esto hace que presente oscilaciones, especialmente para valores pequeños de la velocidad (cuando el motor está parando). Este fenómeno, llamado "digital jitter" (oscilación digital) se aprecia en la figura 5.10.

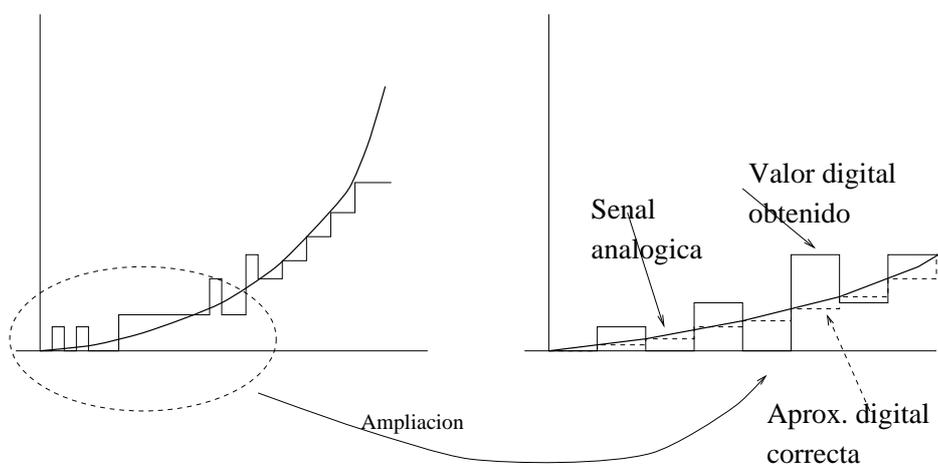


Figura 5.10: Oscilación digital provocada por la cuantización

Como ilustración, imaginemos que a la máxima velocidad tenemos 100 cuentas por periodo de muestreo; deberemos usar un registro de 7 bits. Pero en las fases de aceleración y deceleración (baja velocidad) hay un cambio muy pequeño en el número de cuentas entre dos periodos consecutivos. De hecho, si el periodo de muestreo es pequeño, podemos incluso estar leyendo la misma posición cuantizada del encoder (el ángulo recorrido en el tiempo ΔT es menor que la precisión). Así, la señal de velocidad sería una sucesión de ceros y unos. Esto puede incluso provocar inestabilidad en el sistema de control. Hay varias soluciones, pero ninguna realmente apropiada: usar encoders de más resolución (es caro), reducir la constante de realimentación, K_g , para hacer el sistema más amortiguado y evitar que las oscilaciones le afecten demasiado (enlentece la respuesta), obtener la velocidad con una aproximación de segundo orden o superior (conlleva más cálculo), usar un periodo de

muestreo variable con la velocidad (complica mucho la circuitería y el diseño del sistema de control), o usar varios valores diferentes de K_g según la velocidad medida. Esto es lo más factible; el valor de K_g debería ser grande en aceleraciones y deceleraciones, y más pequeño el resto del tiempo.

+Otros casos:

-Uso de un sensor analógico de velocidad y medida de la posición por integración analógica: se usaría un tacómetro y un integrador hecho con un operacional a la salida. Presenta dos problemas: la integración no es ideal, y la deriva térmica hace que el offset del operacional varíe con el tiempo. Se pierde precisión y repetibilidad. Además, no es completamente lineal, y requiere conversores A/D, puesto que hoy día todo el control es digital.

-Uso de un sensor de posición analógico y medida de la velocidad por derivación: Se usaría, p. ej, un potenciómetro y un derivador analógico a la salida. Tiene el problema de la poca precisión del potenciómetro, y de que la derivación no es ideal, y además amplifica mucho el ruido, especialmente de alta frecuencia.

+Nota final: Recuérdese que estamos controlando la posición de los motores, no la del efector final. Esto significa que nos fiamos de la rigidez del sistema mecánico y de la fiabilidad de las transmisiones mecánicas (engranajes, correas o cadenas). Esto suele ser razonable si tenemos en cuenta que los sensores (normalmente, encoders) se suelen poner acoplados al eje del motor antes de desmultiplicar el movimiento, de modo que una vuelta (N cuentas) es sólo una fracción pequeña de vuelta en el efector final.

5.3. Nota al control adaptativo

Hemos supuesto en todo momento que la carga que el brazo movía, y sus inercias, eran constantes. Esto no es cierto. De hecho, el conjunto de parámetros de control (K_p, K_g o bien K_p, K_I, K_d) diseñados para movimientos alrededor de cierta posición pueden no ser válidos para otra. Lo que se suele hacer es tomar parámetros muy "conservadores", en el sentido de que el punto de funcionamiento (los polos del sistema) estén muy lejos de la inestabilidad; pero esto provoca un fuerte amortiguamiento, y el sistema es lento.

Para evitarlo, se trataría de que los parámetros del controlador dependieran de la posición presente, de la carga (a través del error de posición), y de la señal de control. El esquema sería el mostrado en la figura 5.11

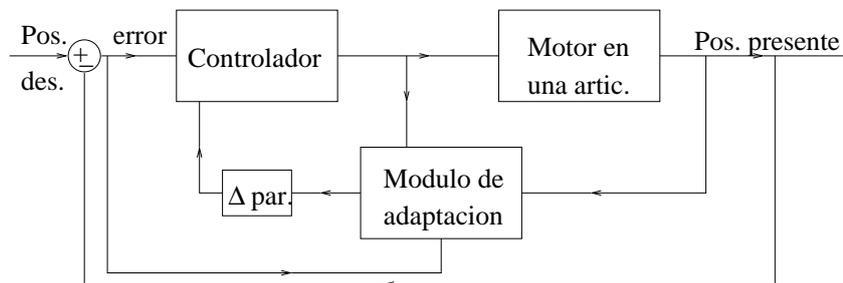


Figura 5.11: Esquema de control adaptativo

El módulo de adaptación que se muestra tendría dos partes: un medidor de cuán bien (o mal) está actuando el controlador, definiendo un índice de eficiencia, y otro módulo que genere cambios en los parámetros del controlador, los cuales dependerían de la posición, del error en la posición y de la señal de control, de tal modo que se consiguiera minimizar el índice de eficiencia. Este módulo puede ser una red neuronal, un sistema basado en reglas que use lógica difusa, u otro similar.

Tema 6: Programación de robots

6.1. Introducción

Hasta ahora hemos visto cómo se modeliza un robot mediante su cinemática directa e inversa; cómo se obtienen caminos en el espacio para el punto terminal que cumplan ciertas condiciones de velocidad, aceleración, etc; qué sensores se usan, y qué señales dan, qué actuadores mueven al robot, y qué señales hay que enviarles. Es el momento de ensamblar todas estas piezas para conseguir que el robot realice tareas útiles. Obviamente, la conexión entre todas ellas está en el software, es decir, en los programas y sistemas operativos que, corriendo sobre uno o más ordenadores o microcontroladores, coordinarán todo el proceso.

Como veremos seguidamente, la programación de robots es sustancialmente diferente a la de ordenadores aislados, aun cuando un programa escrito en un lenguaje para robots semeje a cualquier otro programa convencional. El hecho de estar controlando sistemas reales añade complejidad debido a un cierto grado de impredecibilidad que deberemos asumir y manejar.

Por supuesto, cualquiera de las tareas antedichas podría, en principio, programarse en cualquier lenguaje que pudiese acceder al hardware, pero programar una tarea normal en términos de estas tareas simples sería horriblemente complejo. Veamos un ejemplo: algo aparentemente simple, como pintar al spray un coche. Habría que detallar todos los puntos por los que se debe pasar en función del tiempo, calcular su cinemática inversa, especificar las posiciones, velocidades y aceleraciones de cada articulación como función del tiempo, y programar los parámetros y lazos de realimentación del sistema de control. Además, atender al mismo tiempo a las señales de los sensores de cada articulación y procesarlas, más las posibles señales de interrupción provocadas por sensores externos, si los hay.

Esto explica que sean necesarios lenguajes para especificar las tareas en términos accesibles a los humanos. Estos lenguajes serán usados para dos fines:

- Definir la tarea que el robot tiene que realizar.
- Controlar al robot mientras la realiza.

De lo dicho se entiende que un lenguaje que pretenda ser usado para programar robots deberá ser, en algún sentido, diferente a los lenguajes convencionales. Esas diferencias son debidas a que:

- El entorno del robot no puede describirse sólo en términos cuantitativos; usa relaciones espaciales (arriba de, abajo de, sobre, . . .) e incluso temporales, o de causalidad (A no puede ponerse sobre B hasta que B no esté en X, . . .)
- El robot opera en un mundo real impreciso, y el modelo del mundo que el programa maneja puede no dar cuenta debida de él, precisamente a causa de esa imprecisión.

Además, hay que tener en cuenta que los programas de robot deben atender a señales sensoriales que se producen en instantes impredecibles, y por tanto, deben estar preparados para interrumpirse en cualquier momento. Por otra parte, y como vimos en el tema 2, las posiciones y orientaciones de robot y objetos son adecuadamente descritas mediante transformaciones homogéneas; cualquier lenguaje de robots debería incorporar un fácil manejo de éstas.

6.2. Requerimientos de los lenguajes de programación de robots

Veamos ahora en detalle qué consideraciones deberán tenerse en cuenta al diseñar un lenguaje (o modificar uno existente) para programar robots o sistemas robotizados en general.

- El lenguaje deberá incorporar **estructuras de datos** apropiadas para manejar posiciones, orientaciones, y transformaciones espaciales. En particular, las matrices de transformación homogénea deberán estar presentes, así como las operaciones de rotación y traslación sobre ellas, equivalentes al cambio de sistema de referencia. Por otra parte, deberán poder especificarse de modo simple **órdenes de movimiento**, preferentemente en el espacio cartesiano, y con posibilidad de elegir la forma de la trayectoria del punto terminal entre la posición actual y la deseada. Esto exige el conocimiento de la cinemática inversa, y por tanto, si el lenguaje debe adaptarse a varios manipuladores, también deberá haber formas de especificar la cinemática de un manipulador cualquiera. Finalmente, se deberá conocer la posición actual en todo momento, para continuar desde ella una orden de movimiento anormalmente interrumpida.
- El lenguaje deberá permitir alguna forma de **paralelismo**, puesto que al menos dos niveles distintos de él se dan en todo robot: el control simultáneo de todas sus articulaciones, y el funcionamiento concurrente quizá con otros brazos, y siempre con sus propios sensores externos que pueden mandar señales en cualquier momento. Las formas de implantar este paralelismo son, o bien con varios elementos de computación (caso común en las articulaciones, donde suele haber un microcontrolador por cada dos o tres de ellas), bien con software que simule el paralelismo mediante la compartición del tiempo del único procesador; este software es un sistema operativo multitarea, o un lenguaje que permita concurrencia.
- Si varios procesos (en general, tareas) deben funcionar simultáneamente, el lenguaje deberá permitir la **comunicación** entre ellos; las tres formas más comunes para esto son el uso de memoria compartida, la llamada remota a procedimientos, y el paso de mensajes. Cada uno de éstos presenta ventajas e inconvenientes.
- Al actuar el robot en el mundo real, la activación de mandatos para realizar acciones, e incluso la secuencia en que éstos se ejecuten, depende de que determinados eventos reales hayan sucedido ya, o no (o nunca lleguen a ocurrir). Por ello, es necesario que el lenguaje provea mecanismos de **sincronización** de eventos, que detengan o alteren la ejecución normal en tanto ciertos hechos (por ejemplo, la activación de un sensor) no sucedan. Además, otras acciones deben ser ejecutadas en respuesta a señales de error (p.ej., un sensor de choque que se activa). Por ello, se suelen sincronizar las acciones para atender a cuatro tipos de eventos: activación (iniciar la acción al recibir cierta señal), terminación (cese de la acción ante la señal), error (inicio de acción de seguridad o recuperación ante señal de error) y anulación (cese de la acción al NO recibir señal del terminación después de un tiempo prudencial).
- Determinadas condiciones (sobre todo, sobre los valores de los sensores) deben ser comprobadas constantemente (o al menos, periódicamente). Esto exige que haya mecanismos de **comprobación** de eventos, que pueden ser procesos que corren como un bucle infinito que monitoriza, o interrupciones hardware. Asociado a esto deberá

haber un mecanismo de prioridades, para decidir qué se continúa monitorizando en caso de activación simultánea¹ de varios eventos.

- El lenguaje deberá proveer estructuras y acceso a **variables sensoriales**, es decir, que contienen valores de las señales recogidas por los sensores, y que, a diferencia de las variables convencionales, no se inicializan explícitamente en el programa, y su alcance es siempre global. Se puede dar una actualización continua de algunas o todas estas variables, o bien a intervalos discretos de tiempo. Además, en el caso de que varios procesos (generalmente, todos ellos relacionados con los sensores) puedan actualizar la misma variable, deberán considerarse prioridades entre ellos.
- Deberán estar presentes mecanismos para especificar las acciones de **inicialización y terminación**. Típicamente, al comenzar un programa de robot puede ser necesario el calibrado del mismo (ver tema 3) y su posicionamiento en un lugar concreto, así como un auto-test del hardware, si es posible. Del mismo modo, al acabar también se puede llevar al robot a algún lugar conocido, y luego se suele anular la ganancia de todos los amplificadores de los motores, para evitar el movimiento accidental.

Se verá después cómo algunos de estos requerimientos, pero no todos, se realizan en lenguajes como AL.

6.3. Sistemas operativos

Es necesario hacer un breve comentario sobre qué características debiera tener el sistema operativo que soporte un lenguaje con los requerimientos especificados en el apartado anterior. Es preciso decidir si todos estos requerimientos quedan al lenguaje, o si algunos de ellos son dejados al sistema. Generalmente, esta decisión depende de la velocidad del computador usado; si es suficientemente rápido como para soportar un sistema y un lenguaje manteniendo los requerimientos de respuesta a los eventos externos en tiempo acotado (y suficientemente corto), entonces esta solución vale la pena, pues simplifica enormemente la programación. En otro caso, deberá usarse un lenguaje que mantenga varios procesos con el tiempo de conmutación entre ellos más breve posible.

La característica más importante del sistema operativo que debe verificarse es, en cualquier caso, el que sea de tiempo real (es decir, responda a cualquier petición en un tiempo acotado, y no pueda quedar bloqueado). El ejemplo más típico de esto es la necesidad de mantener los lazos cerrados de realimentación de los controladores de las articulaciones, lo que exige un muestreo de los sensores con un intervalo periódico T , y el envío de la acción de control calculada con ese mismo intervalo. Ignorar las señales durante más tiempo provocaría inestabilidad, y el consiguiente movimiento descontrolado del robot. El usar un microcontrolador dedicado a esa articulación alivia el problema, pero no lo resuelve, puesto que, p. ej. el mantenimiento de una trayectoria rectilínea exige el envío de posiciones de referencia en el espacio de articulación también en instantes específicos.

La conclusión de esto es que no todo estriba en mejorar la velocidad del hardware, y que un diseño cuidadoso de cada proceso para garantizar su terminación en un tiempo acotado es imprescindible.

¹Simultánea en este contexto significa suficientemente próxima en el tiempo, es decir, el tratamiento dado a un evento no ha concluido aún cuando el siguiente se produce

6.4. Clasificación de los lenguajes de prog. de robots

Aunque ningún lenguaje de programación hoy día cumple todos los requerimientos especificados anteriormente, hay varios que resultan útiles, dependiendo de la tarea específica a que se les destine.

Veamos la división y algunos lenguajes principales. En primer lugar, estableceremos una clasificación por la sintáxis y complejidad. Aquí se distinguen tres tipos:

- Secuenciadores de instrucciones: simplemente almacenan y posteriormente repiten una secuencia de posiciones y acciones (apertura/cierre de la pinza, etc.) en un orden más o menos fijo. Tales posiciones y acciones se aprenden, de varias maneras:
 - Mediante movimiento del robot con un joystick, ratón o teclado especial suspendido del techo ("*teach pendant*")
 - Mediante movimiento manual y almacenamiento de las posiciones de los *encoders*. En este caso, se pueden ejecutar los movimientos a diferente velocidad de la que se almacenaron.

Normalmente, estos lenguajes incorporan también órdenes especiales para retraso (dejar que concluya la ejecución de un movimiento) o para especificación de las acciones requeridas ante señales sensoriales específicas (errores, etc.) o ausencia de ellas. La facilidad de "programar un robot de este modo hace común el uso de secuenciadores en entornos industriales poco flexibles (cadenas de montaje de automóviles, etc.).

- Extensiones a lenguajes clásicos: son módulos específicos para el manejo de sensores y actuadores, más estructuras de datos adaptadas (matrices homogéneas, etc.) conservando la sintáxis general y control de flujo del lenguaje escogido. Se pueden basar en BASIC, PASCAL, C, etc. Por las razones expuestas en el apartado anterior, es necesario que estos lenguajes corran sobre un S.O. de tiempo real.
- Lenguajes específicos para robots: fueron diseñados por firmas comerciales (salvo uno de la Univ. de Stanford) para ser vendidos junto con sus manipuladores, teniendo en cuenta los sensores y actuadores a que se debían conectar. Todos ellos incorporan el manejo de las señales de los sensores, y de acuerdo a sus valores pueden cambiar en tiempo real el flujo del programa. Además, incorporan descripción y razonamiento en términos geométricos, e interfaces a sistemas CAD/CAM.

Por otra parte, y respecto al nivel de abstracción que permiten a la hora de especificar la tarea, podemos clasificar estos lenguajes en:

- Orientados al robot: sus primitivas son comandos de movimiento para el robot, o peticiones de lectura de sensores. Se deja al usuario la tarea de establecer explícitamente, y de modo secuencial, cuáles deben ser los movimientos a ejecutar, así como cuál será el comportamiento según los valores de los sensores en cada instante. El nivel de especificación es, sin embargo, suficientemente alto, como veremos en los ejemplos, como para que una tarea simple se pueda programar con rapidez.
- Orientados a la tarea: permiten al usuario especificar qué debe hacer el robot, pero no necesariamente cómo debe hacerlo, al menos, hasta cierto punto. Las especificaciones de tarea se pueden dar en forma textual, o ayudándose de un interface gráfico (simulador del mundo del robot).

6.5. Niveles de programación. Cuadro resumen

Es conveniente ahora dar y comentar un esquema que explique cuál es el software que genéricamente se emplea en la programación y uso de un sistema robotizado, cuáles son las relaciones entre sus partes y a qué nivel se emplea cada una. Lo que sigue explica la figura 6.1. En ella encontramos, de abajo a arriba:

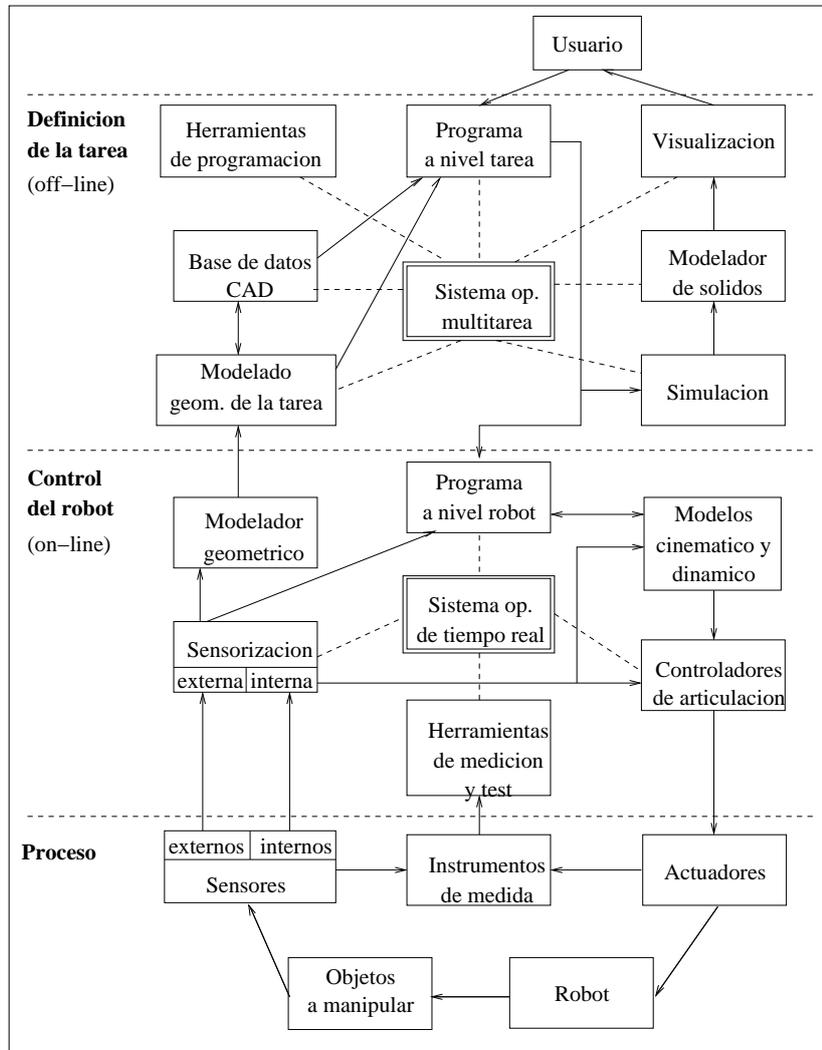


Figura 6.1: Cuadro resumen del software de un sistema robotizado

El **nivel del proceso**: esto no es software, sino hardware actuando sobre el mundo físico (los objetos). Los sensores y actuadores típicos ya fueron comentados en los temas correspondientes. Respecto a los instrumentos de medida externos al robot, que pueden estar presentes o no, se usan en su caso para tareas de monitorización externa, esencialmente para comprobar si el robot realiza su tarea correctamente, y si las condiciones son apropiadas. Pueden ser sensores de temperatura, corriente de los circuitos de alimentación, etc., así como herramientas de verificación y control de calidad, que detecten defectos en la pieza manufacturada.

El **control del robot**: aquí aparecen las herramientas de medición y test (sólo si existían los instrumentos de medida) que procesan sus señales y mandan los oportunos mensajes al usuario en caso de encontrar algo anormal. Además, tenemos los procesos básicos que

tratan la señal en bruto (*raw-data*) de los sensores, y dirigen sus resultados bien al sistema de control, para cerrar los lazos de realimentación de posición/velocidad (sensores internos) bien al programa a nivel robot, que, como hemos dicho, puede necesitar comprobar la señal de sensores externos (p. ej., un microinterruptor que avise del contacto con un objeto), bien a un módulo de modelado geométrico que genere las primitivas geométricas necesarias para el reconocimiento de los objetos de la escena. A este nivel de control de robot se hallan también los controladores de cada articulación, que usualmente son programas escritos en C o ensamblador que corren en microcontroladores dedicados. Por encima, pero aún actuando *on-line* (es decir, al mismo tiempo que se ejecuta la tarea) tenemos el modelo cinemático y dinámico del robot, que suministra las señales de referencia a los bucles de control para conseguir de este modo las trayectorias, velocidades y aceleraciones requeridas. Y directamente conectado a él, el programa a nivel de robot, que como dijimos, está compuesto de la secuencia de acciones (posiblemente variable en función de condiciones externas) que el robot realizará. Todos estos módulos están bajo la supervisión y control de un sistema operativo de tiempo real. Las razones para que tenga que ser así ya han sido suficientemente comentadas antes, excepto en el caso del modelador geométrico: si es necesario que opere en tiempo real es porque los objetos del mundo pueden estar moviéndose, y las trayectorias deben variar en consonancia. Pero si estuviésemos dispuestos a admitir un entorno estático durante todo el desarrollo de la tarea, como hace p. ej. Handey, este módulo podría no operar *on-line*. En cualquier caso, por el momento es irrelevante, puesto que no existe un módulo efectivo de tal tipo, ni siquiera para ambientes industriales genéricos y funcionando *off-line*.

La **definición de la tarea**: En este nivel encontramos dos ramas: una que conecta el modelador geométrico de la tarea, la base de datos de modelos geométricos, y el programa a nivel tarea, y otra, que puede existir, o no, y que hace una simulación del proceso real, bien concurrentemente con él, bien previamente, para depurar errores. La primera rama determina qué objetos están presentes en el mundo del robot, así como su posición y orientación. A partir de este conocimiento, el planificador de tareas incluido en el lenguaje obtiene una secuencia de acciones que deben realizarse teniendo en cuenta cuáles son los objetos y dónde están, para evitar así colisiones. La rama de simulación requiere técnicas de informática gráfica para el modelado y representación de objetos en 3D. Finalmente, el cuadro de herramientas de programación representa los editores, compiladores, y similares que se usan de modo normal. Todos estos módulos no necesariamente deben funcionar en tiempo real, y sincronizados con la ejecución de la tarea, de modo que un sistema operativo multitarea, pero no de tiempo real (p. ej., Unix) puede gestionarlos.

6.6. Lenguajes orientados al robot

6.6.1. Movimientos del robot

Como dijimos, estos lenguajes generan órdenes de movimiento para el robot. Por ello, para entenderlos correctamente es necesario ver cuáles son los tipos fundamentales de movimientos que un manipulador robótico necesita ejecutar. En un entorno con varios objetos, el brazo debe poder moverse, asir y desplazar cualquiera de ellos sin chocar con nada. Antes de asir un objeto debe acercarse cuidadosamente a él, y a la inversa cuando se le deposita. Si se trata de insertarlo en un lugar estrecho, o de la forma justa para que el objeto encaje, ciertas restricciones sobre la fuerza y dirección de movimiento deberán cumplirse. Por ello, los movimientos pueden ser:

- Movimientos en espacio libre (*free motions*): el brazo se mueve sin limitaciones suficientemente lejos de los obstáculos.
- Movimientos condicionados (*guarded motions*): el brazo se mueve comprobando simultáneamente alguna condición que se verificará de modo instantáneo en cierto momento; cuando lo hace, el movimiento debe cesar. El depósito de una pieza descendiendo sobre la mesa hasta que el sensor de fuerza vertical detecta resistencia (la reacción del plano de la mesa) es un ejemplo de esto.
- Movimientos con restricciones (*compliant motions*): el brazo se mueve comprobando una o más condiciones que deben satisfacerse en todo momento; si no lo hacen, entonces o bien el movimiento se interrumpe, o bien se ajusta su dirección o fuerza para que se sigan cumpliendo. La inserción de un pivote en un orificio manteniendo una fuerza de rozamiento constante con las paredes es un ejemplo típico.
- Movimientos de las partes sin ser capturadas (*constrained motions*): consisten en empujar con la pinza algún objeto sucesivamente, de tal modo que, usando su rozamiento con el suelo, se consiga situarlo en un intervalo de posición y orientación pequeño (es decir, reducir su incertidumbre) de modo que la pinza lo capture luego apropiadamente.

La mayoría de los lenguajes orientados a robot (especialmente, los de más alto nivel) incorporan órdenes de movimiento de los tres primeros tipos. El cuarto debe ser programado explícitamente.

6.6.2. Evolución y características

Algunos ejemplos de lenguajes de programación orientados al robot son:

En el nivel más bajo, los lenguajes de aprendizaje, que son secuenciadores de instrucciones ampliados para permitir la lectura automática de variables de posición a petición del usuario, y su modificación en modo texto, además de la incorporación de estructuras de control simples y órdenes de comprobación de sensores que son intercalables mediante interfaces con menú en la secuencia de movimientos aprendida. Un ejemplo de este tipo es APT, desarrollado por las fuerzas aéreas de EEUU para su incorporación en plantas de fabricación y ensamblado de componentes de aviones y que deriva de lenguajes usados antes para control de máquinas-herramienta. Evolucionó después a MCL, que incluía comprobaciones sensoriales.

En el siguiente nivel encontramos los lenguajes con estructuras amigables para la definición de posiciones/orientaciones y su manipulación, cálculo implícito de trayectorias en el espacio cartesiano (lo que permite movimientos de aproximación en cualquier dirección deseada) y comprobación continua o a petición del usuario de señales sensoriales. Entre éstos se encuentran: KAREL, basado en Pascal; SRL y AL, que fueron los primeros en introducir manejo concurrente de los eventos sensoriales. AL fue desarrollado en la Universidad de Stanford, y tiene características de Algol y de Pascal concurrente. ROBEX, otro lenguaje posterior, también maneja los sensores pero por medio de un sistema de señales y semáforos. AML, creado por IBM, y hoy ya en desuso por haber dejado de fabricarse el robot 7535 para el que se diseñó, amplía las estructuras de datos básicas para introducir el "agregado" (una estructura compuesta arbitraria que contiene los datos y métodos necesarios para ejecutar una operación elemental). Por último, VAL y VAL-II, creados por la compañía Unimation para el robot industrial más difundido, el PUMA, tiene una sintaxis similar a

BASIC, con comandos añadidos. Se vende con su propio sistema operativo que corre sólo sobre su propio ordenador de control. Tanto lenguaje como sistema fueron programados en C y en el ensamblador de sus controladores (Motorola 6502). VAL-II posee prácticamente todas las características de tiempo real y manejo de eventos antes citadas.

En el nivel más alto de los lenguajes orientados a robot están dos prototipos de investigación, LM y XPROBE. El primero, creado por J.C. Latombe, parece similar a un lenguaje de aprendizaje, pero su resultado no es una mera repetición de las acciones, sino la generación de un programa con órdenes de movimiento y variables libres, que puede luego editarse para cambiar las posiciones absolutas, de modo que el robot ejecutará la acción en otro lugar de su espacio de trabajo (cosa muy útil cuando las partes están inicialmente en posición desconocida), y se pueden también añadir las órdenes de comprobación continua o puntual de sensores. Un nivel aún más alto es el de XPROBE, en el que el usuario enseña al robot no sólo posiciones y movimientos, sino estrategias sensoriales; esto es posible porque el robot almacena en todo momento los valores de sus sensores, y extrae los conjuntos de valores que considera claves, porque en ellos el usuario ha variado o interrumpido el movimiento.

En una siguiente sección se darán ejemplos de programas escritos en algunos de éstos lenguajes que se consideran representativos.

Nótese que buena parte de la tarea tediosa que se trataba de automatizar es realizada por estos lenguajes. Concretamente:

- Calculan la cinemática inversa, puesto que el usuario especifica posiciones cartesianas (normalmente, en forma de matrices homogéneas).
- Planifican la forma de las trayectorias entre dos puntos (inicial y final) usualmente rectilíneas, y deciden las velocidades y aceleraciones apropiadas, lo cual genera unas funciones de movimiento de las articulaciones con el tiempo.
- Envían los valores pertinentes al controlador: éstos entran como valores de referencia para los lazos de control de cada variable de articulación.
- Procesan los datos de los sensores, dejándolos en una forma útil para su manejo directo por las órdenes o comprobaciones del programa.

6.6.3. Estudios de caso

AL

Como se indicó anteriormente, AL fue desarrollado en la Universidad de Stanford para la programación de su manipulador de 6 g.d.l. (el brazo de Stanford, descrito como ejemplo en el tema 2). Posee sintaxis y características similares a ALGOL y a PASCAL concurrente, y permite las especificaciones de las órdenes a nivel robot, y, muy restringidamente, algo a nivel tarea. Para entender el ejemplo que seguidamente daremos, veamos primero algunas órdenes básicas en AL.

Existe el tipo de datos MH, cuyos elementos son matrices de transformación homogénea. Las posiciones y orientaciones no tienen por qué introducirse explícitamente, sino que pueden calcularse a partir de otras, mediante el uso de ciertas órdenes para la especificación de posición, que son:

VECTOR(x, y, z): devuelve la matriz homogénea (MH) que representa una traslación cuyo

vector de traslación es (x, y, z) .

`ROT(eje, angulo)`: donde `eje` es `x, y` o `z`, y `angulo` es un número real. Devuelve la MH que representa una rotación de ángulo dado alrededor del eje dado.

`TRANS(rotacion, traslacion)`: donde `rotacion` y `traslacion` son MH. Devuelve la MH que representa la rotación y traslación combinadas.

`FRAME(rotacion, traslacion)`: donde `rotacion` y `traslacion` son MH. Devuelve la MH que representa la transformación de coordenadas entre el sistema rotado y trasladado según los argumentos, y el sistema del mundo.

Existen también órdenes para la especificación del movimiento. Las más importantes son:

`MOVE barm TO A`: donde `A` es una MH y `barm` es el punto terminal del brazo. Mueve el brazo interpolando en el espacio de articulaciones (y por tanto, no necesariamente en línea recta) de tal modo que el sistema de la pinza, tal como se definió en el tema 2, quede al final rotado y trasladado respecto al sistema del mundo según especifica la transformación `A`.

`MOVE barm TO A VIA B`: Igual que el caso anterior, pero la trayectoria será tal que en algún momento el sistema de la pinza venga descrito respecto al sistema del mundo por `B`. Esto sería similar a la sucesión de órdenes `MOVE barm TO B`; `MOVE barm TO A`; con la diferencia que en este último caso el brazo se detendría en `B`, no así usando `VIA`.

`MOVE barm TO A`

`WITH DEPARTURE = eje WRT B`

`WITH DURATION = tiempo`: Mueve el punto terminal con las mismas condiciones sobre el sistema de la pinza y la MH `A` que en el apartado anterior, pero lo hace siguiendo la dirección del eje `eje` (que puede ser `x`, `y` o `z`) del sistema cuya posición/orientación define la MH `B`. Además, emplea en realizar el movimiento el tiempo pedido.

`OPEN bhand` y `CLOSE bhand`: Órdenes para abrir y cerrar la pinza, respectivamente,

Por último, las órdenes para el uso de sensores son:

`MOVE barm TO A`

`ON FORCE(eje) >limite_fuerza DO ABORT`: Mueve el punto terminal a `A`, con las mismas consideraciones que antes, pero comprobando en todo momento que la componente de la fuerza de reacción ejercida sobre el brazo en la dirección del eje `eje` del sistema de la mano no cumpla la condición dada (en este caso, ser mayor que un cierto límite). Si la cumpliera, se ejecuta la orden `ABORT`, que interrumpe el programa.

`MOVE barm TO A`

`WITH FORCE(eje) = fuerza`

`WITH DURATION = tiempo`: Mueve el punto terminal a `A`, como siempre, pero ahora de modo que en todo momento se cumplan las condiciones requeridas, en este caso, que la fuerza de reacción que se ejerce sobre el eje `eje` sea la pedida.

Es necesario hacer notar una vez más que estas órdenes requieren la ejecución de pro-

cesos concurrentes, y que además involucran al tiempo, lo que manifiesta la necesidad de uso de un sistema operativo de tiempo real, como dijimos. Por otra parte, la diferencia entre ellas es que la primera responde a lo que antes llamamos movimiento condicionado (la condición se comprueba de modo instantáneo y genera una acción concreta en el tiempo) mientras que la segunda corresponde a lo que denotábamos por movimiento bajo restricciones (las condiciones se comprueban constantemente para garantizar su cumplimiento, es decir, se establece un lazo cerrado de realimentación negativa). Respecto al movimiento libre, lo efectúan las órdenes MOVE sin condiciones.

Como ejemplo de tarea programada en AL, véase la figura 6.2. Consiste en tomar un tornillo del alimentador, e insertarlo (no enroscarlo) en el orificio del bloque marcado con una D.

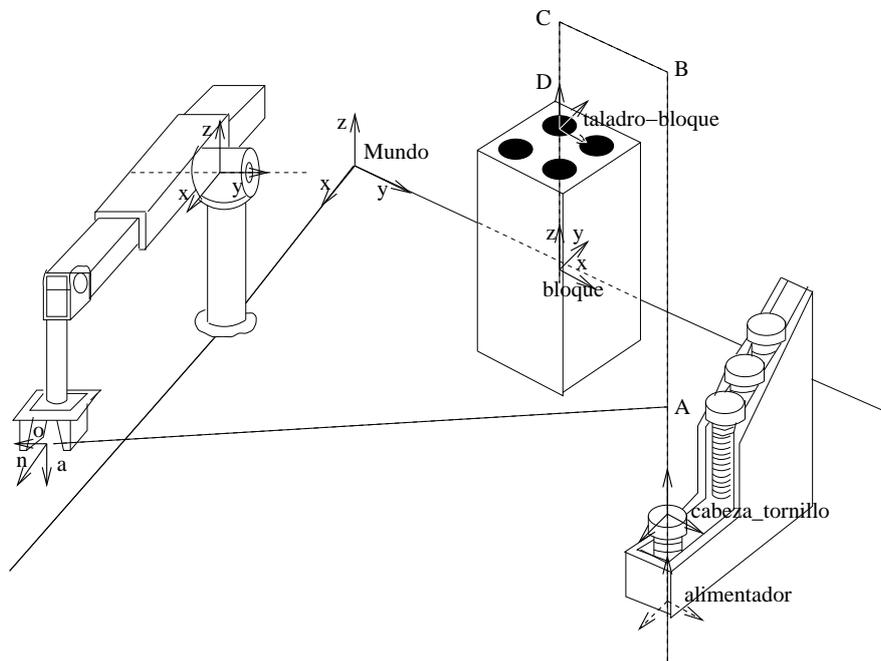


Figura 6.2: Ejemplo de tarea para resolver en AL

El programa sería el que se lista a continuación:

```
BEGIN insercion
```

```
dia_tor := 0.5*cm;
long_tor := 1*cm;
alt_bloque := 5*cm;
intentos := 0;
sujección := false;
```

```
bloque := FRAME(ROT(Z,90*grad),VECTOR(20,5,0)*cm);
alimentador := FRAME(nilrot,VECTOR(25,20,0)*cm);
```

```
punta_tor := alimentador * TRANS(nilrot,nilvect);
cabeza_tor := punta_tor * TRANS(nilrot,VECTOR(0,0,long_tor));
tal_bloque := bloque * TRANS(nilrot,VECTOR(0,0,alt_bloque));
```

```

A := alimentador * TRANS(nilrot,VECTOR(0,0,5)*cm);
B := alimentador * TRANS(nilrot,VECTOR(0,0,alt_bloque+2*long_tor));
C := tal_bloque * TRANS(nilrot,VECTOR(0,0,2*long_tor));
D := tal_bloque * TRANS(nilrot,VECTOR(0,0,long_tor));

OPEN bhand TO dia_tor + 1*cm;
MOVE barm TO cabeza_tor VIA A WITH APPROACH = -Z WRT alimentador;

DO
  CLOSE bhand TO 0.9*dia_tor;
  IF bhand < dia_tor THEN
    BEGIN
      OPEN bhand TO dia_tor + 1*cm;
      MOVE barm TO ⊗ - 1*Z*cm;
    END
  ELSE sujección := verdadero;
  intentos := intentos + 1;
UNTIL sujección OR (intentos > 3);
IF NOT sujección THEN ABORT;

MOVE barm TO B VIA A WITH DEPARTURE = Z WRT alimentador;

MOVE barm TO D VIA C WITH APPROACH = -Z WRT tal_bloque;

MOVE barm TO ⊗ - 0.1*Z*cm ON FORCE(Z) > 10*Nw DO ABORT;

MOVE barm TO tal_bloque DIRECTLY
WITH FORCE(Z) = -10*Nw
WITH FORCE(X) = 0.1*Nw
WITH FORCE(Y) = 0.1*Nw
WITH DURATION = 5*seconds;

```

END inserción

En este programa se comienza inicializando varias variables necesarias, tanto numéricas, como de tipo MH. Obsérvese que, efectivamente, el sistema llamado `bloque` ha sido obtenido trasladando el sistema del mundo a $(20,5,0)$ y rotándolo 90^0 alrededor de su eje Z . A continuación, otros sistemas de referencia útiles se inicializan a partir de los anteriores usando las órdenes de traslación y rotación. A la hora de llevar el brazo a coger el tornillo, no procedemos directamente (lo cual, probablemente, haría chocar a la pinza con el tornillo) sino que, con la pinza abierta suficientemente, nos situamos arriba del tornillo (punto `A`) y bajamos hacia él en la dirección del eje Z del sistema alimentador. Las ventajas de una programación así son obvias: si el alimentador cambiase de lugar, el cambio en la inicialización de su variable de posición/orientación sería lo único que habría que hacer.

A continuación obsérvese el uso de un bucle de comprobación (`DO ...UNTIL`) que baja sucesivamente el brazo si éste, por la razón que fuere, no hubiera podido asir el tornillo, intentando hasta tres veces. El símbolo \otimes usado es una pseudo-variable que contiene la MH de transformación del punto terminal (la posición/orientación presente) en todo momento. El proceso de aproximación para el depósito es similar, buscando un punto superior desde

donde descender verticalmente, y la fase final se compone de un movimiento condicionado, que debería dejar al tornillo exactamente sobre el orificio, sin chocar con el bloque, seguido de un movimiento bajo restricciones, que introduce el tornillo asegurándose de que no toque las paredes (condiciones sobre la fuerza en los ejes X e Y).

VAL-II

Como se explicó antes, VAL y VAL-II son los lenguajes creados por Unimation para su robot PUMA. Su sintaxis recuerda a BASIC o FORTRAN, y las ordenes de movimiento son un poco más compactas que las de AL. Las necesarias para entender el ejemplo son:

REMARK: Introduce un comentario al programa

SET/SETI: Inicializan variables de tipo MH o entero, respectivamente.

TYPE/TYPEI: Imprimen el valor de una string/variable de entero.

SHIFT: Traslada un sistema de coordenadas definido por su transformación DH mediante el vector dado.

OPENI/CLOSEI: Abre/cierra la pinza, atendiendo a los sensores de fuerza.

MOVE: Lleva el punto terminal a la posición/orientación descritas por la MH dada, haciendo coincidir el vector de aproximación de la pinza con el eje Z del sistema dado.

MOVES: Igual que MOVE, pero ejecuta el movimiento en línea recta.

APPRO: Hace que el punto terminal se aproxime al sistema dado en la dirección del eje Z de dicho sistema, es decir, de modo que el vector de aproximación del sistema de la pinza siga la dirección de dicho eje durante la aproximación. Un segundo parámetro real opcional indica la distancia a la que debe quedar el punto terminal del origen del sistema.

DEPARTS: Igual que APPRO, pero separándose del sistema dado.

La tarea esta vez consistirá en tomar cuatro bloques, que algún otro robot, una cinta transportadora, o un humano sitúan en una posición de depósito conocida, y apilarlos formando una torre en otra posición dada. Véase la figura 6.3. El programa, que llamaremos APILA, se lista en la página siguiente.

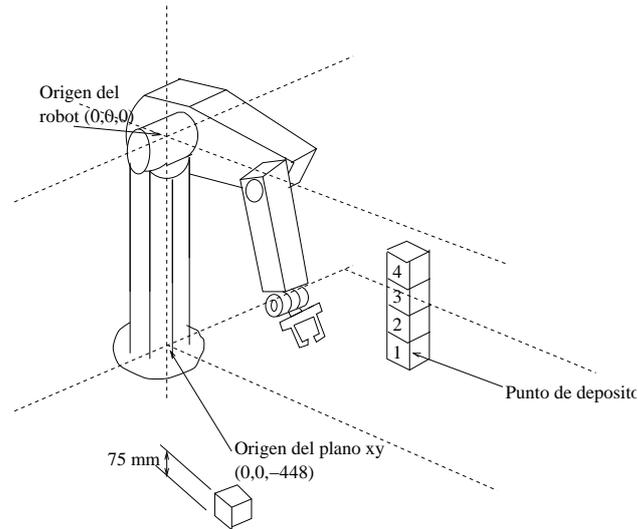


Figura 6.3: Ejemplo de tarea para resolver en VAL

```
.PROGRAM APILA
.LOCATIONS
OK
.LISTP APILA
.PROGRAM APILA
1.      REMARK ESTE PROGRAMA TOMA PIEZAS DE UN PUNTO CONOCIDO
2.      REMARK (PICKUP), Y LAS DEPOSITA EN UNA LOCALIZACION LLAMADA B
3.      REMARK SE ASUME QUE DEBEN APILARSE 4 CUBOS, UNO SOBRE OTRO
4.      OPENI
5.      SET B = DEPOSIT
6.      SETI COUNT=0
7.      10 APPRO PICKUP, 200.00
8.      MOVES PICKUP
9.      CLOSEI
10.     DEPARTS 200.00
11.     APPRO B, 200.00
12.     MOVES B
13.     OPENI
14.     DEPARTS 200.00
15.     SETI COUNT = COUNT + 1
16.     TYPEI COUNT
17.     REMARK COUNT INDICA EL NUMERO DE CUBOS
18.     REMARK YA APILADOS EN ESTE MOMENTO
19.     IF COUNT EQ 4 THEN 20
20.     REMARK ACTUALIZAR LA LOCALIZACION DE SUBIENDOLO 75.00 MM.
21.     SHIFT B BY 0.00, 0.00, 75.00
22.     GOTO 10
23.     20 SPEED 50.00 ALWAYS
24.     READY
25.     TYPE *** FIN DEL PROGRAMA APILA ***
.END
```

```

.LISTL
      X/JT1   Y/JT2   Z/JT3   O/JT4   A/JT5       T
DEPOSIT -445.03 130.59 -448.44 -87.654 88.890 -180.000 .
PICKUP  163.94 433.84 -448.38 178.006 88.896 -180.000
.EXEC APILA
COUNT = 1.
COUNT = 2.
COUNT = 3.
COUNT = 4.
** END OF PROGRAM APILA ***
PROGRAM COMPLETED: STOPPED AT STEP 25
.

```

Obsérvese el uso de `APPRO` y `DEPPART` que suplen el uso de puntos vía, como en `AL`. El texto anterior representa, no sólo el listado, sino también la salida por pantalla, y las órdenes escritas en el *prompt* del sistema (un punto). Las variables de posición/orientación usadas (`DEPOSIT` y `PICKUP`) no se han inicializado, sino que se han cargado de un fichero de localizaciones adjunto al programa con la orden `LOCATIONS`, y pueden ser visualizadas con `LISTL`, así como el texto del programa lo es con `LISTP`.

6.7. Lenguajes orientados a la tarea

6.7.1. Características

Pese a relevar al programador de la realización de todas las tareas que se han citado arriba, la programación de cualquier acción mínimamente compleja en lenguajes orientados al robot sigue siendo tediosa, y está sometida a gran probabilidad de fallo debido a imperfecciones en las partes, o a la aparición de eventos no previstos. Para paliar este problema surgen los lenguajes orientados a tarea, en los que el programador especifica qué quiere hacer, y es el sistema el que decide qué movimientos y comprobaciones sensoriales, y en qué orden, habrá que realizar. Esta decisión se toma en función de dos factores: los objetivos marcados, y el estado del mundo en cada momento.

En los lenguajes orientados a tarea el usuario la describe en un lenguaje de alto nivel, y un módulo planificador consulta una base de datos, conocida como el modelo del mundo, y transforma las especificaciones de la tarea en un programa orientado a robot. Esto está relacionado con las técnicas usadas en Inteligencia Artificial para la generación automática de programas. El esquema sería como se muestra en la figura 6.4.

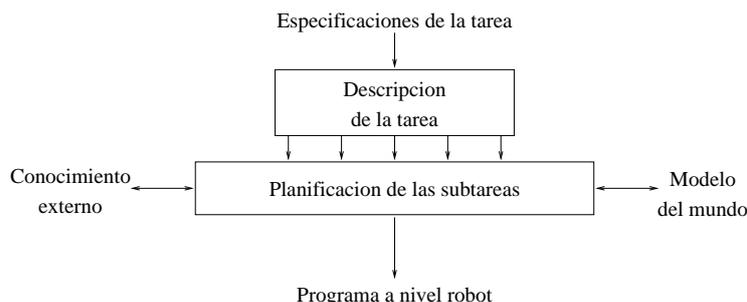


Figura 6.4: Acción de un lenguaje a nivel tarea

Como ejemplo, imaginemos que se trata de tomar un tornillo de un soporte en que se encuentra apoyado e insertarlo en un orificio localizado. Esta tarea global podría descomponerse en:

- Situarse sobre el tornillo.
- Asir el tornillo.
- Llevar el tornillo sobre el bloque.
- Insertar el tornillo.

A su vez, la primera subtarea se dividiría en:

- Conocer dónde está el robot, y dónde está el tornillo.
- Conocer dónde están todos los demás objetos (para no chocar con ellos).
- Determinar una trayectoria libre de colisión.

Las demás subtarear se podrían dividir similarmente, teniendo en cuenta además que la última requeriría información de los sensores de fuerza.

Conocer dónde está el robot es relativamente fácil: hay que leer los sensores internos (codificadores ópticos, normalmente) y usar la cinemática directa. Pero conocer dónde está el tornillo (y los demás objetos) es asunto mucho más complejo.

La aproximación más usada hasta ahora a este problema consiste en almacenar modelos geométricos para los objetos y tratar de obtener a partir de los sensores externos descripciones, también geométricas, que encajen en los objetos almacenados.

A los sistemas que ayudan a la modelización geométrica, almacenamiento, consulta y modificación de los modelos se les llama sistemas CAD (por *Computer Aided Design*, Diseño Asistido por Computador) y son habitualmente usados en ingeniería.

En aplicaciones robóticas la mayoría de los objetos a manejar son los llamados MMO (*man made objects*, objetos hechos por el hombre), y suele ser posible describirlos usando un conjunto discreto y pequeño de primitivas geométricas. Los dos tipos esenciales de descripción son en términos de volúmenes, llamada CGS (*constructive solid geometry*, geometría constructiva de sólidos) y en términos de los límites del objeto, sean éstos dados como bordes o como superficies; esto es el llamado BR (*boundary representation*, representación de límites). Muchos de los formalismos habituales usan:

- Un conjunto de fronteras (bordes) dados como curvas en el espacio cuyas ecuaciones paramétricas se explicitan, además de un conjunto de puntos límite de dichos bordes.
- Un conjunto de superficies, dadas también por sus ecuaciones, así como los planos u otras superficies que las limitan.
- Un conjunto de cilindros generalizados. Un cilindro generalizado es el volumen abarcado por una porción de superficie plana arbitraria, que se mueve a lo largo de una curva alabeada, manteniendo en todo momento su vector normal en la dirección de la tangente a la curva. La forma de la porción de superficie plana puede variar al desplazarse por la curva. Por eso, para describir un cilindro generalizado deberemos dar la ecuación de la curva alabeada, los puntos de inicio y final del desplazamiento, y la ley de variación de los límites de la superficie al desplazarse ésta. Ejemplos:

- + Un cilindro usual de radio r y altura H se puede describir como la circunferencia de radio r constante desplazándose con su centro sobre la recta $x = y = 0$ (el eje z) entre $z = 0$ y $z = H$
 - + Un cono sería una circunferencia de radio r desplazándose con su centro sobre la recta $x = y = 0$ (el eje z) entre $z = 0$ y $z = H$, pero esta vez siendo su radio $r = K(H - z)$, con K y H constantes.
 - + Un prisma poligonal sería un polígono regular con su centro sobre el eje z , en las mismas condiciones que el primer caso.
 - + Una pirámide poligonal sería el mismo polígono sobre la misma recta, pero para el que la apotema fuese variable como $a = K(H - z)$.
- Véase la figura 6.5
- Una descomposición en células: consiste en dividir el espacio en celdillas, normalmente regulares (p. ej., una retícula cúbica) e indicar cuáles de sus elementos pertenecen al sólido que se describe. Esta representación es ineficiente en términos de memoria, pero muy simple para calcular las condiciones de movimiento que no violen la impenetrabilidad de la materia.

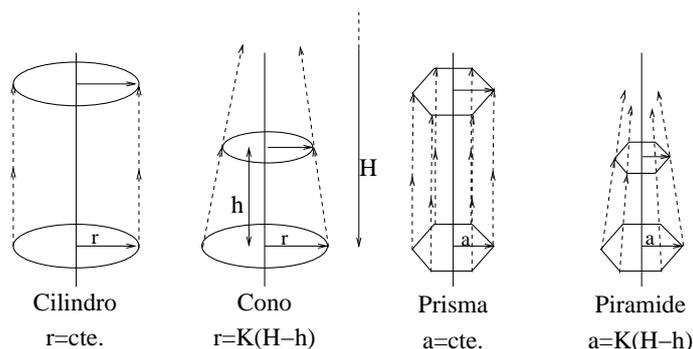


Figura 6.5: Ejemplos de cilindros generalizados

Cada una de las descripciones propuestas tiene sus ventajas e inconvenientes en términos de introducción de los modelos, almacenamiento, modificación, etc. No obstante, la característica que más nos interesa en Robótica es su capacidad de instanciación.

Instanciación es el llenado de una estructura vacía que representa a un objeto con valores para cada uno de sus campos (parámetros definitorios del objeto) para compararla con una o varias estructuras llenas almacenadas. A la hora de la comprobación hay que permitir una cierta tolerancia en los valores de los parámetros, puesto que éstos deben haber sido (supuestamente) obtenidos por los sensores, que, como dijimos, son siempre afectados por ruido.

Ejemplos de sistemas de modelado geométrico destinados a ser usados por lenguajes de programación a nivel tarea son AUTOPASS, y RAPT. Ambos se verán con detalle en los ejemplos. Un caso (prácticamente, el único) de lenguaje (más bien, sistema) de programación a nivel tarea es Handey, desarrollado en el MIT por T. Lozano-Pérez. Se verá con más detenimiento como ejemplo de sistema robótico clásico, en el tema 8.

6.7.2. Estudios de caso

AUTOPASS

Aunque no es realmente un lenguaje orientado a tarea, AUTOPASS fue creado por IBM para usos robóticos, y pretende ser el modelador geométrico que debería ser consultado por las rutinas de planificación de tareas. De hecho, es un sistema CAD, pero con añadidos, que permiten describir los objetos de un modo estructurado, que hace que su forma de almacenamiento represente de modo natural su estructura geométrica, lo cual permita después reconocer objetos instanciando sus estructuras (de hecho, mediante un encaje (*matching*) sintáctico).

Cada objeto se define en AUTOPASS mediante un procedimiento que invoca a otros, bien previamente definidos, bien primitivas del lenguaje. Cada procedimiento admite parámetros que definen las características del objeto. Todos los objetos se consideran poliédricos, y los cilindros se aproximan como prismas de base poligonal (de hecho, la primitiva `CYLIND` es realmente un prisma).

Ejemplo: Un tornillo sería un sólido compuesto por otros tres, todos "cilíndricos": la cabeza, el cuerpo y la punta, Véase figura 6.6

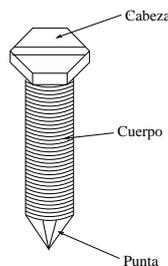


Figura 6.6: Ejemplo de pieza en AUTOPASS

La definición sería:

```
Tornillo: PROCEDURE(l_pun,r_pun,nf_pun,l_cuer,r_cuer,l_cab,r_cab,nf_cab)
DECLARE
  l_pun,r_pun,nf_pun,l_cuer,r_cuer,l_cab,r_cab,nf_cab: FLOAT;
  ag1,ag2: MERGED;
BEGIN
  CALL SOLID(CYLIND,"Punta",l_pun,r_pun,nf_pun);
  CALL SOLID(CYLIND,"Cuerpo",l_cuer,r_cuer,_);
  CALL SOLID(CYLIND,"Cabeza",l_cab,r_cab,nf_cab);
  CALL ag1 := MERGE("Cabeza","Cuerpo",union);
  CALL ag2 := MERGE(ag1,"Punta",union);
  Tornillo := ag2;
END Tornillo;
```

donde el subrayado en la definición de "Cuerpo" representa una variable libre (cualquier valor para ella será posible).

Supongamos que el sistema de sensores de tacto o visión da una descripción del tornillo de la forma

```

MERGE(
  MERGE(
    SOLID(CYLIND(_, 2, 2.5, 6)), SOLID(CYLIND(_, 12, 2, 14))),
  SOLID(CYLIND(1, 2, 6))
);

```

AUTOPASS usa operaciones de lógica y un encaje sintáctico que puede instanciar el dato en alguno de los procedimientos, haciendo así encajar la descripción proporcionada por los sensores con la almacenada. Obsérvese el uso de `MERGE`, que especifica una pieza como fija a la otra (y, por tanto, a la hora de hacer razonamiento geométrico teniendo en cuenta el movimiento de las partes cuando son capturadas por el brazo, el programa tendrá en cuenta que ambas partes se deben mover conjuntamente). En AL existía una orden similar (`AFFIX`) que declara sistemas de coordenadas como fijos unos a otros. Pero las órdenes de AUTOPASS son más potentes, en cuanto que también permiten definir adosamiento condicional, el cual se usa sobre todo para modelizar el comportamiento de la gravedad (un objeto A sobre un objeto B está adosado a él sólo si B tiene cierta orientación. En otro caso, A caería). Lo que no permite AUTOPASS, y sería deseable al trabajar con robots, es la posibilidad de manejar adosamientos no rígidos, y tolerancias.

RAPT

RAPT, uno de los primeros, pero más brillantes intentos de modelización y razonamiento geométrico, fue propuesto por R. Popplestone y otros en la Univ. de Edimburgo. Los objetos pueden ser modelizados de un modo un poco más flexible que en AUTOPASS. Exactamente, pueden ser cuerpos con caras planas o esféricas, o bien ser cilindros generalizados, o tener orificios cilíndricos. Las relaciones entre los objetos se definen mediante primitivas que se consideran naturales desde el punto de vista humano; concretamente, las relaciones de contacto son `AGAINST`, `FIT` y `COPLANAR`. A RAPT deben dársele las especificaciones de un conjunto de objetos en forma de descripciones de sus aspectos (caras y vértices), y declaración explícita de las relaciones entre éstos en un cierto instante (estado inicial), así como la misma descripción para el estado deseado (estado final). RAPT genera un programa compuesto por acciones que son traslaciones o rotaciones referidas al sistema de coordenadas del mundo de los sistemas de los objetos. A partir de ahí es fácil encontrar el conjunto de movimientos del robot que efectúan esto: basta asir el objeto con la pinza, y declarar como unidos los sistemas de objeto y pinza.

Un ejemplo de tarea para RAPT sería la que se muestra en la figura 6.7, en la que hay que pasar de la situación inicial con dos cubos apilados, a la final, en la que el no apilado queda entre los otros dos.

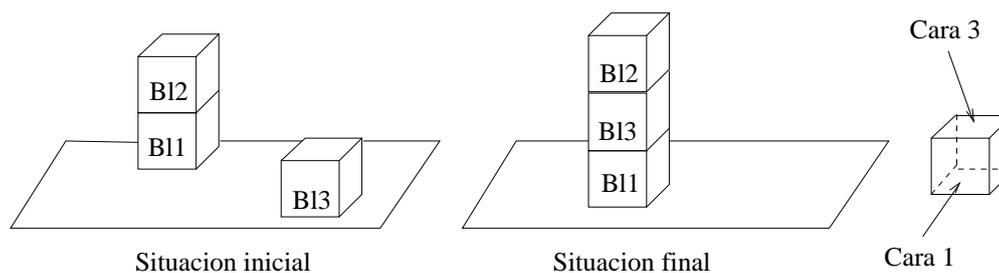


Figura 6.7: Ejemplo de tarea para resolver en RAPT

Las descripciones serían:

Estado inicial	Estado final
(B11_cara1 AGAINST Mesa)	(B11_cara1 AGAINST Mesa)
(B12_cara1 AGAINST B11_cara3)	(B13_cara1 AGAINST B11_cara3)
(B13_cara1 AGAINST Mesa)	(B12_cara1 AGAINST B13_cara3)

El lenguaje generará instrucciones de alto nivel que codifiquen una secuencia de acciones como:

1. Llevar brazo sobre B12	7. Llevar B13 sobre B11
2. Asir B12	8. Soltar B13
3. Llevar B12 sobre Mesa	9. Llevar brazo sobre B12
4. Soltar B12	10. Asir B12
5. Llevar brazo sobre B13	11. Llevar B12 sobre B13
6. Asir B13	12. Soltar B12

A partir de aquí es fácil la generación de un programa de robot. P. ej., las órdenes 6 a 8 se podrían codificar en AL como

```
B12Up := FRAME(B12, VECTOR(0,0,5)*cm)
AFFIX B13 TO barm
MOVE barm TO B12 VIA B12Up WITH DEPARTURE=Z WRT B12
MOVE barm TO ⊗-5.5*cm ON FORCE(Z) >10*Nw DO STOP
OPEN bhand TO 10*cm
DETTACH B13 FROM barm
```

En todos los casos los detalles geométricos que luego aparecen en el programa de robot (cuánto ha de abrirse la pinza, etc.) los resuelve el sistema, puesto que conoce el modelo geométrico de todos los objetos. Para efectuar el razonamiento geométrico RAPT define un sistema de coordenadas asociado a cada aspecto de cada objeto, según su tipo (vértice, borde, plano, esfera, cilindro u orificio) y se definen las relaciones **AGAINST**, **FIT** y **COPLANAR** en términos de las relaciones entre los sistemas de coordenadas asociados a cada aspecto.

Luego se detallan los efectos que sobre cada relación tiene cada acción. Un mecanismo complicado de resolución de restricciones por inferencia lógica genera el plan de acción (secuencia de acciones). Aquí se usan reglas de transformación del tipo $neg(\sin(x)) \rightarrow \sin(neg(x))$ o como $twix(\theta) * twix(\varphi) \rightarrow twix(\theta + \varphi)$, donde neg es el cambio de signo, y $twix$ una rotación de ángulo dado alrededor del eje x .

Como comentario final, hacer notar que la prensión de los objetos es un aspecto no suficientemente tratado por estos lenguajes. Habría que saber cómo debe asirse un objeto (en qué configuración del brazo, con qué fuerza, y en qué punto del objeto). La prensión debe ser posible y estable. Para planearla se usa:

- Conocimiento sobre la geometría del objeto y de la mano: éstos se pueden modelizar como poliedros, y verificar que las intersecciones de sus planos caigan fuera de el espacio ocupado por los propios poliedros.

- Conocimiento de la distribución de masas del objeto: si es homogéneo, coincidirá con su centro geométrico, pero si no, es conveniente que el c.d.m. esté sobre la línea que une los puntos de agarre.
- Reducción de la incertidumbre en la posición del objeto: esto es preciso para cerrar los dedos sobre los puntos correctos. Incluso sin necesidad de sensores que den la posición del objeto con precisión, se puede confinarlo exactamente abriendo y cerrando dos veces la pinza sobre él en direcciones perpendiculares, de modo que el objeto se vea empujado a quedar en un punto conocido. Este es un ejemplo de movimiento de las partes sin ser capturadas (*constrained motion*).

6.8. Planificación global de tareas. Nota sobre sistemas CIM

Los problemas expuestos anteriormente (y sobre los que se incidirá en el tema 8) relativos a la dificultad de construcción de los modelos y el manejo de la incertidumbre hubieran frenado, si no impedido del todo, el uso industrial de los robots, de haber intentado resolverlos completamente. La solución obvia ha sido evitarlos, en aplicaciones que lo permitan. De hecho, los robots industriales suelen funcionar en entornos suficientemente estructurados como para que la incertidumbre en las posiciones de las partes a manipular esté reducida a un mínimo. Además, todas las partes que sufren un paso concreto del proceso de fabricación son casi siempre idénticas. Esto hace que los procesos elementales de manipulación sean simples, y que el estado del proceso (la factoría) completa pueda ser descrito como un conjunto de variables discretas con valores bien definidos, que indican, p. ej., el número de piezas de tipo X que se encuentran en ese momento sobre la cinta C , el número de piezas de tipo Y en almacén, el máximo número de piezas que el robot Z puede procesar en un tiempo T , etc.

Esto significa que cada subprocesso puede entenderse como un consumidor de piezas de los tipos A_1, \dots, A_n que genera piezas tipo B a un ritmo determinado; es misión del sistema de control determinar el flujo y orden óptimo de los procesos para conseguir unos objetivos dados.

Un sistema de esta clase, llamado CIM (*Computer Integrated Manufacturing*, Fabricación Integrada por Computador) debe regular el comportamiento de la factoría como un todo, estableciendo una conexión entre el *stock*, los procesos de fabricación, y los requerimientos del usuario, para generar en todo instante las órdenes de control. Estas son, normalmente, activación o parada de los procesos elementales, que suelen ser rígidos, y fiables. Por ello, la efectividad de los sistemas CIM se sustenta en la confianza en que los procesos básicos funcionen. Para asegurarse de ello, éstos deben ser simples, y la incertidumbre en las posiciones y orientaciones de las partes debe haber sido eliminada mediante el uso de cintas transportadoras especiales, movimientos de alineamiento de las partes, etc.

Los métodos empleados para generación de las acciones necesarias son muy variados; uno de los más usados son las redes de Petri, pero también sistemas de reglas de producción, o cualquier buen algoritmo de búsqueda en grafos, si se puede plantear el problema como la búsqueda en un espacio de estados donde cada nodo representa un estado de la planta, y tiene una o más acciones asociadas.

Tema 7: Robots móviles

7.1. Introducción

En los últimos años la investigación sobre robots móviles está adquiriendo gran desarrollo. Ello se debe, en parte, al abaratamiento del hardware necesario para su construcción, y en parte a la nueva concepción industrial de planta de fabricación flexible, que requiere la reconfiguración de la secuencia de acciones necesarias para una producción variada, lo que a su vez exige facilidad de desplazamiento de los materiales entre cualesquiera puntos de la factoría.

Las soluciones a este problema de transporte de material en entornos "flexibles" son varias. La primera situar las máquinas cerca unas de otras, y organizadas de modo que uno o más brazos robot puedan llevar las piezas entre ellas; esta configuración, un caso particular de las llamadas células de fabricación flexible, es sólo apropiada para un número limitado de máquinas. Otra solución válida es el uso de vehículos autoguiados (denotados usualmente como *AGV*, *Autonomous Guided Vehicles*), los cuales recurren para el guiado a sistemas externos preprogramados, tales como un raíl, cables eléctricos enterrados que crean un campo magnético, etc.. Finalmente, la mejor solución sería disponer de vehículos autónomos (denotados como *ALV*, *Autonomous Land Vehicles*) que se mueven de un punto a otro sin necesidad de ayudas externas (al menos, no en todo momento), lo que los hace capaces de navegación genérica en un entorno dado a partir de órdenes de alto nivel; a la secuencia de tales órdenes se la suele llamar el **plan** de la misión.

Por otra parte, la construcción experimental de (normalmente) pequeños robots móviles en laboratorios universitarios está haciendo surgir un tipo de investigación que aborda los aspectos de conexión senso-motora (como nunca nos cansaremos de recalcar, lo más fundamental en Robótica) desde un punto de vista diferente a aproximaciones anteriores, y que conlleva también un cambio de visión en la concepción clásica de la Inteligencia Artificial, la cual se aborda intentando construir vida artificial. Se incidirá sobre ello en el tema 8.

Aun siendo todavía en gran parte materia de investigación, existen algunas tecnologías, métodos y formulaciones matemáticas para robots móviles que, sin estar tan asentadas como las que se usan para manipuladores, han comenzado ya a ser comunes, e incluidas en algún libro de texto. Ello hace que sea ya posible plantear este tema como parte del curso. En él se estudiará primero la formulación cinemática, que, análogamente a lo que ocurría en los manipuladores, permitirá conocer la posición del robot respecto a un sistema externo a partir de los datos sobre él que, mediante sus sensores, podemos conocer. Después se hablará de navegación, explicando las técnicas que permiten a un robot móvil desplazarse de modo seguro hasta un objetivo dado, en entornos sólo parcialmente conocidos. Se mostrarán después los medios (sensores y actuadores más comunes) que se instalan hoy en robots móviles, y su conexión.

7.2. Cinemática de robots móviles

Aun cuando es posible construir y describir mecánicamente robots que se desplazan sobre patas, por movimiento alternativo de éstas, aquí nos ocuparemos sólo de los que lo hacen sobre ruedas. Las ruedas siempre estarán en contacto con el suelo, lo que hace que

se deba tratar al robot como una cadena cinemática cerrada, donde el suelo actúa también como enlace entre las ruedas. Esto requiere plantearse una formulación para la cinemática bastante diferente a la de los manipuladores. Hablaremos aquí de dos tipos: cinemática interna, que establecerá la relación entre las articulaciones dentro del robot, y externa, que establecerá esta relación entre el robot y el resto del mundo.

Hay una serie de diferencias con los robots estáticos que es preciso señalar:

1. Los robots con ruedas, como ya dijimos, son cadenas cinemáticas cerradas, dado que tienen varias ruedas en contacto simultáneo con el suelo, lo que complica la formulación. Los robots andantes son cadenas cinemáticas alternativamente abiertas y cerradas, según la pata esté o no en contacto con el suelo.
2. Las pseudo-articulaciones del robot móvil tienen dos grados de libertad, es decir, forman un par link-joint de orden superior, con un único punto de contacto (el de la rueda con el suelo).
3. Algunos grados de libertad pueden no estar conectados a ningún actuador.
4. No será necesario medir posición, velocidad y aceleración de cada enlace, porque hay ligaduras entre unos y otros, de modo que no todos los grados de libertad del robot van a ser independientes.

Para que el análisis que vamos a plantear sea válido, necesitamos admitir ciertas restricciones. Concretamente, sólo consideraremos robots capaces de locomoción sobre una superficie mediante la acción de ruedas montadas en el robot y en contacto con dicha superficie. Las ruedas se suponen montadas en dispositivos que proveen o permiten movimiento relativo entre su punto de anclaje (punto de guiado) y una superficie (el suelo) con la cual debe haber un único punto de contacto rodante. Se asume que:

- El robot está construido con mecanismos rígidos.
- Hay no más de un enlace de guiado por cada rueda.
- Todos los ejes de guiado son perpendiculares al suelo.
- La superficie de movimiento es un plano.
- No hay deslizamiento entre las ruedas y el suelo.
- La fricción es suficientemente pequeña como para permitir el giro de cualquier rueda alrededor del eje de guiado.

Todas estas suposiciones se cumplen razonablemente bien en la mayoría de los robots móviles actuales, excepto la penúltima (y, dependiendo del terreno, también la antepenúltima). El deslizamiento es, de hecho, el mayor problema con que se encuentra cualquier robot móvil a la hora de establecer una autolocalización precisa.

Procederemos ahora a asignar sistemas de coordenadas fijos a determinados puntos que nos permitirán, por análisis de las transformaciones entre unos y otros, establecer la cinemática. Sea una rueda montada sobre una pieza, digamos una regleta, fijada al cuerpo del robot. La rueda puede girar alrededor de un eje vertical (el eje de guiado) que se articula en el extremo de la regleta. Véase la figura 7.1

Los sistemas de coordenadas se asignan del modo siguiente:

El eje z de cualquier sistema es siempre perpendicular al suelo. En realidad, analizaremos el movimiento en el plano, y por tanto, la tercera coordenada será ignorada en todos los casos.

Existirá un sistema de referencia fijo, el sistema del mundo, que llamaremos R .

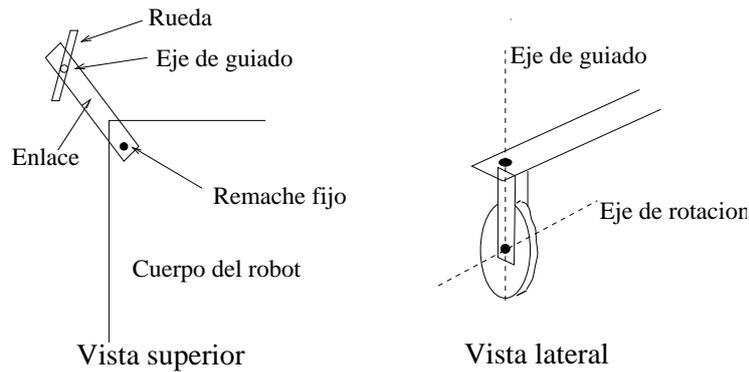


Figura 7.1: Esquema de la rueda de un robot móvil

Un sistema llamado C irá fijo al cuerpo (chasis) del robot, moviéndose con él. Normalmente se fijará su origen en el centro geométrico del robot, y su eje Y en la dirección privilegiada de avance, si la hay.

Se fijarán una serie de sistemas, uno para cada rueda, llamados S_i , cuyos ejes z coincidirán con los ejes de guiado. Sus ejes x e y se suelen poner paralelos a los del sistema C .

Otra serie de sistemas, los W_i , se fijarán a cada rueda con su eje z siendo el de guiado, (como en el caso de los S_i), pero de modo que giran en sentido vertical con la dirección de la rueda, de modo que su eje x coincide en todo momento con la dirección de avance de esta rueda. Como podrá suponerse, el ángulo entre cada W_i y su S_i servirá para representar el ángulo de dirección de la rueda i .

Otro sistema, que llamaremos \bar{C} , está fijo al suelo, pero coincide instantáneamente con C . Esto puede entenderse como si el sistema \bar{C} estuviera "pintado" en el suelo, y en el instante en que vamos a analizar el movimiento, determinando posición y velocidad del robot, éste pasa justo por encima, de modo que C y \bar{C} se superponen. Obsérvese que \bar{C} es estático, y, como se verá luego, se usa como un simple artificio matemático.

Análogamente, se sitúan los sistemas que llamaremos \bar{W}_i como coincidentes instantáneamente con su respectivo W_i en el instante de análisis del movimiento (y, como el \bar{C} , también estáticos).

La razón de introducir estos sistemas es que nos permitirán medir velocidades respecto a la referencia del mundo, R , independientemente de la posición. Para entender todo esto véase la figura 7.2.

Las transformaciones entre dos cualesquiera de estos sistemas se pueden, obviamente, escribir como giros alrededor del eje z con algún ángulo δ , más traslaciones de vector \vec{p} , cuya forma general sería

$$\begin{pmatrix} \cos(\delta) & -\sin(\delta) & 0 & p_x \\ \sin(\delta) & \cos(\delta) & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

pero en lugar de ello, y como p_z es constante, prescindiremos de ella. En su lugar añadiremos un parámetro de orientación φ , que expresará la dirección de avance del robot. Nótese que para describir un sólido rígido como un todo que efectúa un movimiento plano nos basta dar tres coordenadas: las dos que expresen la posición de uno de sus puntos (p. ej., el centro) en un instante, llamémoslas (p_x, p_y) , y la que da la dirección en que el objeto está orientado en tal instante, sea φ . Véase la figura 7.3

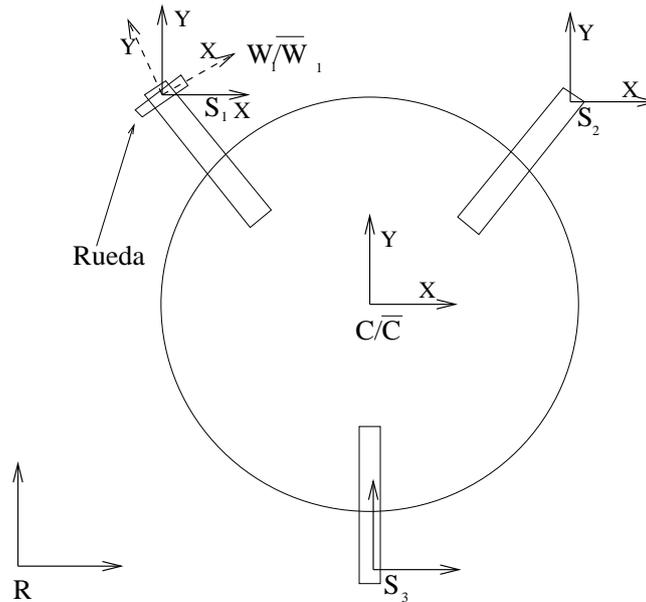


Figura 7.2: Sistemas de referencia para un robot móvil

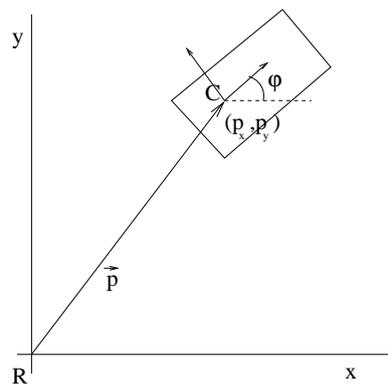


Figura 7.3: Coordenadas de posición-orientación de un robot móvil

El vector de posición-orientación será, pues, $(p_x, p_y, \phi)'$. Nótese que, si los ejes x de dos sistemas cualesquiera forman un ángulo ϕ , y la orientación del robot respecto al primero es φ , respecto al segundo será $\varphi + \phi$. Entonces, la matriz de transformación genérica entre dos sistemas A y B usando las coordenadas antedichas en forma homogénea, o sea, $(p_x, p_y, \phi, 1)'$, sería:

$${}^A_B \mathbf{T} = \begin{pmatrix} \cos(\delta) & -\sin(\delta) & 0 & p_x \\ \sin(\delta) & \cos(\delta) & 0 & p_y \\ 0 & 0 & 1 & \phi \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Su inversa, es decir, la transformación que llevaría del sistema B hasta el A sería:

$${}^B_A \mathbf{T} = \begin{pmatrix} \cos(\delta) & \sin(\delta) & 0 & -p_x \cos(\delta) - p_y \sin(\delta) \\ -\sin(\delta) & \cos(\delta) & 0 & p_x \sin(\delta) - p_y \cos(\delta) \\ 0 & 0 & 1 & -\phi \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Nótese que ${}^A_B \mathbf{T}$ multiplicada por $(0, 0, 0, 1)'$, que es el vector de posición/orientación del

origen de B respecto a sí mismo, da $(p_x, p_y, \phi, 1)$, o sea, la posición/orientación del origen de B respecto a A .

En lo sucesivo, necesitaremos también usar las derivadas de estas transformaciones respecto al tiempo. La expresión general de tal derivada (derivando componente a componente) sería:

$${}^A \dot{\mathbf{T}}_B = \begin{pmatrix} -\dot{\delta} \sin(\delta) & -\dot{\delta} \cos(\delta) & 0 & \dot{p}_x \\ \dot{\delta} \cos(\delta) & -\dot{\delta} \sin(\delta) & 0 & \dot{p}_y \\ 0 & 0 & 0 & \dot{\phi} \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Es fácil ver que, respecto a la derivada de la transformación inversa, ${}^B \dot{\mathbf{T}}_A = -{}^A \dot{\mathbf{T}}_B$.

Usando estas transformaciones como flechas en un grafo cuyos nodos sean los distintos sistemas de referencia, podemos escribir el grafo de la cadena cinemática, según se muestra en la figura 7.4.

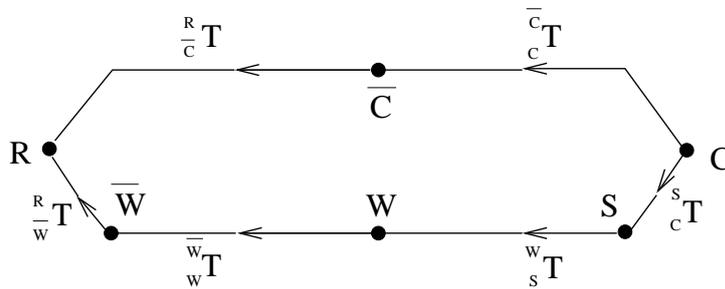


Figura 7.4: Grafo de transformaciones entre sistemas

Nótese que se trata de un grafo cerrado (cíclico), como corresponde al hecho de que tenemos una cadena cinemática cerrada. En realidad, los sistemas W y S indicados son W_i y S_i , habiendo uno de estos grafos por cada rueda.

Ahora estamos en condiciones de plantearnos la obtención de la velocidad del robot (o sea, del sistema C) respecto al sistema de referencia externo R , a partir del conocimiento de las velocidades de cada rueda. Esto será la cinemática (o, en este caso, mejor cinética) directa.

Para ello, escribamos la posición del centro del robot, C , respecto a R

$${}^R p_C = {}^R \mathbf{T}_C \cdot {}^C p_C$$

siendo, naturalmente, ${}^C p_C = (0, 0, 0, 1)'$. Aquí hemos transformado a través de la rama superior del grafo. Ahora, la velocidad se obtendrá por derivación respecto al tiempo:

$${}^R v = {}^R \dot{p}_C = \overline{\left({}^R \mathbf{T}_C \cdot {}^C \mathbf{T}_C \right)} \cdot {}^C p_C = \left({}^R \dot{\mathbf{T}}_C \mathbf{T}_C + {}^R \mathbf{T}_C \dot{\mathbf{T}}_C \right) \cdot {}^C p_C$$

donde se ha usado la regla de derivación del producto, y el hecho de que ${}^C p_C$ es constante. Ahora bien, \bar{C} está fijo al suelo (aun cuando en el preciso instante de la medida coincida con C); luego \bar{C} no varía respecto al sistema fijo, R , y por tanto ${}^R \dot{\mathbf{T}}_{\bar{C}} = 0$, con lo que la expresión de la velocidad queda:

$$v = {}^R \mathbf{T}_{\bar{C}} \cdot \dot{\mathbf{T}}_{\bar{C}} \cdot {}^C p_C \tag{7.1}$$

El producto $\dot{\mathbf{T}}_{\bar{C}} \cdot {}^C p_C$ expresaría la velocidad del robot medida por el sistema fijo el suelo en el preciso momento en que el robot pasa sobre él. La matriz ${}^R \mathbf{T}_{\bar{C}}$ meramente expresa

este vector velocidad en coordenadas del mundo. Con destino a su uso en esta expresión, podremos calcular ${}^C\dot{\mathbf{T}}$, pero no ${}^R\mathbf{T}$, de modo que podremos saber cómo se está moviendo el robot en un cierto instante, pero no su posición absoluta (salvo, como veremos luego, por integración).

Tratemos de trabajar sobre ${}^C\dot{\mathbf{T}}$. Para obtenerla, partiremos del grafo. Escribamos primero la expresión de ${}^R\mathbf{T}$

$${}^R\mathbf{T} = {}^R\mathbf{T} \cdot {}^{\bar{W}}\mathbf{T} \cdot {}^W\mathbf{T} \cdot {}^S\mathbf{T} \cdot {}^C\mathbf{T} \quad (7.2)$$

Para llegar de R a \bar{C} hemos usado la rama más larga de las dos que la unen en el grafo. Ello es precisamente porque a través de la rama corta deberíamos conocer la relación entre el sistema R y el \bar{C} , es decir, la posición absoluta del robot. Esto sólo es posible si disponemos de algún sensor externo que actúe como sistema de posicionamiento global (sistemas de ultrasonidos, una cámara cenital, etc.) lo cual no es corriente. Como se verá luego, usando la otra serie de transformaciones podemos llegar a depender sólo de los sensores internos.

Ahora, derivaremos respecto al tiempo la ecuación 7.2, notando que ${}^R\dot{\mathbf{T}} = {}^{\bar{W}}\dot{\mathbf{T}} = {}^S\dot{\mathbf{T}} = 0$. Esto es porque los sistemas con barra son de coincidencia instantánea con sus homónimos sin barra, pero están, según dijimos, fijos al suelo, R . Y por otra parte, tanto el sistema S como el C están fijos al chásis del robot, de modo que sus posiciones relativas no cambian con el tiempo. Así, la derivada queda:

$$0 = {}^R\mathbf{T} \cdot {}^{\bar{W}}\dot{\mathbf{T}} \cdot {}^W\mathbf{T} \cdot {}^S\mathbf{T} \cdot {}^C\mathbf{T} + {}^R\mathbf{T} \cdot {}^{\bar{W}}\mathbf{T} \cdot {}^W\dot{\mathbf{T}} \cdot {}^S\mathbf{T} \cdot {}^C\mathbf{T} + {}^R\mathbf{T} \cdot {}^{\bar{W}}\mathbf{T} \cdot {}^W\mathbf{T} \cdot {}^S\dot{\mathbf{T}} \cdot {}^C\mathbf{T} + {}^R\mathbf{T} \cdot {}^{\bar{W}}\mathbf{T} \cdot {}^W\mathbf{T} \cdot {}^S\mathbf{T} \cdot {}^C\dot{\mathbf{T}}$$

y como ${}^R\mathbf{T}$ no es idénticamente nula, podemos simplificarla por la derecha, con lo que la ecuación anterior se escribe como

$$0 = {}^{\bar{W}}\dot{\mathbf{T}} \cdot {}^W\mathbf{T} \cdot {}^S\mathbf{T} \cdot {}^C\mathbf{T} + {}^{\bar{W}}\mathbf{T} \cdot {}^W\dot{\mathbf{T}} \cdot {}^S\mathbf{T} \cdot {}^C\mathbf{T} + {}^{\bar{W}}\mathbf{T} \cdot {}^W\mathbf{T} \cdot {}^S\dot{\mathbf{T}} \cdot {}^C\mathbf{T} + {}^{\bar{W}}\mathbf{T} \cdot {}^W\mathbf{T} \cdot {}^S\mathbf{T} \cdot {}^C\dot{\mathbf{T}}$$

Además, recordemos que todo esto se calcula para el instante de tiempo concreto en que W y \bar{W} coinciden, así como C y \bar{C} , con lo que se tiene que ${}^{\bar{W}}\mathbf{T} = {}^C\mathbf{T} = I$, quedando finalmente

$$0 = {}^{\bar{W}}\dot{\mathbf{T}} \cdot {}^W\mathbf{T} \cdot {}^S\mathbf{T} + {}^{\bar{W}}\mathbf{T} \cdot {}^W\dot{\mathbf{T}} \cdot {}^S\mathbf{T} + {}^{\bar{W}}\mathbf{T} \cdot {}^W\mathbf{T} \cdot {}^S\dot{\mathbf{T}} + {}^{\bar{W}}\mathbf{T} \cdot {}^W\mathbf{T} \cdot {}^S\mathbf{T} \cdot {}^C\dot{\mathbf{T}}$$

Por fin, para despejar ${}^C\dot{\mathbf{T}}$, que recordemos que era lo que nos interesa, pues su inversa aparece en 7.1, multiplicaremos ambas partes de la igualdad por ${}^C\mathbf{T} \cdot {}^W\mathbf{T}$, pasando un término a la otra parte y usando la propiedad de que ${}^C\dot{\mathbf{T}} = -{}^{\bar{C}}\dot{\mathbf{T}}$, con lo que

$${}^{\bar{C}}\dot{\mathbf{T}} = {}^C\mathbf{T} \cdot {}^S\mathbf{T} \cdot {}^{\bar{W}}\dot{\mathbf{T}} \cdot {}^W\mathbf{T} \cdot {}^S\mathbf{T} + {}^C\mathbf{T} \cdot {}^S\mathbf{T} \cdot {}^{\bar{W}}\mathbf{T} \cdot {}^W\dot{\mathbf{T}} \cdot {}^S\mathbf{T} + {}^C\mathbf{T} \cdot {}^S\mathbf{T} \cdot {}^{\bar{W}}\mathbf{T} \cdot {}^W\mathbf{T} \cdot {}^S\dot{\mathbf{T}} + {}^C\mathbf{T} \cdot {}^S\mathbf{T} \cdot {}^{\bar{W}}\mathbf{T} \cdot {}^W\mathbf{T} \cdot {}^S\mathbf{T} \cdot {}^C\dot{\mathbf{T}} \quad (7.3)$$

Ahora, se trata de escribir cada una de las matrices cuyo producto forma en la ecuación 7.3 la transformación buscada, ${}^{\bar{C}}\dot{\mathbf{T}}$. Esto lo haremos mirando los esquemas de los sistemas de coordenadas (figura 7.2), y usando la fórmula de la derivada temporal de una matriz de este tipo, así como la siguiente notación:

- β será el ángulo entre el eje x del sistema C y el mismo eje del sistema S . Es fijo, y usualmente nulo.
- α es el ángulo de giro vertical de la rueda (ángulo de dirección). Lo forman S y W , y se mide desde S .
- θ : ángulo instantáneo de giro vertical de la rueda. Lo forman el eje x de \bar{W} y el eje x de W , luego es nulo en el instante de observación, pero su derivada, $\omega = \dot{\theta}$, no. Nótese que, aunque

puede parecerlo, no es igual a α , sino que expresa cómo ve un sistema de referencia externo el giro de una rueda. Si el robot completo está tomando una curva de radio constante, α no cambiará, pero θ sí, sería en ese caso el ángulo de giro del robot como un todo, tal como lo ve un observador externo.

- (d_x, d_y) : vector entre el origen de C y el de S , expresado en C

Véase el esquema de la rueda en la figura 7.5, en donde se añaden otros dos ángulos: θ_y , que es propiamente el ángulo de rotación natural de la rueda sobre su eje, y θ_x , que, según la condiciones impuestas al principio de que el eje de dirección sea siempre perpendicular al suelo, será constante y nulo, por lo que sólo se dibuja por completitud.

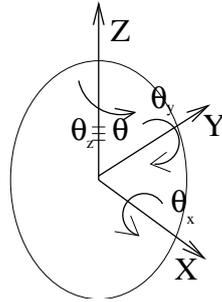


Figura 7.5: Angulos de giro de la rueda

Con esta notación las matrices de transformación resultan ser:

$$\begin{aligned}
 {}^c_s \mathbf{T} &= \begin{pmatrix} c\beta & -s\beta & 0 & d_x \\ s\beta & c\beta & 0 & d_y \\ 0 & 0 & 1 & \beta \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^s_c \mathbf{T} &= \begin{pmatrix} c\beta & s\beta & 0 & -d_x c\beta - d_y s\beta \\ -s\beta & c\beta & 0 & d_x s\beta - d_y c\beta \\ 0 & 0 & 1 & -\beta \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 {}^s_w \mathbf{T} &= \begin{pmatrix} c\alpha & -s\alpha & 0 & 0 \\ s\alpha & c\alpha & 0 & 0 \\ 0 & 0 & 1 & \alpha \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^w_s \mathbf{T} &= \begin{pmatrix} c\alpha & s\alpha & 0 & 0 \\ -s\alpha & c\alpha & 0 & 0 \\ 0 & 0 & 1 & -\alpha \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 {}^w_w \dot{\mathbf{T}} &= \begin{pmatrix} -\omega s\theta & -\omega c\theta & 0 & \bar{w} v_x \\ \omega c\theta & -\omega s\theta & 0 & \bar{w} v_y \\ 0 & 0 & 0 & \bar{w} \omega \\ 0 & 0 & 0 & 0 \end{pmatrix} & {}^w_s \dot{\mathbf{T}} &= \begin{pmatrix} -\dot{\alpha} s\alpha & -\dot{\alpha} c\alpha & 0 & 0 \\ \dot{\alpha} c\alpha & -\dot{\alpha} s\alpha & 0 & 0 \\ 0 & 0 & 0 & -\dot{\alpha} \\ 0 & 0 & 0 & 0 \end{pmatrix}
 \end{aligned}$$

Ahora hay que multiplicar todas estas matrices, de acuerdo a la ecuación 7.3. Pero recordemos, por la ecuación 7.1, que lo que realmente buscábamos era la velocidad del chasis del robot, que denotaremos por

$$\bar{c} v = \begin{pmatrix} \bar{c} v_x \\ \bar{c} v_y \\ \bar{c} \omega \\ 0 \end{pmatrix} = {}^{\bar{c}}_c \dot{\mathbf{T}} \cdot {}^c p_c = {}^{\bar{c}}_c \dot{\mathbf{T}} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = (4^a \text{ columna de } {}^{\bar{c}}_c \dot{\mathbf{T}}) \quad (7.4)$$

por lo que sólo necesitamos la 4ª columna del producto. Multiplicando las matrices según 7.3, simplificando, introduciendo el producto en 7.1, e igualando las expresiones resultantes

según 7.4, se obtiene:

$$\begin{aligned}\bar{c} v_x &= c(\alpha + \beta)^{\bar{w}} v_x - s(\alpha + \beta)^{\bar{w}} v_y + d_y^{\bar{w}} \omega - d_y \dot{\alpha} \\ \bar{c} v_y &= s(\alpha + \beta)^{\bar{w}} v_x + c(\alpha + \beta)^{\bar{w}} v_y - d_x^{\bar{w}} \omega + d_x \dot{\alpha} \\ \bar{c} \omega &= \bar{w} \omega - \dot{\alpha} \\ 0 &= 0\end{aligned}$$

o, escribiéndolo en forma matricial,

$$\begin{pmatrix} \bar{c} v_x \\ \bar{c} v_y \\ \bar{c} \omega \end{pmatrix} = \begin{pmatrix} c(\alpha + \beta) & -s(\alpha + \beta) & d_y & -d_y \\ s(\alpha + \beta) & c(\alpha + \beta) & -d_x & d_x \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} \bar{w} v_x \\ \bar{w} v_y \\ \bar{w} \omega \\ \dot{\alpha} \end{pmatrix} \quad (7.5)$$

$$\bar{c} v = \hat{J} \cdot \dot{q} \quad (7.6)$$

Nótese que la matriz \hat{J} antedicha relaciona velocidades lineales y angulares de la rueda con velocidades del chásis medidas en el sistema de referencia cartesiano. De este modo, y por semejanza con el caso de los manipuladores, podemos considerarla como un Jacobiano (de hecho, un pseudo-jacobiano, en razón de las ligaduras entre articulaciones que aparecen).

El problema ahora es saber si podemos medir las velocidades de la rueda, $(\bar{w} v_x, \bar{w} v_y, \bar{w} \omega, \dot{\alpha})$, y efectivamente esto es posible poniendo en ella sensores (normalmente, codificadores ópticos) apropiados. Para ello es necesario relacionar estas velocidades con los ángulos realmente medidos. Esto depende del tipo concreto de rueda. Veamos dos casos:

Sea una rueda como la mostrada en las figuras 7.1 y 7.5, pero con el eje vertical fijo. En ese caso, la velocidad lineal en la dirección de avance es proporcional a la velocidad angular de la rueda, siendo el coeficiente de proporcionalidad el radio, r . Entonces, se tiene

$$\begin{aligned}\bar{w} v_x &= r \cdot \dot{\theta}_y \\ \bar{w} v_y &= 0 \\ \bar{w} \omega &= \dot{\theta}_z \\ \dot{\alpha} &= 0\end{aligned} \quad \text{o, en forma matricial,} \quad \begin{pmatrix} \bar{w} v_x \\ \bar{w} v_y \\ \bar{w} \omega \\ \dot{\alpha} \end{pmatrix} = \begin{pmatrix} 0 & r & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \\ \dot{\alpha} \end{pmatrix} \equiv W \cdot \dot{\Theta}$$

Obsérvese que, aun siendo la rueda fija, y por tanto α constante (o sea, $\dot{\alpha}$ idénticamente nula), θ_z no es necesariamente constante: el robot puede estar girando respecto al sistema externo.

En el caso de que la rueda también pudiese girar alrededor del eje vertical (eje de guiado), tendríamos

$$\begin{aligned}\bar{w} v_x &= r \cdot \dot{\theta}_y \\ \bar{w} v_y &= 0 \\ \bar{w} \omega &= \dot{\theta}_z \\ \dot{\alpha} &= \dot{\alpha}\end{aligned} \quad \text{o, en forma matricial,} \quad \begin{pmatrix} \bar{w} v_x \\ \bar{w} v_y \\ \bar{w} \omega \\ \dot{\alpha} \end{pmatrix} = \begin{pmatrix} 0 & r & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \\ \dot{\alpha} \end{pmatrix} \equiv W \cdot \dot{\Theta}$$

Al fin, y usando las ecuaciones 7.4 y 7.5, se tendrá que

$$\bar{c} v = \hat{J} \cdot W \cdot \dot{\Theta} \equiv J \cdot \dot{\Theta}$$

donde el producto $J \equiv \hat{J} \cdot W$ es otra forma para el pseudo-Jacobiano, que establece la relación directa entre la velocidad que queremos conocer y las variables medibles. En los dos casos de las ruedas anteriores, y suprimiendo las variables innecesarias, las ecuaciones quedan, para el primer caso,

$$\begin{pmatrix} \bar{w} v_x \\ \bar{w} v_y \\ \bar{w} \omega \\ \dot{\alpha} \end{pmatrix} = \begin{pmatrix} r & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\theta}_y \\ \dot{\theta}_z \end{pmatrix}$$

y para el segundo,

$$\begin{pmatrix} \bar{w} v_x \\ \bar{w} v_y \\ \bar{w} \omega \\ \dot{\alpha} \end{pmatrix} = \begin{pmatrix} r & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{\theta}_y \\ \dot{\theta}_z \\ \dot{\alpha} \end{pmatrix}$$

Las variables que aparecen en el lado derecho de estas ecuaciones son directamente medibles, conectando, p. ej., codificadores ópticos a cada eje de cada rueda (también en el eje de guiado). Esto hace que podamos obtener la velocidad instantánea del robot en su conjunto respecto al sistema externo. Pero observemos que no hay ecuación que nos de la posición; por ello, ésta sólo se podrá conocer por integración de la velocidad. Este es el procedimiento de autolocalización llamado odometría, e incidiremos sobre él en la siguiente sección.

Volviendo ahora al aspecto físico, recordemos que todas estas ecuaciones se deben plantear para cada rueda; dejando aparte la coordenada α , que tiene que ver con el giro individual de esa rueda alrededor de su eje, las otras son un sistema de 3 ecuaciones con 3 incógnitas, que relacionan las variables medibles en una rueda con la velocidad lineal y angular del robot como un todo. Cuando se plantee esto para las N ruedas tendremos un sistema de $3N$ ecuaciones con sólo 3 incógnitas, que deberá ser compatible; en términos físicos, cada rueda debiera ordenar al robot el mismo movimiento. De otro modo, no habrá movimiento alguno, o bien una o más ruedas deslizarán, incumpliendo las condiciones originales.

Veamos ahora un ejemplo. Sea un robot con tres ruedas, según se ve en la figura 7.6, la delantera libre, de modo que puede girar arbitrariamente alrededor de un eje vertical sin estar controlada, y las traseras, orientadas con ángulos α_1 y α_2 respecto a la dirección de avance (fijos o no, se verá luego cada caso).

Los vectores que expresan la velocidad de cada rueda serían

$$\dot{q}_1 = \begin{pmatrix} \bar{w} v_{x1} \\ \bar{w} v_{y1} \\ \bar{w} \omega_1 \\ \dot{\alpha}_1 \end{pmatrix} \quad \text{y} \quad \dot{q}_2 = \begin{pmatrix} \bar{w} v_{x2} \\ \bar{w} v_{y2} \\ \bar{w} \omega_2 \\ \dot{\alpha}_2 \end{pmatrix}$$

Las respectivas matrices \hat{J} , usando la ecuación 7.5, y teniendo en cuenta que $\beta_1 = \beta_2 = 0$, son

$$\hat{J}_1 = \begin{pmatrix} c\alpha_1 & -s\alpha_1 & \frac{l}{2} & -\frac{l}{2} \\ s\alpha_1 & c\alpha_1 & L & -L \\ 0 & 0 & 1 & -1 \end{pmatrix} \quad \text{y} \quad \hat{J}_2 = \begin{pmatrix} c\alpha_2 & -s\alpha_2 & -\frac{l}{2} & \frac{l}{2} \\ s\alpha_2 & c\alpha_2 & L & -L \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

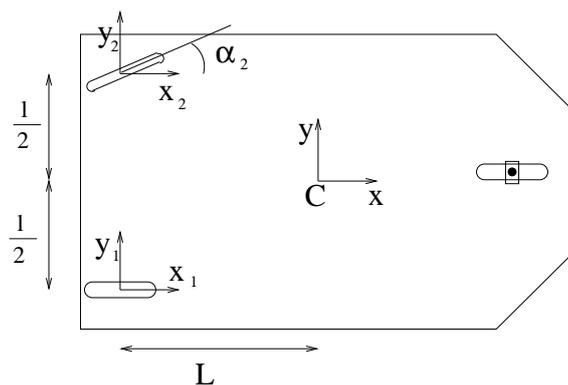


Figura 7.6: Ejemplo de robot móvil

Por otra parte, escribiendo las dos matrices W_i que relacionan las coordenadas propias de cada rueda con su vector velocidad \dot{q}_i , sale:

$$\dot{q}_1 = \begin{pmatrix} r & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \dot{\theta}_{y1} \\ \dot{\theta}_{z1} \\ \dot{\alpha}_1 \end{pmatrix} \quad y \quad \dot{q}_2 = \begin{pmatrix} r & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \dot{\theta}_{y2} \\ \dot{\theta}_{z2} \\ \dot{\alpha}_2 \end{pmatrix}$$

Al multiplicar y escribir por separado cada ecuación resultan los sistemas (obtenidos por sustitución en 7.5):

$$\begin{aligned} \bar{c} v_x &= r c \alpha_1 \dot{\theta}_{y1} + \frac{l}{2} \dot{\theta}_{z1} - \frac{l}{2} \alpha_1 & \bar{c} v_x &= r c \alpha_2 \dot{\theta}_{y2} - \frac{l}{2} \dot{\theta}_{z2} + \frac{l}{2} \alpha_2 \\ \bar{c} v_y &= r s \alpha_1 \dot{\theta}_{y1} + L \dot{\theta}_{z1} - L \alpha_1 & \bar{c} v_y &= r s \alpha_2 \dot{\theta}_{y2} + L \dot{\theta}_{z2} - L \alpha_2 \\ \bar{c} \omega &= \dot{\theta}_{z1} - \alpha_1 & \bar{c} \omega &= \dot{\theta}_{z2} - \alpha_2 \end{aligned}$$

Ahora, si las ruedas traseras mantuvieran constante su orientación en la dirección de movimiento (o sea, $\alpha_1 = \alpha_2 = 0$) se obtiene que

$$\begin{aligned} v_x &= r \dot{\theta}_{y1} + \frac{l}{2} \dot{\theta}_{z1} = r \dot{\theta}_{y2} - \frac{l}{2} \dot{\theta}_{z2} \\ v_y &= L \dot{\theta}_{z1} = L \dot{\theta}_{z2} \\ \omega &= \dot{\theta}_{z1} = \dot{\theta}_{z2} \end{aligned}$$

de donde

$$\omega = \frac{r}{l} (\dot{\theta}_{y1} - \dot{\theta}_{y2})$$

Esto quiere decir que, si ambas ruedas giran a la misma velocidad, el robot avanzará hacia adelante (o atrás, según el sentido) pero no girará ($\omega = 0$), o sea, el avance será en línea recta. Si, por el contrario, cada rueda gira con una velocidad angular diferente, el robot describirá una circunferencia, girando todo él con velocidad angular ω .

Sea ahora el caso en que las orientaciones de las ruedas traseras son constantes e iguales ($\alpha_1 = \alpha_2 \equiv \alpha$), pero no nulas (o sea, no apuntan hacia el frente). En ese caso, las ecuaciones son:

$$\begin{aligned} v_x &= r c \alpha \dot{\theta}_{y1} + \frac{l}{2} \dot{\theta}_{z1} = r c \alpha \dot{\theta}_{y2} - \frac{l}{2} \dot{\theta}_{z2} \\ v_y &= r s \alpha \dot{\theta}_{y1} + L \dot{\theta}_{z1} = r s \alpha \dot{\theta}_{y2} + L \dot{\theta}_{z2} \\ \omega &= \dot{\theta}_{z1} = \dot{\theta}_{z2} \end{aligned}$$

La única forma de hacer compatibles estas ecuaciones es que $\dot{\theta}_{z1} = \dot{\theta}_{z2} = 0$, y que $\dot{\theta}_{y1} = \dot{\theta}_{y2}$, es decir, el robot como un todo ya no puede girar ($\omega = 0$) y además, las ruedas no pueden girar a distintas velocidades. Esto es un ejemplo de las ligaduras que dijimos que aparecían por el hecho de trabajar con cadenas cinemáticas cerradas. Las componentes del vector velocidad global quedan

$$\begin{aligned} v_x &= r c \alpha \dot{\theta}_{y1} \\ v_y &= r s \alpha \dot{\theta}_{y1} \end{aligned}$$

lo que quiere decir que el robot avanza en línea recta, formando un ángulo α con el eje x .

Finalmente, si α_1 y α_2 son constantes, pero no iguales, se puede probar que el sistema es incompatible, es decir, el robot no puede avanzar.

El expuesto es un caso particular en el que aparecen determinadas restricciones. Las condiciones de movilidad de un robot se determinan en general a partir del rango de matrices construidas por yuxtaposición de los jacobianos de cada rueda; la formulación general de este problema se considera fuera del alcance de un curso elemental.

7.3. Navegación

Se llama navegación al conjunto de métodos y técnicas usados para dirigir el curso de un robot móvil a medida que éste atraviesa su entorno. Se supone que se debe llegar a algún destino pedido, sin perderse y sin chocar ni con obstáculos fijos, ni con otros móviles que eventualmente puedan aparecer en el camino.

Para efectuar navegación lo más común es disponer de un mapa, aunque no necesariamente. **Mapa** es cualquier tipo de representación del entorno en la memoria del robot; inmediatamente veremos los principales tipos. A partir de un mapa, se puede determinar un camino apropiado entre dos puntos pedidos, lo cual será más o menos complejo según haya sido la representación escogida. Por último, habrá que seguir ese camino. Para ello, se usa la autolocalización, de varios modos diferentes.

El algoritmo genérico que un robot móvil emplearía para navegar podría ser:

Comienzo (de la tarea)

Si existe un mapa **entonces**

 Buscar caminos en el mapa

 Seleccionar uno usando una función de evaluación

Si el camino es complejo, **entonces**

 descomponer en subcaminos

 Recoger datos sensoriales del entorno

Mientras no estemos en el objetivo **hacer**

 Seguir el primer camino

Si se cumple el subobjetivo **entonces**

 obtener siguiente subobjetivo

 Recoger datos sensoriales del entorno

Si hay un objeto en el camino **entonces**

 Parar el robot

Si el objeto es estacionario **entonces**

 Actualizar el mapa

 Buscar un camino alternativo al subobjetivo

Si tal camino existe **entonces**

 Seguirlo

si no

 Cortar el programa y replanificar la tarea

si no (objeto móvil)

 Parar y esperar a que el camino este libre

fin (mientras)

 (Robot en el objetivo)

si no (no existe un mapa) **entonces**

Mientras no estemos en el objetivo **hacer**

 Recoger datos sensoriales del entorno

Si entorno en la dirección del objetivo está libre **entonces**

 Moverse hacia el objetivo

si no (objeto en el camino)

Si entorno en otras direcciones está libre **entonces**

 Seleccionar una dirección de modo heurístico

 Moverse en esa dirección

```

    si no (el robot está atrapado)
        Cortar el programa y replanificar la tarea
    fin (mientras)
    (Robot en el objetivo)
fin (de la tarea)

```

7.3.1. Mapas del entorno

Será necesario desarrollar estructuras de datos apropiadas para el almacenamiento del tipo de información que el robot va a necesitar, así como algoritmos para su manipulación eficiente; las estructuras usadas deben permitir combinar la información adquirida por los sensores externos con la ya almacenada en el mapa. Las características que debería tener una buena estructura de datos serían:

- Debe encajar con el metodo escogido de planificación de caminos
- Debe minimizar el número de elementos ambiguos (conteniendo a la vez objeto y espacio libre)
- Debe reflejar formas complejas con la máxima precisión, o al menos con la suficiente
- Debe almacenar formas complejas con un número pequeño de elementos
- Debe almacenar formas (o espacio libre) grandes con los mínimos elementos que sea posible
- Debe poder manejar mapas locales y globales
- Debe poder almacenar el estado de las áreas mapeadas
- Debe facilitar el movimiento desde un elemento de la estructura a elementos adyacentes
- Debe incluir subestructuras para almacenar caminos
- Debe poder ser fácilmente manipulada por los algoritmos de localización, búsqueda de caminos y navegación
- Debe poder ser fácilmente extendida para incorporar nueva información procedente del proceso sensorial

Realmente, ninguna de las estructuras de datos propuestas cumple todos los requisitos. En cada aplicación particular hay que elegir la que cumpla los más necesarios para ella.

Veamos ahora los tipos de mapas que son comunmente usados. Primeramente, están los basados en información sensorial, con dos tipos:

- Mapas de marcas en el terreno (*landmarks*): algunas localizaciones particulares fácilmente identificables por el sistema sensorial del robot (cierta esquina, un grupo de objetos bien visibles o tubos de neón, etc.) actúan como marcas relevantes (*landmarks*). Se representan como nodos de un grafo (que pueden tener características asociadas, para garantizar su identificación unívoca), los cuales se unen por los arcos del grafo que normalmente, representan la accesibilidad (si existe arco entre dos nodos, el robot puede desplazarse directamente de uno a otro de los *landmarks* a los que los nodos representan). Estos arcos pueden también estar etiquetados con características del recorrido como distancia, dirección, tiempo de tránsito, etc.

- Mapas de ocupación: Se basan en representar el terreno como una retícula, regular o no, cada una de cuyas casillas contiene un valor útil para el robot, que suele ser la certitud de ocupación, es decir, qué grado de creencia tiene el robot sobre el estado de una determinada casilla, desde -1 (es seguro que está libre) hasta +1 (es seguro que está ocupada) pasando por 0 (no hay evidencia en ningún sentido). Estos mapas se pueden construir por

métodos visuales, mediante la toma de imágenes por un par estereo de cámaras (o una sólo que va a bordo del robot y se sitúa en varias posiciones), a partir de las proyecciones de puntos límite de un objeto, como se ve en la figura 7.7

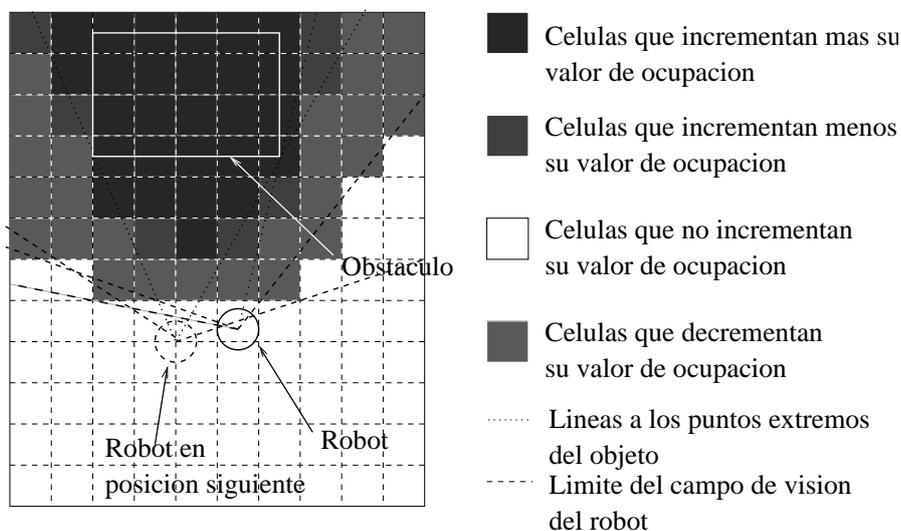


Figura 7.7: Construcción visual del mapa de ocupación

Por otra parte, se pueden clasificar los mapas de acuerdo a lo que se almacena. Aquí hay también varios tipos:

- Mapas de espacio libre: al igual que en los mapas de marcas, la estructura de almacenamiento elegida es también el grafo, pero esta vez cada nodo representa un punto de parada donde el robot pueda detenerse para sensorizar el entorno. Los arcos son líneas rectas que el robot pueda recorrer entre estos puntos sin encontrar obstáculos; evidentemente, limitarán a los posibles obstáculos. Nótese que aquí los datos almacenados sí tienen correspondencia física directa; el dual de uno de estos mapas sería el diagrama de Voronoi, donde cada polígono conteniendo a un punto de detención puede estar o bien completamente libre, o bien puede contener parcialmente a algún objeto. Véase la figura 7.8

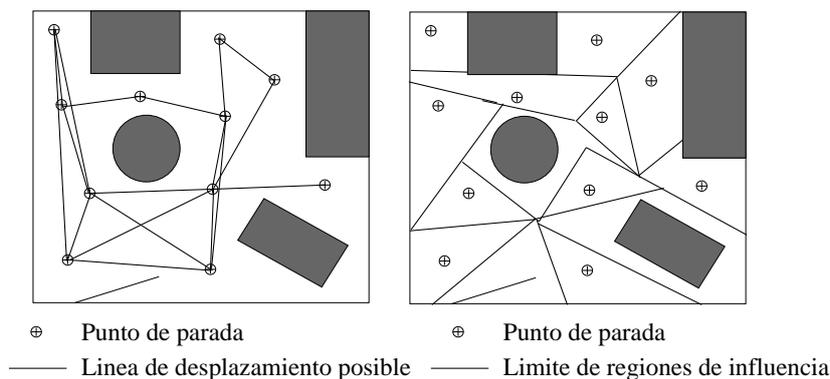


Figura 7.8: Mapa de espacio libre, y su dual

- Mapas de objetos: Como su nombre indica, lo que se almacena en ellos son los objetos (obstáculos) que el robot puede encontrar en su trayectoria, de varios modos; los más normales son considerar al objeto como un polígono, y almacenar su punto central y la extensión máxima en una serie de direcciones desde él; otro modo es caracterizarlo como

una de entre un conjunto de figuras geométricas dadas, y dar su posición y la orientación de un eje propio de esa figura

- Mapas compuestos: almacenan tanto información de objetos como de espacio libre. Una posibilidad es dividir el espacio en regiones arbitrarias, pero conocidas, e indicar en cada una de ellas si está totalmente libre, totalmente ocupada, o parcialmente ocupada. Otra alternativa es una retícula de puntos con un indicador de estado en cada punto, y una lista de a cuáles de los puntos adyacentes se puede acceder directamente; la retícula puede hacerse más o menos densa, en función del tamaño del robot.
- *Quadtrees*: Dividen el espacio mediante una retícula, y proceden por subdivisión recursiva de la misma, mientras la celda resultante sea subdividible, siendo el criterio el que no tenga toda ella el mismo carácter de ocupación.
- Mapas basados en reglas: el entorno del robot se describe como una colección de predicados del cálculo proposicional, como p. ej. `derecha(punto1, mesa)`, `distancia(punto1, punto2, 20)`, `direccion(punto2, armario, 45)`, etc. y se trata de encontrar un camino libre mediante la manipulación simbólica de estas expresiones siguiendo las reglas del cálculo de predicados y una serie de proposiciones que expresan el conocimiento geométrico usual.

7.3.2. Autolocalización

Después de haber descrito los mapas, el paso obvio es explicar cómo un robot móvil es capaz de saber en qué punto del mapa se encuentra. Para ello, se puede recurrir a la información aportada por los dos tipos posibles de sensores: internos y externos. Los principales procedimientos para la autolocalización (*dead reckoning*) son la odometría y el uso de balizas.

- Odometría: como se vió en la sección sobre cinemática, era posible conocer la velocidad instantánea del robot respecto a un sistema externo a partir del conocimiento de la velocidad de las ruedas. Será necesario instalar sensores de posición angular (normalmente, codificadores ópticos) en cada rueda. Conocido el vector velocidad, la posición en el instante presente (t) se obtendrá por integración:

$$\begin{aligned}x(t) &= x_0 + \int_0^t v_x(\tau) d\tau \\y(t) &= y_0 + \int_0^t v_y(\tau) d\tau\end{aligned}$$

donde se supone que en el instante en que se empieza a contar el tiempo ($t = 0$) la posición es conocida, y vale (x_0, y_0) . Este método presenta varios problemas:

* La integral debe hacerse por métodos numéricos, con lo cual se da una pérdida de precisión, que además es acumulativa.

* Es muy sensible al problema del deslizamiento. En ese caso, la distancia computada es mayor que la realmente recorrida.

* Le afectan falsos recorridos computados como más largos debido a la inclinación del

terreno o a pequeños obstáculos. Si una sólo rueda cruza sobre uno de éstos, el ángulo que habrá girado es mayor, y por tanto se recogerán más cuentas en su encoder que en el/los otro/s. Esto hará que el sistema de control interprete que se ha producido un giro hacia el lado de la rueda de menor recorrido, y trate de compensarlo con un giro opuesto, desviando al robot. Véase la figura 7.9.

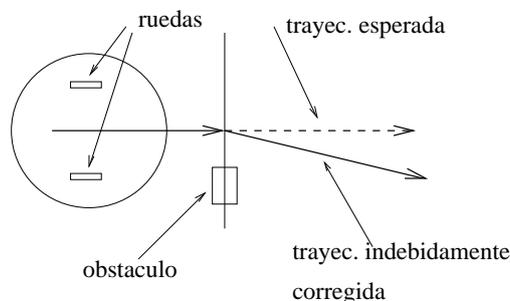


Figura 7.9: Odometría afectada por el terreno

- Balizas: son marcas de fácil localización instaladas en lugares conocidos, que el robot es capaz de detectar con sus sensores, y respecto a las cuáles se sitúa. Pueden ser marcas visuales (tubos de neón, o bandas de colores), o emisores de infrarrojos, cada uno emitiendo una señal modulada con un código conocido. Estas señales pueden ser recogidas por una óptica apropiada y proyectadas sobre una cámara CCD o un array de fotodiodos, que sirve para determinar la dirección de la que proceden. Conociendo al menos dos de éstas direcciones (aunque pueden ser más) y las posiciones absolutas de las balizas es posible determinar por triangulación la posición del robot. Véase la figura 7.10

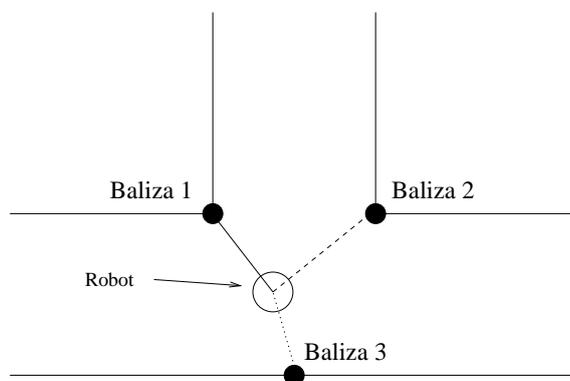


Figura 7.10: Autocalización por balizas

Una ventaja adicional es que la señal modulada de infrarrojos puede usarse también para transmitir información (órdenes, u otra) al modo de una puerta serie.

7.3.3. Planificación y seguimiento de caminos

El problema ahora es, dados un punto inicial y un punto final (meta) especificados sobre el modelo de mapa propuesto, encontrar en dicho mapa un camino libre de colisión que el robot pueda seguir. Para hacerlo físicamente, comprobará continua o intermitentemente que se encuentra sobre los puntos del camino, usando alguna de las técnicas de autocalización ya explicadas.

Mucho de lo hecho en planificación de caminos se deriva de los métodos propuestos para manipuladores, pero simplificados a dos dimensiones. En general, no se considera el caso de que un robot móvil pueda pasar bajo un obstáculo.

Los requerimientos que deberá cumplir un planificador de caminos son:

- Encontrar un camino que el robot pueda atravesar sin colisión.
- Manejar la incertidumbre en el modelo del mundo que debe instanciarse con los datos imprecisos de los sensores.
- Mantener el robot lo más lejos posible de los objetos, para que los sensores den menos datos y así se requiera menos proceso.
- Encontrar el camino óptimo (el mejor entre los posibles) y seguirlo de un modo suave.

Veamos ahora los modos de planificar un camino.

- Por guiado: consiste en llevar al robot físicamente a una serie de lugares preestablecidos, y almacenar las impresiones sensoriales que se reciben en cada uno de ellos, así como la dirección o direcciones de desplazamiento posterior hacia el/los siguiente(s) punto(s) importantes(s). Para alcanzar el punto deseado se pueden implantar lazos de realimentación que operen tomando directamente como entrada las señales sensoriales, y que generen señales de control para los actuadores, evitando el cálculo de la posición absoluta, no útil en este caso.

- Automáticamente: Aquí entran en juego algoritmos que dependen fuertemente de la representación usada para el mapa. En mapas de tipo grafo, siendo los nodos posiciones de referencia a comprobar con los sensores, la planificación consiste en encontrar el camino de mínima distancia en el grafo. La distancia se define en función de los costes de cada arco, que pueden ser bien distancias físicas, o algún otro tipo de penalización asociada a ese desplazamiento (p. ej., debida a la estrechez de un pasillo que obliga a reducir la velocidad, etc.). En mapas que contienen los objetos, los planificadores tratan de encontrar caminos por el espacio libre lo más alejados posible de los objetos. Esto es bueno en pasillos estrechos, pero puede ser ineficiente en zonas anchas, por elegir caminos más largos. Véase, p. ej., la figura 7.11 (parte izquierda), en la que se elegiría el camino A sobre el B, pese a ser más largo.

En planificadores de este tipo, pero más sofisticados, se tienen en cuenta las restricciones cinemáticas del robot, al que no se simplifica como un círculo, sino que se trata con su forma y capacidad de giro real, y se determina si se pueden negociar las esquinas, y en su caso, cómo. Este problema se conoce en la literatura como el del "transportista de pianos".

Una forma de determinar la accesibilidad de un espacio dado por un robot concreto consiste en desplazar de modo ficticio un modelo geométrico del robot de modo que toque a las fronteras de todos los objetos, y esto en todas las orientaciones posibles. Si el robot se modeliza como un círculo, el simple desplazamiento es suficiente. De este modo, se tiene un nuevo mapa con los objetos recreados según la forma del robot, el cual puede entonces considerarse como un punto, que es más fácil de tratar geométricamente.

Una vez decidido qué camino se debe recorrer, hay que proveer los medios para el efectivo seguimiento del mismo. La comparación entre impresiones sensoriales prealmacenadas o

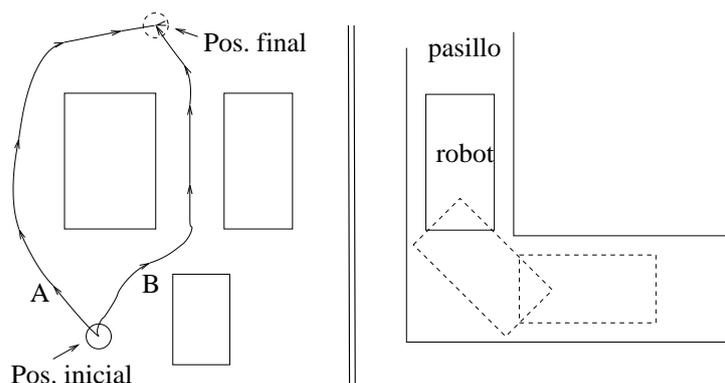


Figura 7.11: Caminos en espacio libre y negociación de esquinas

esperadas y recibidas (especialmente en los *landmarks*, pero también en cualquier otro lugar), es una forma usual, como se comentó antes, pero también hay que mencionar los algoritmos llamados de campos potenciales. Consisten en hacer una analogía física que considera cada objeto como si fuera una colina, lo que asigna una "elevación" ficticia a cada punto del mapa inversamente proporcional a su distancia al objeto. Cuando todos los objetos hacen su contribución, los valles son los puntos de mayor distancia a éstos, y el robot deberá desplazarse por ellos, como lo haría una esferita dejada a la acción de la gravedad en tal entorno. En términos físicos, la elevación asignada juega el papel de una función potencial, V , y su campo asociado sería $\vec{E} = -\nabla V$. Véase la figura 7.12.

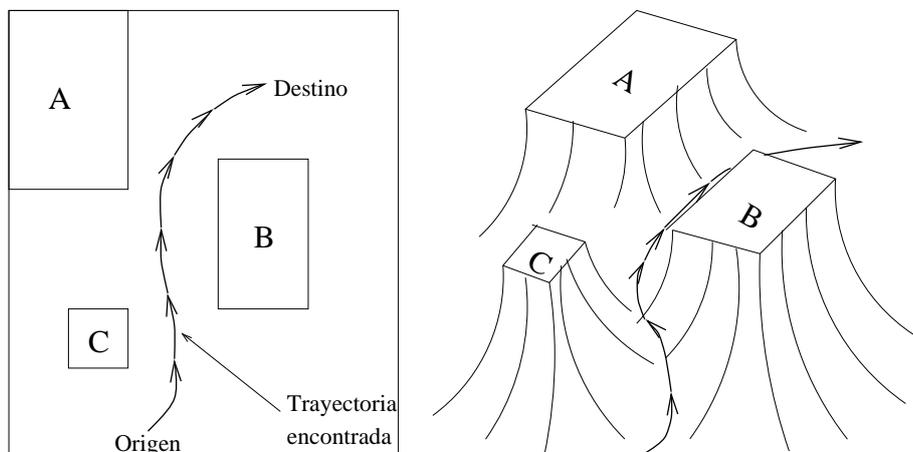


Figura 7.12: Ejemplo de campo potencial

La forma explícita de la función potencial V puede elegirse; se suele tomar una expresión de la forma $V = \frac{K}{d^r}$, donde K es una constante, y r un exponente, del orden de 2. Para encontrar la trayectoria, podemos considerar al robot como si experimentase una fuerza que le hace "caer" en la dirección de máxima pendiente local, y a la vez, una fuerza \vec{G} que le hace avanzar hacia el objetivo, la cual es añadida por el programa para conseguir llegar hasta él. La fuerza resultante sería

$$\vec{F} = \vec{E} + \alpha \vec{G} \quad (7.7)$$

donde α es una constante ajustable, que debe estar en $]0 \dots \infty[$. Si fuese 0, el robot simplemente se mueve hacia el lugar más seguro (el más lejano a cualquier objeto), mientras que si fuese ∞ , avanzaría en línea recta hacia el objetivo, ignorando (atropellando)

los obstáculos. Para mover realmente el robot se le imprime en cada momento una aceleración proporcional a la fuerza definida en la ecuación 7.7, lo que respeta hasta el final la analogía física, o bien se le da una aceleración cuya dirección sea la de la fuerza antedicha, pero cuyo módulo sea proporcional a la diferencia entre la velocidad actual y la deseada. ($|\vec{a}| = \eta(vel_deseada - vel_actual)$) Esto actúa como un controlador proporcional que tiende a mantener una velocidad de crucero fija.

Un inconveniente serio de estos algoritmos es la posibilidad de caer en pozos de potencial, es decir, lugares "seguros", pero desde los que la fuerza de atracción hacia el obstáculo es incapaz de sacar al robot. En entornos estáticos estos lugares podrían ser detectados de antemano, y una forma de evitarlos es situar un pequeño obstáculo ficticio en ellos. En entornos dinámicos, en los que los objetos móviles detectados por los sensores en tiempo real también contribuyen al potencial, la caída en estos mínimos locales no es predecible.

7.4. Tecnologías de construcción y conexión senso-motora

Un robot móvil puede considerarse esencialmente como un sistema (una colección de sensores, actuadores, y elementos computacionales) organizados de tal modo que exhiban acción inteligente en respuesta a ciertos estímulos. Tal sistema no tiene por qué ser complejo. Un buen ejemplo de esto es el robot llamado Tut-Bot. Está constituido por dos motores que hacen girar sendas ruedas independientemente. No lleva procesador, todo su hardware consiste en los drivers de potencia de los motores, y en dos contadores programables que comienzan a contar al recibir una señal externa, y durante el tiempo que dure su cuenta hasta un valor predefinido generan una señal que indica al motor que se mueva en sentido opuesto (B). El resto del tiempo cada motor se mueve hacia adelante (F). Los contadores están conectados a los sensores de choque S1 y S2, como muestra la figura 7.13.

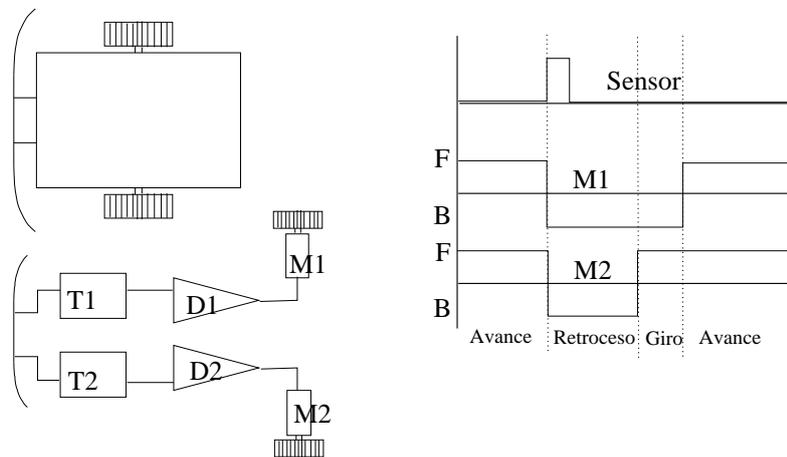


Figura 7.13: Esquema del robot Tut-Bot

Al moverse ambos motores a la misma velocidad y en el mismo sentido (F) el robot avanza. Al chocar con un obstáculo, ambos motores invierten su sentido, y el robot retrocede. Pero el contador de uno de ellos está ajustado a un tiempo menor que el del otro, con lo cual al invertirse ambos giran en distinto sentido durante un instante, lo que hace que el robot gire sobre su propio eje, con lo que es posible que ya no apunte hacia el obstáculo. A continuación ambos motores vuelven a girar hacia adelante, y se continúa el avance en línea recta. Si el tiempo (ángulo) de giro no hubiese sido suficiente, el robot volvería a chocar con el obstáculo, y repetiría la misma maniobra. Véase la figura 7.14, parte izquierda.

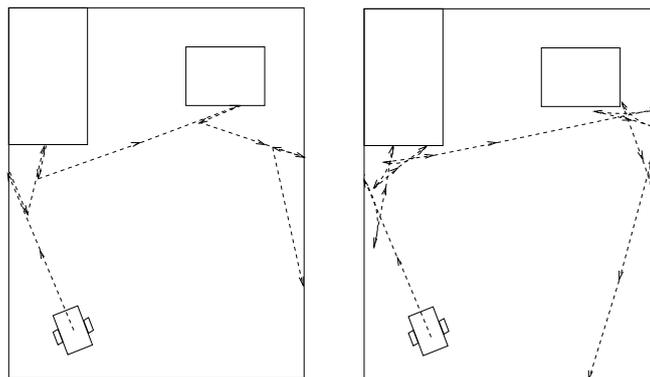


Figura 7.14: Movimiento del Tut-Bot

Si además a uno de los motores se le hiciera girar un poco más lento en el sentido B, tendríamos que el robot retrocede describiendo un arco. Esto haría que siguiese aproximadamente las paredes, como se ve en la figura 7.14, parte derecha. En ese caso, el observador externo asignaría un propósito al robot, cuando en realidad el "programa" no lo establece explícitamente. De hecho, ni siquiera se puede decir que haya un programa en sentido tradicional: todo esto se puede implantar con circuitería analógica, aunque sea mejor hacerlo digitalmente, siempre que se conserve el espíritu de simplicidad.

Este ejemplo ha servido para ilustrar una conexión directa y de bajo nivel entre percepción y acción, pero esta no es la única manera. También se puede (usando sensores más sofisticados) tratar de localizar aspectos físicos externos conocidos por el mapa prealmacenado, es decir, instanciar el modelo del mundo que el mapa representa, en el mismo estilo que se explicó en el tema 6, a partir del modelo, realizar una planificación del camino por los medios ya vistos, y seguirlo, usando *landmarks* u odometría.

Respecto a la tecnología usada, debemos referirnos a los sensores y a los actuadores. Muchos de los sensores explicados en el tema 3 pueden fácilmente ser adaptados, o usados tal cual. Entre ellos son habituales:

- Fotorresistencias o fototransistores, que se usan para implementar fototaxias (seguimiento de fuentes de luz). Su salida se conecta a un conversor A/D, o a un simple comparador, dependiendo del uso que se quiera hacer (si importa el valor de la señal, o sólo si ésta es superior a un umbral).
- Sensores de proximidad por infrarojos: Son sensibles a radiación alrededor de los 880 nm. Existen detectores encapsulados que contienen emisor y receptor; modulan la emisión, y responden sólo a ese patrón de modulación, con lo que evitan interferencias de fuentes externas de infrarojos. El hardware que necesitan es un oscilador (de cuarzo, o astable) para el emisor, y un conversor A/D o comparador para el receptor.
- Sensores piroeléctricos: son resistencias variables con la temperatura. Se usan para seguir fuentes de calor.
- Sensores de contacto por doblez: Constan de un fleje metálico con una capa de pintura conductora que varía su resistencia al doblarse. Se conectan a un conversor A/D.

- Microinterruptores de choque (*bumpers*): se usan con una palanca que los activa al chocar el robot con algún obstáculo. Se conectan directamente a entradas digitales del microcontrolador del robot.
- Sonares: El modelo más usado es el Polaroid TM, ya descrito en el tema 3. El hardware que usan es un contador, para saber el tiempo transcurrido entre la ida y la vuelta del impulso ultrasónico, y circuitos especiales para generar el pulso.
- Codificadores ópticos: normalmente de tipo incremental, se instalan en todas o algunas de las ruedas, tanto en el eje de giro como en el de guiado. Como ya se vió, requieren un hardware específico para la cuenta de pulsos, aunque ésta se pueda hacer también por software, conectado las señales de cada canal a puertos de entrada, y manteniendo un proceso dedicado a monitorizarlos.
- Giróscopos: Son análogos a los usados en los sistemas de navegación inercial de los aviones, pero algo más simples. Son raramente usados por su precio. Existen versiones electrónicas baratas basadas en sensores de estado sólido que sólo miden la velocidad de giro, pero no la orientación absoluta.
- Inclinómetros: Se basan en un codificador óptico en posición vertical con un péndulo colgado de él, o bien en una gota de mercurio sobre un platillo horizontal con contactos repartidos regularmente alrededor de ella.
- Brújulas: deberían dar la orientación absoluta usando el campo magnético terrestre. No son muy usadas, porque aunque en exteriores dan medidas aceptables, en interiores y sobre todo con campos magnéticos provocados por la circuitería o maquinaria circundante no son fiables.
- Cámaras de TV: se suelen usar modelos en miniatura, de tipo CCD. Tienen los inconvenientes de requerir un hardware más complicado (una placa digital de imagen) y generar un volumen de información difícilmente tratable en tiempo real sin hardware específico.

En cuanto a los actuadores, se suelen usar siempre motores eléctricos de CC, por su facilidad de control. Se conectan a engranajes reductores para disminuir la velocidad y aumentar la potencia. Si la corriente que necesitan no es muy alta (robots pequeños no muy pesados con dos motores pueden consumir de 0.5 a 2 A por motor), existen reguladores encapsulados que pueden proporcionarla, los cuales se controlan por modulación en anchura de pulso (PWM).

Los lazos de realimentación para el control de las ruedas se suelen realizar por software, que va leyendo los registros asociados a los codificadores, y envía una señal digital que luego se convierte en analógica y activa los dispositivos de potencia. El control es, por supuesto, siempre discreto.

Los sistemas de locomoción son variados. El modelo más común consiste en usar dos ruedas motrices independientes con sus ejes alineados perpendicularmente a la dirección de avance. La forma del robot suele ser compacta, mejor circular, para ganar maniobrabilidad.

La energía es un punto muy problemático. Hasta ahora, las baterías que podían dar suficiente corriente eran muy pesadas; hoy día existen modelos recargables de Ni-Cd más

ligeros.

Tema 8: Inteligencia en robots

8.1. Introducción

En este tema se pretende dar una visión de la relación entre inteligencia artificial (AI) y Robótica y cómo está cambiando en los últimos años. La naturaleza de este tema, y su carácter de investigación todavía no definitivamente asentada motivan el estilo de exposición discursiva y menos ordenada que en temas anteriores. Algunas de las afirmaciones o sugerencias que se exponen aquí tienen algo de posición personal, y no deben ser tomadas de modo tan estricto como el resto de estas notas.

Veremos cuál ha sido la aproximación tradicional a la Robótica desde la AI clásica, y también algún ejemplo de nuevas tendencias que tratan los problemas de la relación del robot con el mundo real de un modo claramente distinto, y en opinión de algunos, más prometedor que las aproximaciones empleadas hasta ahora. Este cambio de punto de vista es tan profundo que afecta incluso a las propias definiciones de robot y robótica, como se verá inmediatamente. En principio, se piensa en un robot como un dispositivo para facilitar, aliviar, o incluso hacer posibles ciertos tipos de trabajos indeseados, por peligrosos, repetitivos o necesitados de precisión extrema. En esta línea se entiende la definición de robot de la RIA (*Robot Industries Association*) que ya dimos en el tema 1, y que recordamos:

Robot es un manipulador reprogramable multifuncional diseñado para mover material, partes, herramientas o dispositivos especializados mediante movimientos programados variables para la ejecución de diversas tareas.

Como entonces dijimos, la definición es general, porque no restringe la tecnología usada, ni el método de programación, pero olvida el aspecto estrictamente científico de los robots: su uso como herramientas para entender los procesos de la percepción y la acción en entornos reales, no en simulaciones ni en modelos. Evidentemente, nadie quiere robots que no funcionen, o que no hagan nada útil, pero, como será comentado más extensamente después, no podrán hacer nada realmente útil si no aprenden desde el principio a hacerlo en un entorno real. Por eso, una definición alternativa de Robótica podría ser

"La ciencia que estudia los robots como sistemas que operan en algún entorno real, estableciendo algún tipo de conexión inteligente entre percepción y acción"

Descendiendo a temas más concretos, los robots se han usado hasta ahora en instalaciones industriales esencialmente como robots de montaje, soldadura o pintura de maquinaria (coches, etc.). Su característica es la repetición de las acciones preprogramadas sin variación, o a lo sumo con el uso de sensores cuya información detiene el robot en caso de colisión, o ajusta la fuerza o la inclinación del brazo. Una serie de problemas importantes relacionados con el control de bajo nivel (teoría de control de sistemas dinámicos, identificación, modelización, estabilidad, etc.) han sido formulados y resueltos para su uso en estos sistemas, y ello ha permitido una mejora técnica importante. No obstante, tal clase de robots carece por completo de cualquier comportamiento que podamos llamar inteligente, y en este sentido se acercan más a las máquinas-herramienta que a la moderna concepción de un robot.

Un avance sobre ellos lo representan los sistemas para la clasificación o el ensamblado de piezas en las que éstas llegan al entorno de trabajo en posiciones u orientaciones variables,

o con defectos. Aquí ya tenemos un cierto comportamiento inteligente, si bien la mayoría de las veces preprogramado, no adaptativo. Recordemos que la auténtica inteligencia es aquella que aprende, y mejora su eficacia en la ejecución de la tarea requerida con el paso del tiempo.

El siguiente paso lo constituirían los robots móviles, cuya principal actividad es el desplazamiento (navegación) en un entorno no conocido (al menos en todos sus aspectos) e incluso dinámicamente cambiante (gente u otros robots moviéndose). Las aplicaciones de un sistema eficiente de este tipo serían muy variadas: desde carretillas transportadoras en naves industriales, hasta robots de exploración extraterrestre (en este último caso una navegación eficiente sería competencia esencial, dado que es imposible el control en tiempo real).

El último paso, de momento solo atisbado, sería un robot de propósito general, no en el sentido de que ejecutase todas las tareas posibles (lo cual es obviamente imposible, ni siquiera los humanos lo hacemos) sino de que, en un entorno dado, fuese capaz de sobrevivir, enfrentarse a situaciones nuevas y adaptarse a ellas para seguir realizando, al menos en cierta medida, la tarea o tareas para las que se le diseñó.

8.2. La noción de inteligencia y su aplicación en Robótica

Es obvio que un sistema general del tipo apuntado en el apartado anterior requiere un cierto grado de inteligencia, y es por ello el momento de clarificar qué entendemos por inteligencia, y qué queremos decir con la expresión "un cierto grado". Frecuentemente se atribuye a los humanos el monopolio de la inteligencia, y esto es cierto si sólo consideramos como tal al razonamiento de alto nivel, el uso de lenguaje simbólico, y tareas similares. Pero no deberíamos olvidar (y la Inteligencia Artificial clásica lo ha hecho frecuentemente) que todas estas capacidades se asientan en, y necesitan de, facultades inferiores, como el proceso de la información visual (necesario para el establecimiento de relaciones espaciales), el sentido del equilibrio (necesario para la navegación en terreno irregular) o el tacto (para el ajuste de la fuerza en operaciones de prensión). Por eso, en opinión de bastantes psicólogos y etólogos, debería considerarse inteligencia tanto al razonamiento como al conocimiento de sentido común.

Es un error bastante extendido el suponer que la dificultad real estriba en el primero, y que el segundo puede, o podrá en un futuro, programarse con relativa facilidad a partir de los modelos del mundo que el razonamiento haya construido. Lo erróneo de esta afirmación puede comprobarse en el hecho de que se ha obtenido un razonable éxito en la escritura de programas que juegan al ajedrez, o razonan en un dominio particular (sistemas expertos) mientras que los intentos por construir sistemas de visión de alto nivel (capaces de interpretar lo que están viendo) ha fracasado, incluso en entornos restringidos. Ello tiene, sin duda, relación con el hecho de que el córtex visual humano ocupa casi el 20 % de la corteza cerebral, mientras que las neuronas dedicadas al razonamiento analítico parecen ser menos de un 3 % (estas cifras varían bastante con los autores).

En cualquier caso, está apareciendo claramente la idea de que un robot eficiente requerirá un buen sistema de percepción del entorno (lo cual no necesariamente significa muy complejo) que sea capaz de tomar datos de una variedad de fuentes (sensores táctiles, acústicos, olfativos, cinéticos, de distancia, y por supuesto, visuales) y fundirlos en una estructura de información coherente, tomando las piezas necesarias (pero no más) y desechando las erróneas (debidas al ruido, que está invariablemente presente en cualquier medida de cualquier sensor). Este proceso es lo que se conoce como "Data Fusion" (fusión de datos) y es uno de los temas que está despertando atención en la investigación en Robótica

de los últimos años.

Al mismo tiempo que el robot toma sus datos y los procesa, debe ejecutar una o más tareas, la principal de las cuales (especialmente para robots móviles) es sobrevivir, entendiendo por tal no quedarse parado, o atascado en un bucle infinito, o encajado en un lugar sin salida. Para ello no necesariamente tiene que conocer (en el mismo sentido que los humanos) qué se entiende por lugar sin salida. Es, a menudo, suficiente una serie de reflejos que le hagan huir de las paredes si éstas lo rodean (activan simultáneamente varios sensores situados en puntos opuestos del robot). El ejemplo del Tut-Bot suguiendo una pared mencionado en el tema 7 es apropiado aquí.

Por último, las tareas que ejecute un robot deben ser las que se espera de él (si es que se desea, como es lógico, un provecho comercial) pero hay que hacer notar que algunas de éstas, incluso muy complejas, pueden emerger como resultado de la interacción de reflejos simples, de la cooperación de varios organismos (el caso de los insectos sociales) y de la complejidad del entorno. Muchas veces tendemos a antropomorfizar estos comportamientos y a establecer categorías de intencionalidad en comportamientos que de hecho no las tienen.

8.3. La relación Inteligencia Artificial-Robótica

Desde los primeros tiempos de la Inteligencia Artificial (finales de los años 60 y principios de los 70) la relación de ésta con la Robótica ha pasado por vicisitudes diversas. En un primer momento se consideró a la Robótica parte de la IA, y así aparece en varios libros antiguos. Esto es, quizá, herencia de la Cibernética, que trataba de entender los organismos vivos (incluidos los inteligentes) como sistemas de control, frecuentemente organizados en una jerarquía que, idealmente, debía explicar el funcionamiento de sistemas arbitrariamente complejos, incluso la biosfera completa, en términos de sistemas que tratan de conservar el equilibrio (el valor de ciertas variables de estado). Como ejemplo, la tortuga de Walker, un mecanismo descrito por éste en 1951, se construía como un dispositivo móvil con ruedas y detectores luminosos que seguía o huía de fuentes de luz. Reflejos elementales "programados como diferentes cableados de sus conexiones permitían exhibir comportamientos complejos, especialmente cuando dos o más de estos dispositivos dotados de luces interactuaban.

Después la Robótica comenzó a tomar un camino más técnico, centrándose especialmente en problemas de control como los ya formulados desde muy antiguo (la clepsidra, el telar de vapor) pero más complejos, que son abordados con herramientas matemáticas como las transformadas integrales. Desde hace varios años, y con la introducción del control por computadores digitales, han aparecido los problemas de discretización que han sido elegantemente abordados mediante, vg., la transformada Z, que permite generalizar muchos de los resultados conocidos para sistemas contínuos. En cuanto al uso de sensores, o bien se ha hecho acoplándolos directamente a un bucle de realimentación (robots de soldadura dotados del llamado control híbrido) o bien se ha procesado su información según un modelo basado en la idea de la mente que tiene la Inteligencia Artificial clásica.

8.3.1. Robótica clásica

El modelo de inteligencia que la AI clásica pretende contruir está basado en el paradigma, denotado por Haugeland como GOFAI¹ que se sustenta en la hipótesis del Sistema Físico de Símbolos (PSSH) de Newell y Simon, la cual dice que:

¹Good Old Fashioned Artificial Intelligence, (la buena y vieja Inteligencia Artificial)

"Un sistema físico de símbolos posee los medios necesarios y suficientes para producir acción inteligente general"

Esto significa que, si construimos un sistema de símbolos (proposiciones, u otros) que modelicen suficientemente bien un aspecto de la realidad, podremos razonar sobre los modelos, de tal modo que el resultado nos indique el comportamiento del sistema real, y como influirá sobre él cualquiera de nuestras actuaciones. Si el modelo se trata de un modelo del mundo físico (o incluso no físico) en su conjunto, podríamos obtener inteligencia general (en el sentido humano del término). Véase la figura 8.1

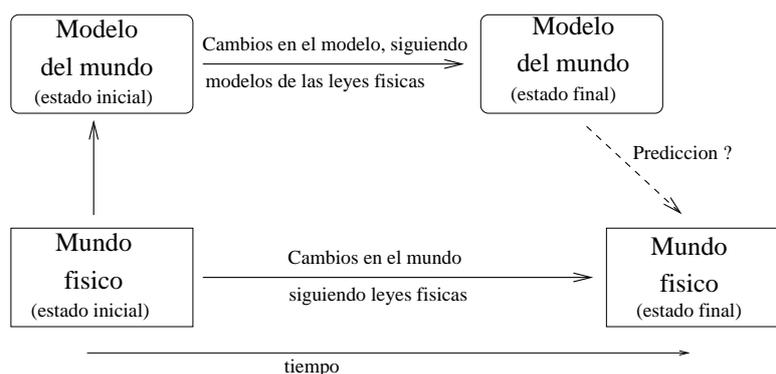


Figura 8.1: Sistemas de símbolos y su relación con el mundo

La aplicación de esto en Robótica da lugar al llamado "*sense-think-act cycle*" (ciclo de recoger información-pensar-actuar) que a partir de los datos de los sensores, los preprocesa para eliminar ruido y extraer los aspectos relevantes, los cuales son pasados a programas que tratan de encajar (instanciar) estos datos en los modelos que conocen, para a continuación, y de acuerdo con el estado presente del mundo, planificar la acción o secuencia de acciones a ejecutar, que son convertidas en comandos para los actuadores, que el sistema de control se encarga de ejecutar fielmente. Clarificaremos esto con un ejemplo:

Hay que poner dos piezas una sobre otra, en un entorno en el que hay varias piezas más. Para ello, dos cámaras toman sendas imágenes, que son filtradas para eliminar ruido y extraer los bordes de los objetos; éstos son preprocesados para obtener dibujos de líneas. Un programa interpreta a partir de ellos la forma tridimensional de cada objeto, funde las representaciones aportadas por ambas cámaras en una sólo estructura de datos representando a cada objeto, e intenta encajarla con el modelo (posiblemente de CAD) que le fue introducido. Una vez encajado, determina (conociendo la transformación de coordenadas de las cámaras) la situación del objeto que ha de asir en el sistema de coordenadas del robot; entonces un programa planificador de trayectorias opera con los modelos geométricos de todas las piezas y del brazo, y sus situaciones actuales, y determina qué secuencia de posiciones debe seguir el brazo para asir la primera pieza, llevarla sobre la segunda sin que ni ella ni el brazo choquen con ninguna otra, y dejarla en su lugar. Esta secuencia de posiciones es transformada a una secuencia de cuentas para los motores del robot usando la cinemática inversa, la cual es por fin enviada y ejecutada. Una descripción de este paradigma aparece en las publicaciones de Lozano-pérez y otros sobre su sistema de ensamblado Handey, que veremos más extensamente después.

Esto es lo que se conoce como descomposición funcional y vertical de las tareas, y se expresa en el esquema adjunto.

Los inconvenientes de la construcción de un modelo simbólico para estos casos deberían ser evidentes simplemente observando la longitud y complejidad del proceso mencionado.

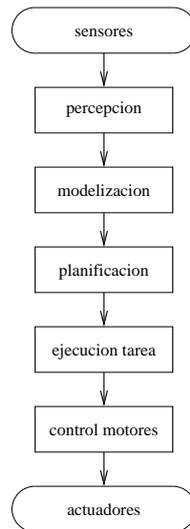


Figura 8.2: Descomposición funcional y vertical

En primer lugar, los datos llegan con ruido, tanto el propio de la digitalización, como el debido a sombras, imperfecciones en las piezas, ocultaciones, puntos de vista anómalos que provocan coincidencias de ejes o bordes sin correspondencia real, falta de contraste, etc. Lógicamente, datos imprecisos crean descripciones simbólicas erróneas, y es difícil encajarlas en los modelos almacenados. Para evitar esto se puede dotar al modelo de más grados de libertad y tolerancias en los valores de las variables, pero ello implica que una misma descripción puede entonces encajar en varios modelos, creando ambigüedad. Esta ambigüedad se intenta resolver tomando más aspectos (datos) del objeto, como color, o textura de la superficie; pero ello implica más cálculo para su extracción y modelos más complicados (dado que, por supuesto, estos nuevos aspectos también están afectados de un error, y sus valores en cierto rango de tolerancias).

Otro inconveniente es que cuando alguna de las descripciones proporcionadas por el bajo nivel no encaja en ningún modelo hay que volver a tomar los datos y analizarlos, orientando el proceso a la resolución de los aspectos que resultaron ambiguos.

Y por último: una descomposición secuencial implica que cada paso no puede empezar hasta que el anterior haya terminado, y que el fallo de uno sólo de ellos hace fracasar el proceso completo: una cadena es tan fuerte como el más débil de sus eslabones. Por ejemplo, existen ingeniosos métodos para la interpretación de dibujos de líneas pero ninguno realmente efectivo para la construcción de tales dibujos a partir de imágenes reales tomadas por cámaras.

Todos estos inconvenientes, unidos al hecho de que en quince años ningún sistema clásico haya resultado realmente eficiente en tareas reales de ensamblado, ha motivado que algunos investigadores se plantearan la posibilidad de usar otros métodos basados en una concepción diferente de la Inteligencia Artificial.

8.3.2. Robótica comportamental

Una visión que siempre ha estado presente en IA, si bien de modo minoritario e incluso un tanto herético, es la que considera la inteligencia no exclusivamente como sucesión de procesos mentales, sino como la organización que emerge de la relación entre un organismo y su entorno. Todos admitiríamos sin discusión que la inteligencia no puede desarrollarse sin una relación con el exterior, que se realiza a través de los órganos de los sentidos, pero

nos cuesta más admitir que no pueda sostenerse sin ellos. Sin embargo, los modelos que hacemos del mundo están fuertemente influenciados por las percepciones sensoriales; esto puede comprobarse analizando la forma en que aprenden a ver los raros casos de invidentes de nacimiento que mediante cirugía o tratamientos recuperan la visión ya adultos.

También es interesante analizar los comportamientos de organismos inferiores, como las termitas, en las que cada uno de los individuos no conoce en modo alguno el plano del termitero (es decir, no tiene idea de qué es lo que está construyendo) pero de hecho una estructura muy compleja, con partes destinadas a funciones específicas, emerge de la cooperación de todas las termitas y de su interacción con el entorno. En estos casos el observador humano tiende a asignar propósitos al comportamiento observado (están haciendo esta cámara para albergar a la reina, etc..) pero ¿ cómo puede decirse tal cosa, si no existe representación simbólica ni siquiera de la morfología de lo construido? (Cuánto menos de su función. Y si alguien piensa que sí hay tal representación, se le reta a que diga dónde y cómo está almacenada).

A la hora de construir un robot, la robótica comportamental prescindiría de programarle explícitamente un comportamiento orientado a un propósito. Mas bien, programaría reflejos o comportamientos muy simples, pero eficaces, que en conexión directa con los sensores y los actuadores, generasen mediante su interacción el comportamiento requerido. Los ejemplos más espectaculares hasta ahora son los trabajos de Rodney Brooks y sus colaboradores, tanto sus robots insectoides como Herbert, un robot móvil para recoger latas vacías. Cada uno de estos módulos de software más hardware que realiza una función elemental se conecta con los sensores, los actuadores, y posiblemente con otros módulos que le pasan una información mínima, o ninguna en absoluto, simplemente lo activan o inhiben. A cada una de estas piezas se la ha empezado a llamar "módulo comportamental", y a la arquitectura de control distribuido que las conecta, arquitectura de subsumción, en el sentido de que cada módulo en muchas ocasiones subsume los comportamientos del anterior, de modo que si alguno falla, es posible que el robot sea todavía capaz de ejecutar algo de su comportamiento, o al menos no pararse completamente; esto se conoce como degradación suave (*"graceful degradation"*).

Por todo lo dicho, se entiende que la robótica comportamental busque habitualmente su fuente de inspiración en la biología, en organismos vivos completos, o bien en comportamientos o reflejos de los mismos que puedan ser estudiados aisladamente sin perder demasiada funcionalidad. Inmediatamente veremos un ejemplo de esto, Herbert.

Es igualmente obvio que un objetivo sustancial deberá ser la conexión de percepción y acción al nivel más bajo posible, dejando que el módulo que los conecte se enfrente al ruido y al problema de la fusión de datos, solucionándolos a ese nivel; esto no es tan extraño como a primera vista puede parecer, dado que al no construir estructuras de datos complicadas los errores introducidos por el ruido no se propagan desmesuradamente, y además, se suelen diseñar los módulos comportamentales con bucles de realimentación para autocorregir sus resultados.

Un esquema de cómo organiza la robótica comportamental una aplicación típica (un robot móvil) puede verse en esta figura:

8.4. Estudios de caso

Seguidamente veremos dos sistemas que pueden ser considerados casi como los más representativos de cada una de las tendencias antes citadas.

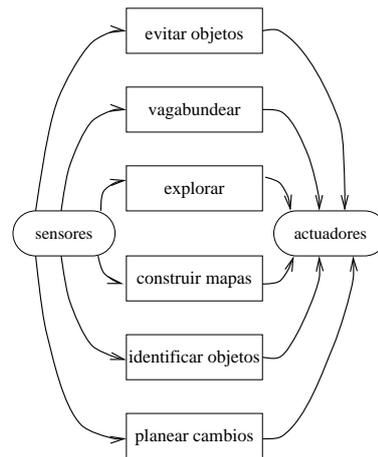


Figura 8.3: Descomposición horizontal

8.4.1. HANDEY: un sistema clásico de ensamblado

Handey es, probablemente, el prototipo de sistema robótico clásico, basado en la descomposición funcional y vertical.

Handey es esencialmente un sistema de planificación. Toma como entrada modelos del mundo construidos por el usuario con sistemas CAD, o a partir de sensores (aunque nunca detalla exactamente cómo), y ejecuta operaciones del tipo *pick & place* (recoger y dejar) generadas por el propio sistema a partir de la especificación por el usuario de dos estados diferentes para el modelo del mundo.

Las operaciones *pick & place* no son las únicas útiles, aunque son las fundamentales en robots de ensamblado; también son importantes las operaciones de ensamblado preciso atendiendo a sensores de fuerza, soldaduras, etc.

Handey pretende ser un sistema de programación de robot a nivel de tarea que, dado el objetivo de conseguir que el mundo quede en un cierto estado dado por un modelo concreto (de poliedros, como luego veremos) genera una secuencia de movimientos apropiados para ejecutar la tarea. Exactamente: Handey genera movimientos para operaciones de recoger y dejar, en entornos modelizados con precisión y en robots de hasta 6 g.d.l. con pinzas de garras plano-paralelas. En todo momento Handey opera con modelos, y de modo secuencial. Lo que sus programadores (T. Lozano-Pérez, Jones, Mazer, O'Donnell y contribuciones de Latombe y otros) dan como propiedad clave es que si el usuario cambia el entorno, una vez modelizado el cambio, el sistema será capaz de generar nuevos movimientos compatibles con el nuevo entorno.

Handey ha sido probado en numerosas tareas, desde apilar cubos de madera hasta ensamblar piezas de motores eléctricos, y para diferentes tipos de robots industriales; incluso se pueden coordinar dos brazos en el mismo espacio de trabajo. El único caso, sin embargo, en que se reporta haberlo usado con un sistema sensorial es la que describe su conexión a un sistema de luz estructurada que genera un mapa de profundidad.²

Es importante señalar qué cosas **no** hace Handey.

- No pretende ser un sistema industrial utilizable (al menos, todavía), sino un experimento.

²Este sistema es similar al descrito en el tema 3, pág. 3/ 13

- No pretende ser rápido (no está optimizado).
- No hace control, pero genera secuencias de puntos en el espacio de articulaciones que se pueden enviar como entradas al sistema de control.
- No genera movimientos restringidos ni condicionados (las órdenes `ON <condition>` o `WITH <condition>` de VAL).
- No genera movimientos que reduzcan la incertidumbre del entorno (*complained motions*, movimientos de las partes sin ser capturadas que las arrastren a una posición conocida).
- No admite variaciones del entorno durante la generación de los movimientos, es decir, no maneja el sistema en tiempo real.
- Requiere que algún otro sistema le genere un modelo completo del entorno, y supone que tal sistema existe, o existirá.

De todas estas restricciones, sólo la última (y, en menor medida, la penúltima) es realmente grave; hasta el momento, no existe un sistema fiable de este tipo, y aún se ignora si la investigación de reconocimiento visual por modelización está orientada en la dirección correcta. El uso de sensores distintos de las cámaras, como los sistemas láser o de luz estructurada antes mencionados, son extremadamente lentos y aún no están suficientemente desarrollados.

Vemos ahora cuáles son las restricciones que habrán de cumplirse (y que Handey comprobará) para que pueda realizarse una tarea de recoger-dejar.

- Debe existir un camino libre de colisión y cinemáticamente alcanzable desde la configuración inicial del robot hasta la posición de agarre.
- El agarre debe ser estable.
- Ninguna parte del robot debe colisionar con nada, ni en la posición de agarre, ni en la de depósito.
- Las posiciones de agarre y depósito deben ser alcanzables.
- Debe existir un camino libre de colisión y cinemáticamente alcanzable desde la posición de agarre hasta la de depósito, teniendo en cuenta al objeto asido por la mano.

Si cualquiera de ellas no puede cumplirse, la operación será imposible.

Las partes en que conceptualmente puede dividirse Handey son:

- + Un planificador de movimiento grueso (en espacio libre).
- + Un planificador de agarre.
- + Un planificador de reorientación y agarre sucesivo.

+ Un coordinador multibrazos.

En cada operación de recoger-dejar el planificador de agarre determina, de acuerdo a la geometría del objeto a agarrar, la de los objetos cercanos, y la de la mano del robot, la forma óptima de asirlo. Igualmente, y basándose en consideraciones similares, determina un punto cercano al punto de depósito del objeto desde el que moverse en forma segura para depositarlo.

El planificador de movimiento grueso determina el camino entre el punto de aproximación al agarre y el punto de inicio del depósito.

A veces el depósito requiere que el objeto sea agarrado de otra manera, de modo que la mano no estorbe para dejarlo. Es necesario entonces buscar un punto libre del espacio de trabajo donde depositarlo temporalmente y recogerlo de nuevo en la forma apropiada para el depósito final; esto lo hace el planificador de reorientación cuando es necesario.

El coordinador multibrazos se encarga de que, si hay dos brazos, éstos no choquen.

Indiquemos brevemente cómo hace Handey estas cosas. Internamente, se usan varios conceptos importantes:

Parte: Un modelo geométrico 3D de un objeto, y el lugar en que está.

Mundo: Un modelo geométrico 3D de una escena entera.

Robot: Un modelo geométrico 3D del manipulador y su cinemática, así como los límites de cada variable de articulación.

Objetivo: Un estado particular del modelo del mundo que se desea alcanzar.

Plan: Una secuencia de movimientos elementales del robot y su pinza que sitúen al mundo en el estado objetivo.

Las llamadas a las funciones de Handey se hacen usando estas estructuras. P.ej.;

el planificador de movimiento grueso se invoca con:	<code>Move(goal,robot,world)</code>
el planificador de agarre con:	<code>Grasp(goal,robot,world,depart?)</code>
donde <code>depart</code> es una variable booleana que indica agarre o depósito	
el planificador de reorientación con:	<code>Regrasp(goal,robot,world)</code>
y el coordinador multibrazos como:	<code>Coordinate(plan1,plan2,world)</code>

Los detalles internos de cada módulo exceden el alcance de esta exposición. Únicamente veremos cómo Handey representa (o mejor, qué representación maneja) para el mundo.

Todo objeto en Handey es representado por poliedros. Un poliedro es un sólido limitado por caras planas. Cada cara plana se caracteriza por el vector asociado a su plano. Es necesario conocer vértices, bordes y caras de cada polígono de cada poliedro. El propio robot es modelizado de esta manera, pero además se conoce su cinemática directa e inversa (medidas, número de articulaciones y tipo, y sus límites).

La colección de modelos poliédricos de todas las partes presentes en el espacio de trabajo accesible al robot, y del robot mismo, forman el modelo del mundo.

Para planificar cualquier movimiento sin colisión hay que hacer las intersecciones de todos los polígonos. Una forma de evitar esto en lo posible es el uso del espacio de configuración (C-space). En él, cada objeto se caracterizará por la posición de uno de sus puntos arbitrariamente escogido, sea p_A , y la orientación θ de una de sus líneas, igualmente arbitraria. Véase la figura 8.4.

Para cada objeto en cada entorno, hay valores de los parámetros p_A y θ que pondrían alguno de sus puntos dentro de alguno de los obstáculos, B_1, \dots, B_n . Entonces, se define el espacio de configuración de obstáculo del objeto A debido a un obstáculo B_j cualquiera

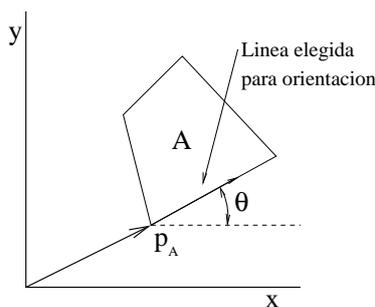


Figura 8.4: Parámetros del espacio de configuraciones

como:

$$CO_A(B_j) = \{C / (A)_C \cap B_j \neq \emptyset\}$$

siendo C una terna de valores de la forma (p_{Ax}, p_{Ay}, θ) y $(A)_C$ el objeto A situado y orientado como indica C . Es claro que $\overline{\bigcap_j CO_A(B_j)}$ es el espacio libre³

Como ejemplos, sea A un círculo de radio r , y sean los obstáculos B_1, B_2, B_3 otros tres círculos, como se muestra en la figura 8.5. Es obvio que aquí el parámetro de orientación θ no es necesario. En este caso, el espacio de configuración se muestra en la misma figura (parte derecha); si cada obstáculo tiene su centro en (x_j, y_j) y un radio r_j , CO_A será la unión de los círculos de centro en (x_j, y_j) y radio $r_j + r$. Entonces, el objeto A puede considerarse reducido a un punto.

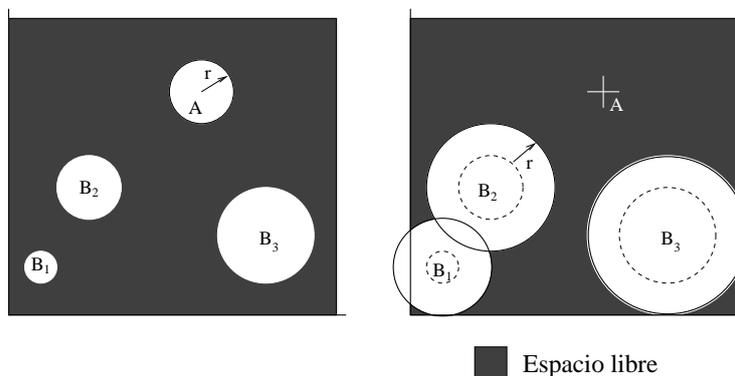


Figura 8.5: Un ejemplo de espacio de configuraciones

Como segundo ejemplo, sea A de nuevo un círculo, y el obstáculo un polígono. CO_A es como se muestra en la figura 8.6.

En el caso de que el objeto A pudiese rotar (o más bien, fuese diferente según su orientación, lo que ocurre en todas las figuras excepto en el círculo) el cálculo de CO_A es mucho más laborioso, porque hay que desplazar el objeto A en todas sus orientaciones distintas. No obstante, nótese que esto puede computarse antes de empezar el movimiento, con lo que durante la tarea lo único que habría que computar es el movimiento de un punto por el espacio libre que haya resultado. Por supuesto, esto exige un entorno estático durante toda la tarea, según se indicó en las restricciones de Handey.

En 3 dimensiones la computación del C -space es aún más compleja. Existen algoritmos paralelos para computar con rapidez el espacio de configuración de un poliedro respecto a otro conjunto de poliedros.

³Para cualquier conjunto X , \overline{X} denota su complementario.

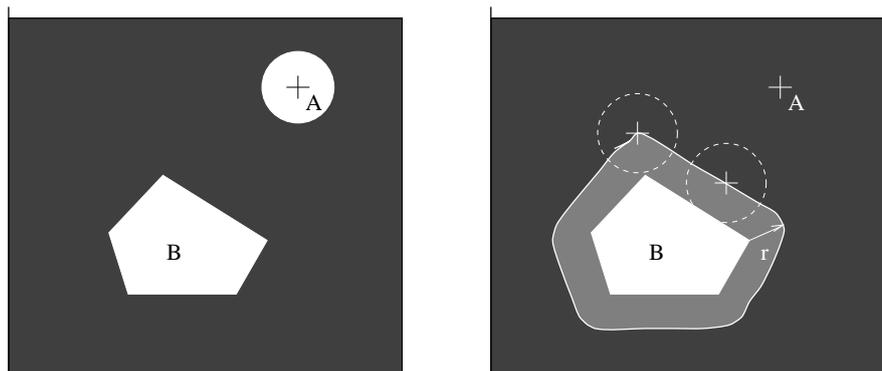


Figura 8.6: Otro ejemplo de espacio de configuraciones

8.4.2. Ejemplo comportamental: Herbert

Dos de los casos más representativos de la descomposición horizontal antes descrita son un robot que circula por un laboratorio recogiendo latas de refresco vacías, y los robots insectoides del prof. R. Brooks. Ambos fueron desarrollados en el MIT (*Massachusetts Institute of Technology*), pues es allí, junto con la Universidad de Edimburgo, donde nacieron las ideas de esta visión.

Herbert

Este robot, diseñado por Jonathan Connell como banco de pruebas para su tesis en el MIT, dirigida por Rodney Brooks, tiene la tarea de deambular por un laboratorio usual, en el que nada ha sido movido o cambiado especialmente, para recoger latas de refresco vacías y llevarlas a un contenedor. Nótese que el propósito de este robot es totalmente específico, aunque su filosofía de diseño trata de ser general. Precisamente la idea de la aproximación comportamental es que los reflejos elementales, al igual que en la naturaleza, son específicos, y dependen del hábitat del robot y de la tarea que debe desempeñar, aun cuando algunos de ellos puedan ser comunes a algunos o casi todos los robots similares.

Herbert está construido enlazando de modo muy hábil módulos elementales, cada uno de los cuales tiene entradas y salidas. Las entradas se conectan directamente a señales dadas por los sensores, y las salidas son o bien comandos para los actuadores, o bien señales que inhiben la salida de otros módulos, o la sustituyen por la suya propia, durante un determinado tiempo fijo. Dentro de Herbert todo esto se implanta por hardware, haciendo correr cada módulo (o grupo de unos pocos de ellos) sobre microprocesadores de 8 bits.

Su estructura mecánica está formada por un cuerpo móvil sobre el que se sitúa un brazo con articulaciones tipo paralelogramo con dos grados de libertad; la mano es una pinza de tipo simple. Véase la figura 8.7

La sensorización para la navegación se realiza con dos anillos, de 14 sensores de ultrasonidos por tiempo de vuelo cada uno, colocados a dos diferentes alturas. La mano está dotada de dos microinterruptores para monitorizar el contacto con la mesa, dos sensores de infrarrojos apuntando hacia el centro, y uno entre dedo y dedo, para controlar si hay un objeto en posición adecuada para ser cogido, además de un sensor de fuerza de tipo ON/OFF (se activa si se rebasa un umbral). Véase la figura 8.8

Además, hay un sistema de visión constituido por un láser y una lente cilíndrica que hace que se proyecte una línea de luz frente al robot, en el área capturada por una cámara de televisión; su señal es recogida y procesada por un hardware específico que realiza opera-

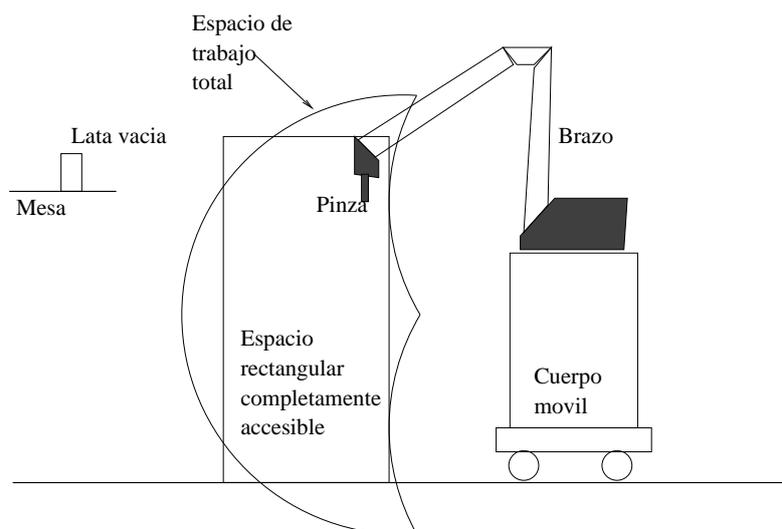


Figura 8.7: Esquema de Herbert

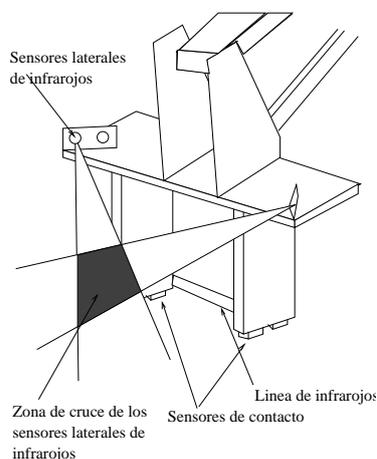


Figura 8.8: La mano de Herbert

ciones muy simples, como detectar el punto más brillante (que se supone corresponde a la línea proyectada por el láser). Véase la figura 8.9.

En cada momento, varios módulos se activan de acuerdo a las señales de los sensores a los que están atentos: p. ej., el módulo de agarrar cierra la pinza si hay algo ocluyendo la línea del sensor de infrarojos entre los dedos; pero al mismo tiempo otro módulo lee los sensores de fuerza para el cierre de la pinza, inhibiendo la señal del anterior cuando la fuerza se hace excesiva. De modo similar, hay módulos controlando que un objeto con forma aproximada de lata se vea en la escena, e instruye a la plataforma completa del robot para moverse hasta que dicho objeto aparezca en el centro del campo visual; en lugar de tratar de averiguar la posición de la lata respecto a algún sistema de coordenadas externo, el robot se sitúa respecto al objetivo de modo que se pueda coger. La aparición de la lata (o de lo que se supone que es tal) en el centro del campo visual activa la compleja red de módulos que ejecutan el comportamiento de agarre. Según Connell y Brooks, este sistema trabaja sin modelos, puesto que no se le ha introducido ninguna descripción explícita de la lata, ni hay ninguna estructura de datos que la contenga. Esto, según otros, no es totalmente cierto: en realidad, hay un modelo, que está implícito en las conexiones entre módulos y en sus condiciones de activación; pero es, al menos, lo que podríamos llamar un "modelo

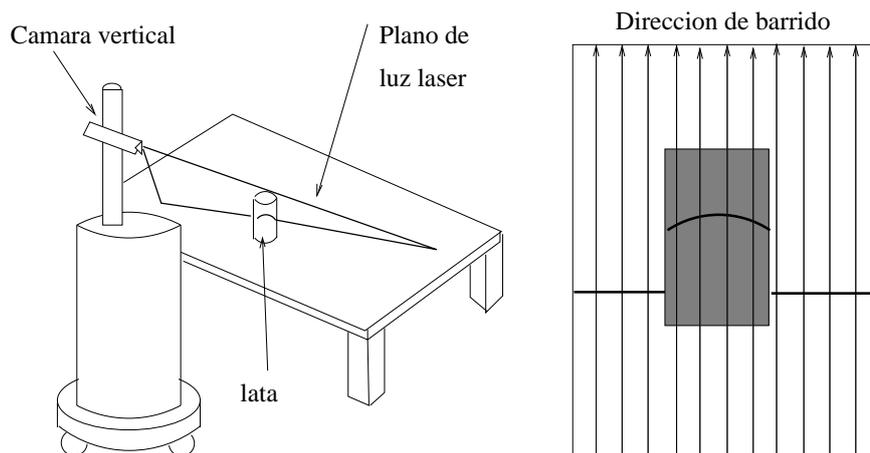


Figura 8.9: Sistema de visión de Herbert

mínimo": es decir, un concepto suficientemente general como para que cualquier objeto con forma aproximadamente de lata encaje en él. En este caso concreto, los módulos de visión detectan objetos que en la imagen capturada aparezcan como situados a una profundidad (distancia desde el punto de observación) menor que su entorno, y para los que se observe en la imagen una proporción dada entre su altura y su anchura.

Herbert efectivamente funcionó en un laboratorio real, y representó uno de los más importantes argumentos en favor de las arquitecturas comportamentales. Pero hay que hacer notar que la rigidez interna de su construcción le hace útil sólo para una tarea, y que la compleja organización de las conexiones entre sus módulos representa mucha más "inteligencia" de la que sus diseñadores desearían admitir.

Apéndices

A. Algunas identidades trigonométricas útiles

Se detallan a continuación las fórmulas trigonométricas que pueden ayudar en la resolución de la cinemática inversa de manipuladores. Las cuatro últimas no suelen figurar las tablas más simples, pero son sin embargo expresiones muy comunes en estos casos.

$$\sin(-\alpha) = -\sin \alpha$$

$$\cos(-\alpha) = \cos \alpha$$

$$\sin^2 \alpha + \cos^2 \alpha = 1$$

$$\sin(2\alpha) = 2 \sin \alpha \cos \alpha$$

$$\cos(2\alpha) = \cos^2 \alpha - \sin^2 \alpha$$

$$\sin \alpha = \frac{\tan \alpha}{\sqrt{1+\tan^2 \alpha}}$$

$$\cos \alpha = \frac{1}{\sqrt{1+\tan^2 \alpha}}$$

$$\sin(\alpha + \beta) = \sin \alpha \cos \beta + \cos \alpha \sin \beta \quad \cos(\alpha + \beta) = \cos \alpha \cos \beta - \sin \alpha \sin \beta$$

$$\sin(\alpha - \beta) = \sin \alpha \cos \beta - \cos \alpha \sin \beta \quad \cos(\alpha - \beta) = \cos \alpha \cos \beta + \sin \alpha \sin \beta$$

$$\tan(2\alpha) = \frac{2 \tan \alpha}{1 - \tan^2 \alpha}$$

$$\tan(\alpha + \beta) = \frac{\tan \alpha + \tan \beta}{1 - \tan \alpha \tan \beta}$$

$$\tan(\alpha - \beta) = \frac{\tan \alpha - \tan \beta}{1 + \tan \alpha \tan \beta}$$

$$\sin \alpha + \sin \beta = 2 \sin \frac{\alpha + \beta}{2} \cos \frac{\alpha - \beta}{2}$$

$$\sin \alpha - \sin \beta = 2 \cos \frac{\alpha + \beta}{2} \sin \frac{\alpha - \beta}{2}$$

$$\cos \alpha + \cos \beta = 2 \cos \frac{\alpha + \beta}{2} \cos \frac{\alpha - \beta}{2}$$

$$\cos \alpha - \cos \beta = -2 \sin \frac{\alpha + \beta}{2} \sin \frac{\alpha - \beta}{2}$$

B. Bibliografía comentada y tabla por temas

Seguidamente se da la referencia bibliográfica y un breve comentario de los libros de los que se ha extraído información para la confección de las clases. Se especifica cuáles son los libros principales que el alumno debe consultar, así como los que pueden servirle para ampliar conocimientos en algún tema concreto, e incluso aquellos que han sido usados sólo en un punto específico. Seguidamente se dará una tabla indicando el libro base y los otros libros útiles para cada tema, siguiendo las abreviaturas usadas en los comentarios. Además, se establece una clasificación en libros genéricos ([G]), específicos de cierto(s) tema(s) ([T]) y de divulgación ([D]).

[G],McK "Introduction to Robotics", Philip J. McKerrow, *Addison-Wesley*, ISBN 0-534-914370-5.

Este es el libro base del curso; de hecho, es probablemente el libro de texto más completo actualmente disponible. La explicación es, en general, clara y ordenada, aunque peca de farragosa en ciertos capítulos. Es el libro cuya adquisición podría recomendarse con mayor provecho a los alumnos interesados en conservar un buen libro genérico.

[G],KCN "Robotic Engineering: An Integrated Approach", R.D. Klafter, T.A. Chmielewski, M. Negin. *Prentice Hall*, ISBN 0-13-468752-3

Este libro contiene la mayor parte de lo tratado por McKerrow, excepto robots móviles; en algunos capítulos (sensores, actuadores y control) es incluso más extenso. Desde el punto de vista de una ingeniería informática, da una visión un poco sesgada hacia la ingeniería de control o electrónica. Los alumnos de ingeniería electrónica pueden planteárselo como adquisición posible si tienen especial interés en la asignatura.

[G],FGL "Robótica: Control, Detección, Visión e Inteligencia", K.S. Fu, R.C. González, C.S.G. Lee. *McGraw-Hill*, ISBN 84-7615-214-0

Durante mucho tiempo, el libro de Robótica por antonomasia. Hoy ha quedado un poco desfasado, pero sigue siendo una fuente de información excelente y muy clara en su exposición de los temas básicos. Está disponible en español.

[G],GWNO "Robótica Industrial: Tecnología, programación y aplicaciones", M. P. Groover, M. Weiss, R. N. Nagel, N. G. Odrey, *McGraw-Hill*, ISBN 0-07-024989-X

También un libro genérico, de corte más bien ingenieril, pero con buenos ejemplos concretos de programas de robot. También en español.

[T],Cra "Introduction to Robotics: Mechanics and Control", J. J. Craig. *Addison-Wesley*, ISBN 0-201-010336-5

Probablemente el más sintético y preciso de los libros sobre mecánica de robots. Comienza desde la base y llega a un nivel elevado. Tiene el inconveniente de que sólo cubre uno de los temas del curso.

[T],GV "Control Systems Design and Simulation", J. Golten, A. Verwer, *McGraw-Hill*, 0-07-707412-2

Este no es un libro de Robótica, sino de Teoría de Control. Se recomienda aquí como ayuda a los alumnos que pudieran no haber visto esta materia anteriormente, y como

repaso para los demás, de modo que puedan entender el tema 5. Se ha preferido, p. ej., al libro clásico de Ogata por su mayor simplicidad y claridad, pero, a diferencia de éste, no es'ta traducido.

[T],Tay "Control Robótico", P. M. Taylor, *Eds. Ceac*, ISBN 0-333043821-3

Un pequeño librito genérico, pero con un capítulo de control de motores muy claro y breve.

[T],AS "Robot Analysis end Control", H. Asada, J. J. E. Slotine, *John Wiley & Sons*, ISBN 0-471-83029-1

Trata exclusivamente cinemática, dinámica y control de robots a un nivel más elevado que el impartido en el curso. Sólo para aquellos alumnos con especial interés en estos temas.

[T],vTu "Autonomous Motion on Wheels", P. van Turenout, *Universiteitsdrukkerij T.U. Delft (Servicio de publicaciones, Univ. de Delft, Holanda)*, ISBN 90-9006530-X

Edición de una tesis doctoral de la Universidad de Delft (Holanda) que contiene en su inicio una exposición de la cinemática de robots móviles más clara y ordenada que la de McKerrow.

[T],JF "Mobile Robots: from inspiration to implementation", J. L. Jones, A. M. Flynn, *A K Peters, Wellesley, Massachusetts*, ISBN 1-56881-011-3

Precioso librito mezcla de manual de hardware y exposición científica, que describe todos los detalles necesarios para la construcción de robots móviles, así como ideas para su programación basadas en la visión reactiva.

[T],LJMO "HANDEY: A Robot Task Planner", T. Lozano-Pérez, J. L. Jones, E. Mazer, P. O'Donnell, *MIT Press, MIT Press Series in AI*

El informe completo sobre Handey escrito por sus diseñadores. Util para profundizar en el tema de robótica funcional.

[T],Bra "Robot Motion: Planning and Control", M. Brady et al. (eds.), *MIT Press, MIT Press Series in AI*, ISBN 0-262-02182-X

Es un conjunto de artículos sobre programación y control de robots desde las trayectorias hasta el razonamiento geométrico. El nivel de la exposición supera un poco al del curso.

[D],Min "Robótica: La última frontera de la alta tecnología", M. Minsky et al., *Ed. Planeta*, ISBN 84-320-7885-9

Un libro escrito para el público en general, pero con rigor. Hoy tiene un interés esencialmente histórico, ilustrativo para comparar las expectativas existentes a finales de los 60 con los logros actuales.

[D],Oll "Evolución y perspectivas de la Robótica", A. Ollero, *Sec. de publicaciones Univ. de Málaga*, ISBN 84-7496-216-1

Una visión divulgativa, pero tratada con rigor, desde el punto de vista español.

En la siguiente tabla, los números entre paréntesis representan el tema o temas del libro en cuestión que se deben consultar. No hay número cuando todo el libro versa sobre ese tema.

Tema	Libro(s) básico(s)	Otros libros de interés
1	FGL (1)	McK (1), KCN (1), Min , Oll
	Cinemática: KCN (8)	McK (3,4,5), FGL (2), Craig ⁽¹⁾ (2,3,4), AS (2)
2	Dinámica: FGL (3)	McK (7), Craig (6), AS (5)
	Trayectorias: Cra (7)	FGL (4), Bra (4)
3	KCN (5), McK (10)	FGL (6)
4	KCN (3,4)	Tay (3)
5	KCN (4)	McK (11), FGL (5), GV , Tay (4), AS (6)
6	McK (9,12)	FGL (9), KCN (7), GWNO (8,9)
7	McK ⁽²⁾ (8)	vTu (3), JF
8	LJMO	FGL (10), GWNO (10)

Notas:

(1): El libro de Craig usa la formulación de Denavit-Hartenberg modificada, que no usa el eje z del sistema del mundo como primer eje de articulación; el primer eje real no es, entonces, el 0 sino el 1, y todos los índices i en la notación usual son en Craig $i + 1$. Esto puede provocar gran confusión si no se advierte.

(2): El tratamiento de la cinemática de robots móviles por McKerrow es confuso; hay errores de imprenta y pasos no explicados que hacen muy difícil seguir el razonamiento. De ahí la necesidad de haber usado una referencia tan extraña como una tesis doctoral (van Turenout) para tratar este tema.

