

*Aníbal Ollero Baturone*



# ROBÓTICA

## Manipuladores y robots móviles



**marcombo**  
BOIXAREU EDITORES

Contiene  
CD-ROM









# **ROBÓTICA**

## **Manipuladores y robots móviles**

*Amigo lector:*

*La obra que tienes en tus manos posee un gran valor. En ella, su autor, ha vertido conocimientos, experiencia y mucho trabajo. El editor ha procurado una presentación digna de su contenido y está poniendo todo su empeño y recursos para que sea ampliamente difundida, a través de su red de comercialización.*

*Puedes obtener fotocopias de las páginas del libro para tu uso personal. Pero desconfía y rehúsa cualquier ejemplar "pirata" o fotocopia ilegal del mismo porque, de lo contrario, contribuirías al lucro de quienes, consciente o inconscientemente, se aprovechan ilegítimamente del esfuerzo del autor y del editor.*

*La reprografía indiscriminada y la piratería editorial, no solamente son prácticas ilegales, sino que atentan contra la creatividad y contra la difusión de la cultura.*

**PROMUEVE LA CREATIVIDAD  
RESPETA EL DERECHO DE AUTOR**

**Aníbal Ollero Baturone**

Catedrático de Ingeniería de Sistemas y Automática  
de la Escuela Superior de Ingenieros  
Universidad de Sevilla

# **ROBÓTICA**

## **Manipuladores y robots móviles**



**marcombo**  
BOIXAREU EDITORES

This One



71WX-ZUC-TY46

© Aníbal Ollero Baturone, 2001  
Reservados todos los derechos de publicación, reproducción, préstamo, alquiler o cualquier otra forma de cesión del uso de este ejemplar en cualquier idioma por MARCOMBO, S.A.  
Gran Vía de les Corts Catalanes, 594  
08007 Barcelona (España)

Quedan rigurosamente prohibidas, sin la autorización de los propietarios del "Copyright", bajo las sanciones establecidas en las leyes, la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la reprografía y el tratamiento informático, y la distribución de ejemplares de ella mediante alquiler o préstamo públicos, así como la exportación e importación de esos ejemplares para su distribución en venta, fuera del ámbito de la Unión Europea.

ISBN: 84-267-1313-0. Marcombo  
Alfaomega

Depósito Legal: B-37.752-2001

Impreso en España

Printed in Spain

Filmación: Gràfic-5,S.L. - Santiago Rusiñol, 23 - 08031 Barcelona

Impresión: GyERSA - Tambor del Bruc, 6 - 08970 Sant Joan Despí

*A Mari Carmen,  
Aníbal, María y Alejandro*



---

# Índice

---

<a href="#"><u>Foreword</u></a> .....	XIII
<a href="#"><u>Prólogo</u></a> .....	XV
<a href="#"><u>Prefacio del autor</u></a> .....	XVII
<a href="#"><u>CAPÍTULO 1. Introducción</u></a> .....	1
1.1 Robótica .....	1
1.2 Esquema general del sistema robot.....	3
1.3 Robots manipuladores.....	5
1.3.1 Sistema mecánico.....	5
1.3.2 Actuadores .....	5
1.3.3 Sensores y sistemas de control.....	6
1.4 Robots móviles .....	8
1.5 Robots autónomos y telerrobótica .....	11
1.6 Conclusiones.....	13
1.7 Referencias.....	13
<a href="#"><u>CAPÍTULO 2. Morfología de los robots</u></a> .....	16
2.1 Estructura de robots manipuladores.....	16
2.1.1 Tipos de articulaciones.....	16
2.1.2 Estructuras básicas .....	18
2.1.2.1 Configuración cartesiana.....	18
2.1.2.2 Configuración cilíndrica.....	20
2.1.2.3 Configuración polar o esférica .....	21

2.1.2.4 Configuración angular .....	21
2.1.2.5 Configuración Scara .....	22
2.1.3 Orientación del efecto final .....	22
2.1.4 Efectores finales .....	24
2.2 Nuevas estructuras para robots manipuladores .....	26
2.2.1 Robots redundantes .....	26
2.2.2 Robots flexibles .....	27
2.2.3 Manos .....	27
2.3 Robots móviles .....	28
2.3.1 Vehículos con ruedas .....	28
2.3.1.1 Ackerman .....	28
2.3.1.2 Triciclo clásico .....	29
2.3.1.3 Direccionamiento diferencial .....	30
2.3.1.4 Skid Steer .....	31
2.3.1.5 Pistas de deslizamiento .....	31
2.3.1.6 Síncronas .....	32
2.3.1.7 Otras configuraciones .....	33
2.3.2 Locomoción mediante patas .....	34
2.3.3 Configuraciones articuladas .....	36
2.3.4 Robots submarinos y aéreos .....	36
2.4 Conclusiones .....	37
2.5 Referencias .....	37
 CAPÍTULO 3. Representación de la posición y orientación .....	39
3.1 Posición y orientación en el plano .....	39
3.2 Posición y orientación en el espacio .....	44
3.3 Transformación inversa .....	52
3.4 Transformaciones compuestas .....	54
3.5 Otras representaciones de la orientación .....	57
3.6 Conclusiones .....	62
3.7 Referencias .....	62
 CAPÍTULO 4. Modelos cinemáticos de robots .....	63
4.1 Introducción .....	63
4.2 Relaciones entre sistemas de referencia .....	64
4.3 Modelo directo del manipulador .....	66
4.3.1 Modelo de un robot manipulador con $n$ articulaciones .....	66
4.4 Modelo inverso del manipulador .....	74
4.5 Velocidades lineales y angulares .....	85
4.5.1 Velocidad lineal .....	85
4.5.2 Velocidad angular .....	86
4.5.3 Propagación de velocidades .....	87
4.6 Jacobiano del manipulador .....	89
4.7 Modelos cinemáticos de robots móviles .....	97
4.7.1 Hipótesis básicas .....	97

<b>4.7.2 Restricciones cinemáticas .....</b>	<b>97</b>
<b>4.7.3 Modelo jacobiano .....</b>	<b>102</b>
<b>4.7.4 Modelos de diferentes configuraciones .....</b>	<b>104</b>
<b>4.7.5 Estimación de la posición y orientación .....</b>	<b>111</b>
<b>4.7.6 Vehículos robóticos con manipuladores .....</b>	<b>113</b>
<b>4.8 Conclusiones .....</b>	<b>114</b>
<b>4.9 Referencias .....</b>	<b>114</b>
<b>CAPÍTULO 5. Modelo dinámico .....</b>	<b>116</b>
<b>5.1 Aspectos dinámicos en robótica .....</b>	<b>116</b>
<b>5.2 Articulación simple de rotación .....</b>	<b>117</b>
<b>5.3 Generalización: modelo de un robot manipulador con <math>n</math> articulaciones ..</b>	<b>119</b>
<b>5.4 Formulación de Lagrange-Euler .....</b>	<b>119</b>
<b>5.5 Obtención del modelo dinámico mediante el método de Newton-Euler ..</b>	<b>120</b>
<b>5.5.1 Aceleraciones .....</b>	<b>120</b>
<b>5.5.2 Ecuaciones de Newton-Euler. Formulación iterativa .....</b>	<b>122</b>
<b>5.6 Análisis del modelo dinámico de los manipuladores .....</b>	<b>131</b>
<b>5.7 Obtención de las trayectorias articulares .....</b>	<b>133</b>
<b>5.8 Modelos dinámicos de vehículos robóticos .....</b>	<b>142</b>
<b>5.9 Conclusiones .....</b>	<b>144</b>
<b>5.10 Referencias .....</b>	<b>145</b>
<b>CAPÍTULO 6. Arquitecturas para control de robots .....</b>	<b>146</b>
<b>6.1 Introducción: funciones básicas y de control inteligente .....</b>	<b>146</b>
<b>6.2 Especificaciones .....</b>	<b>147</b>
<b>6.3 Requerimientos generales de la arquitectura .....</b>	<b>149</b>
<b>6.3.1 Programabilidad .....</b>	<b>149</b>
<b>6.3.2 Eficiencia .....</b>	<b>149</b>
<b>6.3.3 Capacidad de evolución .....</b>	<b>149</b>
<b>6.3.4 Grado de autonomía .....</b>	<b>150</b>
<b>6.3.5 Fiabilidad .....</b>	<b>150</b>
<b>6.3.6 Adaptabilidad .....</b>	<b>150</b>
<b>6.4 Tipos básicos de arquitecturas según reactividad .....</b>	<b>151</b>
<b>6.5 Aproximación al diseño de la arquitectura .....</b>	<b>153</b>
<b>6.5.1 Diseño funcional de la arquitectura .....</b>	<b>153</b>
<b>6.5.2 Gestión de ejecución e implantación .....</b>	<b>156</b>
<b>6.6 Soluciones hardware y software .....</b>	<b>158</b>
<b>6.7 Conclusiones .....</b>	<b>164</b>
<b>6.8 Referencias .....</b>	<b>164</b>
<b>CAPÍTULO 7. Sensores .....</b>	<b>166</b>
<b>7.1 Introducción: sensores y magnitudes .....</b>	<b>166</b>
<b>7.2 Clasificaciones y características de sensores .....</b>	<b>167</b>
<b>7.3 Medidas de desplazamientos lineales y giros .....</b>	<b>168</b>

7.3.1	Potenciómetros para medida de desplazamientos .....	168
7.3.2	Codificadores ópticos .....	170
7.3.2.1	Máquinas síncronas, transformadores y resolvedores...	175
7.3.3	Medida de velocidad de ejes .....	178
7.4	Sensores de presencia y proximidad.....	178
7.5	Sensores de tacto .....	180
7.6	Medidas de fuerza y par .....	182
7.7	Sensores de navegación .....	185
7.7.1	Sensores Doppler.....	185
7.7.2	Compás magnético .....	186
7.7.3	Giróscopos.....	188
7.7.3.1	Giróscopos mecánicos .....	188
7.7.3.2	Giróscopos electrónicos.....	190
7.7.3.3	Giróscopos ópticos.....	190
7.7.4	Sistemas de navegación inercial.....	191
7.7.5	Estimación de posición de vehículos basada en estaciones de transmisión. Aplicación del GPS .....	191
7.7.6	Sensores para vehículos autoguiados industriales.....	194
7.8	Conclusiones.....	196
7.9	Referencias .....	197
<b>CAPÍTULO 8. Control de las articulaciones de un robot manipulador .....</b>		198
8.1	Estrategias de control de articulaciones .....	198
8.2	Control desacoplado de articulaciones .....	199
8.3	Control basado en el modelo dinámico .....	202
8.3.1	Ejemplos introductorios .....	202
8.3.2	Control basado en el modelo. Par computado.....	204
8.4	Control adaptativo de robots .....	224
8.4.1	Introducción .....	224
8.4.2	Par computado adaptativo .....	226
8.4.3	Controlador de inercia adaptativa .....	231
8.5	Control con aprendizaje.....	237
8.6	Control en espacio cartesiano .....	243
8.7	Control de esfuerzos .....	247
8.7.1	Rigidez .....	247
8.7.2	Control de rigidez .....	248
8.7.3	Control híbrido esfuerzos/posición .....	251
8.8	Conclusiones.....	255
8.9	Referencias .....	256
<b>CAPÍTULO 9. Control de robots móviles .....</b>		258
9.1	Control de movimientos de vehículos autónomos.....	258
9.2	Seguimiento de caminos explícitos .....	260
9.3	Seguimiento de caminos empleando métodos geométricos .....	261
9.3.1	Seguimiento mediante persecución pura .....	261

<b>9.3.2 Control geométrico mediante generación de polinomios de orden cinco .....</b>	<b>267</b>
<b>9.4 Aplicación de la teoría de control.....</b>	<b>271</b>
<b>9.4.1 Controlabilidad y estabilización .....</b>	<b>271</b>
<b>9.4.2 Seguimiento de trayectorias.....</b>	<b>275</b>
<b>9.4.2.1 Trayectorias de referencia .....</b>	<b>275</b>
<b>9.4.2.2 Ley de control lineal .....</b>	<b>277</b>
<b>9.4.2.3 Ley de control no lineal .....</b>	<b>282</b>
<b>9.4.3 Seguimiento de caminos .....</b>	<b>284</b>
<b>9.4.3.1 Planteamiento del problema .....</b>	<b>284</b>
<b>9.4.3.2 Ley de control lineal .....</b>	<b>287</b>
<b>9.4.3.3 Ley de control no lineal .....</b>	<b>290</b>
<b>9.4.4 Seguimiento de caminos utilizando el modelo de la bicicleta..</b>	<b>290</b>
<b>9.5 Seguimiento predictivo generalizado .....</b>	<b>293</b>
<b>9.6 Control reactivo.....</b>	<b>300</b>
<b>9.7 Conclusiones.....</b>	<b>301</b>
<b>9.8 Referencias .....</b>	<b>301</b>
 <b>CAPÍTULO 10. Generación de trayectorias.....</b>	<b>303</b>
<b>10.1 Planteamiento del problema .....</b>	<b>303</b>
<b>10.2 Definición paramétrica de las curvas .....</b>	<b>305</b>
<b>10.3 Técnicas de interpolación.....</b>	<b>309</b>
<b>10.3.1 Técnicas básicas de interpolación.....</b>	<b>309</b>
<b>10.3.2 Empleo de funciones spline .....</b>	<b>311</b>
<b>10.4 Generación de caminos en el espacio cartesiano.....</b>	<b>318</b>
<b>10.5 Generación de trayectorias para manipuladores .....</b>	<b>320</b>
<b>10.6 Trayectorias articulares para manipuladores robóticos .....</b>	<b>322</b>
<b>10.6.1 Empleo de polinomios cúbicos.....</b>	<b>323</b>
<b>10.6.2 Empleo de polinomios cúbicos con puntos de paso .....</b>	<b>324</b>
<b>10.6.3 Empleo de polinomios de orden superior .....</b>	<b>327</b>
<b>10.6.4 Empleo de funciones lineales con enlace parabólico .....</b>	<b>330</b>
<b>10.7 Generación en tiempo real .....</b>	<b>338</b>
<b>10.8 Conclusiones.....</b>	<b>339</b>
<b>10.9 Referencias .....</b>	<b>339</b>
 <b>CAPÍTULO 11. Programación de robots.....</b>	<b>341</b>
<b>11.1 Sistemas de programación de robots .....</b>	<b>341</b>
<b>11.2 Programación por guiado .....</b>	<b>342</b>
<b>11.3 Programación textual .....</b>	<b>345</b>
<b>11.4 Sistemas de referencia .....</b>	<b>347</b>
<b>11.5 Especificación de movimientos en robots manipuladores .....</b>	<b>350</b>
<b>11.6 Estructuras de datos .....</b>	<b>352</b>
<b>11.7 Especificación de localizaciones .....</b>	<b>354</b>
<b>11.8 Interacción con el entorno y características tiempo real .....</b>	<b>357</b>
<b>11.9 Programación de vehículos robóticos .....</b>	<b>365</b>

<u>11.10 Conclusiones.....</u>	368
<u>11.11 Referencias .....</u>	368
<b>CAPÍTULO 12. Detección de colisiones y planificación de caminos .....</b>	<b>370</b>
<u>12.1 Detección, evitación y planificación .....</u>	370
<u>12.2 Funciones de detección, evitación y planificación.....</u>	371
<u>12.3 Detección y evitación de colisiones mediante reacción directa a información de sensores.....</u>	373
<u>12.4 Modelos.....</u>	375
<u>12.4.1 Modelado mediante ocupación de celdas .....</u>	375
<u>12.4.2 Empleo de estructuras jerárquicas .....</u>	375
<u>12.4.3 Modelos del entorno basado en primitivas 3-D de sólidos.....</u>	376
<u>12.4.4 Expansión de obstáculos.....</u>	378
<u>12.4.5 Modelado en el espacio de configuraciones .....</u>	379
<u>12.5 Detección y evitación de colisiones empleando modelos.....</u>	382
<u>12.6 Planificación basada en modelo geométrico .....</u>	385
<u>12.6.1 Planteamiento del problema .....</u>	385
<u>12.6.2 Métodos en espacio cartesiano .....</u>	386
<u>12.6.3 Métodos en espacio de configuraciones .....</u>	389
<u>12.7 Métodos reactivos .....</u>	392
<u>12.7.1 Campos potenciales.....</u>	393
<u>12.8 Conclusiones.....</u>	397
<u>12.9 Referencias.....</u>	397
<b>CAPÍTULO 13. Telerrobótica.....</b>	<b>400</b>
<u>13.1 Introducción y conceptos básicos .....</u>	400
<u>13.2 Teleactuación .....</u>	401
<u>13.3 Diseño de sistemas de control de teleoperación.....</u>	403
<u>13.4 Sistemas bilaterales maestro-esclavo.....</u>	403
<u>13.5 Empleo de gráficos predictivos.....</u>	409
<u>13.6 Teleprogramación e interacción con el entorno .....</u>	411
<u>13.7 Control de supervisión .....</u>	412
<u>13.8 Telesensorización.....</u>	413
<u>13.9 Sistemas de visión en teleoperación.....</u>	414
<u>13.9.1 Cámaras .....</u>	414
<u>13.9.2 Visualizadores .....</u>	415
<u>13.9.3 Visualización de imágenes virtuales .....</u>	418
<u>13.9.4 Transmisión de imágenes para teleoperación .....</u>	419
<u>13.10 Conclusiones .....</u>	419
<u>13.11 Referencias.....</u>	420
<b>Apéndice .....</b>	<b>422</b>

---

# Foreword

---

Robots have captured the world's attention by responding to nuclear accidents, finding shipwrecks, exploring volcanoes and traveling into space. Robots are changing the way we build, secure, supply, power and feed the world. They are becoming tools for craft, labor and hazardous duty on earth and beyond. Robots are transforming the way people live and work, and are expanding the bounds of human experience.

Robotics is experiencing explosive growth, a rate of progress fueled by rapid-fire advances in computing, sensing, electronics and software. Robots are now on the verge of revolutionizing industries such as agriculture, mining and transportation. They are capturing not only our imaginations, but major markets as well.

To fulfill the vision, the charter is to evolve the technology, robots, and to create the leaders of tomorrow. Anibal's book is a brilliant means for instilling the vision and passing the technology to the emerging generation of roboticists.

Because Anibal's book is written in first-hand Spanish, it will serve corners of the world without the inevitable losses of translation. Much robotics publication has originated in English, and is at best translated to other languages. The default for this book could have been to inject established English terms into the Spanish text. However, Anibal takes the extra effort to generate precise Spanish expressions for technical robotic concepts that are much more than quoting the English counterpart or presenting simplistic translation.

Until now insufficient attention has been paid to instructing the next generation of roboticists. This splendid book could not have appeared at a better time. Robotics educators, developers and industries are emerging in Spanish-speaking countries like Spain, Mexico, Chile and many others, and the Spanish-speaking community will be special beneficiaries of this book.

Anibal has been an early pioneer and a practitioner. Now he is an author and has the prospect for great impact in the world of robotics education. The breadth and depth of coverage of this book makes it appropriate for several course levels. It is an encyclopedic compendium that broadly covers robotics topics from dynamics and controls

to sensors, perception and planning. It can be used as an educational text, tutorial guide, or reference for individual use. The text also offers practical advice and instruction on programming the algorithms and theory presented, as well as suggesting criteria for robot design, both hardware and software.

Current and future roboticists will benefit from the solid coverage and calibrated perspectives in this outstanding book. "*Robótica. Manipuladores y robots móviles*" is an indispensable addition to every robotics bookshelf. It will catalyze the Spanish-speaking robotics culture. It will have immense impact on the robots and future roboticists that are revolutionizing the world.

WILLIAM "RED" WHITTAKER  
Robotics Institute, Carnegie Mellon University

---

# Prólogo

---

Los robots han logrado captar la atención del mundo cuando se han aplicado a accidentes nucleares, localización de naufragios, exploración de volcanes y viajes espaciales. Están cambiando la forma en la que construimos, mantenemos la seguridad, producimos y distribuimos energía y alimentos al mundo. Se han convertido en herramientas para producir, trabajar y realizar labores peligrosas en la Tierra y fuera de ella. Los robots están transformando la forma de vida y trabajo, y están expandiendo los límites de la experiencia humana.

La robótica está experimentando un crecimiento explosivo impulsado por los avances en computación, sensores, electrónica y software. Los robots están ya en la antesala de revolucionar los procedimientos que se emplean en la agricultura, la minería y el transporte, atrayendo no solo nuestra imaginación sino también nuestros mercados.

Para cumplir estos objetivos, se trata ahora de hacer evolucionar la tecnología y formar a las personas que deben liderarla en el futuro. El libro de Aníbal constituye una importante aportación para inculcar estos objetivos y transmitir la tecnología a la generación emergente de especialistas en robótica.

Debido a que el libro está escrito en español, evita las inevitables pérdidas que se producen en las traducciones. La mayor parte de las publicaciones de robótica se han originado en inglés y después han sido traducidas a otros idiomas. El libro podría haber sido escrito insertando términos en inglés en el texto. Sin embargo, su autor ha dedicado un esfuerzo adicional a generar expresiones precisas en español para conceptos en inglés, lo que tiene mucho más valor que simplemente citar referencias en inglés o realizar traducciones simplistas.

Hasta ahora se ha prestado una atención insuficiente a la enseñanza de las nuevas generaciones de especialistas en robótica. Este espléndido libro no podría haber aparecido en mejor momento. Los profesores de la robótica, y profesionales involucrados en su desarrollo y aplicación en las industrias están emergiendo en países de habla his-

pana tales como España, Méjico, Chile, Argentina y muchos otros. La comunidad de habla hispana se beneficiará especialmente de este libro.

Aníbal ha sido un pionero en los desarrollos y aplicaciones de la robótica. Ahora es un autor que tiene posibilidades de causar un gran impacto en el mundo de la educación en robótica. La amplitud y profundidad del libro lo hace apropiado para cursos de diferentes niveles. Es un compendio enciclopédico que cubre extensamente diferentes tópicos de robótica, desde la dinámica y el control, a los sensores, percepción y planificación. Puede ser utilizado como un libro de texto, una guía tutorial, o como referencia para uso personal. El texto ofrece también soluciones prácticas, así como la enseñanza en la programación de los algoritmos y las soluciones teóricas presentadas, sugiriendo también criterios para el diseño del robot, incluyendo tanto el "hardware" como el "software".

Los actuales y futuros especialistas en robótica se beneficiarán de la solidez y la calibrada perspectiva de este destacado libro. *"Robótica. Manipuladores y robots móviles"* es un elemento indispensable en cualquier biblioteca de robótica. Este libro catalizará la robótica en los países de habla hispana y tendrá un gran impacto en los robots y los especialistas de la robótica del futuro que están revolucionando el mundo.

WILLIAM "RED" WHITTAKER  
Robotics Institute, Carnegie Mellon University

---

# Prefacio del autor

---

Los robots son máquinas en las que se integran componentes mecánicos, eléctricos, electrónicos y de comunicaciones, y dotadas de un sistema informático para su control en tiempo real, percepción del entorno y programación.

La robótica es todavía una disciplina relativamente joven. En efecto, aunque el término robot se acuña en los años veinte del pasado siglo, la robótica industrial nace en los cincuenta y sólo en los setenta comienzan a impartirse cursos de robótica en un gran número de universidades.

En la robótica industrial se trata fundamentalmente de dotar de flexibilidad a los procesos productivos manteniendo al mismo tiempo la productividad que se consigue con una máquina automática especializada.

En los años ochenta y noventa se han diseñado un gran número de máquinas cuyo objetivo no es sustituir la actividad directa de un trabajador en una cadena de producción. Se trata de realizar tareas en lugares difícilmente accesibles, con riesgo de accidentes, en condiciones peligrosas para la salud, o trabajos que resultan difíciles por el tamaño de los objetos que es necesario manipular. Así, se han abordado aplicaciones en la exploración espacial, actividades subacuáticas, manipulación y transporte de materiales peligrosos, minería, agricultura, construcción, cirugía, etc. La mayoría de estas máquinas son fundamentalmente teleoperadas, pero se trata de dotarlas de una mayor autonomía llegando a construir robots. El número de estos robots aumentará de forma importante en los próximos años.

Otro sector de importancia creciente en las aplicaciones de la robótica es el de los robots de servicios, entre los cuales se incluyen los robots domésticos, robots de ayuda a los discapacitados, y robots asistentes en general. Tampoco hay que olvidar que los humanos siempre han sentido una gran atracción por las máquinas que imitan los gestos más visibles de los seres vivos en general y de las personas en particular. Por tanto, tampoco hay que extrañarse de la importancia creciente de las aplicaciones recreativas de la robótica.

La robótica industrial, desde sus orígenes, estuvo muy orientada a las funciones de manipulación. De hecho, suele considerarse que un robot industrial es esencialmente un robot manipulador. Por ello, la mayor parte de los libros de robótica se dedican exclusivamente a robots manipuladores.

Durante los años ochenta y noventa, los aspectos relacionados con la robótica denominada móvil han ido cobrando una importancia creciente. Los robots móviles se diferencian de los robots manipuladores convencionales en que no están anclados, si no que por el contrario pueden desplazarse por el terreno, por el agua o incluso volar libremente. Los aspectos más específicos de la robótica móvil son los relacionados con tales desplazamientos autónomos o navegación del robot. Desde el punto de vista de la robótica industrial, los vehículos robóticos pueden considerarse como una generalización de los vehículos de transporte en fábrica con el objetivo de proporcionar una mayor flexibilidad. Sin embargo, donde el papel de la robótica móvil es más importante es en las nuevas aplicaciones de la robótica, fuera del trabajo en las cadenas de fabricación industrial, tales como las que se mencionaban en párrafos anteriores.

En este libro se ha adoptado como un objetivo básico la consideración tanto de robots manipuladores como de robots móviles. Otro aspecto que se ha considerado fundamental es suministrar herramientas para la mejor comprensión de los conceptos y métodos que se presentan. Por ello, se incluyen numerosos ejemplos que el lector puede de reproducir empleando una herramienta MATLAB-Simulink construida al efecto y que se suministra con el texto.

Este libro es el resultado de maduraciones sucesivas de apuntes empleados en cursos de las Universidades de Sevilla y Málaga. Pasar de unos apuntes de un determinado curso, o cursos, a un libro, requiere frecuentemente un esfuerzo mucho mayor que el previsto. Durante la realización de este esfuerzo la calidad de los libros que se manejan suele parecer cada vez mayor, hasta el punto en que el escritor suele preguntarse una y otra vez si merece la pena continuar. En este caso, se ha creído que los objetivos mencionados en el párrafo anterior justificaban la edición del libro.

Por otra parte, es necesario tener en cuenta que la literatura en castellano sobre robótica es muy escasa. Este comentario puede parecer irrelevante para un buen número de profesores, e incluso alumnos, pero no lo es para el autor de esta obra que considera una cierta obligación transmitir conocimientos en nuestra lengua.

El texto puede considerarse dividido en cuatro partes. La primera de ellas consiste en dos capítulos. El primero introduce nociones básicas sobre robótica y el segundo describe la morfología de los robots, incluyendo tanto robots manipuladores como móviles.

La segunda parte del libro se dedica al modelado de los robots, estando formada por los capítulos 3, 4 y 5. El capítulo 3 expone nociones básicas sobre sistemas de representación y las transformaciones entre ellos. Los conceptos de este capítulo pueden ser ya conocidos por algunos lectores. En este caso su única utilidad es la de familiarizarse con la notación que se emplea en capítulos posteriores. Los capítulos 4 y 5 son los específicos de modelado. El contenido de estos capítulos es quizás el más difícil para estudiantes o especialistas en las tecnologías de la información. Sin embargo, resultará fácil para lectores con conocimientos de mecánica. El capítulo 4 es extenso y considera modelos cinemáticos tanto de robots manipuladores como de robots móviles. El capítulo 5 está dedicado al modelo dinámico. Los lectores exclusivamente interesados en el control y los aspectos de la robótica relacionados con las tecnologías de

la información pueden obviar las secciones 4 y 5 de este capítulo en las cuales se presentan los métodos para la obtención del modelo dinámico. Conviene también poner de manifiesto que la existencia de la herramienta MATLAB-Simulink permite el empleo de los modelos de los robots sin conocer con detalle los métodos para su obtención.

La tercera parte del libro se dedica específicamente al control. El capítulo 6 se ocupa de las arquitecturas de control considerándose tanto las arquitecturas basadas en la planificación con modelo del entorno, como las reactivas. En el capítulo 7 se estudian los sensores, incluyendo sensores básicos para el control de movimientos de los ejes, sensores de manipuladores robóticos y sensores específicos para la navegación de vehículos autónomos. En este capítulo no se tratan los sensores de percepción del entorno, tales como sistemas de visión, o sensores de distancias, para cuyo estudio el lector debe acudir a otros textos especializados. El capítulo 8 es relativamente extenso dedicándose al control de los robots manipuladores, incluyendo la estrategia básica del par computado y sus diferentes particularizaciones, los métodos de control adaptativo, el control con aprendizaje, el control en el espacio cartesiano y el control de esfuerzos. El capítulo 9 está dedicado al control de los robots móviles, incluyéndose diversos métodos desde la técnica básica de la persecución pura hasta los métodos basados en la teoría del control. Finalmente, en el capítulo 10, se consideran métodos para la generación de trayectorias. Partiendo de una especificación del movimiento que debe realizar el robot, se trata de definir de forma precisa una trayectoria en el espacio y en el tiempo. Se estudia la generación de trayectorias en el espacio cartesiano y en el de las articulaciones, y se consideran tanto robots manipuladores como móviles.

La cuarta y última parte del libro está dedicada a la especificación de la tarea que debe realizar el robot. El capítulo 11 se dedica a la programación, con énfasis en la programación de robots manipuladores. En primer lugar se trata la programación por guiado y, a continuación, se consideran los lenguajes de programación textual, empleándose como ejemplo la programación en VAL II. En la última sección del capítulo se considera la programación de vehículos robóticos. El capítulo 12 se ocupa de la detección de colisiones y planificación de caminos. Se trata tanto la detección y evitación de colisiones mediante reacción directa a la información suministrada por los sensores, como los métodos de detección y planificación de caminos libres de colisión basados en el empleo de modelos del robot y su entorno. Finalmente, en el capítulo 13, se presentan conceptos, métodos y tecnologías sobre teleoperación y telerrobótica. Conviene poner de manifiesto que tanto el capítulo 12 como el 13 deben considerarse como introducciones ya que su estudio en profundidad requiere textos especializados.

Antes de terminar, deseo manifestar mi agradecimiento a todas las personas que han contribuido al estado actual del libro.

En primer lugar quiero expresar mi agradecimiento a Iván Maza Alcañiz, actualmente becario de investigación en nuestro grupo, ya que, además de sus numerosas correcciones, ha desarrollado la herramienta MATLAB-Simulink que se utiliza en el texto, programado los ejemplos y obtenido resultados que se han incluido en el mismo.

Agradezco también el trabajo de edición de Lorena de Martino Andrés que ha mecanografiado una buena parte del libro, incluyendo expresiones matemáticas, dibujado figuras y realizado numerosas correcciones durante varios años.

Asimismo, agradezco las revisiones del texto de Guillermo Heredia Benot y Federico Cuesta Rojo, también profesores de las asignaturas de robótica que se han impartido con los apuntes que han dado lugar al libro, así como las revisiones de numerosos alumnos que en los últimos años han utilizado dichos apuntes.

En el libro se utilizan como ejemplos, diversos robots del Instituto Andaluz de Automática y Robótica, así como robots desarrollados en las universidades de Sevilla y Málaga. Mi agradecimiento a los profesores Javier Aracil y Alfonso García Cerezo, responsables de este Instituto en Sevilla y Málaga, así como a los compañeros de estas universidades que han participado en los últimos años en trabajos conjuntos de investigación directamente relacionados con los temas que se tratan en el libro, entre los cuales, además de los profesores ya citados, cabe mencionar a Begoña Arrue, Joaquín Ferruz, Omar Sánchez, Fernando Gómez Bravo, Ángel Rodríguez Castaño, Francisco López Pichaco y Carlos Nogales de la Universidad de Sevilla, así como a Víctor Muñoz, Jorge Martínez, Jesús Gómez de Gabriel y Anthony Mandow en cuyas Tesis presentadas en la Universidad de Málaga se obtuvieron resultados que se han empleado también en la redacción del libro.

Por otra parte, quiero poner de manifiesto que dicha redacción, en su forma actual, tampoco hubiera sido posible sin mi estancia en el "Robotics Institute" de la Carnegie Mellon University (Pittsburgh, EE.UU. de América), en los años 1990 y 1991, durante la cual participé en diversos proyectos adquiriendo conceptos teóricos y prácticos sobre robótica que me han sido muy valiosos. Quisiera mencionar en particular mi agradecimiento a William "Red" Whitaker, director del "Field Robotics Center" en el que estuve trabajando.

Por último, agradezco la financiación para el desarrollo de los Proyectos y actividades de I+D relacionadas con el libro recibida de la Comisión Europea, CICYT, Plan Andaluz de Investigación y Sociedad Sevilla Siglo XXI de la Diputación de Sevilla.

Sevilla, Junio de 2001.

En este capítulo se comienza introduciendo la noción de robótica y tratando sus antecedentes. En el siguiente apartado se presenta un esquema general del sistema robot considerando también la interacción con su entorno. A continuación se tratan aspectos específicos de los robots manipuladores y de los móviles. Por último se introducen conceptos relacionados con la autonomía de un robot distinguiendo entre robots teleoperados, robots de funcionamiento repetitivo y robots autónomos o inteligentes.

## 1.1 ROBÓTICA

---

En el término robot confluyen las imágenes de máquinas para la realización de trabajos productivos y de imitación de movimientos y comportamientos de seres vivos.

Los robots actuales son obras de ingeniería y como tales concebidas para producir bienes y servicios o explotar recursos naturales. Desde esta perspectiva son máquinas con las que se continua una actividad que parte de los propios orígenes de la humanidad, y que desde el comienzo de la Edad Moderna se fundamenta esencialmente en conocimientos científicos.

En nuestro siglo el desarrollo de máquinas ha estado fuertemente influido por el progreso tecnológico. De esta forma se pasa de máquinas que tienen como objetivo exclusivo la amplificación de la potencia muscular del hombre, sustituyéndolo en su trabajo físico, a máquinas o instrumentos que son también capaces de procesar información, complementando, o incluso sustituyendo, al hombre en algunas actividades intelectuales.

Por otra parte, también desde la antigüedad, el hombre ha sentido fascinación por las máquinas que imitan la figura y los movimientos de seres animados. Existe una larga tradición de autómatas desde el mundo griego hasta nuestro siglo, pasando por los autómatas de los artesanos franceses y suizos del siglo XVIII, que ya incorporaban interesantes dispositivos mecánicos para el control automático de movimientos.

El término robot aparece por primera vez en 1921, en la obra teatral R.U.R. (Rossum's Universal Robots) del novelista y autor dramático checo Karel Čapek en cuyo idioma la palabra "robota" significa fuerza del trabajo o servidumbre. Por aquellos años la producción en grandes series se había introducido en numerosas fábricas. Se discute ya del poder de las máquinas y la dominación de los hombres por las máquinas, argumento de ésta y otras obras teatrales y películas de los años veinte en los que aparecen trabajadores robóticos.

El término tiene amplia aceptación y pronto se aplica a autómatas construidos en los años veinte y treinta que se exhiben en ferias, promociones de productos, películas y otras aplicaciones más o menos festivas. Se trata de imitar movimientos de seres vivos pero también de demostrar técnicas de control remoto, incluyéndose en algunos casos funciones sensoriales primarias.

En cualquier caso, interesa recordar que el término robot nace asociado a la idea de trabajo y producción. En 1915, Leonardo Torres Quevedo declaró a la revista "Scientific American" (Eames, 1973):

"Los antiguos autómatas imitaban la apariencia y movimientos de los seres vivos, lo cual no tiene mucho interés práctico; lo que yo busco es una clase de aparatos que, sin necesidad de reproducir los gestos más visibles del hombre, intentan obtener los mismos resultados que una persona".

Los robots industriales surgen de la convergencia de tecnologías del control automático y, en particular, del control de máquinas herramientas, de los manipuladores teleoperados, y de la aplicación de computadores en tiempo real. En los párrafos siguientes se comentan brevemente algunos aspectos significativos en la evolución de estas tecnologías.

Mediante el control automático de procesos se pretende concebir y realizar ingenios que permitan gobernar un proceso sin la intervención de agentes exteriores, especialmente el hombre. En particular, se presentan problemas de seguimiento automático de señales de consigna, mediante los denominados servosistemas. Estos sistemas generan automáticamente señales de control que tratan de anular la diferencia entre la señal de consigna y la señal medida en el proceso u objeto que se pretende controlar.

Un problema similar es el de la regulación automática mediante la cual se trata de mantener una consigna, aunque se presenten perturbaciones que tiendan a separar al sistema de las condiciones deseadas.

Tanto los servosistemas como los reguladores se basan en el principio de la realimentación. Las señales de consigna o referencia se comparan con medidas de variables del proceso u objeto que se pretende controlar y su diferencia se emplea para generar acciones de control sobre el propio proceso u objeto. En los sistemas de control automático esta cadena cerrada de acción-medida-acción se realiza sin intervención del hombre.

La automatización industrial con utilización de sistemas de control automático comienza también en el siglo XIX pero no es hasta el siglo XX y, muy especialmente, después de la Segunda Guerra mundial, cuando empieza a extenderse de forma importante en todos los sectores industriales. De esta forma, se generalizan los sistemas de control automático de variables de procesos industriales y, en particular, sistemas de control de posición y velocidad. Se emplean también sistemas de control realimentado en barcos o aviones que deben seguir automáticamente una determinada trayectoria (pilotos automáticos) o en el posicionamiento de radares.

Tradicionalmente, en la realización de sistemas de control automático se han empleado diversas tecnologías tales como la neumática, hidráulica y, posteriormente, la eléctrica. A finales de los años sesenta y comienzos de los setenta los minicomputadores encuentran una importante acogida en aplicaciones de control. La aparición en 1972 del microprocesador suministra un impulso decisivo al control por computador, haciendo rentables numerosísimas aplicaciones entre las que se cuenta el control de robots. Los avances en microelectrónica de los años ochenta, con la tecnología de los circuitos de gran escala de integración, acentúan esta tendencia.

Con respecto a las máquinas herramientas de control numérico, hay que señalar los proyectos que se desarrollaron en EE.UU. a finales de los años cuarenta y principios de los cincuenta. Se combinaban los progresos en el diseño de servosistemas con las recientes experiencias en técnicas de computación digital. De esta forma, el contorno de corte era codificado en cinta de papel perforado, utilizándose para generar automáticamente las órdenes a los servomecanismos de la máquina. En 1953 se presentaba en el "Massachusetts Institute of Technology" (MIT) una máquina de estas características.

Los teleoperadores se desarrollaron en los años cuarenta para manejar materiales radioactivos. Consistían en un par de pinzas "maestra" y "esclava" acopladas por mecanismos que permitían que la pinza "esclava", en contacto con el material peligroso, reprodujera los movimientos de la pinza "maestra" accionada por un operador detrás de un muro protector con ventanas apropiadas para observar la operación (Gertz, 1963).

El primer teleoperador accionado por servomecanismos eléctricos se presentó en 1947. Poco después, en 1948, se introdujeron servosistemas con realimentación de fuerza hacia la pinza "maestra" para permitir que el operador percibiera el esfuerzo desarrollado.

En 1954 el ingeniero americano George Devol patentó el que se considera el primer robot industrial: un dispositivo que combinaba la articulación de un teleoperador con el eje servocontrolado de una máquina de control numérico.

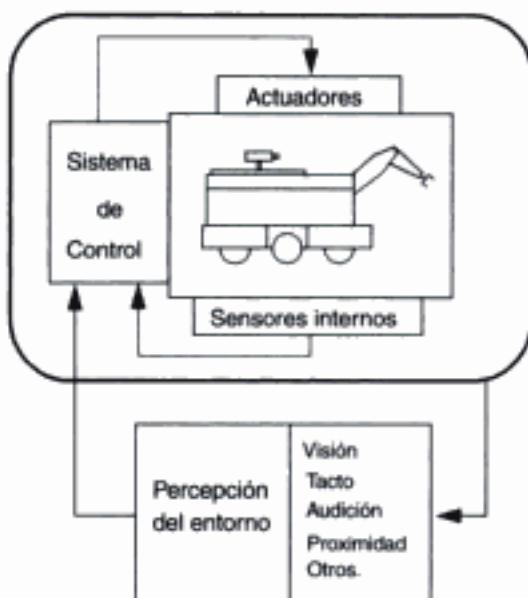
En el próximo apartado se presenta una primera aproximación a los robots actuales y su interacción con el entorno. En los dos siguientes se tratan aspectos específicos de los robots manipuladores y los vehículos robóticos. Posteriormente, se presenta una clasificación de sistemas robóticos teniendo en cuenta su autonomía.

## 1.2 ESQUEMA GENERAL DEL SISTEMA ROBOT

---

En la figura 1.1 se muestra el esquema básico de un robot (Silva, 1984). En ella se identifican un sistema mecánico, actuadores, sensores y el sistema de control como elemento básico necesario para cerrar la cadena actuación-medidas-actuación. En los párrafos siguientes se consideran estos componentes dedicando una atención especial al sistema de control. En los apartados que siguen se tratan problemas más específicos de robots manipuladores o robots móviles. En Ollero (1991) puede encontrarse una descripción más detallada.

En el sistema mecánico puede distinguirse entre el órgano terminal, el brazo articulado, y un vehículo. En la mayor parte de los robots industriales no existe tal vehículo, estando fija la base del brazo.



**Figura 1.1:** Robot y su interacción con el entorno.

Desde el punto de vista del procesamiento de la información, en robótica se involucran funciones de control de movimientos, percepción y planificación.

En un sentido amplio, el sistema de control involucra tanto bucles de realimentación de la información suministrada por los sensores internos, como del entorno.

Los sensores internos miden el estado de la estructura mecánica y, en particular, giros o desplazamientos relativos entre articulaciones, velocidades, fuerzas y pares. Estos sensores permiten cerrar bucles de control de las articulaciones de la estructura mecánica.

Los sensores externos permiten dotar de sentidos al robot. La información que suministran es utilizada por el sistema de percepción para aprehender la realidad del entorno. Los sistemas de percepción sensorial hacen posible que un robot pueda adaptar automáticamente su comportamiento en función de las variaciones que se producen en su entorno, haciendo frente a situaciones imprevistas. Para ello el sistema de control del robot incorpora bucles de realimentación sensorial del entorno, generando automáticamente acciones en función de la comparación de dicha información sensorial con patrones de referencia.

El desarrollo de sistemas de percepción en Robótica surge a partir de los progresos tecnológicos en sensores tales como los de visión, tacto e, incluso, audición. Sin embargo, la percepción involucra no sólo la captación de la información sensorial, sino también su tratamiento e interpretación. Por tanto, es necesario realizar una abstracción a partir de un cierto conocimiento previo del entorno. Es claro que la complejidad de la percepción artificial depende de lo estructurado que esté dicho entorno.

Por último, la planificación tiene como objetivo encontrar una trayectoria desde una posición inicial a una posición objetivo, sin colisiones, y minimizando un determinado índice. En el caso más simple, el problema se plantea en un entorno que se supone conocido (existe un mapa previo) y estático. Se supone además que el robot es omnidireccional, que se mueve suficientemente lento y que es capaz de seguir el camino de forma perfecta.

## 1.3 ROBOTS MANIPULADORES

La mayor parte de los robots industriales actuales son esencialmente brazos articulados. De hecho, según la definición del "Robot Institute of America", un robot industrial es un manipulador programable multifuncional diseñado para mover materiales, piezas, herramientas o dispositivos especiales, mediante movimientos variados, programados para la ejecución de distintas tareas.

En la robótica subyace la idea de sustituir equipos capaces de automatizar operaciones concretas por máquinas de uso más general que puedan realizar distintas tareas. El concepto "programable" es también básico. Como se mencionaba en la sección anterior, la realización por programa de las funciones de control ofrece mucha mayor flexibilidad, y la posibilidad de implantar funciones complejas necesarias para controlar el manipulador.

### 1.3.1 Sistema mecánico

El sistema mecánico está compuesto por diversas articulaciones. Normalmente se distingue entre el brazo y el órgano terminal o efecto final que puede ser intercambiable, empleando pinzas o dispositivos específicos para distintas tareas.

El aumento del número de articulaciones aporta mayor maniobrabilidad pero dificulta el problema del control, obteniéndose normalmente menores precisiones por acumulación de errores. Muchos robots industriales actuales tienen menos de los seis grados de libertad de rotación o traslación que se requieren en general para posicionar y orientar en el espacio el órgano terminal. Sin embargo, también se desarrollan manipuladores altamente redundantes con múltiples articulaciones para aplicaciones en áreas de trabajo de difícil acceso. Entre estos cabe destacar los robots tipo serpiente. Asimismo, se investiga en robots flexibles que permitan un largo alcance con un peso reducido.

En este punto conviene indicar que las ecuaciones que describen el movimiento del brazo articulado son ecuaciones diferenciales no lineales y acopladas, para las que, en un caso general, resulta difícil obtener soluciones analíticas. Físicamente, los términos de acoplamiento representan: pares gravitacionales que dependen de la posición de las articulaciones, pares de reacción debidos a las aceleraciones de otras articulaciones, y pares debidos a la aceleración de Coriolis y fuerzas centrífugas. La magnitud de estas interacciones depende de las características del brazo y de la carga.

### 1.3.2 Actuadores

Los actuadores generan las fuerzas o pares necesarios para animar la estructura mecánica. Se utilizan tecnologías hidráulicas, para desarrollar potencias importantes, y neumáticas, pero en la actualidad se ha extendido el empleo de motores eléctricos, y en particular motores de corriente continua servocontrolados, empleándose en algunos casos motores paso a paso y otros actuadores electromecánicos sin escobillas. Existen también robots industriales de accionamiento directo que permiten eliminar los problemas mecánicos inherentes al empleo de engranajes y otras transmisiones. Se inves-

tiga en nuevos actuadores que disminuyan la inercia, suministren un par elevado, aumenten la precisión, originen menos ruido magnético y sean de bajos peso y consumo (Rosheim, 1989).

Por otra parte, se trata de buscar otras opciones al sistema convencional de accionamiento de articulaciones, empleándose para ello conceptos biomecánicos. De esta forma, se investiga en manipuladores con actuadores tipo músculo tanto para el brazo como para la mano del robot (Nakano y otros, 1984; Akazawa y Fujii, 1986; Caldwell y Taylor, 1988; Galhano y otros, 1991).

### 1.3.3 Sensores y sistemas de control

Los sistemas de control de robots pueden considerarse funcionalmente descompuestos según una estructura jerárquica. En el nivel inferior se realizan las tareas de servocontrol y supervisión de las articulaciones. La mayor parte de los robots industriales actuales emplean servomecanismos convencionales con realimentaciones de posición y velocidad para generar señales de control sobre los actuadores de las articulaciones. Típicamente, los parámetros del controlador son fijos aunque varían significativamente las condiciones de trabajo con la carga o con el propio movimiento. Nótese que las cargas iniciales, acoplamientos entre articulaciones, y efectos de gravedad son todos dependientes de la posición. El problema se amplía al aumentar la velocidad. Como resultado, en la mayor parte de los robots industriales actuales, la velocidad de operación debe ser pequeña.

Para mejorar las prestaciones se investiga en técnicas para identificar modelos suficientemente fiables de la dinámica del robot y en métodos de control de articulaciones que permitan compensar las no linealidades y acoplamientos, y optimizar el comportamiento dinámico (An, Atkinson y Hollerback, 1988). Asimismo, se trabaja en nuevos métodos de control adaptativo, que permitan tener en cuenta los cambios en las condiciones de trabajo (Craig, 1988; Ortega y Spong, 1989), y en métodos de control con aprendizaje para mejorar progresivamente la respuesta en operaciones repetitivas, típicas en robots industriales.

El segundo nivel de control se ocupa de la generación de trayectorias, entendiendo por tal la evolución del órgano terminal cuando se desplaza de una posición a otra. El generador de trayectorias debe suministrar a los servomecanismos las referencias apropiadas para conseguir la evolución deseada del órgano terminal a partir de la especificación del movimiento deseado en el espacio de la tarea. Para obtener las referencias que corresponden a las articulaciones en un determinado punto del espacio de trabajo, es necesario resolver el modelo geométrico inverso que es no lineal.

Los niveles superiores se ocupan de la comunicación con el usuario, interpretación de los programas, percepción sensorial y planificación.

Los primeros robots industriales eran programados exclusivamente por guiado manual, almacenando la secuencia de posiciones en memoria digital. La interacción con la tarea se limitaba a la apertura o cierre de una pinza u otro órgano terminal, indicándolo a un equipo externo, o esperando una señal de sincronización. Las aplicaciones típicas eran de "pick and place", tales como la carga y descarga de máquinas, realizando tareas con movimientos absolutamente definidos y fijos. Es decir, se primaba la repetibilidad sobre la adaptación. En cualquier caso, los robots podían ser reprogramados para la realización de otras tareas.

En la robótica industrial se han integrado los progresos en el control por computador de telemanipuladores, simultáneos al desarrollo de los primeros robots. Entre éstos cabe destacar los trabajos de Shannon y Minsky que, en 1958, propusieron un dispositivo, al que denominaron "sensor-controlled robot", que consistía en un teleoperador equipado con distintos sensores conectados a un computador que le suministraba información suficiente para decidir las acciones necesarias en orden a alcanzar un determinado objetivo. Aunque el dispositivo no llegó a realizarse, motivó a otros investigadores tales como Ernst (1961) que, en su tesis doctoral en el MIT, construyó un robot con sensores de tacto en la mano, que podía ser programado para realizar tareas tales como la localización, agarre, transporte y descarga de pequeñas piezas en cajas. Este robot, que puede ser considerado como el primero controlado mediante sensores externos, se programaba mediante instrucciones parecidas a las de un lenguaje ensamblador, incorporando órdenes relativas a la información de los sensores de tacto.

Continuando con el empleo de sensores de percepción del entorno, puede mencionarse el importante trabajo de Roberts (1963), demostrando la posibilidad de procesar una imagen digitalizada para obtener una descripción matemática de los objetos incluidos en la escena, expresando su posición y orientación mediante transformaciones homogéneas.

Wichman (1967) presentó en Stanford un equipo con cámara de televisión conectada a un computador que podía, en tiempo real, identificar objetos y sus posiciones.

Nótese que interesa que el programador pueda expresar los movimientos en coordenadas cartesianas y no en los ángulos o desplazamientos de las articulaciones. Pieper (1968) obtuvo una solución al problema aplicando resultados teóricos de la Cinemática. De esta forma fue posible facilitar la programación del robot utilizando llamadas a subrutinas que realizan las transformaciones geométricas correspondientes. En general, la utilización de coordenadas cartesianas fue un importante paso que abrió el camino al empleo de modelos del universo para la toma de decisiones, planificación y verificación de tareas.

Desde el comienzo de los años setenta, se investiga en robots con sensores de visión, resolviendo en tiempo real problemas básicos de manipulación con visión en color (Feldman, 1971) y ensamblado de estructuras de bloques (Eijiri y otros, 1972).

Se progresó también en métodos de cálculo de trayectorias con generación de consignas a los servos de las articulaciones y en lenguajes de programación de mayor nivel que incorporan primitivas relacionadas con sensores de percepción del entorno y especificación de movimientos en coordenadas cartesianas.

Al final de los años setenta y comienzo de los ochenta se adoptan lenguajes estructurados (Paul, 1981) con herramientas de programación en tiempo real que progresivamente se introducen en los robots industriales comerciales.

Asimismo, se trabaja en lenguajes de programación orientados a la tarea basados en la incorporación de métodos de la Inteligencia Artificial para generación automática de planes, permitiendo también coordinar la actividad de un robot en sistemas de fabricación flexible.

Por lo que respecta a la planificación de caminos libres de obstáculos, el método típico se basa en construir una estructura de datos que represente la geometría del espacio de trabajo o las restricciones existentes y, a continuación, se utiliza la estructura para encontrar el camino (Lozano-Pérez y Wesley, 1979; Lozano-Pérez 1983; Brooks,

1983). No obstante, estos métodos son, en general, costosos desde el punto de vista computacional, lo que suele impedir su aplicación en tiempo real. Existen también métodos que incorporan la planificación de caminos en el control de bajo nivel utilizando para ello métodos tales como el de los campos potenciales (Khatib, 1986).

En la pasada década, surge también el concepto de realimentación visual en control de manipuladores, con aplicaciones al agarre de objetos en movimiento (Weiss y otros, 1987; Feddema y Mitchell, 1989). En IEEE (1998) pueden encontrarse diversas aplicaciones de realimentación visual.

Por otra parte, desde los años ochenta se progresó en la manipulación diestra de objetos mediante una mejor comprensión de la mecánica de la manipulación y su planificación (Mason y Salisbury, 1985; Sturges, 1990).

Para acabar este apartado conviene poner de manifiesto las importantes demandas en teoría de control, sistemas de percepción y aprendizaje, sistemas informáticos en tiempo real, y nuevos mecanismos, que se requieren para resolver los problemas planteados por el control de estructuras articuladas y la manipulación de objetos.

Se han aplicado también técnicas de control tales como las basadas en redes neuronales (Kawato y Uno, 1988; Goldberg y Pearlmuter, 1988; Miller y otros, 1990), y otras estructuras de control inteligente, que están permitiendo resolver problemas que son de elevada complejidad con métodos tradicionales. En general, junto a los progresos tecnológicos, se requieren desarrollos teóricos que permitan formular una metodología de diseño de estos nuevos sistemas de control, en los que se involucran bucles de realimentación sensorial y procesos de decisión y aprendizaje, que son difíciles de tratar con los métodos convencionales de la teoría de control.

## 1.4 ROBOTS MÓVILES

---

El desarrollo de robots móviles responde a la necesidad de extender el campo de aplicación de la Robótica, restringido inicialmente al alcance de una estructura mecánica anclada en uno de sus extremos. Se trata también de incrementar la autonomía limitando todo lo posible la intervención humana.

Desde el punto de vista de la autonomía, los robots móviles tienen como precedentes los dispositivos electromecánicos, tales como los denominados "micro-mouse", creados desde los años treinta para desarrollar funciones inteligentes tales como descubrir caminos en laberintos. Cabe destacar la tortuga de Walter, presentada en 1948, que podía reaccionar ante la presencia de obstáculos, subir pendientes y, cuando la alimentación comenzaba a ser insuficiente, dirigirse hacia una posición de recarga.

Estos trabajos de investigación no guardan una relación directa con los vehículos autónomos que comenzaron a aplicarse desde los años sesenta en la industria, siendo guiados por cables bajo el suelo o mediante sensores ópticos para seguir líneas trazadas en la planta. Estas aplicaciones, hoy día comunes en muchos procesos de fabricación, se caracterizan por un entorno fuertemente estructurado para facilitar la automatización.

En los años setenta se vuelve a trabajar en el desarrollo de robots móviles dotados de una mayor autonomía. La mayor parte de las experiencias se desarrollan empleando plataformas que soportan sistemas de visión (Moravec, 1981). Sin embargo, el desa-

rrollo tecnológico todavía no era el suficiente para lograr la navegación autónoma de forma eficiente. En los años ochenta el incremento espectacular de la capacidad computacional y el desarrollo de nuevos sensores, mecanismos y sistemas de control, permite aumentar la autonomía. En esta década cabe mencionar los desarrollos de robots móviles, tanto para interiores como para navegación exterior (Bares y otros, 1988; Thorpe, 1990), realizados en la Carnegie Mellon University (Pittsburgh, EE.UU.).

Se trata de que el robot tenga la suficiente inteligencia como para reaccionar y tomar decisiones basándose en observaciones de su entorno, sin suponer que este entorno es perfectamente conocido.

La autonomía de un robot móvil se basa en el sistema de navegación automática. En estos sistemas se incluyen tareas de planificación, percepción y control.

En los robots móviles, el problema de la planificación, en el caso más general, puede descomponerse en planificación global de la misión, de la ruta, de la trayectoria y, finalmente, evitar obstáculos no esperados.

En un robot para interiores, la misión podría consistir en determinar a qué habitación hay que desplazarse, mientras que la ruta establecería el camino desde la posición inicial a una posición en la habitación, definiendo puntos intermedios de paso. El vehículo puede desviarse de la ruta debido a la acumulación de imprecisiones mecánicas y de control.

Existen numerosos métodos de planificación de caminos para robots móviles que se basan en hipótesis simplificadoras similares a las mencionadas en el apartado 1.3: entorno conocido y estático, robots omnidireccionales, con movimiento lento y ejecución perfecta de trayectoria. En particular, hay muchos métodos que buscan caminos libres de obstáculos que minimizan la distancia recorrida en un entorno modelado mediante polígonos. En otros casos, se modela el espacio libre tratando de encontrar caminos por el centro del mismo. Para facilitar la búsqueda existen técnicas de descomposición del espacio en celdas (Thorpe, 1984), utilización de restricciones de varios niveles de resolución y búsqueda jerarquizada (Kambhampati y Davis, 1986; Stenz, 1990) que permiten hacer más eficiente el proceso con vistas a su aplicación en tiempo real.

La planificación de la trayectoria puede realizarse también de forma dinámica, considerando la posición actual del vehículo y los puntos intermedios de paso definidos en la planificación de la ruta. La trayectoria se corrige debido a acontecimientos no considerados. La definición de la trayectoria debe tener en cuenta las características cinemáticas del vehículo. Por ejemplo, en vehículos con ruedas y tracción convencional, interesa definir trayectorias de curvatura continua que puedan ejecutarse con el menor error posible (Kanayama y Hartman, 1989; Nelson, 1989).

Además de las características geométricas y cinemáticas, puede ser necesario tener en cuenta modelos dinámicos de comportamiento del vehículo contemplando la interacción vehículo-terreno. Por otra parte, puede plantearse también el problema de la planificación de la velocidad teniendo en cuenta las características del terreno y del camino que se pretende seguir.

Una vez realizada la planificación de la trayectoria, es necesario planificar movimientos concretos y controlar dichos movimientos para mantener al vehículo en la trayectoria planificada. De esta forma, se plantea el problema del seguimiento de caminos, que para vehículos con ruedas se concreta en determinar el ángulo de dirección teniendo en cuenta la posición y orientación actual del vehículo con respecto a la tra-

yectoria que debe seguir. Asimismo, es necesario resolver el problema del control y regulación de la velocidad del vehículo.

Conviene mencionar también los métodos que permiten la integración de la planificación con el control del vehículo. Entre éstos cabe mencionar el de los campos potenciales (Khatib, 1986; Borenstein y Koren, 1989). La idea consiste en determinar la resultante de fuerzas que atraen el robot hacia el objetivo y que lo repelen de los obstáculos.

En cualquier caso, el problema del control automático preciso de un vehículo con ruedas puede resultar más complejo que el de los manipuladores debido a la presencia de restricciones no holónomas. Los bucles de control se plantean tanto en el espacio de las variables articulares como en coordenadas del mundo, y las ecuaciones de movimiento son complejas, si se considera la interacción con el terreno. Mientras en manipuladores es relativamente fácil el cálculo y medida de los pares y fuerzas que se ejercen sobre la estructura mecánica, la determinación de estos pares en vehículos con ruedas es muy difícil. En la actualidad se emplean fundamentalmente métodos geométricos y modelos cinemáticos simplificados. No obstante, la consideración de aspectos dinámicos es necesaria cuando la velocidad es alta.

Nótese también que el control del vehículo requiere disponer de medidas de su posición y orientación, a intervalos suficientemente cortos. La técnica más simple consiste en la utilización de la odometría a partir de las medidas suministradas por los sensores situados en los ejes de movimiento, típicamente codificadores ópticos. Sin embargo, la acumulación de error puede ser muy grande. Se emplean también sistemas de navegación inercial incluyendo giróscopos y acelerómetros, aunque estos sistemas también acumulan error, especialmente en la determinación de la posición empleando los acelerómetros. No obstante, la combinación de las técnicas odométricas con la medida de los ángulos de orientación puede dar buenos resultados en intervalos de tiempo y distancia viajada suficientemente pequeños.

La corrección de la inevitable acumulación de error hace necesario el empleo de otros sensores. Con este fin, en aplicaciones de exteriores, en las que las distancias que recorre el vehículo autónomo son considerables, se emplean sistemas de posicionamiento global mediante satélites. Asimismo, la incertidumbre de la posición se reduce a intervalos suficientemente grandes empleando el sistema de percepción (Salichs y otros, 1999). En particular, suelen emplearse marcas especiales cuya detección permite estimar con precisión la posición del robot (Cao y otros, 1986). En entornos no estructurados, con ausencia de marcas especiales, la estimación de la posición mediante el sistema de percepción es notablemente más compleja (Asensio y otros, 1999).

El sistema de percepción de un robot móvil o vehículo autónomo tiene un triple objetivo: permitir una navegación segura, detectando y localizando obstáculos y situaciones peligrosas en general, modelar el entorno construyendo un mapa o representación de dicho entorno (fundamentalmente geométrica), y estimar la posición del vehículo de forma precisa. Asimismo, el sistema de percepción de estos robots puede aplicarse no sólo para navegar sino también para aplicaciones tales como el control de un manipulador situado en el robot.

Para el diseño de estos sistemas de percepción deben tenerse en cuenta diferentes criterios, algunos de los cuales son conflictivos entre sí. De esta forma, es necesario considerar la velocidad del robot, la precisión, el alcance, la posibilidad de interpretación errónea de datos y la propia estructura de la representación del entorno.

En muchas aplicaciones se requiere tener en cuenta diversas condiciones de navegación con requerimientos de percepción diferentes. De esta forma, puede ser necesario estimar de forma muy precisa, aunque relativamente lenta, la posición del robot y, a la vez, detectar obstáculos lo suficientemente rápido, aunque no se necesite una gran precisión en su localización.

Existen también arquitecturas en las que el sistema de percepción se encuentra integrado en el controlador de forma que, en entornos estructurados, es posible estimar de forma muy rápida la posición para navegar a alta velocidad (Dickman y Zapp, 1987). Asimismo, se han aplicado redes neuronales (Pomerleau, 1990) para generar el ángulo de dirección a partir del sistema de percepción.

Conviene mencionar también el interés del empleo de técnicas de procesamiento en paralelo para el tratamiento de imágenes en el guiado autónomo de vehículos (Ferruz y Ollero, 2000).

Con respecto a los sensores específicos, además de las características de precisión, rango, e inmunidad a la variación de condiciones del entorno, es necesario tener en cuenta su robustez ante vibraciones y otros efectos originados por el vehículo y el entorno, su tamaño, consumo, seguridad de funcionamiento y desgaste.

Las cámaras de vídeo tienen la ventaja de su amplia difusión y precio, su carácter pasivo (no se emite energía sobre el entorno) y que no es necesario, en principio, el empleo de dispositivos mecánicos para la captación de la imagen. Las desventajas son los requerimientos computacionales, la sensibilidad a las condiciones de iluminación, y los problemas de calibración y fiabilidad.

La percepción activa mediante láser es un método alternativo que ha cobrado una importante significación en robots móviles (Hebert, 1990). Se utilizan dispositivos mecánicos y ópticos de barrido en el espacio obteniéndose imágenes de distancia y reflectancia a las superficies intersectadas por el haz.

Los sensores de ultrasonido son económicos y simples para la navegación (Moravec y Elfes, 1985; Elfes, 1987). Se basan en la determinación del denominado tiempo de vuelo de un pulso de sonido (entre 30 kHz y 1 MHz). Sin embargo, la influencia de las condiciones ambientales puede ser significativa, debiendo corregirse mediante una calibración adecuada. Por otra parte, la relación señal/ruido es normalmente muy inferior a la de los otros sensores, lo que puede hacer necesario el empleo de múltiples frecuencias y técnicas de filtrado y tratamiento de la incertidumbre de mayor complejidad computacional. Asimismo, la resolución lateral es mala, existiendo para evitarlo técnicas de enfoque mediante lentes acústicas o transmisores curvos.

## 1.5 ROBOTS AUTÓNOMOS Y TELEROBÓTICA

De acuerdo con su grado de autonomía, los robots pueden clasificarse en teleoperados, de funcionamiento repetitivo y autónomos o inteligentes.

En los *robots teleoperados* las tareas de percepción del entorno, planificación y manipulación compleja son realizadas por humanos. Es decir, el operador actúa en tiempo real cerrando un bucle de control de alto nivel. Los sistemas evolucionados suministran al operador realimentación sensorial del entorno (imágenes, fuerzas, distancias). En manipulación se emplean brazos y manos antropomórficos con controladores

automáticos que reproducen los movimientos del operador. Alternativamente, el operador mueve una réplica a escala del manipulador, reproduciéndose los movimientos en éste (Hirzinger y otros, 1991).

Estos robots son interesantes para trabajos en una localización remota (acceso difícil, medios contaminados o peligrosos), en tareas difíciles de automatizar y en entornos no estructurados, tales como las que se realizan en la construcción o en el mantenimiento de líneas eléctricas (Peñín y otros, 1998).

Las mayores dificultades radican en las limitaciones del hombre en la capacidad de procesamiento numérico y precisión y, sobre todo, en el acoplamiento y coordinación entre el hombre y robot. En algunas aplicaciones, el retraso de transmisión de información juega también un papel importante y su consideración resulta fundamental en el diseño del sistema de control. El diseño de la interfase persona-máquina suele ser crítico. La investigación actual se dirige a hacer recaer en el operador únicamente las tareas que requieren toma de decisiones en función de información sensorial, experiencia, y habilidad (Vertut y Coiffet, 1985; Bejczy, 1991). No obstante, existen limitaciones por el ancho de banda de la transmisión y, eventualmente, por la complejidad de la tarea del operador.

Los *robots de funcionamiento repetitivo* son la mayor parte de los que se emplean en cadenas de producción industrial. Trabajan normalmente en tareas predecibles e invariantes, con una limitada percepción del entorno. Son precisos, de alta repetibilidad y relativamente rápidos; incrementan la productividad ahorrando al hombre trabajos repetitivos y, eventualmente, muy penosos o incluso peligrosos.

Los *robots autónomos o inteligentes* son los más evolucionados desde el punto de vista del procesamiento de información. Son máquinas capaces de percibir, modelar el entorno, planificar y actuar para alcanzar objetivos sin la intervención, o con una intervención muy pequeña, de supervisores humanos. Pueden trabajar en entornos poco estructurados y dinámicos, realizando acciones en respuesta a contingencias variadas en dicho entorno.

Durante las últimas décadas se han realizado importantes esfuerzos en la aplicación de técnicas de Inteligencia Artificial. Se han empleado métodos simbólicos de tratamiento de la información basados en modelos geométricos del entorno. Las dificultades surgen por la elevada capacidad de procesamiento requerida para tratar en tiempo real problemas suficientemente significativos para muchas aplicaciones y, sobre todo, por la propia incertidumbre de la información del entorno. De esta forma, se resuelven problemas basados en un modelo previo del entorno cuyas soluciones sólo son válidas si el modelo corresponde exactamente a la realidad. La técnica obvia de reducir esta incertidumbre consiste en incrementar la información que se dispone de dicho entorno mediante realimentación sensorial. Existen métodos que permiten intercalar la formulación y ejecución de planes con la captación de la información necesaria para asegurar que el modelo que se utiliza para la planificación sea lo suficientemente fiable. Las limitaciones vienen impuestas por el sistema de percepción y por la propia arquitectura del sistema de información y control del robot.

Desde el punto de vista de la planificación, existen diferentes arquitecturas diseñadas teniendo en cuenta especificaciones sobre el tiempo que tiene el sistema para responder y la disponibilidad de información potencialmente interesante.

La solución se sitúa normalmente entre dos extremos, en uno de los cuales está la planificación puramente estratégica. En este caso, se supone que la situación en la que va a ejecutarse el plan puede ser predecida de forma suficientemente precisa durante la

planificación (Thorpe, 1990). En el otro extremo se sitúa la planificación puramente reactiva en la que se supone que el entorno es incierto, buscándose la mayor flexibilidad posible para reaccionar en cualquier instante lo suficientemente rápido a las discrepancias entre el modelo actual y la realidad observada en el entorno (Fukuda y otros, 1988; Brooks y Flynn, 1989; Dario y otros, 1991).

El problema puede plantearse también en términos de un compromiso entre eficiencia y flexibilidad. En efecto, las arquitecturas diseñadas para conseguir la mayor flexibilidad ante cualquier eventualidad del entorno son mucho menos eficientes que las que utilizan criterios de decisión basados en modelos del entorno suficientemente precisos sin tener demasiado en cuenta la posibilidad de generalizar el comportamiento. En este punto conviene poner de manifiesto el interés de las arquitecturas con capacidad de aprendizaje (Mitchell, 1990) que combinan la planificación estratégica, basada en técnicas de búsqueda, con la planificación puramente reactiva.

## 1.6 CONCLUSIONES

Este capítulo se ha dedicado a introducir nociones básicas sobre la robótica y los problemas que se involucran en el desarrollo de robots.

El sistema robot integra componentes mecánicos, eléctricos, electrónicos e informáticos.

En este capítulo se han introducido las tecnologías que se emplean en los actuadores, sensores y sistemas de control de los robots. Se han considerado tanto robots manipuladores como robots móviles.

Finalmente, se ha presentado una clasificación en robots teleoperados, robots de funcionamiento repetitivo y robots autónomos o "inteligentes". Estos últimos son los más evolucionados desde el punto de vista del procesamiento de la información y objeto de actividades de investigación y desarrollo, pero todavía no se han empleado de forma masiva en las aplicaciones.

## 1.7 REFERENCIAS

- Akazawa K. y K. Fujii, 1986. "Theory of muscle contraction and motor control". Advanced Robotics. Vol. 1, pp. 379-390.
- An C.H., C.G. Atkinson y J.M. Hollerback, 1988. "Model-based control of a robot manipulator". MIT Press, Cambridge, Mass.
- Asensio J.R., J.M. Montiel y L. Montano, 1999. "Goal Directed Reactive Robot Navigation with Relocation Using Laser and Vision". Proc. IEEE International Conference on Robotics and Automation, ICRA'99. Vol. 4, pp. 2905-2910. Detroit, Michigan, USA.
- Bares J., M. Hebert, T. Kanade, E. Krotkov, T. Mitchell, R. Simmons y W. Whittaker, 1988. "Ambler, an autonomous robot for planetary exploration". IEEE Computer.
- Bejczy A., 1991. "Recent developments in advanced teleoperation at JPL". Proc. of the fifth International Conference on Advanced Robotics. Pisa (Italia), pp. 217-218.
- Borenstein J. y Y. Koren, 1989. "Real-time obstacle avoidance for fast mobile robots". IEEE Transactions on Systems Man and Cybernetics. Vol. 19, pp. 1179-1187.
- Brooks R.A., 1983. "Solving the find-path problem by good representation of free space". IEEE Transactions on System Man and Cybernetics. Vol 13, pp. 190-197.

## Referencias

---

- Brooks R.A. y M.A. Flynn, 1989. "Robot Beings". Proc. of IEEE Int. Workshop on Intelligent Robots and Systems, pp. 2-9, Tsukuba (Japón).
- Caldwell D.G. y P.M. Taylor, 1988. "Artificial muscles as robotic actuators". Proc. SYROCO'88. Karlsruhe, pp. 40.1-40.6.
- Cao Z.L., J.J. Roning y E.L. Hall, 1986. "Omnidirectional vision navigation integrating beacon recognition with positioning". Proc. SPIE Conference on Intelligent Robots and Computer Vision.
- Craig J.J., 1988. "Adaptive control of mechanical manipulators". Addison Wesley.
- Dario P., V. Genovese, F. Ribechni y G. Sandini, 1991. "Instinctive cellular robots". Proc. of the fifth International Conference on Advanced Robotics. Pisa (Italia), pp. 551-555.
- Dickman E. y A. Zapp, 1987. "Autonomous high-speed road vehicle guidance by computer vision". Proc. of the 10 IFAC World Congress.
- Eames, 1973. "A computer perspective". Harvard University Press, Harvard, Mass.
- Eijiri M., T. Uno, H. Yoda, T. Goto y K. Takeyasu, 1972. "A prototype intelligent robot that assembles objects from plane drawings". IEEE Trans. Computers. Vol. C-21, núm. 2, pp. 161-170.
- Elfes A., 1987. "Sonar based real world mapping and navigation". IEEE Journal of Robotics and Automation. Vol. RA-3, núm. 3.
- Ernst H.A., 1961. "A computer operated mechanical hand". Sc.D. Thesis, MIT, Cambridge, Mass.
- Feddeema J.T. y O.R. Mitchell, 1989. "Vision-guided servoing with feature-based trajectory generation". IEEE Trans. Robotics and Automation, Vol. 5, núm. 5, pp. 691-700.
- Feldman J., 1971. "The use of vision and manipulation to solve the puzzle". Proc. Second Int. Joint Conference on Artificial Intelligence. Londres, pp. 359-364.
- Ferruz J. y A. Ollero, 2000. "Real-time feature matching in image sequences for non-structured environments. Applications to vehicle guidance". Journal of Intelligent and Robotics Systems, Kluwer Academic Publisher. Vol. 28, Issue 1/2, pp. 85-123.
- Fukuda T., S. Nakagawa, Y. Kawauchi y M. Buss, 1988. "Self organizing robots based on cell structures-CEBOT". Proc. IEEE INT. Workshop on Intelligent Robots and Systems, pp. 145-150, Tokio (Japón).
- Galhano A.M.S.F., J.A.T. Machado y J.L.M. Carvalho, 1991. "On the analysis of muscle-actuated manipulators". Proc. of the fifth International Conference on Advanced Robotics. Pisa (Italia), pp. 67-71.
- Goertz R.C., 1963. "Manipulators used for handling radioactive materials". En "Human Factors in Technology", E.M. Bennett (ed.), McGraw-Hill, pp. 425-443.
- Goldberg K. y B. Pearlmuter, 1988. "Using a neural network to learn the dynamics of the CMU direct-drive arm II". Technical Report CMU-CS-88-160, Carnegie Mellon University.
- Hebert M., 1990. "3-D vision techniques for autonomous vehicles". En "Vision and Navigation: The CMU NavLab". Kluwer Academic Publishers, capítulo 8.
- Hirzinger G., J. Heindl y K. Landzettel, 1991. "Control Structures in sensor-based telerobotic-systems". Proc. of the fifth International Conference on Advanced Robotics. Pisa (Italia), pp. 267-276.
- IEEE Robotics and Automation Magazine, 1998. "Special issue on visual servoing". IEEE Robotics and Automation Magazine.
- Kambhampati S. y L.S. Davis, 1986. "Multiresolution path planning for mobile robots". IEEE Journal of Robotic and Automation. Vol. 2, pp. 135-145.
- Kanayama I. y B. Hartman, 1989. "Smooth local path planning for autonomous vehicles". Proc. IEEE Conference on Robotics and Automation, pp. 1265-1270.
- Kawato, M. y T. Uno, 1988. "Hierarchical neural network model for voluntary movement with applications to robotics". IEEE Control System Magazine. Vol. 8, pp. 8-15.

- Khatib O., 1986. "Real time obstacle avoidance for manipulators and mobile robots". IEEE Journal of Robotics and Automation. Vol. 2, pp. 90-98.
- Lozano-Pérez T. y M.A. Wesley, 1979. "An algorithm for planning collision free path among polyhedral obstacles". Communications of ACM. Vol. 22, pp. 560-570.
- Lozano-Pérez T., 1983. "Spatial planning: a configuration approach". IEEE Trans. Computers. Vol. 32, pp. 108-120.
- Mason M.J. y J.K. Salisbury, 1985. "Robots hands & the mechanics of manipulation". MIT Press.
- Miller T., R. Hewes, F. Glanz y G. Kraft, 1990. "Real-time dynamic control of an industrial manipulator using a neural-network-based learning controller". IEEE Transactions on Robotics and Automation. Vol. 6, núm. 1.
- Mitchell, 1990. "Becoming increasingly reactive". Proc. of the 1990 AAAI Conference, Agosto 1990, Boston.
- Moravec H., 1981. "Robot rover visual navigation". UMI Research Press.
- Moravec H. y A. Elfes, 1985. "High resolution maps from a wide-angle sonar". Proc. IEEE Conference on Robotics and Automation.
- Nakano Y., M. Fujie y Y. Hosodo, 1984. "Hitachi's robothand". Robotics Age, pp. 18-20.
- Nelson W., 1989. "Continuous curvature path for autonomous vehicles". Proc. IEEE Int. Conference on Robotics and Automation, pp. 1260-1264.
- Ollero A., 1991. "Evolución y perspectivas de la robótica". Texto de la lección inaugural del curso 1991-1992 en la Universidad de Málaga. Secretariado de Publicaciones de la Universidad de Málaga.
- Ortega R. y M.W. Spong, 1989. "Adaptive motion control of rigid robots: a tutorial". Automatica. Vol. 25, pp. 887-888.
- Paul R.P., 1981. "Robot manipulators: mathematics, programming and control". MIT Press.
- Pefián, L.F., R. Aracil, M. Ferre, E. Pinto, M. Hernando y A. Barrientos, 1998. "Telerobotic system for live power lines maintenance: ROBTET". Proc. IEEE International Conference on Robotics & Automation, pp. 2110-2115. Leuven, Bélgica.
- Pomerleau D., 1990. "Neural network based autonomous navigation". En "Vision and Navigation: The CMU NavLab". Kluwer Academic Publishers, capítulo 5.
- Rosheim M.E., 1989. "Robot wrist actuators". Wiley.
- Salichs M.A., J.M. Armingol, L.E. Moreno y A. de la Escalera, 1999. "Localization system for mobile robots in indoor environments". Integrated Computer Aided Engineering. Vol. 6, núm. 4, pp. 303-318.
- Silva M., 1984. "Curso de introducción y aplicaciones de la robótica". Primer Salón internacional de tecnología y aplicaciones de la robótica". Zaragoza, 1984.
- Stenz A., 1990. "Multiresolution constraint modelling for mobile robot path-planning". En "Vision and Navigation: The CMU NavLab". Kluwer Academic Publishers, capítulo 11.
- Sturges R.H., 1990. "A quantification of machine dexterity applied to an assembly task". Int. Journal of Robotic Research. Vol. 9, núm. 3.
- Thorpe C., 1984. "FIDO: Vision and navigation for a robot rover". Ph. D. Thesis, Carnegie Mellon University.
- Thorpe C., 1990. "Vision and Navigation: The CMU NavLab". Kluwer Academic Publishers.
- Vertut J. y P. Coiffet, 1985. "Teleoperations and Robotics". Vol. 1 y 2. Kogan Page, Prentice Hall.
- Weiss L.E., A.C. Sanderson y C.P. Neuman, 1987. "Dynamic sensor-based control of robots with visual feedback". IEEE Journal of Robotics and Automation, RA-3, núm 5, pp. 404-417.

# Morfología de los robots

---

En este capítulo se describen las características básicas de la estructura de los robots incluyendo tanto robots manipuladores como móviles. Se comienza considerando la estructura de los primeros comentando brevemente los tipos de articulaciones y configuraciones clásicas de brazos de robots industriales y órganos terminales. Se incluyen también comentarios sobre las nuevas estructuras de robots manipuladores. A continuación se tratan los robots móviles, estudiando los diferentes sistemas de locomoción de los robots con ruedas. Posteriormente se introducen vehículos robóticos con otros tipos de sistemas de locomoción.

## **2.1 ESTRUCTURA DE ROBOTS MANIPULADORES**

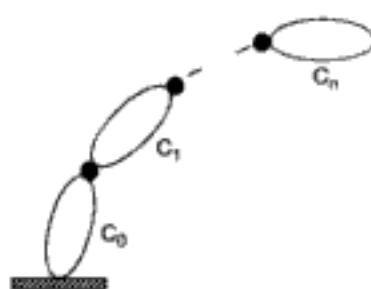
---

Los robots manipuladores son, esencialmente, brazos articulados. De forma más precisa, un manipulador industrial convencional es una cadena cinemática abierta formada por un conjunto de eslabones o elementos de la cadena interrelacionados mediante articulaciones o pares cinemáticos, tal como se ilustra en la figura 2.1. Las articulaciones permiten el movimiento relativo entre los sucesivos eslabones.

### **2.1.1 Tipos de articulaciones**

Existen diferentes tipos de articulaciones. Las más utilizadas en robótica son las que se indican en la figura 2.2 (Ferraté y otros, 1986). La articulación de rotación suministra un grado de libertad consistente en una rotación alrededor del eje de la articulación. Esta articulación es, con diferencia, la más empleada.

En la articulación prismática el grado de libertad consiste en una traslación a lo largo del eje de la articulación.

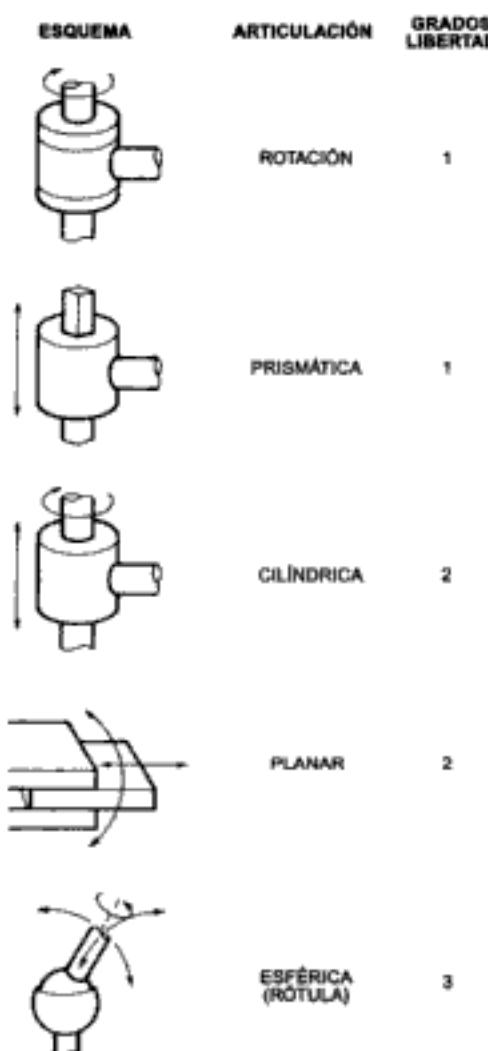


**Figura 2.1:** Cadena cinemática abierta.

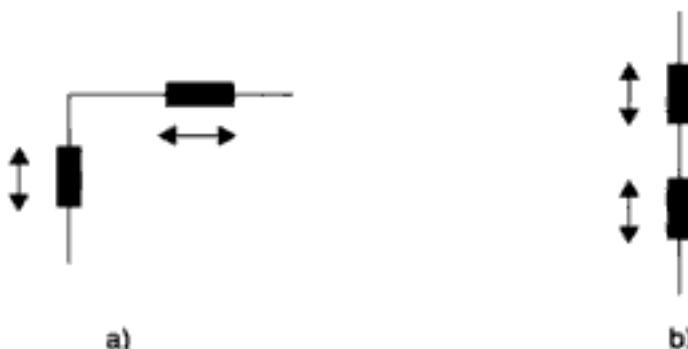
En la articulación cilíndrica existen dos grados de libertad: una rotación y una translación, como se indica en la figura 2.2.

La articulación planar está caracterizada por el movimiento de desplazamiento en un plano que se muestra en la figura 2.2, existiendo, por tanto, dos grados de libertad.

Por último, la articulación esférica combina tres giros en tres direcciones perpendiculares en el espacio.



**Figura 2.2:** Tipos de articulaciones (Ferraté y otros, 1986).



**Figura 2.3:** Pérdida de grados de libertad en estructura con dos eslabones: a) dos grados de libertad y b) un grado de libertad.

Los grados de libertad son el número de parámetros independientes que fijan la situación del órgano terminal. El número de grados de libertad suele coincidir con el número de eslabones de la cadena cinemática. Así, en la figura 2.3a, se ilustra una estructura con dos eslabones, dos articulaciones prismáticas y dos grados de libertad.

Sin embargo, pueden existir casos degenerados, tal como el que se ilustra en la figura 2.3b en la cual se aprecia que, aunque existan dos eslabones y dos articulaciones prismáticas, tan sólo se tiene un grado de libertad. Por consiguiente, en general, el número de grados de libertad es menor o igual que el número de eslabones de la cadena cinemática.

### 2.1.2 Estructuras básicas

La estructura típica de un manipulador consiste en un brazo compuesto por elementos con articulaciones entre ellos. En el último enlace se coloca un órgano terminal o efector final tal como una pinza o un dispositivo especial para realizar operaciones.

Se consideran, en primer lugar, las estructuras más utilizadas como brazo de un robot manipulador. Estas estructuras tienen diferentes propiedades en cuanto a espacio de trabajo y accesibilidad a posiciones determinadas. En la figura 2.4 se muestran cuatro configuraciones básicas.

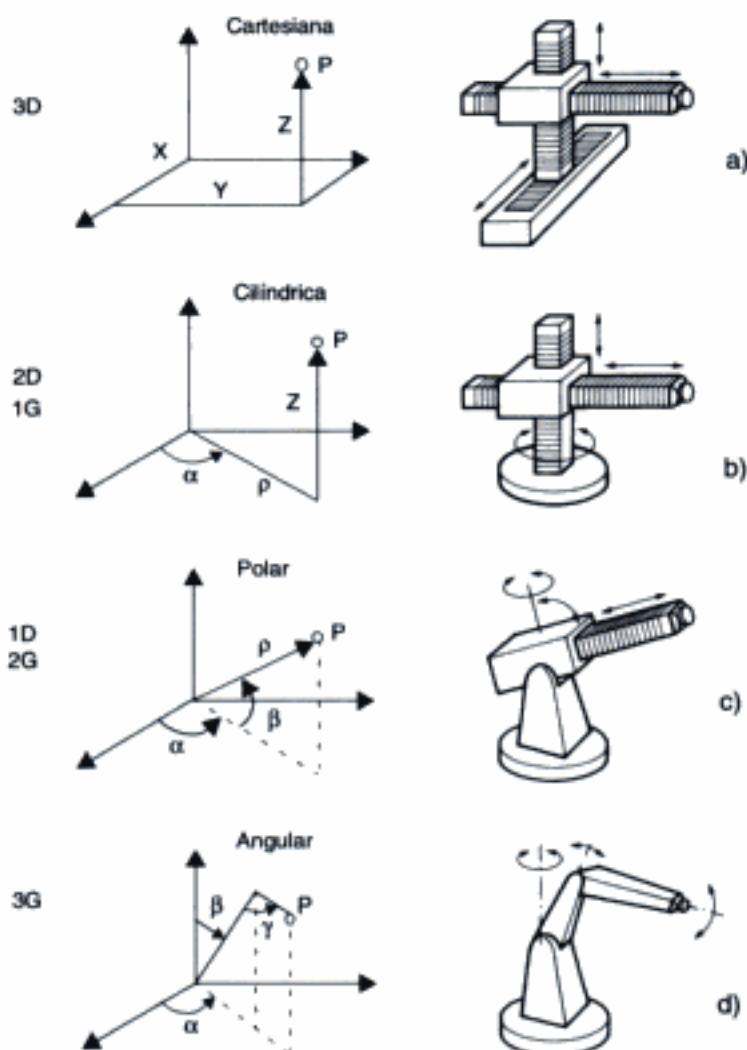
El espacio de trabajo es el conjunto de puntos en los que puede situarse el efecto final del manipulador. Corresponde al volumen encerrado por las superficies que determinan los puntos a los que accede el manipulador con su estructura totalmente extendida y totalmente plegada.

Por otra parte, todos los puntos del espacio de trabajo no tienen la misma accesibilidad. Los puntos de accesibilidad mínima son los de las superficies que delimitan el espacio de trabajo ya que a ellos sólo puede llegar con una única orientación.

#### 2.1.2.1 Configuración cartesiana

Se ilustra en la figura 2.4a. La configuración tiene tres articulaciones prismáticas (3D o estructura PPP). Esta configuración es bastante usual en estructuras industriales.

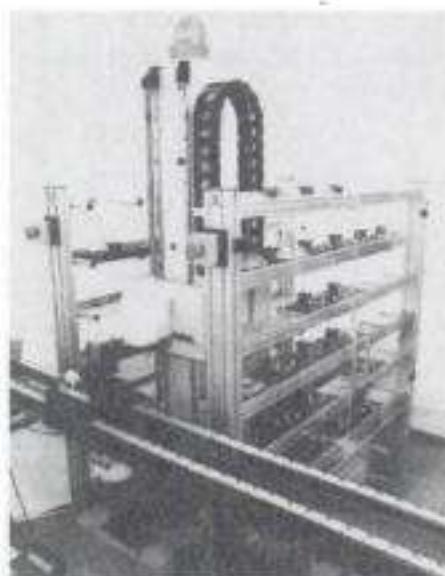
les, tales como pórticos, empleadas para el transporte de cargas voluminosas. En la figura 2.5 se muestra un almacén automático servido por un robot que tiene básicamente la configuración cartesiana.



**Figura 2.4:** Configuraciones básicas de robots manipuladores industriales (Ferraté y otros, 1986).

La especificación de la posición de un punto se efectúa mediante las coordenadas cartesianas ( $x$ ,  $y$ ,  $z$ ). Los valores que deben tomar las variables articulares corresponden directamente a las coordenadas que toma el extremo del brazo. Por consiguiente, en esta configuración, se simplifica la tarea del controlador del robot que debe generar las órdenes para ejecutar una trayectoria definida mediante una secuencia de puntos expresados en coordenadas cartesianas.

Sin embargo, la configuración no resulta adecuada para acceder a puntos situados en espacios relativamente cerrados y su volumen de trabajo es pequeño cuando se compara con el que puede obtenerse con otras configuraciones. Nótese que si se tienen tres segmentos de variación  $L$ , el volumen de trabajo sería  $L^3$ .



**Figura 2.5:** Almacén automático con manipulador cartesiano en los laboratorios del Instituto Andaluz de Automática y Robótica, Escuela Superior de Ingenieros de la Universidad de Sevilla.

#### 2.1.2.2 Configuración cilíndrica

Esta configuración tiene dos articulaciones prismáticas y una de rotación (2D, 1G). La primera articulación es normalmente de rotación (estructura RPP), como se muestra en la figura 2.4b. La posición se especifica de forma natural en coordenadas cilíndricas.

Obsérvese que esta configuración puede ser de interés en una célula flexible, con el robot situado en el centro de la célula sirviendo a diversas máquinas dispuestas radialmente a su alrededor.



**Figura 2.6:** Robot industrial con configuración cilíndrica RT3300 de Seiko.

En la figura 2.6 se muestra un robot industrial con configuración cilíndrica. El volumen de trabajo de esta estructura RPP (o de la PRP), suponiendo un radio de giro de 360 grados y un rango de desplazamiento de  $L$ , es el de un toro de sección cuadrada de radio interior  $L$  y radio exterior  $2L$ . Se demuestra que el volumen resultante es  $3\pi L^3$ .

### 2.1.2.3 Configuración polar o esférica

Esta configuración se caracteriza por dos articulaciones de rotación y una prismática (2G, 1D o estructura RRP). En este caso, las variables articulares expresan la posición del extremo del tercer enlace en coordenadas polares, tal como se muestra en la figura 2.4c.

En la figura 2.7 (Asociación Española de Robótica, 1987) se muestra un robot industrial con configuración esférica. Esta configuración permite un buen volumen de trabajo, sólo inferior a la angular, que se tratará a continuación.



Figura 2.7: Robot industrial con configuración polar.

En efecto, en un manipulador con tres enlaces de longitud  $L$ , el volumen de trabajo de esta estructura, suponiendo un radio de giro de 360 grados y un rango de desplazamiento de  $L$ , es el que existe entre una esfera de radio  $2L$  y otra concéntrica de radio  $L$ . Por consiguiente, el volumen es  $(28/3)\pi L^3$ .

### 2.1.2.4 Configuración angular

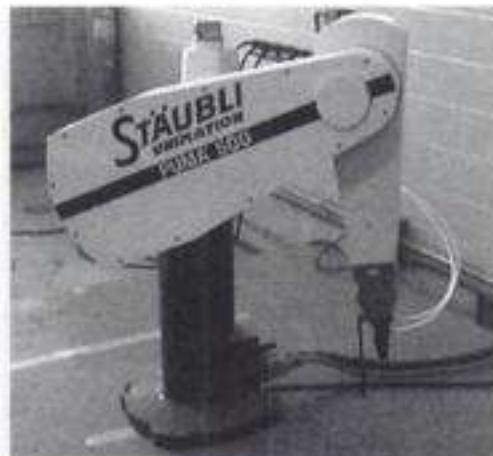
Esta configuración es una estructura con tres articulaciones de rotación (3G o RRR), tal como se muestra en la figura 2.4d. La posición del extremo final se especifica de forma natural en coordenadas angulares.

La estructura tiene un mejor acceso a espacios cerrados y es fácil desde el punto de vista constructivo. Es muy empleada en robots manipuladores industriales, especial-

mente en tareas de manipulación que tengan una cierta complejidad. En la figura 2.8 se muestra un robot industrial típico, existente en un gran número de laboratorios de investigación y desarrollo en robótica. De hecho, la configuración angular es, con mucho, la más utilizada en educación y actividades de investigación y desarrollo.

Obsérvese como, con esta estructura, es posible conseguir un gran volumen de trabajo. En efecto, si la longitud de los tres enlaces es de  $L$ , suponiendo un radio de giro de 360 grados, el volumen de trabajo sería el de una esfera de radio  $2L$ , es decir  $(32/3)\pi L^3$ .

El controlador de un robot con un brazo de estructura angular debe realizar tareas más complejas debido a que, como ya se ha mencionado antes, las trayectorias se especifican normalmente en coordenadas cartesianas, por lo cual deben realizarse las transformaciones adecuadas.



**Figura 2.8:** Robot industrial PUMA en los laboratorios del Instituto Andaluz de Automática y Robótica, Escuela Superior de Ingenieros de la Universidad de Sevilla.

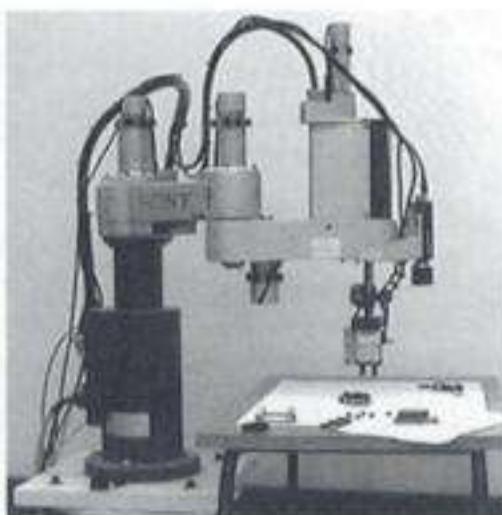
### 2.1.2.5 Configuración Scara

Esta configuración está especialmente diseñada para realizar tareas de montaje en un plano. Está constituida por dos articulaciones de rotación con respecto a dos ejes paralelos, y una de desplazamiento en sentido perpendicular al plano.

En la figura 2.9 se muestra un robot industrial Scara. El volumen de trabajo de este robot, suponiendo segmentos de longitud  $L$ , un radio de giro de 360 grados y un rango de desplazamiento de  $L$  es de  $4\pi L^3$ .

### 2.1.3 Orientación del efecto final

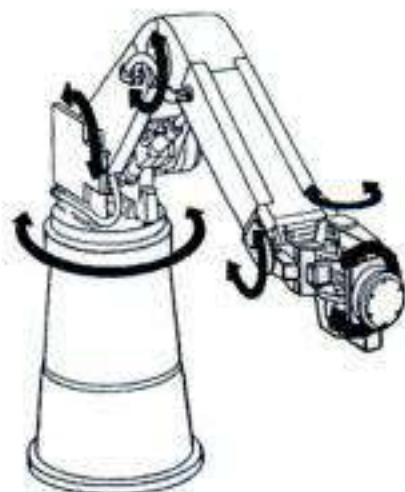
Como se ha mencionado anteriormente, el movimiento de un brazo robótico provisto de una muñeca con un efecto final es frecuente tratarlo en dos pasos. En primer lugar, se mueve el brazo para posicionar el extremo del último enlace y, posteriormente, se orienta la muñeca para que el efecto final tenga la orientación adecuada. No obstante, existen tareas que pueden requerir el movimiento simultáneo del brazo y la muñeca.



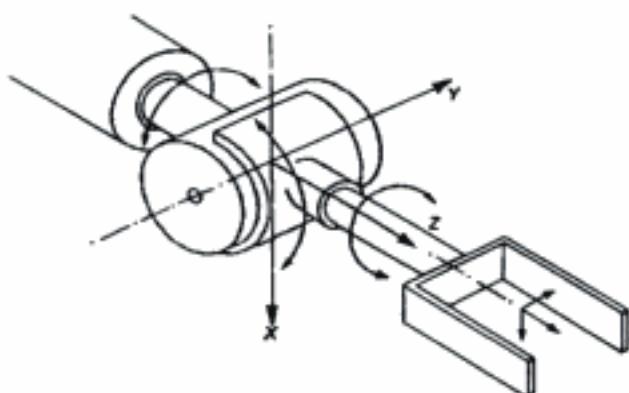
**Figura 2.9:** Robot Scara en los laboratorios del Instituto Andaluz de Automática y Robótica, Escuela Superior de Ingenieros de la Universidad de Sevilla.

En las tareas de montaje en un plano que se han mencionado, puede que no sea necesario ningún grado de libertad adicional ya que se trabaja siempre en dirección perpendicular al plano de montaje, tal como sucede en el montaje de componentes electrónicos.

Sin embargo, en otras tareas de manipulación, suele ser necesario que el efecto final tenga una determinada orientación en el espacio. Esto se consigue con la muñeca del manipulador. En la figura 2.10 se muestra un manipulador angular provisto de una muñeca que añade tres grados de libertad de rotación a la estructura. Estos tres ángulos permiten especificar la orientación del efecto final en el espacio. Obsérvese que, de esta forma, se llega a los seis grados de libertad (tres del brazo y tres de la muñeca) que se necesitan en un caso general para especificar una posición y una orientación en el espacio.



**Figura 2.10:** Manipulador con configuración angular y muñeca con tres grados de libertad.



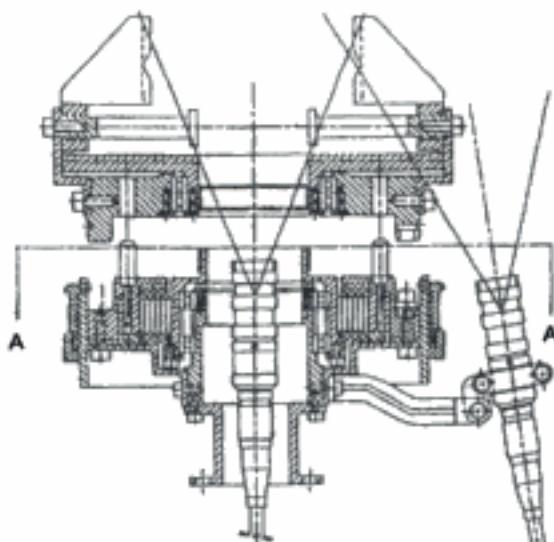
**Figura 2.11:** Muñeca en configuración degenerada (Ferraté y otros, 1986).

Al igual que sucede con la especificación de la posición mediante articulaciones consecutivas, pueden existir configuraciones degeneradas, como es el caso de la que se ilustra en la figura 2.11. Obsérvese cómo, en esta configuración, se pierde un grado de libertad.

#### 2.1.4 Efectores finales

Es el elemento que se coloca en el extremo del último enlace del manipulador y que suministra la capacidad de agarre del objeto que se pretende manipular, o la colocación de una herramienta apropiada para la tarea (pintura, soldadura, etc.).

Las características que hay que tener en cuenta para su diseño son: capacidad de carga, fuerza de agarre, geometría y dimensiones de los objetos que debe manejar, tolerancias, tipos de movimientos que puede realizar, alimentación (neumática, eléctrica, hidráulica), tiempo de actuación del mecanismo de agarre y características de la superficie de contacto.



**Figura 2.12:** Efecto final del Miniman. Se han montado dos cámaras y LED de iluminación.

Los efectores finales más simples son pinzas mecánicas, típicamente con dos dedos y accionamiento neumático todo/nada.

Se emplean también accionamientos eléctricos con control proporcional. En la figura 2.12 se muestra una pinza diseñada en el Proyecto Miniman, en el cual se han desarrollado tecnologías de manipulación robótica para aplicaciones espaciales y, en particular, para el mantenimiento de satélites (Gómez Elvira y Ollero, 1993).

Existen también dedos con material deformable para evitar que se produzca el cierreimiento de la pieza. Asimismo, es posible utilizar sensores táctiles en los dedos, empleando un bucle de control del esfuerzo de agarre. Se dispone también de pares de dedos con diferentes cavidades para piezas de distintos tamaños.

Se dispone de numerosas pinzas mecánicas para su empleo en el agarre de piezas pesadas o voluminosas, dedos de apertura amplia (figura 2.13), manos con sujetaciones interiores y exteriores, o manos dobles que pueden utilizarse para soltar una pieza y agarrar otra al mismo tiempo.

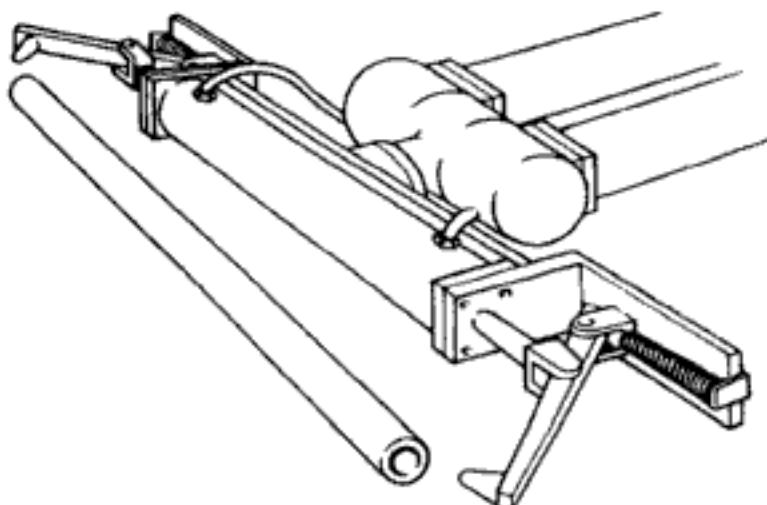


Figura 2.13: Pinza especial para agarre de piezas de elevada longitud.

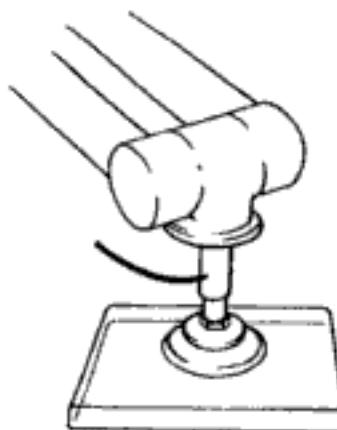


Figura 2.14: Pinza para sujeción por vacío.

Otra técnica clásica de agarre es la de sujeción por succión. Se emplean ventosas elásticas (figura 2.14) para manipular piezas rígidas y ventosas rígidas para manipular piezas elásticas. La sujeción por vacío se prefiere a la magnética para piezas de vidrio, frágiles (tales como tubos de rayos catódicos), cargas ligeras o moderadas.

## 2.2 NUEVAS ESTRUCTURAS PARA ROBOTS MANIPULADORES

---

En el apartado anterior se han presentado las estructuras convencionales en robótica y, en particular, en robótica industrial. Sin embargo, en los últimos años se han desarrollado numerosas tecnologías robóticas que han permitido abordar aplicaciones no convencionales. En este apartado se comentan brevemente algunos de estos desarrollos.

### 2.2.1 Robots redundantes

Para poder trabajar en lugares de difícil acceso es necesario añadir grados de libertad a la estructura. Por ello se utilizan manipuladores redundantes con más de seis grados de libertad.

El incremento del número de articulaciones ha llevado al desarrollo de robots manipuladores tipo serpiente, tal como el que se muestra en la figura 2.15 desarrollado por Miller (2000). Los manipuladores redundantes presentan complejos problemas de control (Hirose y Ma, 1991) y planificación autónoma.



Figura 2.15: Robot manipulador tipo serpiente.

Otra alternativa para tareas complejas de manipulación es el empleo de múltiples manipuladores. Obviamente, se presentan problemas de coordinación, evitación de colisiones y control dinámico coordinado de múltiples manipuladores actuando sobre un objeto (Luh y Zheng, 1987; Koivo y Bekey, 1988; Nakamura y otros, 1989; Yoshikawa y Zheng, 1991; Feliu y otros, 1999).



**Figura 2.16:** Robot manipulador flexible, construido en la ETS de Ingenieros Industriales de la Universidad de Castilla la Mancha.

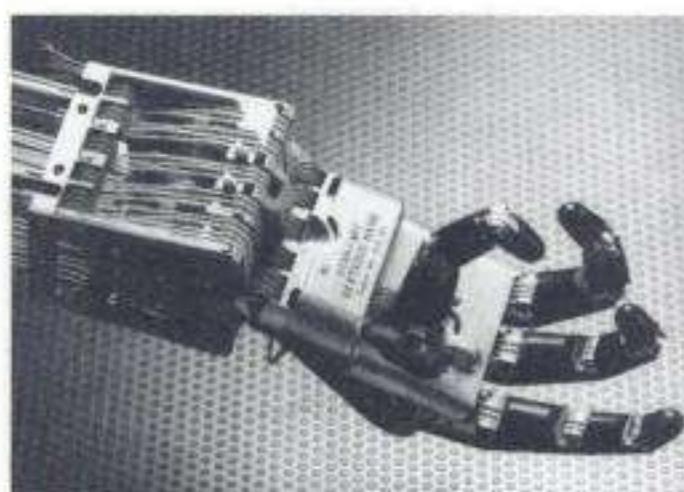
## 2.2.2 Robots flexibles

El empleo de robots flexibles (figura 2.16) resulta necesario cuando se requiere bajo peso y gran alcance. La situación es típica en manipuladores espaciales, pero también se presenta en otras aplicaciones de la robótica tales como la construcción.

En este punto conviene mencionar que el control de los robots flexibles es un problema mucho más complejo que el de los convencionales (ver, por ejemplo, Book, 1984; Das y Singh, 1989; Feliu, 1999).

## 2.2.3 Manos

Desde finales de los ochenta se han desarrollado órganos terminales muy evolucionados, tales como manos con múltiples dedos para manipulación diestra como la que se muestra en la figura 2.17 (Jacobsen y otros, 1988).



**Figura 2.17:** Mano robótica desarrollada en la Universidad de Utah.

## 2.3 ROBOTS MÓVILES

### 2.3.1 Vehículos con ruedas

Los vehículos con ruedas son la solución más simple y eficiente para conseguir la movilidad en terrenos suficientemente duros y libres de obstáculos, permitiendo conseguir velocidades relativamente altas.

Como limitación más significativa cabe mencionar el deslizamiento en la impulsión. Dependiendo de las características del terreno pueden presentarse también deslizamientos y vibraciones. La locomoción mediante ruedas es poco eficiente en terrenos blandos.

Por otra parte, excepto en configuraciones muy especiales, no es posible alterar internamente el margen de estabilidad para adaptarse a la configuración del terreno, lo que limita de forma importante los caminos aceptables del soporte.

Los robots móviles emplean diferentes tipos de locomoción mediante ruedas que les confieren características y propiedades diferentes respecto a la eficiencia energética, dimensiones, cargas útiles y maniobrabilidad. La mayor maniobrabilidad se consigue en vehículos omnidireccionales. Un vehículo omnidireccional en el plano es capaz de trasladarse simultáneamente en cada eje del sistema de coordenadas, y rotar según el eje perpendicular.

A continuación se comentan brevemente las características más significativas de los sistemas de locomoción más comunes en robots móviles.

#### 2.3.1.1 Ackerman

Es el utilizado en vehículos de cuatro ruedas convencionales. De hecho, los vehículos robóticos para exteriores resultan normalmente de la modificación de vehículos convencionales tales como automóviles o incluso vehículos más pesados. Este sistema de locomoción se ilustra en la figura 2.18. La rueda delantera interior gira un ángulo ligeramente superior a la exterior ( $\theta_1 > \theta_0$ ) para eliminar el deslizamiento. Las prolongaciones de los ejes de las dos ruedas delanteras intersectan en un punto sobre la prolongación del eje de las ruedas traseras. El lugar de los puntos trazados sobre el suelo por los centros de los neumáticos son circunferencias concéntricas con centro el eje de rotación  $P_1$  en la figura. Si no se tienen en cuenta las fuerzas centrífugas, los vectores de velocidad instantánea son tangentes a estas curvas.

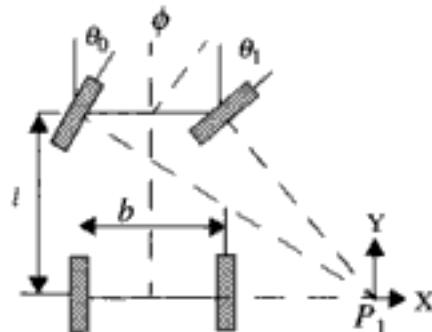


Figura 2.18: Sistema Ackerman.



Figura 2.19: *NavLab 1.*

En la figura 2.19 se muestra el NavLab 1, un vehículo autónomo que ha venido empleándose en el Robotics Institute de Carnegie Mellon University desde 1986 para experimentos de navegación autónoma en exteriores. En la figura 2.20 puede observarse el NavLab 2, vehículo todoterreno que se emplea desde 1991.

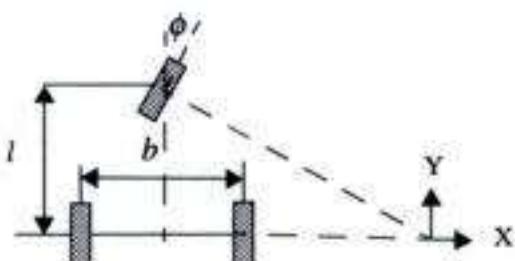
El mayor problema de la locomoción Ackerman es la limitación en la maniobrabilidad.



Figura 2.20: *NavLab 2.*

### 2.3.1.2 Triciclo clásico

Este sistema de locomoción se ilustra en la figura 2.21. La rueda delantera sirve tanto para la tracción como para el direccionamiento. El eje trasero, con dos ruedas laterales, es pasivo y sus ruedas se mueven libremente. La maniobrabilidad es mayor que en la configuración anterior pero puede presentar problemas de estabilidad en terrenos difíciles. El centro de gravedad tiende a desplazarse cuando el vehículo se desplaza por una pendiente, causando la pérdida de tracción.



**Figura 2.21:** Locomoción de triciclo clásico.

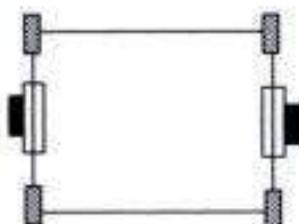
Este sistema de locomoción es la del vehículo robótico Romeo 3R (Ollero y otros, 1999) desarrollado en los laboratorios de la Escuela de Ingenieros de la Universidad de Sevilla (figura 2.22). Debido a su simplicidad, es bastante frecuente en vehículos robóticos para interiores y exteriores pavimentados.

### 2.3.1.3 Direccionamiento diferencial

El direccionamiento viene dado por la diferencia de velocidades de las ruedas laterales. La tracción se consigue también con estas mismas ruedas. Adicionalmente, existen una o más ruedas para soporte. En la figura 2.23 se ilustra el sistema de locomoción de la plataforma Labmate . Esta configuración es la más frecuente en robots para interiores.



**Figura 2.22:** Romeo 3R desarrollado en los laboratorios de la Escuela Superior de Ingenieros de la Universidad de Sevilla.



**Figura 2.23:** Locomoción con direccionamiento diferencial en dos ruedas laterales. El sistema emplea cuatro ruedas de castor no actuadas para soporte de la plataforma.

### 2.3.1.4 Skid Steer

Se disponen varias ruedas en cada lado del vehículo que actúan de forma simultánea. El movimiento es el resultado de combinar las velocidades de las ruedas de la izquierda con las de la derecha. En la figura 2.24 se muestra el "Terregator", un vehículo robótico desarrollado en el Robotics Institute de la Carnegie Mellon University para aplicaciones en exteriores tales como la minería. Este robot se ha aplicado también (Jamshidi y Eicker, 1993) para inspección y obtención de mapas de tuberías enterradas empleando para ello un sistema radar ("Ground Penetrating Radar").



**Figura 2.24:** Robot Terregator con locomoción tipo "skid steer".

### 2.3.1.5 Pistas de deslizamiento

Son vehículos tipo oruga en los que tanto la impulsión como el direccionamiento se consiguen mediante pistas de deslizamiento. Pueden considerarse funcionalmente análogas al skid steer. De forma más precisa, las pistas actúan de forma análoga a ruedas



**Figura 2.25:** Robot AURIGA con locomoción mediante pistas de deslizamiento, desarrollado en los laboratorios del Instituto Andaluz de Automática y Robótica, ETS de Ingenieros Industriales, Universidad de Málaga.

das de gran diámetro. La locomoción mediante pistas de deslizamiento es útil en navegación "campo a través" o en terrenos irregulares, en los cuales presenta un buen rendimiento. En este caso, la impulsión está menos limitada por el deslizamiento y la resistencia al desgaste es mayor. En la figura 2.25 se muestra un robot con locomoción mediante pistas de deslizamiento.

#### 2.3.1.6 Síncronas

Consiste en la actuación simultánea de todas las ruedas, que giran de forma síncrona. La transmisión se consigue mediante coronas de engranajes (synchro drive) o con correas concéntricas. En la figura 2.26 se muestra el "Locomotion Emulator", vehículo robótico también desarrollado en el Robotics Institute de la Carnegie Mellon University.



Figura 2.26: *Locomotion Emulator* desarrollado en la Carnegie Mellon University.

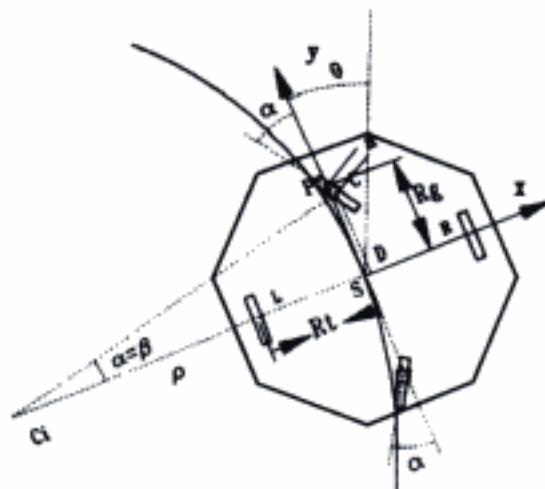


Figura 2.27: *Robot móvil RAMI* desarrollado en la Universidad de Málaga.

### 2.3.1.7 Otras configuraciones

Entre éstas puede señalarse la empleada por el robot RAM1 (Ollero y otros, 1995) que se muestra en la figura 2.27.

El sistema de locomoción de RAM1 se ilustra en la figura 2.28. Consiste en cuatro ruedas dispuestas en las diagonales de un rombo, con la diagonal principal según el eje longitudinal del vehículo y dos ruedas laterales actuadas de forma independiente. Al actuar en el eje longitudinal las ruedas delantera y trasera giran en sentido contrario. El sistema combina este direccionamiento con el diferencial, que se consigue actuando de forma independiente las ruedas laterales que impulsan el vehículo.



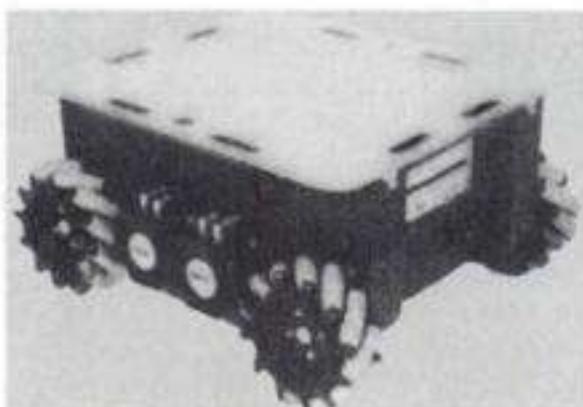
**Figura 2.28:** Cinemática y sistema de locomoción del RAM1.

Con este sistema es posible tener un radio de giro nulo y se mantienen las buenas propiedades del guiado diferencial en el seguimiento de caminos. El sistema se ha aplicado también en el robot AURORA (figura 2.29) para la realización de trabajos en invernaderos (Ollero y otros, 1995; Madow y otros, 1996).



**Figura 2.29:** Robot móvil AURORA.

Otra configuración consiste en el empleo de ruedas especiales, tales como las denominadas "ruedas suecas", que permiten conseguir el movimiento omnidireccional de un vehículo (figura 2.30) con cuatro ruedas de este tipo adecuadamente controladas (Muir y Neuman, 1990).

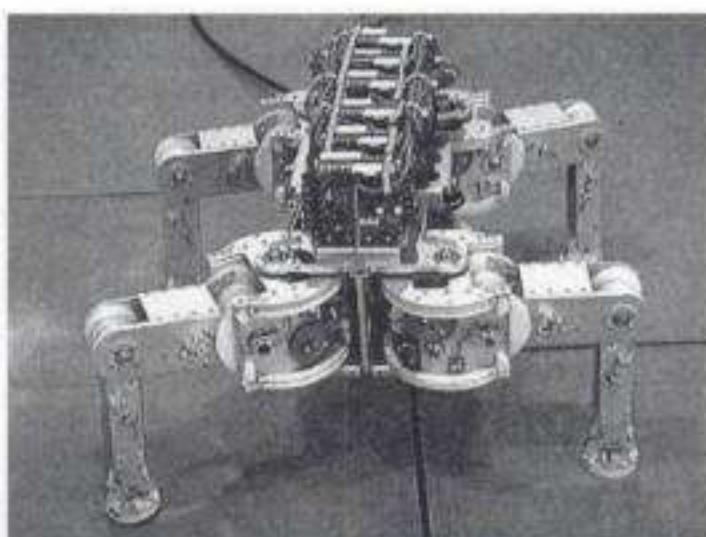


**Figura 2.30:** Robot *Uranus* (*Carnegie Mellon University*) con ruedas especiales para locomoción omnidireccional.

### 2.3.2 Locomoción mediante patas

Permiten aislar el cuerpo del terreno empleando únicamente puntos discretos de soporte. Es posible adaptar el polígono de soporte para mantener la estabilidad y pasar sobre obstáculos. Por consiguiente, tiene mejores propiedades que las ruedas para atravesar terrenos difíciles llenos de obstáculos.

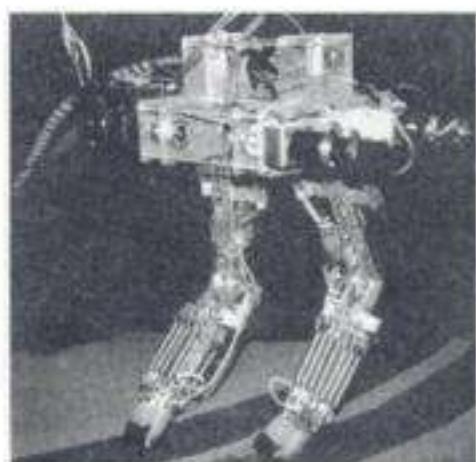
Asimismo, mediante patas, es posible conseguir la omnidireccionalidad y el deslizamiento en la locomoción es mucho menor.



**Figura 2.31:** Robot cuadrúpedo desarrollado en el *Tokyo Institute of Technology*.

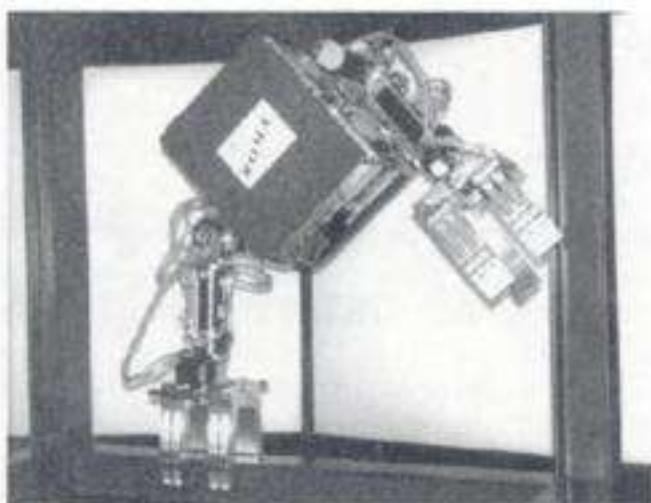
En los robots con patas la complejidad de los mecanismos necesarios es mayor, así como el consumo de energía en la locomoción. En principio, los problemas de planificación y control son más complejos que en los vehículos con ruedas.

La configuración más común es la de seis patas. Existen también de ocho patas, tales como el construido en el Robotics Institute de la Carnegie Mellon University que ha sido empleado para la exploración de un volcán en Alaska (1994). Existen también robots cuadrúpedos (figura 2.31) y bípedos (figura 2.32) que mantienen el equilibrio de forma dinámica.



**Figura 2.32:** Robot bípedo desarrollado en el Massachusetts Institute of Technology (EE.UU.).

En este punto conviene mencionar también a los robots trepadores (figura 2.33), que son de interés para realizar operaciones tales como inspecciones (Abderrahim y otros, 1999) y reparaciones en paredes verticales. Estos robots se sostienen mediante garras o bien empleando dispositivos de succión o magnéticos en las patas.



**Figura 2.33:** Robot trepador desarrollado en la Universidad Carlos III de Madrid.

Existen también sistemas mixtos de locomoción con ruedas para terrenos lisos y patas para salvar obstáculos, subir escaleras, trepar, etc.

### 2.3.3 Configuraciones articuladas

Las configuraciones articuladas son de interés para terrenos difíciles a los que debe adaptarse el cuerpo del robot. La solución más simple consiste en articular dos o más módulos con locomoción mediante ruedas. Las configuraciones articuladas con gran cantidad de eslabones (figura 2.34) son apropiadas para caminos estrechos. La seguridad de funcionamiento puede ser mayor debido a la redundancia de su estructura, ofreciendo la posibilidad de intercambiar segmentos. El transporte de la estructura se ve también facilitado. Como casos particulares cabe mencionar las estructuras compuestas por segmentos con patas o con ruedas. Las configuraciones articuladas son aún recientes y su aplicación, fuera de los laboratorios de robótica, requiere todavía progresos significativos para dotarlas de suficiente autonomía y resolver los complejos problemas de control que se presentan. En Moraleda y otros (1999) se presenta un nuevo diseño de un robot que emplea tracción y direccionamiento hidráulicos, mediante eyección de chorros, para desplazarse por el interior de tuberías de pequeño diámetro.



Figura 2.34: *MAKRO*, robot móvil articulado de GMD (Alemania).

### 2.3.4 Robots submarinos y aéreos

El interés de las aplicaciones tales como la inspección, recogida de datos o mantenimiento de instalaciones en entornos naturales a los que el acceso del hombre resulta muy difícil, o incluso imposible, ha motivado también el desarrollo de vehículos robóticos submarinos o aéreos.

Estos vehículos son normalmente el resultado de la evolución de vehículos completamente teleoperados o pilotados por el hombre. En la figura 2.35 se muestra un robot submarino desarrollado en la Universidad Politécnica de Cataluña (Amat y otros, 1999). Recientemente se han realizado también helicópteros robóticos experimentales (Amidi, 1994; Sugeno y otros, 1993) dotando de autonomía a helicópteros teleoperados.



**Figura 2.35:** Robot submarino desarrollado en la Universidad Politécnica de Cataluña.

## 2.4 CONCLUSIONES

Este capítulo se ha dedicado a estudiar aspectos morfológicos de los robots. Se han considerado tanto los robots manipuladores como los móviles. En el primer caso, se han presentado, en primer lugar, las estructuras más comunes de los robots industriales y, a continuación, las nuevas estructuras que actualmente son objeto de actividades de investigación y desarrollo.

Por lo que respecta a los robots móviles, se han presentado una gran diversidad de plataformas robóticas, desde robots para interiores hasta vehículos robóticos para exploración en terrenos naturales o en fondos marinos.

En el diseño de un nuevo sistema robótico es necesario tener en cuenta la movilidad necesaria para realizar la tarea, la eficiencia energética, la relación peso/carga y las dimensiones.

Asimismo, resulta fundamental considerar de forma conjunta la morfología del robot con la mayor o menor complejidad de las funciones de percepción, planificación y control necesarias en la tarea que se pretende desarrollar.

## 2.5 REFERENCIAS

- Abderrahim M., C. Balaguer, A. Giménez, J.M. Pastor y V.M. Padrón. "ROMA: A climbing robot for inspection operations". IEEE International Conference on Robotics and Automation. ICRA'99. Detroit, Michigan.
- Amat, J., A. Monferrer, J. Batlle, y X. Cuf, 1999. "Garbi: A low-cost underwater vehicle". Microprocessors and Microsystems. Elsevier Sciences. Vol. 23, núm. 2-Special Issue. Septiembre 1999, pp. 61-67.
- Amidi O., 1994. "An autonomous vision guided helicopter". Doctoral Dissertation. Robotics Institute. Carnegie Mellon University.
- Asociación Española de Robótica, 1987. "Robots en la industria". Vídeo editado en colaboración con la Universidad Politécnica de Cataluña.
- Book W.J., 1984. "Recursive Lagrangian dynamics of flexible manipulators arms". Int. Journal of Robotic Research. Vol. 3, núm. 3, pp. 87-101.

- Das A. y S.N. Singh, 1989. "Nonlinear adaptive control and linear stabilization of an elastic arm". *Int. Journal of Robotics and Automation*. Vol. 5, núm. 3, pp. 115-123.
- Feliu J.J., V. Feliú y C. Cerrada, 1999. "Load Adaptive Control of Single-Link Flexible Arms Based on a New Modeling Technique". *IEEE Transactions on Robotics and Automation*. Vol. 15, núm. 5, octubre 1999, pp. 793-804.
- Ferraté G. y otros, 1986. "Robótica Industrial", Marcombo.
- Gómez-Elvira J. y A. Ollero, 1993 "Proyecto MINIMAN. Demostrador de la tecnología de pequeños manipuladores espaciales". *Automatización y Robótica Integrada. Comunicaciones del Tercer Congreso de la Asociación Española de Robótica*, pp. 305-310.
- Hirose S. y S. Ma, 1991. "Moray drive for multijoint manipulators". Proc. of the fifth International Conference on Advanced Robotics. Pisa (Italia), pp. 521-526.
- Jacobsen S.C., I.D. McCammon, K.B. Biggers y R. Phillips, 1988. "Design of tactil sensing system for dexterous manipulators". *IEEE Control System Magazine*. Vol. 8, núm. 1, pp. 3-13.
- Jamshidi M. y P.J. Eicker (editores), 1993. "Robotics and Remote Systems for Hazardous Environments". Prentice Hall Series on Environmental and Intelligent Manufacturing Systems. Prentice Hall.
- Koivo A.J. y G.A. Bekey, 1988. "Report of the Workshop on coordinated multiple robot manipulators: planning, control and applications". *IEEE Trans. of Robotic and Automation*. Vol. 4, núm. 1, pp. 91-93.
- Luh J.Y.S. e Y.Z. Zheng, 1987. "Constrained relations between two coordinated industrial robots for motion control". *Int. Journal of Robotic Research*. Vol. 6, núm. 3, pp. 60-70.
- Mandow, A. y otros, 1996. "The Autonomous Mobile Robot Aurora for Greenhouse Operation". *IEEE Robotics and Automation Magazine*. Vol. 3, núm. 4, pp. 18-28.
- Miller, G.S.P., 2000. "Snake robots for search and rescue". En "Neurotechnology for biomimetic robots". J. Ayers, J. Davis, y A. Rudolph. Será publicado por MIT Press.
- Moraleda J., A. Ollero y M. Orte, 1999. "A robotic system for internal inspection of water pipelines". *IEEE Robotics and Automation Magazine*. Vol. 6, núm. 3, pp. 30-41.
- Muir P.F. y C.P. Neuman, 1990. "Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot". *Autonomous Robot Vehicles*. Editado por I.J. Cox y G.T. Wilfona. Springer-Verlag.
- Nakamura Y., K. Nagai y T. Yoshikawa, 1989. "Dynamics and stability in coordination of multiple robotics mechanism". *Int. Journal of Robotic Research*. Vol. 8, núm. 2, pp. 44-61.
- Ollero A., Arrue B.C, Ferruz J., Heredia G., Cuesta F., L-Pichaco F. y Nogales C., 1999. "Control and perception components for automation vehicle guidance. Application to the Romeo vehicle". *Control Engineering Practice*. Vol. 7, núm. 10. Octubre, pp. 1291-1299.
- Ollero A. y otros, 1995. "The autonomous robots for spraying Aurora". Proc. of the IARP Conference on Robotics in the Agriculture and the Food Industry, pp. 193-198, Toulouse, France.
- Ollero A. y E.F. Camacho, 1993. "Intelligent Components and Instruments for Control Applications". Pergamon Press.
- Sugeno M., M.F. Griffin y A. Bastian, 1993. "A fuzzy hierarchical control of an unmanned helicopter". Proc. of the IEEE 3rd. Conference of Fuzzy Systems, Korea.
- Yoshikawa T. y X. Zheng, 1991. "Coordinated dynamic control for multiple robot manipulators handling an object". Proc. of the fifth International Conference on Advanced Robotics. Pisa (Italia), pp. 579-584.

# Representación de la posición y orientación

---

En robótica es necesario poder describir de forma conveniente las posiciones y orientaciones de los objetos en el espacio. Este capítulo está dedicado a introducir conceptos básicos que se emplearán en los modelos cinemáticos y dinámicos de los capítulos posteriores. Se comienza por estudiar la posición y orientación en el plano introduciendo las operaciones elementales de rotación y traslación, y considerando los cambios de sistemas de referencia. A continuación se trata el problema en tres dimensiones y se introduce el concepto de transformación homogénea. En los siguientes apartados se consideran las transformaciones compuestas y la inversa de una transformación, respectivamente. Finalmente, se estudian formas alternativas para representar la orientación de un cuerpo en el espacio. En este capítulo se emplea por primera vez la herramienta MATLAB que se presenta en el apéndice y que se utilizará en los capítulos posteriores para el tratamiento de las transformaciones y modelos de robots.

## **3.1 POSICIÓN Y ORIENTACIÓN EN EL PLANO**

---

En este capítulo y en los dos siguientes se empleará la notación de Craig (1989). En Fu y otros (1988), Barrientos y otros (1997) y Paul (1981) pueden encontrarse otras notaciones alternativas que son también de interés para representar los modelos empleados en robótica.

En Corke (1996) se presenta una herramienta MATLAB, denominada "Robotics Toolbox" en la cual se emplea la notación de Paul (1981). Esta herramienta es de notable interés, aunque presenta diversos problemas para su utilización en este texto. En el apéndice se presenta una nueva herramienta, que es una versión mejorada del "Robotics Toolbox", y que permite además ser aplicada directamente con la notación de Craig (1989) empleada en este texto. Esta nueva herramienta se emplea en numerosos ejemplos del libro. Su utilización permitirá al lector reproducir con facilidad los resultados obtenidos en dichos ejemplos.

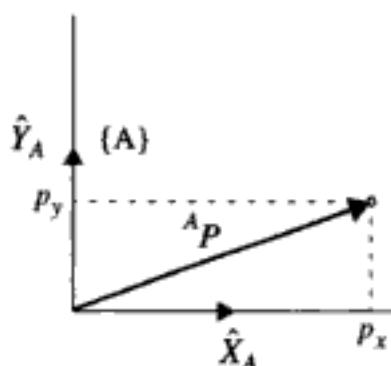


Figura 3.1: Vector de coordenadas en el plano.

En este apartado se considera la localización de objetos en el plano. El problema es frecuente en el estudio de robots móviles que navegan por terrenos planos. En este caso se necesitan dos coordenadas y un ángulo de orientación.

Supóngase un sistema de coordenadas de referencia fijo al que se designará como sistema  $\{A\}$ . La posición con respecto a este sistema se representará mediante un vector de posición  $^A P$ , tal como se ilustra en la figura 3.1, cuyas componentes son las coordenadas de este punto:

$$^A P = \begin{bmatrix} p_x \\ p_y \end{bmatrix} \quad (3.1)$$

Este sistema tiene como vectores unitarios  $\hat{X}_A$ ,  $\hat{Y}_A$ .

Otra posible forma de expresar las coordenadas de un punto con respecto a un sistema de referencia es mediante el empleo de coordenadas polares, tal como se ilustra en la figura 3.2. En este caso las coordenadas son la distancia al origen  $r$  y el ángulo  $\theta$  que forma el vector con el eje  $\hat{X}_A$ .

Considérese también otro sistema de coordenadas con vectores unitarios  $\hat{X}_B$ ,  $\hat{Y}_B$  tal como se muestra en la figura 3.3. La dirección del vector  $\hat{Y}_B$  forma un ángulo  $\theta$  con el vector  $\hat{Y}_A$ . Obsérvese que si, por ejemplo, se trata de localizar a un robot móvil que se desplaza en el plano, este segundo sistema puede ser solidario al robot con  $\hat{Y}_B$  en la orientación del robot.

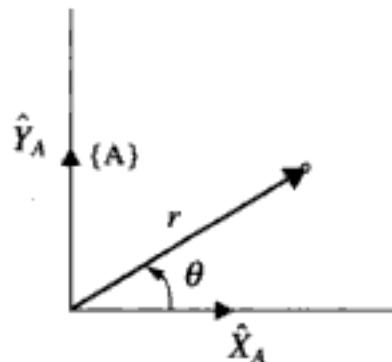


Figura 3.2: Coordenadas polares.

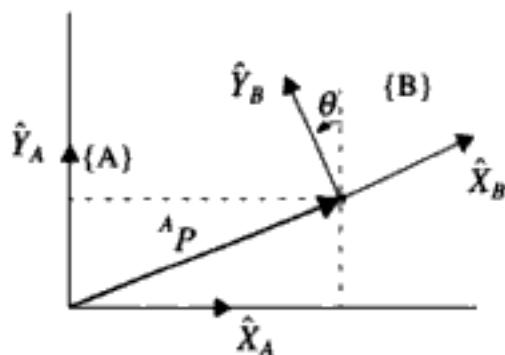


Figura 3.3: Localización de un objeto en posición y orientación.

Si se expresan los vectores unitarios del sistema {B} en el {A}, se escribirá  ${}^A\hat{X}_B$ ,  ${}^A\hat{Y}_B$ . Estos dos vectores se disponen según las columnas de una matriz,

$${}^A_R = \begin{bmatrix} {}^A\hat{X}_B & {}^A\hat{Y}_B \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \quad (3.2)$$

a la que se conoce como *matriz de rotación*. Estas matrices juegan un papel importante en los modelos empleados en robótica.

Obsérvese que si el ángulo de orientación es  $\theta$ , de la figura 3.3 se deduce:

$${}^A_R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (3.3)$$

Teniendo en cuenta que los vectores columna son ortonormales, puede escribirse también

$${}^A_R = {}^B_R^{-1} = {}^B_R^T \quad (3.4)$$

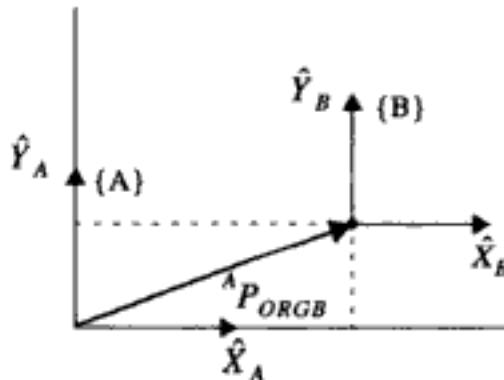


Figura 3.4: Traslación del sistema de coordenadas.

Supóngase que el sistema {B} tiene sus vectores de dirección coincidentes con los del {A}, como se ilustra en la figura 3.4. Este problema se presentaría, por ejemplo, cuando el robot se desplazara sin cambiar de orientación. El origen del sistema {B} se localizará con respecto al {A} mediante el vector

$${}^A P_{ORG} = \begin{bmatrix} {}^A p_x \\ {}^A p_y \end{bmatrix}^T \quad (3.5)$$

Por consiguiente, las coordenadas de un punto cualquiera del plano en los dos sistemas están relacionadas mediante

$${}^A p_x = {}^B p_x + {}^A p_{xORG} \quad (3.6)$$

$${}^A p_y = {}^B p_y + {}^A p_{yORG} \quad (3.7)$$

expresiones en las cuales las componentes de los vectores pueden sumarse por estar estos vectores en la misma dirección.

Las ecuaciones (3.6) y (3.7) definen la transformación de traslación del sistema de coordenadas.

Supóngase ahora que el origen del sistema {B} coincide con el del {A} pero la orientación es diferente, como se ilustra en la figura 3.5. Esta circunstancia se presenta al estudiar movimientos de rotación. Asimismo, este sería el caso cuando el sistema de referencia {A} se desplazara con el robot pero mantuviera su orientación constante, mientras que el sistema {B} además de desplazarse con el robot girara también con él.

En este caso, las coordenadas expresadas en el sistema {A} en función de las del {B} vienen dadas por:

$${}^A p_x = {}^B \hat{X}_A \cdot {}^B P \quad (3.8)$$

$${}^A p_y = {}^B \hat{Y}_A \cdot {}^B P \quad (3.9)$$

expresión en la cual el punto indica el producto escalar. Nótese como (3.8) y (3.9) indican las proyecciones del vector  ${}^A P$  que define el punto sobre los ejes unitarios del sistema que se toma como referencia. Estas expresiones pueden escribirse de forma compacta como:

$${}^A P = {}^B R {}^B P = \begin{bmatrix} {}^B \hat{X}_A^T \\ {}^B \hat{Y}_A^T \end{bmatrix} {}^B P \quad (3.10)$$

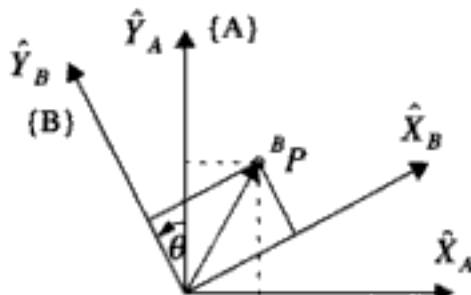


Figura 3.5: Rotación del sistema de coordenadas.

En un caso general, en el cambio de sistemas de referencia existirán tanto traslaciones como rotaciones.

En todas las expresiones anteriores se ha empleado la notación de Craig (1989), que es la que se continuará usando en el texto.

### Ejemplo 3.1

Un robot móvil provisto de sensores de proximetría detecta un obstáculo a una distancia  $d$  en la dirección de marcha. Se sabe que el ángulo de orientación del robot en el instante de la medida es  $\theta$ . Se trata de determinar las coordenadas absolutas del obstáculo con respecto a un sistema de ejes de referencia solidario al vehículo pero con la misma orientación que el sistema absoluto.

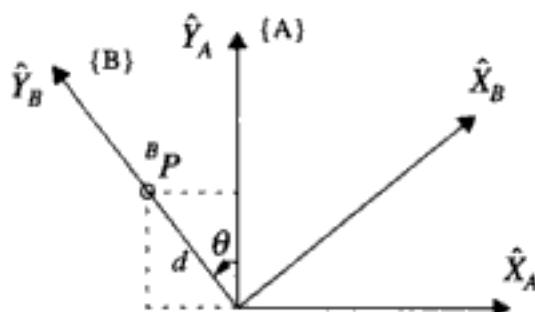


Figura 3.6: Detección de un obstáculo desde un robot móvil.

De acuerdo con la figura 3.6 se tendrá:

$${}^B P = [0 \ d]^T; \quad {}^B \hat{X}_A = [\cos \theta \ -\sin \theta]^T; \quad {}^B \hat{Y}_A = [\sin \theta \ \cos \theta]^T$$

Por consiguiente,

$${}^A p_{x_{obs}} = {}^B \hat{X}_A \cdot {}^B P = -d \sin \theta; \quad {}^A p_{y_{obs}} = {}^B \hat{Y}_A \cdot {}^B P = d \cos \theta$$

Considérese ahora la situación que se ilustra en la figura 3.7. Se supone que el sistema {A} está fijo y, en el instante en que se toma la medida, el robot está situado con

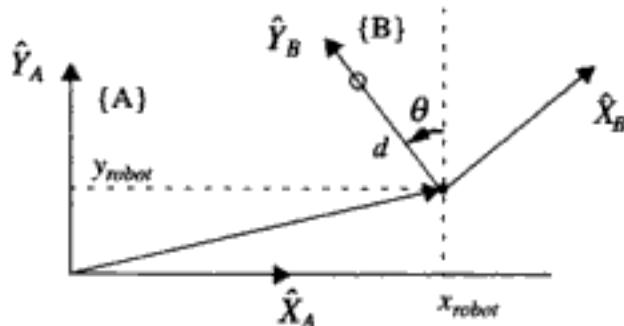


Figura 3.7: Cambio de sistema de referencia en navegación de robots móviles.

respecto a  $\{A\}$  en unas coordenadas  $(x_{robot}, y_{robot})$ . El obstáculo se encuentra con respecto al sistema  $\{A\}$  en:

$${}^A p_{xobs} = x_{robot} - d \operatorname{sen} \theta; \quad {}^A p_{yobs} = y_{robot} + d \cos \theta$$

### Ejemplo 3.2

Considérese ahora un manipulador plano con una articulación de traslación y otra de rotación, como el que se muestra en la figura 3.8.

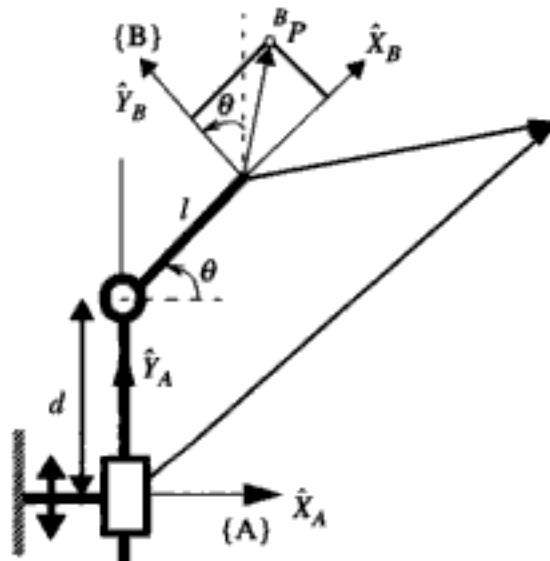


Figura 3.8: Cambio de sistemas de referencia en un manipulador plano.

Sean  $d$  y  $\theta$  las variables de la primera y segunda articulación. Las coordenadas respecto al sistema  $\{A\}$  están relacionadas con las coordenadas respecto al sistema  $\{B\}$  según:

$${}^A p_x = ({}^B p_x + l) \cos \theta - {}^B p_y \operatorname{sen} \theta;$$

$${}^A p_y = {}^B p_y \cos \theta + {}^B p_x \operatorname{sen} \theta + d + l \operatorname{sen} \theta$$

## 3.2 POSICIÓN Y ORIENTACIÓN EN EL ESPACIO

En robótica resulta imprescindible poder representar posiciones y orientaciones en el espacio. En efecto, es evidente que los movimientos de los manipuladores presentados en el capítulo 2 requieren poder representar de forma conveniente las posiciones y orientaciones en el espacio. Asimismo, en robótica móvil, cada vez tiene mayor interés considerar los modelos en tres dimensiones. Este interés es evidente en vehículos autónomos aéreos o submarinos, pero también se emplean estos modelos para estudiar la navegación en terrenos no planos de robots con ruedas o con patas.

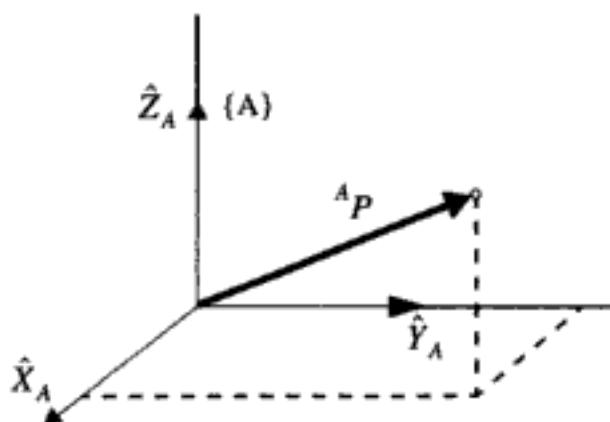


Figura 3.9: Posición en el espacio.

Sea  ${}^A P$  el vector de posición en un sistema  $\{A\}$ , tal como se ilustra en la figura 3.9, para un vehículo autónomo. En este caso se escribe

$${}^A P = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (3.11)$$

siendo  $p_x$ ,  $p_y$  y  $p_z$  las coordenadas.

La posición de un punto en el espacio también puede representarse mediante coordenadas cilíndricas tal como se muestra en la figura 3.10. En este caso las coordenadas son la distancia  $r$ , el ángulo  $\theta$  y la distancia  $z$  entre el punto y su proyección sobre el plano  $\hat{X}_A - \hat{Y}_A$ . Las dos primeras tienen el mismo significado que las coordenadas polares de la figura 3.2.

Por último, en la figura 3.11 se presentan las coordenadas esféricas que permiten representar la posición de un punto mediante la distancia  $r$  y los dos ángulos  $\phi$  y  $\theta$ .

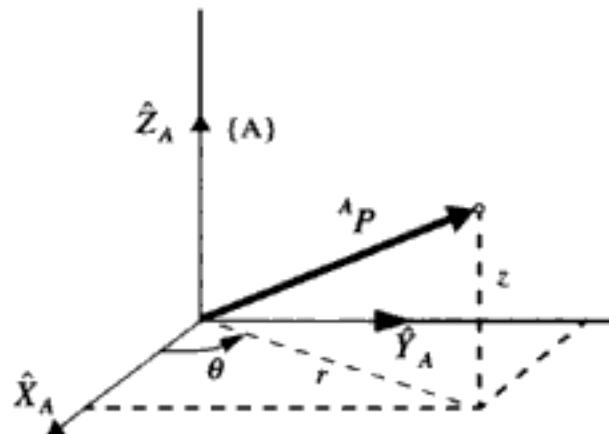
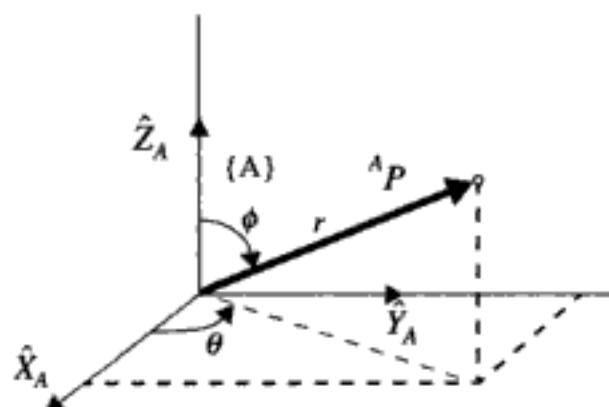


Figura 3.10: Representación de la posición empleando coordenadas cilíndricas.



**Figura 3.11:** Representación de la posición empleando coordenadas esféricas.

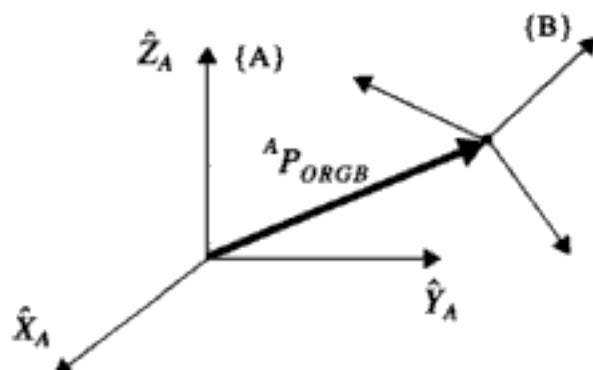
En esta sección y en las tres siguientes se siguen los desarrollos de Craig (1986).

La *orientación* de un cuerpo se describe mediante el sistema de coordenadas {B} solidario al cuerpo. Si se desea representar la orientación con respecto a un sistema de referencia {A}, el problema consiste simplemente en expresar {B} con respecto a {A}.

Sean  $\hat{X}_B, \hat{Y}_B, \hat{Z}_B$  los vectores unitarios en la dirección de los ejes de un sistema de coordenadas {B}. Si se desea expresar que estos vectores se representan en el sistema {A}, se escribirá  ${}^A\hat{X}_B, {}^A\hat{Y}_B, {}^A\hat{Z}_B$ . En este caso la *matriz de rotación* es:

$$\{{}^B_R\} = \begin{bmatrix} {}^A\hat{X}_B & {}^A\hat{Y}_B & {}^A\hat{Z}_B \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (3.12)$$

Como resumen de lo anterior cabe añadir que la localización (posición y orientación) en el espacio se determina mediante cuatro vectores de tres elementos, formando tres de ellos la matriz de rotación  ${}^B_R$  del sistema {B} que se mueve solidario al sólido rígido con respecto a un sistema de referencia {A}, y el cuarto la posición del origen  ${}^A P_{ORG_B}$  del sistema {B} solidario al cuerpo expresada en el sistema de referencia {A}, tal como se ilustra en la figura 3.12.



**Figura 3.12:** Posición y orientación en el espacio.

Obsérvese que la expresión de la posición y orientación de un sólido rígido puede considerarse como la descripción de un sistema de coordenadas asociado al sólido con respecto a otro de referencia. Un cambio en la posición manteniendo constante la orientación puede representarse mediante la matriz de rotación unidad y un vector con los cambios en las coordenadas del punto.

En la mayor parte de los problemas que surgen en robótica es necesario realizar cambios entre sistemas de referencia en el espacio. Así, en robots manipuladores convencionales, los modelos del manipulador involucran cambios entre sistemas asociados a las diferentes articulaciones de la cadena cinemática para describir la posición y orientación del extremo del manipulador con relación a la base.

Tal como se ha visto en el apartado 3.1, estas transformaciones consisten esencialmente en traslaciones y rotaciones. Suponiendo que  $\{A\}$  y  $\{B\}$  tienen la misma orientación, la traslación puede expresarse mediante:

$${}^A P = {}^B P + {}^A P_{ORG} \quad (3.13)$$

como se ilustra en la figura 3.13. Obsérvese que los dos vectores de la derecha de la expresión pueden sumarse debido a que ambos sistemas de referencia tienen la misma orientación.

La rotación puede expresarse mediante

$${}^A R = {}^B R^{-1} = {}^B R^T \quad (3.14)$$

igualdades que se justifican teniendo en cuenta que las columnas son vectores ortonormales. En términos de estos vectores puede escribirse

$${}^A R = \begin{bmatrix} {}^A \hat{X}_B & {}^A \hat{Y}_B & {}^A \hat{Z}_B \end{bmatrix} = \begin{bmatrix} {}^B \hat{X}_A^T \\ {}^B \hat{Y}_A^T \\ {}^B \hat{Z}_A^T \end{bmatrix} \quad (3.15)$$

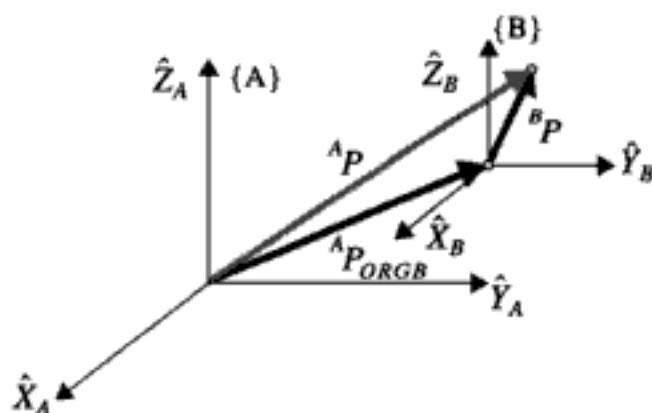


Figura 3.13: Cambio de sistemas de referencia en el espacio.

Suponiendo que los orígenes de {A} y {B} son coincidentes, puede escribirse:

$${}^A P = {}^B R {}^B P \quad (3.16)$$

Si se recuerda que los componentes de  ${}^A P$  pueden considerarse como las proyecciones del vector que define el punto en los vectores unitarios que definen el sistema de referencia, se tiene:

$$\begin{aligned} {}^A p_x &= {}^B \hat{X}_A \cdot {}^B P \\ {}^A p_y &= {}^B \hat{Y}_A \cdot {}^B P \\ {}^A p_z &= {}^B \hat{Z}_A \cdot {}^B P \end{aligned} \quad (3.17)$$

expresiones en las cuales el punto indica el producto escalar.

En un caso general se involucran tanto rotación como traslación. Si  ${}^A \Pi$  y  ${}^B \Pi$  son respectivamente los vectores  ${}^A P$  y  ${}^B P$  extendidos con una cuarta componente de valor 1:

$${}^A \Pi = \begin{bmatrix} {}^A P \\ 1 \end{bmatrix}; {}^B \Pi = \begin{bmatrix} {}^B P \\ 1 \end{bmatrix} \quad (3.18)$$

puede escribirse:

$${}^A \Pi = {}_B T {}^B \Pi \quad (3.19)$$

siendo

$${}_B T = \begin{bmatrix} {}_B R & {}^A P_{ORG} \\ 0 & 1 \end{bmatrix} \quad (3.20)$$

una matriz  $4 \times 4$  a la que se denomina *matriz de transformación homogénea*. Esta matriz es útil para expresar de una forma compacta la rotación (submatriz  $3 \times 3$  superior izquierda) y la traslación (vector columna  $3 \times 1$  formado por los tres primeros elementos de la cuarta columna).

En la herramienta MATLAB que se presenta en el apéndice,  ${}^A \Pi$  (que indica la posición de un punto en el sistema de referencia {A}) se expresa mediante el correspondiente vector de cuatro componentes. Así es posible definir  $\text{PenA}=[x \ y \ z \ 1]'$  como el vector extendido que indica las coordenadas del punto P en el sistema {A}.

Asimismo, el sistema de referencia con una posición y una orientación dadas se define y representa mediante la función `frame (TT, color, tam)`, donde TT es la matriz de transformación (3.20), color especifica el color en que se representará gráfica-

mente el sistema de referencia, y  $\text{tam}$  indica el tamaño deseado para las flechas que representan el sistema de coordenadas.

Las expresiones anteriores de rotación y traslación pueden interpretarse mediante operadores (Craig, 1989). Así, el operador de traslación de un vector  ${}^A P_1$  mediante el vector  ${}^A Q$ :

$${}^A P_2 = {}^A P_1 + {}^A Q; \quad (3.21)$$

puede expresarse mediante

$${}^A \Pi_2 = \text{Tras}({}^A Q, |Q|) {}^A \Pi_1 \quad (3.22)$$

siendo el operador de traslación

$$\text{Tras}({}^A Q, |Q|) = \begin{bmatrix} 1 & 0 & 0 & q_x \\ 0 & 1 & 0 & q_y \\ 0 & 0 & 1 & q_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.23)$$

En la herramienta MATLAB del apéndice existen diferentes funciones que permiten realizar traslaciones. Así, la función:

$$T = \text{transl}(x, y, z)$$

devuelve una matriz de transformación  $T$  que representa una traslación dada por los tres escalares ( $x, y, z$ ), mientras que

$$T = \text{transl}(v)$$

devuelve la matriz de transformación resultado de la traslación por el vector  $v$ .

De igual forma, en la rotación  $R$  de un vector  ${}^A P_1$ :

$${}^A P_2 = R {}^A P_1 \quad (3.24)$$

la matriz de rotación  $R$  es la misma que la que describe un marco rotado con respecto al marco de referencia.

La rotación según el eje  $\hat{K}$  en una magnitud  $\theta$  puede expresarse como

$${}^A \Pi_2 = \text{Rot}(\hat{K}, \theta) {}^A \Pi_1 \quad (3.25)$$

Como caso particular, el operador que rota alrededor del eje  $\hat{Z}$  se expresa mediante la transformación homogénea

$$Rot(\hat{Z}, \theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.26)$$

En la práctica, basta con emplear la matriz de rotación  $3 \times 3$  para aplicar el operador de rotación.

El operador de traslación y rotación es la matriz de transformación homogénea.

$${}^A\Pi_2 = T {}^A\Pi_1 \quad (3.27)$$

En la herramienta MATLAB existen también funciones específicas para la realización de rotaciones. Así, la transformación de rotación de un ángulo theta en torno a un vector v se expresa mediante:

$$T = \text{rotvec}(v, \text{theta})$$

siendo el resultado una transformación homegénica T.

Asimismo, las rotaciones de un ángulo theta alrededor de los ejes  $\hat{X}$ ,  $\hat{Y}$  y  $\hat{Z}$  se realizan mediante las funciones específicas `rotx(theta)`, `roty(theta)` y `rotz(theta)` respectivamente.

### Ejemplo 3.3

Si, en el ejemplo 3.1, la posición del obstáculo con respecto a un sistema de coordenadas solidario al robot es

$${}^A\Pi_1 = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 1 \end{bmatrix} \quad (3.28)$$

y se realiza una rotación de 20 grados alrededor del eje  $\hat{Z}$ , se tendrá:

$${}^A\Pi_2 = Rot(\hat{Z}, 20^\circ) {}^A\Pi_1 = \begin{bmatrix} 0,940 & -(0,342) & 0 & 0 \\ 0,342 & 0,940 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -(0,684) \\ 1,879 \\ 0 \\ 1 \end{bmatrix} \quad (3.29)$$

Empleando la herramienta MATLAB que se presenta en el apéndice, basta utilizar las instrucciones:

$$\begin{aligned} P1enA &= [0 \ 2 \ 0 \ 1]'; \\ P2enA &= rotz(pi/9)*P1enA \end{aligned}$$

Si el robot se encuentra en unas coordenadas (8,4), las coordenadas del obstáculo en el sistema de referencia {A} se obtendrían haciendo:

$$\begin{aligned} P1enA &= [0 \ 2 \ 0 \ 1]'; \\ P2enA &= transl(8,4,0)*rotz(pi/9)*P1enA; \end{aligned}$$

Asimismo, si el robot se encuentra en unas coordenadas (8,4), la matriz de transformación entre el sistema {B} asociado al robot y el sistema de referencia fijo {A} (ver ejemplo 3.1) viene dada por:

$${}^A_B T = \begin{bmatrix} 0,940 & -(0,342) & 0 & 8 \\ 0,342 & 0,940 & 0 & 4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.30)$$

Nótese que la relación entre sistemas de referencia puede visualizarse claramente en MATLAB mediante las instrucciones:

```
TA = [1 0 0 0;
      0 1 0 0;
      0 0 1 0;
      0 0 0 1];
TB = transl(8,4,0)*rotz(pi/9)*TA;
frame(TA, 'c', 1);
frame(TB, 'b', 1);
axis([0 9 -2 6 0 5]) % Rangos de representación
rotate3d % Permite girar libremente el punto de vista
grid on % Dibujo de rejilla
```

El resultado es el que se muestra en la figura 3.14.

Conviene poner de manifiesto que las matrices de transformación homogénea y los operadores representados mediante matrices  $4 \times 4$  tienen interés para escribir de forma más compacta las relaciones involucradas, pero no suelen utilizarse para la implantación de sistemas de control de robots ya que requieren operaciones y memoria adicional en la realización de las transformaciones.

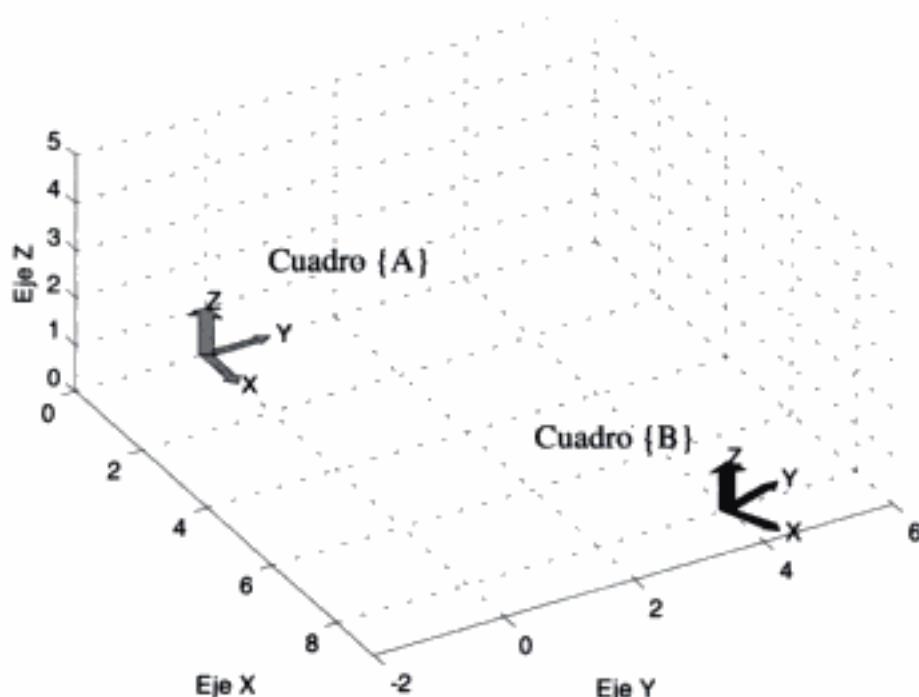


Figura 3.14: Transformaciones entre sistemas de referencia.

### 3.3 TRANSFORMACIÓN INVERSA

Otra operación importante es la obtención de la transformación inversa. Suponiendo  $\{B\}$  conocido con respecto a  $\{A\}$  ( ${}^B_T$  conocida) se desea expresar  $\{A\}$  con respecto a  $\{B\}$ .

En otros términos, se trata de obtener

$${}^B_A T = {}^A_B T^{-1} \quad (3.31)$$

sin necesidad de realizar la operación de inversión de la matriz.

Se sabe que

$${}^B_A R = {}^A_B R^T \quad (3.32)$$

Expresando  ${}^A P_{ORG_B}$  en  $\{B\}$ :

$${}^B({}^A P_{ORG_B}) = {}^B_A R {}^A P_{ORG_B} + {}^B P_{ORG_A} \quad (3.33)$$

Como la parte izquierda de la igualdad debe ser cero (origen de  $\{B\}$  expresado en  $\{B\}$ ) se tiene:

$${}^B P_{ORG_A} = -{}^B_A R {}^A P_{ORG_B} = -{}^B_R^T {}^A P_{ORG_B} \quad (3.34)$$

Por consiguiente, la expresión de la inversa de la transformación homogénea es:

$$\begin{matrix} {}^B_A T = \begin{bmatrix} {}^B_A R & {}^B_A P_{ORG A} \\ 0 0 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^A_B R^T & -{}^A_B R^T {}^A_A P_{ORG B} \\ 0 0 0 & 1 \end{bmatrix} \end{matrix} \quad (3.35)$$

La función de MATLAB `trinv(T)` devuelve la matriz de transformación inversa de aquella ( $T$ ) que se le pasa como parámetro, efectuando el cálculo tal como se indica en (3.35).

#### Ejemplo 3.4

Considérese el sistema de los ejemplos 3.1 y 3.3. La matriz de transformación homogénea se calcula en el ejemplo 3.3. La inversa de esta matriz es

$$\begin{matrix} {}^B_A T = \begin{bmatrix} 0,940 & 0,342 & 0 & -(8,886) \\ -(0,342) & 0,940 & 0 & -(1,023) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix} \quad (3.36)$$

Por otra parte, se tiene

$$\begin{matrix} {}^B_A R = {}^A_B R^T = \begin{bmatrix} 0,940 & 0,342 & 0 \\ -(0,342) & 0,940 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{matrix} \quad (3.37)$$

$$\begin{matrix} {}^A_B R^{TA} P_{ORG B} = {}^A_B R^T \begin{bmatrix} 8 \\ 4 \\ 0 \end{bmatrix} = \begin{bmatrix} -(8,886) \\ -(1,023) \\ 0 \end{bmatrix} \end{matrix} \quad (3.38)$$

con lo cual se comprueba la expresión de la transformación inversa

Si se emplean las líneas de MATLAB:

```
TBenA = [ 0.940 -0.342 0 8;
           0.342 0.940 0 4;
           0 0 1 0;
           0 0 0 1];
TAenB = trinv (TBenA)
```

el resultado es el mismo.

### 3.4 TRANSFORMACIONES COMPUSTAS

En numerosos problemas de robótica es necesario involucrar transformaciones compuestas generadas mediante una serie de operaciones elementales de traslación o rotación. Así, por ejemplo, resulta natural asociar un sistema de referencia a cada articulación de un manipulador. Si se desea conocer la localización de la pinza situada en el extremo del último enlace, es necesario estudiar la transformación compuesta que relaciona este extremo con la base del manipulador.

Considérese, por ejemplo, la transformación compuesta que se muestra en la figura 3.15.

Se supone que  $\{C\}$  es conocido con respecto a  $\{B\}$  y  $\{B\}$  conocido con respecto a  $\{A\}$ . Por consiguiente, si se tiene  ${}^B P$  y se desea obtener  ${}^A P$ , se procederá según

$${}^B P = {}_C^B T {}^C P; \quad (3.39)$$

$${}^A P = {}_B^A T {}^B P; \quad (3.40)$$

Sustituyendo la primera en la segunda

$${}^A P = {}_B^A T {}_C^B T {}^C P = {}_C^A T {}^C P; \quad (3.41)$$

siendo

$${}_C^A T = {}_B^A T {}_C^B T \quad (3.42)$$

una transformación compuesta que puede expresarse mediante la matriz homogénea:

$${}_C^A T = \begin{bmatrix} {}_B^A R {}_C^B R & {}_B^A R {}_C^B P_{ORG_C} + {}_A^B P_{ORG_B} \\ 0 & 1 \end{bmatrix} \quad (3.43)$$

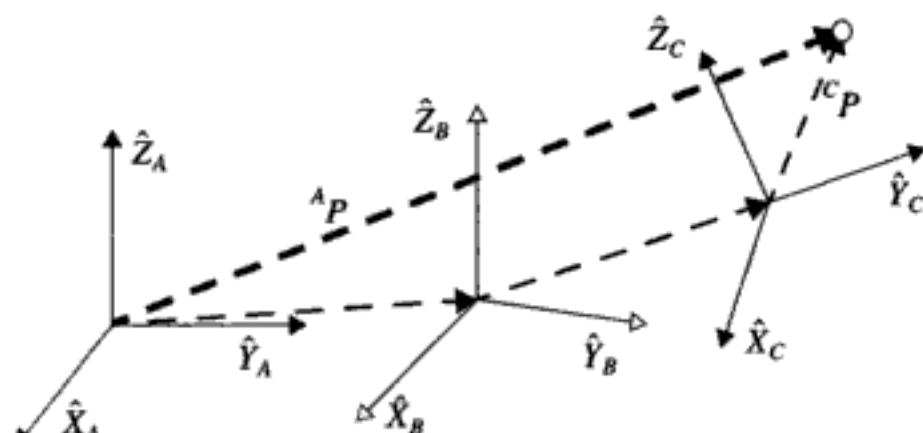
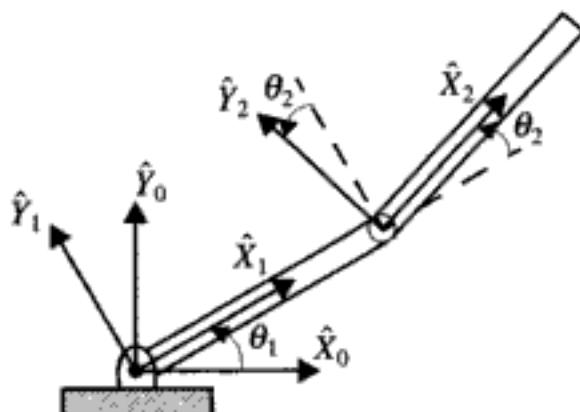


Figura 3.15: Transformaciones compuestas.

**Ejemplo 3.5**

Considérese un robot manipulador con dos articulaciones que se mueve en el plano con dos grados de libertad, tal como se muestra en la figura 3.16.



**Figura 3.16:** Manipulador plano con dos articulaciones de rotación.

En este caso se asocia un cuadro de referencia a cada articulación, tal como se indica en la figura 3.16.

Las matrices de rotación correspondientes son:

$${}^0_1R = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.44)$$

$${}^1_2R = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.45)$$

La transformación compuesta viene dada por la matriz de rotación:

$${}^0_2R = {}^0_1R {}^1_2R = \begin{bmatrix} \cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 & -\cos \theta_1 \sin \theta_2 - \sin \theta_1 \cos \theta_2 & 0 \\ \sin \theta_1 \cos \theta_2 + \cos \theta_1 \sin \theta_2 & \cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.46)$$

En lo que sigue se simplificará la notación escribiendo  $s_i$  en lugar de  $\sin \theta_i$ ,  $c_i$  en lugar de  $\cos \theta_i$ , y

$$\cos(\theta_1 + \theta_2) = c_1 c_2 - s_1 s_2 = c_{12} \quad (3.47)$$

$$\sin(\theta_1 + \theta_2) = c_1 s_2 + s_1 c_2 = s_{12} \quad (3.48)$$

Por consiguiente, la expresión anterior se escribirá:

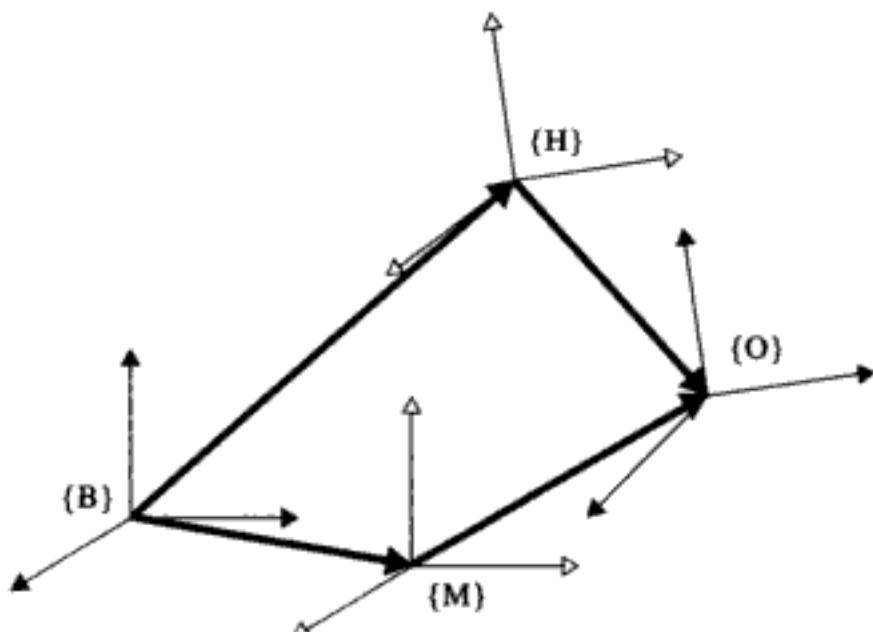
$${}^0_2R = {}^0_1R_2^I R = \begin{bmatrix} c_{12} & -s_{12} & 0 \\ s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.49)$$

\* \* \*

El empleo de transformadas compuestas y la transformada inversa permite resolver ecuaciones de transformación que tienen gran utilidad en numerosos problemas de robótica. Así, supóngase que un manipulador robótico provisto de una pinza  $\{H\}$  en su extremo agarra un objeto  $\{O\}$  definido con respecto a un sistema de referencia  $\{M\}$  asociado a una mesa. La situación de los sistemas de referencia se ilustra en la figura 3.17 (Craig, 1986).

Se suponen conocidas  ${}^B_H T$ , que describe la localización del sistema de referencia  $\{H\}$  asociado a la pinza del manipulador con respecto al de la base,  ${}^M_B T$  que describe el sistema de referencia  $\{M\}$  asociado a la mesa con respecto al de la base  $\{B\}$ , y  ${}^O_M T$  que describe el sistema de referencia  $\{O\}$  asociado al objeto con respecto al de la mesa  $\{M\}$ .

Se trata de obtener la posición y orientación del objeto con respecto a los dedos del manipulador.



**Figura 3.17:** Representación de ecuaciones de transformación.

La localización del objeto con respecto a la base viene dada por

$$\overset{B}{_O}T = \overset{B}{_M}T \overset{M}{_O}T \quad (3.50)$$

Por otra parte, la localización del objeto puede expresarse como

$$\overset{H}{_O}T = \overset{B}{_H}T \overset{H}{_O}T \quad (3.51)$$

Por consiguiente, igualando el lado derecho de ambas ecuaciones, puede obtenerse la posición y orientación del objeto con respecto a la pinza del manipulador mediante:

$$\overset{H}{_O}T = \overset{B}{_H}T^{-1} \overset{B}{_M}T \overset{M}{_O}T \quad (3.52)$$

### 3.5 OTRAS REPRESENTACIONES DE LA ORIENTACIÓN

Como se ha mencionado anteriormente, la orientación en el plano viene dada por un ángulo. Así, en un robot móvil que navega por un terreno plano sólo se necesita un ángulo de orientación, típicamente el ángulo con la dirección del Norte. En el espacio, la orientación viene dada por tres ángulos o, en general, por tres parámetros independientes.

Se sabe que una forma de representación de la orientación es la *matriz de rotación*. Esta matriz es  $3 \times 3$ , por lo que sus elementos no son independientes. Se recuerda que la matriz de rotación  $R = \{\hat{X} \hat{Y} \hat{Z}\}$  puede expresarse mediante los vectores unidad perpendiculares entre sí. Por consiguiente, existen seis restricciones en los elementos de la matriz de rotación:

$$\begin{aligned} |\hat{X}| &= 1; & |\hat{Y}| &= 1; & |\hat{Z}| &= 1; \\ \hat{X} \cdot \hat{Y} &= 0; & \hat{X} \cdot \hat{Z} &= 0; & \hat{Y} \cdot \hat{Z} &= 0 \end{aligned} \quad (3.53)$$

Nótese que las tres primeras restricciones corresponden a los módulos unidad, mientras que las tres restantes imponen un producto escalar cero (vectores perpendiculares).

Las matrices de rotación son interesantes para realizar operaciones de rotación, pero para representar la orientación se prefieren sistemas de ángulos tales como los que se indican a continuación:

#### Ángulos RPY

Se emplean los ángulos RPY: "roll" (balanceo), "pitch" (inclinación) y "yaw" (orientación) que, en terminología náutica, corresponden respectivamente a alabeo, cabeceo y guinada.

Supóngase que se desea expresar la orientación de un objeto, que tiene un sistema de ejes asociado  $\{B\}$ , con respecto a un sistema de referencia  $\{A\}$ . Se comienza con

{B} coincidente con la referencia {A} y se efectúa una rotación de {B} alrededor de  $\hat{X}_A$  un ángulo  $\gamma$  (balanceo). A continuación se efectúa una rotación alrededor de  $\hat{Y}_A$  un ángulo  $\beta$  (inclinación) y finalmente una rotación alrededor de  $\hat{Z}_A$  un ángulo  $\alpha$  (orientación). En la figura 3.18 (Craig, 1986) se ilustra el procedimiento.

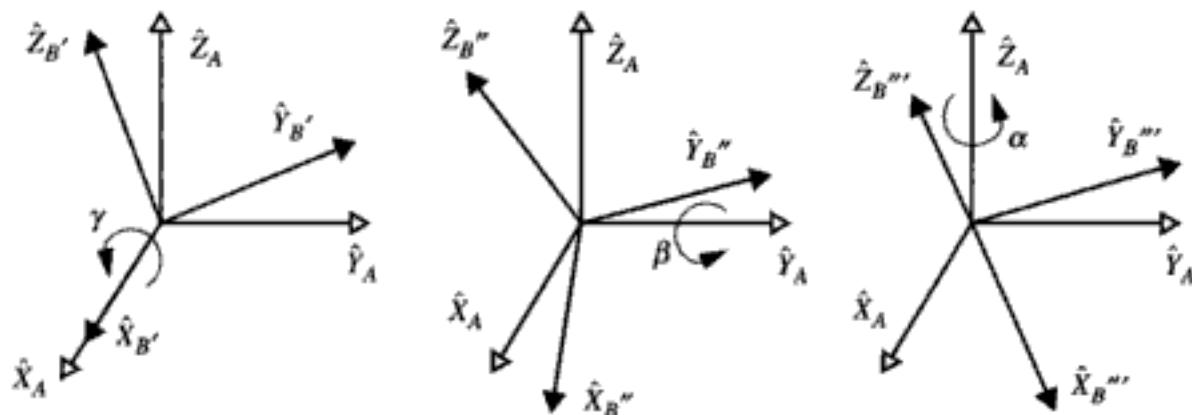


Figura 3.18: Ángulos RPY.

Empleando el operador de rotación, se tendría

$${}^A_B R_{RPY}(\gamma, \beta, \alpha) = Rot({}^A \hat{Z}_A, \alpha) Rot({}^A \hat{Y}_A, \beta) Rot({}^A \hat{X}_A, \gamma). \quad (3.54)$$

que puede expresarse como producto de matrices de rotación

$${}^A_B R_{RPY}(\gamma, \beta, \alpha) = \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix} \quad (3.55)$$

Efectuando la multiplicación de matrices se obtiene:

$${}^A_B R_{RPY}(\gamma, \beta, \alpha) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \quad (3.56)$$

En la herramienta MATLAB del apéndice existe una función que permite obtener la transformación homogénea a partir de los ángulos de RPY. Para utilizarla basta escribir:

`T = rpy2tr(gamma, beta, alpha)`

El problema inverso también tiene interés. Se trata ahora de obtener los ángulos de orientación conocida la matriz de rotación

$${}^A_B R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (3.57)$$

Para obtener las tres incógnitas pueden plantearse nueve ecuaciones de las cuales seis son dependientes. En la práctica, basta plantear tres ecuaciones con tres incógnitas. Por ejemplo, suponiendo  $c\beta \neq 0$ , es posible emplear la función arco tangente para calcular  $\gamma$  a partir de  $r_{32}$  y  $r_{33}$ . De forma más precisa, si se define la función  $\text{arctg } 2(r_{32}, r_{33})$  como el arco tangente del ángulo  $r_{32}/r_{33}$  teniendo en cuenta el signo de  $r_{32}$  y  $r_{33}$  para determinar el cuadrante, se tendrá:

$$\gamma = \text{arctg } 2(r_{32}, r_{33}) \quad (3.58)$$

Así, por ejemplo,

$$\gamma = \text{arctg } 2(-2, -2) = -135^\circ \quad (3.59)$$

mientras que

$$\text{arctg } 2(2, 2) = 45^\circ \quad (3.60)$$

y la función es indefinida cuando ambos argumentos son cero.

Asimismo, es posible obtener

$$\beta = \text{arctg } 2(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \quad (3.61)$$

escogiendo la solución  $-90^\circ \leq \beta \leq 90^\circ$ , que corresponde a tomar la raíz como positiva.

Finalmente, puede aplicarse la misma función para obtener

$$\alpha = \text{arctg } 2(r_{21}, r_{11}) \quad (3.62)$$

Existen soluciones degeneradas para  $\beta = \pm 90^\circ$  (siendo, por tanto,  $c\beta = 0$ ). En estos casos sólo puede calcularse la suma o la diferencia de  $\alpha$  y  $\gamma$ . Suponiendo que se elige  $\alpha = 0$ , se tienen las soluciones:

$$\alpha = 0, \quad \beta = 90^\circ, \quad \gamma = \text{arctg } 2(r_{12}, r_{22}) \quad (3.63)$$

$$\alpha = 0, \quad \beta = -90^\circ, \quad \gamma = -\text{arctg } 2(r_{12}, r_{22}) \quad (3.64)$$

En la herramienta MATLAB del apéndice, para obtener los ángulos RPY a partir de una transformación homogénea dada ( $T$ ) basta escribir:

$$[r \ p \ y] = \text{tr2rpy}(T)$$

**Ángulos de Euler Z-Y-X**

En esta representación, en lugar de realizar tres rotaciones consecutivas alrededor de los ejes del sistema de referencia {A}, las rotaciones se efectúan alrededor de los ejes del sistema {B} solidario al cuerpo, como se ilustra en la figura 3.19 (Craig, 1986).

Se comienza con {B} coincidente con la referencia {A} y se realiza una rotación de {B} alrededor de  $\hat{Z}_B$  un ángulo  $\alpha$  (balanceo). A continuación se efectúa una rotación alrededor de  $\hat{Y}_{B'}$ , resultante del primer giro, un ángulo  $\beta$  (inclinación). Finalmente, se realiza una rotación alrededor de  $\hat{X}_{B''}$ , resultante del segundo giro, un ángulo  $\gamma$  (orientación).

La rotación total puede expresarse como:

$${}^A_R = {}^A_{B'} R {}^{B'}_{B''} R {}^{B''}_B \quad (3.65)$$

de donde:

$${}^A_{B'} R_{zyx}(\alpha, \beta, \gamma) = Rot({}^B_{B'} \hat{Z}, \alpha) Rot({}^B_{B'} \hat{Y}, \beta) Rot({}^B_{B'} \hat{X}, \gamma) \quad (3.66)$$

Por consiguiente,

$${}^A_{B'} R_{zyx}(\alpha, \beta, \gamma) = \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix} \quad (3.67)$$

Operando se obtiene:

$${}^A_{B'} R_{zyx}(\alpha, \beta, \gamma) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \quad (3.68)$$

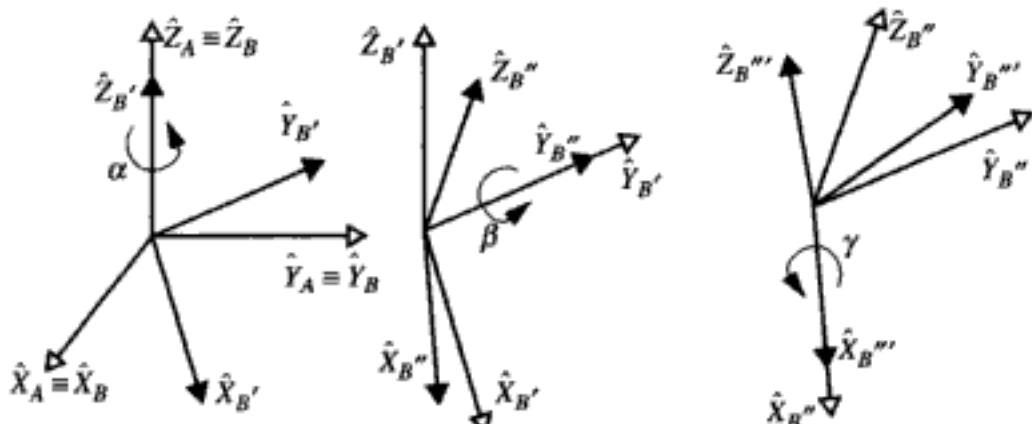


Figura 3.19: Ángulos de Euler Z-Y-X.

Podemos observar que la fórmula resultante es exactamente la misma que la calculada anteriormente para los ángulos RPY, por lo que para obtener los ángulos Z-Y-X a partir de la matriz (problema inverso) es posible aplicar las mismas expresiones que resultaron para los ángulos RPY.

### Ángulos de Euler Z-Y-Z

Los ángulos de Euler presentados anteriormente definen la orientación mediante tres giros consecutivos en los tres ejes de coordenadas. Realizando de forma diferente los giros, se obtienen otras representaciones. En el caso de los denominados ángulos de Euler Z-Y-Z, se comienza también con {B} coincidente con la referencia {A} y se efectúan sucesivamente la rotación de {B} alrededor de  $\hat{Z}_B$  un ángulo  $\alpha$ , y a continuación alrededor de  $\hat{Y}_B$  un ángulo  $\beta$ . Sin embargo, la tercera rotación es diferente ya que ahora vuelve a rotarse alrededor de  $\hat{Z}_B$  un ángulo  $\gamma$ .

En este caso llega a obtenerse una matriz de rotación (Craig, 1986):

$${}^A_R_{zyz}(\alpha, \beta, \gamma) = \begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix} \quad (3.69)$$

En la herramienta MATLAB del apéndice, la función que devuelve la transformación homogénea que corresponde a los ángulos de Euler Z-Y-Z (alpha, beta y gamma), que se le pasan como parámetros, es  $T = \text{eul2tr}(\text{alpha}, \text{beta}, \text{gamma})$ .

El problema inverso se resuelve ahora mediante las expresiones:

$$\gamma = \operatorname{arctg} 2(r_{32}, -r_{31}); \quad \beta = \operatorname{arctg} 2(\sqrt{r_{31}^2 + r_{32}^2}, r_{33}); \quad \alpha = \operatorname{arctg} 2(r_{23}, r_{13}) \quad (3.70)$$

suponiendo  $\sin \beta \neq 0$ .

El ángulo  $\beta$  se elige de forma que  $0 \leq \beta \leq 180^\circ$ , que corresponde a la raíz positiva.

De igual forma que se ha visto anteriormente, si  $\beta = 0$  o  $\beta = 180^\circ$  la solución es degenerada. En este caso sólo puede calcularse la suma o la diferencia de  $\alpha$  y  $\gamma$ .

Eligiendo  $\alpha = 0$  se obtienen las soluciones:

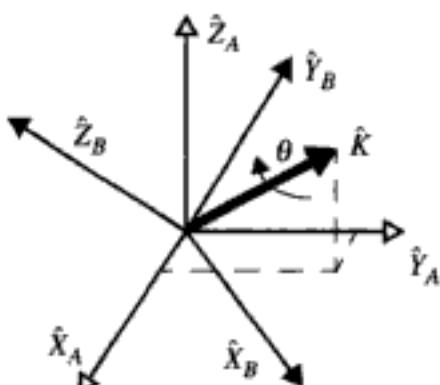
$$\alpha = 0, \quad \beta = 0, \quad \gamma = \operatorname{arctg} 2(-r_{12}, r_{11}) \quad (3.71)$$

$$\alpha = 0, \quad \beta = 180^\circ, \quad \gamma = \operatorname{arctg} 2(r_{12}, -r_{11}) \quad (3.72)$$

En la herramienta MATLAB del apéndice, los ángulos de Euler Z-Y-Z se obtienen a partir de una transformación homogénea ( $T$ ) dada, empleando la función: [alpha beta gamma] = tr2eul(T).

### Pares de rotación y cuaternios

Por último, conviene mencionar otras representaciones posibles de la orientación tales como los denominados pares de rotación o los cuaternios.



**Figura 3.20:** Representación de la orientación mediante pares de rotación.

Cuando se emplean *pares de rotación*, la orientación de un sistema {B} con respecto a un sistema {A} se define mediante el par  $(\hat{k}, \theta)$ , siendo  $\hat{k}$  un vector que define el eje de rotación y  $\theta$  el ángulo de rotación, como se muestra en la figura 3.20. Si se gira el sistema {A} un ángulo  $\theta$  sobre  $\hat{k}$ , se obtiene el sistema {B}.

Los cuaternios están constituidos también por cuatro componentes: un escalar y un vector de tres componentes. De forma más precisa se emplea el par:

$$[\cos(\theta/2), \hat{k} \sin(\theta/2)]$$

Para más detalles puede consultarse el texto Barrientos y otros (1997).

## 3.6 CONCLUSIONES

El modelado y control de robots hace necesario considerar posiciones y orientaciones en el espacio. En este capítulo se han introducido la notación y las expresiones básicas que se utilizarán con profusión en los capítulos posteriores.

Asimismo, se han presentado las transformaciones compuestas y la transformación inversa que se requieren también para el modelado y control tanto de los robots manipuladores como de los móviles.

## 3.7 REFERENCIAS

- Barrientos A. y otros, 1997. "Fundamentos de Robótica". McGraw-Hill.
- Corke P.I., 1996. "A Robotics Toolbox for MATLAB". IEEE Robotics and Automation Magazine". Vol. 3, núm. 1, pp. 24-32.
- Craig J.J., 1986. "Introduction to Robotics. Mechanics and Control". Addison Wesley.
- Fu K.S., R.C. González y C.S.G. Lee, 1988. "Robótica: control, detección, visión e inteligencia". McGraw-Hill.
- Paul R.P., 1981. "Robot manipulators: mathematics, programming and control". MIT Press.

# Modelos cinemáticos de robots

---

En este capítulo se estudian modelos geométricos y cinemáticos de los robots. Estos modelos se emplean tanto para la simulación como para el control. Los modelos se basan en el empleo de las transformaciones entre sistemas de referencia estudiadas en el capítulo precedente. En la primera parte, después de la introducción y de un apartado dedicado a los sistemas de referencia, se estudia el *modelo directo* que permite obtener la posición y orientación del efecto final del robot en función de las variables de las articulaciones. A continuación, se contempla el *modelo inverso*: obtención de las variables articulares que hacen que la posición y orientación del efecto final sea la deseada. En la segunda parte del capítulo se tiene en cuenta también la *velocidad del movimiento* considerando la relación entre las variaciones de las variables articulares y las variaciones de las variables en el espacio cartesiano. Siguiendo el mismo enfoque que en otros capítulos de este texto, se consideran tanto robots manipuladores como *vehículos robóticos* a los cuales se dedica la última sección del capítulo. En los ejemplos de este capítulo se emplea la herramienta MATLAB-Simulink, que se describe en el apéndice del texto.

## 4.1 INTRODUCCIÓN

---

Se entiende por modelo geométrico del robot la relación entre los valores de las variables asociadas a las articulaciones del robot y la situación (posición y orientación) de un sistema de referencia, solidario al robot, que se define teniendo en cuenta la tarea que se pretende desarrollar con el robot. En un robot manipulador, este sistema suele elegirse asociado al efecto final del robot, con lo cual se trata de estudiar la relación entre las variables articulares y la posición y orientación del efecto final con relación a la base del robot.

En vehículos robóticos, el sistema de referencia se elige normalmente asociado al punto de guía deseado.

Se entiende por *cinemática* el estudio del movimiento sin considerar las fuerzas que lo producen. Por tanto, se trata de estudiar tanto las propiedades geométricas como las temporales del movimiento. En estos términos, se considera, además del problema puramente geométrico involucrado en el posicionamiento estático, las variaciones en el tiempo de las posiciones y orientaciones; es decir, las velocidades.

En los modelos geométricos y cinemáticos se involucra esencialmente el estudio de las relaciones existentes entre el espacio de las variables articulares y el espacio de trabajo, o espacio operacional, que suele ser un espacio cartesiano.

Como se ha visto en el capítulo anterior, en un caso general, el espacio de las variables articulares tendrá  $n$  dimensiones, con lo que se empleará un vector de dimensión  $n$  para especificar la posición y orientación del robot.

El espacio de trabajo de un manipulador en el espacio tendrá dimensión seis, necesitándose seis valores para especificar la posición y orientación. En manipuladores planares, o en robots móviles que navegan en el plano, bastaría con un espacio de dimensión tres para especificar la posición y el ángulo de orientación.

Algunos autores (Craig, 1986) distinguen entre el espacio de variables articulares y el espacio de los actuadores, asociado directamente a las características de los actuadores. En este caso, la variable articular no es directamente la asociada a un determinado actuador, sino que se obtiene a partir de uno o varios actuadores, que pueden ser de diferente naturaleza. Así, se consideraría una articulación de traslación aunque el movimiento se genere inicialmente mediante un motor rotacional.

## 4.2 RELACIONES ENTRE SISTEMAS DE REFERENCIA

En la figura 4.1 se ilustra una aplicación robótica general en la que se involucra un robot móvil consistente en un vehículo robótico dotado de un manipulador. Se trata de estudiar la relación entre los diferentes sistemas de referencia.

Sea  $\{V\}$  el sistema o cuadro de referencia asociado al vehículo y  $\{B\}$  el asociado a la base del manipulador. Normalmente no existirá ningún movimiento entre ellos, por lo que se utilizará únicamente el  $\{B\}$  que se relaciona con el  $\{V\}$  empleando una transformación constante.

El manipulador está definido por una cadena cinemática en cuyo extremo existe una muñeca de sistema de referencia  $\{M\}$ .

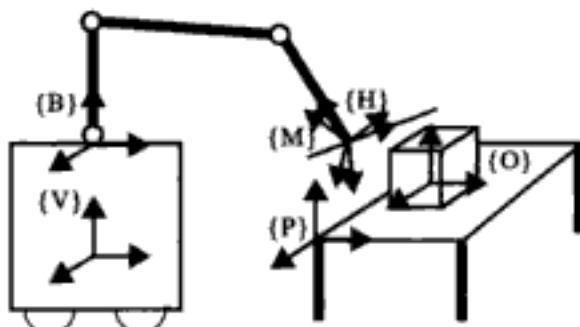


Figura 4.1: Asignación de sistemas de referencia.

Otro sistema asociado al manipulador es el  $\{H\}$ , solidario a la herramienta que soporta la muñeca del manipulador. Este cuadro se define, con respecto al de la muñeca  $\{M\}$ , dependiendo de las características de la herramienta y de cómo se sostenga. En pinzas convencionales suele elegirse con el origen del sistema en el centro del espacio entre los dedos.

Por otra parte, se tienen los sistemas asociados a la tarea u objeto que se pretende manipular. Así, puede identificarse un sistema objetivo  $\{O\}$  que permite definir dónde debe colocar el robot la herramienta para realizar la tarea. Este sistema es solidario al objeto que se pretende manipular. Normalmente resulta conveniente definir  $\{O\}$  con respecto a un sistema  $\{P\}$  característico del puesto de trabajo en el que se realiza la tarea. De esta forma, si se desea realizar la misma tarea en otro puesto de trabajo basta con modificar  $\{P\}$ .

A su vez, el sistema característico del puesto de trabajo está relacionado con respecto al robot mediante una transformación  ${}^P_T$ .

Al final del movimiento,  $\{H\}$  debe coincidir con  $\{O\}$ .

Para posicionar la herramienta interesa calcular  $\{H\}$  relativo a  $\{P\}$ . Con este objetivo pueden efectuarse las transformaciones:

$${}^P_H T = {}^P_T {}^{-1} \cdot {}^P_M T \cdot {}^M_H T \quad (4.1)$$

Esta ecuación permite definir un modelo generalizado del robot a partir del cual es posible calcular dónde está la herramienta que lleva el manipulador con respecto a un sistema de referencia de la estación de trabajo.

La transformación que involucra la cadena cinemática formada por las articulaciones del manipulador es la  ${}^P_M T$ . En su formulación se emplean las ecuaciones cinemáticas de los enlaces del manipulador.

Si los sistemas de referencia de las articulaciones son  $\{0\}, \{1\}, \dots, \{n\}$ , el sistema  $\{B\}$  coincide con el  $\{0\}$  y el  $\{M\}$  con el  $\{n\}$ . El sistema  $\{M\}$  se define con respecto a la base mediante una transformación:

$$\{M\} = {}^P_M T = {}^0_n T \quad (4.2)$$

que se obtiene mediante las ecuaciones cinemáticas del manipulador. En los apartados que siguen se estudia con detalle la obtención de esta transformación que relaciona el espacio de las variables articulares con el espacio cartesiano en el que se define  $\{M\}$ .

En la última parte del capítulo se estudiará la relación que existe entre las variables asociadas a los ejes del vehículo y la posición de éste con relación a  $\{P\}$ . En este punto conviene poner de manifiesto que, debido a la redundancia existente en el número de grados de libertad, una vez que el objetivo esté suficientemente cerca es posible planificar diferentes movimientos alternativos combinando los del vehículo y los del manipulador. Sin embargo, el problema suele simplificarse resolviéndolo en dos pasos: se realiza primero el posicionamiento del vehículo  ${}^P_T$  para, a continuación, posicionar el manipulador.

## 4.3 MODELO DIRECTO DEL MANIPULADOR

El modelo directo viene dado por una función que permite expresar la posición y orientación del sistema de referencia objetivo en el espacio cartesiano (operacional o de la tarea)  $p$  en términos de las variables articulares  $q$ :

$$p = \varphi(q) \quad (4.3)$$

siendo  $\varphi$  un conjunto de funciones no lineales.

Nótese el interés del modelo directo para la *simulación del manipulador*: Conocidas las variables articulares (entradas), se desea saber dónde está el manipulador.

Para construir el modelo directo de un determinado robot existen dos alternativas típicas:

- Utilización de relaciones específicas del robot: Consiste en el empleo de las relaciones geométricas que puedan establecerse en el robot de que se trate.
- Método general: Se trata de emplear matrices de transformación que relacionan sistemas de referencia tal como se definieron en el capítulo anterior.

En lo que sigue se estudia el método general.

### 4.3.1 Modelo de un robot manipulador con $n$ articulaciones

Un robot articulado puede describirse definiendo cuatro magnitudes asociadas a cada articulación. Una de estas magnitudes es la variable de la articulación y las restantes son parámetros fijos para cada robot.

En la figura 4.1 se muestran estos parámetros que permiten definir una representación de las relaciones de traslación y rotación entre los enlaces adyacentes. A esta relación se le denomina representación de Denavit-Hartenberg (1955).

Así, la variable de una articulación  $i$  de rotación se representará mediante el ángulo  $\theta_i$  y la de una prismática mediante el desplazamiento  $d_i$ . Los otros dos parámetros de la articulación son la distancia  $a_{i-1}$  entre el eje de la articulación  $i-1$  y el eje de la articulación  $i$ , medida sobre la línea perpendicular común, y el ángulo  $\alpha_{i-1}$  entre estos dos ejes (ángulo entre las proyecciones de los dos ejes en un plano cuya normal es la perpendicular común) medido como rotación alrededor de la perpendicular común hasta hacer coincidir las direcciones de los ejes. Cuando el eje  $i-1$  y el  $i$  intersectan, el valor del parámetro  $a_{i-1}$  es cero.

Las cadenas cinemáticas se describirán indicando los cuatro parámetros de Denavit-Hartenberg de las articulaciones. En la primera articulación de la cadena, el valor de los parámetros  $a_0, \alpha_0$ , es arbitrario y se toma como cero. Si la articulación es de rotación, el parámetro  $d_i$  correspondiente se toma también cero. Cuando la articulación es prismática, el parámetro  $\theta_i$  se hace igual a cero.

Los sistemas de referencia se asignan haciendo coincidir uno de los ejes del sistema de coordenadas, típicamente el  $\hat{Z}_i$ , con el eje de la articulación. El origen de  $\{i\}$  se escoge en el punto en el que la línea sobre la que se define  $a_i$  interseca el eje de la articulación  $i$ . El eje  $\hat{X}_i$  se elige en la dirección de la perpendicular común entre el eje de la articulación y la siguiente. Para elegir el eje  $\hat{Y}_i$  se sigue la regla de la mano derecha.

Por consiguiente,  $a_i$  y  $\alpha_i$  resultan ser respectivamente la distancia desde  $\hat{Z}_i$  hasta  $\hat{Z}_{i+1}$  y el ángulo entre estos dos ejes medida sobre el eje  $\hat{X}_i$ . El signo de  $\alpha_i$  será positivo si al llevar  $\hat{Z}_i$  sobre  $\hat{Z}_{i+1}$  por el camino más corto, el sentido que resulte de aplicar la regla de la mano derecha es el mismo que el del vector  $\hat{X}_i$ . Asimismo,  $d_i$  y  $\theta_i$  son, respectivamente, la distancia desde  $\hat{X}_{i-1}$  hasta  $\hat{X}_i$  y el ángulo entre estos dos ejes medidos sobre el eje  $\hat{Z}_i$ . El signo de  $\theta_i$  será positivo si al llevar  $\hat{X}_{i-1}$  sobre  $\hat{X}_i$  por el camino más corto, el sentido que resulte de aplicar la regla de la mano derecha es el mismo que el del vector  $\hat{Z}_i$ .

El sistema de referencia {0} se elige de forma que  $\hat{Z}_0$  coincide con  $\hat{Z}_1$  cumpliéndose siempre que  $a_0 = \alpha_0 = 0$ . Cuando la última articulación es de rotación, el sistema de referencia {n} se elige con la dirección de  $\hat{X}_n$  alineada con  $\hat{X}_{n-1}$  cuando  $\theta_n = 0$ . Si {n} es prismática, su origen se elige en la intersección entre su eje y  $\hat{X}_{n-1}$  cuando  $d_n = 0$ .

Conviene poner de manifiesto que la elección de los sistemas de referencia no es única. Existen variantes según se tome el sentido del eje  $\hat{Z}_i$ , o bien en situaciones tales como la que se presenta cuando intersectan dos ejes consecutivos de articulaciones  $\hat{Z}_i$  y  $\hat{Z}_{i+1}$  ( $a_i = 0$ ) y puede escogerse el signo de  $\alpha_i$  en uno u otro sentido de la normal al plano formado por  $\hat{Z}_i$  y  $\hat{Z}_{i+1}$ .

En lo anterior se ha escogido la convención de Craig (1986), para la elección de los sistemas de referencia y parámetros de Denavit-Hartenberg, que es la que se utilizará a lo largo de todo este texto. Existen otras formas de seleccionar dichos parámetros tales como las que se emplean en Fu y otros (1988) y en Paul (1981) que es la utilizada en el "Robotics Toolbox" de Corke (1996). En la herramienta MATLAB que se presenta en el apéndice se hace uso de la notación de Craig, por lo cual el lector podrá trabajar directamente con los ejemplos del texto.

Como se ha mencionado al comienzo de este apartado, el modelado de manipuladores hace necesario representar {i} con respecto a {i-1}. Cada transformación puede definirse según tres parámetros y una variable de la articulación. Cabe hablar también de una transformación compuesta por cuatro transformaciones elementales, tal como se ilustra en la figura 4.2. En la notación del capítulo anterior, si se supone que entre la localización inicial {i-1} y la final {i} existen tres localizaciones intermedias P, Q y R (figura 4.2), se tendrá (Craig, 1986):

$${}^{i-1}_iT = Rot(\hat{X}_{i-1}, \alpha_{i-1})Tras(\hat{X}_{i-1}, a_{i-1})Rot(\hat{Z}_i, \theta_i)Tras(\hat{Z}_i, d_i) \quad (4.4)$$

Si se componen estas transformaciones aplicando las matrices de transformación elementales para las rotaciones y las traslaciones, se obtiene la siguiente forma general asociada a la articulación:

$${}^{i-1}_iT = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

donde  $s$  significa seno y  $c$  coseno.

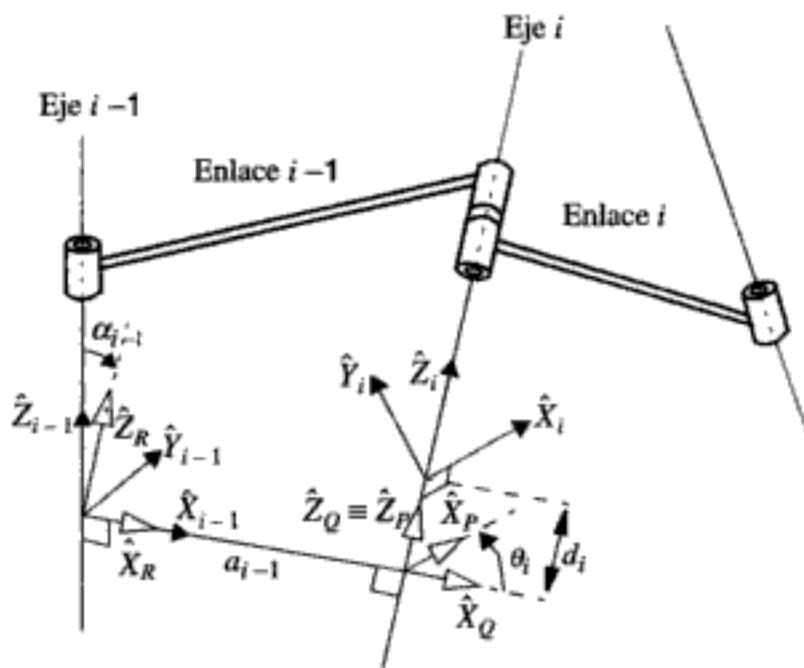


Figura 4.2: Asignación de cuadros de referencia a articulaciones consecutivas.

Para construir el modelo directo de un robot con  $n$  articulaciones es necesario definir un sistema de referencia solidario a cada segmento y elegir sus parámetros. A continuación puede procederse a la obtención de matrices de transformación de cada articulación.

A partir de las matrices de cada articulación puede obtenerse la transformación compuesta  ${}^nT$  que relaciona la localización  $\{n\}$  con la  $\{0\}$ . Esta transformación será función de las  $n$  variables de las articulaciones:  $q_1, q_2, \dots, q_n$ .

Por consiguiente, dados los valores de  $q_1, q_2, \dots, q_n$ , puede calcularse la posición y orientación cartesiana de la última articulación como:

$$p = \varphi(q) = {}^0T \cdot {}^1T \cdots {}^{n-1}T \cdot {}^nT \quad (4.6)$$

La aplicación de estas ecuaciones permite estimar la posición y orientación del efecto final del manipulador conociendo los valores de las variables articulares. No obstante, desde el punto de vista de la implantación, resulta más interesante desarrollar las ecuaciones involucradas eliminando las expresiones redundantes.

En efecto, obsérvese que para representar la orientación sólo se necesitan tres parámetros, mientras que existen nueve términos en la submatriz de rotación  $3 \times 3$  de  $T$  que representa la orientación.

Asimismo, la factorización de las ecuaciones permite reducir el número de multiplicaciones y sumas creando variables auxiliares, lo que elimina la repetición del cálculo de los mismos términos.

Una vez efectuadas estas simplificaciones, el mayor coste computacional suele estar en el cálculo de los senos y cosenos, por lo que, si es necesario acelerar más el cálculo, puede resultar interesante tabular estas funciones con un nivel de resolución apropiado.

Por último, en sistemas con un “hardware” mínimo, puede resultar interesante emplear exclusivamente operaciones en punto fijo. Nótese que el rango de variación de las variables involucradas suele ser relativamente pequeño y conocido previamente, lo que facilita la programación de las operaciones en punto fijo.

### Ejemplo 4.1

Considérese el manipulador plano con tres articulaciones de rotación que se muestra en la figura 4.3 (Craig, 1986). Las longitudes de los dos primeros segmentos son  $l_1$  y  $l_2$ .

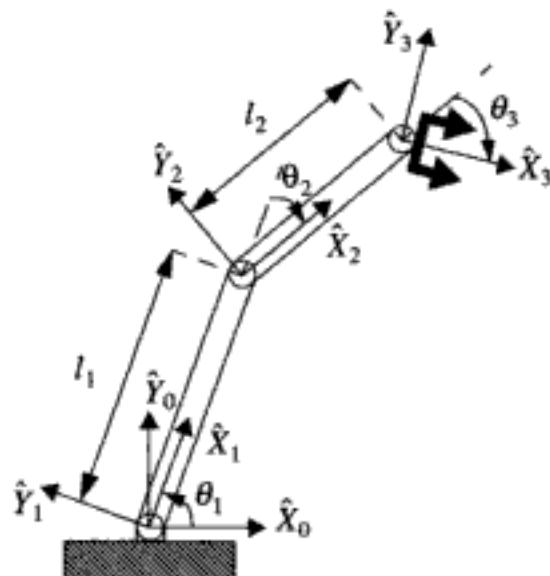


Figura 4.3: Manipulador plano con tres articulaciones de rotación.

La asignación de los marcos de referencia se muestra también en la figura 4.3. Los ejes  $\hat{Z}_0$ ,  $\hat{Z}_1$  y  $\hat{Z}_2$  de los marcos de referencia {0}, {1} y {2} son paralelos y en la misma dirección que los ejes de las tres articulaciones apuntando hacia fuera. Por consiguiente, los parámetros  $d_i$  y los  $\alpha_i$  son todos nulos. En la tabla 4.1 se indican los parámetros de Denavit-Hartenberg resultado de la asignación de los sistemas de referencia que se muestran en la figura 4.3. Nótese que las variables  $\theta_1$ ,  $\theta_2$  y  $\theta_3$  tienen signo. En la configuración que se muestra en la figura 4.3,  $\theta_1$  es positiva y  $\theta_2$  y  $\theta_3$  son negativas.

Tabla 4.1: Parámetros Denavit-Hartenberg del manipulador planar con tres articulaciones de rotación.

$i$	$\alpha_{i-1}$	$a_{i-1}$	$\theta_i$	$d_i$
1	0	0	$\theta_1$	0
2	0	$l_1$	$\theta_2$	0
3	0	$l_2$	$\theta_3$	0

Sea  $c_i = \cos \theta_i$  y  $s_i = \sin \theta_i$ . Las matrices de transformación de las tres articulaciones son:

$${}^0T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.7)$$

$${}^1T = \begin{bmatrix} c_2 & -s_2 & 0 & l_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

$${}^2T = \begin{bmatrix} c_3 & -s_3 & 0 & l_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.9)$$

Por tanto, el modelo directo viene dado por

$${}^0T = {}^0T {}^1T {}^2T = \begin{bmatrix} c_{123} & -s_{123} & 0 & l_1c_1 + l_2c_{12} \\ s_{123} & c_{123} & 0 & l_1s_1 + l_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.10)$$

siendo

$$c_{12} = c_1c_2 - s_1s_2 \quad (4.11)$$

$$s_{12} = c_1s_2 + s_1c_2 \quad (4.12)$$

$$c_{123} = c_{12}c_3 - s_{12}s_3 \quad (4.13)$$

$$s_{123} = s_{12}c_3 - c_{12}s_3 \quad (4.14)$$

A continuación se trata el ejemplo con la herramienta MATLAB que se describe en el apéndice. Para ello es necesario introducir los parámetros de Denavit-Hartenberg en una matriz que se utiliza como parámetro en las funciones MATLAB de la mencionada herramienta. En la tabla 4.2 se muestra la matriz de parámetros correspondiente al ejemplo.

**Tabla 4.2:** Formato de las matrices de parámetros que hay que pasarles a las funciones de MATLAB.

$i$	$\alpha_{i-1}$	$a_{i-1}$	$\theta_i$	$d_i$	$\sigma_i$
1	0	0	$\theta_1$	0	0
2	0	$l_1$	$\theta_2$	0	0
3	0	$l_2$	$\theta_3$	0	0

La última columna indica el tipo de articulación: 0 si es de rotación y distinta de 0 si es prismática. Si no se escribe la última columna, las funciones de la herramienta MATLAB tratan la articulación como si fuera de rotación.

Para resolver el problema se escribirían las siguientes líneas:

```

syms t1 t2 t3 real; % Variables articulares  $\theta_1$ ,  $\theta_2$  y  $\theta_3$ 
syms l1 l2 real; % Parámetros  $l_1$  y  $l_2$ 
% Introducimos la matriz con los parámetros de Denavit-Hartenberg
dh = [0 0 t1 0 0;
       0 l1 t2 0 0;
       0 l2 t3 0 0];

```

A continuación se llama a la función con la que se puede obtener la matriz  ${}^0T_3$ . Dicha función es `fkine(dh,q)` y sus parámetros son la matriz de parámetros de Denavit-Hartenberg y el vector de variables articulares. Por tanto, para solucionar el problema sólo resta escribir:

```

q=[t1 t2 t3]; % Vector de variables articulares
T = simple( fkine(dh,q) )

```

La función `simple` se encarga de simplificar la expresión resultante.

Es importante que el lector tenga claro que la función `fkine` toma los valores para las variables articulares del segundo parámetro que se le pasa, y no de los que aparecen en la matriz `dh`. Esto quiere decir que si la articulación es de rotación no se tendrá en cuenta el valor del tercer elemento de la fila correspondiente de la matriz `dh`. En caso de que la articulación sea prismática, será el cuarto elemento de la fila el que no se tenga en cuenta. Por tanto, el resultado obtenido habría sido el mismo si se hubiera escrito la siguiente matriz de parámetros de Denavit-Hartenberg:

```

dh = [0 0 376 0 0;
       0 l1 212 0 0;
       0 l2 789 0 0];

```

### Ejemplo 4.2

Considérese el robot que se muestra en la figura 4.4 (Craig, 1986).

El sistema de referencia {0} se supone rígidamente unido al enlace 0 (anclaje del

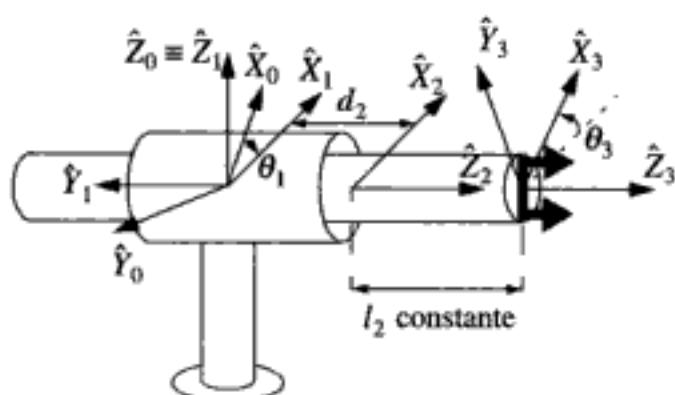


Figura 4.4: Esquema del robot del ejemplo 4.2.

robot) y el  $\{n\}$  al último enlace. El sistema  $\{1\}$  coincide con el  $\{0\}$  para  $\theta_1 = 0$ , pero gira con respecto a éste al variar  $\theta_1$ .

Al igual que en el ejemplo 4.1, para simplificar la representación, los ejes de movimiento se eligen coincidentes con algún eje de los sistemas de referencia, y en particular con el eje  $\hat{Z}$ . Así, en la primera y tercera articulación (rotacionales), existen rotaciones alrededor de  $\hat{Z}_1$  y de  $\hat{Z}_3$ , y en la segunda (prismática) un desplazamiento a lo largo de  $\hat{Z}_2$ .

Siguiendo la notación que se introdujo para los parámetros de Denavit-Hartenberg, en la articulación prismática la variable es la  $d_2$ , siendo fijo el valor de  $\theta_2$ . El origen del sistema de referencia  $\{2\}$  se elige de forma que, cuando  $d_2$  es cero, corresponda a la extensión mínima del enlace, con lo cual la variable  $d_2$  mide la extensión de la articulación. Si se escogen los sistemas de referencia como en la figura 4.4, los ejes de las articulaciones  $\{1\}$  y  $\{2\}$  intersectan y, por consiguiente, el desplazamiento  $a_1$  es cero. Nótese también cómo  $\hat{Z}_1$  y  $\hat{Z}_2$  están rotados 90 grados:  $\alpha_1 = 90^\circ$ . De la observación de las figuras 4.2 y 4.4 se deducen también el resto de los valores de los parámetros que se muestran en la tabla 4.3.

Tabla 4.3: Parámetros Denavit-Hartenberg del manipulador cilíndrico de la figura 4.4.

$i$	$\alpha_{i-1}$	$a_{i-1}$	$\theta_i$	$d_i$
1	0	0	$\theta_1$	0
2	$90^\circ$	0	0	$d_2$
3	0	0	$\theta_3$	$l_2$

En este punto conviene poner de manifiesto que existen otras posibles asignaciones de sistemas de referencia que dan origen a otros valores de los parámetros de Denavit-Hartenberg. En Craig (1986) se muestran otras cuatro posibles asignaciones para este mismo manipulador.

Las transformaciones resultantes, tomando como variables las de la tabla 4.3, son:

$${}^0_1T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.15)$$

$${}^1_2T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -d_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.16)$$

$${}^2_3T = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.17)$$

A continuación se indica cómo puede emplearse la herramienta del apéndice con el objetivo de comprobar los resultados obtenidos. Para obtener las matrices de transformación entre enlaces intermedios se puede emplear la función `linktrans(dh,q)`, teniendo presente que `dh` sería un vector fila con los parámetros de Denavit-Hartenberg del enlace en estudio y `q` el valor de la variable articular de dicho enlace. Es importante resaltar que la función `linktrans` ha sido elaborada de tal modo que ignora el valor de las variables articulares que se le pasan en el vector fila `dh` y solamente usa el valor que se le da en `q`. De este modo, si la articulación es de rotación se ignora el tercer elemento del vector `dh`, y si es prismática no se tendrá en cuenta el cuarto elemento. Por tanto, no debe extrañar al lector el resultado de las siguientes órdenes:

```

syms t1 d2 t3 real;      d2 y θ3           % Variables articulares θ1,  

syms l2 real;             % Parámetro l2  

T1 = linktrans ([0 0 125 0 0],t1)          % Matriz {}01T  

T2 = linktrans ([pi/2 0 0 426 1],d2)         % Matriz {}12T. Hemos dado  

                                              a σi un valor distinto de  

                                              cero (1) porque la  

                                              segunda articulación es  

                                              prismática.  

T3 = linktrans ([0 0 876 l2 0],t3)          % Matriz {}23T

```

Obsérvese que cuando se usa la función `linktrans` en la forma `linktrans(dh,q)`, es necesario indicar en el parámetro `dh` el tipo de cada articulación (rotacional o prismática) mediante el uso de  $\sigma_i$  tal como se mostró en el ejemplo 4.1.

## 4.4 MODELO INVERSO DEL MANIPULADOR

Como se ha mencionado anteriormente, en la mayor parte de las aplicaciones, interesa definir los movimientos del robot en el espacio cartesiano con relación a la tarea que se pretende desarrollar. Por consiguiente, el control del robot hace necesario obtener los valores de las variables articulares para que la posición y orientación del robot, o en particular de su efecto final, sea la deseada.

Se trata, por tanto, de resolver el problema inverso al del apartado anterior. Es decir, en la figura 4.1, se trata de calcular las variables articulares para situar  $\{H\}$  con respecto a un sistema de referencia  $\{P\}$ . El problema puede descomponerse en dos:

- 1) Determinar las transformaciones que permiten obtener  $\{M\}$ . De forma más precisa, suponiendo conocido un objetivo  $\{O\}$  se efectuarían las siguientes operaciones:
  - 1.1) Cálculo de la posición en la que debe estar la herramienta  $\{H\}$  conociendo el objetivo  $\{O\}$ .
  - 1.2) Cálculo de la posición de la muñeca  $\{M\}$  a partir de  $\{H\}$ .
- 2) Obtener los valores de las variables articulares conocida la posición y orientación en el espacio cartesiano definida por  $\{M\}$ . En el caso más general del robot móvil de la figura 4.1, el problema involucra tanto la consideración de la base móvil como de todas las articulaciones del manipulador. Sin embargo, normalmente, el problema se simplifica resolviéndolo secuencialmente: se obtendría en primer lugar la posición del vehículo tal que  $\{M\}$  sea alcanzable y a continuación se obtendrían las variables articulares del manipulador.

Nótese que la relación entre el sistema  $\{M\}$  y la base del manipulador viene dada por el modelo directo de éste. Se trata ahora de resolver el modelo inverso del manipulador

$$q = F \varphi^{-1}(p)$$

para obtener los valores de las variables articulares  $q = [q_1, q_2, \dots, q_n]^T$ .

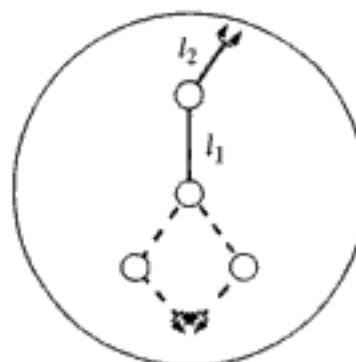
En lo que sigue se trata el problema planteado en el punto 2) para el manipulador. Nótese que se involucra la resolución de un sistema de ecuaciones no lineales. Así, en un robot manipulador con seis articulaciones rotacionales, tal como el Puma 500, la matriz  ${}^0T$  que define el modelo directo tiene 16 elementos, de los cuales cuatro son triviales (última fila de la matriz de transformación). Por consiguiente, se plantean doce ecuaciones no lineales para obtener los seis ángulos de las seis articulaciones. Las doce ecuaciones corresponden a tres de traslación y nueve de rotación de las cuales sólo tres son independientes. Es decir, el problema puede reducirse a uno con seis incógnitas y seis ecuaciones independientes no lineales con senos y cosenos tales como las que se involucran en el modelo directo.

La resolución de este problema hace necesario considerar, en primer lugar, la existencia de soluciones. En efecto, se trata de asegurar que la posición y orientación  $\{M\}$  puede o no ser alcanzada por la muñeca del operador; es decir, si está o no dentro de lo que se denomina *espacio de trabajo alcanzable*.

En los casos en que la localización objetivo no pertenece al espacio de trabajo alcanzable, se intentaría buscar una solución lo más "cercana" posible de la deseada  ${}^p_T$  con la cual se resolvería el modelo inverso. Obviamente, esta solución cercana debe ser libre de obstáculos y, en general, satisfacer las restricciones de la tarea.

De forma más precisa, en Craig (1986) se distingue entre espacio de trabajo alcanzable o volumen que puede ser alcanzado con el órgano terminal en al menos una orientación, y el *espacio de trabajo de manipulación con destreza* o volumen que puede ser alcanzado con el órgano terminal en *cualquier orientación*.

Nótese que la existencia o no de soluciones depende del número de articulaciones del manipulador. Así, en general, un manipulador con menos de seis grados de libertad no puede situar y orientar el órgano terminal en una posición cualquiera del espacio 3D. Considérese, por ejemplo, el manipulador plano con dos articulaciones de rotación que se muestra en la figura 4.5. Suponiendo que los enlaces tienen longitudes  $l_1$  y  $l_2$ , resulta evidente que el espacio alcanzable es un anillo de radio exterior  $l_1 + l_2$  y radio interior  $|l_1 - l_2|$ . Para el caso particular en que  $l_1 = l_2$  el espacio alcanzable es un disco de radio  $2l_1$ .



**Figura 4.5:** Espacio de trabajo de un manipulador plano con dos grados de libertad de rotación.

Obsérvese que a los puntos del interior se puede llegar con dos posibles orientaciones, mientras que a los puntos del límite del espacio de trabajo sólo con una. Si las longitudes de los enlaces son iguales,  $l_1 = l_2$ , el espacio de trabajo de manipulación con destreza es exclusivamente el origen, mientras que si estas longitudes son diferentes no existe espacio de trabajo de manipulación diestra.

En el ejemplo del párrafo anterior, se ha supuesto que las articulaciones pueden rotar 360 grados, lo que normalmente no será posible. En la práctica se establecen restricciones adicionales que limitan el espacio de trabajo. Suponiendo que en el ejemplo la primera articulación varía entre 0 y 360 grados mientras que la segunda sólo puede rotar entre 0 y 180°, el espacio alcanzable sería el mismo pero sólo podría llegarse con una orientación a cada punto.

El subespacio alcanzable de un manipulador de  $n$  grados de libertad es un subespacio del espacio de trabajo que puede expresarse como función de las  $n$  variables de las articulaciones. De esta forma, dando valores a las variables articulares en todo su rango se determinaría, mediante el modelo directo, el subespacio alcanzable del manipulador. Sin embargo si se pretende determinar las variables articulares que correspon-

den a cada punto del espacio alcanzable, o a una región de este, se necesita resolver el modelo inverso.

### Ejemplo 4.3

Considérese el manipulador del ejemplo 4.2. El sistema de referencia {0} coincide con el {1} para el valor de  $\theta_1 = 0$  que se muestra en la figura 4.4. Se trata de estudiar la posición  $(x, y)$  del extremo del segundo enlace. Por consiguiente, el giro de la tercera articulación (sistema de referencia {3}) no se considera. Supuesta la segunda articulación en su máxima extensión, el origen de {2} coincidirá con el de {3} en la figura 4.4. Las coordenadas  $x$  e  $y$  de este origen expresadas en el sistema {0} pueden tomar cualquier valor sólo limitado por los rangos de variación de las variables articulares. Sin embargo, la orientación del extremo según  ${}^0\hat{Z}_2$ , viene determinada en una dirección que depende de  $x$  e  $y$ . El vector unitario  ${}^0\hat{Z}_2$  viene dado por:

$${}^0\hat{Z}_2 = \begin{bmatrix} \frac{x}{\sqrt{x^2 + y^2}} \\ \frac{y}{\sqrt{x^2 + y^2}} \\ 0 \end{bmatrix} \quad (4.18)$$

Nótese que se dividen las coordenadas por  $\sqrt{x^2 + y^2}$  para que el módulo sea la unidad.

El vector  ${}^0\hat{Y}_2$  apunta siempre hacia arriba según la dirección de  $\hat{Z}_0$ :

$${}^0\hat{Y}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.19)$$

El vector  ${}^0\hat{X}_2$  puede calcularse como el producto vectorial:

$${}^0\hat{X}_2 = {}^0\hat{Y}_2 \times {}^0\hat{Z}_2 = \begin{bmatrix} \frac{-y}{\sqrt{x^2 + y^2}} \\ \frac{x}{\sqrt{x^2 + y^2}} \\ 0 \end{bmatrix} \quad (4.20)$$

La matriz de rotación es

$${}^0{}_2R = \begin{bmatrix} {}^0\hat{X}_2 & {}^0\hat{Y}_2 & {}^0\hat{Z}_2 \end{bmatrix} \quad (4.21)$$

Por consiguiente, la posición y orientación del extremo con respecto al origen viene dada por la transformación:

$${}^0_2T = \begin{bmatrix} \frac{-y}{\sqrt{x^2 + y^2}} & 0 & \frac{x}{\sqrt{x^2 + y^2}} & x \\ \frac{x}{\sqrt{x^2 + y^2}} & 0 & \frac{-y}{\sqrt{x^2 + y^2}} & y \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.22)$$

que define el subespacio alcanzable.

Por otra parte, el modelo directo desde la articulación 0 a la 2 viene dado por:  ${}^0_2T = {}^0_1T \cdot {}^1_2T$  donde  ${}^1_2T$  y  ${}^0_1T$  son las del ejemplo 4.2. Multiplicando estas dos matrices de transformación se obtiene:

$${}^0_2T = \begin{bmatrix} c_1 & 0 & s_1 & d_2s_1 \\ s_1 & 0 & -c_1 & -d_2c_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.23)$$

Dividiendo el elemento (2, 1) entre el (1, 1) en esta matriz y en la (4.22) e identificando ambos valores se tiene:

$$\theta_1 = \operatorname{arctg} \frac{x}{-y} \quad (4.24)$$

Asimismo, a partir de las ecuaciones (4.22) y (4.23), se deduce la variable  $d_2$ :

$$d_2 = \sqrt{x^2 + y^2} \quad (4.25)$$

En lo que sigue se aplica la herramienta MATLAB del apéndice para comprobar que los resultados obtenidos son coherentes. En primer lugar, se calcula el modelo directo del manipulador (ejemplo 4.1) con las siguientes líneas:

```

syms t1 d2 real; % Variables articulares theta_1 y d_2
dh = [0 0 t1 0 0;
      pi/2 0 0 d2 1]; % Matriz con los parametros de
                      % Denavit-Hartenberg (en este caso
                      % la matriz tiene solo dos filas
                      % porque en este ejemplo no
                      % consideramos el tercer enlace)
q = [t1 d2];
T = simple(fkine(dh,q)) % Vector de variables articulares

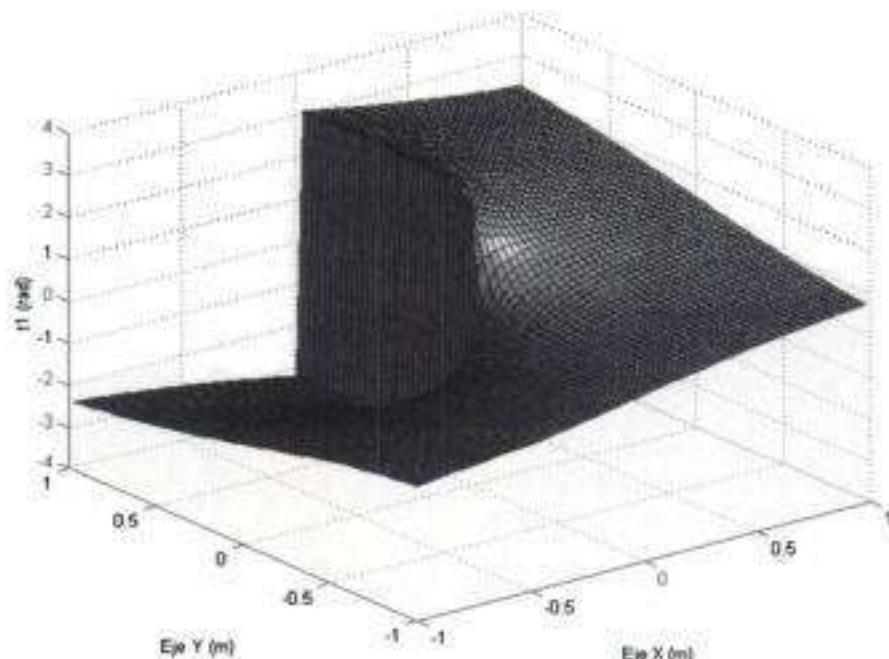
```

El resultado obtenido es la (4.23).

Para representar los valores que toma  $\theta_1$  en función de las coordenadas  $(x, y)$  se emplean las siguientes líneas en MATLAB:

```
[x,y] = meshgrid(-1:0.03:1,-1:0.03:1);
t1 = atan2(x,-y);
surf(x,y,t1); xlabel('Eje X (m)'), ylabel('Eje Y (m)'), zlabel('t1 (rad)')
```

El resultado obtenido es el que se muestra en la figura 4.6.



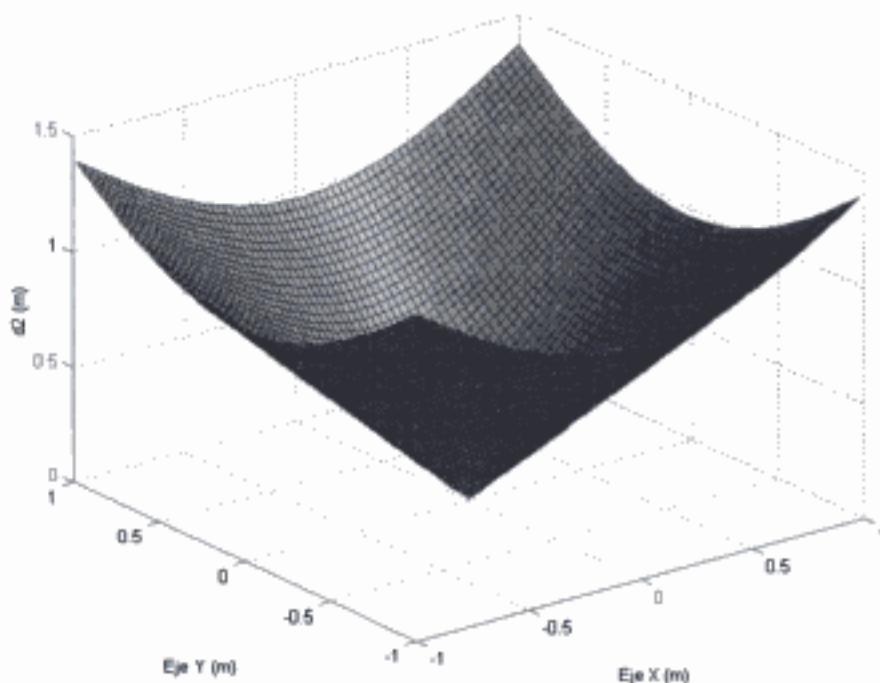
**Figura 4.6:** Representación de  $\theta_1$  en función del punto  $(x, y)$  del extremo del manipulador.

Hay que tener presente que realmente los puntos  $(x, y)$  alcanzables por este manipulador están en una circunferencia de radio 1 (y no en el rectángulo  $[-1, 1] \times [-1, 1]$ ). Por otro lado, en la gráfica puede apreciarse cómo los valores de  $\theta_1$  son los que cabría esperar. Así, por ejemplo, para el punto  $(0, -1)$  se tiene:  $\theta_1 = 0$ , y para el  $(1, 0)$  se tiene:  $\theta_1 = \frac{\pi}{2}$ , valores que eran de esperar sin mas que observar el manipulador.

Con respecto a la variable articular  $d_2$  puede escribirse:

```
[x,y]=meshgrid(-1:0.03:1,-1:0.03:1);
d2=sqrt(x.^2+y.^2);
surf(x,y,d2); xlabel('Eje X (m)'), ylabel('Eje Y (m)'), zlabel('d2 (m)')
```

resultando de su ejecución la gráfica que se muestra en la figura 4.7.



**Figura 4.7:** Representación de  $d_2$  en función del punto  $(x, y)$  del extremo del manipulador.

\* \* \*

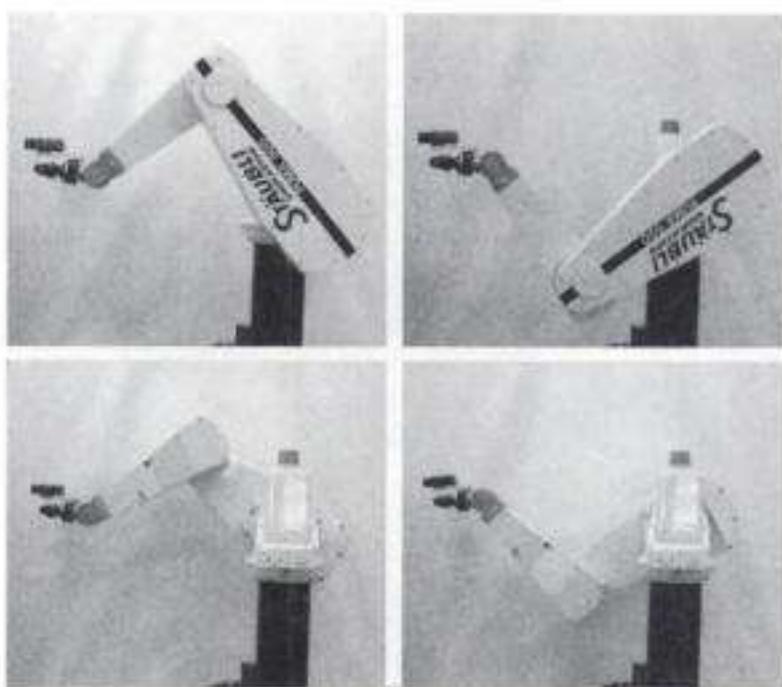
Otro aspecto del problema que es necesario estudiar con detalle es la existencia de múltiples soluciones del problema inverso. Es decir, diferentes conjuntos de valores de variables articulares permiten obtener la misma posición y orientación del extremo del manipulador en el espacio cartesiano. En general, existirán más soluciones cuanto mayor sea el número de articulaciones.

Así, por ejemplo, en robots manipuladores tales como el Puma 500 existen localizaciones que pueden ser alcanzadas con ocho soluciones diferentes. En la figura 4.8 se muestran cuatro soluciones. Para cada solución existe otra con las últimas tres articulaciones cambiadas según:

$$\theta'_4 = \theta_4 + 180^\circ; \quad \theta'_5 = -\theta_5; \quad \theta'_6 = \theta_6 + 180^\circ$$

En general, cuantos más parámetros de Denavit-Hartenberg sean diferentes de cero, existirán más soluciones distintas. Para un manipulador con seis articulaciones de rotación existirían 16 soluciones diferentes. Sin embargo, en la práctica, es necesario tener en cuenta el rango de variación de las variables, ya que, debido a límites físicos, algunas soluciones puede que no sean accesibles.

La existencia de múltiples soluciones hace necesario establecer criterios que permitan escoger la solución más idónea entre las diferentes posibles. Entre estos criterios está el que minimiza los movimientos articulares desde la posición actual. Conviene tener en cuenta también que no todos los movimientos articulares pueden ser igualmente aconsejables. Por consiguiente, en la mencionada minimización pueden tenerse en cuenta criterios energéticos, de fiabilidad, etc. Así, por ejemplo, se trataría de mover menos los enlaces de mayor longitud.



**Figura 4.8:** Cuatro diferentes configuraciones de un robot angular dan lugar a la misma posición y orientación del extremo.

Obviamente, es necesario tener en cuenta los *obstáculos* que pueden producir colisiones en el movimiento del robot. Por tanto, se trata de escoger la solución más cercana, dentro del conjunto de soluciones cuyo movimiento desde la posición actual es libre de colisión. Para ello es necesario el conocimiento previo del entorno del robot. Desde un punto de vista práctico, es muy difícil mantener un modelo actualizado del entorno que pueda ser consultado en tiempo real por el controlador del robot cada vez que tenga que resolver el modelo inverso. Por consiguiente, la gran mayoría de los sistemas de control de robots industriales no tienen esta capacidad y escogen las soluciones del modelo inverso sin tener en cuenta en tiempo real el modelo del entorno.

No obstante, en aplicaciones en las que se requiere una gran fiabilidad y pueden generarse movimientos variados, resulta imprescindible efectuar las mencionadas comprobaciones. En Ollero y otros (1995) se presenta un sistema de simulación predictiva que, en tiempo real, verifica las propiedades de las soluciones teniendo en cuenta el modelo del robot, los rangos de variación de sus variables articulares, y un modelo simplificado del entorno que puede consultarse fácilmente.

Desde un punto de vista teórico se sabe que es posible resolver el modelo inverso de todo sistema con articulaciones prismáticas y de rotación que suministren un total de seis grados de libertad en una cadena cinemática simple. Sin embargo, las consideraciones antes mencionadas sobre los parámetros de la estructura y los rangos de variación admisible de las variables articulares pueden hacer que no sea posible alcanzar una solución teórica.

Conviene mencionar que los métodos numéricos para la resolución general del problema cinemático inverso de un manipulador son normalmente de convergencia más lenta que la admisible en el control en tiempo real de manipuladores robóticos.

Por tanto, normalmente se prefieren soluciones analíticas que permitan obtener directamente los valores de las variables articulares.

En general, no puede asegurarse que sea posible resolver analíticamente cualquier robot con seis grados de libertad. No obstante, se han desarrollado métodos de resolución tales como el de Pieper (1968) para ciertas clases de manipuladores como los que tienen tres ejes consecutivos que intersectan en un punto o son paralelos. Esta situación se da en numerosos manipuladores industriales tales como el Puma, en el cual las articulaciones 4, 5 y 6 intersectan en un punto.

Los métodos algebraicos de resolución del modelo inverso se basan en especificar una transformación  ${}^B_M T$  que define un subespacio en el que deben estar todas las soluciones alcanzables.

A continuación, se obtienen analíticamente expresiones que resuelven las ecuaciones que resultan de igualar  ${}^B_M T$  con las expresiones cinemáticas generales del manipulador  ${}^0_N T$ . Es necesario comprobar la existencia de soluciones. Como se ha mencionado anteriormente, podrán existir diferentes soluciones que satisfacen las ecuaciones. Estas soluciones corresponden normalmente a expresiones alternativas con diferente signo en las ecuaciones.

Otra técnica alternativa para resolver el modelo inverso consiste en estudiar geométricamente el manipulador en cuestión para formular las ecuaciones. Normalmente el estudio en el espacio de tres dimensiones se trata mediante diversas formulaciones en el plano. El problema se reduce a la resolución de triángulos. La existencia de soluciones y la elección entre múltiples soluciones alternativas se realiza también teniendo en cuenta consideraciones geométricas. La complejidad de este método geométrico depende de las características de la estructura cinemática. Así, cuando los parámetros  $\alpha$  toman como valores  $0, +90^\circ, -90^\circ$ , el problema se simplifica.

#### Ejemplo 4.4

Considérese el manipulador del ejemplo 4.1. La especificación de la muñeca relativa a la base viene dada por el modelo directo  ${}^B_M T$  (ejemplo 4.1):

$${}^B_M T = {}^0_3 T = \begin{bmatrix} c_{123} & -s_{123} & 0 & l_1 c_1 + l_2 c_{12} \\ s_{123} & c_{123} & 0 & l_1 s_1 + l_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.26)$$

donde las expresiones de  $c_{12}$ ,  $s_{12}$ ,  $c_{123}$  y  $s_{123}$  vienen dadas por (4.11), (4.12), (4.13) y (4.14) respectivamente.

En un manipulador plano, la especificación de la posición y orientación deseadas en el extremo del manipulador se realiza indicando tres números  $x$ ,  $y$ ,  $\phi$ , siendo los dos primeros las coordenadas en el plano del extremo del manipulador (origen del sistema de referencia {3}) y  $\phi$  su orientación con relación al eje  $+X$ . Por tanto, en vez de considerar como dato una matriz general  ${}^B_M T$ , se supondrá una transformación con la estructura (Craig, 1986):

$${}^B_M T = \begin{bmatrix} c\phi & -s\phi & 0 & x \\ s\phi & c\phi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.27)$$

siendo  $s\phi$  y  $c\phi$  el seno y el coseno del ángulo de orientación  $\phi$  respectivamente.

Igualando los elementos no triviales de las matrices (4.26) y (4.27) (diferentes de 0 y 1) se obtienen cuatro ecuaciones en las cuales existen tres incógnitas  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ . Nótese como dos ecuaciones corresponden a las coordenadas  $x$  e  $y$ :

$$\begin{aligned} x &= l_1 c_1 + l_2 (c_1 c_2 - s_1 s_2) \\ y &= l_1 s_1 + l_2 (c_1 s_2 + s_1 c_2) \end{aligned} \quad (4.28)$$

Las dos restantes corresponden a la orientación y pueden resumirse en

$$\theta_1 + \theta_2 + \theta_3 = \phi = \operatorname{arctg} 2(s\phi, c\phi) \quad (4.29)$$

donde la función  $\operatorname{arctg} 2$  se define como en el capítulo anterior (arco tangente teniendo en cuenta el signo de los argumentos).

Para obtener las ecuaciones que resuelven el modelo inverso se elevan al cuadrado las ecuaciones (4.28) y se suman, con lo cual se obtiene:

$$x^2 + y^2 = l_1^2 + l_2^2 + 2l_1 l_2 c_2 \quad (4.30)$$

Despejando  $c_2$  se obtiene

$$c_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2} \quad (4.31)$$

Para que exista una solución, el lado derecho de la ecuación debe dar un valor entre  $-1$  y  $+1$ . Si no se cumple esta condición, el punto objetivo está demasiado lejos para que lo alcance el manipulador. Suponiendo que existe solución, para determinar  $\theta_2$  teniendo en cuenta el signo, se hace

$$s_2 = \pm \sqrt{1 - c_2^2} \quad (4.32)$$

de donde puede calcularse  $\theta_2$  mediante:

$$\theta_2 = \operatorname{arctg} 2(s_2, c_2) \quad (4.33)$$

Dependiendo del signo de  $s_2$  (positivo o negativo), se tiene una solución diferente (arriba o abajo). Una vez que se conoce  $\theta_2$ , la posición y orientación pueden calcularse mediante:

$$x = (l_1 + l_2 c_2) c_1 - (l_2 s_2) s_1 \quad (4.34)$$

$$y = (l_1 + l_2 c_2) s_1 + (l_2 s_2) c_1 \quad (4.35)$$

Dividiendo (4.35) entre (4.34) se obtiene:

$$\frac{y}{x} = \frac{(l_1 + l_2 c_2)s_1 + (l_2 s_2)c_1}{(l_1 + l_2 c_2)c_1 - (l_2 s_2)s_1}. \quad (4.36)$$

Suponiendo  $c_1$  diferente de cero, si se divide el lado derecho de (4.36) por  $c_1$  se obtiene:

$$\frac{y}{x} = \frac{(l_1 + l_2 c_2)\operatorname{tg} \theta_1 + (l_2 s_2)}{(l_1 + l_2 c_2) - (l_2 s_2)\operatorname{tg} \theta_1} = \frac{\operatorname{tg} \theta_1 + \frac{l_2 s_2}{l_1 + l_2 c_2}}{1 - \frac{l_2 s_2}{l_1 + l_2 c_2}\operatorname{tg} \theta_1} \quad (4.37)$$

Nótese que el lado derecho de (4.37) coincide con la expresión de la tangente de la suma de dos ángulos, por lo cual puede escribirse:

$$\theta_1 = \operatorname{arctg} 2(y, x) - \alpha \quad (4.38)$$

siendo

$$\alpha = \operatorname{arctg} 2(l_2 s_2, l_1 + l_2 c_2) \quad (4.39)$$

Nótese que al escoger el signo de  $\theta_2$  también se afecta a  $\theta_1$ . Obsérvese además que, cuando  $x = y = 0$ , no puede determinarse el lado derecho de (4.38), por lo que  $\theta_1$  es arbitrario.

Finalmente, conociendo  $\theta_1$  y  $\theta_2$ , se calcula  $\theta_3$  mediante la (4.29).

Se trata ahora de resolver el mismo manipulador plano con tres articulaciones de rotación, pero empleando ahora consideraciones geométricas (Craig, 1986). Aplicando relaciones trigonométricas en los triángulos de la figura 4.9 se obtiene:

$$x^2 + y^2 = l_1^2 + l_2^2 - 2l_1 l_2 \cos(180^\circ + \theta_2) = l_1^2 + l_2^2 + 2l_1 l_2 \cos \theta_2 \quad (4.40)$$

Por tanto, puede despejarse

$$\cos \theta_2 = c_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2} \quad (4.41)$$

El algoritmo debe comprobar que la distancia  $\sqrt{x^2 + y^2}$  sea menor o igual que

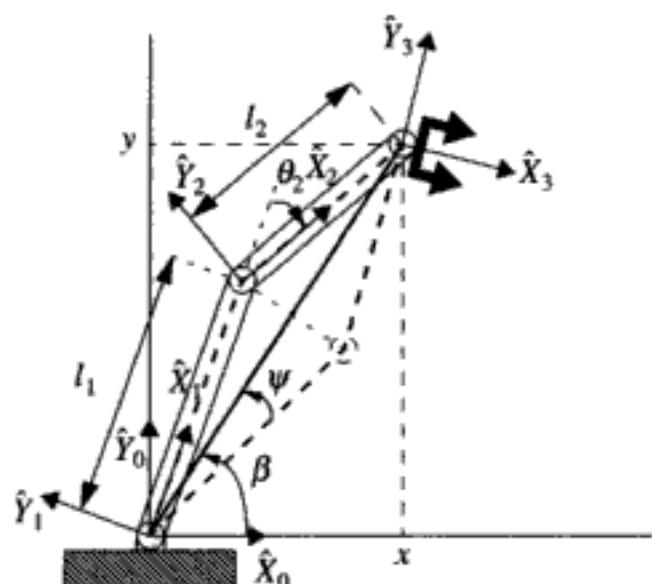


Figura 4.9: Resolución mediante métodos geométricos.

$l_1 + l_2$  antes de aplicar las expresiones. Suponiendo que existe solución, se resuelve para un valor de  $\theta_2$  entre  $0$  y  $-180^\circ$  ya que el triángulo sólo existe para estos valores.

La otra posible solución (línea discontinua en la parte inferior) corresponde a  $+\theta_2$ .

Para obtener  $\theta_1$ , se encuentran las expresiones de los ángulos  $\beta$  y  $\psi$ . Nótese que  $\beta = \text{arctg } 2(y, x)$ . Por otra parte, aplicando la ley de cosenos se obtiene:

$$l_2^2 = l_1^2 + (x^2 + y^2) - 2l_1\sqrt{x^2 + y^2} \cos \psi$$

de donde

$$\cos \psi = \frac{x^2 + y^2 + l_1^2 - l_2^2}{2l_1\sqrt{x^2 + y^2}} \quad (4.42)$$

El arco coseno se resuelve para  $0 \leq \psi \leq 180^\circ$ . Finalmente se hace

$$\theta_1 = \beta \pm \psi \quad (4.43)$$

aplicando el signo  $+$  si  $\theta_2 < 0$  y el signo  $-$  si  $\theta_2 > 0$ .

Por último, teniendo en cuenta

$$\theta_1 + \theta_2 + \theta_3 = \phi \quad (4.44)$$

se obtiene  $\theta_3$ .

\* \* \*

Una vez planteadas las ecuaciones por métodos algebraicos o geométricos, es importante efectuar simplificaciones que minimicen el número de operaciones que es necesario efectuar en tiempo real.

En efecto, nótese que el controlador del robot debe resolver el modelo cinemático inverso en períodos que son típicamente de algunas decenas de milisegundos. Por consiguiente, las técnicas que permiten reducir el tiempo de procesamiento son particularmente interesantes.

De esta manera, se ordenan las ecuaciones de forma que a partir de una solución se simplifique la obtención de otras, por ejemplo mediante la simple suma o resta de un ángulo.

Una técnica para acelerar el cálculo es la simple tabulación de la función arco tangente, que normalmente aparece en las expresiones del modelo cinemático inverso.

Por último, cabe mencionar que el modelo inverso puede resolverse también empleando métodos numéricos. No obstante, estos métodos tienen frecuentemente problemas de convergencia. En la herramienta que se describe en el apéndice, se presenta una función (*ikine*) que permite la aplicación de un método numérico poniendo de manifiesto los problemas de convergencia en la resolución del modelo inverso del manipulador del ejemplo 4.4.

## 4.5 VELOCIDADES LINEALES Y ANGULARES

Se trata ahora de relacionar las velocidades del efecto final en el espacio cartesiano con las velocidades del movimiento de las articulaciones. Para ello se comienza por tratar la expresión de las velocidades lineales y angulares de acuerdo con la notación introducida en los apartados precedentes. A continuación se trata la propagación de las velocidades en las articulaciones. En todo este apartado se empleará también la notación y desarrollos de Craig (1986).

### 4.5.1 Velocidad lineal

Se estudia en primer lugar la *velocidad lineal o de traslación* de un punto  ${}^BQ$  con respecto al sistema fijo  $\{A\}$ , suponiendo conocidas su velocidad lineal  ${}^BV_Q$  con respecto al  $\{B\}$  y la velocidad lineal  ${}^AV_{ORGB}$  del origen del sistema  $\{B\}$  con respecto al  $\{A\}$ . La situación se ilustra en la figura 4.10.

Supóngase que la localización  $\{B\}$  relativa a  $\{A\}$  está descrita por un vector de posición  ${}^AP_{ORGB}$  y una matriz de rotación  ${}^BR$ . La velocidad  ${}^BV_Q$  del punto  $Q$  en el sistema de referencia  $\{B\}$  se define como

$${}^BV_Q = \frac{d}{dt} {}^BQ = \lim_{\Delta t \rightarrow 0} \frac{{}^BQ(t + \Delta t) - {}^BQ(t)}{\Delta t} \quad (4.45)$$

expresión en la cual la derivada se toma con respecto al sistema de referencia  $\{B\}$ . Suponiendo que el origen de  $\{B\}$  se mantiene fijo con respecto al  $\{A\}$ , el vector velocidad puede expresarse en el sistema de referencia  $\{A\}$  como:

$${}^A({}^B V_Q) = {}^A \left( \frac{d}{dt} {}^B Q \right) = {}^A {}_B R {}^B V_Q \quad (4.46)$$

Si ahora se considera que el origen del sistema {B} se mueve con respecto al sistema {A} con una velocidad  ${}^A V_{ORG_B}$ , y se supone que la orientación relativa de {B} con respecto a {A} no varía ( ${}^A {}_B R$  constante), la velocidad lineal del punto  $Q$  con respecto al sistema {A} viene dada por:

$${}^A V_Q = {}^A V_{ORG_B} + {}^A {}_B R {}^B V_Q \quad (4.47)$$

#### 4.5.2 Velocidad angular

Supóngase ahora que los sistemas {A} y {B} mantienen sus orígenes coincidentes en el tiempo (la velocidad de traslación del {B} con respecto al {A} es nula), pero el sistema {B} está rotando con respecto al {A}. Se trata de estudiar la velocidad del punto  $Q$  con respecto al sistema {A}, conociendo su velocidad lineal  ${}^B V_Q$  con respecto al sistema {B} y la *velocidad angular o de rotación*  ${}^A \Omega_B$  del sistema {B} con respecto al {A}. La situación se muestra en la figura 4.10.

La velocidad angular o de rotación de un cuerpo rígido se expresa como el movimiento de rotación de un sistema de referencia solidario al cuerpo. La dirección de la velocidad angular  ${}^A \Omega_B$  del sistema {B} con respecto al {A} expresada en el sistema {A}, proporciona el eje instantáneo de rotación y su magnitud la velocidad de rotación.

Considérese el movimiento de un punto  $Q$  debido a la rotación, tal como se muestra en la figura 4.10 (Craig, 1986). El cambio diferencial en  ${}^A Q$  es perpendicular tanto a  ${}^A \Omega_B$  como a  ${}^A Q(t)$  y su magnitud viene dada por

$$|\Delta Q| = (|{}^A Q| \sin \theta) (|{}^A \Omega_B| \Delta t) \quad (4.48)$$

Por consiguiente, conociendo la posición  ${}^B Q$  del punto en el sistema {B}, la velocidad de  $Q$  visto desde {A} debido a la rotación  ${}^A \Omega_B$  puede expresarse mediante el producto vectorial:

$${}^A V_Q = {}^A \Omega_B \times {}^A Q = {}^A \Omega_B \times {}^A {}_B R {}^B Q \quad (4.49)$$

Por otra parte, el cambio del vector  $Q$  con respecto a {B} viene dado por la velocidad  ${}^B V_Q$  que expresada en el sistema {A} es  ${}^A {}_B R {}^B V_Q$ .

Por tanto, la velocidad resultante del cambio del vector  $Q$  con respecto a {B} y de la rotación de {B} con respecto a {A}, expresada en {A}, es:

$${}^A V_Q = {}^A {}_B R {}^B V_Q + {}^A \Omega_B \times {}^A {}_B R {}^B Q \quad (4.50)$$

Considérese ahora la *combinación de la velocidad de traslación con la de rotación*. Teniendo en cuenta (4.47) y (4.50) la velocidad resultante es:

$${}^A V_Q = {}^A V_{ORG_B} + ({}^A {}_B R {}^B V_Q + {}^A \Omega_B \times {}^A {}_B R {}^B Q) \quad (4.51)$$

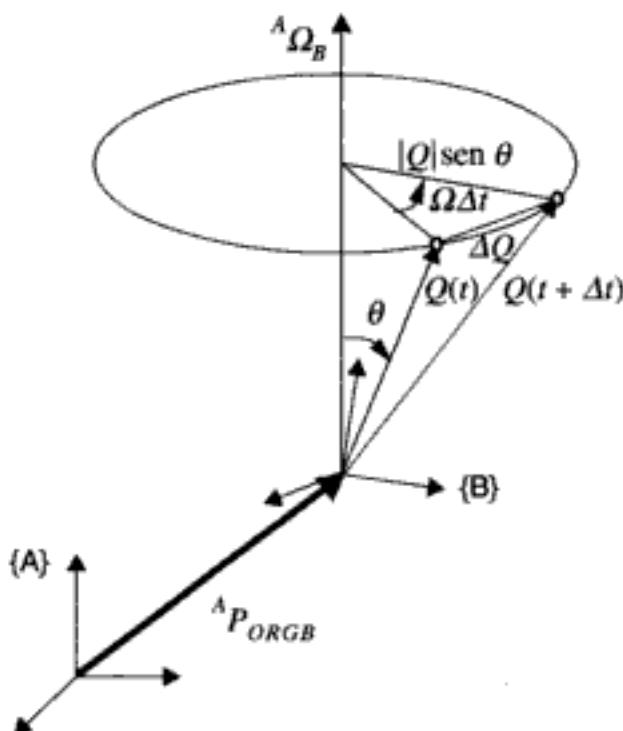


Figura 4.10: Velocidad de traslación y de rotación.

#### 4.5.3 Propagación de velocidades

Se trata ahora de estudiar la propagación de las velocidades por las articulaciones de un robot. Los enlaces entre dos articulaciones se suponen cuerpos rígidos con movimiento descrito por vectores de velocidades lineales y angulares.

Sea  $\{0\}$  la articulación que se toma como referencia. La velocidad lineal del origen del sistema  $\{i\}$  se representará por  $v_i$  y la velocidad angular de este sistema por  $\omega_i$ . La velocidad del enlace  $i+1$  es la del  $i$  más los componentes que añade la articulación  $i+1$ . En lo que sigue se considera en, primer lugar, la propagación de velocidades en las articulaciones de rotación y a continuación en las de traslación. En la figura 4.11 se representan los vectores velocidad.

Se supondrá, en primer lugar, que la articulación considerada es de *rotación*. Si el eje de esta rotación coincide con el eje  $\hat{Z}$ , la expresión de la *velocidad angular* del enlace  $i+1$  en términos del sistema en  $\{i\}$  es

$${}^i\omega_{i+1} = {}^i\omega_i + {}_{i+1}R \ {}^{i+1}\dot{\theta}_{i+1} \hat{Z}_{i+1} \quad (4.52)$$

siendo  $\dot{\theta}_{i+1}$  la derivada con respecto al tiempo de la variable articular que, en forma vectorial, se representa como:

$${}^{i+1}\dot{\theta}_{i+1} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \theta'_{i+1} \end{bmatrix} \quad (4.53)$$

y  ${}^{i+1}R$  la matriz que rota el eje de  $\{i+1\}$  a su descripción en  $\{i\}$  para poder sumar las dos componentes de velocidades angulares.

Asimismo, la *velocidad angular* del enlace  $i+1$  expresado en  $\{i+1\}$  es:

$${}^{i+1}\omega_{i+1} = {}^{i+1}R \cdot {}^i\omega_i + \theta'_{i+1} {}^{i+1}\hat{Z}_{i+1} \quad (4.54)$$

Por otra parte, también en la articulación de rotación, la *velocidad lineal* del origen de  $\{i+1\}$  es la del origen de  $\{i\}$  más una componente originada por la velocidad angular de  $i$ :

$${}^i v_{i+1} = {}^i v_i + {}^i \omega_i \times {}^i P_{i+1} \quad (4.55)$$

siendo  ${}^i P_{i+1}$  el vector que expresa la posición del origen de  $\{i+1\}$  en el sistema  $\{i\}$  (figura 4.11).

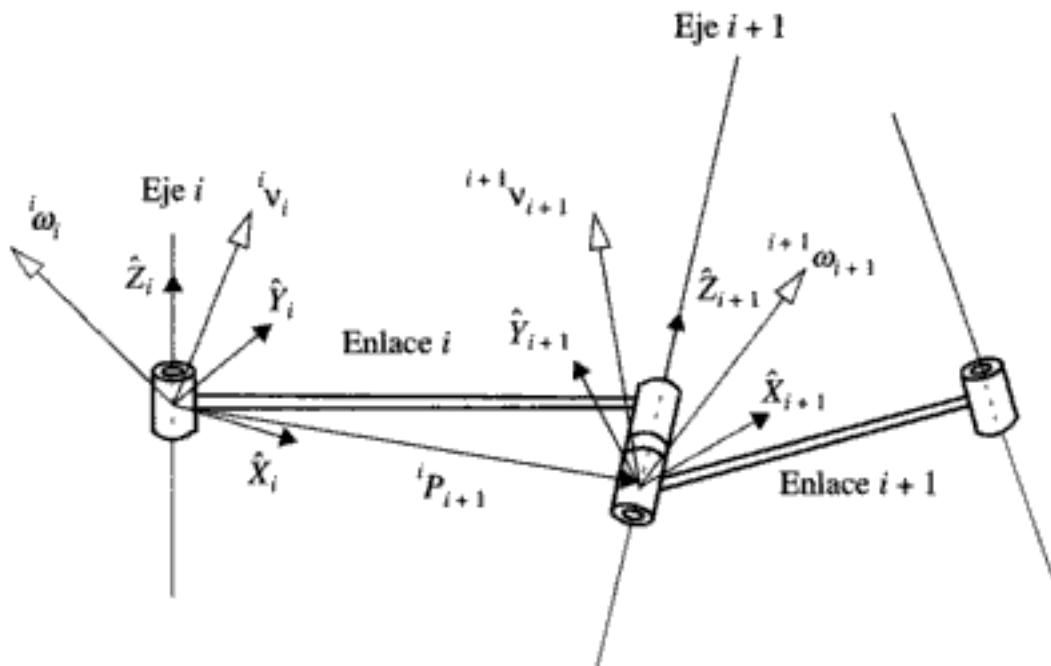


Figura 4.11: Vectores velocidad de dos articulaciones consecutivas.

La expresión resultante de la velocidad con respecto al sistema  $\{i+1\}$  es:

$${}^{i+1}v_{i+1} = {}^{i+1}R({}^i v_i + {}^i \omega_i \times {}^i P_{i+1}) \quad (4.56)$$

Se consideran ahora *articulaciones de traslación*. En este caso, las velocidades lineales y angulares son:

$${}^{i+1}\omega_{i+1} = {}^{i+1}R \cdot {}^i\omega_i \quad (4.57)$$

$${}^{i+1}v_{i+1} = {}^iR({}^iv_i + {}^i\omega_i \times {}^iP_{i+1}) + d_{i+1}{}^{i+1}\hat{Z}_{i+1} \quad (4.58)$$

Suponiendo  ${}^0\omega_0 = 0$ ,  ${}^0v_0 = 0$  la aplicación sucesiva de las expresiones (4.54), (4.56) (articulaciones de rotación) y (4.57), (4.58) (articulaciones de traslación) permite propagar las velocidades desde la  $i = 1$  hasta la  $i = N$  obteniendo finalmente  ${}^N\omega_N$ ,  ${}^Nv_N$ . Para referirlas a la base se multiplica por  ${}^0R$ .

## 4.6 JACOBIANO DEL MANIPULADOR

El jacobiano del manipulador relaciona las velocidades articulares con las velocidades cartesianas del extremo:

$${}^0v = {}^0J(q) \cdot q' \quad (4.59)$$

expresión en la cual se ha tomado como referencia el sistema {0}. Obsérvese como el Jacobiano se expresa en un determinado sistema de referencia. Asimismo, nótese como, para unos determinados valores de las variables articulares, la relaciones entre las velocidades articulares y las velocidades en el espacio cartesiano son lineales.

El vector de velocidades en el espacio cartesiano puede escribirse como:

$${}^0v = \begin{bmatrix} {}^0v \\ {}^0\omega \end{bmatrix} \quad (4.60)$$

siendo  ${}^0v$  un vector  $3 \times 1$  de velocidades lineales, y  ${}^0\omega$  un vector  $3 \times 1$  de velocidades rotacionales.

En un robot con seis articulaciones,  $q'$  es un vector  $6 \times 1$  de velocidades de las articulaciones. Por consiguiente, en este caso,  $J$  es una matriz  $6 \times 6$ .

En general, el número de columnas de  $J$  es el número de articulaciones y el número de filas el de grados de libertad.

El jacobiano no es necesariamente cuadrado. En efecto, en robots redundantes, el número de columnas es mayor que el de filas.

Existen diferentes métodos para obtener el jacobiano de un determinado robot. El más evidente es el de la obtención analítica de expresiones de las velocidades cartesianas en función de las velocidades articulares. Para ello, puede aplicarse la técnica de propagación de las velocidades expuesta anteriormente.

La obtención de las velocidades articulares en función de las velocidades cartesianas en el extremo involucra la consideración de la inversa del jacobiano. Si la matriz es no singular puede escribirse:

$$q' = J^{-1}(q) \cdot v \quad (4.61)$$

Sin embargo, es frecuente que existan valores de  $q$  para los que el jacobiano es singular. Estas singularidades se presentan en los límites del espacio de trabajo, o en su interior cuando dos o más ejes de articulaciones están alineados.

En las singularidades se produce pérdida de grados de libertad en el espacio cartesiano; es decir, direcciones o subespacios en los que no es posible mover el órgano terminal. Se recuerda, por ejemplo, el caso en que los ejes de la cuarta y la sexta articulación del robot Puma están alineados (capítulo 2).

### Ejemplo 4.5

Considérese otra vez el manipulador plano con tres articulaciones de rotación de los ejemplos 4.1 y 4.4 pero, en este caso, con los ejes del marco de referencia {3} en la misma dirección que los del {2}. Se desean encontrar la velocidad en el origen del sistema {3} y el jacobiano del manipulador.

Las matrices de rotación de las transformaciones son:

$${}^0R = \begin{bmatrix} c_1 & s_1 & 0 \\ -s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.62)$$

$${}^1R = \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.63)$$

$${}^2R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.64)$$

Suponiendo que {0} tiene velocidad nula, y propagando desde esta articulación empleando las expresiones (4.54) y (4.56) se obtiene:

$$\begin{aligned} {}^1\omega_1 &= {}^0R \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \theta'_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \theta'_1 \end{bmatrix}; {}^1v_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ {}^2\omega_2 &= {}^1R \cdot \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \theta'_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \theta'_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \theta'_1 + \theta'_2 \end{bmatrix} \end{aligned} \quad (4.65)$$

Si  ${}^1P_2$  es el vector desde el origen del sistema {1} al del sistema {2} expresado en el sistema {1} se tiene:

$${}^1\omega_1 \times {}^1P_2 = \begin{bmatrix} 0 \\ 0 \\ \theta'_1 \end{bmatrix} \times \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ l_1\theta'_1 \\ 0 \end{bmatrix} \quad (4.66)$$

$${}^2v_2 = \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left( \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ l_1\theta'_1 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} l_1s_2\theta'_1 \\ l_1c_2\theta'_1 \\ 0 \end{bmatrix} \quad (4.67)$$

$${}^3\omega_3 = {}^2\omega_2 \quad (4.68)$$

Si ahora  ${}^2P_3$  es el vector desde el origen del sistema {2} al del sistema {3} expresado en el sistema {2} se tiene:

$${}^2\omega_2 \times {}^2P_3 = \begin{bmatrix} 0 \\ l_2(\theta'_1 + \theta'_2) \\ 0 \end{bmatrix} \quad (4.69)$$

$${}^3v_3 = \begin{bmatrix} l_1s_2\theta'_1 \\ l_1c_2\theta'_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ l_2(\theta'_1 + \theta'_2) \\ 0 \end{bmatrix} = \begin{bmatrix} l_1s_2\theta'_1 \\ l_1c_2\theta'_1 + l_2(\theta'_1 + \theta'_2) \\ 0 \end{bmatrix} \quad (4.70)$$

Por tanto, el jacobiano expresado en {3} es:

$${}^3J(\theta) = \begin{bmatrix} l_1s_2 & 0 \\ l_1c_2 + l_2 & l_2 \end{bmatrix} \quad (4.71)$$

La velocidad del extremo expresada con respecto al sistema {0} es

$${}^0v_3 = {}^0R {}^3v_3 = {}^0R {}^1R {}^2R {}^3v_3 = \begin{bmatrix} -(l_1s_1\theta'_1 + l_2s_{12}(\theta'_1 + \theta'_2)) \\ l_1c_1\theta'_1 + l_2c_{12}(\theta'_1 + \theta'_2) \\ 0 \end{bmatrix} \quad (4.72)$$

Por consiguiente, el jacobiano expresado en {0}, tal como está definido en (4.59), es:

$${}^0J(\theta) = \begin{bmatrix} -(l_1s_1 + l_2s_{12}) & -l_2s_{12} \\ l_1c_1 + l_2c_{12} & l_2c_{12} \end{bmatrix} \quad (4.73)$$

La inversa del jacobiano es

$${}^0J^{-1}(\theta) = \frac{1}{l_1 l_2 s_2} \begin{bmatrix} l_2 c_{12} & l_2 s_{12} \\ -(l_1 c_1 + l_2 c_{12}) & -(l_1 s_1 + l_2 s_{12}) \end{bmatrix} \quad (4.74)$$

Obsérvese como existe una singularidad para  $\theta_2 = 0$  que corresponde al brazo completamente estirado (límite del espacio de trabajo) o bien cuando  $\theta_2 = 180^\circ$  (brazo plegado sobre sí mismo).

A continuación se realiza el estudio empleando las funciones de la herramienta MATLAB que se presenta en el apéndice. En particular, se hace uso de la función `jacobn(dh, q)` que corresponde a la aplicación de un método alternativo para la obtención del jacobiano del manipulador (Paul, 1981). Para aplicar esta función se emplean las instrucciones:

```

syms t1 t2 t3 real; % Variables articulares θ1, θ2 y θ3
syms l1 l2 real; % Parámetros l1 y l2
% Matriz con los parámetros de Denavit-Hartenberg del robot del Ejemplo 4.1
dh = [0 0 t1 0 0;
      0 l1 t2 0 0;
      0 l2 t3 0 0];
q = [t1 t2 t3];
J = simple(jacobn(dh,q)) % Jacobiano expresado en {3}
J = simple(jacob0(dh,q)) % Jacobiano expresado en {0}

```

cuya ejecución proporciona como resultados las siguientes matrices:

$${}^3J(\theta) = \begin{bmatrix} l_1 s_{23} + l_2 s_3 & l_2 s_3 & 0 \\ l_1 c_{23} + l_2 c_3 & l_2 c_3 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (4.75)$$

$${}^0J(\theta) = \begin{bmatrix} -(l_1 s_1 + l_2 s_{12}) & -l_2 s_{12} & 0 \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (4.76)$$

Con el fin de comprobar que los resultados alcanzados con este algoritmo son correctos, se aplica la función MATLAB denominada *velprop* que emplea la técnica tradicional de propagación de velocidades descrita anteriormente. El listado de dicha función es el siguiente:

```
function z = velprop(dh,q,qd,v0,w0)

n = numrows(dh);
for i=1:n
    T = linktrans(dh(i,:), q(i));
    if dh(i,5) == 0
        % articulación de rotación
        w = T(1:3,1:3)' *w0+qd(i)*[0 0 1]';
        v = T(1:3,1:3)' *(v0+cross(w0,T(1:3,4)));
        w0 = w;
        v0 = v;
    else
        % articulación prismática
        w = T(1:3,1:3)' *w0;
        v = T(1:3,1:3)' *(v0+cross(w0,T(1:3,4)))+qd(i)*[0 0 1]';
        w0 = w;
        v0 = v;
    end
end
z = [v;w];
```

Esta función devuelve un vector de la forma " $v = \begin{bmatrix} v \\ w \end{bmatrix}^T$ ", a partir del cual es inmediata la obtención del jacobiano, ya que:

$$\dot{v} = \dot{J}(q) \cdot \dot{q} \quad (4.77)$$

Para aplicar la función *velprop* al manipulador del ejemplo 4.1 se usa el siguiente listado:

```
syms t1 t2 t3 real; % Variables articulares  $\theta_1$ ,  $\theta_2$  y  $\theta_3$ 
syms td1 td2 td3 real; % Derivadas de las variables articulares
                        (%  $\dot{\theta}_1$ ,  $\dot{\theta}_2$  y  $\dot{\theta}_3$ )
syms l1 l2 real; % Parámetros  $l_1$  y  $l_2$ 

% Matriz con los parámetros de Denavit-Hartenberg del robot del Ejemplo 4.1
dh = [ 0 0 t1 0 0;
       0 l1 t2 0 0;
       0 l2 t3 0 0];
```

```

q = [t1 t2 t3];           % Vector de variables articulares
qd = [td1 td2 td3];       % Vector con las derivadas de las variables articulares
v0 = [0 0 0]';            % Vector velocidad lineal del {0}
w0 = [0 0 0]';            % Vector velocidad angular del {0}
z = simple(velprop(dh,q,qd,v0,w0))

```

El vector resultante de aplicar estas instrucciones es:

$$\begin{bmatrix} {}^3v \\ {}^3\omega \end{bmatrix} = \begin{bmatrix} (l_1 s_{23} + l_2 s_3) \theta'_1 + l_2 s_3 \theta'_2 \\ (l_1 c_{23} + l_2 c_3) \theta'_1 + l_2 c_3 \theta'_2 \\ 0 \\ 0 \\ 0 \\ \theta'_1 + \theta'_2 + \theta'_3 \end{bmatrix} \quad (4.78)$$

del que se puede extraer de forma inmediata  ${}^3J(\theta)$  en virtud de la ecuación (4.77) comprobándose que el resultado coincide con (4.75).

#### Ejemplo 4.6

En este caso se aplican las funciones `velprop` y `jacobn` al manipulador del ejemplo 4.2:

Utilizando `velprop`:

```

syms t1 d2 t3 real;          % Variables articulares θ₁, d₂ y θ₃
syms td1 dd2 td3 real;      % Derivadas de las variables articulares
                             (θ'₁, d'₂ y θ'₃)
syms l2 real;                % Parámetro l₂
                             % Matriz con los parámetros de Denavit-Hartenberg del robot del Ejemplo 4.2
dh = [0 0 t1 0 0;
      pi/2 0 0 d2 1;
      0 0 t3 l2 0];
q = [t1 d2 t3];              % Vector de variables articulares
qd = [td1 dd2 td3];          % Vector con las derivadas de las variables articulares
v0 = [0 0 0]';                % Vector velocidad lineal del marco {0}
w0 = [0 0 0]';                % Vector velocidad angular del marco {0}
z = simple(velprop(dh,q,qd,v0,w0))

```

El vector resultante es:

$$\begin{bmatrix} {}^3v \\ {}^3\omega \end{bmatrix} = \begin{bmatrix} c_3(l_2 + d_2)\theta'_1 \\ -s_3(l_2 + d_2)\theta'_1 \\ d'_2 \\ s_3\theta'_1 \\ s_3\theta'_1 \\ \theta'_3 \end{bmatrix} \quad (4.79)$$

Empleando jacobn:

```

syms t1 d2 t3 real; % Variables articulares θ₁, d₂ y θ₃
syms l2 real; % Parámetro l₂
% Matriz con los parámetros de Denavit-Hartenberg del robot del Ejemplo 4.2
dh = [0 0 t1 0 0;
      pi/2 0 0 d2 1;
      0 0 t3 l2 0];
q = [t1 d2 t3]; % Vector de variables articulares
J = simple(jacobn(dh,q))

```

El resultado es la siguiente matriz:

$${}^3J(\theta) = \begin{bmatrix} c_3(l_2 + d_2) & 0 & 0 \\ -s_3(l_2 + d_2) & 0 & 0 \\ 0 & 1 & 0 \\ s_3 & 0 & 0 \\ c_3 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.80)$$

Obsérvese como ambos resultados coinciden.

### Ejemplo 4.7

Como último ejemplo se presenta la obtención del jacobiano del robot manipulador Puma 560 (capítulo 2):

Empleando velprop:

```

% Variables articulares θ₁, θ₂, θ₃, θ₄, θ₅ y θ₆
syms t1 t2 t3 t4 t5 t6 real;
% Derivadas de las variables articulares θ'_₁, θ'_₂, θ'_₃, θ'_₄, θ'_₅ y θ'_₆
syms td1 td2 td3 td4 td5 td6 real

```

```

% Parámetros a2, a3, d3 y d4
syms a2 a3 d3 d4 real;
% Matriz con los parámetros de Denavit-Hartenberg del Puma 560
dh = [0 0 t1 0 0;
       -pi/2 0 t2 0 0;
       0 a2 t3 d3 0;
       -pi/2 a3 t4 d4 0;
       pi/2 0 t5 0 0;
       -pi/2 0 t6 0 0];

q = [t1 t2 t3 t4 t5 t6]; % Vector de variables articulares
qd = [td1 td2 td3 td4 td5 td6]; % Vector con las derivadas de las
                                   % variables articulares
v0 = [0 0 0]'; % Vector velocidad lineal del marco {0}
w0 = [0 0 0]'; % Vector velocidad angular del marco {0}

z = simple( velprop(dh,q,qd,v0,w0) )

```

Utilizando jacobn:

```

% Variables articulares θ1, θ2, θ3, θ4, θ5 y θ6
syms t1 t2 t3 t4 t5 t6 real;
% Parámetros a2, a3, d3 y d4
syms a2 a3 d3 d4 real;
% Matriz con los parámetros de Denavit-Hartenberg del Puma 560
dh = [0 0 t1 0 0;
       -pi/2 0 t2 0 0;
       0 a2 t3 d3 0;
       -pi/2 a3 t4 d4 0;
       pi/2 0 t5 0 0;
       -pi/2 0 t6 0 0];
q = [t1 t2 t3 t4 t5 t6]; % Vector de variables articulares
J = simple( jacobn(dh,q) )

```

Tabla 4.4: Número de flops de MATLAB para cada caso (Pentium MMX 200 MHz).

	Método de R.P. Paul (función jacobn.m)	Método de propagación de velocidades
Manipulador ejemplo 4.1	21374	29994
Manipulador ejemplo 4.2	9883	23534
Puma 560	255093	322611

Finalmente en la tabla 4.4 se comparan el número de operaciones requeridas con el método de propagación de velocidades y las necesarias empleando la función jacobn, correspondiente al método de Paul (1981), en cada uno de los manipuladores considerados en los ejemplos.

Se observa como la complejidad computacional del método de Paul (1981) aplicado mediante jacobn es menor que la del método de propagación de velocidades aplicando velprop.

## 4.7 MODELOS CINEMÁTICOS DE ROBOTS MÓVILES

### 4.7.1 Hipótesis básicas

En el capítulo 2 se presentaron diversas configuraciones posibles de los robots móviles o vehículos robóticos que en su acepción más amplia incluyen vehículos aéreos, marinos, o robots terrestres con patas o pistas de deslizamiento. Se consideran aquí exclusivamente robots móviles con ruedas. Asimismo, se adoptan las siguientes hipótesis simplificadoras:

- El robot se mueve sobre una superficie plana.
- Los ejes de guiado son perpendiculares al suelo.
- Se supone que las ruedas se mueven con rodadura pura; es decir, el deslizamiento es despreciable en el periodo de control.
- El robot no tiene partes flexibles.
- Durante un periodo de tiempo suficientemente pequeño en el que se mantiene constante la consigna de dirección, el vehículo se moverá de un punto al siguiente a lo largo de un arco de circunferencia.
- El robot se comporta como un sólido rígido, de forma que si existen partes móviles (ruedas de dirección), éstas se situarán en la posición adecuada mediante el sistema de control.

### 4.7.2 Restricciones cinemáticas

Sea  $p = [p_1 \dots p_r]^T$  un vector de las variables necesarias para determinar completamente la posición y orientación de todas las partes de un sistema físico (robot). Cuando se utilizan estas variables para describir un movimiento, es necesario tener en cuenta que las variables pueden no ser independientes. Así, por ejemplo, en el movimiento del péndulo simple con una varilla que se ilustra en la figura 4.12, existe un único grado de libertad y las coordenadas ( $x, y$ ) de la masa del extremo de la varilla deben cumplir la restricción

$$x^2 + y^2 - l^2 = 0 \quad (4.81)$$

siendo  $l$  la longitud de la varilla del péndulo. Nótese que en el modelo geométrico de los robots manipuladores se involucraban restricciones análogas.

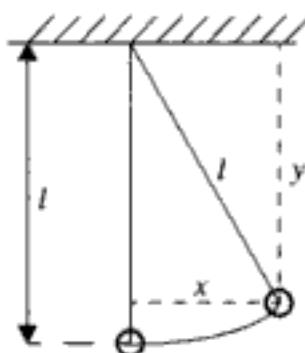


Figura 4.12: Péndulo simple.

En general (Rañada, 1994), es posible formular restricciones del tipo:

$$G_k(p, p', t) = 0; \quad k = 1, \dots, s;$$

es decir, se deben satisfacer  $s$  restricciones o ecuaciones en las variables  $p$ , sus derivadas  $p'$  y, posiblemente, el tiempo  $t$ .

Para que se cumpla una restricción debe haber una fuerza que obligue a ello, tal como la tensión de la varilla del péndulo. En general estas fuerzas pueden plantear problemas complejos ya que no son conocidas previamente y dependen del movimiento. En algunos casos el problema se resuelve eliminando las variables redundantes, quedándose con un conjunto mínimo de variables  $p = [p_1 \dots p_n]^T$ ,  $n < r$  y prescindiendo de las fuerzas. Sin embargo, esto no es siempre posible o deseable.

Las restricciones pueden ser holónomas o no holónomas. Las *holónomas* son aquellas en las que no intervienen las velocidades; es decir, tienen la forma:

$$G_k(p, t) = 0; \quad k = 1, \dots, s; \quad (4.82)$$

Las *no holónomas* dependen de las velocidades. Para que una restricción sea no holónoma se exige además que no sea integrable; es decir, que no se deduzca por derivación total con respecto al tiempo de una holónoma.

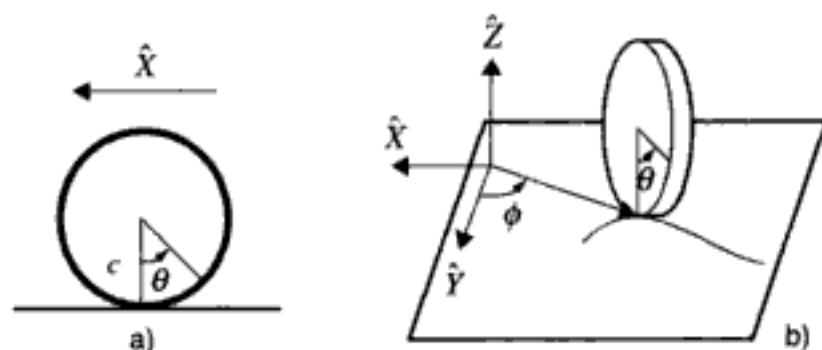
Los modelos cinemáticos de los robots manipuladores con base fija se caracterizan porque las restricciones del movimiento son holónomas. Sin embargo, este no es el caso de numerosos robots móviles en los cuales existen relaciones no holónomas entre las coordenadas en el espacio de los actuadores y las coordenadas en el espacio cartesiano.

Considérese el movimiento de la rueda de radio  $r$  en una dimensión tal como el que se ilustra en la figura 4.13a. La variable del actuador podría ser el giro  $\theta$  y la variable en el espacio cartesiano la  $x$  que indica el espacio recorrido. Las dos variables obedecen a la condición de rodadura

$$x' = c\theta' \quad (4.83)$$

que depende de las velocidades, pero puede deducirse por derivación de la restricción holónoma

$$x - c\theta = \text{constante} \quad (4.84)$$



**Figura 4.13:** Restricciones no holónomas. a) Movimiento de rueda en una dirección; b) movimiento de rueda en plano.

Por consiguiente, en este caso, no existen restricciones no holónomas. Nótese que se necesita una única coordenada ( $x$  o  $\theta$ ) y existe también un grado de libertad.

Sin embargo, si se estudia el movimiento de la rueda en el plano, tal como se muestra en la figura 4.13b, aparecen restricciones no holónomas. En efecto, considérese el movimiento de la rueda de forma que el diámetro correspondiente al punto de contacto con el suelo esté siempre en posición vertical. En este caso, pueden emplearse cuatro coordenadas para especificar completamente la posición y orientación de la rueda: las coordenadas ( $x$ ,  $y$ ) del punto de contacto, el ángulo  $\theta$  entre la vertical y un radio de referencia, el cual indica cuánto ha girado el disco, y el ángulo de orientación  $\phi$  de la rueda. La condición de rodadura sin deslizamiento introduce dos restricciones, ya que el espacio que el punto de contacto recorre sobre el borde de la rueda es igual al que recorre en el plano. Por consiguiente, proyectando la velocidad del punto de contacto en el plano, paralela y perpendicularmente al disco, se obtiene:

$$-x' \operatorname{sen} \phi + y' \cos \phi = \theta' c \quad (4.85)$$

$$x' \cos \phi + y' \operatorname{sen} \phi = 0 \quad (4.86)$$

Estas dos restricciones no son integrables (no se pueden obtener como las derivadas de dos condiciones holónomas), lo cual se comprueba observando que no se satisface el teorema de Frobenius. Por tanto, no se pueden obtener relaciones funcionales entre las variables  $(x, y, \theta, \phi)$  a partir de las (4.85) y (4.86). Dados unos valores  $(x_0, y_0, \theta_0, \phi_0)$ , haciendo rodar la rueda sin deslizar y girándola alrededor del eje vertical, es posible llegar a cualquier otra configuración  $(x_f, y_f, \theta_f, \phi_f)$ , lo cual implica que no existe una relación funcional entre estos valores. Sin embargo, las direcciones de movimiento deben satisfacer las ecuaciones anteriores y, por tanto, el camino no puede ser cualquiera.

Nótese como, en los manipuladores con base fija estudiados en las secciones precedentes, la ecuación del modelo jacobiano (4.6) constituye una restricción que puede ser integrada para obtener el modelo geométrico. Si el manipulador tiene un número suficiente de articulaciones, podrá moverse en su espacio de trabajo haciendo que su efecto final describa cualquier camino, exceptuando los puntos en que existan singularidades del jacobiano y teniendo en cuenta limitaciones físicas, tales como las saturaciones en los actuadores y las limitaciones de ángulos de giro de las articulaciones.

En la práctica, existen diferentes tipos de ruedas cuya consideración tiene una notable influencia en el modelo cinemático del vehículo. Así, cabe distinguir entre cuatro tipos de ruedas (ver, por ejemplo, Bastin y otros, 1997). Las ruedas fijas sólo pueden rotar sobre su eje (ángulo  $\theta$  en la figura 4.13). Las ruedas de direccionamiento se caracterizan por la rotación alrededor del eje vertical (ángulo  $\phi$  en la figura 4.13) que pasa por el centro de la rueda y que permite orientarla con respecto al vehículo. Las ruedas de castor también son orientables respecto al vehículo, pero el eje vertical de rotación no pasa por el centro de la rueda. Finalmente, las denominadas ruedas suecas (ver ruedas del robot Uranus en el capítulo 2) permiten variar la dirección de la velocidad del punto de contacto con relación al plano de la rueda (el vector velocidad puede tener un ángulo con respecto a la tangente).

Considérese un sistema de referencia  $\{G\}$  y un sistema  $\{L\}$  con centro en el punto de guiado del vehículo y eje  $\hat{Y}_L$  en la dirección del eje longitudinal del vehículo (figura 4.14).

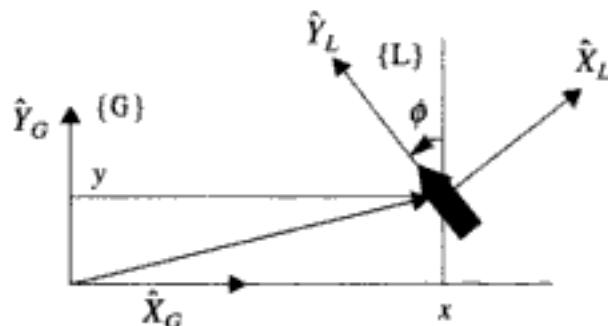


Figura 4.14: Cambio de sistema de referencia en navegación de robots móviles.

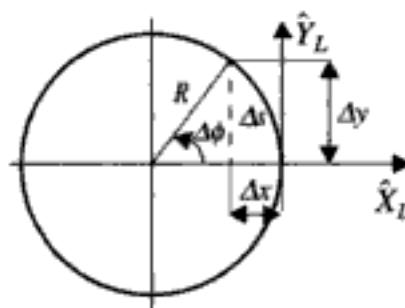


Figura 4.15: Círculo osculador.

Supóngase que el vehículo se desplaza en un intervalo de control según un arco de circunferencia, tal como se muestra en la figura 4.15. Esta suposición es válida para intervalos de control suficientemente pequeños.

La velocidad lineal del vehículo viene dada por:

$$v = \frac{\Delta s}{\Delta t} \quad (4.87)$$

y la velocidad angular por:

$$\omega = \frac{\Delta\phi}{\Delta t} \quad (4.88)$$

siendo  $\Delta s$  y  $\Delta\phi$  respectivamente el espacio recorrido por el punto de guiado del vehículo y su cambio de orientación durante el intervalo de control  $\Delta t$ .

La longitud  $\Delta s$  del arco recorrido por el robot en  $\Delta t$  viene dada por:

$$\Delta s = R\Delta\phi \quad (4.89)$$

siendo  $R$  el radio de giro o radio de la circunferencia que describe el punto de guiado.

La curvatura se define como la inversa del radio de giro:

$$\gamma = \frac{1}{R} = \frac{\Delta\phi}{\Delta s} \quad (4.90)$$

Las ecuaciones de movimiento en el sistema  $\{L\}$  de la figura 4.15 en la posición inicial son:

$${}^L(\Delta x) = -(R - R \cos(\Delta\phi)) \quad (4.91)$$

$${}^L(\Delta y) = R \sin(\Delta\phi) \quad (4.92)$$

Si la orientación inicial del vehículo con respecto al sistema  $\{G\}$  es de  $\phi$ , el movimiento en el sistema  $\{G\}$  se determina rotando  $\phi$ :

$$\begin{aligned} \Delta x &= R [\cos(\Delta\phi) - 1] \cos(\phi) - R \sin(\Delta\phi) \sin(\phi) \\ \Delta y &= R [\cos(\Delta\phi) - 1] \sin(\phi) + R \sin(\Delta\phi) \cos(\phi) \end{aligned} \quad (4.93)$$

Suponiendo que el intervalo de control es suficientemente pequeño, también lo será el cambio de orientación  $\Delta\phi$  con lo cual se tendrá que

$$\begin{aligned} \cos(\Delta\phi) &\equiv 1 \\ \sin(\Delta\phi) &\equiv \Delta\phi \end{aligned} \quad (4.94)$$

Sustituyendo en las anteriores ecuaciones se tiene que

$$\begin{aligned} \Delta x &= -R\Delta\phi \sin(\phi) \\ \Delta y &= R\Delta\phi \cos(\phi) \end{aligned} \quad (4.95)$$

y teniendo en cuenta (4.89)

$$\begin{aligned} \Delta x &= -\Delta s \sin(\phi) \\ \Delta y &= \Delta s \cos(\phi) \end{aligned} \quad (4.96)$$

Dividiendo ambas ecuaciones por  $\Delta t$ , teniendo en cuenta (4.87) y haciendo tender  $\Delta t$  a cero se llega a:

$$x' = -v \sin \phi \quad (4.97)$$

$$y' = v \cos \phi \quad (4.98)$$

ecuaciones a las que puede añadirse la que se obtiene a partir de la (4.88):

$$\dot{\phi} = \omega \quad (4.99)$$

la cual proporciona la variación de la orientación.

Nótese que en el modelo anterior no se considera el ángulo de direccionamiento de las ruedas, variables que se utilizan en diversas configuraciones de robots móviles. Estos ángulos se emplean en modelos tales como el de la bicicleta que se ilustra en la figura 4.16.

El ángulo de direccionamiento  $\alpha$  es el que forma el eje longitudinal del vehículo con la rueda delantera, la cual se supone direccionable.



**Figura 4.16:** Modelo de la bicicleta. Rd rueda delantera, Rt rueda trasera.

#### 4.7.3 Modelo jacobiano

Sea  $p$  el vector que representa un punto en el espacio de  $n$  coordenadas generalizadas y  $q$  el vector de  $m$  variables de actuación, siendo  $n > m$ . Sean  $p'$  y  $q'$  las derivadas temporales correspondientes. En lo que sigue, por omisión, se considera que las variables se expresan en el sistema de referencia global ( $G$ ). El *modelo directo* en la forma (4.59) es

$$p' = J(p)q' \quad (4.100)$$

siendo  $J(p)$  el jacobiano. Este jacobiano puede escribirse en la forma (Zhao y Bement, 1992):

$$p' = f(p) + \sum_{i=1}^m g(p)_i q'_i \quad (4.101)$$

siendo  $f$  y  $g$  funciones vectoriales analíticas.

Si  $p = [x \ y \ \phi]^T$  es el vector con las coordenadas globales del punto de guía del vehículo y la orientación, las ecuaciones (4.97), (4.98) y (4.99) pueden expresarse en la forma (4.101) con  $f(p) = 0$ ;  $m = 2$  como

$$p' = \begin{bmatrix} -\sin \phi \\ \cos \phi \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \quad (4.102)$$

siendo  $v$  la velocidad lineal del vehículo y  $\omega$  la angular.

Las ecuaciones (4.102) pueden expresarse también en la forma del modelo (4.100) como:

$$\begin{bmatrix} x' \\ y' \\ \phi' \end{bmatrix} = \begin{bmatrix} -\sin \phi & 0 \\ \cos \phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (4.103)$$

siendo  $q' = [v \ \omega]^T$  el vector de variables de entrada.

Combinando las dos primeras ecuaciones de (4.103) es posible obtener la siguiente restricción independiente de  $v$ :

$$x' \cos \phi + y' \sin \phi = 0 \quad (4.104)$$

que constituye la restricción no holónoma del movimiento (4.86) según la cual el vehículo debe moverse en cada instante según la dirección de su eje longitudinal de simetría.

$$\operatorname{tg} \phi = -\frac{x'}{y'} \quad (4.105)$$

En efecto, teniendo en cuenta (4.86), la posición  $(x, y)$  y la orientación  $\phi$  del vehículo no son independientes.

El modelo inverso, tal como el (4.61) para robots manipuladores, involucra la inversa del jacobiano. Para resolverlo de una forma general sería necesario invertir el jacobiano. Cuando el jacobiano no es cuadrado, es necesario emplear la seudoinversa. Multiplicando ambos miembros de (4.100) por  $J^T$  y despejando  $q'$  se obtiene:

$$q' = \{[J(p)]^T J(p)\}^{-1} [J(p)]^T p' \quad (4.106)$$

Así, para el modelo (4.103) se obtiene:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ \phi' \end{bmatrix} \quad (4.107)$$

de la primera se deduce

$$v = -x' \sin \phi + y' \cos \phi \quad (4.108)$$

que coincide con la ecuación (4.85).

#### 4.7.4 Modelos de diferentes configuraciones

En la figura 4.17 se muestran cuatro configuraciones diferentes de robots móviles. Las coordenadas  $(x, y)$  suministran la posición del robot con respecto a las coordenadas globales y el ángulo  $\phi$  su orientación con respecto a un eje paralelo al  $\hat{Y}$ .

La configuración de la figura 4.17a es la denominada *síncrona* o “synchro-drive” en la cual existen transmisiones que permiten orientar las tres ruedas simultáneamente con una velocidad angular  $\omega$  y hacer que el vehículo se desplace con una velocidad lineal  $v$ . En esta configuración el modelo viene dado por las ecuaciones (4.102) o (4.103).

Nótese que para especificar la configuración hay que indicar los valores de las tres variables  $(x, y, \phi)$ . Se tiene una restricción no holónoma y dos grados de libertad.

En la figura 4.17b se representa la *locomoción con guiado diferencial*, tal como la de la plataforma comercial Labmate mencionada en el capítulo 2. En este caso, las variables de control son las velocidades de las ruedas laterales. Sean  $\omega_l$  y  $\omega_d$ , las velocidades de giro de la rueda izquierda y derecha, respectivamente. Si el radio de la rueda es  $c$ , las velocidades lineales correspondientes son  $v_l = \omega_l c$  y  $v_d = \omega_d c$ . En este caso, la velocidad lineal y la velocidad angular correspondientes en el modelo (4.101) vienen dadas por:

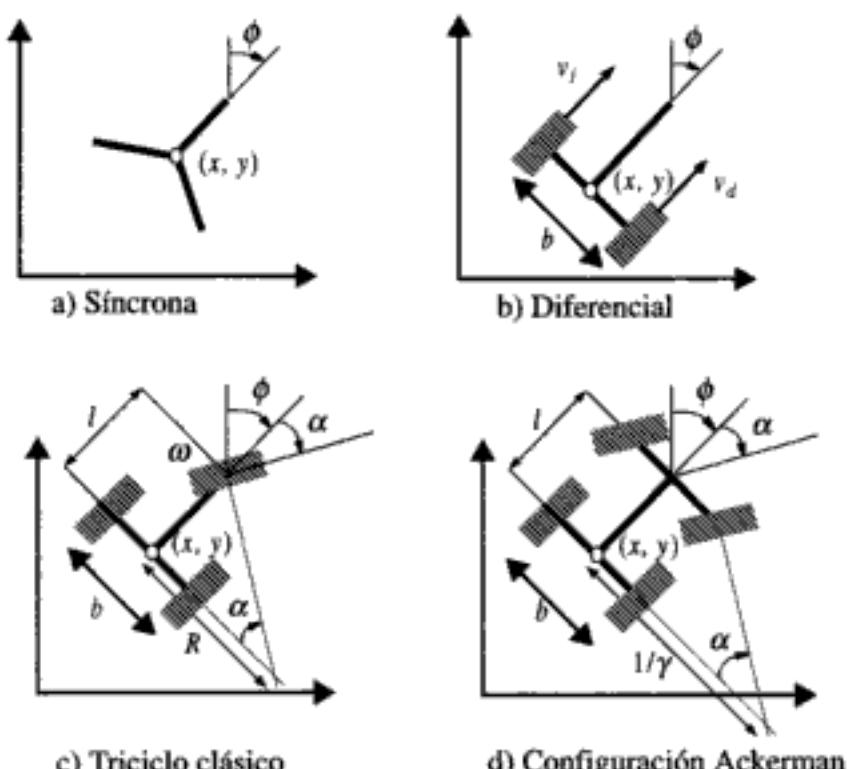


Figura 4.17: Configuraciones de robots móviles.

$$v = \frac{v_d + v_i}{2} = \frac{(\omega_d + \omega_i)c}{2} \quad (4.109)$$

$$\omega = \frac{v_d - v_i}{b} = \frac{(\omega_d - \omega_i)c}{b} \quad (4.110)$$

siendo  $b$  la vía del vehículo (distancia que separa las dos ruedas centrales). Por consiguiente, si se especifican la velocidad lineal  $v$  y angular  $\omega$  del vehículo, las velocidades de giro que hay que aplicar a las ruedas izquierda y derecha son:

$$\omega_i = \frac{v - (b/2)\omega}{c} \quad (4.111)$$

$$\omega_d = \frac{v + (b/2)\omega}{c} \quad (4.112)$$

Sustituyendo (4.109) y (4.110), el modelo (4.102) puede expresarse en función de estas variables de control como

$$\begin{bmatrix} x' \\ y' \\ \phi \end{bmatrix} = \begin{bmatrix} -(c \operatorname{sen} \phi)/2 \\ (c \cos \phi)/2 \\ -c/b \end{bmatrix} \omega_i + \begin{bmatrix} -(c \operatorname{sen} \phi)/2 \\ (c \cos \phi)/2 \\ c/b \end{bmatrix} \omega_d \quad (4.113)$$

$$= \begin{bmatrix} -(c \operatorname{sen} \phi)/2 & -(c \operatorname{sen} \phi)/2 \\ (c \cos \phi)/2 & (c \cos \phi)/2 \\ -c/b & c/b \end{bmatrix} \begin{bmatrix} \omega_i \\ \omega_d \end{bmatrix} \quad (4.114)$$

El esquema de la figura 4.17c representa el *triciclo convencional*, en el cual la rueda delantera se utiliza tanto para la orientación como para la tracción. Esta configuración corresponde al robot Romeo 3R mencionado en el capítulo 2. En este caso las variables de control suelen tomarse como el ángulo  $\alpha$  de dirección de la rueda delantera (o su velocidad angular  $\omega_\alpha$ ) y la velocidad de giro de la misma rueda  $\omega_i$  (o su velocidad lineal correspondiente  $v_i = c\omega_i$ ). Se supondrá que el punto de guía ( $x, y$ ) está en el centro del eje trasero.

Las velocidades lineal  $v$  y angular  $\omega$  del vehículo que corresponden a las entradas en el modelo (4.102) son respectivamente

$$v = v_i \cos \alpha = c\omega_i \cos \alpha \quad (4.115)$$

y

$$\alpha' = \omega_\alpha \quad (4.116)$$

Asimismo, el ángulo de orientación del vehículo varía según:

$$\dot{\phi}' = \frac{c\omega_t}{l} \sin \alpha = \frac{v_t}{l} \sin \alpha \quad (4.117)$$

Por consiguiente, sustituyendo en (4.102), se obtiene el siguiente modelo en función de las velocidades de control  $v_t$  y  $\omega_\alpha$ :

$$\begin{bmatrix} x' \\ y' \\ \dot{\phi}' \\ \alpha' \end{bmatrix} = \begin{bmatrix} -\sin \phi & \cos \alpha & 0 \\ \cos \phi & \cos \alpha & 0 \\ (\sin \alpha)/l & 0 & 1 \end{bmatrix} v_t + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega_\alpha = \begin{bmatrix} -\sin \phi & \cos \alpha & 0 \\ \cos \phi & \cos \alpha & 0 \\ (\sin \alpha)/l & 0 & 1 \end{bmatrix} \begin{bmatrix} v_t \\ \omega_\alpha \\ 0 \end{bmatrix} \quad (4.118)$$

Obsérvese también que, conocidas las velocidades deseadas lineal  $v$  y angular  $\omega$ , las variables de control  $\alpha$  y  $\omega_t$  pueden obtenerse mediante:

$$\alpha = \operatorname{arctg} \left( \frac{l}{R} \right) = \operatorname{arctg} \left( \frac{l\omega}{v} \right) \quad (4.119)$$

$$\omega_t = \frac{v_t}{c} = \frac{\sqrt{v^2 + \omega^2 l^2}}{c} \quad (4.120)$$

En efecto, obsérvese como la relación entre la velocidad de traslación y la de rotación es el radio de giro  $R$ . Asimismo,  $R = 1/\gamma$  siendo  $\gamma$  la curvatura.

El triciclo puede representarse también mediante el denominado modelo de la bicicleta que se ilustra en la figura 4.16. En este modelo se adoptan las expresiones (4.97) y (4.98). Con respecto a la orientación, se tiene en cuenta la definición de la curvatura (4.90) de donde se obtiene la ecuación  $\dot{\phi}' = v\gamma$ . Por tanto, las ecuaciones son:

$$\begin{aligned} x' &= -v \sin \phi \\ y' &= v \cos \phi \\ \dot{\phi}' &= v\gamma \end{aligned} \quad (4.121)$$

siendo  $\gamma$  la curvatura, que puede obtenerse en función del ángulo de direccionamiento mediante  $\gamma = (\operatorname{tg} \alpha)/l$ .

En general, si se emplea el modelo de la bicicleta, para obtener los modelos inversos puede aplicarse la expresión (4.106), donde el jacobiano viene dado por las correspondientes ecuaciones del modelo directo: (4.114) para el guiado diferencial y (4.118) para el triciclo.

### Ejemplo 4.8

Se trata de obtener el modelo inverso del triciclo convencional. En esta configuración, teniendo en cuenta (4.118), el jacobiano puede escribirse como:

$$J = \begin{bmatrix} -\sin \phi & \cos \alpha & 0 \\ \cos \phi & \cos \alpha & 0 \\ (\sin \alpha)/l & 0 & 1 \end{bmatrix} \quad (4.122)$$

Aplicando (4.106) se obtiene la siguiente expresión para el modelo inverso:

$$\begin{bmatrix} v_t \\ \omega_a \end{bmatrix} = \begin{bmatrix} \frac{-l^2 \sin \phi}{(l \cos \alpha)^2 + \sin^2 \alpha} & \frac{l^2 \cos \phi}{(l \cos \alpha)^2 + \sin^2 \alpha} & \frac{l \sin \alpha}{(l \cos \alpha)^2 + \sin^2 \alpha} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ \phi \\ \alpha \end{bmatrix} \quad (4.123)$$

Obsérvese como la ecuación de la segunda fila de (4.123) suministra una relación trivial. Nótese también como, de la tercera de las expresiones (4.121), se deduce:  $\phi' = v\gamma = v(\tan \alpha)/l$ . Asimismo, es necesario tener en cuenta la restricción (4.105). Por tanto, se tendrá  $\phi = \arctan(-x'/y')$ .

Supóngase que se desea que el centro de guiado del triciclo, situado en el punto medio del eje entre las dos ruedas traseras, describa un arco de circunferencia de radio 1, definido por  $x(t) = \cos t$ ,  $y(t) = \sin t$ , desde el punto inicial  $(1, 0)$  hasta el punto final  $(0, 1)$ .

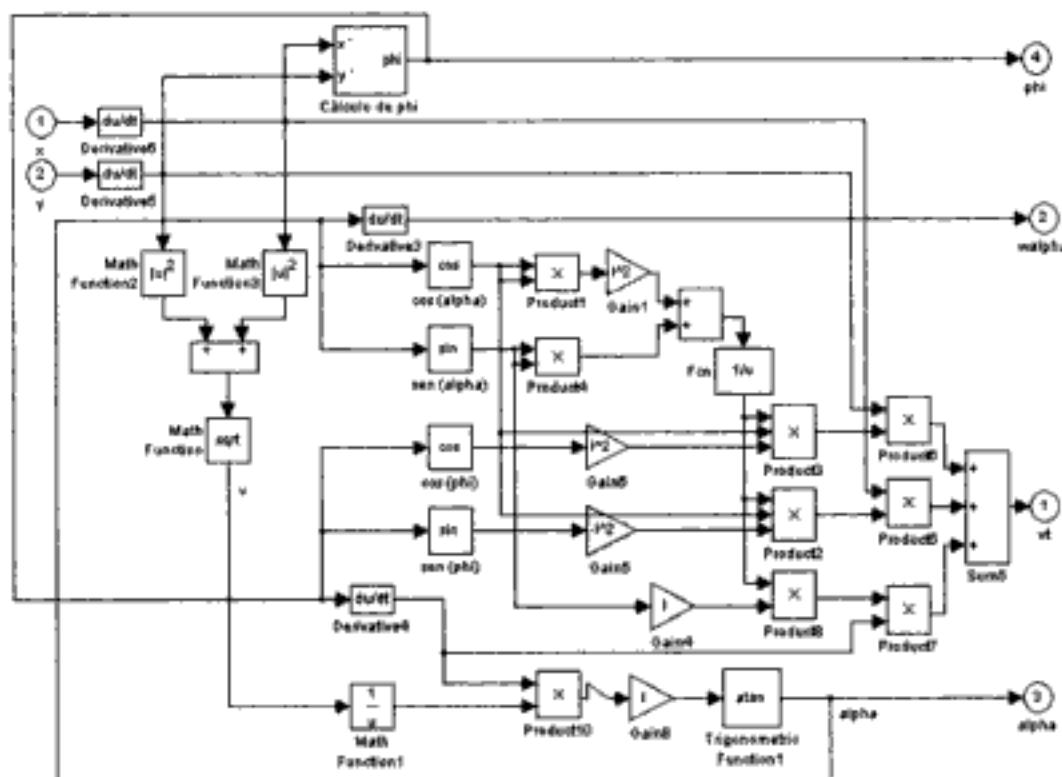
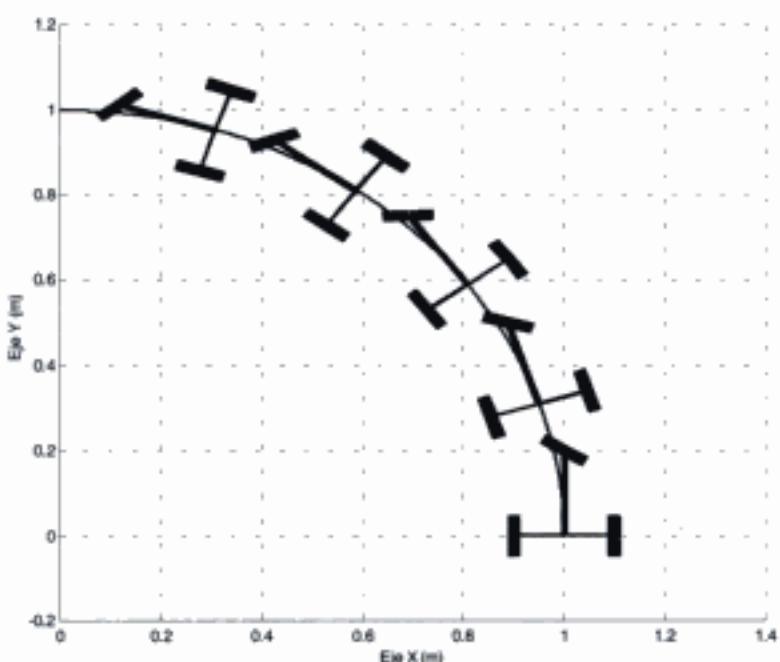


Figura 4.18: Diagrama en Simulink del modelo inverso del triciclo convencional.



**Figura 4.19:** Resultados de la simulación del modelo inverso del triciclo convencional en el seguimiento de un arco de circunferencia.

En la figura 4.18 se muestra el diagrama Simulink correspondiente al modelo inverso del triciclo. Obsérvese como la velocidad  $v$  se obtiene según:  $v = \sqrt{x^2 + y^2}$ . Al efectuar la simulación se comprueba que el ángulo  $\alpha$  es constante y de valor igual a  $56,31^\circ$ , por lo cual  $\omega_\alpha$  resulta ser cero. Asimismo,  $v_r$  también permanece constante e igual a 1,8028 m/s.

En la figura 4.19 se presentan los resultados de la simulación con el diagrama la figura 4.18 siendo la distancia entre el eje de la rueda delantera y el eje de las traseras de  $l = 1,5$  m. Las posiciones del triciclo a lo largo de la trayectoria se representan mediante la función `trirep` de la herramienta MATLAB-Simulink.

Es interesante comprobar que, derivando con respecto al tiempo en la expresión  $\phi = \arctg(-x'/y')$ , se obtiene  $\dot{\phi} = (x'y'' - x''y')/(x'^2 + y'^2)$ , que resulta ser la misma expresión que la  $\dot{\phi} = v\gamma = \sqrt{x'^2 + y'^2}\gamma$  cuando se sustituye la curvatura  $\gamma$  en un determinado punto de una curva por su expresión en función de las derivadas  $x', y', x'', y''$  en dicho punto, tal como se verá en el capítulo 10.

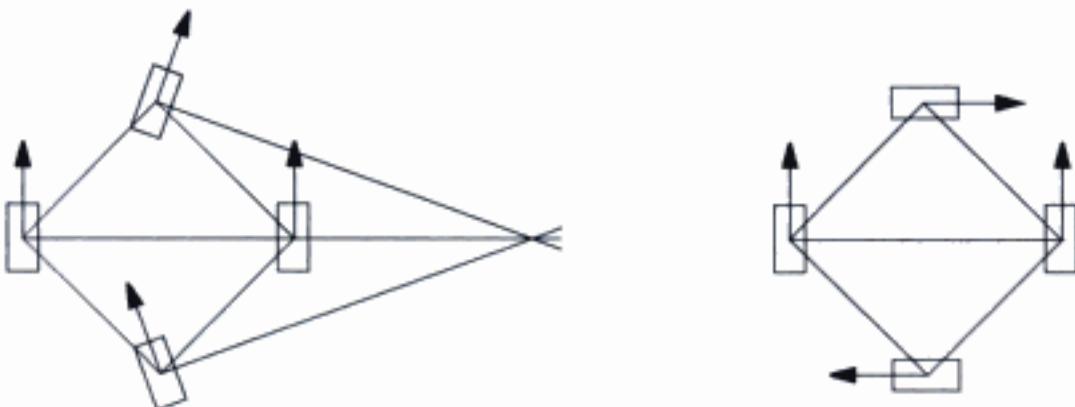
\* \* \*

Considérese ahora un vehículo de cuatro ruedas con sistema de locomoción con *configuración Ackerman*, tal como el que se representa en la figura (4.17). Se supone que el centro de guiado del vehículo (origen del sistema de referencia local  $\{L\}$ ) está situado en la mitad del eje de las ruedas de tracción (ruedas traseras). El vehículo puede representarse también según el modelo de la bicicleta (Shin y Singh, 1990), siendo  $\alpha = (\operatorname{tg} \alpha)/l$  con  $\alpha$  el ángulo de direccionamiento. Sin embargo, en general no existen expresiones explícitas de la cinemática inversa de la configuración Ackerman.

man, debiendo aplicarse la integración numérica. Hay que tener en cuenta que la velocidad real de las cuatro ruedas es diferente y, por consiguiente, el cálculo de la velocidad del centro de guiado del vehículo a partir de la velocidad de una rueda da lugar a errores. Asimismo, el radio de curvatura tampoco puede medirse de forma directa a partir de la curvatura de las ruedas directrices ya que cada una de ellas se mueve según un arco de diferente radio. Por último, hay que señalar que efectos tales como la deformación de los neumáticos hacen que la orientación de la rueda de dirección no sea la prevista.

Las configuraciones del triciclo y de la locomoción Ackerman pueden representarse de forma más o menos simplificada mediante el modelo de la bicicleta y tienen dos grados de libertad.

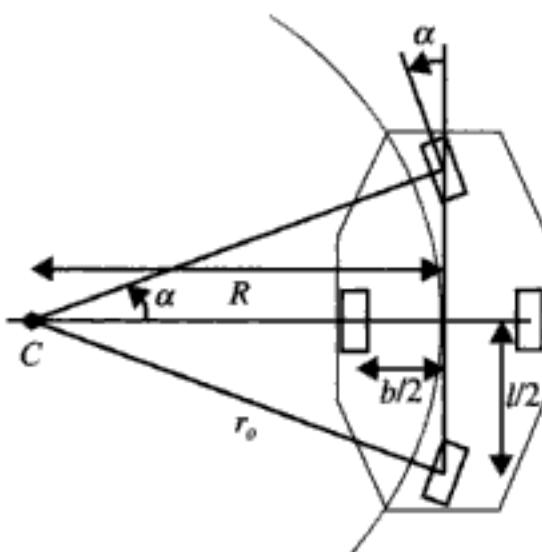
Otra configuración de cuatro ruedas es la de los robots móviles RAM-1 (Ollero y otros, 1993) y AURORA (Mandow y otros, 1996) mencionadas en el capítulo 2. Como se puso de manifiesto en el mencionado capítulo, estos robots tienen una configuración, a la que se ha denominado rómbica, tal como se muestra en la figura 4.20.



**Figura 4.20:** Locomoción con configuración en rombo.

La configuración rómbica tiene cuatro ruedas situadas en los vértices de un rombo imaginario cuya diagonal principal coincide con el sentido de la marcha. El punto de guiado se elige en el centro del vehículo. Las dos ruedas centrales, situadas en el eje transversal, son motrices, y cada una de ellas está movida por un motor independiente. Girando los motores en distinto sentido se consigue que el vehículo pueda girar sobre su propio centro geométrico minimizando el espacio necesario. Se dice que el radio de giro es cero. Para girar en esta situación límite, las ruedas delantera y trasera deben estar giradas 90°. Las ruedas delantera y trasera son las de dirección, accionándose conjuntamente mediante un tercer motor. Las dos ruedas de dirección giran el mismo ángulo, pero en distinto sentido. Por consiguiente, se pueden orientar hacia el centro de la trayectoria requerida.

RAM-1 tiene dos modos de locomoción. El primero de ellos es el diferencial, empleando las ruedas laterales de forma análoga a la figura 4.17b. El segundo es el modo



**Figura 4.21:** Configuración cinemática del Aurora.

maniobra. En este segundo caso se emplea la cinemática inversa para determinar el valor de las variables de control (ángulo de dirección  $\alpha$  o velocidad angular de las ruedas laterales  $\omega_l$  y  $\omega_d$ ) que hacen que la posición y orientación del robot sea la deseada. Si se pretende que el robot tenga una curvatura deseada  $\gamma$ , el ángulo de direccionamiento deberá ser (figura 4.21):

$$\alpha = \operatorname{arctg} \left( \frac{l}{2} \gamma \right) \quad (4.124)$$

Para que el movimiento se ejecute de forma apropiada y la curvatura sea  $\gamma$ , es necesario que las velocidades de las ruedas laterales sean:

$$v_i = \omega \left( \frac{1}{\gamma} - \frac{b}{2} \right) = v \left( 1 - \gamma \frac{b}{2} \right) \quad (4.125)$$

$$v_d = \omega \left( \frac{1}{\gamma} + \frac{b}{2} \right) = v \left( 1 + \gamma \frac{b}{2} \right) \quad (4.126)$$

siendo  $\omega$  la velocidad angular y  $b$  la vía del vehículo. La velocidad lineal del centro del vehículo viene dada por

$$v = \frac{v_i + v_d}{2} \quad (4.127)$$

y la velocidad angular de dicho centro

$$\omega = \frac{v_d - v_i}{b} \quad (4.128)$$

Sustituyendo en (4.102) se obtiene:

$$\begin{bmatrix} x' \\ y' \\ \phi' \\ \alpha' \end{bmatrix} = \begin{bmatrix} -(\sin \phi)/2 \\ (\cos \phi)/2 \\ -1/b \\ 0 \end{bmatrix} v_i + \begin{bmatrix} -(\sin \phi)/2 \\ (\cos \phi)/2 \\ 1/b \\ 0 \end{bmatrix} v_d + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega_a \quad (4.129)$$

que constituye el modelo jacobiano del vehículo.

#### 4.7.5 Estimación de la posición y orientación

La estimación de la posición y orientación de los robots móviles hace necesaria la integración de las ecuaciones de los modelos presentados en 4.7.3 y 4.7.4.

Así, en el modelo (4.103), conociendo la posición y orientación inicial  $p_0 = [x_0 \ y_0 \ \phi_0]^T$ , la integración daría:

$$\begin{bmatrix} x \\ y \\ \phi \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ \phi_0 \end{bmatrix} + \begin{bmatrix} \int_0^t -v \sin \phi d\tau \\ \int_0^t v \cos \phi d\tau \\ \int_0^t \omega d\tau \end{bmatrix} \quad (4.130)$$

De igual forma, en el caso de direccionamiento diferencial, se tiene:

$$\begin{bmatrix} x \\ y \\ \phi \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ \phi_0 \end{bmatrix} + \begin{bmatrix} \int_0^t \frac{-c \sin \phi}{2} (\omega_d + \omega_i) d\tau \\ \int_0^t \frac{c \cos \phi}{2} (\omega_d + \omega_i) d\tau \\ \int_0^t \frac{c}{b} (\omega_d - \omega_i) d\tau \end{bmatrix} \quad (4.131)$$

En el caso del modelo del triciclo se aplican las ecuaciones:

$$v = c \omega_i \cos \alpha \quad (4.132)$$

$$\begin{bmatrix} x \\ y \\ \phi \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ \phi_0 \end{bmatrix} + \begin{bmatrix} \int_0^t -v \sin \phi d\tau \\ \int_0^t v \cos \phi d\tau \\ \int_0^t \frac{v_i \sin \alpha}{l} d\tau \end{bmatrix} \quad (4.133)$$

En la configuración rómbica se tiene:

$$\begin{bmatrix} x \\ y \\ \phi \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ \phi_0 \end{bmatrix} + \begin{bmatrix} \int_0^t -v \sin \phi \, d\tau \\ \int_0^t v \cos \phi \, d\tau \\ \int_0^t \frac{2v \tan \alpha}{l} \, d\tau \end{bmatrix} \quad (4.134)$$

No obstante, cabe señalar que la estimación de la posición mediante las expresiones anteriores es una aproximación válida sólo para períodos de tiempo suficientemente pequeños, ya que en caso contrario se produce una acumulación de error debido a inevitables deslizamientos y otros efectos provocados por comportamientos dinámicos, relacionados fundamentalmente con la interacción entre el vehículo y el terreno, que no están contemplados en el modelo.

#### Ejemplo 4.9

Se trata de obtener las trayectorias seguidas por la posición y orientación de un robot móvil con guiado diferencial cuando se aplican diversas combinaciones de velocidades angulares a las ruedas izquierda y derecha. Para ello se emplea el esquema de Simulink que aparece en la figura 4.22. Se considera un radio de las ruedas de  $c = 0,25$  m y una separación entre las mismas de  $b = 1$  m. Asimismo, la posición inicial será el origen de coordenadas y la orientación inicial tendrá un valor de  $-\pi/4$  rad. Se simulará la evolución del robot durante 30 segundos.

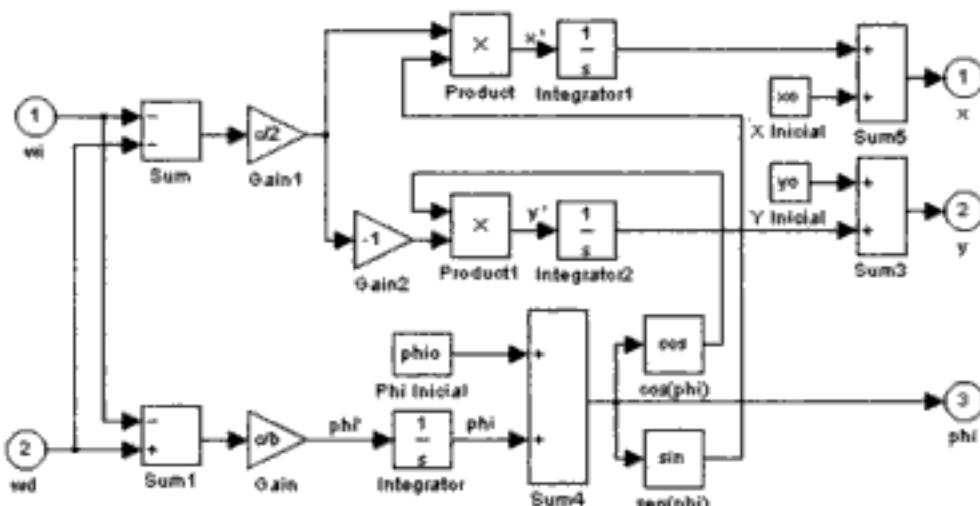


Figura 4.22: Diagrama de Simulink para el cálculo de posición y orientación de robots móviles con guiado diferencial.

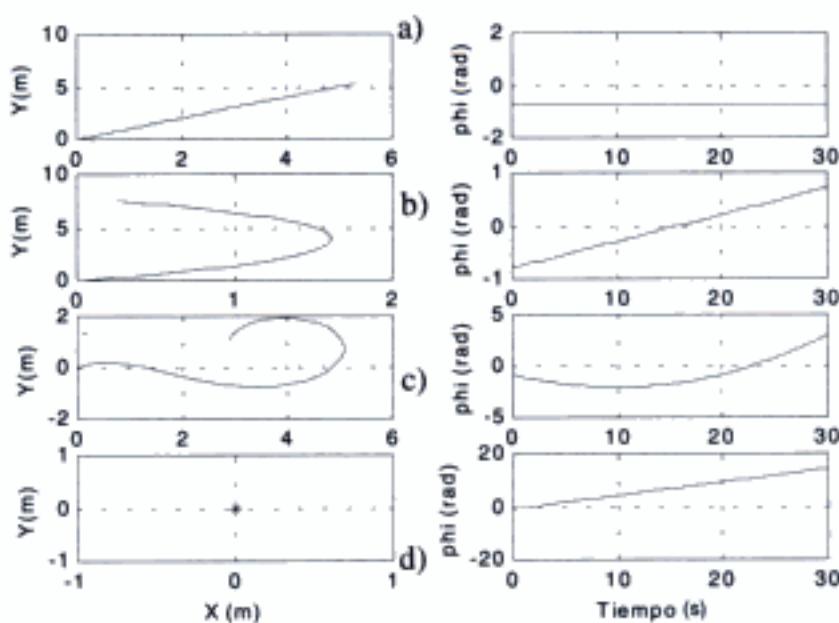


Figura 4.23: Casos simulados para el guiado diferencial.

En lo que sigue se comentan los resultados obtenidos en los cuatro casos de la figura 4.23:

- Se tiene una misma velocidad angular ( $1 \text{ rad/s}$ ) en el mismo sentido en las ruedas izquierda y derecha, por lo que el robot avanza en línea recta con la orientación inicial.
- Se aplica una velocidad angular a la rueda derecha un poco superior ( $1.2 \text{ rad/s}$ ) a la que se aplica a la rueda izquierda ( $1 \text{ rad/s}$ ), con lo que el robot gira continuamente hacia la izquierda.
- Se tiene una velocidad angular constante de  $1 \text{ rad/s}$  en la rueda izquierda, mientras que a la rueda derecha se aplica una rampa que empieza en  $0$  y que crece continuamente con pendiente  $0.1$ . En este caso, el robot gira hacia la derecha hasta el instante en que la velocidad angular de la rueda derecha supera a la de la izquierda, momento en el cual se invierte el sentido de giro del robot.
- Se aplican iguales velocidades angulares en ambas ruedas ( $1 \text{ rad/s}$ ), pero en sentidos opuestos, con lo cual se pone de manifiesto una de las principales características del guiado diferencial: la posibilidad de que el robot gire sobre sí mismo.

#### 4.7.6 Vehículos robóticos con manipuladores

Para acabar el capítulo conviene considerar robots móviles tal como el que se muestra en la figura 4.1, en la cual se representa un vehículo autónomo con un manipulador a bordo. La posición y orientación de la muñeca del manipulador en coordenadas cartesianas  ${}^G p_M$  puede calcularse según:

$${}^G p_M = {}^G p_V + {}^V p_M \quad (4.135)$$

siendo  ${}^V p_M$  la posición y orientación de la muñeca con respecto al sistema  $\{V\}$  asociado al punto de guía del vehículo, que puede calcularse conociendo las ecuaciones cinemáticas del manipulador, y  ${}^G p_V$  la posición y orientación del vehículo con respecto al sistema global  $\{G\}$ . La consideración de la cinemática del vehículo hace necesario tener en cuenta la restricción no holónoma definida por la ecuación diferencial (4.104). Nótese que, a diferencia de las ecuaciones definidas para los manipuladores, la restricción es una ecuación diferencial.

Existen autores que han tratado de forma conjunta las ecuaciones cinemáticas y dinámicas del manipulador y del vehículo, para lo cual consideran el vehículo robótico como una articulación planar con tres grados de libertad que, a su vez, se trata como tres articulaciones de un grado de libertad (dos prismáticas y una de rotación) puestas en serie mediante enlaces de longitud y masa nulas. Por consiguiente, se utiliza el vector de coordenadas generalizadas  $[x \ y \ \phi \ \theta_1 \ \theta_2 \ \theta_3]$  en el cual las tres primeras componentes son la posición y orientación y las tres últimas los ángulos girados por las tres articulaciones del manipulador.

## 4.8 CONCLUSIONES

---

El estudio de los modelos cinemáticos es básico en robótica. Los modelos directos se emplean para simular el comportamiento del robot cuando se le aplican estímulos en sus articulaciones. Estos estímulos, o variables de control, suelen ser pares de control en las articulaciones, o bien velocidades de giro en los ejes de los robots móviles. Se consideran modelos que relacionan las posiciones y orientaciones de un sistema de referencia en el espacio cartesiano con las variables de control que se aplican en las articulaciones. Asimismo, se consideran modelos que relacionan las variaciones entre unas y otras (jacobiano). Los modelos inversos permiten determinar qué valores hay que aplicar a las variables de control para que las posiciones y orientaciones del sistema de referencia sean las deseadas. En este caso, suelen presentarse múltiples soluciones que es necesario resolver teniendo en cuenta consideraciones físicas adicionales sobre el movimiento del robot y el espacio de trabajo.

En este capítulo se han considerado tanto robots manipuladores como robots móviles, presentando también un modelo jacobiano de éstos y teniendo en cuenta las restricciones no holónomas que se presentan en muchas configuraciones de vehículos robóticos.

El empleo de la herramienta MATLAB/Simulink que se presenta en el apéndice es muy útil para el estudio de los modelos cinemáticos. El uso de esta herramienta se ilustra con numerosos ejemplos de manipuladores y robots móviles.

## 4.9 REFERENCIAS

---

Bastin G., G. Campion y B. d'Andréa-Novel, 1997. "Theory of Robot Control", Capítulo 7. Editado por C. Canudas de Wit, B. Siciliano y G. Bastin. Springer.

Corke P.I., 1996. "A Robotics Toolbox for MATLAB". IEEE Robotics and Automation Magazine". Vol. 3, núm. 1, pp. 24-32.

- Craig J.J., 1986. "Introduction to Robotics. Mechanics and Control". Addison Wesley.
- Denavit J. y R.S. Hartenberg, 1955. "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices", *Journal of Applied Mechanism*.
- Fu K.S., R.C. González y C.S.G. Lee, 1988. "Robótica: control, detección, visión e inteligencia". McGraw-Hill.
- Mandow, A. y otros, 1996. "The Autonomous Mobile Robot Aurora for Greenhouse Operation". *IEEE Robotics and Automation Magazine*. Vol. 3, núm. 4, pp. 18-28.
- Ollero A. y otros, 1995. "The autonomous robots for spraying Aurora". Proceedings of the IARP Conference on Robotics in the Agriculture and the Food Industry, pp. 193-198. Toulouse, France.
- Ollero A., A. Simón, F. García y V. Torres, 1993. "Integrated mechanical design and modelling of a new mobile robot". Proceedings of the IFAC Symposium on Intelligent Components and Instruments for Control Applications. SICICA'92. Pergamon Press, pp. 461-466. Málaga, España.
- Paul R.P., 1981. "Robot manipulators: mathematics, programming and control". MIT Press, Massachusetts.
- Pieper D.L., 1968. "The kinematics of manipulators under computer control". Stanford Artificial Intelligence Lab., Stanford, California, Memo AIM-72.
- Rañada A., 1994. "Dinámica Clásica". Alianza Universidad Textos.
- Shin D.H. y S. Singh, 1990. "Position based path tracking for wheeled mobile robots". Proceedings of the IEEE International Workshop on Intelligent Robot and Systems, pp. 386-391. Tsukuba, Japón.
- Zhao y Bement, 1992. "Kinematics, Dynamics and Control of Wheeled Mobile Robots". Proceedings of the IEEE International Conference on Robotics and Automation, pp. 91-96. Niiza, Francia.

# Modelo dinámico

---

Este capítulo se dedica a introducir los modelos dinámicos empleados en robótica. El objetivo no es profundizar en el comportamiento dinámico con vistas al diseño o simulación de robots, sino exclusivamente tratar los fundamentos de estos modelos para su empleo en el control. Al igual que en capítulos precedentes se consideran tanto manipuladores como robots móviles. En este capítulo se emplean también las funciones MATLAB de la herramienta del apéndice, así como Simulink para la simulación del modelo dinámico del manipulador.

## 5.1 ASPECTOS DINÁMICOS EN ROBÓTICA

---

En el capítulo anterior se ha considerado la geometría de los robots, así como el movimiento sin analizar las fuerzas que lo producen. Como se sabe, las velocidades lineales y angulares vienen dadas por las fuerzas y pares que se aplican a la estructura mecánica y dependen también de las magnitudes de las masas y su distribución. Las relaciones involucradas constituyen el modelo dinámico del manipulador. La identificación del modelo dinámico de un robot es en general difícil. Asimismo, la consideración de dicho modelo complica el desarrollo e implantación del sistema de control. Por ello, muchos sistemas de control de robots han sido diseñados utilizando fundamentalmente el modelo cinemático. Los resultados son aceptables cuando los movimientos del robot son suaves no produciéndose aceleraciones significativas. Sin embargo, cuando es necesario realizar movimientos rápidos involucrándose aceleraciones y masas importantes, la consideración del modelo dinámico resulta imprescindible.

En este punto cabe señalar también que la evolución de la robótica ha llevado al desarrollo de máquinas en las que los aspectos dinámicos son los más importantes en el diseño del sistema de control, tal como sucede en los manipuladores flexibles o en vehículos robóticos como helicópteros.

En el estudio del comportamiento dinámico de los robots es necesario involucrar conceptos relacionados con las distribuciones de masas y las inercias. Un cuerpo rígido en un espacio tridimensional puede rotar alrededor de un número infinito de ejes. En la rotación alrededor de estos ejes es básico el concepto de *momento de inercia*. Su generalización conduce a la noción de *tensor de inercia* que se define con respecto a un sistema de referencia asociado al cuerpo rígido en estudio, como una matriz  $3 \times 3$  cuyos elementos son los momentos de inercia con respecto a cada uno de los ejes (elementos diagonales) y los productos de inercia entre cada dos ejes. Estos elementos se calculan como una integral de volumen que depende de las distribuciones de las masas alrededor de los ejes de rotación (Craig, 1986).

Los métodos de obtención del modelo dinámico emplean formulaciones tales como la de Lagrange-Euler (Uicker, 1964) o la de Newton-Euler (Lu, 1980). En este capítulo sólo se tratará con cierto detalle la segunda formulación en su versión iterativa.

En los apartados que siguen se comienza por considerar el modelo de una articulación simple de rotación y su generalización al caso de  $n$  articulaciones. A continuación se introduce la formulación de Lagrange-Euler. El apartado siguiente se dedica al método de Newton-Euler, presentando, en primer lugar, las expresiones básicas para el tratamiento vectorial de las aceleraciones y, a continuación, las ecuaciones de Newton-Euler en su forma iterativa. Más tarde se estudia el papel de cada uno de los términos del modelo dinámico de un robot manipulador de  $n$  articulaciones y se trata la obtención de las trayectorias articulares mediante su integración numérica. En el último apartado se trata la dinámica de los vehículos robóticos.

En los textos de Paul (1981), Craig (1986), Fu y otros (1988) y Barrientos y otros (1997) se trata con más detalle el modelo dinámico de los robots manipuladores.

## 5.2 ARTICULACIÓN SIMPLE DE ROTACIÓN

Considérese el posicionamiento de un eje correspondiente a una articulación caracterizada por un momento de inercia  $I$  y un coeficiente de fricción viscosa  $b$  en el que se aplica un par  $\tau$ , tal como se ilustra en la figura 5.1.

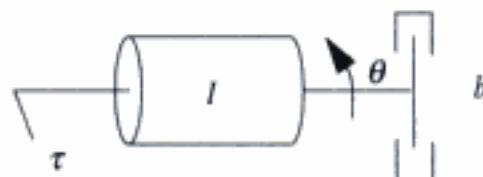


Figura 5.1: Articulación de rotación.

El par aplicado  $\tau$  y el ángulo girado  $\theta$  están relacionados mediante:

$$\tau = I\theta'' + b\theta' \quad (5.1)$$

El primer término del lado derecho representa el par de inercia, y el segundo el de rozamiento viscoso.

En este caso, el modelo es lineal. Tomando transformadas de Laplace para condiciones iniciales nulas, se tiene que la función de transferencia viene dada por:

$$\frac{\theta(s)}{\tau(s)} = \frac{1}{Is^2 + bs} \quad (5.2)$$

siendo  $\tau(s)$  la transformada de Laplace del par de entrada, y  $\theta(s)$  la transformada de Laplace del ángulo girado.

Este modelo simple puede complicarse si se considera, por ejemplo, un modelo más completo de las fricciones, incluyendo tanto la fricción viscosa como la de Coulomb. En este caso, se tendría el modelo no lineal

$$\tau = I\theta'' + b\theta' + c \operatorname{sgn}(\theta') \quad (5.3)$$

siendo  $\operatorname{sgn}(\theta')$  la función signo definida como:

$$\begin{aligned}\operatorname{sgn}(\theta') &= 1 && \text{para } \theta' > 0 \\ \operatorname{sgn}(\theta') &= -1 && \text{para } \theta' < 0\end{aligned} \quad (5.4)$$

Considérese ahora un péndulo invertido, que puede contemplarse también como un manipulador con una única articulación de rotación, como se ilustra en la figura 5.2.

Supuesto que la masa está concentrada en el extremo, se tendrá un momento de inercia de  $I = ml^2$ . La gravedad genera un par sobre el eje de la base que viene dado por  $mgl \cos \theta$ . Considérese ahora un modelo de las fricciones en el que se incluyen fricciones viscosas y de Coulomb. En este caso, el modelo vendrá dado por

$$\tau = ml^2\theta'' + mgl \cos \theta + b\theta' + c \operatorname{sgn}(\theta') \quad (5.5)$$

La ecuación anterior puede escribirse como

$$\tau = I\theta'' + G(\theta) + F(\theta') \quad (5.6)$$

en la cual

$$G(\theta) = mlg \cos \theta$$

representa los efectos gravitatorios y

$$F(\theta') = b\theta' + c \operatorname{sgn}(\theta')$$

son los pares generados por las fricciones.

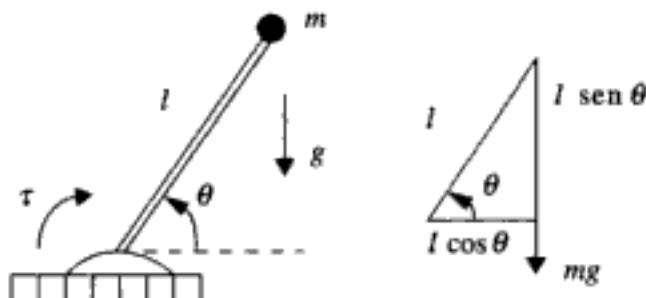


Figura 5.2: Manipulador con una articulación.

### 5.3 GENERALIZACIÓN: MODELO DE UN ROBOT MANIPULADOR CON $n$ ARTICULACIONES

Supóngase ahora que, continuando con la hipótesis de masas concentradas, se considera un robot con  $n$  articulaciones.

En este caso, cuando se estudia una articulación será necesario tener en cuenta los pares generados sobre ella por el resto de las articulaciones. Como se verá más adelante en este capítulo, el modelo dinámico completo de un manipulador de  $n$  articulaciones tiene la forma

$$\tau = M(\theta)\dot{\theta}'' + G(\theta) + V(\theta, \dot{\theta}') + F(\theta') \quad (5.7)$$

en la cual  $\tau$ ,  $G$ ,  $V$  y  $F$  son ahora vectores de  $n$  elementos, y  $M(\theta)$  es una matriz de  $n \times n$  a la que se conoce como matriz de masas del manipulador. Los elementos de  $M$  dependen de las distintas variables articulares y tienen como coeficientes expresiones en las que intervienen los valores de las masas concentradas y las longitudes entre ellas.

Los elementos del vector de términos gravitatorios  $G(\theta)$  también dependen de las diferentes variables articulares.

Por lo que respecta al vector de fricciones  $F(\theta')$ , sus elementos dependen sólo de la variable articular correspondiente; es decir, pueden desacoplarse.

Por último, nótese que se ha añadido un vector  $V(\theta)$  que permite considerar los pares generados por las fuerzas centrífugas y de Coriolis.

### 5.4 FORMULACIÓN DE LAGRANGE-EULER

Se basa en una interpretación de la energía involucrada (ver, por ejemplo, Lewis y otros, 1993).

En el ejemplo del manipulador con una única articulación del apartado 5.2, la energía cinética viene dada por:

$$E_c = \frac{1}{2} I\dot{\theta}'^2 \quad (5.8)$$

y la energía potencial por

$$E_p = mgl \operatorname{sen} \theta \quad (5.9)$$

La función lagrangiana se define como

$$L = E_c - E_p = \frac{1}{2} ml^2 \dot{\theta}'^2 - mgl \operatorname{sen} \theta \quad (5.10)$$

La formulación dinámica se basa en la ecuación

$$\tau = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}'} - \frac{\partial L}{\partial \theta} \quad (5.11)$$

siendo  $\tau$  el par aplicado.

Por tanto, derivando en (5.10) se tendrá

$$\frac{\partial L}{\partial \theta} = -mgl \cos \theta \quad (5.12)$$

$$\frac{\partial L}{\partial \dot{\theta}} = ml^2 \theta' \quad (5.13)$$

Sustituyendo en (5.11) se obtiene

$$\tau = ml^2 \theta'' + mgl \cos \theta \quad (5.14)$$

que coincide con (5.5) cuando no se consideran las fricciones.

La formulación (5.11) se generaliza para el caso de un manipulador con  $n$  articulaciones según:

$$\tau_i = \frac{d}{dt} \frac{\partial L}{\partial q'_i} - \frac{\partial L}{\partial q_i}, \quad i = 1 \dots n \quad (5.15)$$

siendo  $q_i$ ,  $i = 1 \dots n$  las variables articulares,  $q'_i$ ,  $i = 1 \dots n$  las velocidades articulares y  $L = E_c - E_p$  la diferencia entre la energía cinética total y la energía potencial total.

La expresión (5.15) continúa siendo válida para el movimiento de sólidos con  $n$  coordenadas generalizadas.

Conociendo las matrices de transformación entre los sistemas de referencia asociados a cada articulación y los tensores de inercia, es posible formular un algoritmo (Barrrientos y otros, 1997) para obtener las matrices  $M$ ,  $G$  y  $V$  en el modelo (5.7).

## 5.5 OBTENCIÓN DEL MODELO DINÁMICO MEDIANTE EL MÉTODO DE NEWTON-EULER

La obtención de un modelo dinámico como el (5.7) requiere generalizar a las aceleraciones los conceptos y notación introducidos en el capítulo anterior para las velocidades. En este apartado se comienza por presentar las expresiones correspondientes a las aceleraciones. A continuación, se presenta el método de Newton-Euler en su formulación iterativa, ilustrándose su aplicación en el manipulador plano con dos articulaciones de rotación estudiado en capítulos precedentes. En este apartado, se emplea también la misma notación y desarrollos que en Craig (1986).

### 5.5.1 Aceleraciones

#### *Aceleración lineal*

Se recuerda que la velocidad  ${}^A V_Q$  del punto  $Q$  con respecto al sistema de referencia  $\{A\}$  conocidos la posición  ${}^B Q$  y velocidad  ${}^B V_Q$  con respecto a  $\{B\}$  cuando los orígenes de  $\{A\}$  y  $\{B\}$  son coincidentes, viene dada por:

$${}^A V_Q = {}^B R {}^B V_Q + {}^A \Omega_B \times {}^B R {}^B Q \quad (5.16)$$

que puede escribirse como

$$\frac{d}{dt}({}^B R {}^B Q) = {}^B R {}^B V_Q + {}^A \Omega_B \times {}^B R {}^B Q \quad (5.17)$$

Si se sustituye la posición por la velocidad, la ecuación (5.17) puede escribirse también para la aceleración como:

$$\frac{d}{dt}({}^B R {}^B V_Q) = {}^B R {}^B V'_Q + {}^A \Omega_B \times {}^B R {}^B V_Q \quad (5.18)$$

Tomando derivadas con respecto al tiempo en (5.16) se obtiene:

$${}^A V'_Q = \frac{d}{dt}({}^B R {}^B V_Q) + {}^A \Omega'_B \times {}^B R {}^B Q + {}^A \Omega_B \times \frac{d}{dt}({}^B R {}^B Q) \quad (5.19)$$

Sustituyendo (5.17) en el último término del lado derecho de (5.19) se obtiene:

$$\begin{aligned} {}^A V'_Q &= {}^B R {}^B V'_Q + {}^A \Omega_B \times {}^B R {}^B V_Q + {}^A \Omega'_B \times {}^B R {}^B Q + {}^A \Omega_B \times ({}^B R {}^B V_Q + {}^A \Omega_B \times {}^B R {}^B Q) \\ &= {}^B R {}^B V'_Q + 2 {}^A \Omega_B \times {}^B R {}^B V_Q + {}^A \Omega'_B \times {}^B R {}^B Q + {}^A \Omega_B \times ({}^A \Omega_B \times {}^B R {}^B Q) \end{aligned} \quad (5.20)$$

Si los orígenes de {A} y {B} no son coincidentes, hay que añadir el término  ${}^A V_{ORG_B}$  que representa la aceleración lineal del origen:

$$\begin{aligned} {}^A V'_Q &= {}^A V_{ORG_B} + {}^B R {}^B V'_Q + 2 {}^A \Omega_B {}^B R {}^B V_Q + {}^A \Omega'_B \times {}^B R {}^B Q + \\ &\quad + {}^A \Omega_B \times ({}^A \Omega_B \times {}^B R {}^B Q) \end{aligned} \quad (5.21)$$

Para el caso particular en que el punto  $Q$  no varía con respecto a  $B$  se tendrá

$${}^B V_Q = {}^B V'_Q = 0 \quad (5.22)$$

de donde

$${}^A V'_Q = {}^A V_{ORG_B} + {}^A \Omega_B \times ({}^A \Omega_B \times {}^B R {}^B Q) + {}^A \Omega'_B \times {}^B R {}^B Q \quad (5.23)$$

expresión de interés cuando hay que calcular la aceleración lineal de un manipulador.

### Aceleración angular

Supóngase que el sistema {B} está rotando con relación a {A} con  ${}^A \Omega_B$ , y el sistema {C} rotando con respecto a {B} con  ${}^B \Omega_C$ . Para calcular  ${}^A \Omega_C$  se suman los vectores expresados con relación al sistema {A}:

$${}^A \Omega_C = {}^A \Omega_B + {}^B R {}^B \Omega_C \quad (5.24)$$

Derivando con respecto al tiempo se obtiene:

$${}^A\dot{\Omega}'_C = {}^A\dot{\Omega}'_B + \frac{d}{dt}({}^A_R {}^B\Omega_C) = {}^A\dot{\Omega}'_B + {}^A_R {}^B\dot{\Omega}'_C + {}^A\Omega_B \times {}^A_R {}^B\Omega_C \quad (5.25)$$

expresión de interés para calcular la aceleración angular del manipulador.

### 5.5.2 Ecuaciones de Newton-Euler. Formulación iterativa

#### Ecuaciones básicas

Considérese un cuerpo rígido de masa total  $m$  cuyo centro de masas tiene una aceleración  $v'_c$ . De acuerdo con la ecuación de Newton, la fuerza que actúa en el centro de masas viene dada por:

$$F = mv'_c \quad (5.26)$$

Por otra parte, si un cuerpo rígido rota con velocidad angular  $\omega$  y aceleración angular  $\omega'$ , el momento angular que actúa en el cuerpo viene dado por la ecuación de Euler:

$$N = {}^C I \omega' + \omega \times {}^C I \omega \quad (5.27)$$

siendo  ${}^C I$  el tensor de inercias del cuerpo en un sistema  $\{C\}$  cuyo origen está en el centro de masas.

Se trata ahora de aplicar estas ecuaciones para calcular los pares que hay que aplicar a las articulaciones de un robot para que la posición  $\theta$ , velocidad  $\omega = \dot{\theta}$  y aceleración  $\omega' = \ddot{\theta}$  de las articulaciones sean las deseadas. Se supondrá que la distribución de masas es conocida.

El problema puede resolverse de forma iterativa realizando, en primer lugar, iteraciones hacia fuera, desde la base a la última articulación, con objeto de calcular velocidades y aceleraciones y, a continuación, se calculan las fuerzas y pares actuando en cada articulación.

En las iteraciones hacia fuera se calculan la velocidad angular y las aceleraciones lineales y angulares del centro de masas, desde la articulación 1 a la  $n$ . Para ello es necesario propagar las velocidades angulares de la misma forma que se trató en el modelo cinemático. Considérese el enlace, o eslabón de la cadena cinemática,  $i$  entre las articulaciones  $i$  e  $i+1$  tal como se ilustra en la figura 5.3. La relación entre las velocidades angulares en los cuadros de referencia  $\{i\}$  e  $\{i+1\}$  asociados a las articulaciones viene dada por:

$${}^{i+1}\omega_{i+1} = {}^{i+1}R^i \omega_i + \theta'_{i+1} {}^{i+1}\hat{Z}_{i+1} \quad (5.28)$$

siendo  ${}^{i+1}\hat{Z}_{i+1}$  el vector unitario según el eje de la articulación  $\{i+1\}$

Asimismo, es necesario propagar las aceleraciones. Teniendo en cuenta los resultados del apartado anterior, la relación entre las aceleraciones de dos articulaciones consecutivas viene dada por:

$${}^{i+1}\omega_{i+1} = {}^{i+1}\omega_i + \theta^r_{i+1} {}^{i+1}\hat{Z}_{i+1} + {}^{i+1}R^i \omega_i \times \theta'_{i+1} {}^{i+1}\hat{Z}_{i+1} \quad (5.29)$$

de donde

$${}^{i+1}\omega_{i+1} = {}^{i+1}R^i \omega_i + \theta^r_{i+1} {}^{i+1}\hat{Z}_{i+1} + {}^{i+1}R^i \omega_i \times \theta'_{i+1} {}^{i+1}\hat{Z}_{i+1} \quad (5.30)$$

Cuando la articulación  $i+1$  es prismática (5.30) se simplifica a:

$${}^{i+1}\omega'_{i+1} = {}^{i+1}R^i \omega'_i \quad (5.31)$$

Por otra parte, la aceleración lineal del origen de  $\{i+1\}$  viene dada por:

$${}^{i+1}v'_{i+1} = {}^{i+1}R [{}^i\omega_i \times {}^iP_{i+1} + {}^i\omega_i \times ({}^i\omega_i \times {}^iP_{i+1}) + {}^iV_i] \quad (5.32)$$

siendo  ${}^iP_{i+1}$  el vector que expresa la posición del origen del sistema  $\{i+1\}$  en el sistema  $\{i\}$  tal como se muestra en la figura 5.3.

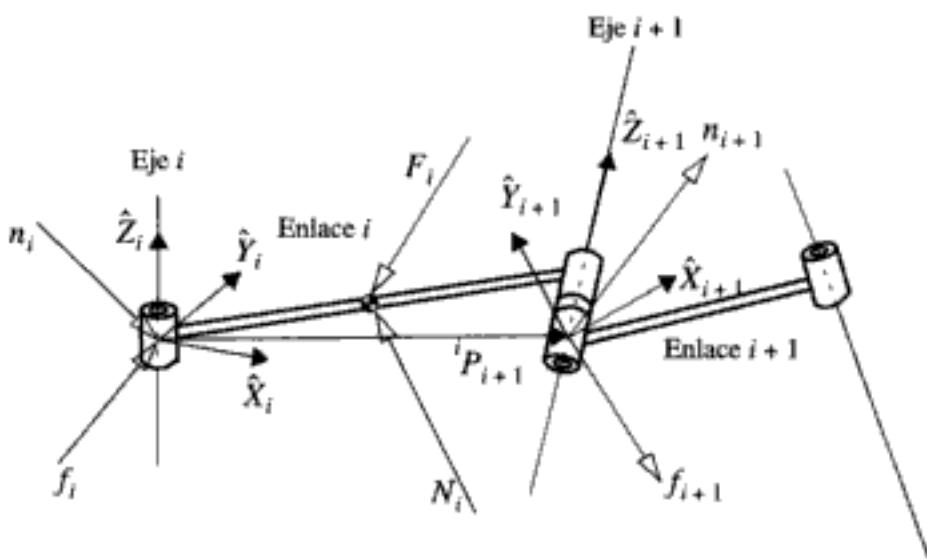


Figura 5.3: Balance de fuerzas en dos articulaciones consecutivas.

Si la articulación es prismática, hace falta añadir los términos correspondientes al desplazamiento  $d_i$  según la variable articular. Utilizando la expresión general de la aceleración lineal es posible escribir

$$\begin{aligned} {}^{i+1}v'_{i+1} &= {}^{i+1}R ({}^i\omega_i \times {}^iP_{i+1} + {}^i\omega_i \times ({}^i\omega_i \times {}^iP_{i+1}) + {}^iV_i) + \\ &+ 2{}^{i+1}\omega_{i+1} \times d'_{i+1} {}^{i+1}\hat{Z}_{i+1} + d''_{i+1} {}^{i+1}\hat{Z}_{i+1} \end{aligned} \quad (5.33)$$

Por último, suponiendo que existe un sistema  $\{C_i\}$  solidario a cada segmento, con su origen situado en el centro de masas y con la misma orientación que la  $\{i\}$ , la aceleración de este centro viene dada por la misma expresión que la del origen, pero en este caso expresada en el sistema  $\{i\}$ :

$${}^i v_{ci} = {}^i \omega_i \times {}^i P_{ci} + {}^i \omega_i \times ({}^i \omega_i \times {}^i P_{ci}) + {}^i v_i \quad (5.34)$$

En el segmento 1 las expresiones se simplifican ya que

$${}^0 \omega_0 = {}^0 \omega_0 = 0 \quad (5.35)$$

Una vez calculada la aceleración del centro de masas, pueden aplicarse las ecuaciones de Newton-Euler para determinar la fuerza y el par que actuarán en el centro de masas de cada enlace:

$$F_i = m v'_{ci} \quad (5.36)$$

$$N_i = {}^{C_i} I \omega_i + \omega_i \times {}^{C_i} I \omega_i \quad (5.37)$$

A continuación se procede a realizar las iteraciones hacia dentro; es decir, desde la última articulación a la base.

Sean  $f_i$  y  $n_i$  la fuerza y el par ejercidos en  $i$  por  $i-1$ . El equilibrio de fuerzas en el eslabón  $i$  (figura 5.3) es:

$${}^i F_i = {}^i f_i - {}_{i+1} {}^i R \cdot {}^{i+1} f_{i+1} \quad (5.38)$$

El equilibrio de pares en el centro de masas es:

$${}^i N_i = {}^i n_i - {}^i n_{i+1} + (-{}^i P_{ci}) \times {}^i f_i - ({}^i P_{i+1} - {}^i P_{ci}) \times {}^i f_{i+1} \quad (5.39)$$

Por consiguiente, se obtiene:

$${}^i N_i = {}^i n_i - {}_{i+1} {}^i R \cdot {}^{i+1} n_{i+1} - {}^i P_{ci} \times {}^i F_i - {}^i P_{i+1} \times {}_{i+1} {}^i R \cdot {}^{i+1} f_{i+1} \quad (5.40)$$

Estas ecuaciones se aplican desde la última articulación hasta la primera. Si el extremo del manipulador está libre, se tendrá:  ${}^{N+1} f_{N+1} = 0$ ,  ${}^{N+1} n_{N+1} = 0$ ; en caso contrario, se comenzaría con los valores de las fuerzas y pares externos que actúan en el extremo.

Las expresiones en forma iterativa son:

$${}^i f_i = {}_{i+1} {}^i R \cdot {}^{i+1} f_{i+1} + {}^i F_i \quad (5.41)$$

$${}^i n_i = {}^i N_i + {}_{i+1} {}^i R \cdot {}^{i+1} n_{i+1} + {}^i P_{ci} \times {}^i F_i + {}^i P_{i+1} \times {}_{i+1} {}^i R \cdot {}^{i+1} f_{i+1} \quad (5.42)$$

El par en una articulación rotacional es la componente  $\hat{Z}$  del par aplicado por un enlace en el siguiente:

$$\tau_i = {}^i n_i^T \hat{Z}_i \quad (5.43)$$

Asimismo, la fuerza en una articulación prismática es:

$$\tau_i = {}^i f_i^T \hat{Z}_i \quad (5.44)$$

Estas ecuaciones permiten calcular las actuaciones necesarias en todas las articulaciones.

Las expresiones anteriores hacen posible formular un algoritmo iterativo. Este algoritmo tiene dos partes. En la primera, se calculan las velocidades y aceleraciones de los enlaces directamente desde el enlace 1 al  $n$ , aplicando las ecuaciones de Newton-Euler en cada enlace.

En la segunda, se calculan las fuerzas y pares de interacción y finalmente los pares que hay que aplicar en los actuadores de las articulaciones, todo ello comenzando en la articulación  $n$  y terminando en la 1.

Por consiguiente, el algoritmo para un robot manipulador con seis articulaciones consiste en ejecutar las siguientes operaciones (Craig, 1986):

### Iteraciones hacia fuera

Para  $i = 0$  hasta 5:

$${}^{i+1}\omega_{i+1} = {}^i\omega_i + \theta_{i+1} {}^{i+1}\hat{Z}_{i+1} \quad (5.45)$$

$${}^{i+1}\omega_{i+1} = {}^iR {}^i\omega_i + {}^iR {}^i\omega_i \times \theta_{i+1} {}^{i+1}\hat{Z}_{i+1} + \theta_{i+1} {}^{i+1}\hat{Z}_{i+1} \quad (5.46)$$

$${}^{i+1}v_{i+1} = {}^iR [{}^i\omega_i \times {}^iP_{i+1} + {}^i\omega_i \times ({}^i\omega_i \times {}^iP_{i+1}) + {}^iV_i] \quad (5.47)$$

$${}^{i+1}v_{C_{i+1}} = {}^{i+1}\omega_{i+1} \times {}^{i+1}P_{C_{i+1}} + {}^{i+1}\omega_{i+1} \times ({}^{i+1}\omega_{i+1} \times {}^{i+1}P_{C_{i+1}}) + {}^{i+1}V_{i+1} \quad (5.48)$$

$${}^{i+1}F_{i+1} = m_{i+1} {}^{i+1}V_{C_{i+1}} \quad (5.49)$$

$${}^{i+1}N_{i+1} = C_{i+1} I_{i+1} {}^{i+1}\omega_{i+1} + {}^{i+1}\omega_{i+1} \times C_{i+1} I_{i+1} {}^{i+1}\omega_{i+1} \quad (5.50)$$

### Iteraciones hacia dentro

Para  $i = 6$  hasta 1.

$${}^i f_i = {}_{i+1} {}^i R {}^{i+1} f_{i+1} + {}^i F_i \quad (5.51)$$

$${}^i n_i = {}^i N_i + {}_{i+1} {}^i R {}^{i+1} n_{i+1} + {}^i P_{c_i} \times {}^i F_i + {}^i P_{i+1} \times {}_{i+1} {}^i R {}^{i+1} f_{i+1} \quad (5.52)$$

$$\tau_i = {}^i n_i^T {}^i \hat{Z}_i \quad (5.53)$$

El efecto de las cargas gravitatorias en las articulaciones puede tenerse en cuenta haciendo  ${}^0 v_0 = G$ , siendo  $G$  el vector de la aceleración de la gravedad.

Las ecuaciones presentadas pueden utilizarse como método numérico o de forma analítica para desarrollar ecuaciones simbólicas.

El empleo como método numérico puede aplicarse de forma general para cualquier robot manipulador. Una vez que se han especificado tensores de inercia, masas de enlaces, vectores  $P_C$ , y matrices  $R$  de un determinado manipulador, las ecuaciones pueden aplicarse directamente para calcular los pares de las articulaciones en cualquier movimiento.

### Ejemplo 5.1

Considérese el manipulador con dos articulaciones de rotación que se muestra en la figura 5.4 (Craig, 1986). Para facilitar la resolución del problema se supondrá que las masas  $M_1, M_2$  están concentradas en los extremos de los enlaces, tal como se ilustra en la figura 5.4.

La posición de los centros de masas viene dada por los vectores:

$${}^1P_{C_1} = l_1 \hat{X}_1; \quad {}^2P_{C_2} = l_2 \hat{X}_2 \quad (5.54)$$

Teniendo en cuenta la hipótesis de las masas concentradas, el tensor de inercia en cada centro de masa es nulo:

$${}^{C_1}I_1 = {}^{C_2}I_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (5.55)$$

Con objeto de resolver el manipulador de acuerdo con las ecuaciones anteriores, se comienza por tener en cuenta que en el extremo libre:

$${}^3f_3 = 0; \quad {}^3n_3 = 0 \quad (5.56)$$

Asimismo, en la base fija se tiene:

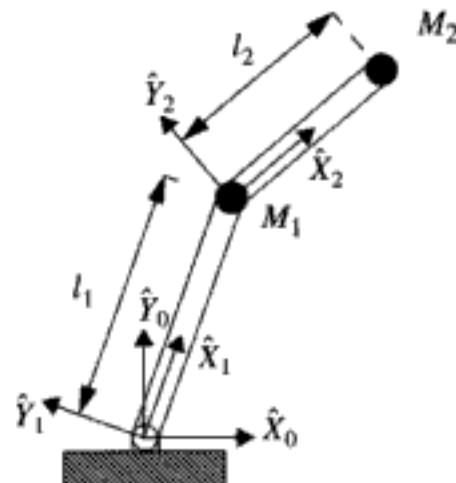


Figura 5.4: Manipulador plano con dos articulaciones de rotación.

$${}^0\omega_0 = 0; \quad {}^0\alpha_0 = 0 \quad (5.57)$$

La aceleración lineal en el origen es:

$${}^0v'_0 = g\hat{Y}_0 \quad (5.58)$$

siendo  $g$  la aceleración de la gravedad.

Por otra parte, se sabe que las matrices de rotación entre los cuadros correspondientes a dos enlaces consecutivos vienen dadas por:

$${}_{i+1}{}^iR = \begin{bmatrix} c_{i+1} & -s_{i+1} & 0 \\ s_{i+1} & c_{i+1} & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad {}^{i+1}{}^iR = \begin{bmatrix} c_{i+1} & s_{i+1} & 0 \\ -s_{i+1} & c_{i+1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.59)$$

Se procede ahora a la aplicación del procedimiento iterativo expuesto anteriormente.

### Iteraciones hacia fuera

En el eslabón 1 se tiene:

$${}^1\omega_1 = \theta'_1 {}^1\hat{Z}_1 = \begin{bmatrix} 0 \\ 0 \\ \theta'_1 \end{bmatrix}; \quad {}^1\omega'_1 = \theta''_1 {}^1\hat{Z}_1 = \begin{bmatrix} 0 \\ 0 \\ \theta''_1 \end{bmatrix} \quad (5.60)$$

$${}^1v'_1 = \begin{bmatrix} c_1 & s_1 & 0 \\ -s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix} = \begin{bmatrix} gs_1 \\ gc_1 \\ 0 \end{bmatrix} \quad (5.61)$$

Por consiguiente:

$$\begin{aligned} {}^1v'_{C1} &= \begin{bmatrix} 0 \\ 0 \\ \theta''_1 \end{bmatrix} \times \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \theta'_1 \end{bmatrix} \times \left( \begin{bmatrix} 0 \\ 0 \\ \theta'_1 \end{bmatrix} \times \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \right) + \begin{bmatrix} gs_1 \\ gc_1 \\ 0 \end{bmatrix} = \\ &= \begin{bmatrix} 0 \\ l_1\theta''_1 \\ 0 \end{bmatrix} + \begin{bmatrix} -l_1\theta'^2_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} gs_1 \\ gc_1 \\ 0 \end{bmatrix} = \begin{bmatrix} -l_1\theta'^2_1 + gs_1 \\ l_1\theta''_1 + gc_1 \\ 0 \end{bmatrix} \end{aligned} \quad (5.62)$$

$${}^1F_1 = m_1 {}^1v_{c1} = \begin{bmatrix} -m_1 l_1 \theta_1^2 + m_1 g s_1 \\ m_1 l_1 \theta''_1 + m_1 g c_1 \\ 0 \end{bmatrix} \quad (5.63)$$

$${}^1N_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (5.64)$$

En el segundo eslabón:

$${}^2\omega_2 = \begin{bmatrix} 0 \\ 0 \\ \theta'_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \theta'_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \theta'_1 + \theta'_2 \end{bmatrix} \quad (5.65)$$

$${}^2\omega_2 = \begin{bmatrix} 0 \\ 0 \\ \theta''_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \theta''_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \theta''_1 + \theta''_2 \end{bmatrix} \quad (5.66)$$

$$\begin{aligned} {}^2v_2 &= \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left( \begin{bmatrix} 0 \\ 0 \\ \theta''_1 \end{bmatrix} \times \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \theta'_1 \end{bmatrix} \times \left( \begin{bmatrix} 0 \\ 0 \\ \theta'_1 \end{bmatrix} \times \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \right) + \begin{bmatrix} g s_1 \\ g c_1 \\ 0 \end{bmatrix} \right) = \\ &= \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -l_1 \theta_1^2 + g s_1 \\ l_1 \theta''_1 s_2 - l_1 \theta_1^2 c_2 + g s_{12} \\ l_1 \theta''_1 c_2 + l_1 \theta_1^2 s_2 + g c_{12} \end{bmatrix} \end{aligned} \quad (5.67)$$

$${}^2v_{c2} = \begin{bmatrix} 0 \\ 0 \\ \theta''_1 + \theta''_2 \end{bmatrix} \times \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \theta'_1 + \theta'_2 \end{bmatrix} \times \left( \begin{bmatrix} 0 \\ 0 \\ \theta'_1 + \theta'_2 \end{bmatrix} \times \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} \right) + {}^2v_2 \quad (5.68)$$

$${}^2v_{c2} = \begin{bmatrix} 0 \\ l_2(\theta''_1 + \theta''_2) \\ 0 \end{bmatrix} + \begin{bmatrix} -l_2(\theta'_1 + \theta'_2)^2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} l_1 \theta''_1 s_2 - l_1 \theta_1^2 c_2 + g s_{12} \\ l_1 \theta''_1 c_2 + l_1 \theta_1^2 s_2 + g c_{12} \\ 0 \end{bmatrix} \quad (5.69)$$

$${}^2F_2 = m_2 {}^2v_{c2} = \begin{bmatrix} m_2 l_1 \theta''_1 s_2 - m_2 l_1 \theta'^2_1 c_2 + m_2 g s_{12} - m_2 l_2 (\theta'_1 + \theta'_2)^2 \\ m_2 l_1 \theta''_1 c_2 + m_2 l_1 \theta'^2_1 s_2 + m_2 g c_{12} + m_2 l_2 (\theta''_1 + \theta''_2) \\ 0 \end{bmatrix} \quad (5.70)$$

$${}^2N_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (5.71)$$

### Iteraciones hacia dentro

Se comienza con el eslabón 2:

$${}^2f_2 = {}^2R {}^3f_3 + {}^2F_2 = {}^2F_2 \quad (5.72)$$

$$\begin{aligned} {}^2n_2 &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} \times {}^2F_2 + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \\ &= \begin{bmatrix} 0 \\ 0 \\ m_2 l_1 l_2 \theta''_1 c_2 + m_2 l_1 l_2 s_2 \theta'^2_1 + m_2 l_2 g c_{12} + m_2 l_2^2 (\theta''_1 + \theta''_2) \end{bmatrix} \end{aligned} \quad (5.73)$$

Finalmente, en el eslabón 1 puede escribirse:

$${}^1f_1 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^2f_2 + \begin{bmatrix} -m_1 l_1 \theta'^2_1 + m_1 g s_1 \\ m_1 l_1 \theta''_1 + m_1 g c_1 \\ 0 \end{bmatrix} \quad (5.74)$$

$$\begin{aligned} {}^1n_1 &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + {}^1R {}^2n_2 + \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} -m_1 l_1 \theta'^2_1 + m_1 g s_1 \\ m_1 l_1 \theta''_1 + m_1 g c_1 \\ 0 \end{bmatrix} + \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \times {}^1R {}^2f_2 = \\ &= \begin{bmatrix} 0 \\ 0 \\ m_2 l_1 l_2 c_2 \theta''_1 + m_2 l_1 l_2 s_2 \theta'^2_1 + m_2 l_2 g c_{12} + m_2 l_2^2 (\theta''_1 + \theta''_2) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ m_1 l_1^2 \theta''_1 + m_1 l_1 g c_1 \end{bmatrix} + \\ &+ \begin{bmatrix} 0 \\ 0 \\ m_2 l_1^2 \theta''_1 - m_2 l_1 l_2 s_2 (\theta'_1 + \theta'_2)^2 + m_2 l_1 g s_2 s_{12} + m_2 l_1 l_2 c_2 (\theta''_1 + \theta''_2) + m_2 l_1 g c_2 c_{12} \end{bmatrix} \end{aligned} \quad (5.75)$$

de donde, considerando las componentes según  $\hat{Z}$ , se llega a que los pares de control son:

$$\begin{aligned}\tau_1 &= m_2 l_2^2 (\theta''_1 + \theta''_2) + m_2 l_1 l_2 c_2 (2\theta''_1 + \theta''_2) + (m_1 + m_2) l_1^2 \theta''_1 - m_2 l_1 l_2 s_2 \theta''_2^2 - \\ &\quad - 2m_2 l_1 l_2 s_2 \theta'_1 \theta'_2 + m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1 \\ \tau_2 &= m_2 l_1 l_2 c_2 \theta''_1 + m_2 l_1 l_2 s_2 \theta''_1^2 + m_2 l_2 g c_{12} + m_2 l_2^2 (\theta''_1 + \theta''_2)\end{aligned}\quad (5.76)$$

que pueden expresarse en forma matricial como:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} l_2^2 m_2 + 2l_1 l_2 m_2 c_2 + l_1^2 (m_1 + m_2) & l_2^2 m_2 + l_1 l_2 m_2 c_2 \\ l_2^2 m_2 + l_1 l_2 m_2 c_2 & l_2^2 m_2 \end{bmatrix} \begin{bmatrix} \theta''_1 \\ \theta''_2 \end{bmatrix} + \\ + \begin{bmatrix} m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1 \\ m_2 l_2 g c_{12} \end{bmatrix} + \begin{bmatrix} -m_2 l_1 l_2 s_2 \theta''_2^2 - 2m_2 l_1 l_2 s_2 \theta'_1 \theta'_2 \\ m_2 l_1 l_2 s_2 \theta''_1^2 \end{bmatrix} \quad (5.77)$$

Este modelo tiene la estructura

$$\tau = M(\theta) \theta'' + G(\theta) + V(\theta, \theta') \quad (5.78)$$

que coincide con la de la expresión (5.7) excepto en el término de las fricciones  $F(\theta)$ , que es nulo en el ejemplo. Nótese la forma que tiene la matriz de masas  $M(\theta)$ . El término  $V(\theta, \theta')$  depende en el ejemplo exclusivamente del cuadrado de las velocidades de rotación de las articulaciones. Finalmente, nótese como en  $G(\theta)$  aparecen los términos que incluyen la aceleración de la gravedad  $g$ .

En el apartado siguiente se estudia con algo más de detalle este modelo.

Considérese ahora el empleo de la herramienta MATLAB del apéndice. La función `me` permite obtener el modelo dinámico. Para ello simplemente hay que pasarle una matriz de parámetros `dyn` con el formato apropiado.

Adicionalmente, es necesario pasarle un vector con la aceleración de la gravedad que sufre el manipulador, así como los valores de las variables articulares y de las velocidades y aceleraciones articulares. La función `rne` devuelve como resultado los pares ejercidos en cada articulación.

En este ejemplo, la matriz de parámetros `dyn` se forma utilizando los vectores y las matrices:

$$\begin{aligned}{}^1P_{C_1} &= \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} & {}^2P_{C_2} &= \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} \\ {}^{C_1}I_1 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & {}^{C_2}I_2 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\end{aligned}\quad (5.79)$$

donde  $C_1$  y  $C_2$  son, respectivamente, los centros de masas de los enlaces 1 y 2. Asimismo, hay que tener presente que la aceleración de la gravedad a la que se ve sometido el manipulador es  $-g \hat{Y}_0$ .

Para obtener el modelo dinámico simbólico bastan las siguientes líneas:

```

syms t1 t2 real; % Variables articulares θ1 y θ2
syms td1 td2 real; % Velocidades articulares θ'₁ y θ'₂
syms tdd1 tdd2 real; % Aceleraciones articulares θ''₁ y θ''₂
syms l1 l2 real; % Longitudes de los enlaces (l1 y l2)
syms m1 m2 real; % Masas de los enlaces (m1 y m2)
syms g real; % Aceleración de la gravedad
% Escribimos la matriz con los parámetros dinámicos del manipulador
dyn = [0 0 t1 0 0 m1 l1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0;
       0 l1 t2 0 0 m2 l2 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0];
q = [t1 t2]; % Vector de variables articulares
qd = [td1 td2]; % Vector de velocidades articulares
qdd = [tdd1 tdd2]; % Vector de aceleraciones articulares
grav = [0 -g 0]; % Vector aceleración de la gravedad
tau = rne(dyn, q, qd, qdd, grav)
simple(tau)

```

De esta forma se obtiene una expresión simbólica de los pares que coincide exactamente con la que aparece en la ecuación (5.76).

También es posible evaluar cada uno de los términos que intervienen en la expresión del par por separado. Para ello se escribe:

$M = \text{inertia}$ (dyn, q)	% Matriz de masas $M(\theta)$
$G = \text{gravedad}$ (dyn, q, grav)	% Término gravitatorio $G(\theta)$
$V = \text{coriolis}$ (dyn, q, qd)	% Términos centrífugos y de Coriolis $V(\theta, \theta')$

## 5.6 ANÁLISIS DEL MODELO DINÁMICO DE LOS MANIPULADORES

Considérese nuevamente el modelo dinámico expresado en (5.7), al cual se añade ahora un nuevo término  $T_d$  para tener en cuenta dinámicas no modeladas y perturbaciones externas:

$$\tau = M(\theta)\theta'' + G(\theta) + F(\theta') + V(\theta, \theta') + T_d \quad (5.80)$$

siendo  $\tau = [\tau_1 \dots \tau_n]^T$  los pares ejercidos por los actuadores en las  $n$  articulaciones,  $\theta = (\theta_1, \dots, \theta_n)^T$  el vector de variables articulares,  $M(\theta)$  la matriz  $n \times n$  de masas del manipulador,  $V(\theta, \theta')$  los pares debidos a las fuerzas centrífugas y de Coriolis,  $F(\theta')$  los pares debidos a las fricciones y  $G(\theta)$  los pares debidos a la gravedad.

La dificultad principal de estos modelos es la determinación de los parámetros.

La matriz de masas  $M(\theta)$  del manipulador es una matriz simétrica, definida positiva. Está acotada superior e inferiormente ya que sus elementos dependen de las variables  $\theta_i$  de las articulaciones mediante funciones trigonométricas de senos y cosenos (funciones acotadas para cualquier valor del argumento) que aparecen siempre en el numerador. La matriz es siempre invertible y su inversa es también definida positiva y acotada. Puede demostrarse que la energía cinética del manipulador viene dada por:

$$\frac{1}{2} \dot{\theta}^T M(\theta) \dot{\theta} \quad (5.81)$$

Los términos centrífugos y de Coriolis  $V(\theta, \dot{\theta})$  son cuadráticos en  $\dot{\theta}$ . En particular, los términos en  $\dot{\theta}_i^2$  corresponden a las fuerzas centrífugas de las articulaciones, y los términos en  $\dot{\theta}_i \dot{\theta}_j$  a las fuerzas de Coriolis. Estas funciones pueden factorizarse como:

$$V(\theta, \dot{\theta}) = V_m(\theta, \dot{\theta}) \dot{\theta} \quad (5.82)$$

donde  $V_m(\theta, \dot{\theta})$  está relacionado con la derivada en el tiempo de la matriz de masas según:

$$V_m(\theta, \dot{\theta}) = \frac{1}{2}(M'(\theta, \dot{\theta}) + S) \quad (5.83)$$

donde  $S$  es una matriz que satisface la propiedad ("skew symmetric")  $x^T S x = 0$  para todo  $x \in \mathbb{R}^n$ . Esta matriz puede determinarse (Lewis y otros, 1993) según  $S = U - U^T$ , siendo  $U$  una matriz de dimensiones  $n \times n$  que se obtiene mediante  $U = (I_n \otimes \dot{\theta}^T)(\partial M / \partial \theta)$ , expresión en la cual el primer paréntesis del lado derecho es una matriz  $n \times n^2$  que resulta del producto de Kronecker de la matriz unidad por el vector de variables articulares (cada elemento de la matriz unidad se multiplica por la traspuesta del vector de variables articulares), y  $\partial M / \partial \theta = [\partial M / \partial \theta_1 \dots \partial M / \partial \theta_n]^T$  es una matriz  $n^2 \times n$  que se obtiene derivando la matriz de masas con respecto a las variables articulares.

Los términos de fricción  $F(\theta')$  son complejos y difíciles de determinar. En principio, dependen de las velocidades de las variables articulares. Un posible modelo es:

$$F(\theta') = \tau_{fv} + \tau_{fc} \quad (5.84)$$

donde  $\tau_{fv}$  es el par de fricción viscosa, que resulta ser proporcional a la velocidad  $\dot{\theta}'$ :

$$\tau_{fv} = k \dot{\theta}' \quad (5.85)$$

siendo  $k$  una constante, y  $\tau_{fc}$  el par de fricción de Coulomb:

$$\tau_{fc} = c \operatorname{sgn}(\dot{\theta}') \quad (5.86)$$

siendo  $\operatorname{sgn}(\dot{\theta}')$  la función signo. La fricción tiene un efecto local.  $F(\theta')$  está desacoplada y cada elemento puede escribirse como  $f_i(\theta'_i)$ . En algunos casos se supone que las fuerzas de fricción son sólo viscosas, con lo cual puede escribirse:

$$F(\theta') = F_m \theta' + T_{df} \quad (5.87)$$

donde  $F_m$  es una matriz diagonal de fricción viscosa y  $T_{df}$  representa otros efectos no estructurales de fricción (resonancias). A veces se modela también el efecto de las ex-

centricidades en los engranajes y otros efectos que hacen que la fricción dependa, además de la velocidad  $\dot{\theta}$ , de las propias variables articulares  $\theta$ , con lo cual se escribe  $F(\theta, \dot{\theta})$ .

Por último, los términos  $G(\theta)$  son en los que aparece la gravedad  $g$ . Estos términos dependen en senos y coseños de las variables articulares.  $G(\theta)$  está acotada para cualquier valor de  $\theta$ .

## 5.7 OBTENCIÓN DE LAS TRAYECTORIAS ARTICULARES

El modelo (5.7) relaciona los pares de control en las articulaciones con la evolución de las variables articulares. Si se pretende simular el comportamiento dinámico del manipulador ante determinados pares, es necesario resolver el modelo para obtener las trayectorias articulares. Obsérvese que, a partir de (5.7), se obtiene:

$$\ddot{\theta}^n = M^{-1}(\theta)[\tau - V(\theta, \dot{\theta}) - G(\theta) - F(\theta, \dot{\theta})] \quad (5.88)$$

Se recuerda que, como se indicó en la sección anterior, la matriz de masas  $M(\theta)$  es invertible.

Las trayectorias articulares pueden obtenerse mediante integración numérica. Suponiendo conocidas las condiciones iniciales:

$$\theta(0) = \theta_0; \quad \dot{\theta}(0) = 0; \quad \ddot{\theta}^n(0) = 0 \quad (5.89)$$

puede aplicarse (5.88) para obtener las derivadas segundas y, a partir de éstas, las velocidades y las posiciones empleando técnicas tales como las de Euler o las de Runge -Kutta.

El método presentado en el apartado 5.5 de este capítulo puede aplicarse para resolver cualquier manipulador sin más que conocer sus parámetros. En particular, el método de Newton-Euler iterativo es más eficiente que otras formulaciones existentes en la literatura. Sin embargo, si sólo se pretende resolver un determinado manipulador, resulta más apropiado emplear la forma cerrada (5.7).

Así, en el ejemplo del manipulador plano con dos articulaciones de rotación, en el procedimiento iterativo serían necesarias 153 multiplicaciones y 120 sumas, mientras que en forma cerrada sólo se requieren 30 multiplicaciones y 13 sumas (Craig, 1986). Obviamente, este número de operaciones es muy pequeño cuando se compara con el necesario en manipuladores con seis grados de libertad. Sin embargo, también en estos manipuladores se produce un ahorro importante del número de operaciones, del orden del 50% (Craig, 1986). No obstante, la obtención de (5.7) para un determinado manipulador es una tarea laboriosa. Como se sabe, en la actualidad existen programas, tales como el empleado en el ejemplo 5.1, que permiten obtener expresiones simbólicas.

### Ejemplo 5.2

Se trata de obtener la trayectoria que seguiría el manipulador plano de la figura 5.4 al aplicarle los pares:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} 30 + (1 + e^{-t})10 \sin t \\ 10 + 10 \cos t \end{bmatrix} \quad (5.90)$$

Suponiendo  $m_1 = 2 \text{ kg}$ ,  $m_2 = 1 \text{ kg}$ ,  $l_1 = 3 \text{ m}$  y  $l_2 = 1 \text{ m}$ , las líneas que deben escribirse son:

```

m1 = 2; m2 = 1; l1 = 3; l2 = 1;
% Cargamos los parámetros del manipulador
dyn = [0 0 0 0 m1 l1 0 0 0 0 0 0 0 0 0 0 1 0 0 0;
       0 l1 0 0 0 m2 l2 0 0 0 0 0 0 0 0 0 0 1 0 0 0];
% Integraremos la dinámica
[tsim,q,qd] = fdyn (dyn,0,10,'taucap5',[0 0],[0 0],[0 -9.81 0]);

```

Previamente es necesario crear un archivo `taucap5.m` con el par que se aplica:

```

function tau = taucap5 (t,x)
tau = [30+(1+exp(-t))*10*sin(t);
       10+10*cos(t)];

```

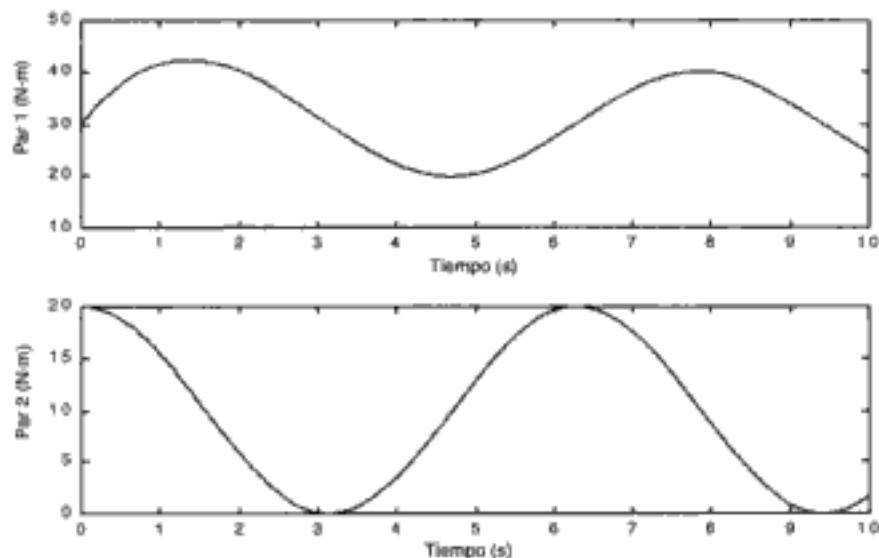
Los pares aplicados (figura 5.5) corresponden, evidentemente, a la ecuación (5.90). Al integrar la ecuación (5.88) se obtienen las variables articulares de la figura 5.6.

En la matriz  $Q$  se tienen almacenados los valores de las variables articulares, por lo que, para lograr una animación del manipulador, basta con escribir:

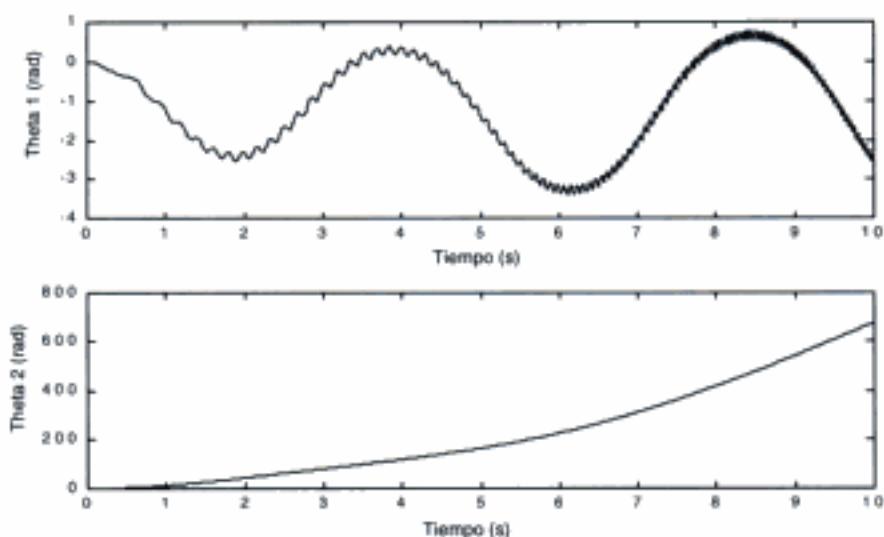
```

dh = [0 0 0 0; % Parámetros de Denavit-Hartenberg del manipulador
      0 3 0 0 0];
plotbot (dh, q, 'd')

```



**Figura 5.5:** Pares aplicados al manipulador plano.

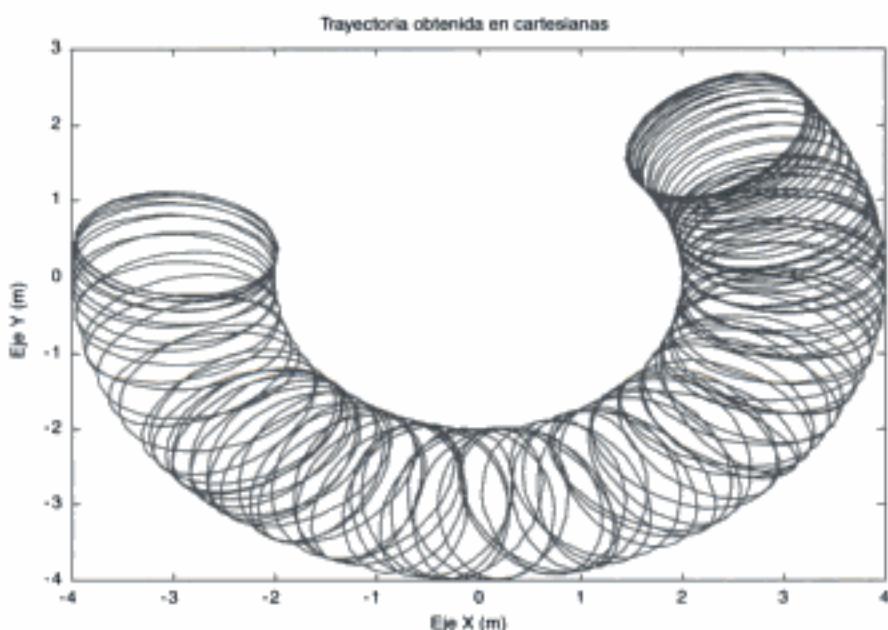


**Figura 5.6:** Valores de las variables articulares correspondientes a la aplicación de los pares.

Como resultado, el extremo del segmento de longitud  $l_2$  seguirá una trayectoria dada por:

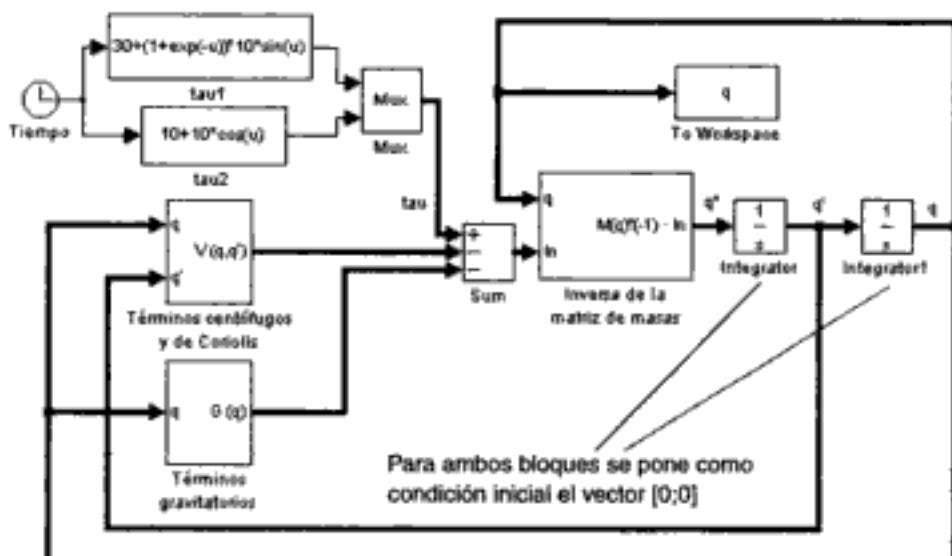
$$\begin{aligned}x &= l_1 c_1 + l_2 c_{12} \\y &= l_1 s_1 + l_2 s_{12}\end{aligned}\tag{5.91}$$

que se representa en la figura 5.7.



**Figura 5.7:** Representación en coordenadas cartesianas de las posiciones del extremo del manipulador.

## Obtención de las trayectorias articulares

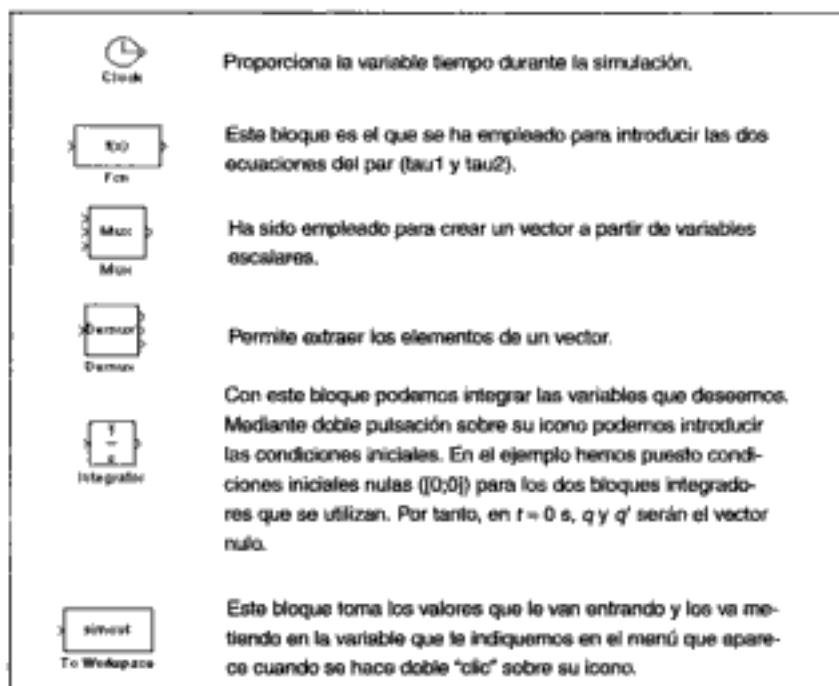


**Figura 5.8:** Diagrama de bloques que se emplea para resolver el ejemplo 5.2.

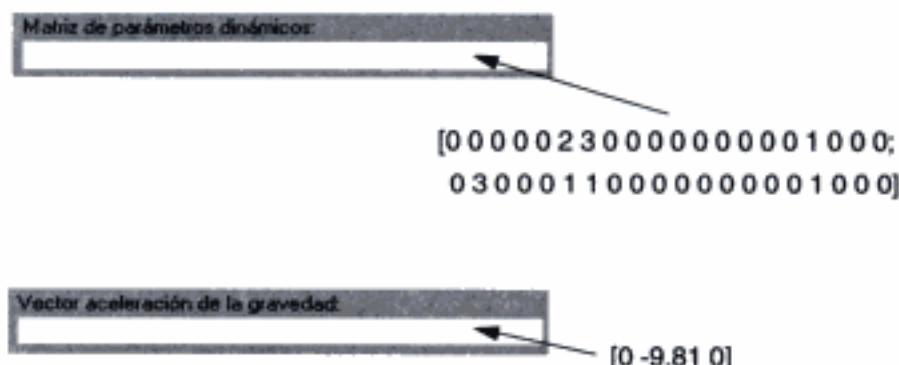
Es posible también emplear Simulink para la simulación del comportamiento dinámico del manipulador. Para ello hay que dibujar un diagrama de bloques que implemente la ecuación:

$$\theta'' = M^{-1}(\theta)[\tau - V(\theta, \theta') - G(\theta)]$$

En la figura 5.8 se presenta un posible diagrama en el cual aparece una serie de bloques. En la figura 5.9 se muestra una breve explicación acerca de cada uno de estos bloques.



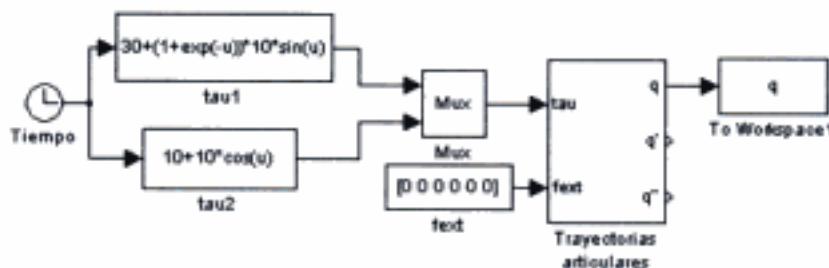
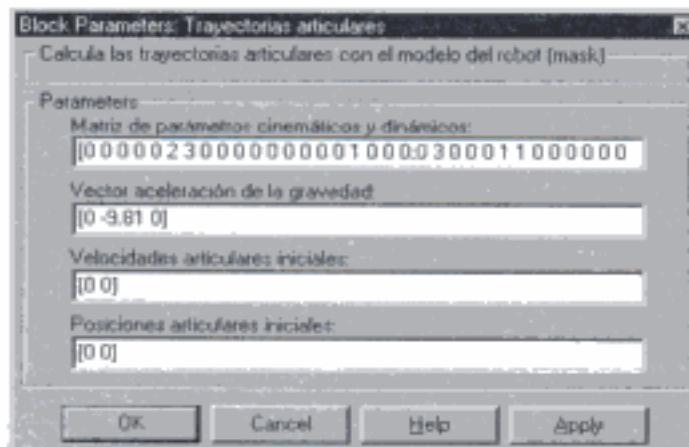
**Figura 5.9:** Notas aclaratorias sobre algunos bloques del diagrama de la figura 5.8.

**Figura 5.10:** Parámetros en el ejemplo 5.2.

Para introducir los parámetros en los bloques correspondientes al modelo dinámico hay que hacer doble “clic” sobre ellos. En este ejemplo los valores de los parámetros son los que se indican en la figura 5.10.

Es posible también usar un esquema alternativo basado en la utilización del bloque “Trayectorias Articulares” (figura 5.11).

El cuadro de diálogo correspondiente a dicho bloque es el que se muestra en la figura 5.12.

**Figura 5.11:** Esquema alternativo al de la figura 5.8.**Figura 5.12:** Cuadro de diálogo del bloque *Trayectorias Articulares*.

Tanto si se usa un esquema como otro, para arrancar la simulación basta con seleccionar *Start* dentro del menú *Simulation*. Una vez que finalice la simulación (o se la pare), se obtendrán los resultados en el espacio de trabajo (variable  $q$ ).

### Ejemplo 5.3

Se estudia ahora la dinámica de un manipulador plano con tres articulaciones de rotación (figura 5.13). Los valores numéricos de los parámetros del manipulador son los siguientes:

$$\begin{aligned}l_1 &= l_2 = 0,8 \text{ m} \\m_1 &= 4,7 \text{ kg}, m_2 = 2,6 \text{ kg}, m_3 = 1,1 \text{ kg} \\g &= 9,8 \text{ m/s}^2\end{aligned}$$

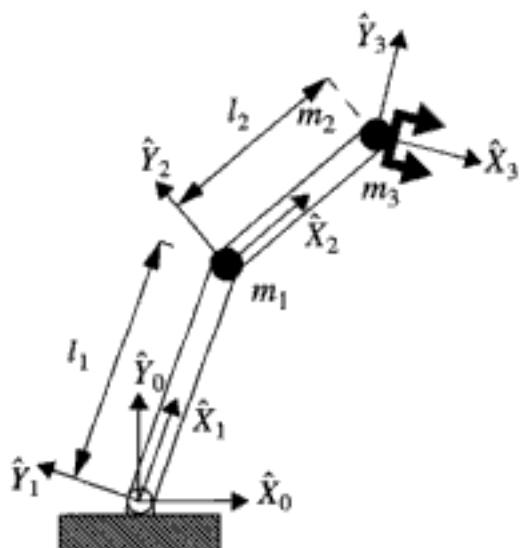


Figura 5.13: Manipulador plano del ejemplo 5.3.

Los parámetros de Denavit-Hartenberg son:

$i$	$\alpha_{i-1}$	$a_{i-1}$	$\theta_i$	$d_i$
1	0	0	$\theta_1$	0
2	0	$l_1$	$\theta_2$	0
3	0	$l_2$	$\theta_3$	0

Para las dos primeras articulaciones se supone que la masa está concentrada al final del enlace, siendo nulos los tensores de inercia (ver ejemplo 5.1). Si el centro de masas de la tercera articulación está localizado en el origen del sistema de referencia {3}, el tensor de inercia para este tercer enlace es:

$${}^c_1 I_3 = \begin{bmatrix} 0,05 & 0 & 0 \\ 0 & 0,1 & 0 \\ 0 & 0 & 0,1 \end{bmatrix} (\text{kg/m}^2) \quad (5.92)$$

Por tanto, las coordenadas de los centros de masas y los tensores de inercia del manipulador son:

$$\begin{aligned} {}^1 P_{C_1} &= \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} & {}^2 P_{C_2} &= \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} & {}^3 P_{C_3} &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ {}^c_1 I_1 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & {}^c_2 I_2 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & {}^c_3 I_3 &= \begin{bmatrix} 0,05 & 0 & 0 \\ 0 & 0,1 & 0 \\ 0 & 0 & 0,1 \end{bmatrix} \end{aligned} \quad (5.93)$$

Se tienen ahora todos los datos necesarios para construir la matriz de parámetros cinemáticos y dinámicos.

El listado que permitiría obtener los pares que se ejercen en las tres articulaciones será:

```

syms t1 t2 t3 real;
syms td1 td2 td3 real;
syms tdd1 tdd2 tdd3 real;
dyn = [0 0 0 0 0 4.7 0.8 0 0 0 0 0 0 0 0 0 1 0 0 0;
       0 0.8 0 0 0 2.6 0.8 0 0 0 0 0 0 0 0 0 1 0 0 0;
       0 0.8 0 0 0 1.1 0 0 0 0.05 0.1 0.1 0 0 0 0 1 0 0 0];
q = [t1 t2 t3];
qd = [td1 td2 td3];
qdd = [tdd1 tdd2 tdd3];
grav = [0 -9.8 0];
tau = rne(dyn, q, qd, qdd, grav)

```

#### Ejemplo 5.4

Se trata de simular el comportamiento dinámico del manipulador del ejemplo 5.3. Se estudiará qué sucede cuando no se aplica ningún par en las articulaciones del brazo y se le deja evolucionar a partir de diferentes posiciones iniciales y en diferentes condiciones.

- a) En primer lugar se establece como posición inicial la definida por:

$$[\theta_1 \ \theta_2 \ \theta_3] = [-\pi/2 \ 0 \ 0]$$

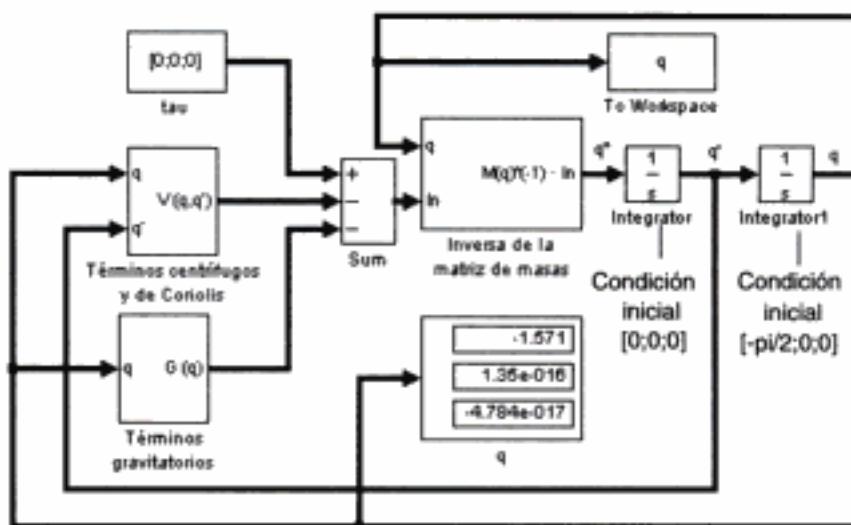


Figura 5.14: Diagrama de bloques Simulink para el ejemplo 5.4 en los casos a y b.

El correspondiente esquema de Simulink es el que se muestra en la figura 5.14. Para introducir la  $\tau_{au}$  se ha empleado un bloque *Constant* de la librería *Sources*. Se ha usado un bloque *Display* de la librería *Sinks* con objeto de ir visualizando los valores que se van obteniendo para  $q$  en cada instante.

Los valores de los parámetros de los bloques del modelo dinámico son los que se indican en la figura 5.15.

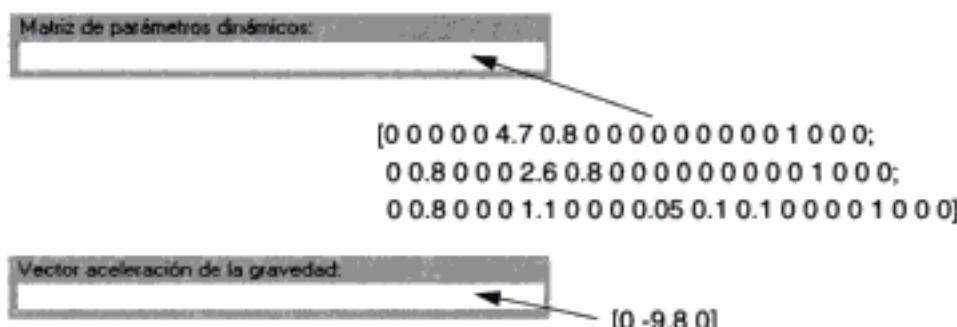


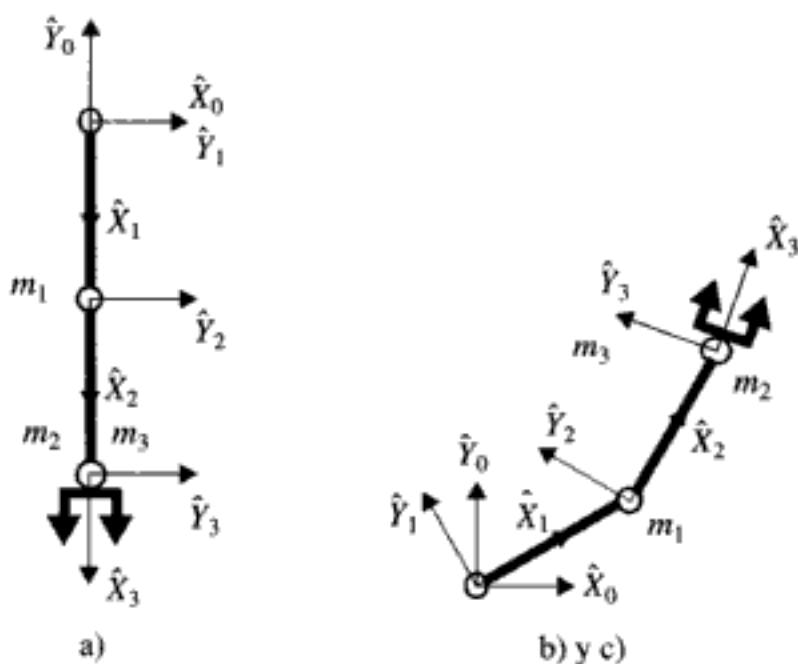
Figura 5.15: Parámetros en el ejemplo 5.4.

Todos los valores calculados para las variables articulares son almacenados en una variable  $q$  del espacio de trabajo mediante el bloque *To Workspace* de la librería *Sinks*. Así, una vez terminada la simulación, puede escribirse:

```

dh = [0 0 0 0 0;
      0 0.8 0 0 0;
      0 0.8 0 0 0];
plotbot (dh, q, 'd')
  
```

para obtener una animación del movimiento del brazo.



**Figura 5.16:** Posiciones iniciales para los apartados a, b y c del ejemplo 5.4.

Los resultados obtenidos son los que cabría esperar, ya que la posición inicial es la del brazo “caído” (figura 5.16a), y como no se aplica ningún par en las articulaciones, el brazo permanece en esa posición.

**b)** Si ahora se parte de una posición inicial (figura 5.16b) definida por:

$$[\theta_1 \ \theta_2 \ \theta_3] = [\pi/6 \ \pi/6 \ \pi/18]$$

es posible emplear el mismo esquema de la figura 5.14, sin más que cambiar la condición inicial del bloque “Integrator1” por la nueva posición inicial (es decir, por  $[\pi/6; \pi/6; \pi/18]$ ). Es posible obtener una animación del brazo de mayor calidad modificando algunas opciones en *Simulation → Parameters*. Por ejemplo, el tiempo de simulación se aumenta incrementando *Stop Time*, y el paso de simulación puede hacerse más pequeño (de 0,01 s, por ejemplo) con el parámetro *Max Step Size*.

Una vez finalizada la simulación, pueden emplearse las mismas líneas que en el apartado a) para ejecutar la animación del robot. El resultado es el esperado; es decir, que el brazo cae por acción de la gravedad, y luego continúa moviéndose indefinidamente debido a que no hay fricciones.

**c)** Se introducen ahora fricciones viscosas en las articulaciones y se realiza la simulación con las mismas condiciones iniciales que en el caso b). La fricción viscosa en cada articulación será de 5,0 N·m s.

El diagrama de bloques en este caso es el que se representa en la figura 5.17.

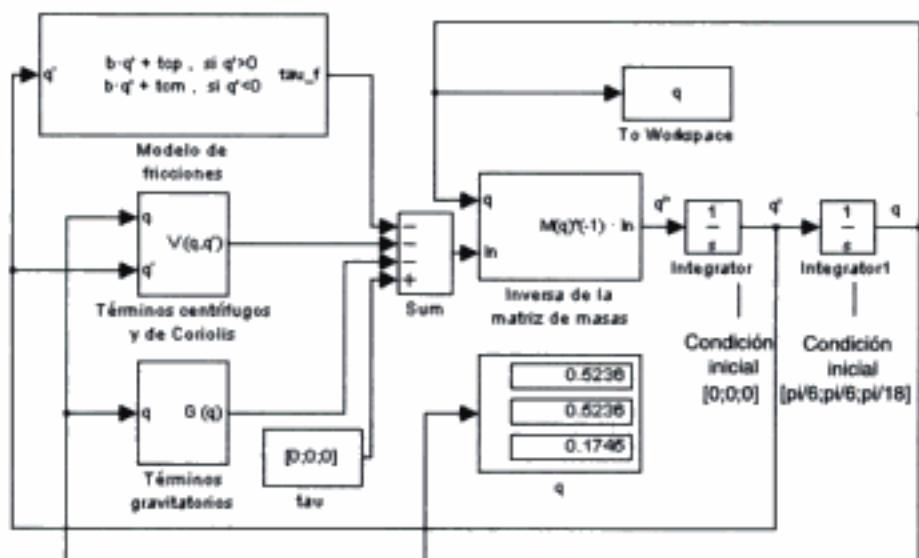


Figura 5.17: Esquema para simular el ejemplo 5.4 en el caso c.

En el esquema de la figura 5.17 hay un bloque nuevo correspondiente a las fricciones. Los parámetros de dicho bloque son los que se indican en la figura 5.18.

Para la simulación y posterior animación pueden seguirse los mismos pasos indicados en el apartado b). En este caso, el brazo cae por acción de la gravedad desde la posición inicial y después se comporta de modo parecido a un péndulo, hasta acabar por detenerse.

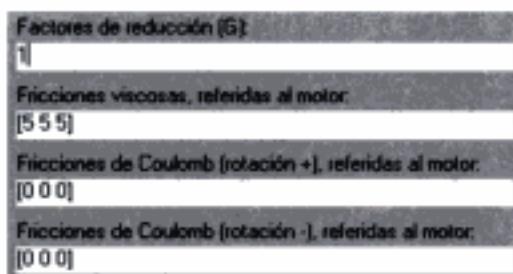


Figura 5.18: Parámetros de fricción en el ejemplo 5.4.

## 5.8 MODELOS DINÁMICOS DE VEHÍCULOS ROBÓTICOS

El estudio de la dinámica de los vehículos con ruedas hace necesario considerar tanto la dinámica del chasis del vehículo como la interacción con el terreno.

Los vehículos convencionales tienen sistemas de tracción y direccionamiento relativamente desacoplados que suelen aproximarse por modelos linealizados de primer o segundo orden. En particular, el sistema de direccionamiento de vehículos tales como

ROMEO 3R o RAMI (ver capítulo 2) se aproxima mediante un sistema de primer orden tal como:

$$\gamma' = -\frac{\gamma}{\tau} + \frac{K}{\tau} u \quad (5.94)$$

siendo  $\gamma$  la curvatura del vehículo,  $u$  la señal de control y  $\tau$  la constante de tiempo. Asimismo, el sistema de tracción también puede aproximarse por el sistema lineal de primer orden:

$$v' = -\frac{v}{\tau_v} + \frac{K}{\tau_v} v_d \quad (5.95)$$

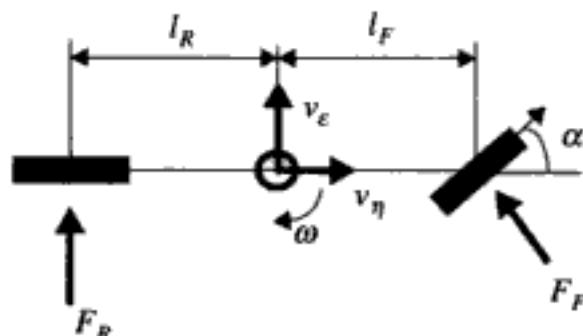
siendo  $v$  la velocidad del vehículo,  $v_d$  la velocidad ordenada y  $\tau_v$  la constante de tiempo.

Por lo que respecta a la interacción del vehículo con el terreno, se han utilizado modelos con muy diferente grado de complejidad. En Shin (1990) se estudia el desplazamiento lateral del vehículo empleando el modelo de la bicicleta para las ecuaciones cinemáticas y las siguientes ecuaciones dinámicas:

$$v'_e = -\frac{C_{af} + C_{ar}}{mv_\eta} v_e + \left( \frac{l_R C_{ar} - l_F C_{af}}{mv_\eta} - v_\eta \right) \omega + \frac{C_{af}}{m} \alpha \quad (5.96)$$

$$\omega' = \frac{l_R C_{ar} - l_F C_{af}}{mv_\eta} v_e + \left( \frac{l_R^2 C_{ar} + l_F^2 C_{af}}{mv_\eta} - v_\eta \right) \omega + \frac{C_{af}}{I} \alpha \quad (5.97)$$

siendo  $v_e$  la velocidad lateral del vehículo, en la dirección perpendicular al eje longitudinal,  $v_\eta$  la velocidad longitudinal,  $\alpha$  el ángulo de direccionamiento,  $\omega = \dot{\theta}$  la velocidad angular,  $m$  la masa del vehículo,  $I$  el momento de inercia del vehículo,  $C_{af}$  y  $C_{ar}$  los coeficientes de los neumáticos delanteros y traseros del vehículo, y  $l_F$  y  $l_R$  las distancias de los neumáticos delanteros y traseros al centro de masas del vehículo según el eje longitudinal, tal como se muestra en la figura 5.19. El modelo puede linearizarse para velocidad longitudinal  $v_\eta$  constante.



**Figura 5.19:** Ilustración de variables y parámetros en el modelo dinámico del vehículo (5.96)-(5.97).

Las ecuaciones anteriores, tratadas conjuntamente con las cinemáticas del modelo de la bicicleta que se presentó en el capítulo 4, permiten formular un modelo dinámico del vehículo en el que el vector de variables de estado es:

$$\begin{bmatrix} x & y & v_e & \theta & \theta' & \alpha \end{bmatrix}^T, \quad (5.98)$$

siendo  $x$ ,  $y$ ,  $\theta$  las coordenadas globales del centro del vehículo y el ángulo de orientación del eje longitudinal del vehículo (ángulo de los ejes de coordenadas locales del vehículo con los globales). Este modelo se ha utilizado (Shin, 1990) para el seguimiento autónomo de caminos, problema que consiste esencialmente en el control del desplazamiento lateral del vehículo y que se tratará en el capítulo 9.

En Campion y otros, (1996) y Heredia (1999) se presentan clasificaciones de modelos cinemáticos y dinámicos de robots móviles con ruedas.

## 5.9 CONCLUSIONES

Aunque la mayor parte de los sistemas de control de robots convencionales se han diseñado teniendo en cuenta fundamentalmente el modelo cinemático, el estudio del comportamiento dinámico resulta imprescindible cuando se desea diseñar robots de altas prestaciones, teniendo en cuenta aceleraciones significativas y masas considerables. Existen también un buen número de robots, tales como los robots flexibles, o los robots con comportamientos inestables, en los cuales la consideración de la dinámica es imprescindible.

Existen diferentes métodos que permiten formular el modelo dinámico de un robot manipulador. En este capítulo sólo se introducen los fundamentos del método de Lagrange y se considera con un cierto detalle el método Newton-Euler iterativo. La aplicación de este método permite obtener las velocidades y aceleraciones de los diferentes enlaces desde la base al extremo del manipulador y, una vez obtenidas, partiendo del extremo, es posible ir calculando los pares y fuerzas de los diferentes enlaces. Asimismo, la aplicación del método de Newton-Euler, permite formular un modelo simbólico del manipulador en cuestión, que, posteriormente, puede ser integrado para obtener el comportamiento (aceleraciones, velocidades y posiciones) ante determinados pares de control en las articulaciones. En el capítulo se presentan programas en MATLAB que permiten obtener el modelo simbólico y esquemas de Simulink para integrar el modelo realizando simulaciones del comportamiento dinámico. Por otra parte, en el capítulo, se realiza también un análisis de la influencia de los distintos términos del modelo dinámico lo cual tiene interés para el diseño del sistema de control.

En el capítulo se considera sólo de forma introductoria el modelo dinámico de robots móviles con ruedas, presentando modelos simplificados de interés para su empleo en el sistema de control en tiempo real. El planteamiento de modelos más completos requiere el estudio detallado de la interacción con el terreno, lo cual se sale del objetivo de este capítulo. Por otra parte, hay que poner de manifiesto que los modelos que involucran la interacción con el terreno en tres dimensiones son, en general, demasiado complejos para su empleo en un controlador en tiempo real del robot.

## 5.10 REFERENCIAS

- Barrientos A. y otros, 1997. "Fundamentos de Robótica". McGraw-Hill.
- Campion y otros, 1996. "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots". IEEE Trans. on Robotics and Automation. Vol. 12, núm. 1, pp. 47-62.
- Craig J.J., 1986. "Introduction to Robotics. Mechanics and Control". Addison Wesley.
- Fu y otros, 1988. "Robótica: control, detección, visión e inteligencia". McGraw-Hill.
- Heredia G., 1999. "Seguimiento autónomo de caminos en robots móviles. Análisis de estabilidad". Tesis Doctoral.
- Lewis F.L. y otros, 1993. "Control of Robots Manipulators". Macmillan, Inc.
- Luh J.Y.S., M.W. Walker, and R.P. Paul, 1980. "On-Line Computational Scheme for Mechanical Manipulators". Trans. of the ASME Journal of Dynamic Systems, Measurement, and Control.
- Paul R.P., 1981. "Robot manipulators: mathematics, programming and control". MIT Press.
- Shin D.H., 1990. "High performance tracking of explicit paths by roadworthy mobile robots". Tesis doctoral, Carnegie Mellon University.
- Uiker J.J., J. Denavit y R.S. Hartenberg, 1964. "An iterative method for the displacement analysis of spatial mechanisms", Trans. of the ASME Journal of Applied Mech. Vol. 31, Serie B.

# Arquitecturas para control de robots

---

En este capítulo se comienza por introducir algunas consideraciones sobre las distintas funciones que pueden considerarse incluidas en un marco general de control de robots. A continuación se tratan las especificaciones para el control de robots, incluyéndose tanto los manipuladores como los robots móviles. El resto del capítulo está dedicado a presentar aspectos significativos relacionados específicamente con las arquitecturas de control, incluyendo sus requerimientos, tipos y características del diseño, así como la implantación, presentando diversos ejemplos de arquitecturas y en especial las de los robots móviles RAM-1 y AURORA.

## 6.1 INTRODUCCIÓN: FUNCIONES BÁSICAS Y DE CONTROL INTELIGENTE

---

En la literatura sobre robótica y control de robots, son frecuentes términos tales como “funciones inteligentes” o “control inteligente de robots” (Kortenkamp y otros, 1998). Se pretende con ello hacer alusión a la incorporación de un mayor número de funciones autónomas de las que se consideran habitualmente como básicas.

Las *funciones básicas* son el servocontrol de las articulaciones y la generación de trayectorias, incluyéndose en ésta tanto la interpolación articular como la cartesiana en el espacio de trabajo.

Asimismo, en el caso de la teleoperación, se sustituiría la generación de trayectorias del control automático por el control manual del operador, que también puede ser en el espacio articular o en el espacio cartesiano.

Se entiende por funciones inteligentes las que permiten la realización parcial o totalmente autónoma de operaciones tales como la planificación de tareas, planificación de movimientos, percepción sensorial, y la reacción ante la presencia de obstáculos y condiciones no previstas en general.

Estas funciones no deben contemplarse de forma aislada, sino integradas en una arquitectura del sistema de control en la cual se incluya también la posibilidad de la teleoperación y las funciones de control de bajo nivel.

Si se considera que un determinado robot debe realizar funciones inteligentes, cabe plantearse qué funciones específicas deben incorporarse y con qué grado de autonomía.

En muchas aplicaciones que involucran control inteligente, el enfoque más adecuado consiste en una transición entre funciones autónomas y teleoperadas. Normalmente interesa que sean automáticas las funciones en las que la intervención humana es difícil por carecer de la suficiente capacidad de percepción, o bien por requerirse una respuesta más rápida de la que es capaz de generar un operador, o de mayor precisión que la que puede alcanzarse manualmente.

Asimismo, por razones de seguridad, y aunque se realice teleoperación, se cree necesario incorporar técnicas para reaccionar de forma autónoma a situaciones peligrosas en general y, en particular, para evitar colisiones inminentes en tiempo real.

## 6.2 ESPECIFICACIONES

Las especificaciones más frecuentes en manipuladores robóticos (ver, por ejemplo, Groover y otros, 1989) son las siguientes:

### *Accesibilidad*

Está directamente relacionada con el número de grados de libertad. Es claro que, cuanto mayor sea el número de grados de libertad, se tiene mayor accesibilidad. El problema es la dificultad de control que representa tener más grados de libertad. Esta dificultad es relevante en el caso de manipuladores redundantes con un gran número de grados de libertad.

### *Capacidad de carga*

La carga tiene una notable influencia en la respuesta dinámica del sistema. Si, además, las velocidades y aceleraciones son significativas, la consideración del modelo dinámico resulta imprescindible para el control apropiado del manipulador.

### *Rapidez de respuesta*

Esta especificación es importante para la productividad del robot, lo cual es de particular interés en robótica industrial. La rapidez de respuesta suele considerarse en el control de articulaciones y en la generación de trayectorias articulares, siendo menos frecuente en los niveles superiores de control.

En la práctica, la rapidez de la respuesta del robot está limitada por el par disponible en las articulaciones y por los tirones y oscilaciones no amortiguadas que se producen debido a los comportamientos dinámicos que no se consideran de forma apropiada en el control. En este punto conviene recordar que, a velocidad elevada, es imprescindible tener en cuenta los términos de fuerzas centrífugas y de Coriolis en el modelo dinámico del robot.

**Precisión y características relacionadas**

Existen distintas características relacionadas. La primera es la *resolución*, que, para el robot en su conjunto, se entiende como la menor variación posible en el posicionamiento del efecto final. La resolución viene dada por los sensores de posición empleados, los actuadores y los componentes electrónicos, incluyendo tanto los de almacenamiento y procesamiento digital como los de conversión analógica-digital y digital-analógica.

Se entiende por *precisión* la diferencia entre la posición ordenada y la realmente alcanzada por el efecto final. El origen de la imprecisión es tanto la limitación de la resolución como las dificultades en calibración, los desgastes, las deformaciones y los propios errores de control. Por consiguiente, la cifra que expresa la magnitud de la precisión es mayor que la de la resolución.

La precisión es una especificación típica del servocontrol de articulaciones, pero también se utiliza en la generación de trayectorias y niveles superiores de control.

Se entiende por *repetibilidad* la capacidad para volver a la misma posición a la que fue dirigido en las mismas condiciones. La repetibilidad es, obviamente, muy importante en robótica industrial, en la cual el robot debe repetir un gran número de veces la misma operación en distintas piezas.

La falta de repetibilidad se debe normalmente a problemas mecánicos en transmisiones, rozamientos, histéresis, zonas muertas, etc.

La cifra que indica la repetibilidad es típicamente menor que la de la precisión, como se ilustra en la figura 6.1.



**Figura 6.1:** Precisión ( $p$ ) y repetibilidad ( $r$ ).  $T$  es la posición alcanzada en el primer posicionamiento y  $R$  la alcanzada en la repetición.

**Interacción con el entorno**

Estas restricciones hacen referencia a operaciones concretas tales como el posicionamiento preciso requerido en aplicaciones como las de agarre, inserción y manipulación en general, el seguimiento preciso de trayectorias, la detección suficientemente rápida de obstáculos imprevistos, y la interacción involucrando esfuerzos como los que se requieren en operaciones tales como atornillar, cortar o pulir.

## 6.3 REQUERIMIENTOS GENERALES DE LA ARQUITECTURA

En el diseño de la arquitectura se emplean frecuentemente requerimientos sobre programabilidad, eficiencia, capacidad de evolución, autonomía, adaptabilidad y reactividad.

### 6.3.1 Programabilidad

Con la programabilidad se hace alusión a la posibilidad de ejecutar múltiples tareas, a la especificación de tareas como acciones que es necesario ejecutar, o simplemente formulando objetivos que es necesario cumplir, y a la posibilidad de planificar acciones de acuerdo con el estado actual del robot y de su entorno.

Por ejemplo, en aplicaciones de manipuladores, es claro que la arquitectura deberá soportar operaciones tales como la transferencia de piezas, ensamblado y actuación con determinadas herramientas para aplicaciones específicas. Se trata de que sea posible expresar estas operaciones de forma flexible y cómoda.

Existen distintas técnicas para secuenciar operaciones, con mayor o menor intervención del operador humano, desde la especificación completa de la secuencia de movimientos y sus parámetros, a la simple especificación de objetivos generándose automáticamente las especificaciones de las operaciones y su secuenciamiento para cumplir con dicho objetivo.

### 6.3.2 Eficiencia

Se refiere tanto a la eficiencia en la realización de una tarea (tiempo de ejecución, precisión) como a los recursos empleados (potencia, "hardware" y "software" necesarios).

En principio, la eficiencia se consigue disponiendo de modelos apropiados del robot y de su entorno, y empleando planificadores eficientes que permitan determinar soluciones óptimas de acuerdo con estos modelos.

En cualquier caso, es interesante llegar a definir, de la forma más precisa posible, límites sobre los tiempos de ejecución y recursos empleados para realizar las funciones de control necesarias para la ejecución de las operaciones especificadas.

Conviene también poner de manifiesto que la eficiencia está también relacionada con modos de comportamiento consistentes con planes y objetivos, no puramente dirigido por estímulos del entorno o, en otros términos, que la reacción ante estímulos externos esté también guiada por los objetivos que se pretenden cumplir.

### 6.3.3 Capacidad de evolución

Es un requerimiento de especial interés en arquitecturas concebidas para proyectos innovadores, o de demostración, en los cuales pueden producirse revisiones significativas, e incorporación de nuevas tecnologías, equipos y componentes.

### 6.3.4 Grado de autonomía

El grado de autonomía es otra especificación básica en el diseño de la arquitectura. En general, todas las propuestas de arquitecturas de control inteligente consideran un elevado grado de autonomía. Como se ha mencionado anteriormente, es frecuente contemplar una transición gradual desde un control esencialmente teleoperado hacia comportamientos autónomos. Puede considerarse también un modo dual con posibilidad de teleoperación y funcionamiento autónomo. A este respecto conviene insistir aquí en los comentarios de la introducción sobre las funciones autónomas básicas cuando existen dificultades para teleoperación eficiente y razones de seguridad en funcionamiento teleoperado.

Obviamente, el funcionamiento dual (autónomo/teleoperado) tiene implicaciones en el diseño de la arquitectura, especialmente en los niveles superiores. Por ello, se hace necesario considerar tanto las arquitecturas de control inteligente propuestas teniendo en cuenta una operación esencialmente autónoma, como las que contemplan de forma explícita la teleoperación.

### 6.3.5 Fiabilidad

Está relacionada con la no dependencia de un único sistema o subsistema para todas sus acciones. De esta forma, se tiende a incluir redundancias en cuanto a funciones, estableciendo formas diferentes de realizar la misma tarea. La redundancia puede llegar hasta duplicar recursos "hardware" para aumentar la seguridad de funcionamiento.

Nótese que la redundancia en los sensores está también relacionada con la disminución de la incertidumbre asociada a todo proceso de percepción.

En algunos sistemas móviles, tales como los aéreos o los terrestres con patas, las redundancias del "hardware" están restringidas por las limitaciones de carga a bordo. No obstante, además de las redundancias oportunas en el "hardware" de la estación de control y supervisión, parece interesante, por razones de seguridad, considerar también ciertas redundancias en el "hardware" a bordo, y entre éste y el de supervisión.

### 6.3.6 Adaptabilidad

Está relacionada con la flexibilidad, entendida como capacidad de funcionamiento en entornos diversos, poco conocidos, con capacidad de responder ante cualquier eventualidad.

En general, la adaptabilidad puede concebirse como la modificación de comportamientos de acuerdo con las circunstancias en las que se encuentre el robot o, en otros términos, como la capacidad de razonamiento para analizar situaciones y producir/seleccionar las reacciones oportunas ante cambios en el entorno y, en general, condiciones de trabajo del robot. Por consiguiente, la reactividad de una determinada arquitectura viene dada por la capacidad de detectar temporalmente acontecimientos y reaccionar de acuerdo con el contexto y la tarea, lo cual está relacionado fundamentalmente con la capacidad sensorial y la capacidad de generación suficientemente rápida de respuestas.

\* \* \*

Algunos de los requerimientos mencionados en los apartados anteriores pueden llegar a ser conflictivos entre sí. Así, la eficiencia en la ejecución de una determinada tarea empleando unos determinados recursos debe estudiarse frente a la capacidad de ejecutar múltiples tareas, capacidad de evolución, robustez, adaptabilidad y reactividad.

En particular, en el diseño actual de arquitecturas, se discute el compromiso entre la eficiencia (robots dirigidos por objetivos o planes) y la adaptabilidad y capacidad de reacción (robots más dirigidos por datos), lo cual ha dado origen a diversos tipos de arquitecturas que se consideran de forma introductoria en el próximo apartado.

## 6.4 TIPOS BÁSICOS DE ARQUITECTURAS SEGÚN REACTIVIDAD

Desde el punto de vista de la planificación, existen diferentes arquitecturas diseñadas teniendo en cuenta especificaciones sobre eficiencia, tiempo de respuesta ante acontecimientos no previstos, y disponibilidad de información del entorno.

En un extremo se sitúan las arquitecturas orientadas esencialmente a soportar una planificación *puramente estratégica*, utilizando modelos del entorno suficientemente precisos, suponiendo que la situación en la que va a ejecutarse el plan puede ser conocida a priori, y sin tener demasiado en cuenta la posibilidad de generalizar el comportamiento. Son arquitecturas funcionales basadas en planificación con modelo centralizado para verificación de la información suministrada por los sensores y generación de acciones. Esta información es utilizada por un planificador para producir una secuencia de acciones (plan).

En el otro extremo se sitúan las arquitecturas orientadas esencialmente a suministrar flexibilidad, adaptabilidad y capacidad de reacción ante condiciones no previstas. Las arquitecturas *puramente reactivas* implementan una estrategia de control como una colección de pares condiciones-acciones. El sistema consiste en una colección de reglas reactivas, conteniendo una descripción interna mínima del estado. El sistema no mantiene un modelo interno, no realiza las típicas estrategias de búsqueda en árbol, y simplemente busca y ordena la acción apropiada para cada conjunto de lecturas de sensores. Se basa en un acoplamiento directo entre sensores y actuaciones mediante bucles rápidos de realimentación.

Las arquitecturas basadas en *comportamientos* ("behaviors") (Arkin, 1998) pueden considerarse como una evolución de las puramente reactivas. Los comportamientos describen la forma en que se reacciona ante un determinado estado de los sensores, y las actuaciones se determinan de forma más elaborada que la simple consulta a una tabla de correspondencias, utilizándose distintas formas de representaciones internas y realizándose con ellas cálculos para decidir qué acción hay que realizar. De esta forma, cabe mencionar comportamientos tales como "evitar obstáculos", o "cerrar pinza". En las arquitecturas basadas en comportamientos, se trata de emplear controladores sencillos fáciles de depurar y utilizar. En estas arquitecturas, suele existir una colección de comportamientos paralelos ejecutándose concurrentemente.

Entre las arquitecturas de comportamientos cabe mencionar la denominada "*subsumption*". Esta arquitectura se debe a Brooks (1986) y está basada en la idea de descomponer un problema utilizando comportamientos que ejecutan tareas. La arquitec-

tura consiste en capas de comportamientos. Cada capa supone un nivel más elaborado de comportamiento que puede inhibir o actuar sobre los niveles inferiores. Cada comportamiento puede representarse como una máquina de estados finitos que tiene como entradas señales de sensores y produce como salidas señales a los actuadores. Se trata de definir, probar y depurar comportamientos sin necesidad de modificarlos posteriormente. Para dotar de una mayor competencia al sistema, se van añadiendo capas de comportamientos, tal como se ilustra en la figura 6.2, sin modificar los comportamientos de capas inferiores. El término “subsumption” se refiere a la capacidad de los comportamientos de mayor nivel para subsumir la funcionalidad de los comportamientos de las capas inferiores.

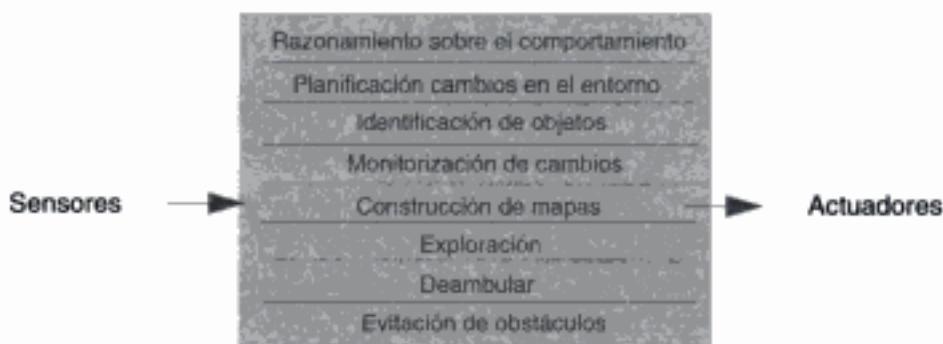


Figura 6.2: Jerarquía de comportamientos.

En Arkin (1992a) se presenta una arquitectura basada en comportamientos para la navegación de robots móviles. MacKenzie y Arkin (1993) definen el concepto de *esquema* para combinar comportamientos sencillos que producen una acción coherente para la navegación de robots móviles. Existen también arquitecturas basadas en comportamientos para el control de robots manipuladores (Stein y Paul, 1994).

El control basado en comportamientos encuentra dificultades cuando se involucran comportamientos conflictivos, en cuyo caso es necesaria una representación de mayor nivel para resolver los conflictos. Otra dificultad se presenta cuando en la ejecución de una tarea es necesario emplear representaciones simbólicas.

Las denominadas *arquitecturas híbridas* ofrecen un compromiso entre las puramente reactivas y las orientadas a planes. Se emplea un sistema reactivo para control de bajo nivel, y un planificador para toma de decisiones de un nivel superior. Se separa el sistema de control en dos o más partes comunicadas, pero básicamente independientes. En la mayor parte de los casos, el proceso reactivo de bajo nivel se encarga de funciones de seguridad, mientras que el de alto nivel utiliza un planificador para seleccionar la acción. En realidad, estas arquitecturas son básicamente extensiones directas de las basadas en planificación.

Connell y Viola (1990) presentan una arquitectura híbrida combinando una arquitectura de tipo “subsumption” con una capa de manipulación simbólica. En Mataric (1992a, 1992b, 1997) se integran también representaciones simbólicas de tareas para implantar comportamientos dirigidos por un objetivo “*goal driven*”. Arkin (1992b) presenta una arquitectura híbrida que permite resolver los conflictos mencionados anteriormente.

En aplicaciones en las que se conoce el entorno y puede modelarse, parece que la arquitectura de su sistema de control debería ser esencialmente orientada a la ejecución de planes, con cierta capacidad reactiva para funciones tales como la detección de colisiones inminentes que pueden producirse debido a las imprecisiones en la ejecución de planes.

## 6.5 APROXIMACIÓN AL DISEÑO DE LA ARQUITECTURA

En el diseño de la arquitectura del sistema de control inteligente de un robot puede distinguirse entre el diseño funcional y la implantación.

En el primer caso, se trata de definir una estructura lógica que cumpla con requerimientos de diseño, estudiando las interrelaciones entre las distintas funciones que es necesario realizar, definiendo niveles de jerarquización, y estableciendo un compromiso entre planes para cumplir objetivos y reacciones ante situaciones no previstas en los planes.

En el segundo, se trata esencialmente de asignar unos recursos "software" y "hardware" a las funciones especificadas en la arquitectura funcional. Esta segunda fase es igualmente imprescindible en cualquier proyecto de robótica que deba concluir con una realización práctica.

### 6.5.1 Diseño funcional de la arquitectura

El enfoque típico consiste en partir de las tareas que es necesario realizar y descomponerlas en funciones, definiendo las interacciones entre ellas. Los problemas típicos que hay que abordar en el diseño de arquitecturas de control inteligente son los siguientes:

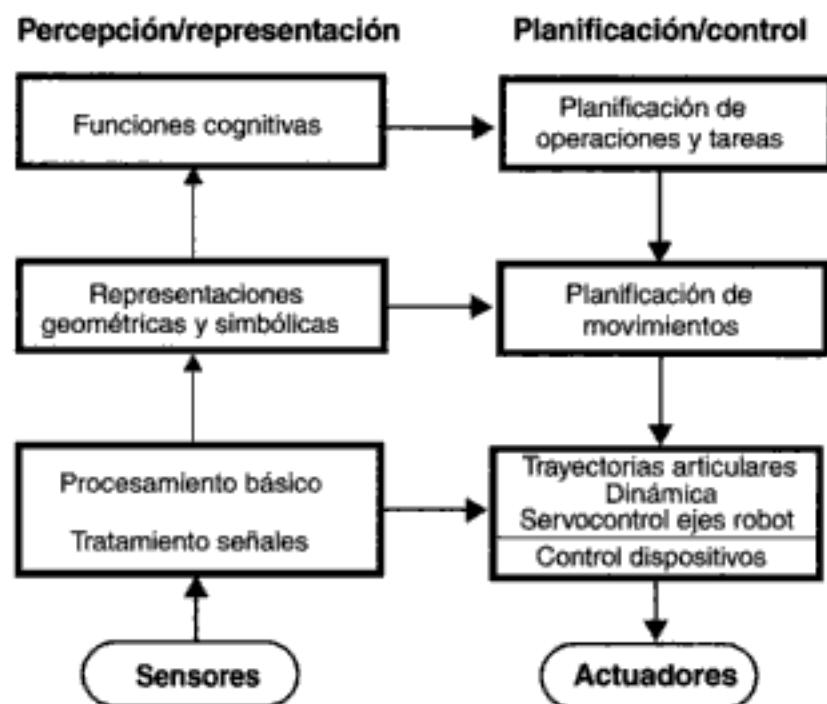
#### *1) Definición de niveles de jerarquización*

Para ello, se suele seguir un enfoque descendente con refinamientos jerárquicos y planes descendentes hasta llegar a los bucles de control de los actuadores de las articulaciones. Obviamente, el ancho de banda y características dinámicas asociadas a las funciones, así como su complejidad computacional, juegan un papel importante en la definición de esta estructura jerárquica. El ciclo de tiempo para realizar una determinada función puede variar entre minutos, para razonamiento simbólico, y milisegundos, en el control de articulaciones.

Por consiguiente, parece necesario especificar de la forma más precisa y cuantitativa posible los requerimientos necesarios y evaluar la complejidad asociada a cada tarea. Es claro que una definición que no contemple funciones que después pudieran ser relevantes dificultaría la implantación posterior de dichas funciones, mientras que una especificación con niveles carentes de suficiente contenido puede llevar a una complejidad artificial de la arquitectura que origine rendimientos inferiores.

En la figura 6.3 se ilustra una estructura jerárquica con tres zonas en la que se contemplan tanto funciones de percepción/representación como de planificación/control, considerándose también los acoplamientos horizontales.

La zona inferior de la jerarquía corresponde al procesamiento de señal de los sensores y a la generación de órdenes de control sobre los actuadores. Desde el punto de vista de planificación/control, las acciones generadas pueden responder a consignas del nivel superior, o incluso a reacciones directas a las entradas suministradas por los sensores.



**Figura 6.3: Niveles de la arquitectura de control.**

En esta zona pueden considerarse, a su vez, varios niveles jerárquicos, situándose en el inferior el servocontrol de ejes, con un periodo típicamente del orden de milésimas de segundo en los manipuladores. En un nivel superior se contemplaría la generación de trayectorias articulares, con un periodo de centésimas de segundos. Dependiendo de la estrategia de control, la consideración de modelos de la dinámica del manipulador se realizaría también en este nivel.

En la zona inferior es también posible considerar funciones que suministren una reacción rápida como respuesta a sensores externos (proximetría, fuerzas/pares, etc.). Nótese la posibilidad de establecer sistemas de control en bucle cerrado con estrategias mixtas de control posición/esfuerzos, que pueden resultar de interés en las operaciones de ensamblado e inserción, en las que se realizan movimientos manteniendo el contacto del efecto final con objetos del entorno.

Por último, obsérvese que en la zona inferior deben incluirse las funciones de control del efecto final, así como de otros dispositivos mecánicos.

La *zona intermedia* se ocupa de la planificación de movimientos. Desde el punto de vista de la percepción/representación, en estos niveles podrían generarse representaciones geométricas y simbólicas del robot y su entorno. Nótese que las entradas para la planificación pueden proceder tanto de la representación obtenida en esta misma

zona como de los planes suministrados por la zona superior. Cabe distinguir entre la planificación de caminos libres de obstáculos y la especificación de movimientos tales como "posicionar-manipulador-en X", o "mover-herramienta-a Y".

En esta zona es necesario emplear programas para la planificación de caminos teniendo en cuenta una descripción del entorno en términos geométricos o simbólicos. Alternativamente, esta planificación podría realizarse por un operador remoto. Dependiendo de la complejidad del entorno, podría hablarse de periodos del orden de segundos o decenas de segundos para la planificación. En cualquier caso, también cabe contemplar planificaciones locales en tiempo real a partir de la interpretación de la información suministrada por los sensores, modificando caminos previamente planificados.

Adicionalmente, en esta zona intermedia, se realizaría la supervisión de los niveles inferiores de control de movimientos. En particular, dependiendo de los movimientos planificados, puede seleccionarse una determinada estrategia de control en la zona inferior. De esta forma, se ordenaría un control posicional puro con velocidad relativamente elevada en un movimiento de aproximación libre de contacto, conmutando a un control híbrido con realimentación de posición y esfuerzos y velocidad menor cuando se produce el contacto o la actuación de la herramienta, todo ello de acuerdo con la secuencia de operaciones elementales que se recibe de los niveles superiores. Asimismo, en zonas con proximidad de obstáculos, podría disminuirse la velocidad y emplear realimentación sensorial.

En la *zona superior* se involucran funciones cognitivas en lo que respecta a la percepción/representación, y funciones estratégicas en planificación. En el caso más general, cabe distinguir también diversos niveles. En el nivel inferior se partiría de las órdenes contempladas en un plan general planificando su ejecución. Nótese que la complejidad de las funciones involucradas puede ser muy variable. De este modo, en aplicaciones de manipulación, existen funciones tales como las de transferencia de objetos en las que bastaría prácticamente con la planificación de movimientos considerada en la zona intermedia, mientras otras, como las de ensamblado, pueden requerir un importante esfuerzo de planificación en este nivel.

Asimismo, en la zona superior, se realizaría la supervisión de los niveles de planificación de movimientos de la zona intermedia. De esta forma, si un programa de planificación de caminos no puede encontrar un camino libre de obstáculos, lo cual es relativamente frecuente, podrían emplearse diferentes alternativas tales como la modificación de los parámetros de la estrategia de búsqueda, el ensayo de otra planificación diferente de la tarea en curso, o la devolución del control al operador.

Podría existir también un nivel superior que generaría una secuencia de órdenes para cumplir con un determinado objetivo del plan general. Típicamente se le asignan periodos de decenas de segundos, o incluso centenas en misiones complejas que requieren descomponer el servicio de la misión en órdenes de administración de servicios que, a su vez, se descomponen en grupos de tareas que se realizan sobre objetos individuales.

## 2) Compromiso planes/reacciones

Este compromiso es hoy día típico en el diseño de arquitecturas de sistemas de control inteligente de robots, ocupando a numerosos investigadores. Se trata de esta-

blecer un compromiso entre la ejecución de planes mediante estrategias de búsqueda, de la forma más aproximada posible a la óptima, para alcanzar unos objetivos previamente definidos, y la capacidad de reacción a contingencias no previstas a partir de información sensorial. Este compromiso ya se ha discutido en los apartados 6.3 y 6.4. En principio, para entornos básicamente conocidos y estáticos, es posible disponer de modelos suficientemente precisos, por lo que el sistema de control podría tener una arquitectura orientada fundamentalmente a la ejecución de planes, manteniendo una cierta capacidad de reacción ante imprevistos.

### **3) Conocimiento del estado**

Se trata de definir la información que debe mantener el sistema de control sobre el estado del robot y su entorno.

De especial interés es la representación del entorno mediante mapas y/o conocimiento simbólico, así como la definición de estructuras de datos que representen de forma adecuada dicho estado para su actualización y empleo para la generación de acciones en tiempo real. Nótese como en la representación del entorno a partir de un sistema de visión pueden involucrarse datos tan diversos como imágenes, objetos geométricos y símbolos.

Aunque el entorno sea básicamente estático, la definición del estado y estructuras para representarlo afectan de forma importante a la eficiencia de la planificación, y a la fiabilidad del sistema, incluyendo la detección de obstáculos en tiempo real.

### **4) Otras características**

En la definición funcional de la arquitectura es necesario involucrar también la definición de prioridades (mayores a las funciones relacionadas con la seguridad), la asignación de tareas, típicamente continua en algunas funciones, tales como la evitación de obstáculos, y bajo demanda en otras, tales como el reconocimiento de un determinado objeto. Asimismo, puede distinguirse entre funciones sincronas y asíncronas.

#### **6.5.2 Gestión de ejecución e implantación**

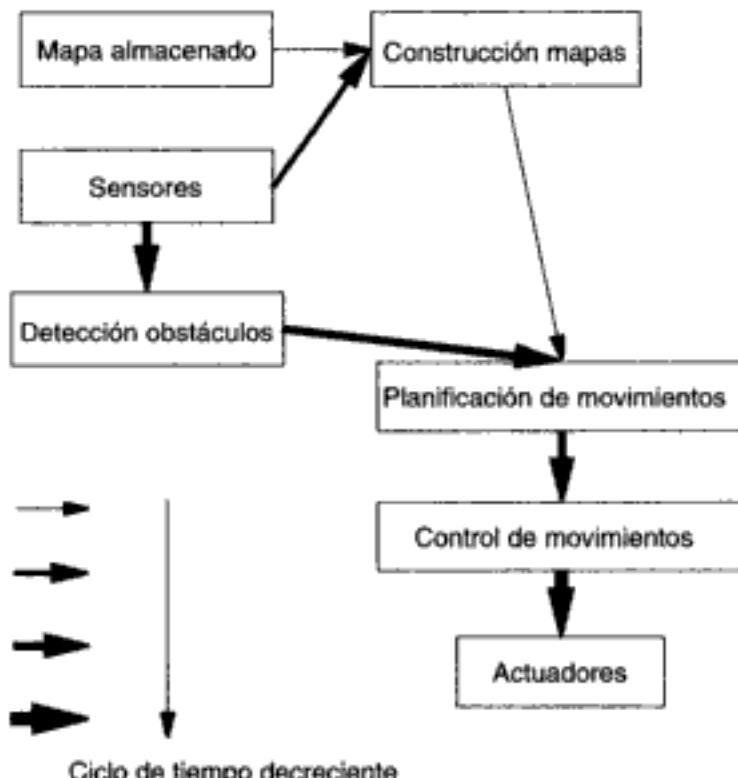
Se trata esencialmente de definir las relaciones entre las funciones del sistema y los recursos "software" y "hardware" disponibles.

El tema puede abordarse desde la perspectiva típica de las arquitecturas para control en tiempo real. De esta forma, el primer compromiso que hay que adoptar es el grado de distribución de los recursos (distribuido frente a centralizado).

En un sistema centralizado existen una serie de módulos dedicados a percepción, planificación de tareas/misiones, planificación de trayectorias y control de movimientos, los cuales se comunican todos con un módulo central en el que se mantiene (o se tiene acceso) a toda la información relevante del robot y su entorno (geometría, conocimiento simbólico, posición y trayectoria del robot, etc.). Este módulo central se encarga de comunicar y sincronizar los restantes módulos. En la figura 6.4 se ilustra esta implantación.

**Figura 6.4:** Arquitectura centralizada.

Alternativamente, es posible concebir una implantación en la cual no existe un módulo central de las características mencionadas en el punto anterior. Esta situación se muestra en la figura 6.5 de forma simplificada. El énfasis en ella está en el periodo con que se comunican los distintos módulos y en la representación apropiada de los datos. En la figura se muestran conexiones de distinta prioridad. Las de mayor prioridad se ejecutan con mayor frecuencia (menor periodo de espera para ser atendidas).

**Figura 6.5:** Arquitectura descentralizada.

Obsérvese como los sensores se emplean tanto para la detección de obstáculos como para la construcción de mapas del entorno, pero la primera de estas funciones es de mayor prioridad. La planificación de movimientos se realiza de acuerdo con el mapa del entorno pero debe responder también a la presencia de obstáculos imprevistos.

En general, la implantación de la figura 6.4 favorece la planificación de tareas y movimientos mediante estrategias de búsqueda manteniendo un modelo suficientemente preciso, mientras que el esquema descentralizado de la figura 6.5 es apropiado para arquitecturas reactivas.

Por tanto, de acuerdo con las consideraciones de los apartados precedentes, cuando el entorno es conocido e invariante, el sistema de control suele implantarse, en lo que se refiere a las funciones inteligentes, según un esquema básicamente centralizado.

Por otra parte, en el diseño de la arquitectura de control, es necesario tomar decisiones tales como las relativas al típico compromiso procesamiento/memoria, que es particularmente relevante cuando es necesario implantar funciones de percepción en tiempo real y mantener descripciones geométricas y simbólicas. Estas decisiones pueden tener una notable influencia en la eficiencia del sistema resultante.

Asimismo, es conveniente estudiar otros aspectos tales como la utilización de una memoria compartida entre procesos o la comunicación entre procesos mediante paso de mensajes.

En el apartado siguiente se trata la elección del hardware y software necesario para implantar la arquitectura de control.

## 6.6 SOLUCIONES HARDWARE Y SOFTWARE

---

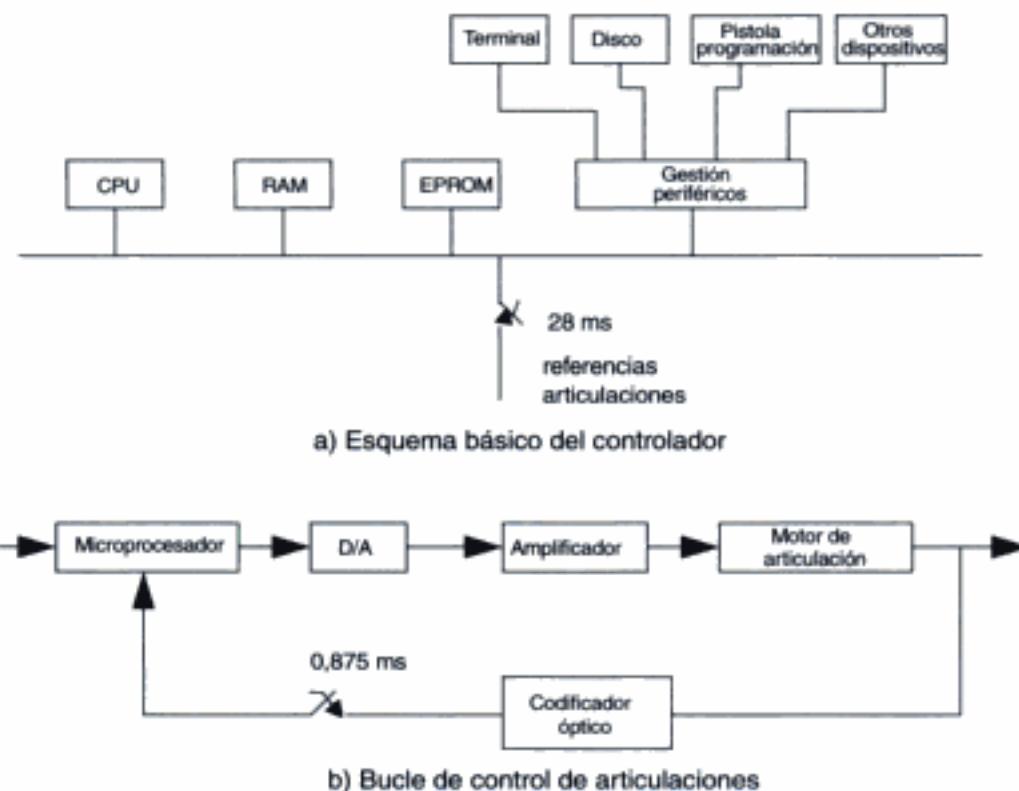
Para la elección del "hardware" y del "software" básico es necesario tener en cuenta características similares a otros sistemas en tiempo real tales como la eficiencia y velocidad de ejecución, facilidad de desarrollo de programas, fiabilidad, posibilidades de expansión y reconfiguración, existencia de recursos específicos (mejor "hardware" para determinadas tareas) y coste.

Existen aplicaciones, tales como las de robótica espacial, en las que las especificaciones de fiabilidad, tolerancia a fallos y protecciones a las condiciones ambientales son particularmente relevantes. En estas aplicaciones la elección está fuertemente restringida.

Las soluciones adoptadas en control de robots para aplicaciones generales han variado significativamente. Así, en los años setenta y ochenta, los controladores de los robots manipuladores industriales empleaban un "hardware" basado en microprocesadores convencionales, de prestaciones modestas, implantando funciones de servocontrol de articulaciones, generación de trayectorias a partir de órdenes relativamente simples, comunicación con el usuario y gestión de periféricos. Las funciones de percepción y planificación era necesario implantarlas en un "hardware" separado que, típicamente, se conectaba al controlador básico del robot mediante una línea serie. La solución más habitual era un computador que realizaba funciones de supervisión, ejecución de programas de planificación y coordinación con otros equipos y, eventualmente, de percepción dotándole de tarjetas especializadas, tales como las de adquisi-

ción de imágenes. Asimismo, se incluían equipos con "hardware" especializado, tales como los de procesamiento de imágenes.

Un ejemplo típico de sistema de control de robots industriales de los años ochenta es el de los robots Puma de Unimation, que se mencionaron en el capítulo 2 de este texto. En la figura 6.6 se muestra el esquema básico de control del Puma. En la unidad central existe un bus al que se conecta la CPU, accediéndose a memoria y a un dispositivo que gestiona los periféricos entre los cuales cabe distinguir el terminal del operador, una unidad de almacenamiento externo y una pistola de programación. Asimismo, existe un dispositivo que, a través de una interfase, permite suministrar las referencias a los bucles de control de las articulaciones. Estas referencias se transmiten cada 28 ms. Se dispone de un bucle de control digital de cada una de las articulaciones. El microprocesador genera la señal de control que se aplica a través de un convertidor digital-analógico a un amplificador que acciona el motor de la articulación. Se emplean codificadores ópticos para medida del giro de la articulación. El bucle de control de la articulación se cierra cada 0,875 ms.



**Figura 6.6:** Sistema de control del robot Puma.

Existen otros sistemas de control de robots más simples en los que un único microprocesador cierra los bucles de control de todas las articulaciones.

En cualquier caso, conviene poner de manifiesto que en los sistemas de control de manipuladores robóticos, mencionados en párrafos precedentes, se emplea el modelo cinemático del manipulador para generar las referencias de las articulaciones a partir del movimiento especificado desde el terminal del operador (instrucciones en lenguaje

je de programación del robot o bien órdenes de movimiento en botonera de programación), pero el control en bucle cerrado de cada una de las articulaciones se realiza en muchos casos de forma desacoplada, sin tener en cuenta el modelo dinámico del manipulador.

En el comienzo de los años noventa se desarrollaron un buen número de aplicaciones empleando chasis con bus VME con tarjetas de CPU (tales como los Motorola 68030 o 68040) y memoria, tarjetas con distintos módulos de control de movimientos y tarjetas específicas para sensores. Este "hardware" ha sido bastante empleado para los niveles inferiores de la arquitectura. Para los niveles superiores, en los cuales los requerimientos computacionales para percepción y planificación son superiores, se han empleado chasis que pueden contener la tarjeta base de una estación de trabajo suficientemente potente y tarjetas especializadas tales como las de procesamiento de imágenes. Una solución bastante empleada consiste en comunicar mediante red ETHERNET el "hardware" de los niveles inferiores a esta estación de trabajo.

Existen productos "software", tales como, por ejemplo, el VxWorks, o el LynxOS, que suministran para el "hardware" mencionado en el párrafo precedente un sistema operativo en tiempo real, una capa de servicios de comunicación interprocesos y herramientas de desarrollo. Sobre este sistema puede desarrollarse una capa de comunicación entre tareas que facilite la implantación de los niveles superiores de la arquitectura.

Asimismo, la configuración mencionada en los párrafos precedentes facilita el desarrollo y ejecución de los programas desde estaciones de trabajo UNIX conectadas mediante red ETHERNET, con lo cual el usuario puede elaborar sus programas (típicamente en lenguaje C), depurarlos y cargarlos en el "hardware" de control desde su puesto de trabajo habitual, e incluso ejecutarlos en dicho "hardware".

La arquitectura del sistema de control del robot móvil RAM-1 (Ollero y otros, 1994), que se muestra en la figura 6.7, tiene la estructura que se indicaba en los párrafos precedentes.

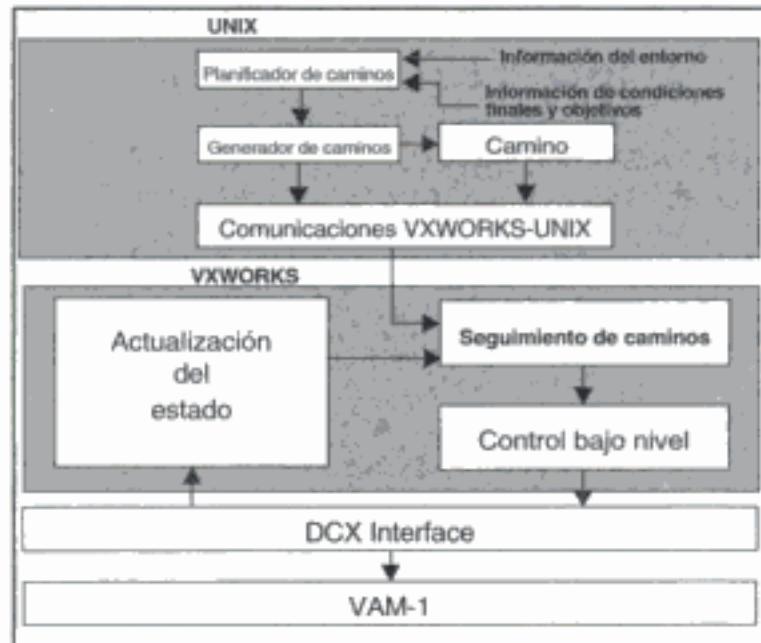
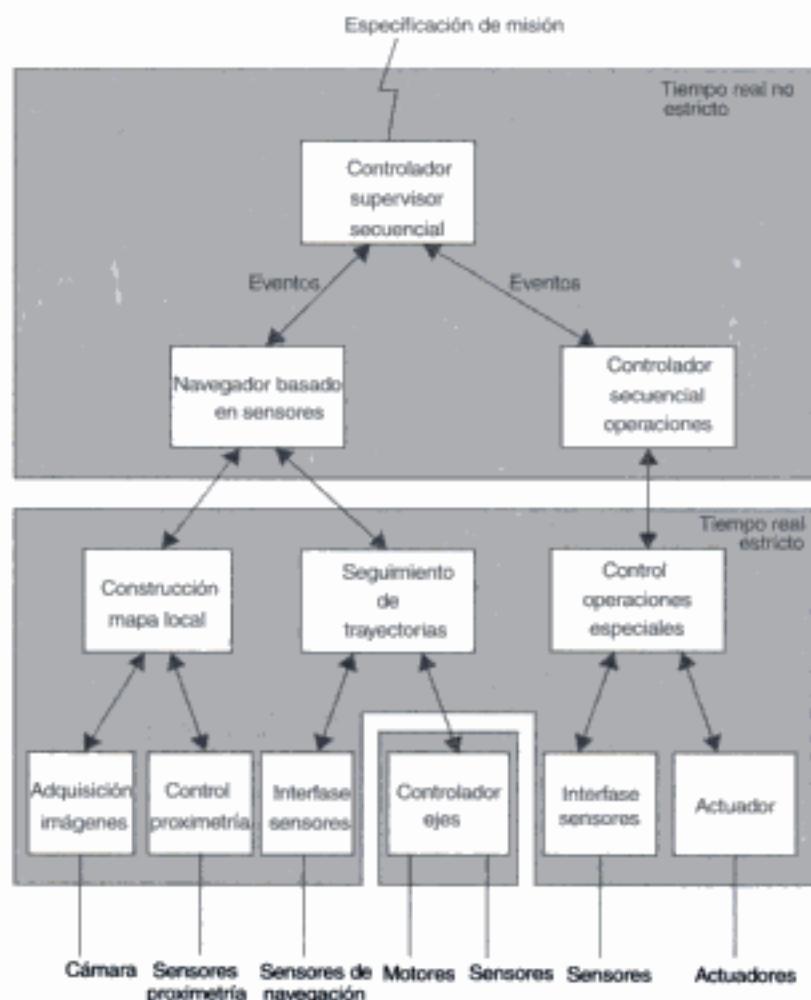


Figura 6.7: Arquitectura de control del RAM-1.

En el nivel superior se realiza la planificación de caminos a partir de un modelo del entorno. El resultado es una secuencia de puntos de paso que se interpolan o aproximan empleando un generador de caminos. Tanto el planificador como el generador de caminos tienen en cuenta las restricciones cinemáticas y dinámicas del vehículo. La planificación de caminos puede descomponerse en planificación global, teniendo en cuenta un modelo estático del entorno, y planificación local, basándose en medidas adquiridas con el sistema de percepción durante la navegación del robot. Todas estas funciones de percepción y planificación se ejecutan mediante procesos en UNIX que se han implantado en estaciones de trabajo con bus VME.

En el nivel inferior se ejecuta el seguimiento de los caminos planificados en el nivel superior, generándose órdenes a la dirección del vehículo. Tanto estas órdenes como las de velocidad se envían al "hardware" de control de movimientos, que emplea microcontroladores especializados. Todos estos procesos se ejecutan bajo el sistema operativo en tiempo real VXWorks, en un sistema basado en el procesador Motorola 68030 conectado a un bus VME, al cual se conectan también tarjetas especializadas tales como la DCX para el control del movimiento de los ejes.

En la figura 6.8 se muestra una versión evolucionada de la arquitectura de control del robot RAM-1 (Ollero y otros, 1994). Obsérvese que existe también una estructura



**Figura 6.8:** Arquitectura jerárquica del robot RAM-1.

jerárquica cuyo nivel superior consiste en un controlador secuencial, que coordina un segundo nivel en el cual existen módulos de navegación y de realización de operaciones especiales. Estos dos niveles, que constituyen la capa superior de la arquitectura de control, han sido implantados en UNIX.

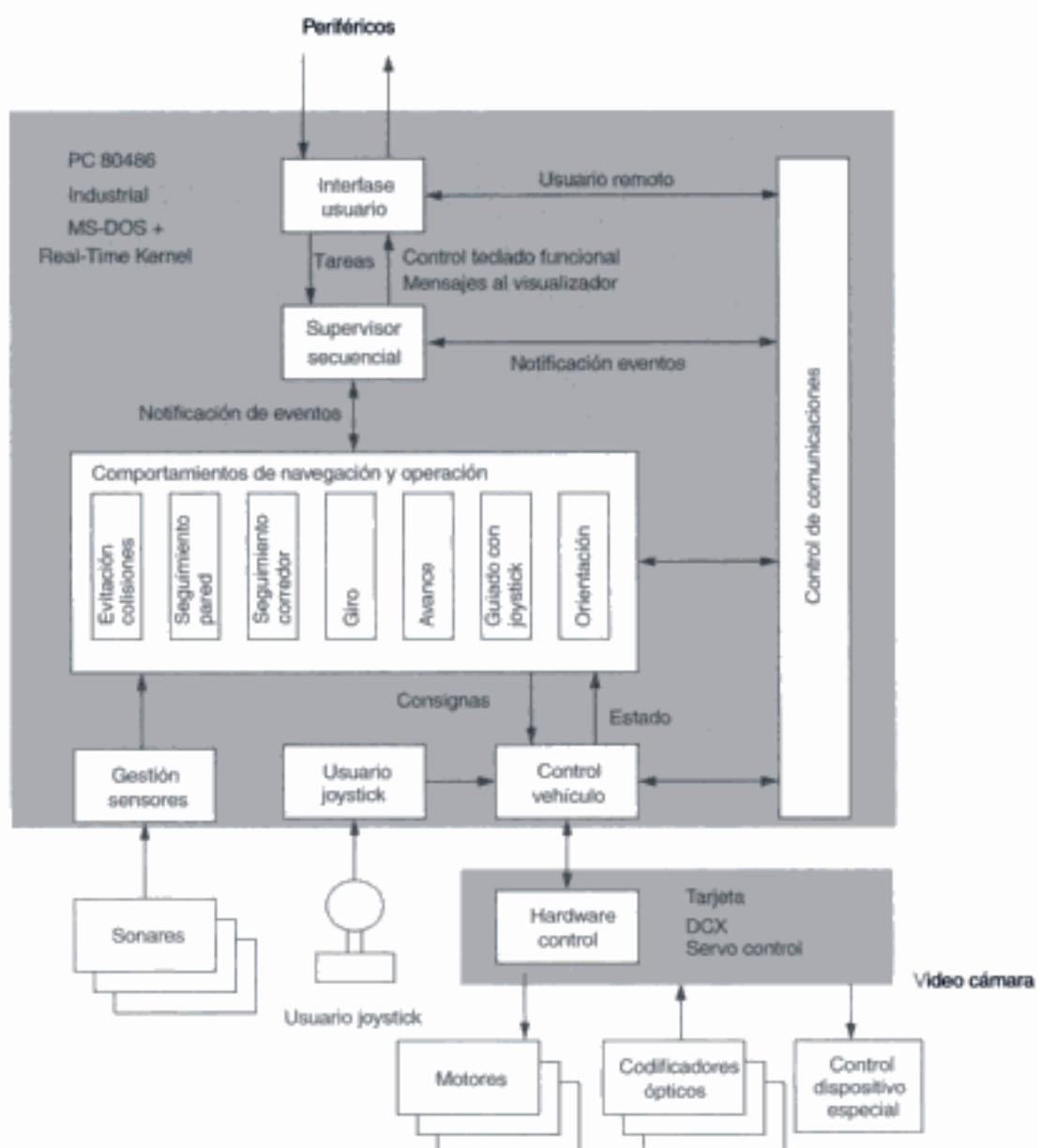
Las capas inferiores están constituidas por procesos que se ejecutan en sistemas operativos en tiempo real. Obsérvese que existe un nivel en el que se incluyen el seguimiento de trayectorias y la construcción de mapas locales. Igualmente, en el mismo nivel, se realiza el secuenciamiento de las operaciones con dispositivos especiales durante la navegación. Finalmente, en el nivel inferior, tiene lugar la interfase con los sensores y controladores de movimiento de los dispositivos especiales. Así, el generador de mapas locales utiliza un módulo de control de sensores de proximetría y, en particular, de sonares, y un sistema de adquisición de imágenes. El módulo de seguimiento de trayectorias interacciona con una interfase a los sensores de navegación y con el controlador de los motores del vehículo. Por último, nótese cómo el controlador de operaciones especiales interacciona también con sensores y actuadores especializados.

En los últimos años se han utilizado también arquitecturas PC con procesadores 486 y Pentium. El sistema de control del robot móvil Romeo (Ollero y otros, 1999) también dispone de un controlador de bajo nivel encargado del control de movimientos y comportamientos básicamente reactivos, y un controlador de alto nivel fundamentalmente dedicado a percepción y planificación.

El sistema de control de movimientos está constituido por un computador compatible PC de tipo industrial, con chasis de tamaño relativamente reducido y bus pasivo compatible AT. El procesador es 486 a 33 MHz y 8 MB de RAM. Existe una tarjeta de interfaz analógica y digital para lectura de sensores, y una tarjeta DCX en la que existen módulos para el control de motores. El PC realiza el control de los movimientos del robot bajo el sistema operativo en tiempo real LynxOS. Este sistema operativo es compatible con Unix, permitiendo además la programación en tiempo real y la obtención de tiempos de respuesta predecibles. Entre los programas de control de movimientos cabe mencionar los de seguimiento automático de caminos y los de ejecución de comportamientos reactivos basados en respuestas de sensores de proximetría y, en particular, de sonares tanto analógicos como digitales.

El control de alto nivel está organizado en torno a una CPU-3CE, tarjeta compatible SUN, que dispone de recursos suficientes como para considerarla una estación de trabajo. El sistema operativo es Solaris, idéntico a las máquinas de sobremesa, lo cual facilitó el desarrollo. Sin embargo, la tarjeta tiene un formato estándar (doble Eurocard) que permite su instalación en un chasis industrial y también de interfaz VME. En este bus VME existen tarjetas específicas para realizar funciones de digitalización y procesamiento de imágenes. Existe también una tarjeta que permite ampliar el número de líneas serie en ocho adicionales, lo que se emplea para controlar el movimiento de las cámaras, así como para entradas de sensores. La conexión con el controlador de movimientos se realiza mediante Ethernet.

Existe también la posibilidad de soluciones de bajo coste en las que un único computador se emplea para implantar las funciones de todos los niveles de la figura, excepto algunas funciones en las que se emplean tarjetas y procesadores especializados. Este es el caso de la arquitectura de control del robot AURORA, que se muestra en la



**Figura 6.9: Arquitectura de control de AURORA.**

figura 6.9 (Mandow y otros, 1996). Esta arquitectura está diseñada para que sea posible tanto el funcionamiento autónomo como la teleoperación.

En AURORA existe una interfase que permite que el usuario introduzca programas e instrucciones, tanto localmente como de forma remota, a través de un sistema de comunicaciones. El controlador secuencial del nivel superior interpreta estas órdenes y secuencia comportamientos que se ejecutan en un nivel inferior de forma autónoma o bajo la intervención del operador remoto. Estos comportamientos corresponden a los de una arquitectura de tipo “subsumption”. Entre ellos cabe mencionar los de seguimiento de una pared, seguimiento de un corredor, giro, avance en línea recta, cambio de orientación o evitación de una colisión.

Los comportamientos se ejecutan de una forma autónoma, admitiendo como entra-

das las procedentes de sensores de percepción del entorno, tales como los sensores de proximetría, del "hardware" de control del vehículo y, eventualmente, de un operador local.

El resultado de la ejecución de los comportamientos es un conjunto de instrucciones al "hardware" de control del vehículo. Este "hardware" suministra servicios para controlar el sistema mecánico del vehículo y, en particular, el eje de dirección y la velocidad. Normalmente, estas funciones se realizan en microcontroladores especializados dispuestos en una tarjeta que se conecta al bus.

Para implantar las funciones en tiempo real se ha empleado un "Kernel" de libre distribución.

En la segunda mitad de los años noventa se han empleado también arquitecturas basadas en microcontroladores y procesadores digitales de señal (DSP) especializados en distintas funciones. Se utilizan también arquitecturas con los DSP en paralelo. Estas arquitecturas son especialmente apropiadas para tareas que requieren una gran cantidad de procesamiento y que pueden ser paralelizadas total o parcialmente, tales como las de procesamiento de imágenes (Ferruz y Ollero, 2000).

## 6.7 CONCLUSIONES

---

La elección de una arquitectura "hardware" y "software" apropiada tiene una importante influencia en la eficiencia del sistema de control de robots.

En la primera parte de este capítulo se han considerado las diferentes funciones de control y las especificaciones funcionales de la aplicación.

Estas especificaciones permiten formular los requerimientos generales de la arquitectura.

Se han presentado diferentes tipos de arquitecturas según su reactividad, desde las puramente reactivas, basadas en reacciones simples a estímulos de sensores, hasta las orientadas a la ejecución de planes elaborados a partir de modelos del entorno.

A continuación se han efectuado consideraciones generales sobre el diseño de una arquitectura de control y su implantación.

En la última parte del capítulo se han presentado ejemplos de soluciones "hardware" y "software" existentes en los robots actuales.

## 6.8 REFERENCIAS

---

- Arkin R.C., 1992a. "Behavior-based Robot Navigation in Extended Domains", *J. of Adaptive Behavior*. Vol. 1, núm. 2.
- Arkin R.C., 1998. "Behavior-based Robotics", MIT Press.
- Brooks R.A., 1986. "A Robust Layered Control System for a Mobile Robot", *IEEE Journal on Robotics and Automation*. Vol. RA-2, núm. 1, pp. 14-23, marzo.
- Connell J. y P. Viola, 1990. "Cooperative Control of a Semi-Autonomous Mobile Robot", *IEEE Intl. Conf. on Robotics and Automation*, mayo 13-18, Cincinnati, Ohio, USA.
- Ferruz J. y A. Ollero, 2000. "Integrated real-time vision system for vehicle control in non-structured environments", *Engineering Applications of Artificial Intelligent*. Vol. 13, núm. 3, pp. 215-236.

- Groover M.P., M. Weiss, R.N. Nagel y N.G. Odrey, 1989. "Robótica industrial", McGraw-Hill.
- Kortenkamp D., R.P. Bonasso y R. Murphy, 1998. "Artificial intelligence and mobile robots", MIT Press.
- MacKenzie D.C. y R.C. Arkin , 1993. "Formal Specification for Behavior-Based Mobile Robots", SPIE. Vol. 2058 Mobile Robots VIII.
- Mandow A., J. Gómez de G., J.L. Martínez, V.F. Muñoz, A. Ollero y A. García-Cerezo, 1996. "The autonomous mobile robot AURORA for greenhouse operation", IEEE Robotics and Automation Magazine. Vol. 3, núm. 4, pp. 18-28, diciembre.
- Mataric M.J., 1992a. "Integration of Representation into Goal-Driven Behavior-Based Robots", IEEE Transactions on Robotics and Automation. Vol. 8, núm. 3, pp. 302-312.
- Mataric M.J., 1992b. "Behavior-Based Control: Main Properties and Implications", Workshop on Architectures for Intelligent Control Systems, IEEE Intl. Conf. on Robotics and Automation, Niza, Francia.
- Mataric M.J., 1997. "Behavior-Based Control: Examples from Navigation, Learning, and Group Behavior", Journal of Experimental and Theoretical Artificial Intelligence. Vol. 9, núms. 2-3, Hexmoor, Horswill, and Kortenkamp, ed.
- Ollero A., A. Mandow, V.F. Muñoz, J. Gómez de Gabriel, 1994. "Control architecture for mobile robot operation and navigation", Robotics and Computer-Integrated Manufacturing. Vol. 11, núm. 4, pp. 259-269, Pergamon.
- Ollero A., B.C. Arrue, J. Ferruz, G. Heredia, F. Cuesta, F. L-Pichaco, C. Nogales, 1999. "Control and perception components for autonomous vehicles guidance. Application to the Romeo vehicles", Control Engineering Practice. Vol. 7, núm. 10.
- Stein M.R. y R.P. Paul , 1994. "Operator Interaction, for Time-Delayed Teleoperation with a Behavior-Based Controller", IEEE Proc. of the Intl. Conf. on Robotics and Automation, mayo.

Este capítulo se dedica al estudio de sensores en robótica con mayor énfasis a sensores internos necesarios para cerrar los bucles de control de las articulaciones. En primer lugar, se tratan aspectos y características generales de los sensores. A continuación, se consideran los sensores básicos que permiten medir los ángulos y desplazamientos de las articulaciones y ejes de movimiento. En el apartado siguiente se tratan los sensores de velocidad angular. Los apartados siguientes involucran sensores que suministran una cierta percepción del entorno inmediato del robot; son los sensores de contacto y los sensores de fuerza y par. En la última parte del capítulo se estudian sensores de navegación que son específicos de los robots móviles.

## 7.1 INTRODUCCIÓN: SENsoRES Y MAGNITUDES

Los sensores permiten la adquisición de la información necesaria para el control del robot. En el estudio de los sensores debe involucrarse la medida de las magnitudes y su representación en forma compatible para su procesamiento.

En la toma de medidas siempre existe un cierto grado de incertidumbre. En principio, el incremento de la información hace posible la reducción de la incertidumbre. Para ello se trata de tomar más medidas o de emplear sensores redundantes.

Existen diferentes portadores de información basados en distintos principios físicos y químicos. Así, entre los principios y parámetros involucrados cabe mencionar:

- Mecánica: posición, velocidad, tamaño, fuerza,...
- Termotecnia: temperatura, calor, entropía,...
- Electricidad: voltaje, intensidad, resistencia, capacidad,...
- Magnetismo: intensidad de campo, densidad de flujo, permeabilidad,...
- Química: concentración de un material, estructura cristalina,...

- Radiación (ondas electromagnéticas) de todas las frecuencias, desde ondas de radio a rayos γ: intensidad, frecuencia, polarización, fase,...

Con respecto al procesamiento y transmisión de la información, pueden emplearse también distintas tecnologías con limitaciones físicas diferentes:

- Hidráulica mediante el empleo de componentes fluídicos. En este caso, existe el límite de la velocidad del sonido en un fluido, que es de aproximadamente  $10^3$  m/s.
- Eléctrica y electrónica. En la actualidad se emplean circuitos electrónicos. El límite de velocidad viene dado por la movilidad de las cargas en un material semiconductor, que es de aproximadamente  $10^5$  m/s.
- Radiante empleando componentes ópticos. El límite es la velocidad de la luz en la guía: aproximadamente  $10^8$  m/s.

En la actualidad, se emplea casi con exclusividad el procesamiento electrónico. Para su empleo es necesario traducir las magnitudes a señales eléctricas. Nótese que los sensores realizan frecuentemente transformaciones de energía. Así, por ejemplo, en un codificador óptico, la entrada es la rotación mecánica del eje y la salida una señal eléctrica. En esta transformación se emplea un haz de luz como fuente auxiliar de energía. Esta energía auxiliar, modulada por la rotación mecánica, produce la señal de la salida.

En numerosas aplicaciones, además de las propias magnitudes, interesa conocer sus derivadas en el espacio o en el tiempo. Para ello puede procederse a la medida de la magnitud y al cálculo de la derivada mediante procesamiento. También es posible la medida directa de la derivada, tal como la velocidad de giro de un eje mediante un tacómetro.

## 7.2 CLASIFICACIONES Y CARACTERÍSTICAS DE SENsoRES

El empleo de fuentes de energía adicional a la de la señal de entrada permite realizar una primera clasificación de sensores. Así, existen sensores que realizan la conversión directa de la energía de la señal. Como ejemplo, cabe mencionar las células fotoeléctricas de selenio. Para el funcionamiento básico de estos sensores no se requiere fuente de potencia adicional. Producen salida cero para entrada cero.

Sin embargo, en otros casos, se requiere una modulación con un aporte de energía con fuente diferente de la señal entrada. Este principio es de interés para realizar medidas con señal débil. Como ejemplo, cabe mencionar un puente de galgas extensiométricas. Se necesita hacer pasar corriente por resistencias para producir la salida.

Otra clasificación posible de sensores es según el carácter absoluto o relativo de las medidas que produzcan. Así, cabe distinguir entre los sensores, tales como los potenciómetros, cuya salida es un valor absoluto que representa sin ambigüedad a la señal de entrada, y los sensores incrementales cuya salida sólo indica la magnitud del cambio en la señal de entrada.

Entre las características más significativas para evaluar los sensores (Berney, 1985) se encuentran:

- Linealidad. La característica entrada/salida es lineal. Normalmente se evalúa la separación máxima de la línea recta.
- Histéresis. La salida del sensor para una determinada entrada varía dependiendo de que la entrada esté aumentando o disminuyendo.
- Repetibilidad. Es la variabilidad de la salida ante la misma entrada.
- Resolución. Es el cambio más pequeño en la entrada que puede ser detectable a la salida.
- Sensibilidad. Un pequeño cambio en la entrada causa un pequeño cambio en la salida. Normalmente se cuantifica por la relación entre el cambio en la salida dividido por el cambio en la entrada.
- Ruido. Es el nivel de señal espuria en la salida que no corresponde a un cambio en la entrada.

En este punto, conviene tratar el concepto que en la literatura en inglés se conoce como “smart sensors”. Se trata de sensores que no sólo producen la magnitud que se pretende medir, sino que también incorporan funciones de procesamiento tales como:

- Comunicación bidireccional. La transmisión de la información medida se realiza bajo petición, controlándose funciones locales.
- Autocalibración: Se trata de compensar variaciones de características con el tiempo o con condiciones ambientales.
- Filtrados, reducción de información en general y compensación de no linealidades.
- Fusión multisensorial. Medidas diferentes con diversos sensores y obtención de una estimación en función de todas las medidas.

Para realizar estas funciones se emplean normalmente circuitos electrónicos especializados, microcontroladores y otros componentes electrónicos.

Interesa también mencionar aquí la tecnología de *sensores de silicio* (Muller y otros, 1991), que permite integrar sensores en circuitos electrónicos. Se emplean principios físicos tales como el efecto fotoeléctrico para medir la luz, el efecto Hall para medir un campo magnético, o piezoresistencias para medir fuerza o presión. Estos sensores integran normalmente funciones tales como entrada/salida paralela, conversión analógico-digital, UART (Universal Asynchronous Receiver Transmitter), microcontroladores, o incluso microcomputadores completos.

Se consiguen con ello beneficios tales como la reducción de tamaño, disminución drástica de conexiones necesarias, mejor relación señal/ruido (evitar transmisiones de pequeña señal por cable), compensaciones internas (no linealidad, derivadas, desplazamiento) mediante circuitos integrados y acondicionamiento de señal: conversión analógico-digital, procesamiento, etc.

### 7.3 MEDIDAS DE DESPLAZAMIENTOS LINEALES Y GIROS

---

#### 7.3.1 Potenciómetros para medida de desplazamientos

Actúan como divisores de potencial. Se mide el desplazamiento mediante la variación de una resistencia.

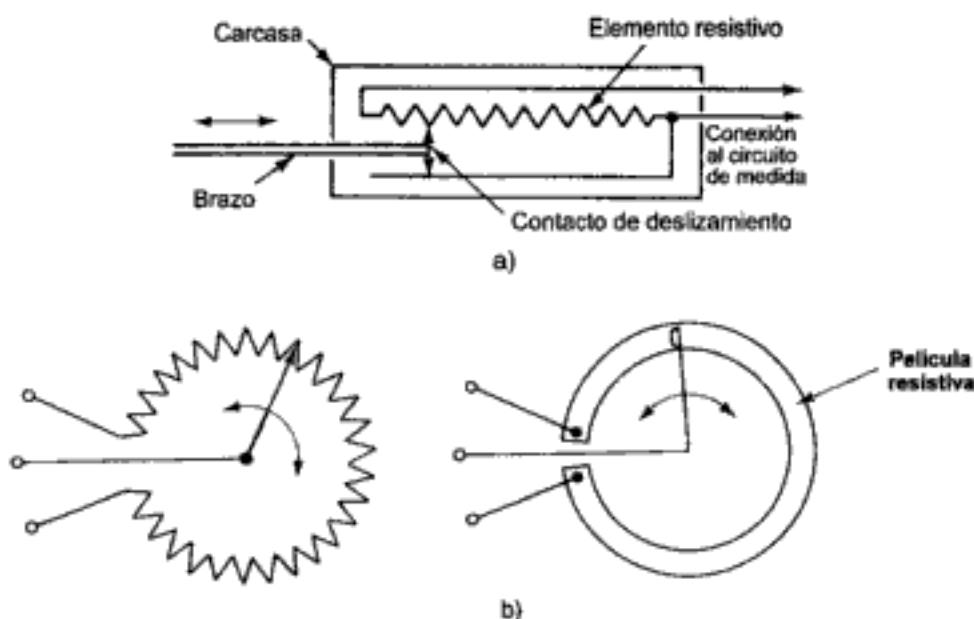


Figura 7.1: Potenciómetro: a) lineal; b) giratorio.

De esta forma puede emplearse una pista resistiva con un contacto móvil sobre la pista. Cabe distinguir entre potenciómetros lineales y rotatorios, según se pretenda medir un giro o un desplazamiento, tal como se muestra en la figura 7.1 (Hunter, 1987).

En la figura 7.2 se ilustra el principio del potenciómetro lineal. Si se aplica una tensión  $V_e$  en la pista resistiva, se obtiene  $V_s$  en el contacto como función de la posición. El desplazamiento se obtiene como

$$L_i = \frac{R_i L_T}{R_T} = \frac{V_s L_T}{V_e} \quad (7.1)$$

siendo  $R_T$  la resistencia total entre extremos de la pista resistiva,  $R_i$  la resistencia entre el contacto deslizante y el extremo y  $L_T$  la longitud total.

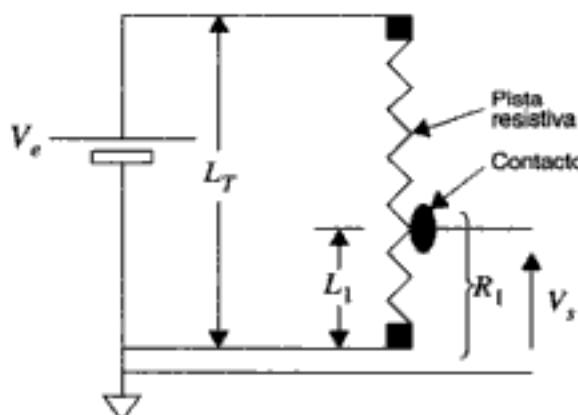
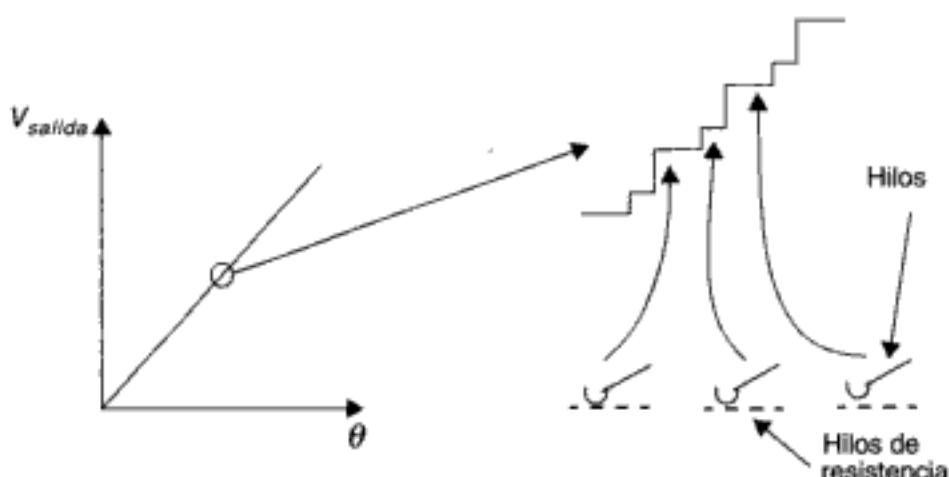


Figura 7.2: Potenciómetro lineal.



**Figura 7.3:** Potenciómetro con hilo enrollado.

Para potenciómetros rotatorios el ángulo se obtiene como:

$$\theta_I = \frac{R_I \theta_T}{R_T} = \frac{V_s \theta_T}{V_e} \quad (7.2)$$

Para realizar físicamente las pistas resitivas se emplean películas de material resistivo: carbón, metal, cerámica conductora, plástico conductor, etc.

Existen también potenciómetros en los que la pista resistiva consiste en hilos de alta resistencia sobre aluminio anodizado. Empleando un hilo enrollado, la salida cambia en escalones al saltar de una espira a la otra. La resolución viene dada por la magnitud del escalón, tal como se ilustra en la figura 7.3 (Russell, 1990).

Conviene poner de manifiesto que las ecuaciones anteriores son correctas si no hay derivación de corriente en los contactos. Para evitar estas derivaciones suelen emplearse amplificadores de alta impedancia de entrada a la salida de los potenciómetros.

Otro aspecto de carácter práctico es la dificultad para medir giros de más de  $300^\circ$  en una vuelta debido a la necesidad de espacio para los contactos.

Los potenciómetros son sensores de relativamente bajo coste. Sin embargo, la precisión es limitada. En potenciómetros de calidad pueden conseguirse errores lineales del 0,1%.

En general, presentan problemas de fiabilidad debido a desgaste, fricciones, polvo, etcétera.

Nótese también que la salida que suministran los potenciómetros convencionales es analógica, por lo que es necesario digitalizar la señal para aplicar control digital.

### 7.3.2 Codificadores ópticos

Estos sensores convierten un desplazamiento rotacional en una señal digital sin necesidad de convertidor analógico-digital. La medida del desplazamiento se realiza contando las interrupciones de un haz de luz.

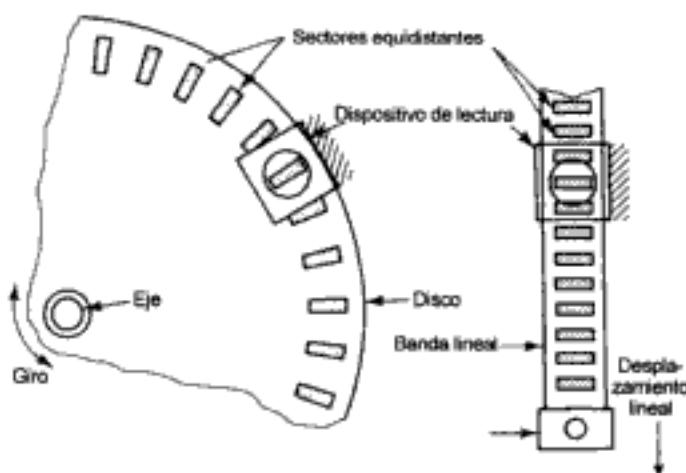


Figura 7.4: Codificadores ópticos rotacionales y lineales.

Es posible también distinguir entre codificadores ópticos rotacionales y lineales, tal como se muestra en la figura 7.4 (Hunter, 1987).

Como emisor puede emplearse una fuente incandescente o un diodo. El detector es típicamente un fotodiodo. Para medir el ángulo girado se cuenta el número de transiciones oscuro-luz y luz-oscuro. Es necesario un circuito que suministre un pulso positivo en cada transición. En la figura 7.5 se muestra un posible esquema.

Con el esquema anterior se conoce el ángulo pero no se tiene información sobre el sentido de rotación. Por tanto, no es válido en aplicaciones en las que el sentido de giro puede cambiar.

Otra posible técnica para medir el ángulo de giro es el empleo de un disco transparente de vidrio o plástico sobre el que se imprimen áreas de material opaco dispuestas en pistas concéntricas. En la figura 7.6 (Russell, 1990) se ilustra el principio de funcionamiento de estos codificadores, que pueden ser absolutos o incrementales.

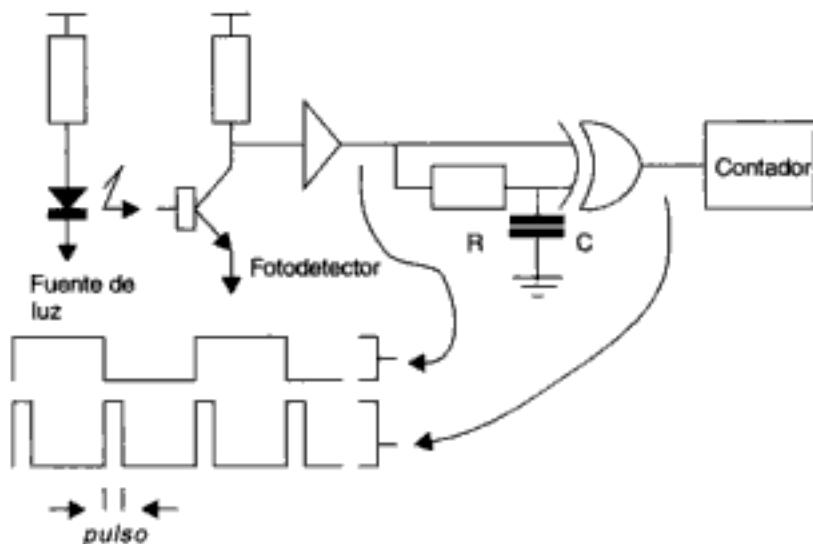


Figura 7.5: Circuito para codificador óptico de ranura.

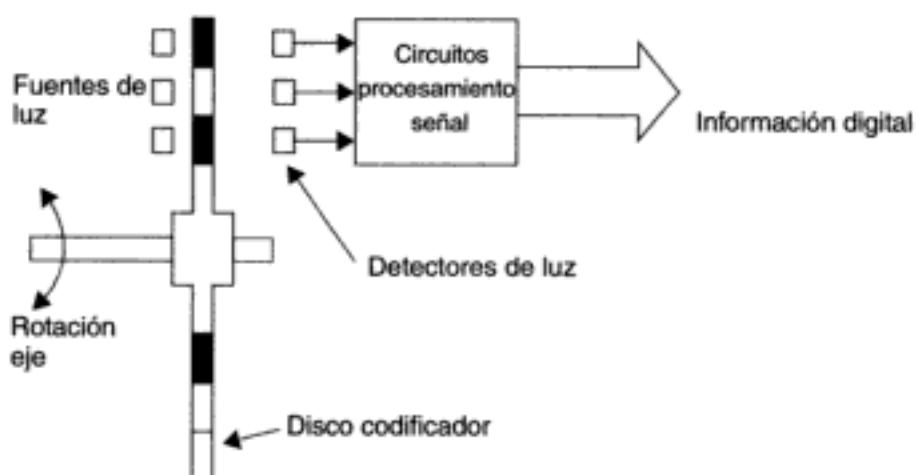


Figura 7.6: Principio del codificador óptico.

En los *codificadores ópticos absolutos* se produce un único código para cada posición angular del eje. En la figura 7.7 (Russell, 1990) se muestra un disco giratorio de 4 bits. Nótese como permite contar de 0 a 15; es decir 16 desplazamientos diferentes.



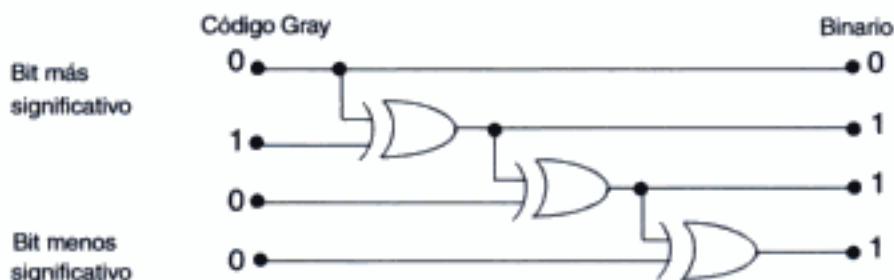
Figura 7.7: Codificador absoluto.

Tabla 7.1: Codificación Gray.

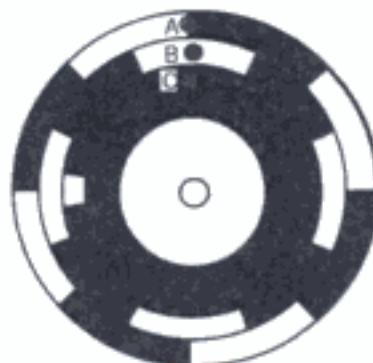
Número decimal	Codificación binaria	Código Gray	Número decimal	Codificación binaria	Código Gray
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

El principio de funcionamiento es la codificación binaria de áreas transparentes y opacas. Nótese la posibilidad de errores de lectura en el posicionamiento entre dos códigos. Así, al pasar de 0 a 15, cambiarían simultáneamente 4 bits. En la práctica, la alineación no es perfecta, con lo cual podrían producirse errores de lectura. Para evitarlo se emplea el código Gray, en el cual sólo cambia 1 bit entre posiciones digitales consecutivas. En la tabla 7.1 se muestra la codificación Gray de números entre 0 y 15. Obsérvese que se emplea de forma recurrente el operador O exclusivo.

En la figura 7.8 se muestra un circuito simple para convertir de código Gray a binario.



**Figura 7.8:** Circuito para convertir el código Gray a binario.



**Figura 7.9:** Codificador incremental.

La resolución del codificador depende del número de pares emisor-receptor. Un codificador con diez pistas tendría  $2^{10} = 1024$  códigos por revolución.

Los codificadores absolutos son relativamente costosos. Una técnica más simple desde el punto de vista constructivo es la de los *codificadores relativos*. En la figura 7.9 (Russell, 1990) se ilustra el funcionamiento de uno de estos codificadores.

En los codificadores relativos se requieren sólo dos o tres pares detector-emisor. Sin embargo, es necesario emplear más circuitería y se requiere una inicialización.

Los pares en A y B producen ondas cuadradas separadas un ángulo de fase de  $90^\circ$ . Si A adelanta a B en  $90^\circ$ , el disco rota en sentido antihorario. Por el contrario, si B se adelanta a A en  $90^\circ$ , el disco rota en sentido horario. En la figura 7.10 se muestran las secuencias en uno y otro caso.

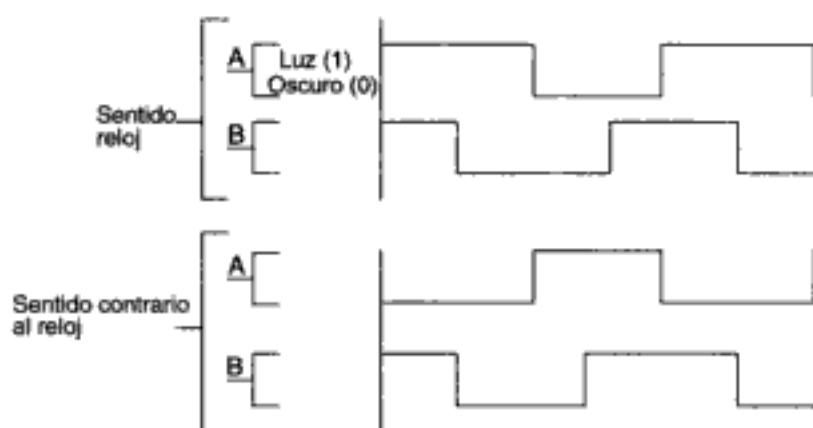


Figura 7.10: Señales en el codificador.

En la figura 7.11 (Russell, 1990) se muestra el diagrama de estados de un circuito secuencial que decodifica la salida de los dos commutadores A y B y produce dos salidas I (incremento) y D (decremento) que pueden controlar un contador que sigue los movimientos.

Obsérvese que el circuito necesita ser inicializado. El fotodetector C, en la figura 7.9, produciría un pulso en cada revolución.

En la figura 7.12 (Russell, 1990) se muestra un circuito lógico con biestables R-S para decodificar salidas de A y B produciendo salidas I y D que controlan un contador (incrementos y decrementos) para seguir el movimiento.

El circuito secuencial puede hacerse también por programa con ejecución suficientemente rápida.

Existen técnicas para incrementar la resolución de los codificadores incrementales sin necesidad de reducir áreas sensitivas o aumentar el diámetro del codificador. Se emplean para ello las denominadas franjas de moaré, utilizando otro disco fijo transparente con líneas paralelas con diferente espaciado. En este caso, un pequeño movimiento produce un gran desplazamiento de las franjas.

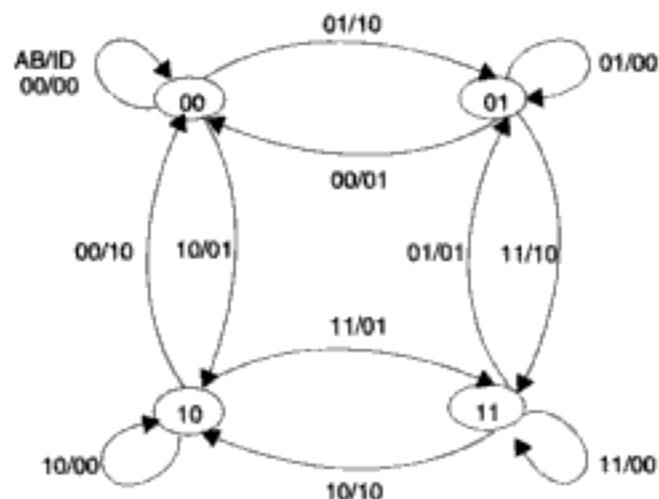


Figura 7.11: Circuito secuencial asíncrono del codificador.

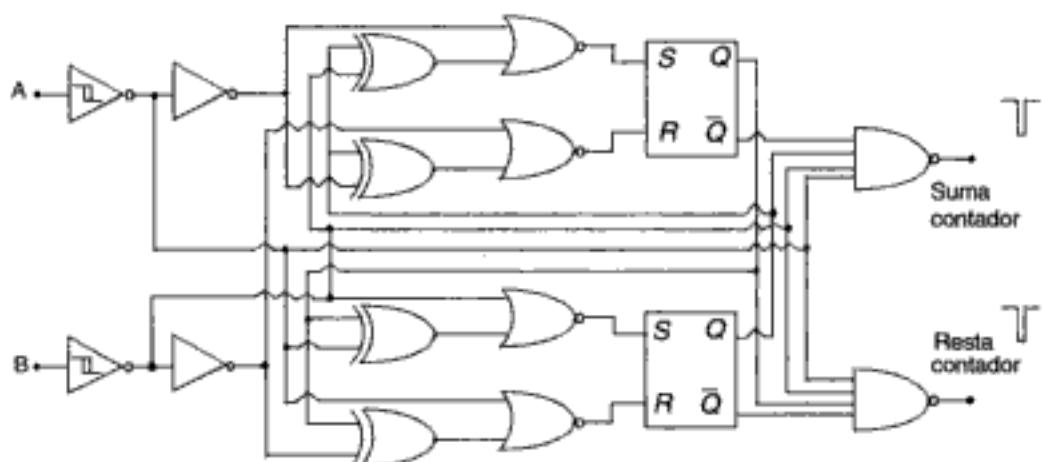


Figura 7.12: Implementación como circuito secuencial asincrónico.

### 7.3.2.1 Máquinas síncronas, transformadores y resolvedores

Se emplean máquinas eléctricas síncronas para medir y transmitir eléctricamente y con elevada precisión información angular. En estas máquinas existe un rotor excitado por corriente alterna (devanado primario) y dos o más estatores (devanados secundarios) orientados simétricamente alrededor del rotor. El acoplamiento magnético entre el devanado del rotor y los devanados estatores varía en función del giro del rotor. El acoplamiento máximo se produce cuando los campos son paralelos y el mínimo cuando el campo del rotor es ortogonal con respecto a un devanado del estator. Las salidas de los estatores son corrientes alternas cuyas magnitudes definen de forma única el ángulo que gira el rotor en un determinado instante.

Una configuración muy conocida es la del *transmisor/receptor de tres fases*, que se emplea para transmitir un giro a una localización remota, tal como se ilustra en la figura 7.13 (Klafter y otros, 1989). Existe un transmisor maestro conectado a un receptor esclavo, de forma que los estatores son paralelos. Se excitan ambos motores mediante una corriente alterna, normalmente entre 400 Hz y varios kHz. Cuando los motores están en una alineación idéntica a la de los respectivos estatores, las tensiones de salida de los estatores serán iguales y, por tanto, no circulará corriente. Si se aplica un



Figura 7.13: Transmisor/receptor síncrono.

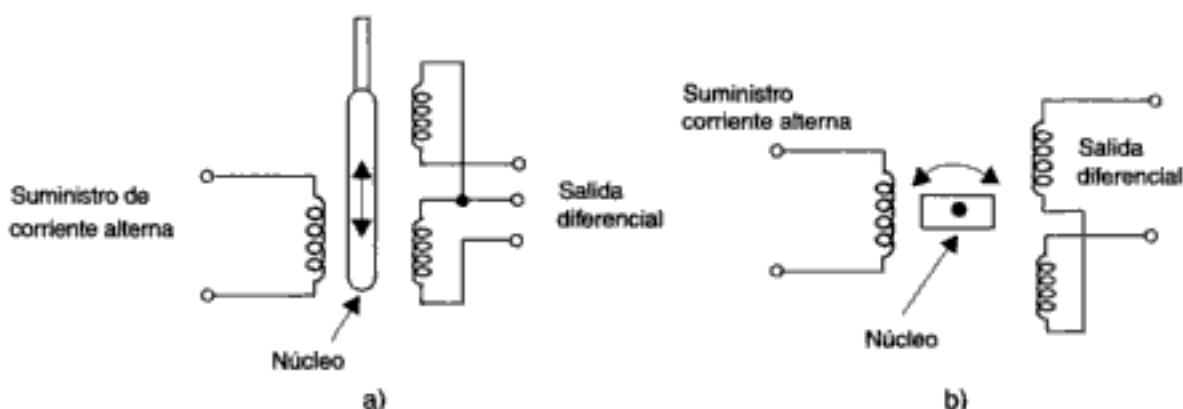


Figura 7.14: Esquema básico de un transformador diferencial variable.

par al eje del rotor del transmisor de forma que gire con respecto al del receptor, se generará una diferencia de tensión que producirá una corriente en los estatores. Estas corrientes hacen que se produzca un equilibrio de pares generándose en el receptor un par idéntico al que se aplica en el transmisor. Como resultado, el eje del receptor girará el mismo ángulo que el del transmisor.

Los *transformadores lineales variables* producen una tensión proporcional al desplazamiento de una armadura de hierro que se mueve en el campo creado por una bobina.

En la figura 7.14a se muestra una bobina con un devanado primario y dos devanados secundarios conectados en oposición. La salida es la diferencia de potenciales. La diferencia es nula cuando la inductancia mutua entre el devanado primario y cada uno de los secundarios es nula, lo que corresponde a que el núcleo esté en la posición central o completamente fuera.

Existen también *transformadores rotatorios* (figura 7.14b) que permiten medir rotaciones del núcleo de hasta 180°. En este caso, el cambio de fase indica la dirección.

Estos sensores necesitan circuitos para excitar el devanado primario y detectar la amplitud y fase de la salida. Nótese también que la salida es analógica, por lo que debe digitalizarse posteriormente para aplicar control digital.

El sensor permite obtener precisiones entre 0,5 y 0,05%. El rango de medida es entre  $\pm 0,1$  mm y  $\pm 250$  mm.

Los *resolvedores* se emplean para medición precisa del desplazamiento angular. Son máquinas síncronas en las que los devanados adoptan una disposición especial, tal como se ilustra en la figura 7.15 (Everett, 1995). El estator consiste en dos devanados primarios a 90° a los que se aplica corriente alterna con una diferencia de fase de 90°. En el rotor existe un único devanado en el que se toma la tensión de salida mediante escobillas de anillos deslizantes. La salida es de la misma amplitud que la entrada pero con una fase cambiada. Normalmente se excita con señal de alterna a 50, 400, o 1000 Hz.

En la figura se muestra el rotor girado un ángulo de aproximadamente  $\theta = 45^\circ$ . En general, cuando el rotor gira un ángulo  $\theta$ , se genera en él la suma de los componentes de las entradas seno y coseno:

$$V_r = V \cos \theta \cos \omega t + V \sin \theta \sin \omega t = V \cos (\theta - \omega t) \quad (7.3)$$

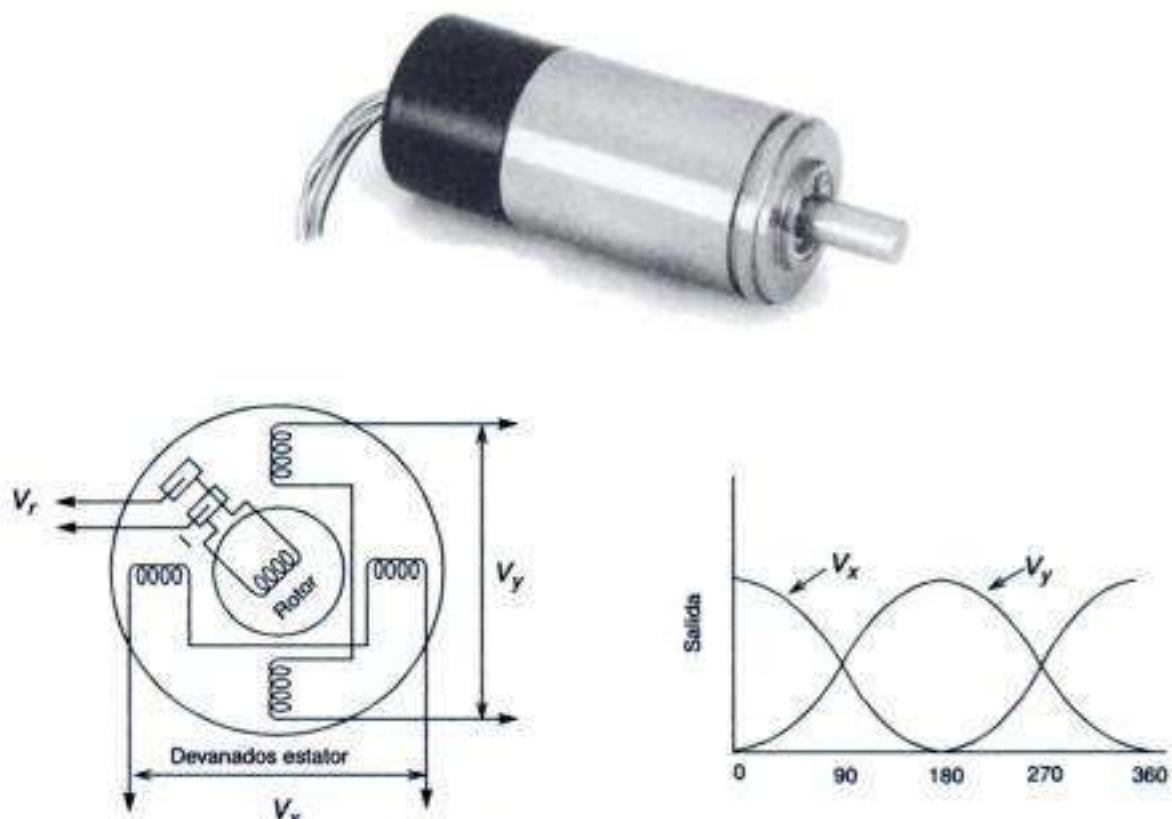


Figura 7.15: Resolvedor síncrono.

El ángulo entre el rotor y el campo resultante es  $\theta - \omega t$ . El potencial inducido es  $V \cos(\theta - \omega t)$ . Obsérvese cómo en este sensor también se necesita digitalizar la señal de salida para aplicar control digital.

Existen también resolviédores sin escobillas que emplean un transformador giratorio, como se ilustra en la figura 7.16 (Everett, 1995).

En estos sensores no existen escobillas ni anillos deslizantes. Se aplica la señal de alterna al rotor mediante un oscilador. La salida se obtiene tomando las componentes seno y coseno inducidas en estator. El ángulo del rotor puede determinarse a partir de la amplitud de la señal del coseno o del seno. No obstante, para aumentar la precisión de la medida se considera la relación entre ambas señales.

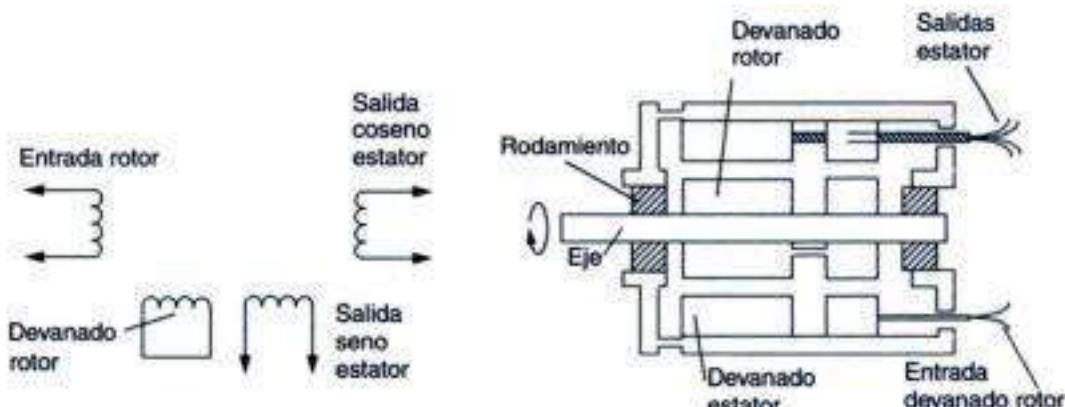


Figura 7.16: Resolvedor sin escobillas.

### 7.3.3 Medida de velocidad de ejes

En la figura 7.17 se muestran esquemas básicos de funcionamiento de sensores de velocidad.

En la figura 7.17a se muestra un resolvedor que genera una tensión proporcional a la velocidad que, una vez amplificada, proporciona realimentación directa de velocidad para el control de un eje.

En la figura 7.17b se representa un contador incremental cuya frecuencia de pulsos proporciona la velocidad. Así, una frecuencia de 2000 Hz desde el codificador de 1000 pulsos/rev indica una velocidad de 2 rev/s.

En la figura 7.17c se representa un tacogenerador que genera una salida eléctrica relacionada con la velocidad de entrada mecánica. Consiste en un generador eléctrico de corriente continua acoplado (directamente o mediante engranajes) al eje cuya velocidad pretende medir. La tensión es proporcional a la velocidad y su polaridad indica la dirección de giro. En tacogeneradores de alta calidad la linealidad es del 0,1%.

Por último, en la figura 7.17d se ilustra un tacogenerador de reluctancia variable. El empleo de corriente alterna evita caídas de tensión y ruidos en transmisiones que tendrían un efecto mayor con corriente continua.

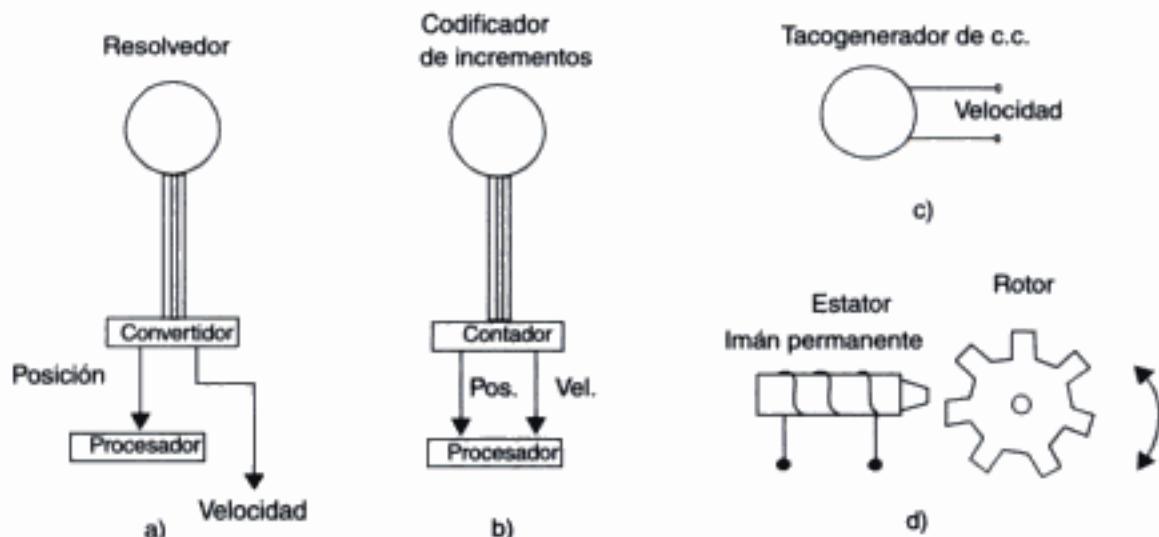


Figura 7.17: Medida de velocidad de ejes.

## 7.4 SENSORES DE PRESENCIA Y PROXIMIDAD

En primer lugar, cabe mencionar los sensores que detectan la presencia de un objeto. Los más simples son posiblemente los interruptores mecánicos tales como los habituales *finales de carrera*. En este caso, existe un contacto mecánico con el objeto que hace que se cierre un contacto eléctrico que genera una señal binaria.

Se emplean también otros sensores que detectan la presencia de objetos sin necesidad de que exista un contacto mecánico. Si el objeto es metálico se utilizan *sensores inductivos* basados en el cambio de inductancia que se produce por la presencia de un

objeto de material ferromagnético en un campo creado por una bobina arrollada situada junto a un imán permanente. La presencia del objeto modifica el campo induciendo en la bobina una corriente que se detecta midiendo la tensión en la bobina. Estos sensores, que son muy utilizados en la industria, permiten detectar la presencia de objetos a distancias muy cortas.

Otros sensores para detectar la presencia de objetos de materiales ferromagnéticos son los basados en el *efecto Hall*, que relaciona la tensión entre dos puntos de un material conductor o semiconductor con un campo magnético a través del material. Si se utiliza un imán permanente para crear un campo, el efecto Hall permite detectar dicho campo cuando no existe en él un material ferromagnético. Sin embargo, la presencia de un objeto de material ferromagnético hace que el campo se debilite, disminuyendo también la tensión a través del semiconductor, lo cual se utiliza para detectar la presencia del objeto. De esta forma, es posible construir sensores de silicio en los que se integran circuitos de detección.

Si el objeto no es de material ferromagnético, pueden emplearse *sensores capacitivos*, que permiten detectar cambios de capacidad inducidos por superficies cercanas. Los cambios de capacidad se detectan mediante circuitos apropiados, tales como osciladores que hacen que se inicie la oscilación cuando la capacidad es superior a un determinado umbral, o en circuitos en los cuales los desfases son proporcionales a las variaciones de capacidad.

La presencia de objetos puede detectarse también mediante *sensores ópticos* tipo barrera en los que el objeto intersecta la línea entre el emisor y el receptor, tal como se ilustra en la figura 7.18a. El principio de funcionamiento de interrupción del haz es el mismo que el que se empleaba en los codificadores ópticos. Se emplean diodos emisores de luz y fotodetectores tales como fotorresistores, fotodiódos o fototransistores. Los fotorresistores, o fotocélulas, son resistencias cuyo valor cambia con la intensidad de la luz recibida. Su empleo y conexión a microcontroladores es muy sencilla. Los fotodiódos tienen una mayor sensibilidad, y producen una señal lineal en un rango muy amplio de niveles de intensidad. Sin embargo, su salida necesita ser amplificada.

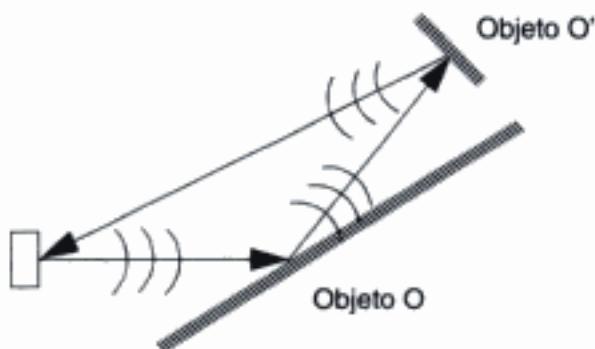
Se emplean también sensores ópticos en los que el emisor y el receptor se montan sobre el robot detectándose la presencia del objeto por la reflexión de la luz en el objeto tal como se ilustra en la figura 7.18b. El enfoque se consigue mediante lentes apropiadas. En particular, se utilizan sensores de infrarrojo cercano que son sensibles en longitudes de onda, tales como 880 nm, inmediatamente inferiores a la visible. A estos sensores se les denomina también *sensores de proximidad*. Sin embargo, conviene poner de manifiesto que, en principio, no suministran ninguna medida de distancia,



**Figura 7.18:** Sensores ópticos de presencia y proximidad.

sino tan solo una señal binaria indicando si existe o no un objeto próximo en un rango de distancia característico del sensor. No obstante, existen otros sensores de distancia con principios de funcionamiento similares. De esta forma forma, si se coloca un emisor y una línea de detectores, es posible calcular la distancia a un objeto próximo determinando cual de los receptores es el que recibe mayor intensidad luminosa del haz reflejado y aplicando triangulación. Existen también otros sensores ópticos de distancia, tales como los *sensores láser de tiempo de vuelo*, basados en la determinación del tiempo entre los pulsos emitidos y los que se reciben después de ser reflejados en el objeto, o los *sensores láser de medida de desfase* entre una onda emitida y la recibida reflejada por el objeto. Estos sensores de distancia no se consideran en este texto, dejándose para otros especializados en percepción del entorno.

Por último, conviene mencionar a los *sensores de ultrasonidos*. Estos sensores funcionan mediante el principio del tiempo de vuelo, emitiendo pulsos de sonido y determinando el tiempo hasta que se detecta una vez que ha sido reflejado por el objeto. De esta forma, teniendo en cuenta la velocidad de propagación del sonido, puede llegar a determinarse una distancia, tal como se efectúa en los sensores láser de medida de distancias. Sin embargo, se cuenta también con sensores de proximidad que indican si existe o no un objeto a una distancia menor que una dada, la cual puede programarse en el sensor. Conviene poner de manifiesto que las características de la superficie que refleja la onda y el ángulo de incidencia tienen una notable influencia en la eficiencia de estos sensores. En efecto, si el ángulo de incidencia excede un cierto valor crítico, la energía reflejada no entrará en la zona de detección. Pueden recibirse también reflexiones desperdigadas de otros objetos generando señales falsas tal como se pone de manifiesto en la figura 7.19. Nótese como, en este caso, pudiera no detectarse que el objeto O está muy próximo debido a que se recibe el rebote de O'.

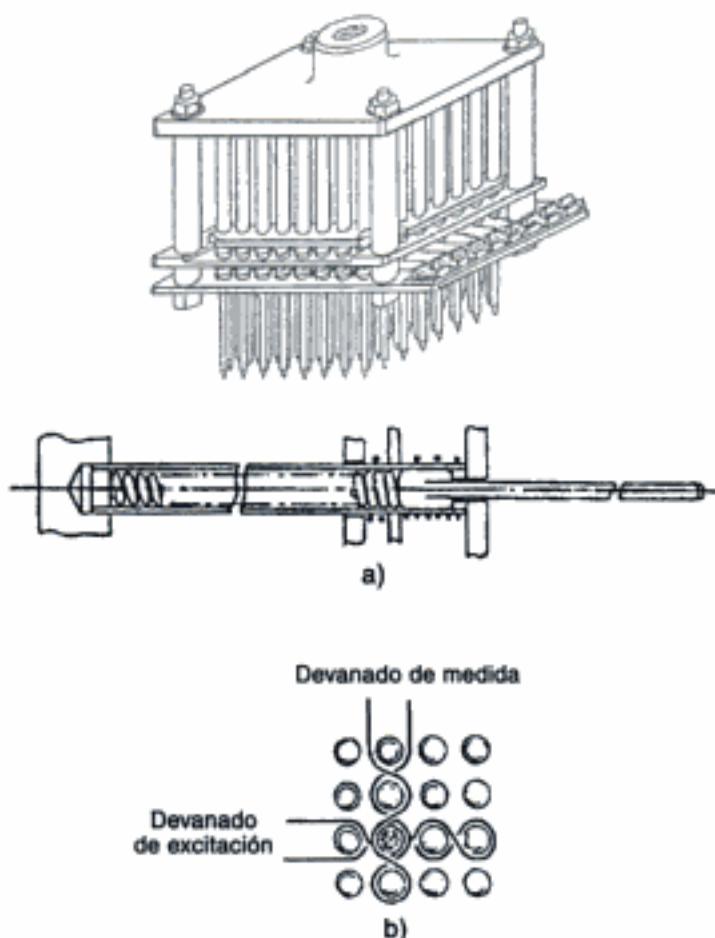


**Figura 7.19:** Problemas de rebotes en sensores de ultrasonidos.

## 7.5 SENSORES DE TACTO

Son sensores que pretenden suministrar al robot, típicamente un robot manipulador, una cierta percepción de tacto.

En la figura 7.20 (Klafter y otros, 1989) se ilustra un sensor de matriz de agujas. Las posiciones de las agujas se leen de forma inductiva o por microcontactos.

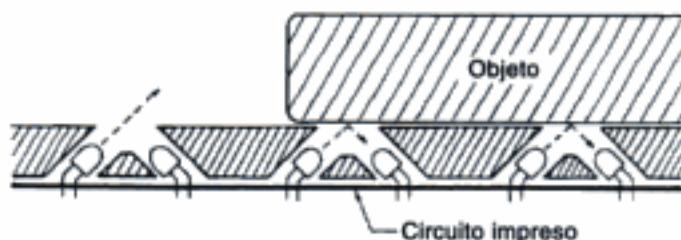


**Figura 7.20:** Sensor de contacto mediante matriz de agujas.

El sensor táctil óptico consta de una serie de orificios por los que actúan una serie de detectores ópticos por reflexión, tal como se ilustra en la figura 7.21 (Ferraté y otros, 1986).

Otro sensor de tacto es el elastómetro conductor. Consiste en una lámina de goma conductora que al ser presionada hace contacto con una base que dispone de zonas metalizadas, tal como se ilustra en la figura 7.22 (Klafter y otros, 1989).

Finalmente, cabe mencionar los sensores de barras de silicona. Constan de un retículo de barras de silicona dispuestas perpendicularmente sobre barras metálicas como se ilustra en la figura 7.23 (Ferraté y otros, 1986). La resistencia entre las barras de silicona y las metálicas varía según la presión ejercida. Obsérvese como este sensor puede ser considerado como de medida de esfuerzos tales como los que se tratan en el siguiente apartado.



**Figura 7.21:** Sensor táctil óptico.

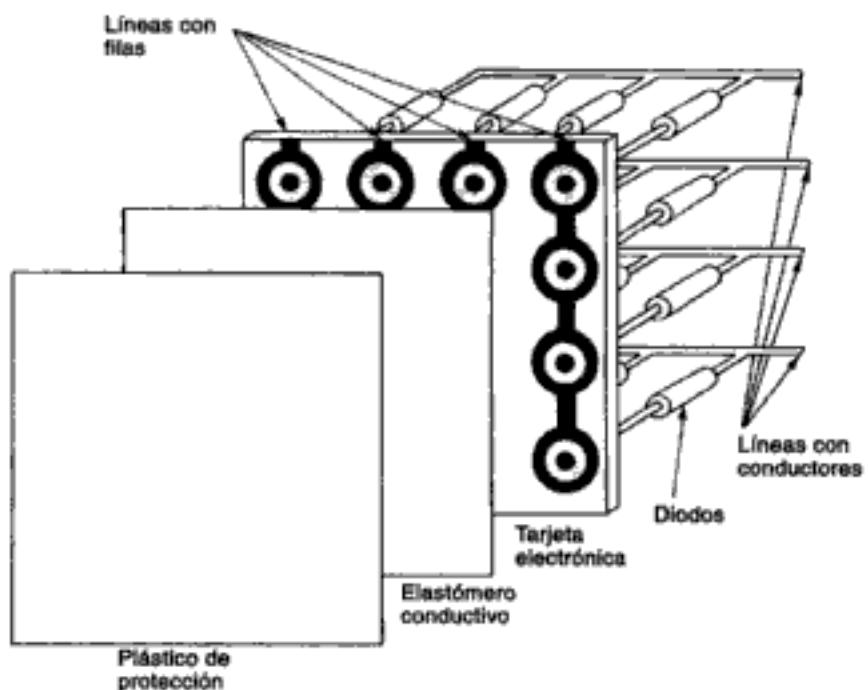


Figura 7.22: Elastómetro conductorio.

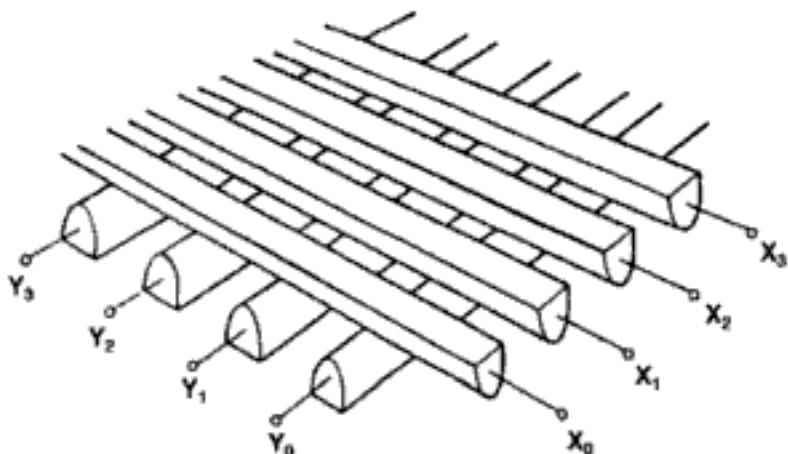


Figura 7.23: Barras de silicona.

## 7.6 MEDIDAS DE FUERZA Y PAR

En robótica suelen emplearse *galgas extensiométricas* en las cuales se relacionan también las deformaciones producidas por la aplicación de fuerzas con las variaciones de resistencia eléctrica (Russell, 1990). En particular, se emplea el efecto piezoresistivo. La resistencia entre dos caras opuestas de un bloque de material conductor viene dada por:

$$R = \frac{\rho l}{A} \quad (7.4)$$

siendo  $R$  la resistencia,  $\rho$  la resistividad,  $l$  la distancia entre caras, y  $A$  el área de las caras. El incremento de la resistencia por unidad de resistencia  $(\Delta R)/R$  viene dado por:

$$\frac{\Delta R}{R} = \frac{\Delta \rho}{\rho} + \frac{\Delta l}{l} - \frac{\Delta A}{A} \quad (7.5)$$

siendo  $\Delta \rho$  la variación de resistividad,  $\Delta l$  la variación de longitud, y  $\Delta A$  la variación de área.

Si  $\mu$  representa la relación de Poisson (extensión transversal/extensión axial) puede escribirse:

$$\frac{\Delta A}{A} = -2\mu \frac{\Delta l}{l} \quad (7.6)$$

sustituyendo (7.6) en (7.5) se obtiene:

$$\frac{\Delta R}{R} = \frac{\Delta \rho}{\rho} + (1 + 2\mu) \frac{\Delta l}{l} \quad (7.7)$$

de donde se deduce que el cambio de resistencia es el resultado del cambio en resistividad y en forma.

La sensibilidad de la galga viene dada por:

$$K = \frac{\Delta R/R}{\Delta l/l} \quad (7.8)$$

Cada material tiene una determinada sensibilidad. Así, la del cromo es  $K = 2,5$ , la del manganeso  $+0,5$ , la del platino  $4,8$  y la del níquel  $-12$ .

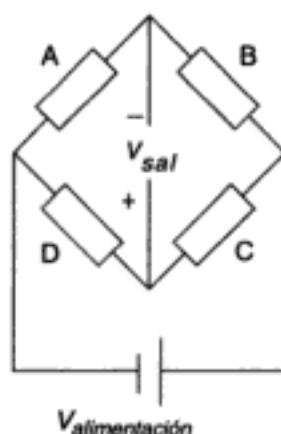
Existen también galgas extensiometrías de hoja metálica en las cuales la forma de la galga se diseña para minimizar la sensibilidad transversal.

Otro tipo de galgas extensiometrías son las de semiconductor. En ellas se emplea silicio o germanio dopados. En estas galgas la variación de la resistividad con la longitud es 100 veces mayor que en las metálicas. Sin embargo, las variaciones no son lineales y las medidas son sensibles a las variaciones de temperatura.

Para obtener la variación de resistencia suele emplearse un puente de Wheatstone, tal como se ilustra en la figura 7.24. Supóngase que las resistencias tienen un valor nominal  $R$ . Si se produce un cambio en  $\Delta R$  de la resistencia en la rama A, el cambio resultante en la tensión es:

$$\Delta V_{sal} = \left( \frac{V_{sal}}{2R} \cdot \frac{\Delta R}{(2 + \Delta R/R)} \right) = \frac{V_{sal} \Delta R}{4R} \quad (7.9)$$

El empleo de un puente de Wheatstone hace que la medida sea menos sensible con respecto a efectos tales como las variaciones de la temperatura.

**Figura 7.24:** Empleo de galgas extensiometrías.

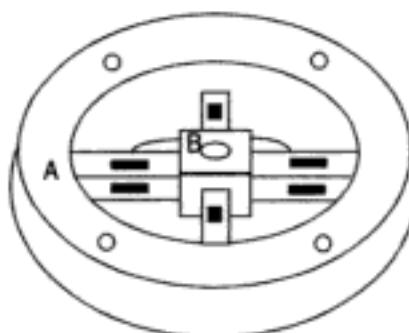
Las galgas extensiometrías se emplean también en sensores para medida de fuerzas y pares. De este tipo son los sensores de “muñequera” para robots manipuladores tal como el que se ilustra en la figura 7.25 (Russell, 1990).

En estos sensores existe un anillo externo A acoplado al brazo y un anillo interno B acoplado a la pinza. Suelen montarse cuatro galgas extensiometrías, una en cada radio. Existen también versiones miniaturizadas con galgas para medida de fuerzas y pares en un dedo.

Un sensor de tipo “muñequera” produce voltajes de salida relacionados con las fuerzas y pares mediante (Russell, 1990):

$$V = H \cdot F; \begin{bmatrix} v_1 \\ v_2 \\ \dots \\ v_n \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{16} \\ h_{21} & h_{22} & \dots & h_{26} \\ \dots & \dots & \dots & \dots \\ h_{n1} & h_{n2} & \dots & h_{n6} \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_2 \\ \dots \\ f_6 \end{bmatrix} \quad (7.10)$$

Para poder aplicar esta relación es necesario primero calcular los elementos de  $H$  en un proceso de calibración que requiere aplicar fuerzas y pares conocidos midiendo los voltajes que se obtienen. A partir de estos datos se obtiene la matriz  $H$ .

**Figura 7.25:** Sensor fuerza/parr de tipo muñequera.

Una vez conocida  $H$  es posible determinar  $F$  a partir de las medidas  $V$ . Nótese que, en el caso habitual en que  $H$  no es cuadrada, para poder resolver la ecuación es necesario emplear la seudoinversa

$$H^S = (H^T H)^{-1} H^T \quad (7.11)$$

## 7.7 SENsoRES DE NAVEGACIÓN

Estos sensores son de interés en robótica móvil. Se trata de medir posiciones, orientaciones, velocidades y aceleraciones de un determinado vehículo. Para ello pueden emplearse muy diversas tecnologías. En lo que sigue se describen algunas de estas tecnologías resumiendo los principios físicos involucrados.

### 7.7.1 Sensores Doppler

Son muy utilizados en vehículos aéreos y marítimos. Permiten medir una velocidad absoluta con respecto a la tierra. Se basan en el principio de Doppler por el cual se sabe que cuando una onda se refleja en una superficie que se está moviendo con respecto al emisor, se produce un desplazamiento en frecuencia que puede medirse para suministrar una medida de la velocidad del movimiento.

En la figura 7.26 se ilustra el principio de funcionamiento (Everett, 1995):

La velocidad del terreno con respecto al vehículo  $V_A$  se obtiene a partir de la velocidad  $V_D$  Doppler medida según:

$$V_A = \frac{V_D}{\cos \alpha} = \frac{c F_D}{2 F_0 \cos \alpha} \quad (7.12)$$

siendo  $F_D$  el desplazamiento en frecuencia observado (desplazamiento Doppler),  $F_0$  la frecuencia de transmisión,  $c$  la velocidad de la luz y  $\alpha$  el ángulo de incidencia.

En la práctica existen ondas reflejadas de diferentes frecuencias, por lo que es necesario un filtrado previo para quedarse con las frecuencias más representativas y medir  $F_D$ . Existen diferentes causas de errores en la medida, entre las que cabe mencio-

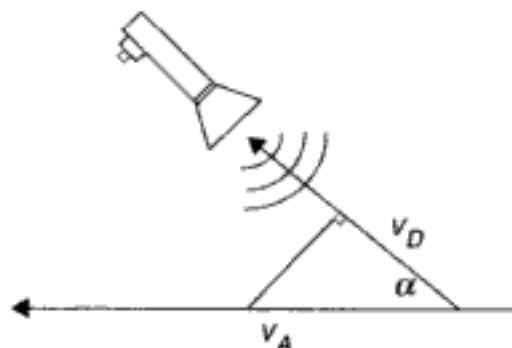


Figura 7.26: Sensor Doppler.

nar las interferencias de los lóbulos laterales de la respuesta en frecuencia, las componentes verticales de la velocidad introducidas por los movimientos verticales del vehículo ocasionados por las irregularidades del terreno, y las incertidumbres en el ángulo real de incidencia.

Los sensores que se emplean en vehículos marinos emplean energía acústica que se refleja en el fondo del mar. Sin embargo, en vehículos aéreos, se emplea radiofrecuencia, reflejándose las microondas en la superficie de la tierra.

Normalmente se disponen diversos sensores en orientaciones diferentes. Así, en los sistemas de navegación marítimos o aéreos suelen emplearse cuatro sensores colocados a 90° entre sí de diferencia de azimut y con el mismo ángulo de inclinación hacia abajo respecto al plano horizontal. El coste de estos sistemas de navegación es elevado.

En vehículos terrestres, y en particular en vehículos robóticos, ha sido frecuente emplear un único sensor en la dirección de marcha hacia abajo con un cierto ángulo con respecto al terreno, con el cual se mide la velocidad. Esta técnica ha sido empleada en vehículos agrícolas (Everett, 1995), en los cuales se trata de realizar de forma uniforme una operación durante la marcha del vehículo (sembrado, fumigación, etc.). Nótese que, en estos vehículos, debido a las características del terreno, normalmente se producen deslizamientos que hacen que la medida de la velocidad mediante odometría no sea posible.

### 7.7.2 Compás magnético

Se emplean para medir la orientación de un vehículo. Los compases son magnetómetros; es decir, sensores de medida de campo magnético, que se emplean para medir el campo de la Tierra.

Como se sabe, el campo magnético terrestre hace que una barra imanada se sitúe paralela con respecto a sus líneas de fuerza. Este principio se ha utilizado desde épocas remotas en las brújulas, en las cuales una aguja imanada colocada sobre un soporte vertical, que le sirve de apoyo y le permite girar libremente, apunta hacia el Norte. Si la brújula se coloca en un plano horizontal, puede emplearse para determinar orientaciones en dicho plano sobre una escala graduada.

Los primeros compases para la navegación consistían esencialmente en agujas magnetizadas que flotaban en agua. En la actualidad, los compases mecánicos emplean anillos magnéticos, o pares de barras, sujetos a un disco graduado de mica. Este conjunto flota en una mezcla de agua, alcohol o glicerina, de forma que puede rotar alrededor de un pivote. El fluido soporta el peso del conjunto y amortigua su movimiento. En navegación marítima es frecuente suspender el compás de un mecanismo con dos grados de libertad de rotación (*gimbal*) para conseguir un desacoplamiento del movimiento del barco.

Un aspecto muy importante es tratar de eliminar las perturbaciones del campo magnético terrestre, debidas a los materiales metálicos existentes en las proximidades, que generan una desviación en la medida de la orientación. Para ello se utilizan esferas de hierro con imanes permanentes ajustables en la base, mediante los cuales se compensan las desviaciones de las líneas de flujo geomagnéticas. Otro aspecto que es necesario tener en cuenta es la eliminación de las *declinaciones magnéticas* que hace

que el ángulo marcado varíe con el lugar y con el tiempo. Para ello se emplean cartas de *declinaciones*.

Los compases más populares, debido a su precio y reducidas dimensiones, son los basados en magnetómetros de núcleo saturable. Se basan en la saturación magnética del núcleo de un material altamente permeable empleando para ello una corriente alterna. Cuando un material altamente permeable se introduce en un campo magnético uniforme, las líneas de fuerza se dirigen por el camino de menor resistencia. Sin embargo, si el material está saturado por un efecto magnetizador suplementario, las líneas de flujo del campo externo no se modifican. El magnetómetro hace uso de este efecto de saturación para medir directamente la intensidad del campo magnético de los alrededores.

Existen distintos tipos de magnetómetros, dependiendo de las clases de núcleos y de devanados (Everett, 1995). Los núcleos con geometría de anillo son muy utilizados. Existe un devanado de excitación, así como devanados de medida enrollados sobre el anillo que se disponen de forma orthogonal. Las tensiones de continua en las salidas  $V_x$  y  $V_y$  de los devanados de medida, varían como funciones del seno y del coseno del ángulo  $\theta$  del sensor con el campo magnético terrestre. El valor instantáneo del ángulo se obtiene digitalizando los valores de las tensiones y realizando la operación de arco tangente:

$$\theta = \arctg \frac{V_x}{V_y} \quad (7.13)$$

Otra tecnología empleada en la realización de compases es la de los magnetómetros magnetoinductivos. Estos sensores sólo necesitan un solenoide por cada eje, por lo cual su consumo es mucho menor. El devanado de medida sirve como elemento inductivo en un oscilador de relajación  $L/R$  de baja potencia cuya inductancia efectiva viene influenciada por la componente, paralela al eje del devanado, del campo magnético del medio. Se demuestra (Everett, 1995) que la inductancia efectiva  $L$  viene dada por:

$$L = \mu_0 n^2 V \frac{dB}{dH} \quad (7.14)$$

siendo  $\mu_0$  la permeabilidad magnética del aire,  $n$  el número de vueltas del devanado por unidad de longitud,  $V$  el volumen del material del núcleo,  $B$  el flujo magnético total y  $H$  la fuerza de magnetización.

En la ecuación anterior  $dB/dH$  es la permeabilidad relativa  $\mu$  del material del núcleo. Esta permeabilidad depende, a su vez, de la fuerza de magnetización  $H$ , siendo esta dependencia lineal en un cierto rango.

Se aplica una corriente continua que genera un flujo en el solenoide provocando una fuerza de magnetización  $H_0$  centrada en la zona en la que  $\mu$  varía linealmente con  $H$ . La presencia de un campo exterior  $H_e$  provoca una desviación con respecto a  $H_0$ , dependiendo del cambio asociado de la permeabilidad efectiva.

El periodo de la salida del oscilador es proporcional a  $L/R$ , siendo  $L$  aproximadamente directamente proporcional a la permeabilidad efectiva  $\mu$ , que es una función de  $H$ . Cuanto mayor sea el campo magnético externo  $H_e$  menor es la inductancia efectiva aumentando la frecuencia del oscilador. En la práctica, el cambio de la frecuencia per-

mite medir la orientación del devanado de medida, o ángulo con respecto al campo geomagnético.

Otra tecnología diferente es la de los compases de efecto Hall. Éstos emplean sensores de efecto Hall los cuales, en presencia de un campo magnético externo, suministran una tensión de continua, a través de una región de semiconductor. Esta tensión es proporcional a la componente del campo magnético que es perpendicular a la dirección del flujo.

Por último, cabe mencionar a los compases magnetorresistivos y magnetoelásticos basados en propiedades de los materiales.

### 7.7.3 Giróscopos

Los giróscopos presentan la ventaja, con respecto a los compases magnéticos, de su inmunidad en relación con las anomalías electromagnéticas y ferromagnéticas que afectan a los compases. Por otra parte, pueden aplicarse en situaciones en las que no existe campo geomagnético apreciable (aplicaciones espaciales), o cuando existen campos locales que provocan importantes perturbaciones.

#### 7.7.3.1 Giróscopos mecánicos

Los giróscopos, o giroscopios, mecánicos están constituidos por un volante o masa que rota suficientemente rápido alrededor de un eje estando la masa distribuida en la periferia con objeto de que el momento de inercia del eje de rotación sea alto.

En la figura 7.27 se muestra una configuración típica de giróscopo. El rotor es accionado mediante un motor eléctrico, suspendiéndose mediante un par de cojinetes que deben ser de bajo rozamiento en cada extremo del eje. Estos cojinetes están soportados por un anillo circular, conocido como anillo gimbal interno, el cual, a su vez, pivota en un segundo juego de cojinetes que están unidos rígidamente a un anillo gimbal externo. Nótese que existen tres ejes: el eje de rotación del volante, o rotor, un eje perpendicular al de rotación del volante, que define el pivote interno, horizontal en la figura 7.27 (Everett, 1995), y un tercer eje perpendicular a los dos anteriores que define el pivote externo, vertical en la citada figura.

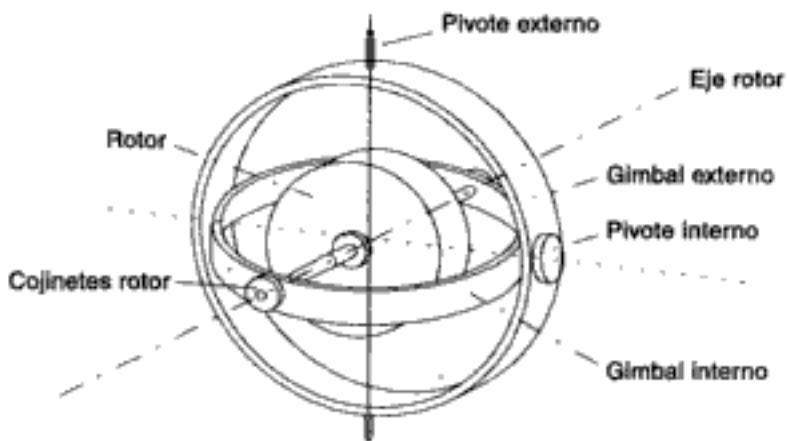


Figura 7.27: Giróscopo de volante con dos ejes.

Cuando el eje de rotación de la masa que gira se suspende en una estructura que le permite variar libremente su orientación, tal como sucede en la figura 7.27, aunque la estructura gire en el espacio, la dirección del eje de rotación de la masa que está girando tiende a mantenerse constante debido al equilibrio de momentos. Para ello es necesario que los cojinetes que sustentan el eje del rotor produzcan un rozamiento despreciable (eje flotando en el espacio), así como que la velocidad de giro sea alta y que la masa esté concentrada en la periferia (alto momento de inercia del rotor). En efecto, si la velocidad de giro se disminuye se observa como el eje de giro pierde su orientación inicial.

La propiedad de que el eje de rotación de la masa mantenga su orientación, se ha utilizado desde principios de este siglo para construir sensores de dirección. Si el eje de rotación del giróscopo es visto por un observador sobre la Tierra, observaría un movimiento aparente de rotación que es debido a la propia rotación de la Tierra sobre su eje. Sin embargo, si el eje del giróscopo está en la dirección del eje de giro de la Tierra (apuntando a la estrella polar) no se produce el movimiento aparente. Esta propiedad se utiliza en el *girocompás*, que puede considerarse como una configuración especial de giróscopo para realizar una búsqueda del norte. Cuando el eje de giro del rotor del giróscopo está apuntando al norte, es insensible a la rotación de la Tierra y, por tanto, no debe producirse en él ninguna inclinación. Por consiguiente, en caso de producirse, su medida debe suministrar el ángulo de orientación con el norte. Un giróscopo como el que se ilustra en la figura 7.27, se convierte en un buscador de norte fijando un peso a la parte inferior del rotor y creando así un péndulo que suministra una referencia gravitatoria.

En este punto conviene mencionar la diferencia que existe entre el girocompás, mencionado en el punto anterior, que no depende del campo magnético de la Tierra y el *compás giromagnético* en el cual se emplean giróscopos para mantener el compás en una plataforma horizontal, lo cual es condición necesaria para realizar de forma precisa la medida del campo magnético terrestre.

Otra propiedad interesante de los giróscopos es que si en la rueda que gira se le aplica un par de fuerzas perpendicular al eje, tendiendo a volcarla, se aprecia el fenómeno, en principio sorprendente, de que la rueda adquiere un lento movimiento de rotación, pero no alrededor del eje del par aplicado, sino alrededor del otro eje perpendicular a él y al eje de giro de la rueda. Este movimiento, al cual se conoce con el nombre de movimiento de precesión, se mantiene mientras existe la inercia giroscópica. Esta propiedad también puede emplearse para la medida del ángulo girado.

Así, si en la figura 7.27 se aplica una fuerza vertical hacia abajo en el extremo de eje del rotor, se produce un par en el eje horizontal que origina un movimiento de precesión de rotación alrededor del eje vertical. Si a la vez se gira la base del instrumento sobre el mismo eje vertical existirá una relación entre el movimiento de precesión y la velocidad de giro de la base. En efecto, cuanto más rápido se gire la base, mayor fuerza es necesaria para parar la precesión. Esta relación proporcional puede emplearse para medir la velocidad de giro sobre el eje vertical, o variación de azimuth. Asimismo, si en un barco se coloca el rotor del giróscopo en una dirección perpendicular al eje longitudinal del barco, se mediría la variación en el ángulo de alabeo ("roll") o rotación sobre el eje longitudinal del barco. Conviene poner de manifiesto que este giróscopo proporciona la variación de ángulo de rotación de un eje perpendicular al eje del rotor, mientras que en los giróscopos mencionados anteriormente, tales como el girocompás, el eje sobre el que se desea medir el giro coincide con el eje del rotor.

Existen otros giróscopos para medir la variación de ángulos que se basan en el efecto de la aceleración de Coriolis. Consisten esencialmente en dos masas oscilando radialmente en los extremos de fibras similares a cartílagos formando un diapasón. Cuando se produce un cambio en la orientación, las fuerzas de Coriolis generan pares proporcionales a las velocidades angulares de giro.

#### 7.7.3.2 Giróscopos electrónicos

Son normalmente sensores de velocidad angular que emplean el efecto de Coriolis mencionado en el párrafo anterior. Para ello se realizan micromecanizados del silicio configurando un anillo que se hace vibrar a una frecuencia de resonancia. El movimiento de rotación produce fuerzas de Coriolis que dependen de la velocidad de giro. La medida de la velocidad se obtiene determinando la diferencia de las vibraciones a diferentes ángulos. Un sensor típico puede tener dimensiones entre 2 y 3 milímetros y permite medir hasta 100 grados por segundo.

#### 7.7.3.3 Giróscopos ópticos

El dispositivo básico consiste en dos haces de láser en direcciones opuestas formando un circuito cerrado. Los patrones de interferencia formados dividiendo y mezclando porciones de los haces se utilizan para determinar la magnitud y dirección de la velocidad de rotación del dispositivo.

El anillo láser puede concebirse de forma ideal como un espejo en forma toroidal, hueco, como se muestra en la figura 7.28, donde la luz sigue un camino cerrado circular (Everett, 1995). Suponiendo una superficie de espejo ideal que refleja el 100% de la luz, la energía óptica dentro de la cavidad no se afecta teóricamente por el giro del espejo. Los haces de luz que se propagan en sentidos contrarios se refuerzan unos y otros para crear una onda estacionaria con picos y valores nulos de intensidad.

Las franjas de luz y oscuridad son análogas a las bandas opacas y transparentes en los discos rotatorios de los codificadores ópticos, y pueden ser contadas de forma óptica similar.

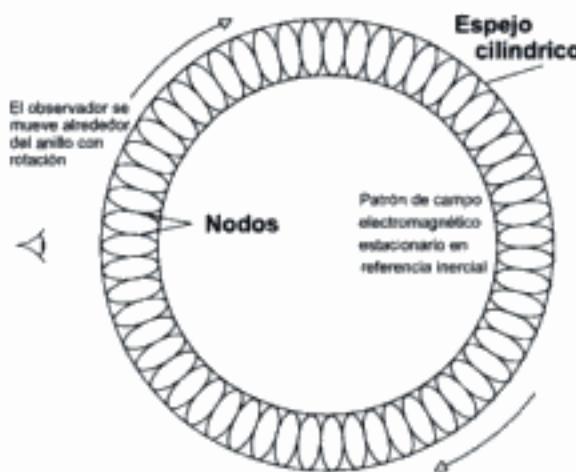


Figura 7.28: Ondas creadas por la propagación en sentido contrario de haces de luz.

En cada rotación completa del “donut”, el detector vería un número de picos igual a dos veces la longitud del camino óptico del haz dividida por la longitud de onda de la luz. En un láser He-Ne de longitud de onda 632,8 nanómetros, en un típico camino cerrado de 2,4 pulgadas de diámetro, existirían 300.000 longitudes de onda y, por tanto, 600.000 nodos, permitiendo conseguir aproximadamente medio millón de cuentas por revolución.

En Everett (1995) se distinguen cinco diferentes tecnologías para realizar giróscopos ópticos: resonadores ópticos activos (giróscopos de anillo de láser), resonadores ópticos pasivos, interferómetros de fibra óptica en bucle abierto (analógico), interferómetro de fibra óptica en bucle cerrado (digital) y resonadores de fibra óptica.

#### 7.7.4 Sistemas de navegación inercial

Un Sistema de Navegación Inercial (INS) emplea acelerómetros para determinar la aceleración en cada uno de los tres ejes de movimiento. Las velocidades y posiciones se obtienen mediante doble integración.

Existen diferentes tecnologías para construir acelerómetros. Una posible estructura consiste en una masa colocada en el extremo de una viga en voladizo, situada entre dos electrodos fijos. El amortiguamiento del sistema se consigue mediante películas de gas entre la masa y los electrodos. La masa y la viga en voladizo se fabrican en silicio. En esta estructura capacitiva el movimiento de la masa se detecta midiendo la capacidad variable en el condensador.

Los acelerómetros se montan en una plataforma estabilizada giroscópicamente para mantener constante la orientación.

En vehículos robóticos terrestres es frecuente que las aceleraciones sean reducidas, por lo que la relación señal/ruido es baja, dificultando la medida correcta. Asimismo, las posiciones estimadas acumulan derivas, necesitándose sensores de gran calidad para obtener valores aceptables del error de posición. Es también importante que la plataforma esté perfectamente estabilizada, por lo cual es necesario emplear giróscopos de calidad para la medida de las orientaciones.

Por todo ello, los sistemas de navegación inercial, para obtener precisiones razonables, son muy costosos. Sin embargo, el empleo de giróscopos ópticos y acelerómetros de estado sólido tiende a mejorar la situación.

#### 7.7.5 Estimación de posición de vehículos basada en estaciones de transmisión. Aplicación del GPS

La estimación de la posición absoluta mediante estaciones de transmisión ha venido utilizándose desde hace decenios en aplicaciones marítimas y aeronaúticas. Puede distinguirse entre los que emplean estaciones fijas y los de estaciones móviles.

La estimación mediante estaciones fijas se basa en el empleo del método conocido como navegación hiperbólica (Forssell, 1991), en el cual se mide la diferencia de tiempos en que una señal de radio emitida desde dos estaciones distintas tarda en ser recibida por el observador. Si la velocidad de propagación es constante, la diferencia de tiempos es equivalente a la diferencia en distancias entre cada una de las estaciones

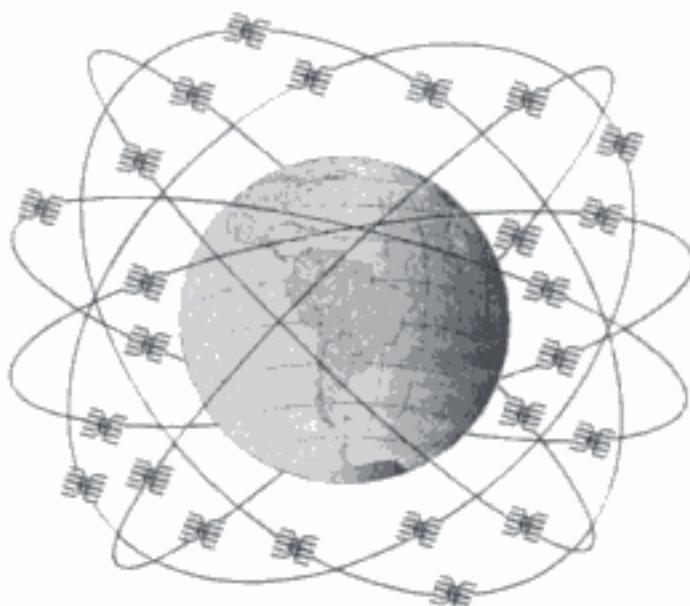
y el observador. Utilizando distintos pares de estaciones es posible estimar la posición del vehículo de forma bastante fiable. El orden de magnitud de los errores es de varias decenas de metros.

Entre los sistemas de posicionamiento mediante estaciones móviles destaca el denominado Global Positioning System, más conocido por sus siglas GPS (Logsdon, 1995), en el cual las estaciones de transmisión operan desde satélites. En los últimos años la estimación de posición mediante GPS ha cobrado un gran interés en vehículos autónomos navegando en exteriores (Schonberg y otros, 1999; Rodríguez-Castaño y otros, 1999).

El GPS fue creado por el Departamento de Defensa de Estados Unidos y suministra información de la posición y velocidad 24 horas al día, en todas las condiciones meteorológicas y con cobertura en todo el mundo. En el GPS cabe distinguir entre los segmentos espacial, de control y de usuario.

El segmento espacial consta de una constelación de 24 satélites (figura 7.29), en seis planos separados 55 grados con períodos orbitales de 12 horas a una altura aproximada de unos 20.200 km. Esto proporciona a un receptor de GPS de seis a doce satélites a la vista en cualquier punto de la superficie del planeta en cualquier instante. Cada satélite envía una señal característica que lo identifica (Space Vehicle Number y Pseudorandom Code Number) en la que va incluida información sobre su posición, estado, tiempo, parámetros orbitales y otros datos. Los satélites transmiten en dos bandas de frecuencia: la L1, centrada en 1.572,42 MHz, y la L2, centrada en 1.227,60 MHz. La portadora L1 está modulada por el código C/A ("Coarse Acquisition"), y por el código P ("Precision") que está encriptado para usos militares o especialmente autorizados. La portadora L2 está modulada sólo por el código P.

El segmento de control consta de una estación maestra, situada en Colorado Springs (existe otra de reserva en la costa oeste de EE.UU.), cinco estaciones monitor, que supervisan los satélites gracias a las señales difundidas por éstos (posición, movi-



**Figura 7.29:** Constelación de satélites del GPS.

miento orbital y temporización) y tres estaciones para envío de datos a los satélites. Los datos de las estaciones monitor se envían a la estación maestra donde se recalculan parámetros orbitales, y se vuelven a enviar a los satélites a través de las estaciones de envío. Los satélites difunden los nuevos datos recibidos en vez de los antiguos.

El segmento de usuario consta de los equipos que reciben las señales de los satélites. Estos equipos deben procesar simultáneamente las señales de un mínimo de cuatro satélites para obtener medidas precisas.

El receptor de GPS mide el tiempo que tarda la señal del satélite en llegar desde éste hasta el receptor. La distancia a que se halla el equipo del satélite se calcula conociendo el tiempo en que la señal sale del satélite, el tiempo en que es recibida y la velocidad de propagación de la señal. Si el receptor tuviera un reloj perfecto, sincronizado de forma perfecta con los de los satélites, bastarían tres medidas para determinar las tres coordenadas de la posición. Sin embargo, un reloj de elevada precisión es muy costoso, por lo cual se emplea un cuarto satélite para determinar el error en el reloj del receptor o "bias"  $B$ . El receptor mide el tiempo que tarda en llegar la información GPS de cada uno de los cuatro satélites y obtiene las medidas denominadas pseudorangos ( $PR_1, PR_2, PR_3, PR_4$ ) multiplicando por la velocidad de la luz  $c$ . De esta forma, se plantean cuatro ecuaciones con cuatro incógnitas:

$$(X_i - U_X)^2 + (Y_i - U_Y)^2 + (Z_i - U_Z)^2 = (PR_i - Bc)^2; \quad i = 1 \dots 4 \quad (7.15)$$

donde  $(X_i, Y_i, Z_i)$  son las posiciones de cada uno de los satélites, transmitidas de forma codificada a una frecuencia de 50 Hz por los mismos satélites, y las cuatro incógnitas son  $(U_X, U_Y, U_Z, B)$ .

El GPS tiene diferentes fuentes de error, entre las que cabe mencionar: actividad de la ionosfera (cambios en el retraso de propagación por los niveles de ionización produciendo menores errores durante la noche y afectando más a las regiones polares y ecuatoriales); retrasos en la troposfera o parte más baja de la atmósfera (varía en función de condiciones atmosféricas, altura sobre nivel del mar y ángulo de elevación del satélite, siendo el retraso menor para satélites cercanos a 90 grados con respecto al horizonte y mayor para satélites situados a 0 grados); multirayecto (la señal GPS llega a la antena después de rebotar en alguna superficie u objeto cercano produciendo interferencia con la señal que llega directamente del satélite, lo que origina errores bastante apreciables); radiointerferencias (degradación de señal producida por una emisión de radio o de radar dirigida a la antena de GPS); disponibilidad selectiva (degradación intencionada de la precisión introducida por el Departamento de Defensa de Estados Unidos consistente en que la información difundida por los satélites es incorrecta, pudiendo obtenerse la máxima precisión sólo en el caso de conocer un código encriptado); errores entre el tiempo real del reloj del satélite y el tiempo que se envía en el mensaje difundido por el satélite; error en el reloj del receptor (diferencia entre el tiempo GPS del receptor y el verdadero tiempo GPS, que puede ser diferente para cada receptor, dependiendo del tipo de oscilador utilizado); y errores de "ephemeris", o serie de parámetros orbitales que permiten calcular la velocidad y posición del satélite (diferencia entre posición y velocidad verdaderas del satélite y la que se difunde). Por todo ello, la precisión del sistema está entre 50 y 100 metros.

La mencionada precisión del GPS puede aumentarse de forma importante empleando un sistema diferencial (DGPS). En particular, el DGPS permite reducir la ma-

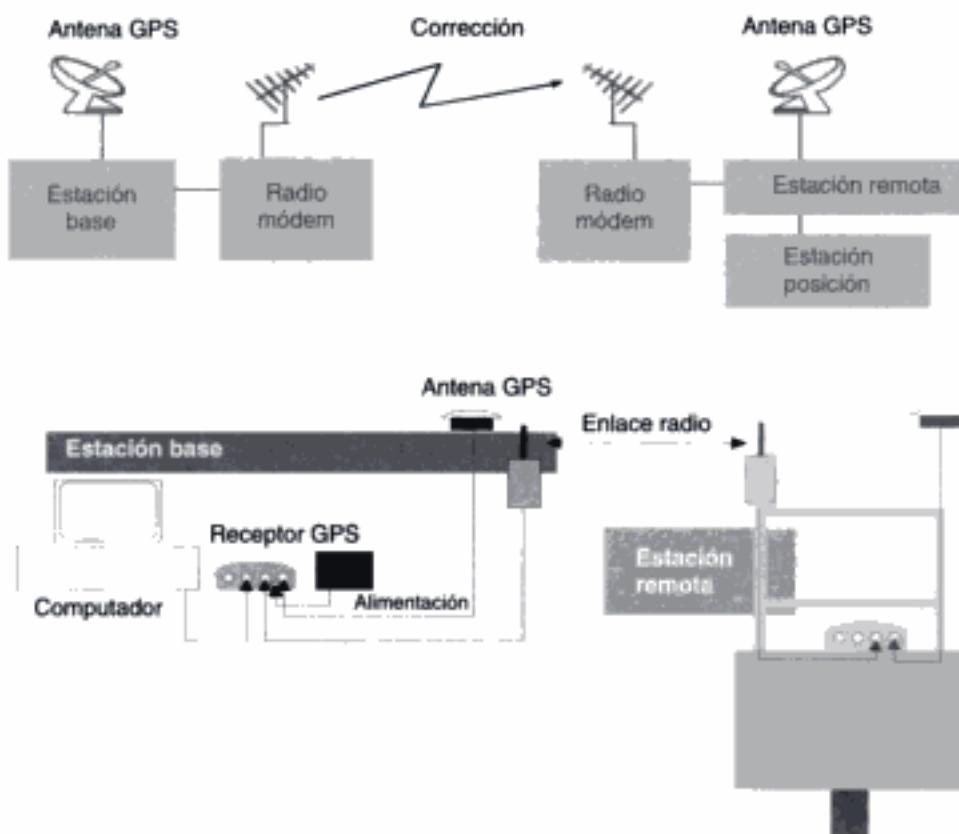


Figura 7.30: GPS diferencial.

yor parte de los errores mencionados en el punto anterior, excluyendo los errores por multirayecto y los debidos al reloj del receptor.

El esquema básico se muestra en la figura 7.30. En el DGPS las estaciones de recepción funcionan a pares. Cada pareja consta de una estación base y una estación remota. Es posible también conectar varias estaciones remotas por cada estación base. Para que el funcionamiento en modo diferencial sea efectivo, es necesario que las estaciones base y remota reciban simultáneamente las señales de los mismos satélites. Si las estaciones están suficientemente próximas (separadas menos de 50 km) los errores se consideran iguales, eliminándose mediante correcciones diferenciales. Si la separación entre las estaciones base y remota se hace mayor, los errores están menos correlacionados, reduciendo la efectividad de las correcciones diferenciales con lo cual la precisión que se obtiene es menor. Utilizando DGPS es posible obtener precisiones en la posición entre 1 y 5 m, llegándose incluso hasta precisiones de pocos centímetros.

### 7.7.6 Sensores para vehículos autoguiados industriales

Los vehículos autónomos industriales o AGV ("Autonomous Guided Vehicles") se diseñan para que se desplacen autónomamente por un entorno, típicamente industrial, siguiendo un determinado circuito.

El circuito se establece empleando distintos medios. El más habitual es el de un cable enterrado que crea un campo magnético que debe ser detectado por un sensor si-

tuado en la parte inferior del vehículo. Son los denominados *vehículos filoguiados*. Se entierran uno o más cables en una ranura realizada en el suelo de 0,32 a 1,9 cm de ancho y de 1,27 a 3,81 cm de profundidad. Estos cables forman el circuito que debe seguir el AGV.

Un emisor de baja frecuencia produce en el cable una corriente con intensidad y frecuencia determinadas, que induce un campo magnético en el suelo que es captado por las bobinas detectoras del dispositivo de guiado del vehículo, tal como se ilustra en la figura 7.31 (Castleberry, 1991).

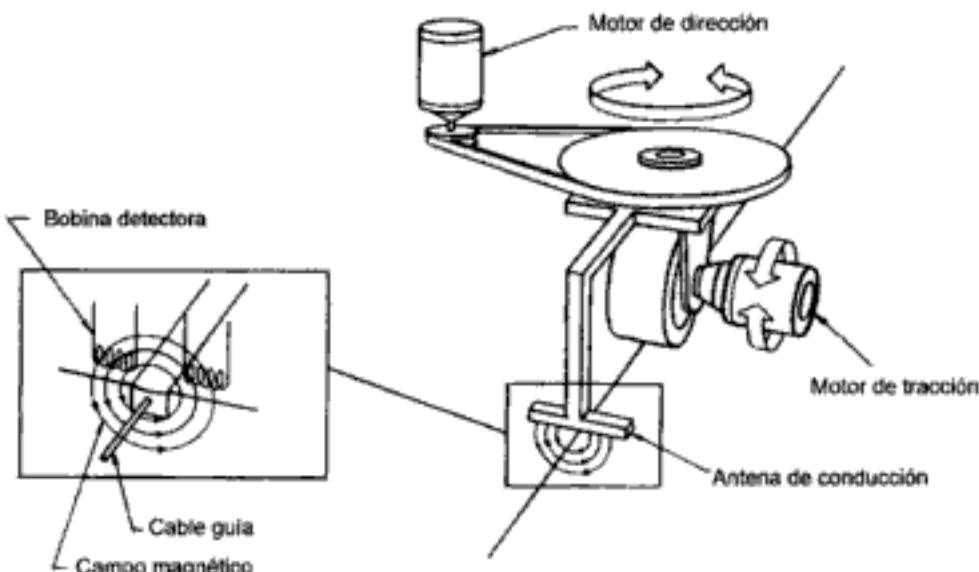


Figura 7.31: Sensor de vehículo filoguiado.

Para evitar la necesidad de enterrar un cable, existen otras técnicas de guiado que pretenden obtener una mayor flexibilidad. Así, se utilizan técnicas basadas en el establecimiento de marcas en la superficie tales como las bandas magnéticas, ópticas, o de partículas con determinadas propiedades químicas, que requieren sensores específicos.

La técnica de guiado con bandas magnéticas en el suelo sufre muchas interferencias. En el optoguiado, el AGV incorpora un sensor que permite detectar una línea o banda de un color apropiado pintada en el suelo. En este caso, no hay fuente de energía en el circuito y se evitan interferencias, pero la banda debe mantenerse limpia para que pueda ser detectada.

El denominado guiado químico emplea una cinta con partículas fluorescentes sobre la superficie. Estas partículas se estimulan con la luz ultravioleta generando una emisión luminosa en un espectro en el que normalmente no existen otras emisiones en el ambiente.

Existen también AGV con técnicas de guiado mediante balizas, que constituyen un paso más en el incremento de flexibilidad de los vehículos industriales autónomos ya que no requieren mantener una línea en el suelo. En este caso, se combina la odometría con la detección de balizas dispuestas en el trayecto que debe recorrer el vehículo.

Se emplean distintos tipos de emisiones tales como láser de baja potencia, infrarrojos, ultrasonidos o emisiones de radio. Así, por ejemplo, existen sistemas en los que un receptor giratorio instalado sobre el vehículo barre continuamente el espacio en busca de las señales emitidas por las balizas. En McGuillem y Rappaport (1998) se describe un sistema consistente en una serie de balizas de luz infrarroja y un dispositivo óptico giratorio sobre el vehículo que es capaz de detectar esta emisión. Conocida la velocidad de giro del detector óptico, el sistema determina los ángulos entre balizas consecutivas a partir del tiempo que transcurre entre las detecciones de éstas.

La técnica descrita en el párrafo anterior emplea rastreo con un detector sobre el vehículo y balizas fijas que emiten. Es posible también utilizar sobre el vehículo un mecanismo de rastreo con un sensor emisor/detector en cuyo caso las balizas pueden ser simplemente balizas pasivas que reflejen la emisión de forma apropiada, o bien transpondedores. Otra posibilidad es el empleo de emisores rotantes y detectores fijos en las balizas.

Asimismo, es posible colocar marcas especiales tales como señales luminosas, códigos de barras, etc., y situar una o varias cámaras sobre el vehículo con objeto de detectar estas marcas.

La posición del vehículo se calcula empleando relaciones trigonométricas a partir de medidas de distancias o ángulos a un cierto número de balizas (González y Ollero, 1996). Para navegación en el plano, los ángulos de observación con respecto al eje horizontal X de dos balizas son suficientes. Si las medidas se refieren a ángulos entre las balizas, serán necesarias tres de ellas. Cuando el sensor permite obtener distancias a las balizas, con sólo dos balizas pueden presentarse ambigüedades (dos intersecciones de las circunferencias de distancia constante a las balizas). Sin embargo, si se combina información angular y de distancia puede ser suficiente con una baliza, aunque en este caso el sensor es más complejo.

En general, es conveniente que puedan detectarse simultáneamente más del número mínimo de balizas mencionado en el párrafo precedente con objeto de reducir la incertidumbre y minimizar el efecto del ruido en las medidas (Case, 1986).

## 7.8 CONCLUSIONES

---

La robótica involucra una gran cantidad de sensores. En este capítulo se han considerado, en primer lugar, los sensores internos que colocados sobre el robot, suministran información sobre las posiciones, velocidades y fuerzas desarrolladas, prescindiendo de los sensores de percepción del entorno. Entre los sensores que suministran los ángulos de las articulaciones se ha prestado especial atención a los codificadores ópticos, ya que son los más utilizados. Se introducen también máquinas eléctricas que permiten medir posiciones y velocidades. A continuación se han introducido sensores de proximidad, contacto, fuerza y par, que suministran una información primaria sobre la interacción con el entorno. En el capítulo se han considerado también sensores de navegación, que son fundamentales en robótica móvil. Entre ellos, se incluyen medidas de orientación mediante giróscopos y compases magnéticos, medidas de velocidad e incluso medidas de posición absoluta mediante GPS. Por último, se han presentado también sensores específicos para el guiado de vehículos industriales (AGV).

## 7.9 REFERENCIAS

- Berney G.C., 1985. "Intelligent instrumentation". Prentice Hall.
- Case M.P., 1986. "Single Landmark Navigation by Mobile Robots". SPIE Conference on Intelligent Robots and Computer Vision. Vol. 727.
- Castleberry G.A., 1991. "The AGV handbook". Braun-Brumfield, Inc. Ann Arbor, Michigan.
- Everett H.R., 1995. "Sensors for mobile robots. Theory and application". A.K. Peters.
- Ferraté G. y otros, 1986. "Robótica Industrial", Marcombo.
- Forssell B., 1991. "Radionavigation Systems". Prentice Hall International.
- González J. y A. Ollero, 1996. "Estimación de la posición de un robot móvil". Informática y Automática. Vol. 29, núm. 4, pp. 3-18.
- Hunter R.P., 1987. "Automated process control systems. Concepts and Hardware". Prentice Hall.
- Klafter R., T.A. Chmielewsky y M. Negin, 1989. "Robotic Engineering An Integrated Approach". Prentice Hall.
- Logsdon T., 1995. "Understanding the Navstar: GPS, GIS and IVHS". Van Nostrand Reinhold.
- McGuillem C.D. y T.S. Rappaport, 1998. "Infra-red Location System for Navigation of Autonomous Vehicles". IEEE Int. Conference on robotics and Automation.
- Muller R.S., R.T. Howe, S.D. Senturia, R.L. Smith y R.M. White, 1991. "Microsensors". IEEE Press.
- Rodríguez-Castaño A., A. Ollero y G. Heredia, 1999. "Fuzzy Path Tracking of Autonomous Vehicles using GPS". Proceedings of the 1999 Eusflat-Estylf Joint Conference, pp. 477-480. Palma de Mallorca, 22-25 septiembre.
- Russell R.A., 1990. "Robot Tactil Sensing". Prentice Hall.
- Schonberg, T., M. Ojala, J. Suomela, A. Torpo y A. Halme, 1999, Positioning an autonomous off-road vehicle by using fused DGPS and inertial navigation, *Proc. of the 2nd. Int. Conf. on Intelligent Autonomous Vehicles (IAV95)*, Helsinki, Finlandia, pp. 226-231.

# Control de las articulaciones de un robot manipulador

---

En este capítulo se tratan métodos para generar las actuaciones que permiten controlar las articulaciones de un robot manipulador. En primer lugar, se presentan las diversas estrategias de control. A continuación, se contempla el control desacoplado de las articulaciones. En el tercer apartado, se estudia el control basado en un modelo, presentándose la técnica del par computado. En los apartados cuarto y quinto se introducen las técnicas de control adaptativo y control con aprendizaje. En el quinto apartado, se tratan las técnicas de control en el espacio cartesiano. Finalmente, se introduce el control de esfuerzos. En este capítulo también se utiliza MATLAB y Simulink para simular las estrategias de control.

## 8.1 ESTRATEGIAS DE CONTROL DE ARTICULACIONES

---

Se consideran estrategias de control en bucle cerrado de las articulaciones. La estrategia más simple es el control de cada articulación de forma independiente o “desacoplada” del resto de las articulaciones. Es habitual emplear un controlador Proporcional + Integral + Derivada (PID) digital en cada articulación. En este controlador, la señal de control en bucle cerrado es el resultado de la suma de tres términos que son proporcionales al error, derivada del error e integral del error. Los pares de acoplamiento se tratan como perturbaciones que es necesario anular mediante el control PID. Se presenta el problema del ajuste más conveniente de las ganancias del controlador PID, ya que, dependiendo de la carga e incluso de la posición dentro del espacio de trabajo, se necesitarían valores diferentes de dichas ganancias.

Si se pretenden considerar de forma apropiada los acoplamientos entre las articulaciones, es necesario emplear el modelo dinámico completo del manipulador. Este modelo se presentó en el capítulo 5. Es un modelo no lineal con gran cantidad de parámetros. La dificultad principal de estas técnicas de control es el conocimiento de los mencionados parámetros que, en algunos casos, son difíciles de medir o estimar. Por

otra parte, pueden producirse variaciones de los parámetros debido al desgaste o modificaciones no previstas en las condiciones de trabajo. Debido a estas razones, en apartados posteriores se presentan métodos de control adaptativo y control con aprendizaje.

Las estrategias mencionadas en los párrafos precedentes son empleadas para el control de la posición y, eventualmente, de la velocidad del manipulador. Sin embargo, en muchos casos, interesa controlar también los esfuerzos que se realizan en la actuación del manipulador. Esta necesidad es particularmente evidente en operaciones en las que tiene lugar un contacto entre el manipulador y los objetos del entorno, ejerciéndose una fuerza, tales como las que se presentan en el pulido de una superficie, inserción, etc. En estos casos, interesa que, además del control de posición, existan articulaciones que se controlen teniendo en cuenta los esfuerzos. Para ello, es muy conveniente el empleo de sensores de esfuerzos.

Conviene también poner de manifiesto que en este capítulo se consideran exclusivamente brazos robóticos rígidos. Es decir, no se tienen en cuenta aspectos relacionados con vibraciones de la estructura y otros problemas que se presentan cuando existen elementos flexibles.

## 8.2 CONTROL DESACOPLADO DE ARTICULACIONES

Considérese el control de una articulación simple, según el modelo que se presenta en el apartado 5.1.2 (figura 5.1).

Supóngase que el par  $\tau$  se genera mediante un motor de corriente continua. Si este motor está controlado por el inducido (figura 8.1), el par  $\tau$  generado por el motor es proporcional a la intensidad  $i$  que circula por el inducido

$$\tau = K_a i \quad (8.1)$$

La intensidad  $i$  está relacionada con la tensión de control  $V$  aplicada en el inducido mediante la ecuación del circuito del inducido, en la cual hay que tener en cuenta la fuerza contraelectromotriz que genera una tensión proporcional a la velocidad  $\theta'$  de giro del motor. Por tanto, se tiene:

$$Ri + L\dot{i} = V - K_b \theta' \quad (8.2)$$

siendo  $K_b$  la constante de fuerza contraelectromotriz.  $R$  y  $L$  son, respectivamente, la resistencia e inductancia del circuito del inducido.

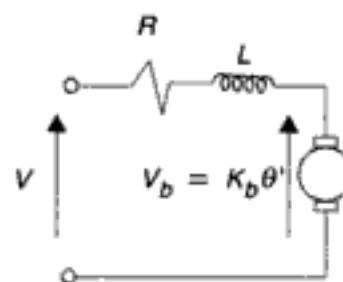
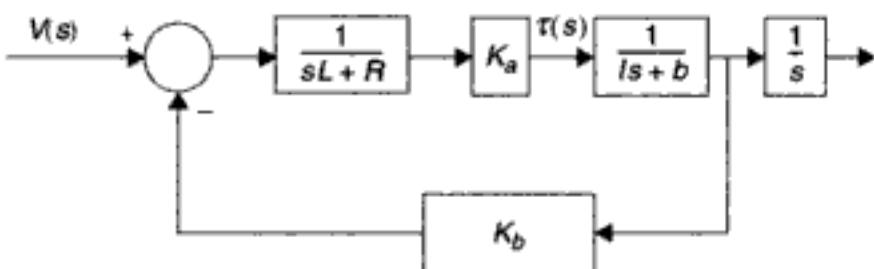


Figura 8.1: Articulación con motor controlado por el inducido.



**Figura 8.2:** *Modelo simple en bucle abierto de una articulación con motor de corriente continua.*

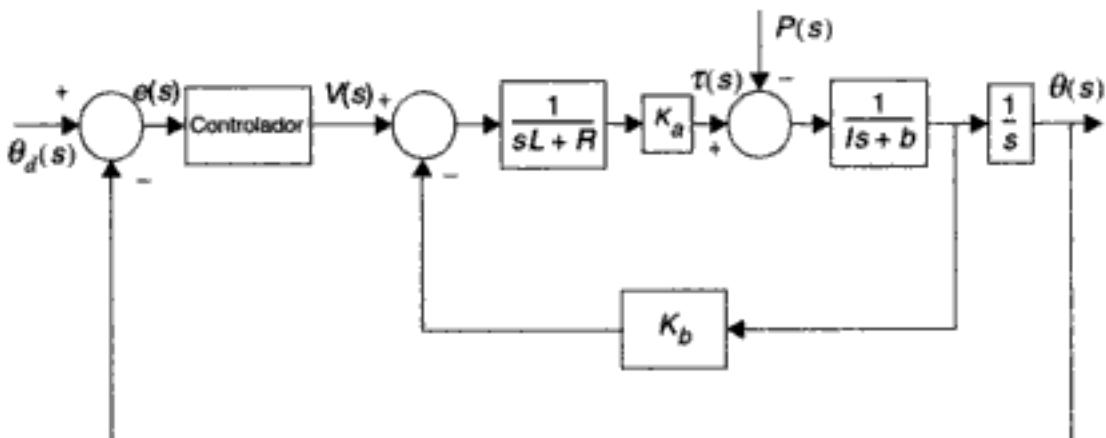
Tomando transformadas de Laplace se llega al diagrama de bloques que se ilustra en la figura 8.2. En sistemas de posicionamiento, es frecuente despreciar la inductancia  $L$ , en cuyo caso la función de transferencia viene dada por:

$$\frac{\theta(s)}{V(s)} = \frac{K_a}{(RIs + Rb + K_a K_b)s} \quad (8.3)$$

siendo  $s$  la variable de la transformada de Laplace,  $I$  el momento de inercia en el eje de salida,  $b$  el coeficiente de rozamiento y  $R$  la resistencia del circuito del inducido.

Lo más frecuente es que entre el eje del motor y el de la articulación existan engranajes de un cierto factor de multiplicación. En el caso ideal, estos engranajes pueden modelarse mediante una simple constante que multiplica el par y divide la velocidad de giro del eje de salida. Sin embargo, los engranajes suelen introducir también zonas muertas, fricciones no modeladas e histéresis. Estos efectos no lineales tienen un notable efecto perjudicial en el comportamiento dinámico del manipulador. Por esta razón, en los últimos años, se han desarrollado robots de accionamiento directo en los cuales se suprimen los engranajes. En el resto de este apartado no se consideran los mencionados efectos no lineales. Por tanto, el modelo que relaciona la transformada de Laplace de la tensión de control  $V(s)$  con la transformada de Laplace de la variable articular  $\theta(s)$  es el que se muestra en la figura 8.2.

El sistema de control en bucle cerrado es el que se muestra en la figura 8.3.



**Figura 8.3:** *Control en bucle cerrado de la articulación.*

Considérese un controlador lineal con función de transferencia  $G_c(s)$ . La función de transferencia en bucle abierto es:

$$\frac{\theta(s)}{E(s)} = \frac{K_a G_c(s)}{s(RI s + Rb + K_a K_b)} \quad (8.4)$$

La estrategia de control más simple es la del control proporcional al error del bucle de control. Si la constante de proporcionalidad es  $K_p$ , la función de transferencia del controlador es  $G_c(s) = K_p$ .

El empleo de controladores PD (proporcional + derivativo) es muy frecuente. En este caso, la acción de control viene dada por:

$$\left[ K_p e(t) + K_v \frac{de(t)}{dt} \right] \quad (8.5)$$

siendo  $K_v$  la constante de proporcionalidad de la derivada del error. La función de transferencia del controlador resulta ser  $G_c(s) = K_p + sK_v$ .

En este caso, la función de transferencia en bucle cerrado es:

$$\frac{\theta(s)}{\theta_d(s)} = \frac{K_a K_v s + K_a K_p}{s^2 RI + s(Rb + K_a K_b + K_a K_v) + K_a K_p} \quad (8.6)$$

Considérese que el efecto de las otras articulaciones se traduce en una perturbación caracterizada mediante un par  $P$ . Este par es el resultado de efectos gravitatorios y centrífugos de las otras articulaciones, así como de otras posibles perturbaciones externas. Se supone que el par afecta al eje del motor, tal como se indica en la figura 8.3.

Se trata ahora de generar una actuación de control que, además de posicionar el eje de la articulación respondiendo a la señal de referencia, anule también el efecto de la perturbación. El par resultante sobre el eje es ahora (figura 8.2):

$$\tau(s) = (s^2 I + sb)\theta(s) + P(s) \quad (8.7)$$

Suponiendo que la señal de referencia es nula, la función de transferencia entre la posición del eje de salida y la perturbación viene dada por:

$$\frac{\theta(s)}{P(s)} \Big|_{\theta_d = 0} = \frac{-R}{s^2 RI + s(Rb + K_a K_b + K_a K_v) + K_a K_p} \quad (8.8)$$

Aplicando el principio de la superposición con la ecuación anterior y la (8.6) se obtiene:

$$\theta(s) = \frac{K_a(K_p + sK_v)\theta_d(s) - RP(s)}{s^2 RI + s(Rb + K_a K_b + K_a K_v) + K_a K_p} \quad (8.9)$$

Finalmente, cabe mencionar el empleo de estrategias de control PID en las cuales existe un término adicional proporcional a la integral del error del bucle de control. En este caso, la señal de control viene dada por:

$$K_p e(t) + K_v \frac{de(t)}{dt} + K_i \int_0^t e(t) dt \quad (8.10)$$

El problema principal de todas las estrategias consideradas en este apartado es que el par de perturbación  $P$  depende en cada instante de las posiciones y velocidades de las distintas articulaciones. Por consiguiente, sería necesario ajustar un valor diferente de las constantes  $K_p$ ,  $K_v$  y  $K_i$ , dependiendo del instante considerado.

En el siguiente apartado se estudian estrategias de control que permiten contemplar el acoplamiento de las articulaciones de forma más apropiada. Para ello se emplea el modelo dinámico completo del manipulador, tal como se presentó en el capítulo 5. En particular, se utiliza el modelo dinámico en forma cerrada.

## 8.3 CONTROL BASADO EN EL MODELO DINÁMICO

### 8.3.1 Ejemplos introductorios

Considérese nuevamente el problema del control de posición de un eje de una articulación. En este caso, se adopta un modelo de fricción en el cual se añade un término no lineal a la fricción viscosa considerada en el apartado anterior. En este caso, se tiene (ver capítulo 5):

$$\tau = I\theta'' + b\theta' + c \operatorname{sgn}(\theta') \quad (8.11)$$

Supóngase que se conocen los parámetros  $b$ ,  $c$ . En este caso, cabe pensar en escoger una ley de control que compense de forma perfecta la fricción, reduciendo el problema al del control en bucle cerrado de un doble integrador. Para ello se adopta como par de control:

$$\tau = I\tau_r(e) + \tau_c \quad (8.12)$$

siendo  $\tau_r(e)$  una cierta función del error del bucle de control y  $\tau_c$  el par compensador que, para el modelo de la ecuación (8.11), se toma:

$$\tau_c = b\theta' + c \operatorname{sgn}(\theta') \quad (8.13)$$

Sustituyendo (8.12) y (8.13) en (8.11) se tiene un comportamiento en bucle cerrado dado por:

$$I\tau_r(e) + b\theta' + c \operatorname{sgn}(\theta') = I\theta'' + b\theta' + c \operatorname{sgn}(\theta') \quad (8.14)$$

Por consiguiente, si el modelo es perfecto, es posible cancelar los sumandos y eliminar también  $I$  obteniéndose:

$$\tau_r(e) = \theta'' \quad (8.15)$$

La situación se ilustra en la figura 8.4, en la cual los bloques representan integradores (equivalentes a bloques de  $1/s$  en el diagrama de bloques con la transformada de Laplace).

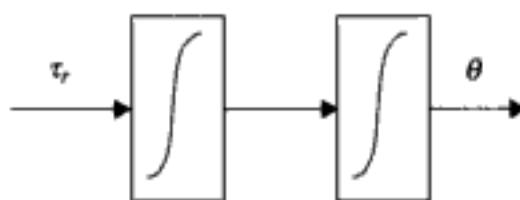


Figura 8.4: Reducción a doble integrador.

Si se escoge  $\tau_r(e)$  como la estrategia de control PD definida mediante:

$$\tau_r = \theta''_d + K_v e' + K_p e \quad (8.16)$$

siendo  $e = \theta_d - \theta$ , y se sustituye en la ecuación (8.15), se tiene la ecuación del error en bucle cerrado:

$$e'' + K_v e' + K_p e = 0 \quad (8.17)$$

Considérese ahora el péndulo invertido, que se contemplaba también en el capítulo 5 como un manipulador con una única articulación de rotación (figura 5.2). El modelo viene dado por

$$\tau = I\theta'' + b\theta' + c \operatorname{sgn}(\theta') + mgl \cos(\theta) \quad (8.18)$$

siendo  $I = ml^2$ ,  $m$  la masa,  $l$  la longitud y  $g$  la aceleración de la gravedad.

En este caso, se escogería la ley de control

$$\tau = ml^2 \tau_r(e) + \tau_c \quad (8.19)$$

donde  $\tau_r(e)$  viene dada por (8.16) y  $\tau_c$  es ahora:

$$\tau_c = b\theta' + c \operatorname{sgn}(\theta') + mgl \cos(\theta) \quad (8.20)$$

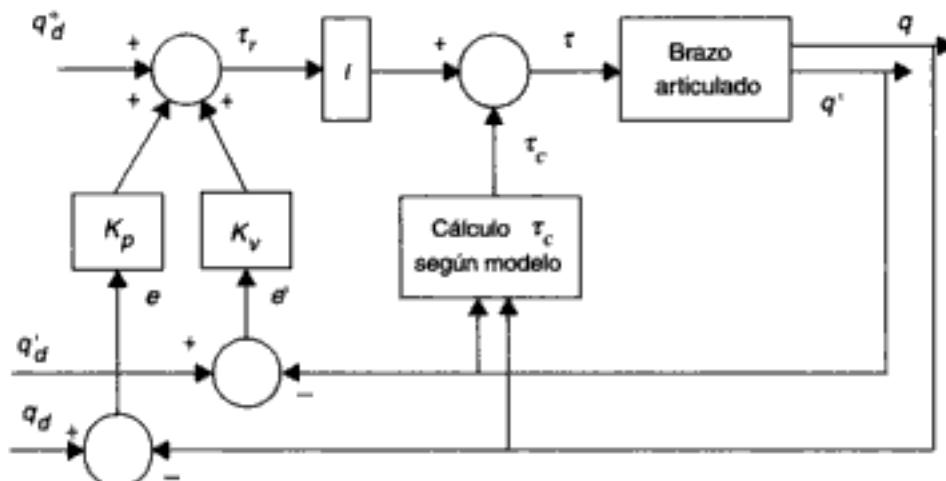


Figura 8.5: Control de una articulación.

De esta forma se llega a también a la ecuación (8.15) y escogiendo la ley de control (8.16) proporcional al error del ángulo y al error de velocidad, se llega a la ecuación del error (8.17).

El esquema de control correspondiente se muestra en la figura 8.5.

Para ajustar los valores de los coeficientes  $K_p$  y  $K_v$  puede buscarse el amortiguamiento crítico, tal como se mencionaba en el apartado anterior.

En efecto, obsérvese que la ecuación (8.17) corresponde a la de un sistema lineal de segundo orden con denominador de la función de transferencia

$$s^2 + 2\delta\omega_n s + \omega_n^2 \equiv s^2 + K_v s + K_p = 0 \quad (8.21)$$

siendo  $\delta$  el coeficiente de amortiguamiento y  $\omega_n$  la frecuencia natural no amortiguada. Identificando coeficientes se tiene:

$$\begin{aligned} \omega_n &= \sqrt{K_p} \\ 2\delta\sqrt{K_p} &= K_v \end{aligned} \quad (8.22)$$

$$\delta = \frac{K_v}{2\sqrt{K_p}} \quad (8.23)$$

Por tanto, si, por ejemplo, se desea amortiguamiento crítico  $\delta = 1$ , se tendrá:

$$K_v = 2\sqrt{K_p} \quad (8.24)$$

Obsérvese que la estrategia presentada en los ejemplos es una técnica de linealización basada en el conocimiento del modelo.

### 8.3.2 Control basado en el modelo. Par computado

La estrategia del par computado constituye una generalización de las leyes de control introducidas en los ejemplos del apartado anterior.

Se trata de compensar los efectos gravitatorios, centrífugos y rozamientos generando un par apropiado.

Considérese un manipulador como un sistema dinámico representado mediante (capítulo 5):

$$\tau = M(q)q'' + V(q, q') + G(q) + F(q, q') \quad (8.25)$$

siendo:

- $q$  un vector de  $n$  variables articulares,
- $M(q)$  una matriz de masas  $n \times n$  simétrica definida positiva,
- $V(q, q')$  un vector de  $n$  fuerzas centrífugas y de Coriolis,
- $G(q)$  un vector de  $n$  términos de gravedad, y
- $F(q, q')$  un vector de  $n$  componentes con efectos de fricciones.

Se supone que se dispone de un modelo dado por:

$$\tau = \hat{M}(q)q'' + \hat{V}(q, q') + \hat{G}(q) + \hat{F}(q, q') \quad (8.26)$$

siendo  $\hat{M}(q)$ ,  $\hat{V}(q, q')$ ,  $\hat{G}(q)$  y  $\hat{F}(q, q')$  estimaciones de las matrices y vectores correspondientes.

El método del *par computado*, o de la dinámica inversa, consiste en la aplicación del par de control:

$$\tau = \hat{M}(q)\tau_r(e) + \tau_c \quad (8.27)$$

siendo

$$\begin{aligned} \tau_c &= \hat{V}(q, q') + \hat{G}(q) + \hat{F}(q, q') \\ \tau_r &= q''_d + K_v e' + K_p e \end{aligned} \quad (8.28)$$

donde  $e$ ,  $e'$  son vectores de  $n$  errores de posición y velocidad:

$$e = q_d - q; \quad e' = q'_d - q' \quad (8.29)$$

y  $K_p$ ,  $K_v$  son ahora las matrices diagonales  $n \times n$ :

$$K_p = \begin{bmatrix} K_{p1} & & & \\ & K_{p2} & & \\ \dots & \dots & \dots & \\ & & & K_{pn} \end{bmatrix} \quad (8.30)$$

$$K_v = \begin{bmatrix} K_{v1} & & & \\ & K_{v2} & & \\ \dots & \dots & \dots & \\ & & & K_{vn} \end{bmatrix} \quad (8.31)$$

Sustituyendo  $\tau$  en la ecuación del sistema dinámico (8.25) se obtiene:

$$\hat{M}[q''_d + K_v e' + K_p e] + \hat{V} + \hat{G} + \hat{F} = Mq'' + V + G + F \quad (8.32)$$

de donde, restando en ambos lados  $\hat{M}q''$ , se llega a:

$$e'' + K_v e' + K_p e = \hat{M}^{-1}[(M - \hat{M})q'' + (V - \hat{V}) + (G - \hat{G}) + (F - \hat{F})] \quad (8.33)$$

Si el modelo es exacto, el lado derecho de la igualdad es cero. Sin embargo, en la práctica, el modelo no se conoce de forma perfecta. Asimismo, existirán fricciones, envejecimiento y variaciones de la dinámica debido a la carga en la pinza. Como resultado de todo ello, es necesario considerar un par de perturbación  $\tau_d$ . En este caso, el comportamiento del error se modela como:

$$\ddot{e}^n + K_v \dot{e}' + K_p e = \hat{M}^{-1} \tau_d \quad (8.34)$$

Nótese que existirán acoplamientos entre articulaciones debido a que, en el lado derecho de la ecuación,  $\hat{M}^{-1}$  no es diagonal.

Si  $\tau_d$  se hace suficientemente pequeña, el comportamiento dinámico de los errores de las articulaciones se aproximarán al de los sistemas lineales de segundo orden:

$$\ddot{e}^n + K_v \dot{e}' + K_p e = 0 \quad (8.35)$$

En este caso, se tienen  $n$  ecuaciones lineales de segundo orden desacopladas, tales como la (8.17). Los coeficientes de estas ecuaciones se determinan de la misma forma que en los ejemplos; es decir, normalmente se trata de conseguir un amortiguamiento crítico.

Nótese también que la acción de control de la segunda de las ecuaciones (8.28) emplea el término  $K_v \dot{e}' + K_p e$ ; es decir, es de tipo PD. Si en lugar de esta acción se empleara la PID:

$$\tau_r = q''_d + K_v \dot{e}' + K_p e + K_i \int_0^t e dt \quad (8.36)$$

siendo  $K_i$  una matriz diagonal, en lugar de la ecuación (8.34) se tendría:

$$\ddot{e}^n + K_v \dot{e}' + K_p e + K_i \int_0^t e dt = M^{-1} \tau_d \quad (8.37)$$

El término integral eliminaría errores en régimen permanente, pero podría empeorar el transitorio.

En la figura 8.6 se muestra la implantación del sistema de control de par computado, según las ecuaciones (8.27) y (8.28).

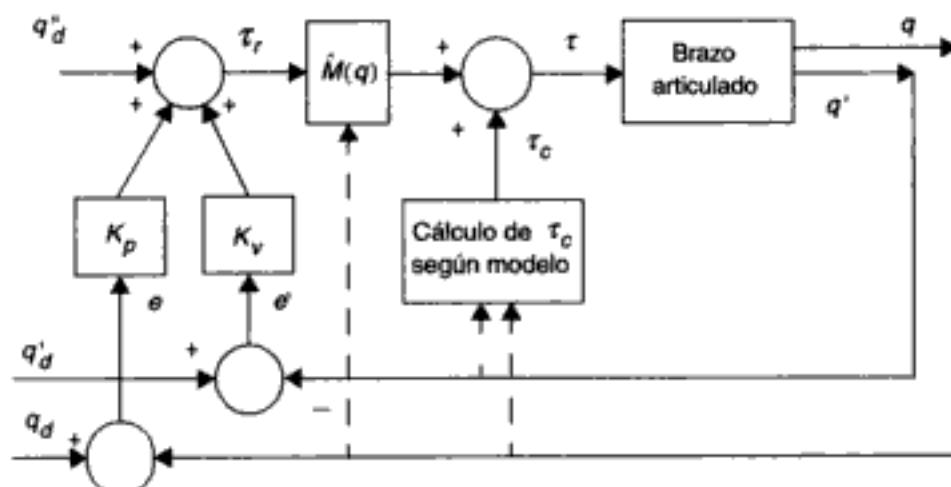


Figura 8.6: Esquema de control del par computado.

Obsérvese que el cálculo de  $\tau_r$  en el bucle exterior de la figura 8.6, según la segunda ecuación (8.28), es de fácil implantación. Sin embargo, para calcular  $\tau_c$  según (8.27) es necesario evaluar en tiempo real  $\hat{M}(q)$ ,  $\hat{V}(q, q')$ ,  $G(q)$ ,  $S\hat{F}(q, q')$ , lo cual requiere una importante cantidad de cálculos en manipuladores con un cierto número de articulaciones. Por consiguiente, la frecuencia a la que puede cerrarse el bucle de control interno en la figura 8.6 es menor.

Las consideraciones del párrafo anterior llevan a pensar en una estructura de control en la cual se intercambie la posición de los bucles de control de la figura 8.6. Es decir, situar un bucle de control interior rápido para el cálculo de  $\tau_r$ , aplicando la segunda ecuación (8.28), y uno exterior más lento que emplee el modelo dinámico. La estrategia se muestra en la figura 8.7.

Una posible implantación del bucle interno consiste en el empleo de controladores PD convencionales de las articulaciones.

Sin embargo, esta estrategia no da como resultado el mismo comportamiento dinámico desacoplado (8.35) de los errores de las variables articulares que se consigue con el esquema de la figura 8.6.

En efecto, en este caso, el par generado por el bucle interno es:

$$K_v e' + K_p e \quad (8.38)$$

al cual se le suma un par generado por el bucle externo

$$\hat{M}(q)q''_d + \hat{V}(q, q') + \hat{G}(q) + \hat{F}(q, q') \quad (8.39)$$

Por tanto, el comportamiento dinámico en bucle cerrado viene dado por:

$$\begin{aligned} \hat{M}(q)q''_d + \hat{V}(q, q') + \hat{G}(q) + \hat{F}(q, q') + K_v e' + K_p e = \\ = M(q)q'' + V(q, q') + G(q) + F(q, q') \end{aligned} \quad (8.40)$$

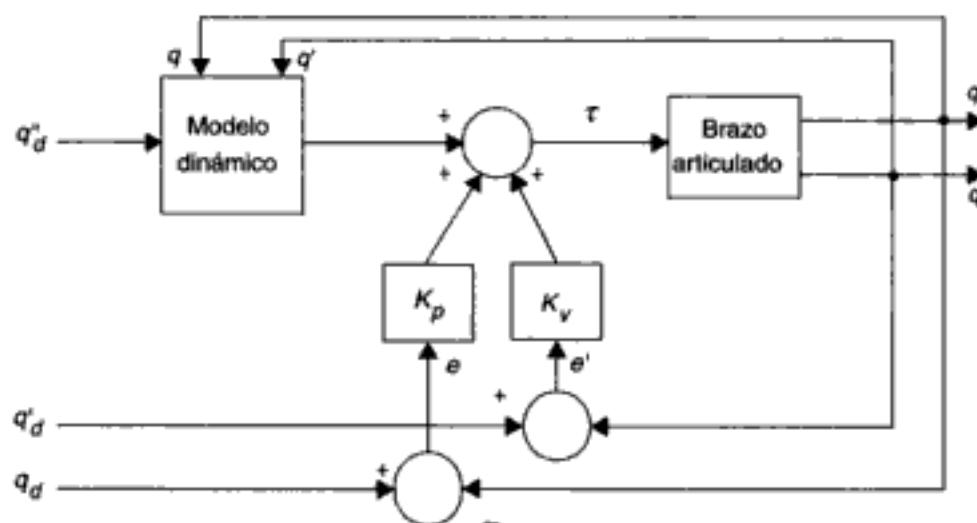


Figura 8.7: Implementación del par computado con cálculo de modelo en bucle externo.

Si el modelo es perfecto y no se producen perturbaciones externas, se tendrá:

$$M(q)\ddot{q} + K_v\dot{e} + K_p e = 0 \quad (8.41)$$

Las ecuaciones de error vienen dadas por:

$$\ddot{e} + M^{-1}(q)[K_v\dot{e} + K_p e] = 0 \quad (8.42)$$

Es decir, no se produce el desacoplamiento deseado de las articulaciones que se conseguía con el esquema de la figura 8.6.

En cualquier caso, conviene poner de manifiesto que la dificultad principal de la estrategia de control mediante el par computado es la necesidad de conocer el modelo. Suponiendo que se conociera solamente la matriz de masas, puede aplicarse como par de control

$$\tau = \hat{M}\tau_r = \hat{M}[q_d'' + K_v\dot{e} + K_p e]$$

Es decir, se anularía  $\tau_c$  en (8.27).

Si se supone que la matriz de masas es exacta,  $\hat{M} = M$ , la ecuación dinámica del error viene dada por:

$$\ddot{e} + (K_v\dot{e} + K_p e) = M^{-1}(V + G + F) \quad (8.43)$$

Asimismo, en algunas implantaciones se consideran también los términos gravitatorios en la ley de control, con lo cual se emplea

$$\tau = \hat{M}(q)\tau_r + \hat{G}(q) \quad (8.44)$$

siendo igual que antes  $\tau_r = q_d'' + K_v\dot{e} + K_p e$ . En este caso, si la estimación de los términos gravitatorios es perfecta, se logra eliminar el término  $G$  en el lado derecho de la ecuación (8.43), con lo cual suelen reducirse los errores en régimen permanente que se producen cuando no se consideran los efectos gravitatorios. Cabe recordar también que, a velocidades reducidas, el efecto del término  $V$  no es apreciable y que los términos de rozamiento  $F$  pueden considerarse de forma desacoplada en las distintas articulaciones. Por todo ello, la técnica de control basada en el empleo de la matriz de masas y términos gravitatorios, tal como se indica en (8.44) puede dar buenos resultados en muchas aplicaciones.

Nótese también que la estrategia de control más simple del PID desacoplado de cada articulación, puede incluirse dentro de la formulación del par computado. Para ello basta emplear el par de control (8.36). El comportamiento dinámico del error viene dado por:

$$q_d'' + K_v\dot{e} + K_p e + K_i \int_0^t edt = M\ddot{q} + V + G + F \quad (8.45)$$

de donde:

$$\ddot{e} + \left( K_v\dot{e} + K_p e + K_i \int_0^t edt \right) = (M - I)\ddot{q} + V + G + F \quad (8.46)$$

siendo  $I$  la matriz unidad de dimensiones  $n \times n$ .

Se recuerda que esta estrategia es de implantación muy simple. Nótese que se ignoran acoplamientos dinámicos de articulaciones. La elección de ganancias es muy importante. Normalmente, estas ganancias se eligen para conseguir un amortiguamiento crítico en el centro del espacio de trabajo. Es claro que en otros puntos se tendrán configuraciones sobre o subamortiguadas.

En Canudas y otros (1996) se estudia la estabilidad del sistema de control en bucle cerrado cuando se aplican las estrategias de control PD, PID y control con compensación de la gravedad.

La técnica del par computado no es la única para contemplar todos los acoplamientos dinámicos entre articulaciones. Así, en Canudas y otros (1996) se presentan otras técnicas basadas en teoría de control, tales como el control basado en el método de Liapunov (Slotine y Li, 1988) y el control basado en pasividad (Ortega y Spong, 1989). Estos métodos pueden considerarse en el marco del control adaptativo, tal como se hará en el apartado 8.4. Otra estrategia es la de control robusto (Spong y Vidyasagar, 1989).

Antes de terminar este apartado, conviene poner de manifiesto que en todo lo anterior se ha empleado una formulación en tiempo continuo. Si las frecuencias del comportamiento dinámico del manipulador son significativas con respecto a la frecuencia con la que se cierra el bucle de control, es necesario emplear una formulación en tiempo discreto para realizar el análisis del sistema de control.

Otro aspecto que conviene tener en cuenta en la selección de la frecuencia con la que se cierra el bucle de control, es la consideración de las características de las señales de referencia. En esta selección deben contemplarse también las características frecuenciales de las perturbaciones y ruidos que afectan al bucle de control.

### Ejemplo 8.1

Considérese el manipulador plano con dos articulaciones de rotación estudiado en capítulos precedentes que se muestra en la figura 8.8. Se supondrá, en primer lugar, que su modelo dinámico se conoce de forma perfecta. Los parámetros son:

$$\hat{m}_1 = m_1 = 2 \text{ kg}; \quad \hat{m}_2 = m_2 = 1 \text{ kg}; \quad \hat{l}_1 = l_1 = 3 \text{ m}; \quad \hat{l}_2 = l_2 = 1 \text{ m}$$

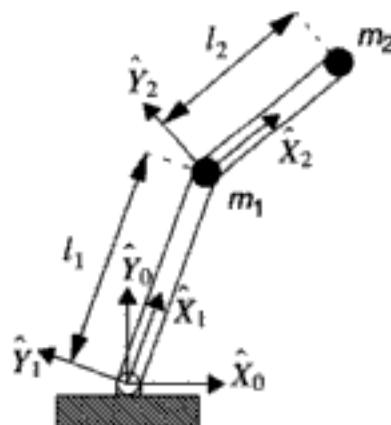


Figura 8.8: Manipulador plano con dos articulaciones de rotación.

Se aplicará la técnica del par computado usando un control PD para seguir la siguiente trayectoria:

$$\begin{aligned}\theta_{1d} &= A \operatorname{sen}\left(\frac{2\pi t}{T}\right) \\ \theta_{2d} &= A \cos\left(\frac{2\pi t}{T}\right)\end{aligned}\quad (8.47)$$

siendo la amplitud  $A = 0,15$  rad y el periodo  $T = 4$  s. Se diseñará un controlador para lograr amortiguamiento crítico y tiempo de establecimiento (en alcanzar  $\pm 5\%$  del valor final) de  $\pi/5$  s.

Teniendo en cuenta que en sistemas de segundo orden se verifica de forma aproximada la relación  $t_s = \pi/(\delta\omega_n)$ , siendo  $t_s$  el tiempo de establecimiento, y que con amortiguamiento crítico se tiene  $\delta = 1$ , se escoge  $\omega_n = 5$ .

Por tanto, teniendo en cuenta (8.22) y (8.24), los parámetros del PD se escogen como:

$$\begin{aligned}k_p &= \omega_n^2 = 25 \\ k_v &= 2\omega_n = 10\end{aligned}\quad (8.48)$$

El par de control será:

$$\tau = \hat{M}(q)(q_d'' + K_v e' + K_p e) + \hat{V}(q, q') + \hat{G}(q) \quad (8.49)$$

donde:

$$\begin{aligned}K_p &= \begin{bmatrix} k_p & 0 \\ 0 & k_p \end{bmatrix} = \begin{bmatrix} 25 & 0 \\ 0 & 25 \end{bmatrix} \\ K_v &= \begin{bmatrix} k_v & 0 \\ 0 & k_v \end{bmatrix} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}\end{aligned}\quad (8.50)$$

Se trata ahora de simular la respuesta del sistema con esta ley de control. Para ello, puede emplearse Simulink, tal como se indica en el apéndice. Para ello se utiliza el esquema que se muestra en la figura 8.9.

Los bloques grises corresponden a la dinámica estimada, mientras que los rayados contienen la dinámica real del manipulador. En este ejemplo, la estimación de los parámetros es perfecta, por lo que la matriz de parámetros cinemáticos y dinámicos para los bloques grises y rayados será la misma:



$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

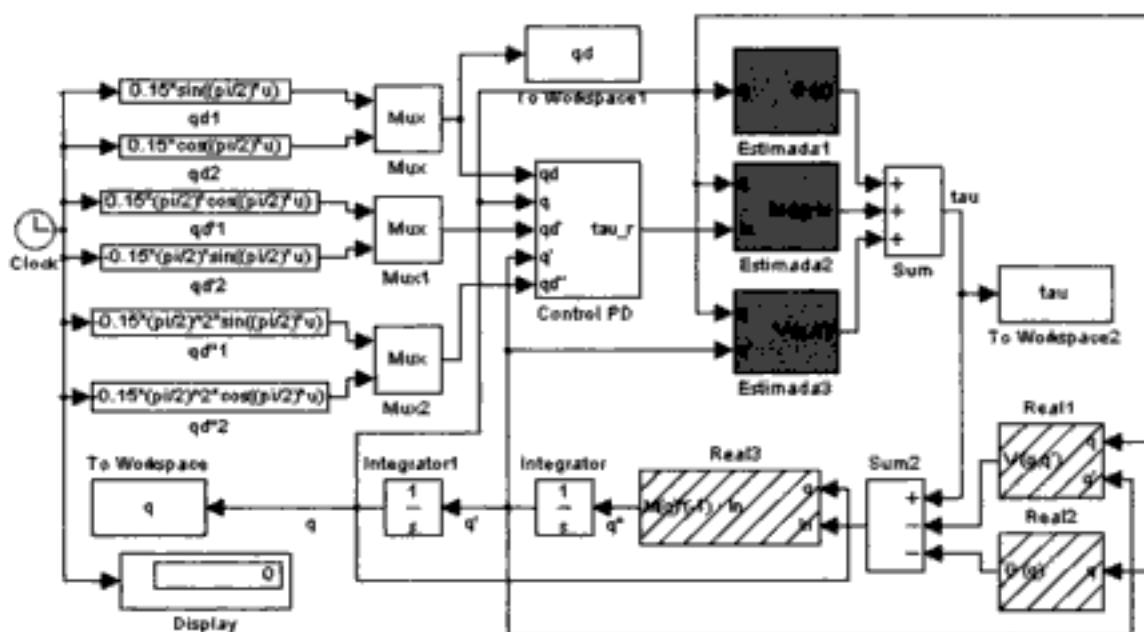


Figura 8.9: Diagrama correspondiente al control mediante la técnica del par computado usando un PD.

Las matrices  $K_p$  y  $K_v$  se introducen en el bloque “Control PD”, y se establecen condiciones iniciales nulas en los bloques integradores. El esquema anterior permite realizar simulaciones para obtener en el espacio de trabajo de MATLAB las variables que se deseen estudiar. En este caso, se ha realizado una simulación de 20 segundos de duración y se han guardado las variables  $q$ ,  $qd$  y  $\tau$  para su estudio posterior. A partir de ellas es posible elaborar una serie de gráficas que se discuten a continuación.

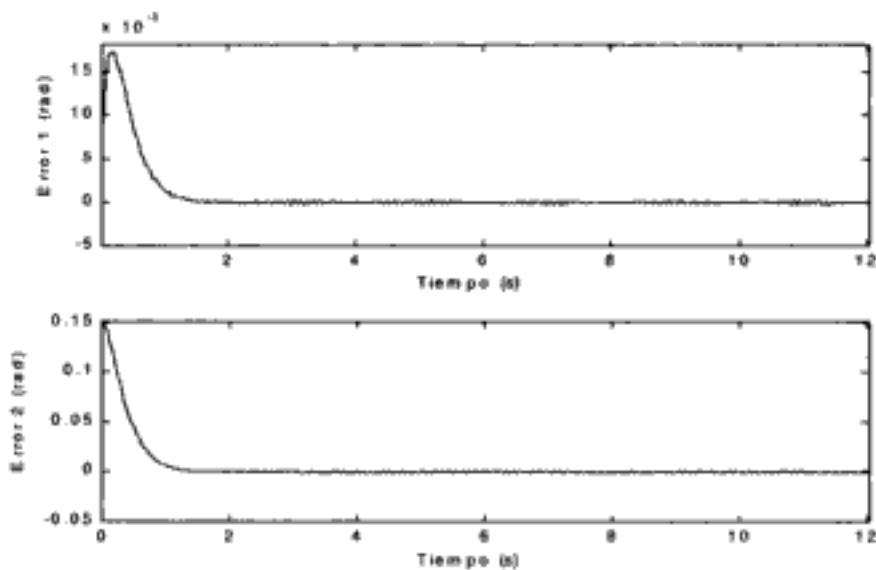
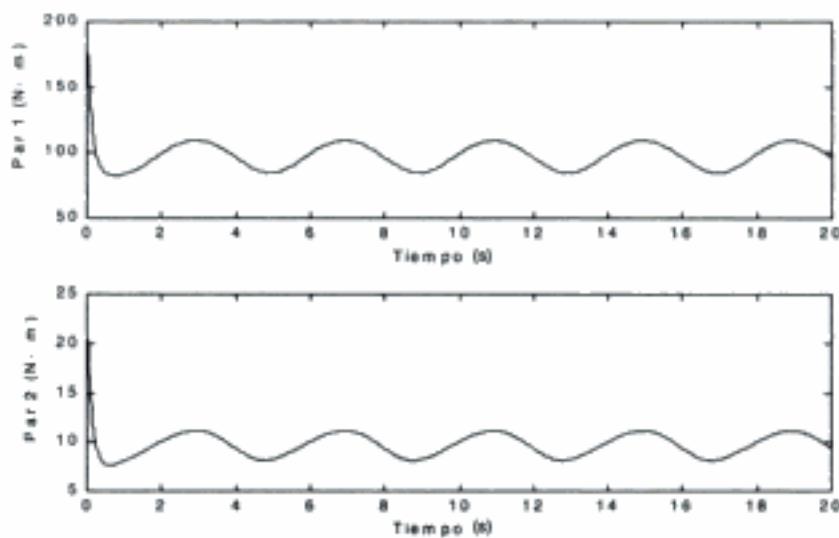
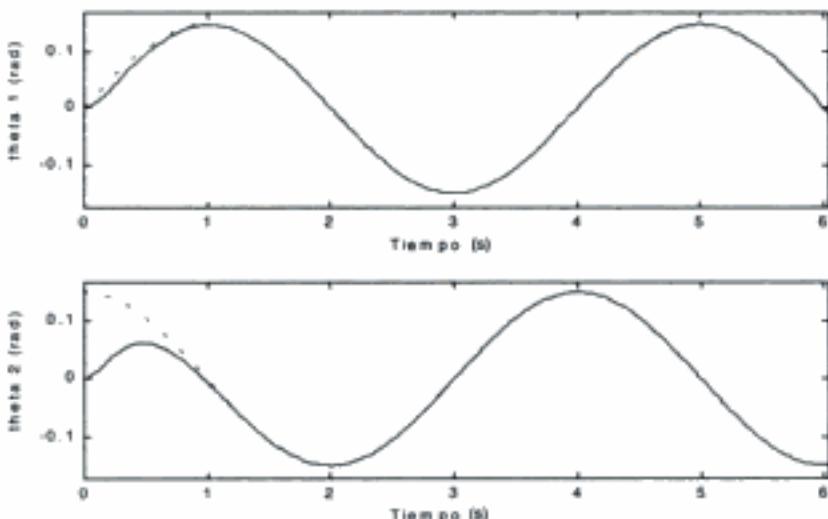


Figura 8.10: Errores en  $\theta_1$  (superior) y en  $\theta_2$  (inferior).

**Figura 8.11:** Representación de los pares aplicados  $\tau_1$  y  $\tau_2$ .

En la figura 8.10 se muestran las gráficas de los errores en las articulaciones. En ellas se observa cómo la dinámica del error es la esperada (respuesta críticamente amortiguada). No obstante, en el error  $e_1$  se aprecian ciertas oscilaciones de pequeña amplitud que no se deben a la naturaleza del sistema, sino a problemas numéricos en el método de integración empleado.

En la figura 8.11 aparecen los pares de control  $\tau_1$  y  $\tau_2$  aplicados a las articulaciones. Se observa que el par aplicado en la articulación 1 es mayor que el de la articulación 2, lo cual es lógico, ya que la articulación 1 tiene que mover dos enlaces, mientras que la articulación 2 sólo tiene que mover uno.

**Figura 8.12:** Evolución de las variables articulares obtenidas (línea continua) junto con las deseadas (seno y coseno).

En la figura 8.12 puede observarse cómo, al cabo de poco tiempo, el manipulador logra seguir perfectamente las trayectorias deseadas.

Finalmente, se ha representado (figura 8.13) la trayectoria en el espacio cartesiano que describiría el extremo del manipulador intentando seguir la trayectoria deseada (que es periódica y se repite varias veces). En la gráfica puede verse cómo, con las condiciones iniciales nulas,  $\theta_1 = \theta_2 = 0$ , el extremo se encuentra en las coordenadas ( $x = 4$ ,  $y = 0$ ) y, desde ahí, se dirige hacia la trayectoria deseada, logrando seguirla satisfactoriamente. Este movimiento inicial hacia la trayectoria deseada es el que requiere mayores pares, como puede apreciarse en la figura 8.11.

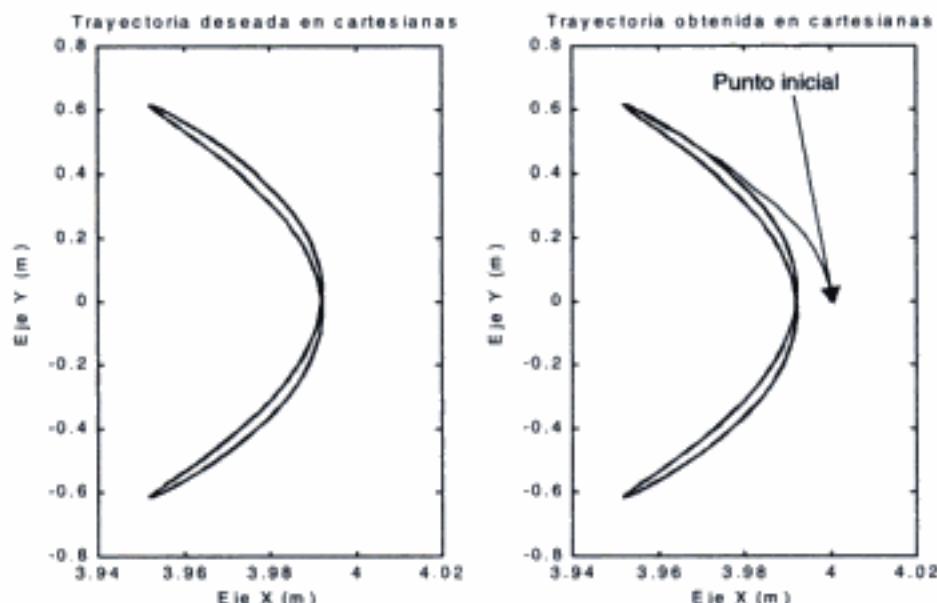


Figura 8.13: Trayectoria en coordenadas cartesianas.

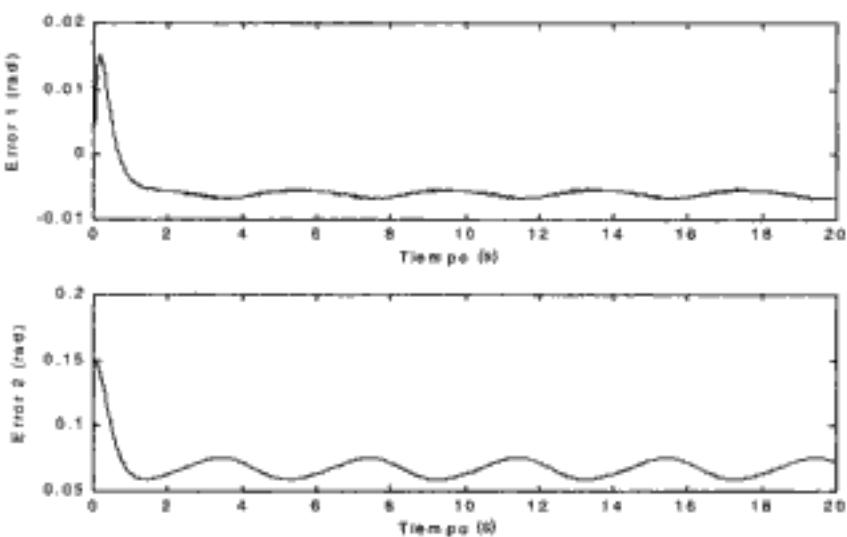
### Ejemplo 8.2

En el ejemplo 8.1 se ha supuesto un conocimiento perfecto del modelo dinámico. Supóngase ahora que se cometen errores en la estimación de las masas. Es decir, que por ejemplo,  $\hat{m}_1 = 2,1$  kg y  $\hat{m}_2 = 0,9$  kg.

Empleando las mismas ganancias del PD ( $K_p = 25$  y  $K_v = 10$ ), para simular el comportamiento del sistema sólo hay que cambiar la matriz de parámetros cinemáticos y dinámicos de los bloques grises del esquema de la figura 8.9. La nueva matriz es:

Matriz de parámetros dinámicos:	
---------------------------------	--

[0 0 0 0 0 2,1 3 0 0 0 0 0 0 0 0 1 0 0 0;  
0 3 0 0 0 0,9 1 0 0 0 0 0 0 0 0 0 1 0 0 0]

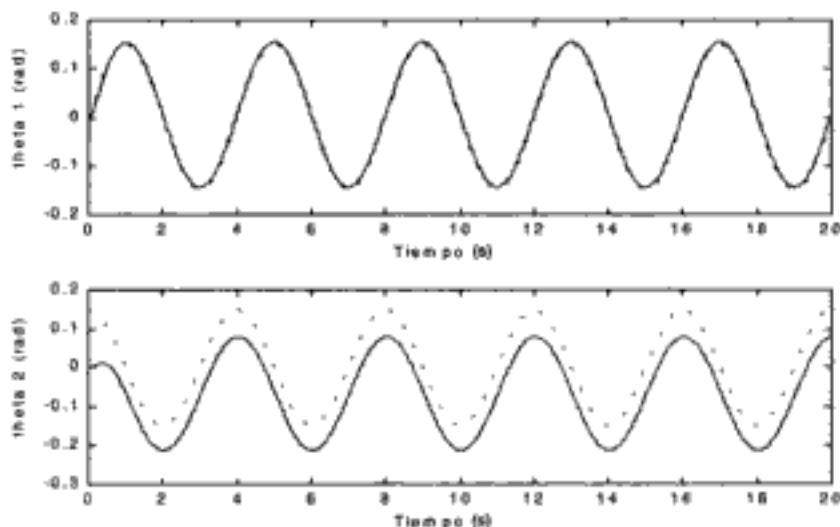


**Figura 8.14:** Errores de posición en el caso de que la estimación de las masas no sea exacta cuando se emplea un PD.

Las gráficas correspondientes al error que se obtiene en este caso aparecen en la figura 8.14. En ellas se aprecia cómo el comportamiento ha empeorado mucho respecto al ejemplo 8.1, apareciendo un importante error en régimen permanente.

Los errores de posición impiden seguir la trayectoria adecuadamente, como se aprecia en la figura 8.15 (trayectorias de articulaciones) y en la 8.16 (trayectoria cartesiana del extremo).

En un ejemplo posterior podrá verse cómo desaparece este problema si, en la técnica del par computado, se emplea un PID en lugar de un PD.



**Figura 8.15:** La trayectoria seguida por el manipulador no alcanza a la deseada debido al error en régimen permanente no nulo.

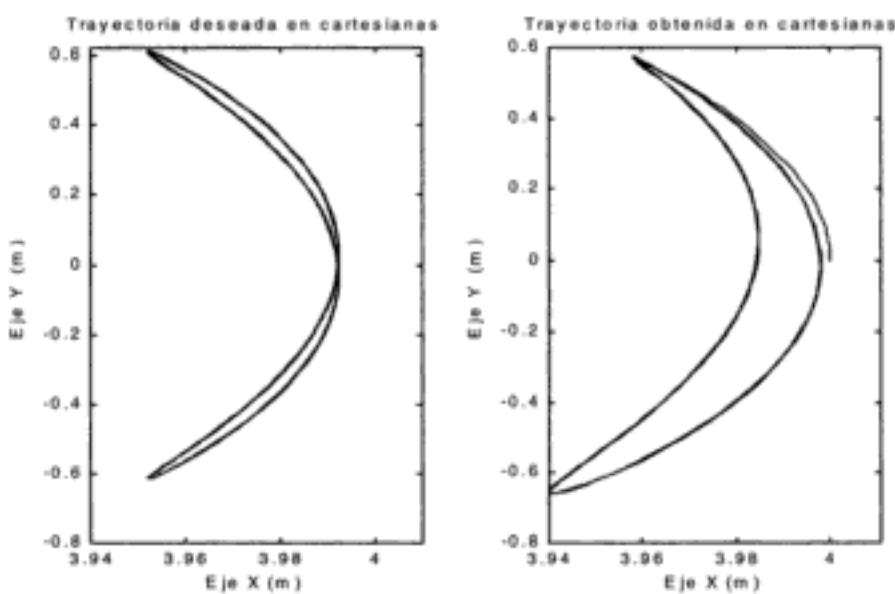


Figura 8.16: Comparación en cartesianas entre la trayectoria obtenida y la deseada.

### Ejemplo 8.3

En este ejemplo puede comprobarse cómo la aplicación del método del par computado con un PID permite resolver los problemas que se presentaban en el ejemplo anterior, cuando se aplicaba un PD que no lograba anular el error en régimen permanente.

Para realizar la simulación hay que construir el diagrama de la figura 8.17, que básicamente es el mismo que el de la figura 8.9, excepto por la sustitución del bloque “Control PD” por “Control PID”.

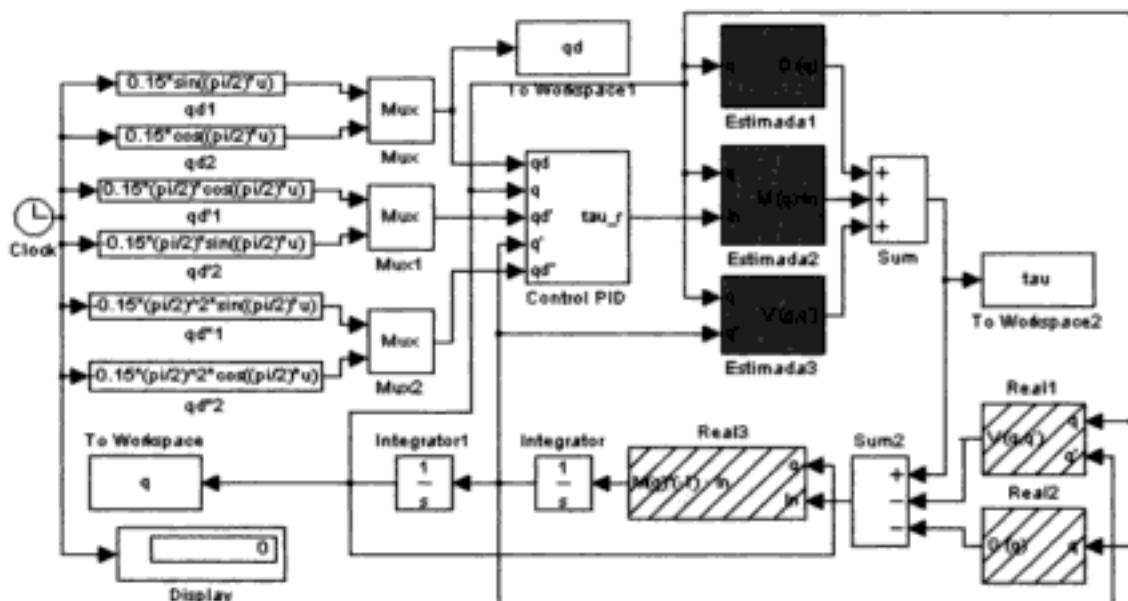
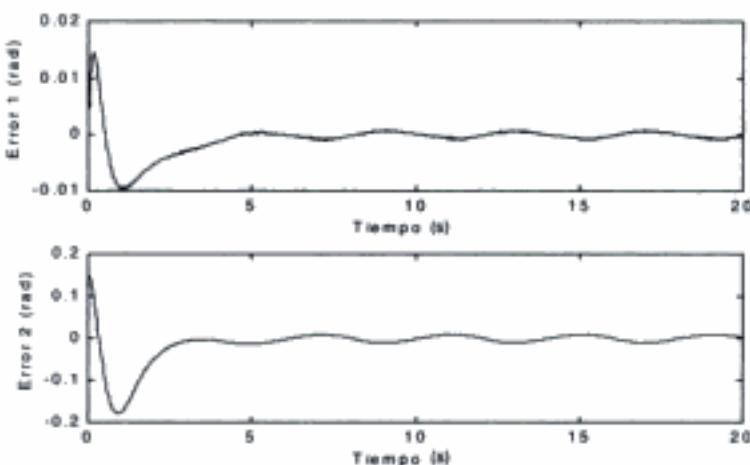
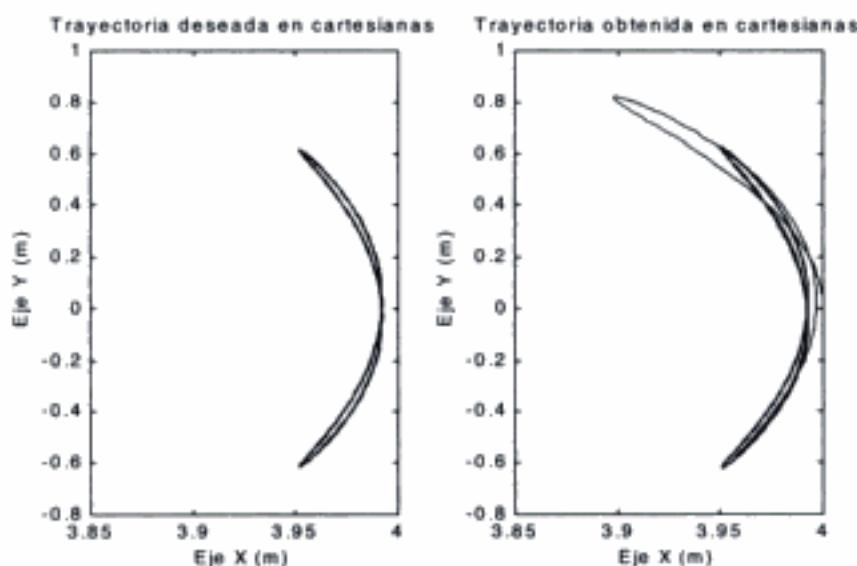


Figura 8.17: Esquema necesario para simular la técnica del par computado con un controlador PID.

**Figura 8.18:** Errores en el caso de emplear un PID.

Los valores de los parámetros elegidos para el PID son:

$$\begin{aligned} K_p &= \begin{bmatrix} k_p & 0 \\ 0 & k_p \end{bmatrix} = \begin{bmatrix} 25 & 0 \\ 0 & 25 \end{bmatrix} \\ K_v &= \begin{bmatrix} k_v & 0 \\ 0 & k_v \end{bmatrix} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \\ K_i &= \begin{bmatrix} k_i & 0 \\ 0 & k_i \end{bmatrix} = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \end{aligned} \quad (8.51)$$

**Figura 8.19:** La trayectoria en cartesianas con el PID es la deseada excepto en el transitorio inicial.

Es decir, se mantienen los mismos valores de  $K_p$  y  $K_v$  empleados en el controlador PD del ejemplo anterior y se elige un valor para  $K_i$  tal que se logre error nulo en régimen permanente en poco tiempo, pero sin llegar a inestabilizar el sistema. Estos valores son los que se introducen en el bloque "Control PID".

En la figura 8.18 se observa la evolución del error en las articulaciones y en la figura 8.19 la evolución en coordenadas cartesianas de extremo (pinza).

La trayectoria de la pinza es más próxima a la deseada que la que se logra con el PD, si bien el transitorio inicial es peor, ya que con el PID se tarda más tiempo en alcanzar el régimen permanente, como puede comprobarse comparando las gráficas de los errores con las que se obtuvieron para el PD.

#### Ejemplo 8.4

En los ejemplos anteriores se mostraba el efecto de cometer un pequeño error en la estimación de las masas del manipulador. Supóngase ahora que el manipulador sostiene una carga en su pinza. En estos casos, el problema se agrava mucho más ya que la diferencia entre la masa real y la estimada es mayor. Para ilustrar este efecto se analiza el comportamiento si el manipulador cogiera un objeto de 2 kg en  $t = 10$  s. Se supone que la forma del objeto es tal que no varían las coordenadas del centro de masas del enlace 2.

En este caso, se tendrá  $m_1 = 2$  kg,  $\hat{m}_1 = 2,1$  kg,  $m_2 = 3$  kg y  $\hat{m}_2 = 0,9$  kg a partir de  $t = 10$  s.

Para simular el comportamiento del sistema se emplea el esquema de la figura 8.9 con un controlador PD. Se ejecutan dos simulaciones:

- La primera de 0 a 10 segundos partiendo de condiciones iniciales nulas en los bloques integradores (posiciones y velocidades articulares nulas inicialmente).

La matriz de parámetros para los bloques grises es:



$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 2,1 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 9 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

mientras que para los bloques rayados es:



$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Los resultados pueden almacenarse en el espacio de trabajo bajo las etiquetas q1, qd1, qprima1 y tau1. Una vez terminada la simulación, se renombra el vector tiempo (tout) resultante (haciendo, por ejemplo, t1 = tout).

- La segunda simulación se realiza de 10 a 20 segundos, siendo necesario cargar los integradores con los últimos valores obtenidos en la primera simulación. Así, se carga el bloque “Integrator” con los valores iniciales  $[-0,2352; -0,0089]$ , que son los valores presentes en la última fila de la matriz  $q_{\text{prima}1}$  (que corresponden al instante  $t = 10 \text{ s}$ ). Análogamente, se carga el bloque “Integrator 1” con un vector columna que contenga los valores de la última fila de  $q_1$ .

También es necesario modificar, con respecto a la primera simulación, la matriz de parámetros cinemáticos y dinámicos de los bloques rayados, que ahora tendrá que ser:

Matriz de parámetros dinámicos:

$[0 \ 0 \ 0 \ 0 \ 0 \ 2 \ 3 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0;$   
 $0 \ 3 \ 0 \ 0 \ 0 \ 3 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]$

Los resultados se almacenan en el espacio de trabajo con las etiquetas  $q_2$ ,  $qd_2$ ,  $q_{\text{prima}2}$  y  $\tau_2$ . Se renombra ahora el vector  $tout$  como  $t2$ .

Finalmente, desde MATLAB se fusionan los resultados de ambas simulaciones con las siguientes órdenes:

```
q = [q1; q2];
qd = [qd1; qd2];
tau = [tau1; tau2];
t = [t1; t2];
```

Con esto ya se tienen las variables necesarias para efectuar las representaciones gráficas que se presentan a continuación.

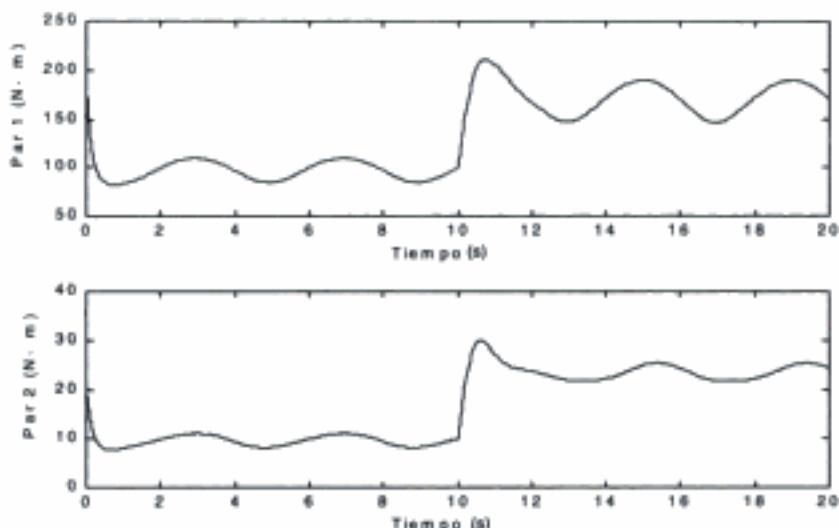
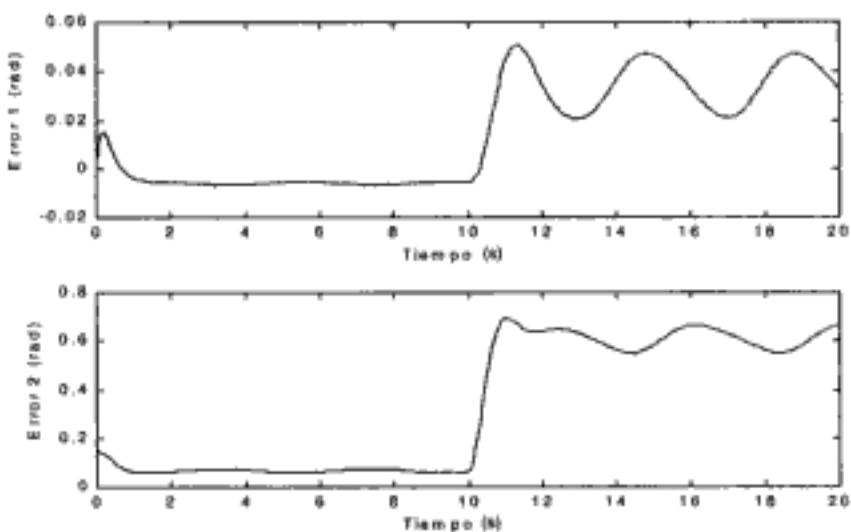


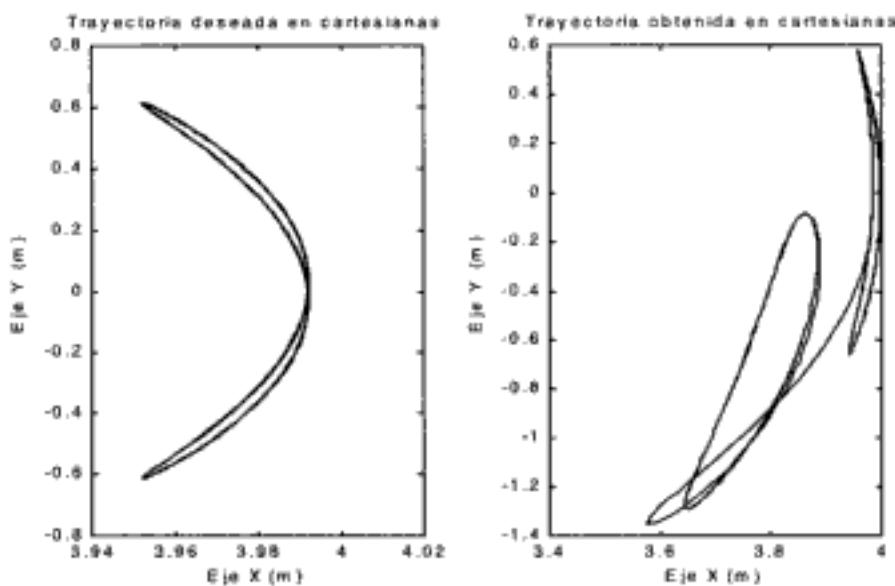
Figura 8.20: Pares aplicados en el caso de coger un objeto de 2 kg.



**Figura 8.21:** Error al agarrar un objeto de 2 kg en  $t = 10$  s.

En la figura 8.20 puede verse que, como era de esperar, cuando el robot coge el objeto aumentan los pares aplicados en las articulaciones. No obstante, como a partir de  $t = 10$  s las estimaciones de las masas se hacen mucho peores, los errores en régimen permanente aumentan de un modo apreciable (figura 8.21).

Como resultado de todo esto, el seguimiento de la trayectoria se ve notablemente afectado, no siendo capaz de seguir la trayectoria deseada como se pone de manifiesto en la figura 8.22.



**Figura 8.22:** El manipulador no es capaz de seguir la trayectoria.

**Ejemplo 8.5**

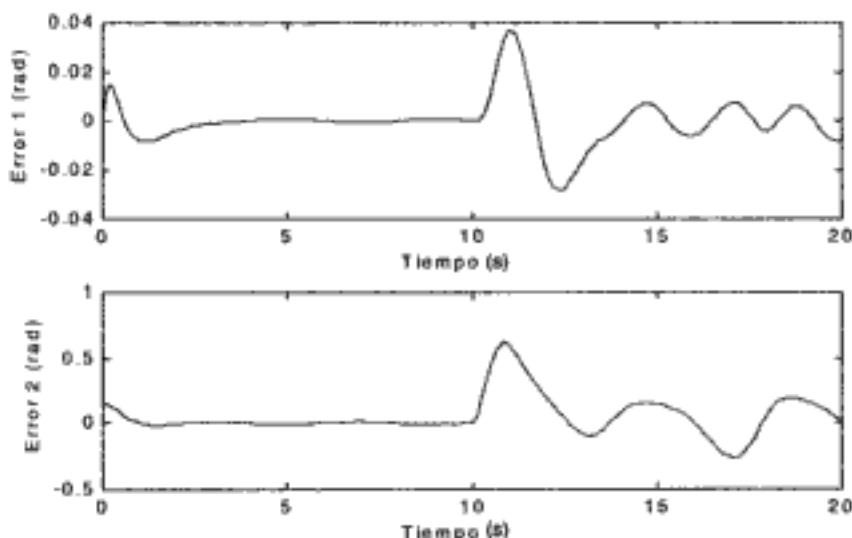
Se trata ahora de mejorar el seguimiento del ejemplo anterior eliminando los errores en régimen permanente. Para ello se emplea una ley de control de par computado con un PID de valores:

$$\begin{aligned} K_p &= \begin{bmatrix} k_p & 0 \\ 0 & k_p \end{bmatrix} = \begin{bmatrix} 25 & 0 \\ 0 & 25 \end{bmatrix} \\ K_v &= \begin{bmatrix} k_v & 0 \\ 0 & k_v \end{bmatrix} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \\ K_i &= \begin{bmatrix} k_i & 0 \\ 0 & k_i \end{bmatrix} = \begin{bmatrix} 25 & 0 \\ 0 & 25 \end{bmatrix} \end{aligned} \quad (8.52)$$

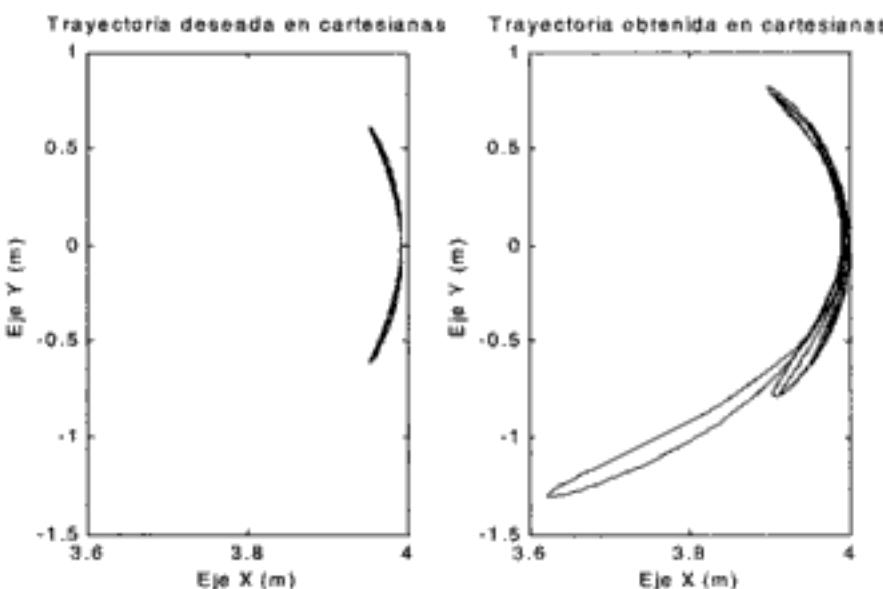
En este caso, la evolución de los errores en las articulaciones es la que se muestra en la figura 8.23.

Se observa un buen comportamiento para los primeros diez segundos, pero en el momento en que el manipulador toma la carga, la dinámica empeora ostensiblemente (se puede comprobar que el error continuaría oscilando casi indefinidamente sin llegar a hacerse cero). Esto se debe a que se emplean los mismos parámetros para el PID todo el tiempo, sin adaptarlos en función de la carga que lleve el robot en cada momento. El resultado es que el movimiento de la pinza describe una trayectoria inaceptable, tal como la que se muestra en la figura 8.24.

Los resultados obtenidos ponen de manifiesto la necesidad de adaptar los parámetros del controlador en función de la región de trabajo en que se encuentre el manipulador. Esta es la idea básica del control adaptativo que se trata en un apartado posterior.



**Figura 8.23:** Errores al tomar un objeto cuando empleamos un PID.



**Figura 8.24:** Recorrido de la pinza al tomar una carga empleando un PID.

Una posible solución es disponer de una tabla en la que se asignan los parámetros más adecuados en función de la masa del objeto que se esté agarrando en cada instante. A esta técnica se la conoce como *asignación de ganancias* (en inglés “gain scheduling”). Cuando se escogen las ganancias, hay que tener también en cuenta el diferente comportamiento dinámico del manipulador dependiendo de la región en que se encuentre. Estas consideraciones se discuten en el ejemplo siguiente.

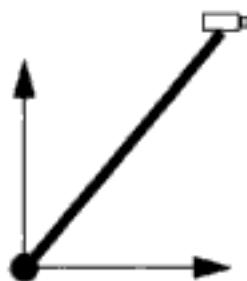
### Ejemplo 8.6

Una cámara de televisión se posiciona en el plano vertical mediante un brazo de longitud  $L$  con una sola articulación de rotación. La masa puede suponerse concentrada en el extremo, estimándose su valor en  $\hat{m}$  kg. Se dispone de estimaciones precisas de los coeficientes de rozamiento. El brazo se controla mediante la técnica del par computado. Si, debido a un cambio de cámara, la masa real durante el movimiento es  $m$  kg, estudiar las variaciones que se producirían en el comportamiento dinámico sobre el previsto, analizando cómo influye la variación de masa  $\mu = m - \hat{m}$ .

Se tratará también de obtener el error en régimen permanente y estudiar cómo varía en el espacio de trabajo en función de  $\mu$  y de los parámetros del controlador. Suponiendo que  $\hat{m} = 1$  kg,  $m = 1,1$  kg y  $l = 1$  m, se pretende determinar qué valores han de tener los parámetros del controlador para que, en la zona del espacio de trabajo en que el error en régimen permanente es máximo, el valor de dicho error en el espacio cartesiano sea inferior a 10 cm.

El manipulador en cuestión es el que se muestra en la figura 8.25. El modelo del manipulador es el siguiente:

$$\tau = ml^2\theta'' + mgl \cos \theta + v\theta' + c \operatorname{sgn}(\theta') \quad (8.53)$$



**Figura 8.25:** Cámara posicionada mediante brazo con una sola articulación.

La estrategia de control es:

$$\tau = \hat{m}l^2(\theta_d'' + K_v e' + K_p e) + \hat{m}lg \cos \theta + \hat{v}\theta' + \hat{c} \operatorname{sgn}(\theta') \quad (8.54)$$

Suponiendo:

$$l = \hat{l}; \quad v = \hat{v}; \quad c = \hat{c} \quad (8.55)$$

y sustituyendo la ley de control en el modelo, se obtiene:

$$\hat{m}l^2(\theta_d'' + K_v e' + K_p e) + \hat{m}lg \cos \theta = ml^2\theta'' + mlg \cos \theta \quad (8.56)$$

Si se hace  $\theta_d'' = e'' + \theta''$  se obtiene:

$$e'' + K_v e' + K_p e = \frac{1}{\hat{m}l^2}[ml^2\theta'' + mlg \cos \theta - \hat{m}lg \cos \theta] - \theta'' \quad (8.57)$$

Sea el error de estimación de masa:

$$\mu = m - \hat{m} \quad (8.58)$$

Sustituyendo:

$$e'' + K_v e' + K_p e = \frac{1}{\hat{m}l^2}[(\hat{m} + \mu)l^2\theta'' + (\hat{m} + \mu)lg \cos \theta - \hat{m}lg \cos \theta] - \theta'' \quad (8.59)$$

Simplificando:

$$e'' + K_v e' + K_p e = \frac{\mu}{\hat{m}}\theta'' + \frac{\mu}{\hat{m}l}g \cos \theta \quad (8.60)$$

El efecto es menor cuanto menor sea la relación  $\mu/\hat{m}$ . Asimismo, el efecto es menor cuanto menor sea  $\theta''$  y en las zonas del espacio de trabajo en las que  $\theta = n\frac{\pi}{2}$  siendo  $n = 1, 2, \dots$

Supuesto que el sistema es estable en régimen permanente, se verificará:

$$e'' = e' = \theta'' = 0 \quad (8.61)$$

Por tanto:

$$K_p e = \frac{\mu}{\hat{m}l} g \cos \theta \quad (8.62)$$

de donde

$$e = \frac{\mu g}{K_p \hat{m}l} \cos \theta \quad (8.63)$$

El error en régimen permanente es máximo cuando  $\theta = n\pi$  siendo  $n = 0, 1, 2, \dots$ , en cuyo caso:

$$\bar{e} = \frac{\mu g}{K_p \hat{m}l} \quad (8.64)$$

Se trata ahora de estudiar cómo varía el error en función de  $\mu$  y de la zona del espacio de trabajo. La variable  $e$  es el error en la variable articular, es decir,  $\Delta\theta = e$ . Para pasar a variables cartesianas, basta observar la figura 8.26.

De la figura se deduce:

$$l^2 = \left(\frac{\Delta s}{2}\right)^2 + l^2 \left[\cos\left(\frac{\Delta\theta}{2}\right)\right]^2 \quad (8.65)$$

Sustituyendo la identidad trigonométrica

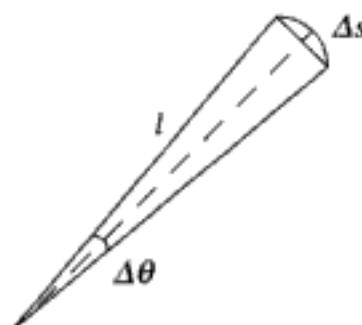
$$\left[\cos\left(\frac{\Delta\theta}{2}\right)\right]^2 = \frac{1 + \cos \Delta\theta}{2} \quad (8.66)$$

se llega a:

$$l^2 = \frac{(\Delta s)^2}{4} + \frac{l^2}{2} + \frac{l^2}{2} \cos \Delta\theta, \quad (8.67)$$

de donde es posible despejar  $\Delta s$ :

$$\Delta s = \sqrt{2}l\sqrt{1 - \cos \Delta\theta} \quad (8.68)$$



**Figura 8.26:** Obtención del error en coordenadas cartesianas.

Sustituyendo ahora la expresión obtenida para  $\Delta\theta = e$  se obtiene:

$$\Delta s = \sqrt{2}l \sqrt{1 - \cos \left( \frac{\mu g}{K_p \hat{m} l} \cos \theta \right)} \quad (8.69)$$

El error máximo se da para  $\theta = n\pi$ ,  $n = 0, 1, 2, \dots$ , en cuyo caso:

$$\overline{\Delta s} = \sqrt{2}l \sqrt{1 - \cos \left( \frac{\mu g}{K_p \hat{m} l} \right)} \quad (8.70)$$

Desarrollando:

$$\frac{\overline{\Delta s}^2}{2l^2} = 1 - \cos \left( \frac{\mu g}{K_p \hat{m} l} \right) \quad (8.71)$$

de donde:

$$\frac{\mu g}{K_p \hat{m} l} = \arccos \left( 1 - \frac{\overline{\Delta s}^2}{2l^2} \right) \quad (8.72)$$

Por tanto,

$$K_p = \frac{\mu g}{\hat{m} l \arccos \left( 1 - \frac{\overline{\Delta s}^2}{2l^2} \right)} \quad (8.73)$$

Si se desea  $\overline{\Delta s} < 0,1$ , se tendrá:

$$K_p > \frac{\mu g}{\hat{m} l \arccos \left( 1 - \frac{0,005}{l^2} \right)} = \frac{0,98}{\arccos (1 - 0,005)} = 9,796 \quad (8.74)$$

## 8.4 CONTROL ADAPTATIVO DE ROBOTS

### 8.4.1 Introducción

Existen numerosos esquemas de control adaptativo. El más simple es el de asignación de ganancias ("gain scheduling" en terminología inglesa) dependiendo, del punto de trabajo. Es decir, se divide el espacio de trabajo en regiones, calculándose durante el diseño las ganancias más apropiadas para cada una de ellas. La implantación más convencional es la de una tabla en la cual las entradas son intervalos discretizados de las variables, que determinan la pertenencia a una región u otra del espacio de trabajo, y las salidas los valores de las ganancias en estas regiones. En tiempo real basta buscar en esta tabla para determinar en cada instante los valores de las ganancias.

Las variables más empleadas para ajustar las ganancias en control de robots manipuladores son las variables articulares y la carga. Nótese que el procedimiento para ajustar los valores de los parámetros de los controladores PID de las articulaciones puede ser muy laborioso ya que, en general, se necesitaría ajustar  $3n$  parámetros, siendo  $n$  el número de articulaciones. Obviamente, el número de ensayos que es necesario realizar depende de la resolución que se desee. En cualquier caso, si no se dispone de un modelo, es necesario realizar una gran cantidad de pruebas.

Una posible solución más general es ajustar las ganancias de acuerdo con una ley de adaptación. La más empleada en robótica es la del modelo de referencia. Se trata de que las salidas del proceso, que en este caso son las variables articulares, reproduzcan las señales generadas por un modelo de referencia. El esquema de control correspondiente se muestra en la figura 8.27.

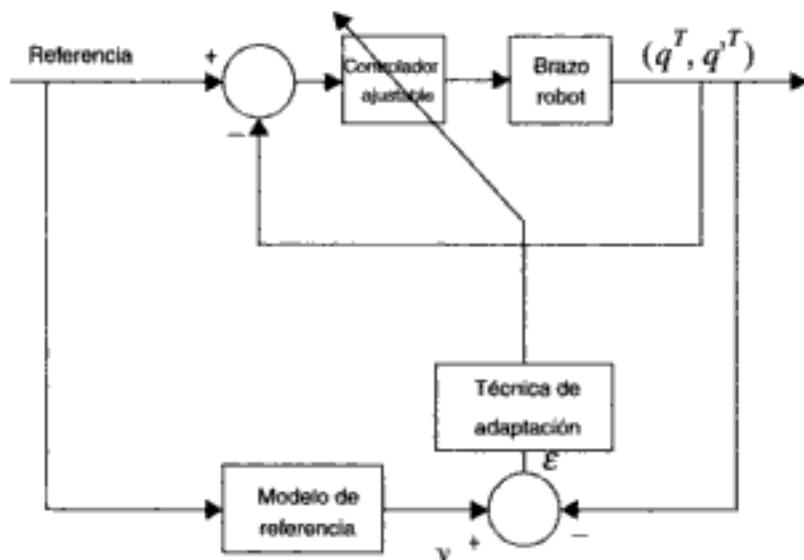


Figura 8.27: Control adaptativo por modelo de referencia.

Las salidas del modelo de referencia se comparan con las medidas de las variables articulares, obteniendo un error que se emplea, según una técnica de adaptación, para ajustar las ganancias del sistema de control que en los apartados anteriores se habían considerado como constantes.

Considérese el siguiente modelo de referencia de la articulación  $i$ :

$$a_i y_i'' + b_i y_i' + y_i = v_i \quad (8.75)$$

siendo  $y_i(t)$  la trayectoria deseada.

Los coeficientes que hacen que la salida tenga una frecuencia natural  $\omega_{ni}$  y un coeficiente de amortiguamiento  $\delta_i$ , son:

$$a_i = \frac{1}{\omega_{ni}^2}; \quad b_i = \frac{2\delta_i}{\omega_{ni}} \quad (8.76)$$

Supóngase ahora que el comportamiento en bucle cerrado de cada articulación está desacoplado del resto de las articulaciones y viene dado por:

$$\alpha_i q_i'' + \beta_i q_i' + q_i = r_i \quad (8.77)$$

variando lentamente  $\alpha_i, \beta_i$ . Las técnicas para ajustar las ganancias de realimentación y obtener  $\alpha_i, \beta_i$  se basan frecuentemente en métodos de gradiente que minimizan el error  $\varepsilon_i$ , o una función del error, entre las salidas del modelo  $y_i$  y las variables  $q_i$  medidas sobre el brazo.

$$J_i(\varepsilon_i) = \frac{1}{2}(w_2^i \varepsilon_i'' + w_1^i \varepsilon_i' + w_0^i \varepsilon_i)^2 \quad i = 1 \dots n \quad (8.78)$$

siendo  $w_0^i, w_1^i, w_2^i$  ponderaciones de los errores y sus derivadas.

#### 8.4.2 Par computado adaptativo

Otro esquema de control adaptativo es el que resulta de considerar la variación de los coeficientes del modelo dinámico utilizado en la ley del par computado, tal como se muestra en la figura 8.28. Es decir, se trata de estimar la matriz de masas  $M(q)$  y la matriz  $N = V + G + F$  que resulta de sumar los pares distintos de inercia (centrífugos y de Coriolis, gravitatorios y rozamientos). De forma más precisa, se trata de determinar los coeficientes de las matrices  $M$  y  $N$  que hacen que se minimicen los errores de seguimiento, tal como se ilustra en la figura 8.28.

Además de la complejidad de estimar todos o algunos de los parámetros de las matrices  $M$  y  $N$ , la dificultad principal de estos esquemas es que resulta difícil asegurar la convergencia del algoritmo de estimación de los parámetros y la estabilidad del sistema de control.

El método más utilizado para estudiar la estabilidad de los sistemas de control adaptativo de robots manipuladores es el de Liapunov. Un sistema  $x' = f(x), x(0) = x_0$  es

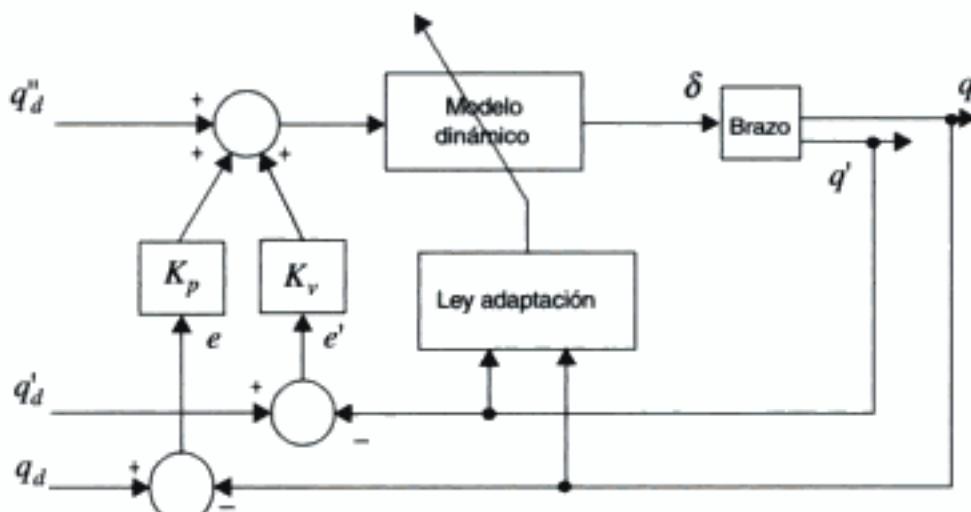


Figura 8.28: Control adaptativo por modelo de referencia en esquema de par computado.

estable, en el sentido de Liapunov, si las trayectorias del sistema se pueden mantener arbitrariamente cerca del origen si se empieza suficientemente cerca del él. En la mayor parte de las aplicaciones, se requiere la estabilidad asintótica, según la cual el sistema debe converger hacia el origen de su espacio de estados  $\forall x_0 \in \mathbb{R}^n, x(t) \rightarrow 0$  cuando  $t \rightarrow \infty$ . En particular, se trata de asegurar que el error del bucle de control tiende asintóticamente a cero. Se dice que un equilibrio es globalmente asintóticamente estable cuando existe una función  $\mathcal{V}(x)$ , denominada función de Liapunov, tal que se cumple:

i)  $\mathcal{V}(0) = 0$ ,

ii)  $\mathcal{V}$  es continuamente diferenciable y cumple las condiciones:

$$\forall x \in \mathbb{R}^n - \{0\}, \mathcal{V}(x) > 0$$

$$\exists r > 0, \inf_{\|x\| \geq r} \mathcal{V}(x) > 0$$

$$\mathcal{V}(x) \rightarrow \infty \text{ cuando } \|x\| \rightarrow \infty.$$

iii)  $\dot{\mathcal{V}}(x) = \frac{d\mathcal{V}(x)}{dt} = (\nabla \mathcal{V}(x))^T f(x) < 0, \forall x \in \mathbb{R}^n - \{0\}$ , siendo  $\nabla$  el vector gradiente

$$\nabla = \frac{\partial \mathcal{V}(x)}{\partial x}.$$

Sea  $\varphi$  un vector con parámetros del modelo, tal como, por ejemplo, las masas de las diferentes articulaciones. Supóngase que el modelo dinámico puede factorizarse escribiéndose en la forma (Lewis y otros, 1993):

$$W(q, q', q'')\varphi = M(q)q'' + V_m(q, q')q' + G(q) + F(q') \quad (8.79)$$

expresión en la cual el término de fuerzas centrífugas y de Coriolis se ha escrito de forma factorizada (ver capítulo 5). Si  $r$  es la dimensión del vector de parámetros  $\varphi$ ,  $W$  es una matriz  $n \times n$  cuyos términos son funciones temporales.

Análogamente, si  $\hat{\varphi}$  es un vector con las estimaciones de los parámetros, se tendrá:

$$W(q, q', q'')\hat{\varphi} = \hat{M}(q)q'' + \hat{V}_m(q, q')q' + \hat{G}(q) + \hat{F}(q') \quad (8.80)$$

Por tanto, la expresión del par de error que resulta de restar (8.79) y (8.80) puede escribirse en la forma

$$\tau_d = W\tilde{\varphi} = (M - \hat{M})q'' + (V - \hat{V}) + (G - \hat{G}) + (F - \hat{F}) \quad (8.81)$$

Por otra parte, aplicando la ley de control del par computado definida mediante (8.27) y (8.28) se tendrá:

$$\tau = \hat{M}\tau_r + \tau_c = \hat{M}(q_d'' + K_v e' + K_p e) + \hat{V} + \hat{G} + \hat{F}$$

y el comportamiento del error se modela mediante (8.33), donde el término de la derecha es cero en caso de modelo perfecto, o bien:

$$e'' + K_v e' + K_p e = \hat{M}^{-1} \tau_d \quad (8.82)$$

siendo  $\tau_d$  un par de error y  $\hat{M}^{-1}$  la inversa de la matriz de masas estimada.

Por tanto, la ecuación de la dinámica del error es:

$$\dot{e}'' + K_v e' + K_p e = \hat{M}^{-1} W \tilde{\varphi} \quad (8.83)$$

Nótese como la ecuación de error es no lineal y, por tanto, no pueden aplicarse los métodos convencionales de la teoría de control de sistemas lineales para elegir los valores de  $K_v$  y  $K_p$ .

Considérese el sistema controlado mediante el par computado. La ecuación de la dinámica del error (8.83) puede escribirse en la forma:

$$\dot{\varepsilon}' = A \varepsilon + B \hat{M}^{-1} W \tilde{\varphi} \quad (8.84)$$

siendo  $\varepsilon$  el vector de error de seguimiento:

$$\varepsilon = \begin{bmatrix} e \\ e' \end{bmatrix}; \quad B = \begin{bmatrix} O_n \\ I_n \end{bmatrix}; \quad A = \begin{bmatrix} O_n & I_n \\ -K_p & -K_v \end{bmatrix} \quad (8.85)$$

donde  $I_n$  es la matriz unidad  $n \times n$  y  $O_n$  la matriz nula de  $n \times n$ .

Empleando la teoría de estabilidad de Liapunov, puede demostrarse que el error de seguimiento disminuye asintóticamente eligiendo adecuadamente la ley de adaptación. Se escoge como función de Liapunov (Lewis y otros, 1993):

$$\mathcal{V} = \varepsilon^T P \varepsilon + \tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi} \quad (8.86)$$

siendo  $P$  una matriz  $2n \times 2n$  constante, definida positiva y simétrica, y  $\Gamma$  una matriz diagonal definida positiva de dimensiones  $r \times r$ :

$$\Gamma = \begin{bmatrix} \gamma_1 & & & \\ & \gamma_2 & & \\ & & \ddots & \\ & & & \gamma_r \end{bmatrix} \quad (8.87)$$

donde  $\gamma_i$  son constantes.

Derivando con respecto al tiempo la ecuación anterior, se obtiene:

$$\dot{\mathcal{V}} = \varepsilon^T P \varepsilon' + \varepsilon'^T P \varepsilon + \tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi}' + \tilde{\varphi}'^T \Gamma^{-1} \tilde{\varphi} \quad (8.88)$$

Teniendo en cuenta que  $\Gamma$  es una matriz diagonal, se verificará

$$(\tilde{\varphi}'^T \Gamma^{-1} \tilde{\varphi})^T = \tilde{\varphi}^T (\Gamma^{-1})^T \tilde{\varphi} = \tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi} \quad (8.89)$$

Por tanto, es posible sumar los dos últimos términos del lado derecho y llegar a:

$$\dot{\mathcal{V}} = \varepsilon^T P \varepsilon' + \varepsilon'^T P \varepsilon + 2 \tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi} \quad (8.90)$$

Sustituyendo  $\epsilon'$  de (8.84) se obtiene

$$\mathcal{V}' = \epsilon^T P (A\epsilon + B\hat{M}^{-1}W\tilde{\varphi}) + (A\epsilon + B\hat{M}^{-1}W\tilde{\varphi})^T P\epsilon + 2\tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi}$$

Se recuerda que la matriz de masas  $M$  es simétrica definida positiva. Si  $\hat{M}^{-1}$  es también simétrica, combinando términos se obtiene:

$$\begin{aligned}\mathcal{V}' &= -\epsilon^T Q\epsilon + \epsilon^T P B \hat{M}^{-1} W \tilde{\varphi} + \tilde{\varphi}^T W^T \hat{M}^{-1} B^T P \epsilon + 2\tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi} = \\ &= -\epsilon^T Q\epsilon + 2\tilde{\varphi}^T (\Gamma^{-1} \tilde{\varphi} + W^T \hat{M}^{-1} B^T P \epsilon)\end{aligned}\quad (8.91)$$

siendo  $Q$  una matriz simétrica definida positiva que satisface la ecuación de Liapunov:

$$A^T P + PA = -Q \quad (8.92)$$

Si se desea la estabilidad asintótica,  $V'$  debe ser negativa.

Si se hace:

$$\tilde{\varphi}' = -\Gamma W^T \hat{M}^{-1} B^T P \epsilon \quad (8.93)$$

el último término del lado derecho de  $V'$  se anula, con lo cual queda:

$$\mathcal{V}' = -\epsilon^T Q\epsilon \quad (8.94)$$

Teniendo en cuenta que  $\hat{\varphi} = \varphi - \tilde{\varphi}$  siendo  $\varphi$  los parámetros verdaderos, si se supone que  $\varphi$  es constante se tendrá que  $\varphi' = 0$ . Por tanto, sustituyendo en la expresión de  $\varphi'$  la ley de adaptación, queda:

$$\hat{\varphi}' = \Gamma W^T \hat{M}^{-1} B^T P \epsilon \quad (8.95)$$

Así, es posible implantar una estrategia de control tal como la que se indica en la figura 8.28. De esta forma, se trata de adaptar los coeficientes del modelo dinámico para minimizar los errores.

En Lewis y otros (1993) se demuestra que  $e(t)$  tiende a cero cuando  $t$  tiende a infinito y que si  $\hat{M}^{-1}$  existe, los errores en la estimación de los parámetros  $\varphi$  están acotados.

### Ejemplo 8.7

Considérese el ejemplo del manipulador con dos articulaciones de rotación del capítulo 5. Las variables articulares son  $q_1 = \theta_1$ ,  $q_2 = \theta_2$ .

El modelo dinámico viene dado por:

$$\tau = Mq'' + V + G + F \quad (8.96)$$

siendo,

$$M = \begin{bmatrix} l_2^2 m_2 + 2l_1 l_2 m_2 c_2 + l_1^2 (m_1 + m_2) & l_2^2 m_2 + l_1 l_2 m_2 c_2 \\ l_2^2 m_2 + l_1 l_2 m_2 c_2 & l_2^2 m_2 \end{bmatrix} \quad (8.97)$$

$$V = \begin{bmatrix} -m_2 l_1 l_2 s_2 q_2^2 - 2m_2 s_2 l_1 l_2 q'_1 q'_2 \\ m_2 l_1 l_2 s_2 q'_1^2 \end{bmatrix} \quad (8.98)$$

$$G = \begin{bmatrix} m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1 \\ m_2 l_2 g c_{12} \end{bmatrix} \quad (8.99)$$

Supóngase que se aplica la ley de control del par computado.

En este ejemplo parece evidente que pueden suponerse conocidas  $l_1$  y  $l_2$ , estimándose  $\hat{m}_1$  y  $\hat{m}_2$ .

Nótese la dependencia lineal del lado derecho del modelo dinámico con los parámetros  $m_1$  y  $m_2$ .

Por consiguiente, si  $\varphi$  es un vector con las masas reales  $\varphi = [m_1 \ m_2]^T$  y  $\hat{\varphi} = [\hat{m}_1 \ \hat{m}_2]^T$  es un vector con las estimaciones de las masas, puede aplicarse el modelo factorizado (8.79), siendo:

$$W = \begin{bmatrix} l_1^2 q_1'' + l_1 g c_1 & l_2^2 (q_1'' + q_2'') + l_1 l_2 c_2 (2q_1'' + q_2'') + l_1^2 q_1'' - l_1 l_2 s_2 q_2^2 - 2l_1 l_2 s_2 q'_1 q'_2 + l_2 g c_{12} + l_1 g c_1 \\ 0 & l_2^2 (q_1'' + q_2'') + l_1 l_2 c_2 q_1'' + l_1 l_2 s_2 q_1'^2 + l_2 g c_{12} \end{bmatrix} \quad (8.100)$$

Si  $\tilde{\varphi} = [\tilde{\varphi}_1 \ \tilde{\varphi}_2]^T$  es el vector de errores de estimación, donde  $\varphi_1 = m_1 - \hat{m}_1$  y  $\varphi_2 = m_2 - \hat{m}_2$ , la ecuación de la dinámica del error viene dada por (8.83), siendo:

$$K_v = \begin{bmatrix} k_{v_1} & 0 \\ 0 & k_{v_2} \end{bmatrix}; \quad K_p = \begin{bmatrix} k_{p_1} & 0 \\ 0 & k_{p_2} \end{bmatrix} \quad (8.101)$$

Nótese como la ecuación de error es no lineal y, por tanto, no pueden aplicarse los métodos convencionales de la teoría de control de sistemas lineales para elegir los valores de  $K_v$  y  $K_p$ . Si se hace:

$$K_v = k_v I_n; \quad K_p = k_p I_n$$

se escogería:

$$P = \frac{1}{2} \begin{bmatrix} (k_p + \frac{1}{2} k_v) I_n & \frac{1}{2} I_n \\ \frac{1}{2} I_n & I_n \end{bmatrix} = \begin{bmatrix} p_1 I_n & p_2 I_n \\ p_3 I_n & p_4 I_n \end{bmatrix} \quad (8.102)$$

que es simétrica y positiva definida si  $k_v > 1$ .

Es necesario comprobar que la matriz  $Q$  resultante en la ecuación de Liapunov es definida positiva. Sustituyendo  $P$  en esta ecuación, se obtiene:

$$Q = \begin{bmatrix} \frac{1}{2}k_p I_n & O_n \\ O_n & (k_v - \frac{1}{2})I_n \end{bmatrix} \quad (8.103)$$

Teniendo en cuenta que  $k_v > 1$ ,  $Q$  es definida positiva y simétrica.

En general, encontrar  $P$  y  $Q$  no es fácil.

Una vez que se ha determinado  $P$ , es posible obtener la ley de adaptación de los parámetros  $\dot{\phi} = [\hat{m}_1 \hat{m}_2]^T$  según (8.95) (Lewis y otros, 1993):

$$\begin{bmatrix} \dot{m}'_1 \\ \dot{m}'_2 \end{bmatrix} = \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} \begin{bmatrix} W_{11} & W_{21} \\ W_{12} & W_{22} \end{bmatrix} \begin{bmatrix} MI_{11} & MI_{21} \\ MI_{21} & MI_{22} \end{bmatrix} \begin{bmatrix} O_2 & I_2 \\ O_2 & I_2 \end{bmatrix} \begin{bmatrix} p_1 I_2 & p_2 I_2 \\ p_2 I_2 & p_3 I_2 \end{bmatrix} \begin{bmatrix} e \\ e' \end{bmatrix} \quad (8.104)$$

siendo:

$$\hat{M}^{-1} = \begin{bmatrix} MI_{11} & MI_{21} \\ MI_{21} & MI_{22} \end{bmatrix} \quad (8.105)$$

Por tanto:

$$\begin{aligned} \dot{m}'_1 &= \alpha_1 [(W_{11}MI_{11} + W_{21}MI_{21})(p_2e_1 + p_3e'_1) + (W_{11}MI_{21} + W_{21}MI_{22})(p_2e_2 + p_3e'_2)] \\ &\quad (8.106) \end{aligned}$$

$$\dot{m}'_2 = \alpha_2 [(W_{12}MI_{11} + W_{22}MI_{21})(p_2e_1 + p_3e'_1) + (W_{12}MI_{21} + W_{22}MI_{22})(p_2e_2 + p_3e'_2)]$$

siendo:

$$MI_{11} = \frac{1}{\Delta}(\hat{m}_2 l_2^2) \quad (8.107)$$

$$MI_{21} = -\frac{1}{\Delta}(\hat{m}_2 l_1 l_2 c_2 + \hat{m}_2 l_2^2) \quad (8.108)$$

$$MI_{22} = \frac{1}{\Delta}[2\hat{m}_2 l_1 l_2 c_2 + \hat{m}_2 l_2^2 + (\hat{m}_1 + \hat{m}_2)l_1^2] \quad (8.109)$$

donde:

$$\Delta = [2\hat{m}_2 l_1 l_2 c_2 + \hat{m}_2 l_2^2 + (\hat{m}_1 + \hat{m}_2)l_1^2(\hat{m}_2 l_2^2) - (\hat{m}_2 l_2^2 + \hat{m}_2 l_1 l_2 c_2)^2] \quad (8.110)$$

### 8.4.3 Controlador de inercia adaptativa

El método de control adaptativo presentado en el apartado anterior es interesante desde el punto de vista conceptual, pero presenta dos inconvenientes para su aplica-

ción práctica. Por una parte, es necesario que la estimación de la matriz de masas sea invertible, lo cual puede ocasionar problemas en aplicaciones en las que se presenta una variación de masa considerable, tal como sucede si se agarra una carga importante. Por otra, es necesario medir las aceleraciones en las variables articulares. Teniendo en cuenta que normalmente sólo se miden directamente mediante sensores las posiciones y velocidades, es necesario obtener las aceleraciones tomando sus derivadas, lo cual introduce ruidos.

Por ello, se han propuesto otros métodos de control adaptativo, entre los que se encuentran los denominados de inercia adaptativa. En Slotine y Li (1988) se propone un método de seguimiento de trayectorias en el que se utiliza la conservación de la energía.

Sea  $r(t)$  una señal auxiliar de error definida como:

$$r = \Lambda e + e' \quad (8.111)$$

siendo  $\Lambda$  una matriz diagonal definida positiva  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ .

La función de Liapunov empleada es

$$\mathcal{V} = \frac{1}{2} r^T M r + \frac{1}{2} \varphi^T \Gamma^{-1} \tilde{\varphi} \quad (8.112)$$

donde  $\Gamma$  es la matriz diagonal definida positiva (8.87) y  $\varphi$  es el vector de errores de estimación de los parámetros  $\varphi = \varphi - \hat{\varphi}$ .

Derivando con respecto al tiempo, la función  $V$ , se obtiene

$$\dot{\mathcal{V}} = r^T M r' + \frac{1}{2} r^T M' r + \tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi}' \quad (8.113)$$

Se trata ahora de escribir la ecuación de la dinámica del manipulador en términos de  $r$ . Teniendo en cuenta la definición (8.111), puede escribirse:

$$r = \Lambda e + (q'_d - q') \quad (8.114)$$

de donde

$$q' = \Lambda e + q'_d - r \quad (8.115)$$

Asimismo,

$$q'' = \Lambda e' + q''_d - r' \quad (8.116)$$

Sustituyendo (8.115) y (8.116) en la ecuación del modelo dinámico del manipulador con el término de los efectos centrífugos y de Coriolis escrito en forma factorizada  $V_m q'$  se tiene:

$$\tau = M q'' + V_m q' + G + F = M(\Lambda e' + q''_d - r') + V_m(\Lambda e + q'_d - r) + G + F \quad (8.117)$$

Por tanto puede escribirse

$$M r' = M(\Lambda e' + q''_d) + V_m(\Lambda e + q'_d) + G + F - V_m r - \tau \quad (8.118)$$

Si se hace

$$Y\dot{\varphi} = M(\Lambda e' + q''_d) + V_m(\Lambda e + q'_d) + G + F \quad (8.119)$$

la ecuación (8.118) puede escribirse como

$$Mr' = Y\dot{\varphi} - V_m r - \tau \quad (8.120)$$

En estas expresiones  $Y$  es una matriz de funciones temporales conocidas. Nótese como el lado izquierdo de (8.119) se escribe en forma factorizada, al igual que se hacía en el apartado anterior con  $WY$  en (8.79). La diferencia fundamental es que ahora en el lado derecho de (8.119) no aparece la aceleración  $q''$  en cuya obtención se introduce ruido.

Sustituyendo (8.120) en (8.113) se obtiene:

$$\mathcal{V}' = r^T(Y\dot{\varphi} - \tau) + r^T\left(\frac{1}{2}M - V_m\right)r + \tilde{\varphi}^T\Gamma^{-1}\tilde{\varphi} \quad (8.121)$$

Teniendo en cuenta la relación existente entre la derivada de la matriz de masas y  $V_m$  (ver capítulo 5), el segundo sumando del lado derecho es nulo y, por tanto, puede escribirse:

$$\mathcal{V}' = r^T(Y\dot{\varphi} - \tau) + \tilde{\varphi}^T\Gamma^{-1}\tilde{\varphi} \quad (8.122)$$

Si se aplica el par de control:

$$\tau = Y\hat{\varphi} + K_v r \quad (8.123)$$

sustituyendo en (8.122) se obtiene:

$$\mathcal{V}' = r^T(Y\dot{\varphi} - Y\hat{\varphi} - K_v r) + \tilde{\varphi}^T\Gamma^{-1}\tilde{\varphi} = -r^T K_v r + r^T Y\tilde{\varphi} - \tilde{\varphi}^T\Gamma^{-1}\tilde{\varphi} \quad (8.124)$$

Por tanto, si los parámetros se adaptan según:

$$\hat{\varphi}' = -\tilde{\varphi}' = \Gamma Y r \quad (8.125)$$

sustituyendo en (8.124) se obtiene:

$$\mathcal{V}' = -r^T K_v r + r^T Y\tilde{\varphi} - \tilde{\varphi}^T\Gamma^{-1}Yr = -r^T K_v r \quad (8.126)$$

En Lewis y otros (1993) se demuestra que los errores de estimación  $\varphi$  están acotados y que se verifica:

$$\lim_{t \rightarrow \infty} \mathcal{V}' = 0 \quad (8.127)$$

$$\lim_{t \rightarrow \infty} r = 0 \quad (8.128)$$

Obsérvese que (8.111) es una ecuación diferencial de primer orden estable excitada por la entrada  $r$ . Por tanto, aplicando la teoría de sistemas dinámicos lineales se deduce:

$$\begin{aligned}\lim_{t \rightarrow \infty} e &= 0 \\ \lim_{t \rightarrow \infty} e' &= 0\end{aligned}\quad (8.129)$$

Por tanto, los errores de seguimiento  $e$  y  $e'$  son asintóticamente estables.

### Ejemplo 8.8

Considérese el mismo sistema de control del ejemplo del apartado anterior con dos articulaciones de rotación. Suponiendo que la fricción es despreciable y que las longitudes de los enlaces son conocidas de forma exacta, si se aplica (8.123) con:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}; \quad K_v = \begin{bmatrix} K_{v1} & 0 \\ 0 & K_{v2} \end{bmatrix}; \quad r = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + \begin{bmatrix} e'_1 \\ e'_2 \end{bmatrix} \quad (8.130)$$

el controlador de inercia adaptativa puede formularse como:

$$\begin{aligned}\tau_1 &= Y_{11}\hat{m}_1 + Y_{12}\hat{m}_2 + k_{v_1}e'_1 + k_{v_1}\lambda_1e_1 \\ \tau_2 &= Y_{21}\hat{m}_1 + Y_{22}\hat{m}_2 + k_{v_2}e'_2 + k_{v_2}\lambda_2e_2\end{aligned}\quad (8.131)$$

La matriz  $Y$ ,

$$Y(q_d^n, q_d', q_d, q, q') = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}, \quad (8.132)$$

se obtiene mediante (8.119), en la cual  $F = 0$  y  $V_m$  se determina aplicando el procedimiento de factorización que se indica en el apartado 5.6, llegándose a:

$$V_m(\theta, \dot{\theta}) = \frac{1}{2}(M(\theta, \dot{\theta}) + S) = \begin{bmatrix} -m_2l_1l_2s_2q'_2 & -m_2l_1l_2s_2(q'_1 + q'_2) \\ -m_2l_1l_2s_2q'_1 & 0 \end{bmatrix} \quad (8.133)$$

Sustituyendo en (8.119) se obtiene:

$$\begin{aligned}Y_{11} &= l_1^2(q_{d_1}'' + \lambda_1e'_1) + l_1gc_1 \\ Y_{12} &= (l_2^2 + 2l_1l_2c_2 + l_1^2)(q_{d_1}'' + \lambda_1e'_1) + (l_2^2 + l_1l_2c_2)(q_{d_2}'' + \lambda_2e'_2) \\ &\quad - l_1l_2s_2(q'_1 + q'_2)(q_{d_2}'' + \lambda_2e_2) + l_2gc_{12} + l_1gc_1 \\ Y_{21} &= 0 \\ Y_{22} &= (l_1l_2c_2 + l_2^2)(q_{d_1}'' + \lambda_1e'_1) + l_2^2(q_{d_2}'' + \lambda_2e'_2) \\ &\quad + l_1l_2s_2q'_1(q_{d_1}'' + \lambda_1e_1) + l_2gc_{12}\end{aligned}\quad (8.134)$$

El vector de estimaciones de los parámetros  $\hat{\varphi} = [\hat{m}_1, \hat{m}_2]^T$  se obtiene aplicando (8.125) con

$$\Gamma = \begin{bmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{bmatrix} \quad (8.135)$$

$$Y = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}$$

y las mismas señales  $r$ , con lo cual se llega a:

$$\begin{aligned} \hat{m}'_1 &= \gamma_1 [Y_{11}(\lambda_1 e_1 + e'_1) + Y_{21}(\lambda_2 e_2 + e'_2)] \\ \hat{m}'_2 &= \gamma_2 [Y_{12}(\lambda_1 e_1 + e'_1) + Y_{22}(\lambda_2 e_2 + e'_2)] \end{aligned} \quad (8.136)$$

En la figura 8.29 se presenta el diagrama Simulink de simulación de la estrategia de control.

Sean  $m_1 = 0.8 \text{ kg}$  y  $m_2 = 2.3 \text{ kg}$ , los valores de los parámetros:  $k_{v_1} = k_{v_2} = 10$ ;  $\lambda_1 = \lambda_2 = 2.5$ ;  $\gamma_1 = \gamma_2 = 20$ .

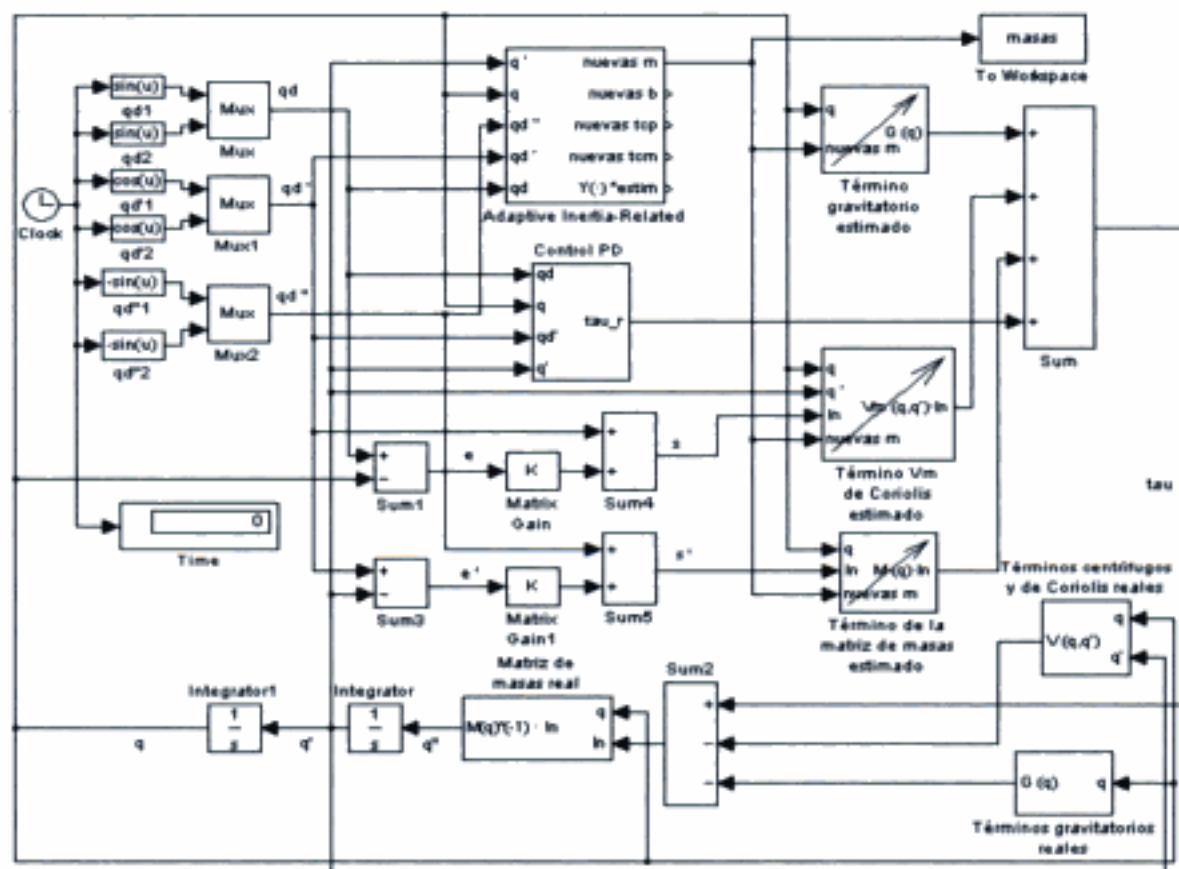
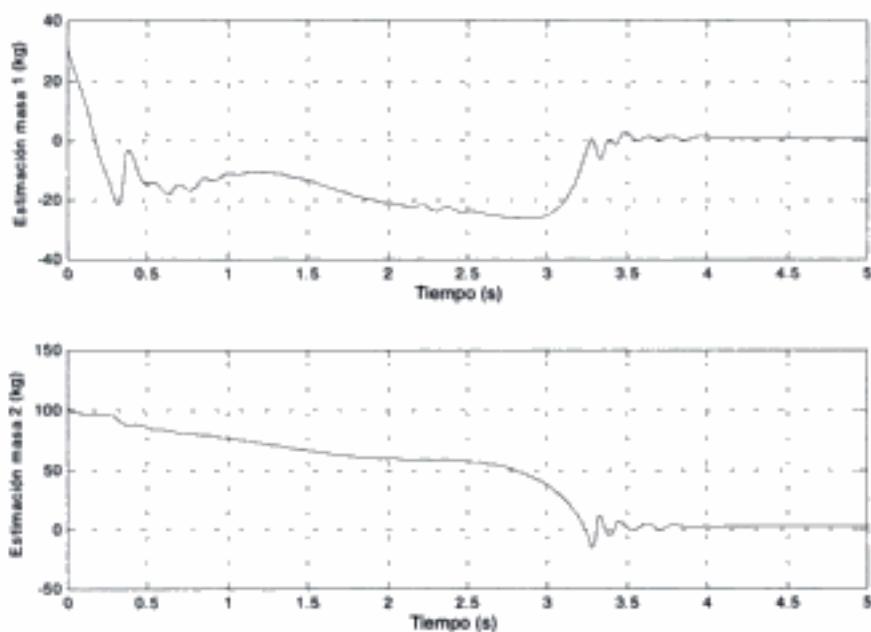


Figura 8.29: Simulación de la estrategia de controlador de inercia adaptativa.

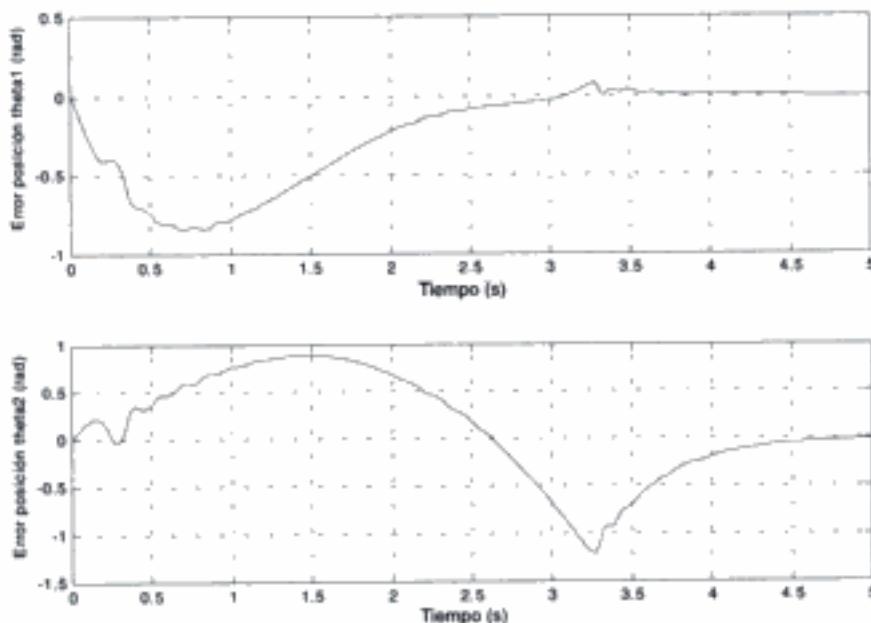


**Figura 8.30:** Estimación de las masas en el controlador de inercia adaptativa.

las condiciones iniciales:

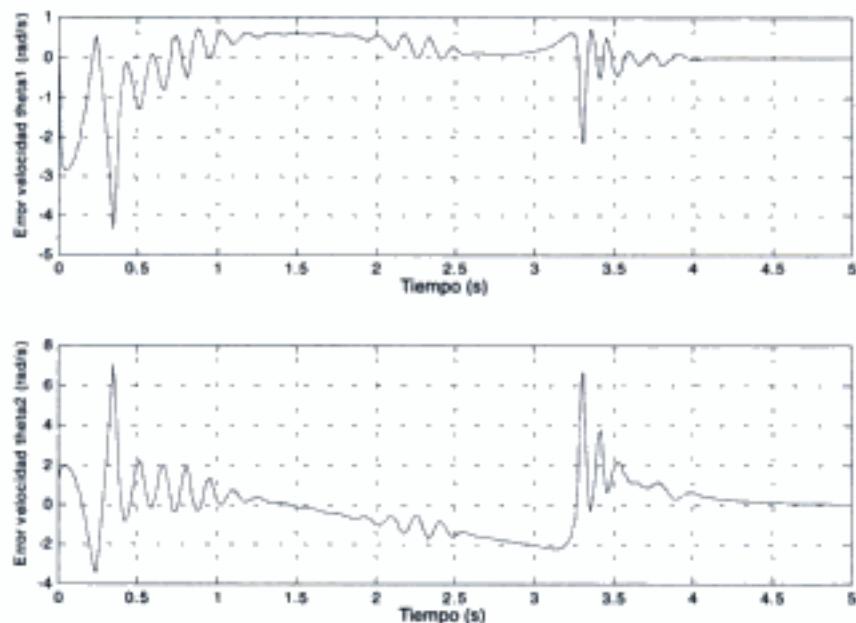
$$\begin{aligned}\hat{m}_1(0) &= 30; & \hat{m}_2(0) &= 100 \\ q_1(0) &= 0; & q_2(0) &= 0 \\ q'_1(0) &= 0; & q'_2(0) &= 0\end{aligned}\tag{8.137}$$

y la trayectoria deseada  $q_{d_1} = q_{d_2} = \sin t$ .



**Figura 8.31:** Evolución de errores de posición empleando el controlador de inercia adaptativa.

La figura 8.30 presenta la evolución temporal de las estimaciones de las masas. En la figura 8.31 se muestra la evolución de los errores de posición y en la figura 8.32 la de los errores de velocidad.



**Figura 8.32:** Evolución de errores de velocidad empleando el controlador de inercia adaptativa.

## 8.5 CONTROL CON APRENDIZAJE

La motivación básica es la misma que en el caso del control adaptativo: las dificultades de conocer de forma suficientemente precisa el modelo. Sin embargo, el problema se resuelve de forma diferente. En el control con aprendizaje se trata de mejorar el comportamiento del sistema de control en repeticiones sucesivas de las operaciones.

Para ello, suele suponerse que una parte del modelo es conocida, generándose mediante ella un par de control, al que se añade otro par calculado con un modelo que se ajusta mediante una ley de aprendizaje en repeticiones sucesivas de la misma operación.

Por tanto, si se utiliza un esquema similar al de las secciones precedentes, con términos de control PD, más términos que dependen del modelo conocido, el par de control sería:

$$\tau = \hat{M}(q)[q_d'' + K_v e' + K_p e] + \hat{V}(q, q') + \hat{G}(q) + \hat{\gamma}_k \quad (8.138)$$

siendo  $\hat{\gamma}_k$  un par obtenido por aprendizaje en repeticiones sucesivas del movimiento  $k = 0, 1, \dots$

Teniendo en cuenta el modelo dinámico (8.25) y considerando además de los términos estimados, un término de fricciones no modeladas,  $\hat{\gamma}_k$ , sustituyendo (8.138) se llega a:

$$e'' + (K_v e' + K_p e) = \hat{M}^{-1}(F - \gamma_k) \quad (8.139)$$

o en forma más compacta:

$$e'' + (K_v e' + K_p e) = D - D_k \quad (8.140)$$

siendo  $D = \hat{M}^{-1}F$  y  $D_k = \hat{M}^{-1}\hat{\gamma}_k$ .

En cada una de las articulaciones se tendrá:

$$e''_i + (K_{vi} e'_i + K_{pi} e_i) = D_i - D_{ik} \quad (8.141)$$

siendo el primer sumando del lado derecho una función del tiempo que se mantiene constante en las repeticiones y el segundo un par que se trata de mejorar en cada repetición.

En Craig (1988) se propone la siguiente ley de aprendizaje para la articulación  $i$ :

$$\hat{D}_{i(k+1)} = \hat{D}_{ik} + P * e_{ik} \quad (8.142)$$

donde el segundo sumando del lado derecho es el producto de convolución entre la respuesta impulsional  $P$  de un filtro y el error en la iteración  $k$ . Tomando transformadas de Laplace en (8.141) y en (8.142) y sustituyendo en la transformada de Laplace de la segunda se obtiene:

$$\hat{D}_{i(k+1)}(s) = \hat{D}_{ik}(s) + P(s)H(s)[D_i(s) - \hat{D}_{ik}(s)] \quad (8.143)$$

siendo:

$$H(s) = \frac{1}{s^2 + K_{vi}s + K_{pi}} \quad (8.144)$$

Sea,

$$G(s) = 1 - P(s)H(s) \quad (8.145)$$

Sustituyendo en (8.143), se obtiene:

$$\hat{D}_{i(k+1)}(s) = G(s)\hat{D}_{ik}(s) + \hat{D}_i(s)[1 - G(s)] \quad (8.146)$$

Comenzando con  $D_{io}$  después de  $k$  iteraciones se obtiene:

$$\hat{D}_{ik}(s) = D_i(s) + CG^k(s) \quad (8.147)$$

donde  $C$  es una constante. En el segundo término del lado derecho de (8.147) aparece  $G^k(s)$ . La convergencia del algoritmo de aprendizaje depende de este factor: en caso de que tienda a cero en las sucesivas iteraciones se producirá la convergencia verificándose que  $\hat{D}_{ik}(s) \rightarrow D_i(s)$ , con lo cual se verificaría la expresión (8.140) con el lado derecho igual a cero. Por tanto, el comportamiento dinámico del error sería el del sistema de segundo orden que se indica en el lado izquierdo de la mencionada expresión.

La transformada inversa de Laplace de  $G^k(s)$  es:

$$g_k(t) = L^{-1}[G^k(s)] \quad (8.148)$$

Se demuestra (Craig, 1988) que, si se escoge de forma apropiada  $P(s)$ , se cumplirá:

$$\lim_{k \rightarrow \infty} |g_k(t)| = 0 \quad (8.149)$$

con lo cual es posible garantizar la convergencia.

Un posible filtro es el:

$$P(s) = s^2 + (K_{vi} - \mu)s + (K_{pi} - \mu) \quad (8.150)$$

siendo  $\mu$  una constante. La convergencia del filtro depende de  $\mu$ . Para escogerlo, se tiene en cuenta  $|G(j\omega)|$  (Craig, 1988). El filtro es no causal, pero puede implantarse empleando todos los errores registrados en la iteración anterior del algoritmo.

En la figura 8.33 se muestra un esquema en el que se ilustra el control mediante aprendizaje.

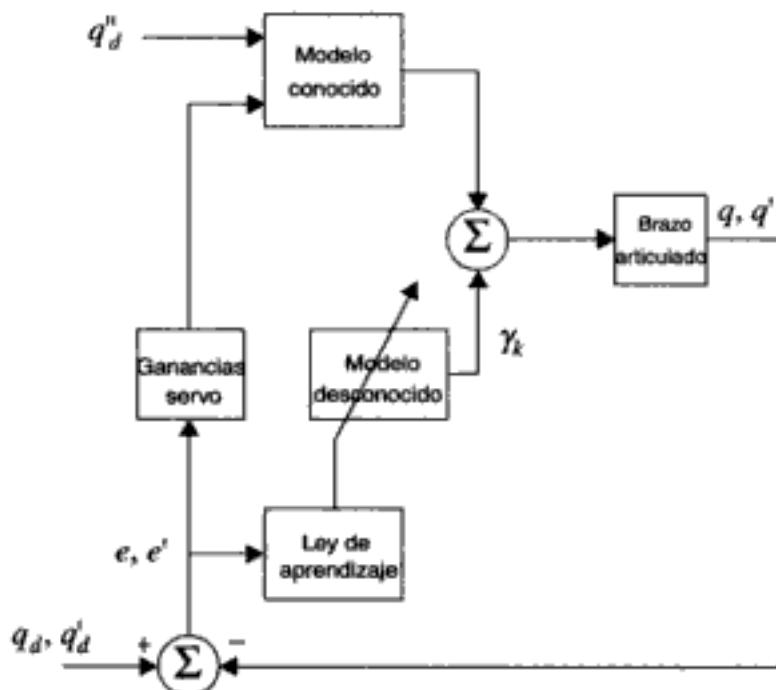


Figura 8.33: Control mediante aprendizaje.

### Ejemplo 8.9

Considérese el manipulador con dos articulaciones de rotación de ejemplos anteriores. Se supone que todos los parámetros del modelo, excepto las fricciones visco-sas, son conocidos. Se pretende que el comportamiento del sistema sea el correspondiente a un sistema de segundo orden críticamente amortiguado y con frecuencia natural no amortiguada de  $\omega_n = 25$ . Por tanto, se escogerán:

$$K_p = \omega_n^2 = 625; \quad K_v = 2\omega_n = 50 \quad (8.151)$$

Supóngase que los valores reales de las fricciones viscosas del modelo son  $b_1 = 10$ ,  $b_2 = 5$ . Como se indicaba en el párrafo anterior estos valores se suponen desconocidos e inicialmente se estiman como cero. Se trata de aprender un par  $\gamma_k$  que compense estas fricciones.

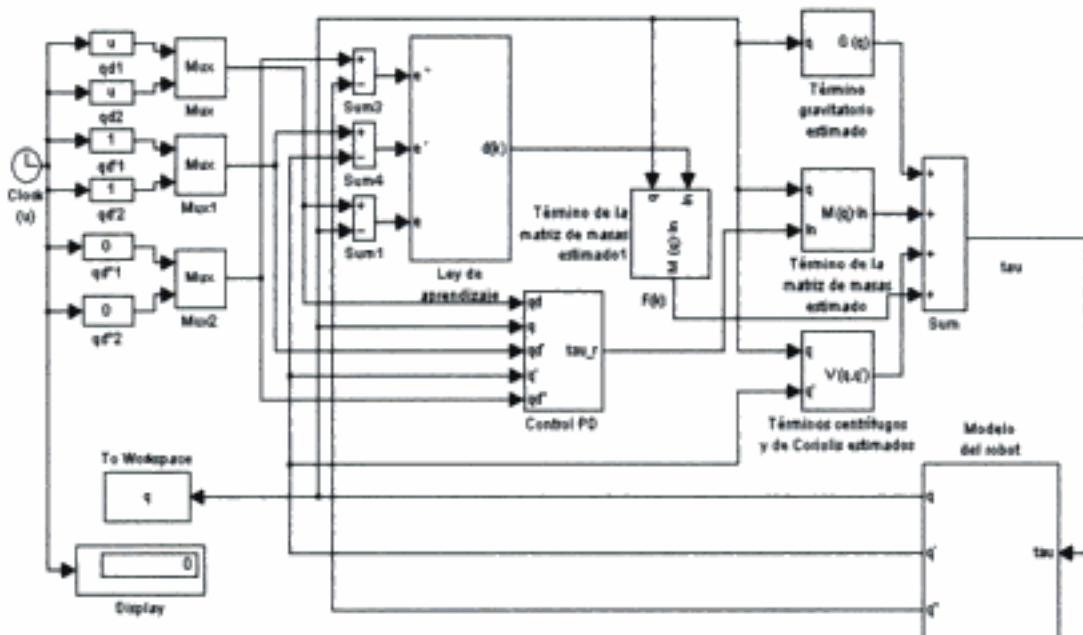
Se pretende seguir la trayectoria dada por:

$$\theta(t) = \begin{bmatrix} t \\ t \end{bmatrix} \quad (8.152)$$

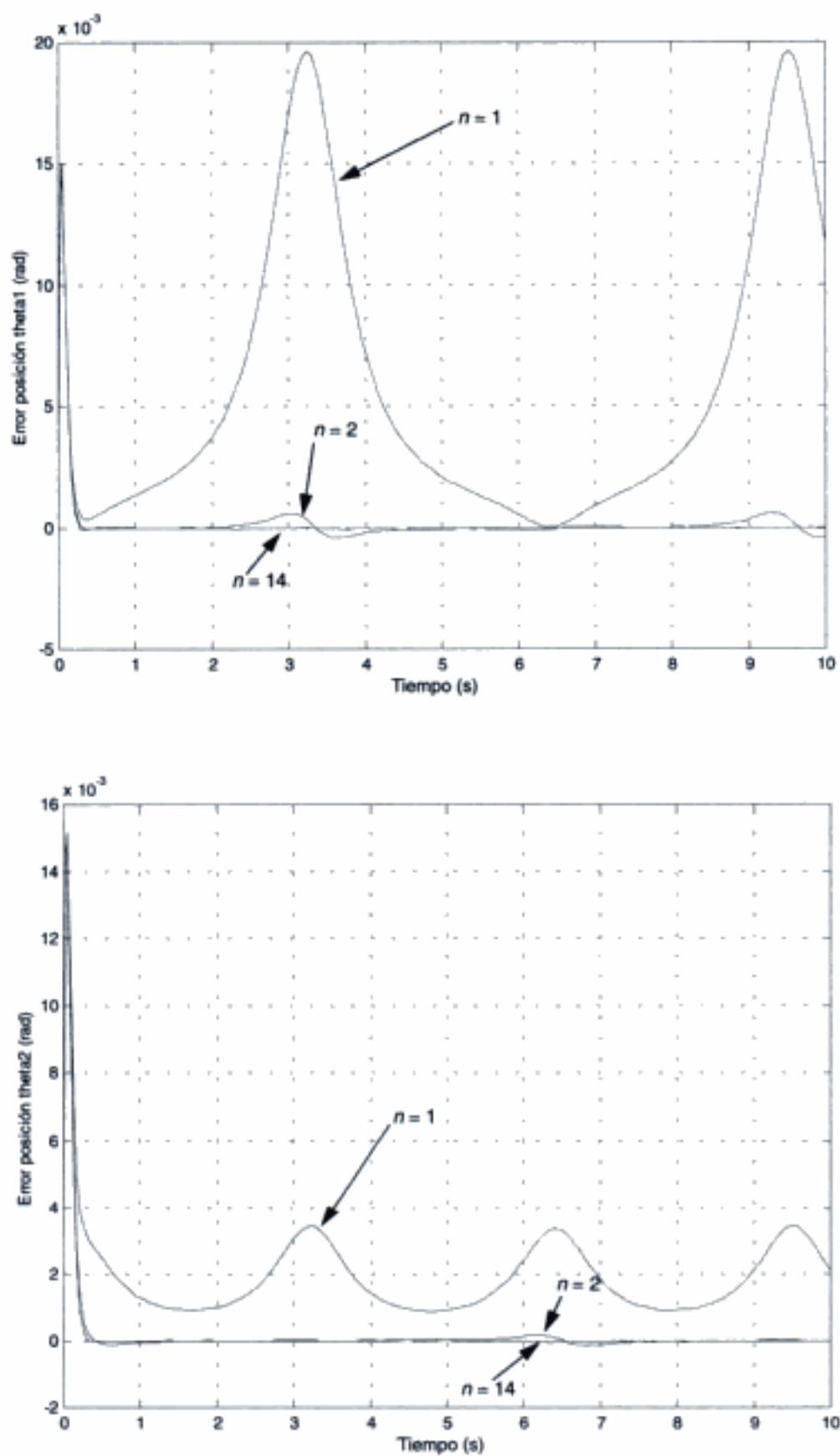
Aplicando la ley de control por aprendizaje mediante el diagrama Simulink de la figura 8.34, con los parámetros de aprendizaje en las dos articulaciones  $\mu_1 = \mu_2 = 3$ , se obtienen los resultados que se indican en la figura 8.35. En la primera iteración, cuando todavía no se aplica el par aprendido, el resultado es el que se indica con  $n = 1$ . Nótese cómo mejora el comportamiento del error en la segunda iteración ( $n = 2$ ) y cómo cuando  $n = 14$  se consigue un comportamiento como el que se especifica mediante el sistema de segundo orden críticamente amortiguado. En la figura 8.36 se muestra la evolución de los errores medios de control:

$$\frac{1}{10} \int_0^{10} e^2 dt \quad (8.153)$$

durante el aprendizaje.



**Figura 8.34:** Diagrama Simulink para la aplicación del método de control con aprendizaje.



**Figura 8.35:** Resultados del algoritmo de control con aprendizaje en el ejemplo.

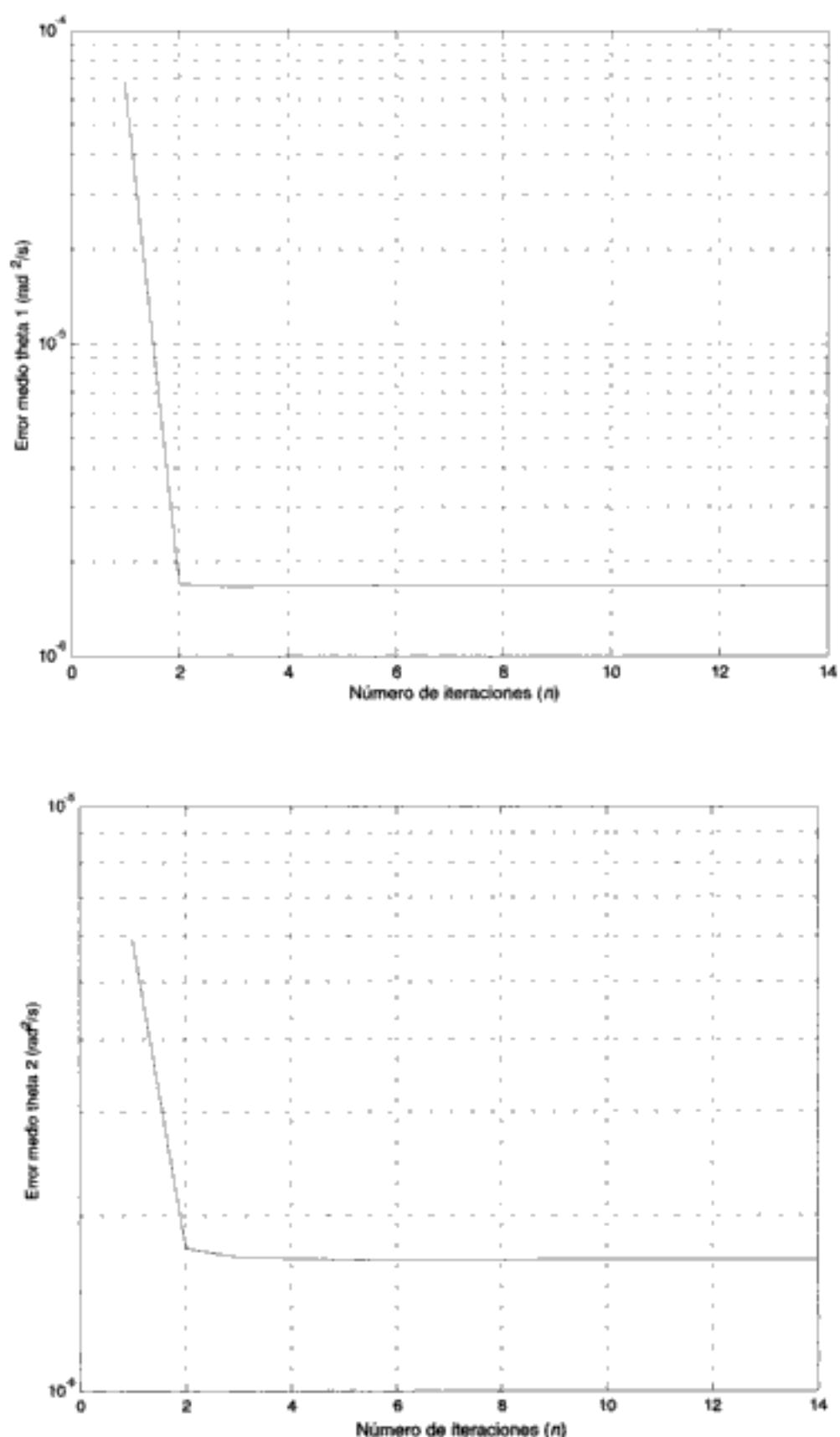


Figura 8.36: Evolución en escala logarítmica de los errores medios durante el aprendizaje.

Otra estrategia diferente de control con aprendizaje consiste en la modificación mediante aprendizaje de los parámetros de un controlador convencional.

En caso de utilizar controladores PID simples se tendría:

$$\tau = q_d^n + K_{vk}e' + K_{pk}e + K_{ik}\int_0^t edt \quad (8.154)$$

Los parámetros en la iteración  $k$  se aprenden empleando los resultados de la iteración anterior (Ollero y otros, 1990).

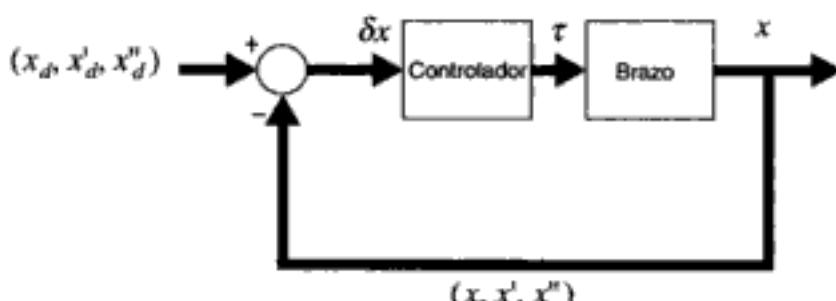
## 8.6 CONTROL EN ESPACIO CARTESIANO

En todos los apartados anteriores se ha considerado el bucle de control en las variables articulares. Para ello, se ha supuesto que existe un generador de trayectorias articulares que obtiene los valores deseados de las variables articulares a partir de los puntos de una trayectoria expresada en el espacio cartesiano, que es en el que normalmente se especifican los movimientos. Como se sabe, la relación entre el espacio de las variables articulares y el de las variables cartesianas viene dado por el modelo geométrico del manipulador (ver capítulo 4).

Sin embargo, existen aplicaciones en las que interesa expresar los bucles de control en el espacio cartesiano. En este caso, la trayectoria que se pretende seguir viene dada por las variables  $x_d, x'_d, x''_d$ , que pueden representar tanto posiciones como orientaciones, así como sus variaciones en el tiempo. Los bucles de control se cierran calculando el error entre estas variables y las variables  $x, x', x''$  asociadas a las posiciones y orientaciones del manipulador, o de forma más precisa de su efecto final, así como de sus variaciones en el tiempo.

Los valores de las variables  $x, x', x''$ , o incluso directamente los de los errores  $x_d - x, x'_d - x', x''_d - x''$ , pueden obtenerse a partir de sensores externos tales como cámaras o sensores activos de distancia, que miden posiciones y orientaciones relativas a rasgos del entorno. En la figura 8.37 se ilustra el bucle de control.

Alternativamente, cabe pensar en obtener las posiciones y orientaciones del efecto final en el espacio a partir de las medidas de las variables articulares y el empleo del modelo cinemático directo, tal como se ilustra en la figura 8.38 en la cual  $\phi(q)$  representa el modelo directo del manipulador  $x = \phi(q)$ .



**Figura 8.37:** Bucle de control en el espacio de trabajo empleando sensores de percepción del entorno.

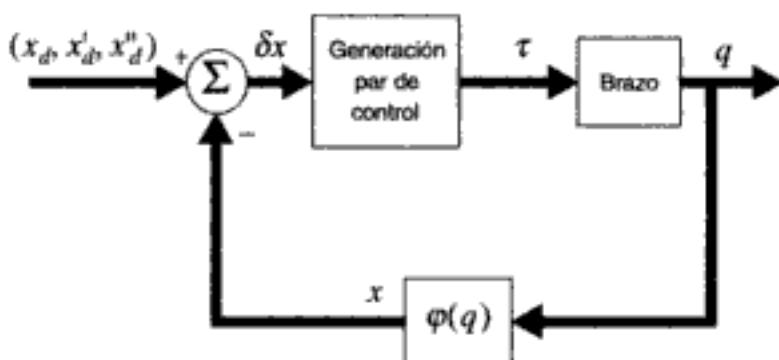


Figura 8.38: Control en espacio cartesiano empleando modelo cinemático.

Nótese que el problema de control con referencias deseadas en el espacio cartesiano puede reducirse también al considerado en los apartados anteriores de este capítulo, aplicando el modelo inverso para generar referencias en el espacio articular:

$$q_d = \varphi^{-1}(x_d) \quad (8.155)$$

$$q'_d = J^{-1}(q)x'_d \quad (8.156)$$

$$q''_d = [J^{-1}(q)]'x'_d + J^{-1}(q)x''_d \quad (8.157)$$

expresiones en las cuales  $\varphi^{-1}$  representa el modelo geométrico inverso, y  $J$  el jacobiano del manipulador. La realización de todas las operaciones que se expresan en las ecuaciones anteriores es laboriosa. En la práctica suele calcularse sólo  $q_d$  según el modelo inverso, aproximando mediante diferencias las velocidades y aceleraciones.

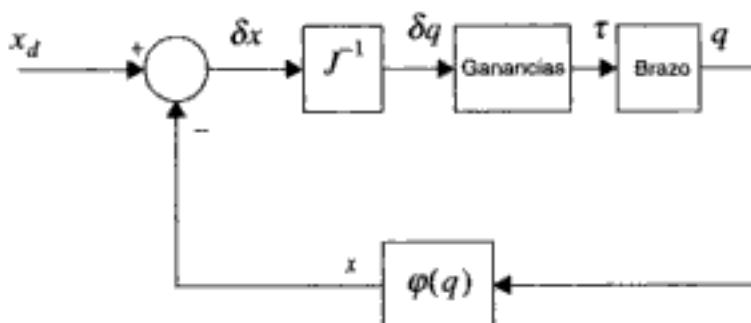
En la figura 8.39 se muestra otro esquema alternativo de control en el cual los errores en las variables articulares se determinan a partir de los errores en el espacio cartesiano empleando la inversa del jacobiano.

La dificultad principal es la transformación de coordenadas en el bucle de control, que tiene una notable complejidad computacional debido a que es necesaria la inversión del jacobiano en tiempo real. El resultado, a igual capacidad computacional, es un bucle de control más lento, lo cual da origen a peores propiedades dinámicas.

Por otra parte, nótese que si el jacobiano no es una matriz cuadrada, es necesario acudir a la seudoinversa (ver capítulo 4). En Klein y Huang (1983) se presentan métodos de control de robots redundantes basados en la seudoinversa.

En la vecindad de las singularidades del jacobiano, o su seudoinversa, los métodos mencionados en los párrafos precedentes presentan problemas numéricos. Para superar estos problemas se han presentado los métodos denominados de "inversa de matrices cuadradas amortiguadas" (Wampler, 1986; Nakamura y Hanafusa, 1986) en los cuales en vez del modelo jacobiano  $x' = Jq'$ , se resuelve la ecuación cinemática modificada  $J^T x' = (J^T J + \lambda^2 I)q'$ , siendo  $\lambda$  el denominado factor de amortiguamiento (cuando  $\lambda = 0$  la ecuación es la habitual).

Otro esquema posible es el que resulta de considerar los esfuerzos que se producen en el efecto final debidos a los pares de control en las articulaciones. Si  $\psi$  es el vec-



**Figura 8.39:** Bucle de control con inversión del jacobiano (realimentación de velocidad no mostrada).

tor de fuerzas y pares que se produce en el efecto final, la relación con los pares de las articulaciones viene dada por:

$$\tau = J^T(q)\psi \quad (8.158)$$

Esta expresión puede demostrarse considerando la conservación de la energía en el movimiento del manipulador:

$$q'^T\tau = x'^T\psi \quad (8.159)$$

Teniendo en cuenta que el jacobiano del manipulador se define mediante

$$x' = Jq' \quad (8.160)$$

Sustituyendo (8.160) en (8.159) se obtiene:

$$q'^T\tau = q'^TJ^T\psi \quad (8.161)$$

La expresión (8.158) se consigue teniendo en cuenta que (8.161) debe verificarse para todo  $q'$ .

Por consiguiente, si los errores entre la posición-orientación del efecto final, con respecto a los deseados, vienen dados por  $\delta x$  y se considera que para anularlos hay que ejercer unos esfuerzos  $\psi$  en el efecto final, se aplicaría la matriz traspuesta del jacobiano, como se indica en la expresión (8.158), para calcular los pares de control necesarios. El esquema simplificado correspondiente se muestra en la figura 8.40.

Hay que poner de manifiesto que los esquemas de las figuras 8.39 y 8.40 coinciden cuando se trata de un manipulador cartesiano, ya que en este caso se verifica:

$$J^T = J^{-1} \quad (8.162)$$

El comportamiento dinámico de los anteriores esquemas de control es de difícil análisis.

Se considera a continuación un esquema con controlador de desacoplamiento en el espacio cartesiano (Craig, 1986). En este caso, la ecuación dinámica en el espacio cartesiano es (Khatib, 1985):

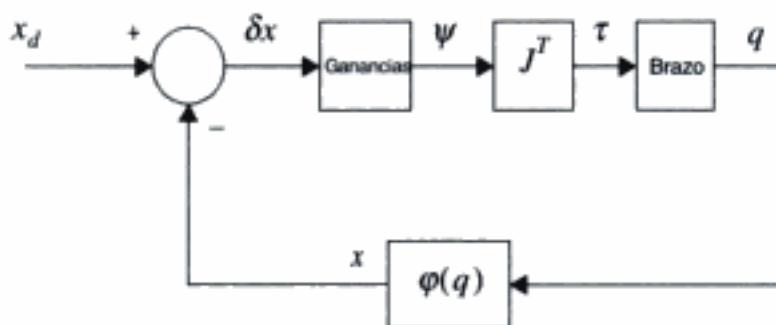


Figura 8.40: Controlador en espacio cartesiano empleando traspuesta del jacobiano.

$$\psi = M_x(q)x'' + V_x(q, q') + G_x(q) \quad (8.163)$$

siendo  $\psi$  el vector de fuerzas/pares actuando en el elemento terminal,  $M_x$  la matriz de masas en espacio cartesiano,  $V_x$  el vector con los términos de velocidad en el espacio cartesiano,  $G_x$  el vector gravedad en espacio cartesiano y  $x$  un vector cartesiano representando posición y orientación del órgano terminal.

Al igual que en el apartado anterior puede emplearse el modelo dinámico para desacoplar y linealizar. Nótese que  $\psi$  es un vector de esfuerzos en el espacio cartesiano. Por tanto, se necesita calcular los pares necesarios en las articulaciones, para lo cual se emplea la (8.158).

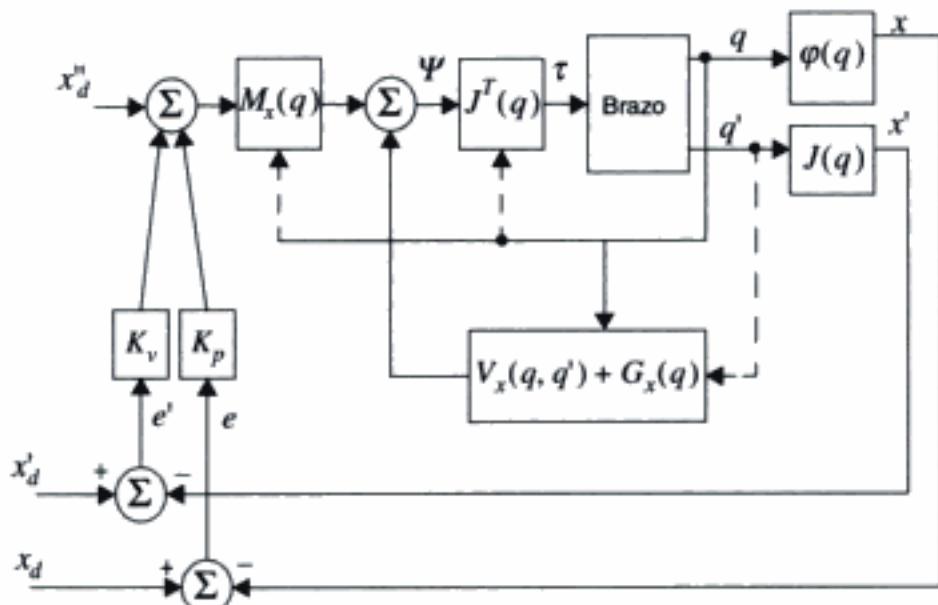


Figura 8.41: Controlador con modelo dinámico en el espacio cartesiano.

En la figura 8.41 se muestra un posible esquema de control. Obsérvese cómo, a diferencia de los esquemas de control convencionales, en el esquema de la figura no es necesario emplear un generador de trayectorias para obtener las consignas en el espacio articular, utilizándose directamente las trayectorias en el espacio cartesiano.

## 8.7 CONTROL DE ESFUERZOS

Esta estrategia de control es de gran interés en aplicaciones en las que el manipulador debe mantener contacto o aplicar fuerzas. Se basa en el empleo de sensores de esfuerzos que permiten medir el vector  $\psi$  en un espacio de la tarea que normalmente se definirá en el espacio cartesiano. En lo que sigue no se efectúan distinciones entre espacio de la tarea y espacio cartesiano, empleando únicamente la denominación espacio cartesiano con una dimensión  $N$ .

### 8.7.1 Rigidz

En las estrategias de control de esfuerzos se emplea el concepto de rigidz. Para introducir este efecto, considérese la relación entre la fuerza ejercida  $\psi_e$  y el desplazamiento ( $x - x_e$ ) en el espacio cartesiano:

$$\psi_e = K_e(x - x_e) \quad (8.164)$$

siendo  $x_e$  la posición del objeto en contacto con el efecto final del manipulador y  $K_e$  una matriz constante  $N \times N$  diagonal, semidefinida positiva, que representa la rigidz del entorno. Si el movimiento del manipulador no está restringido en una determinada dirección del espacio, entonces es nulo el elemento diagonal correspondiente de la matriz.

El modelo dinámico del manipulador interaccionando con objetos en el espacio puede representarse mediante:

$$\tau = M(q)q'' + V_m(q, q')q' + G(q) + F(q') + \tau_e \quad (8.165)$$

siendo

$$\tau_e = J^T(q)\psi_e \quad (8.166)$$

un vector que representa los pares que es necesario suministrar en las articulaciones debido a la interacción con el entorno.

En la ecuación (8.165), el vector de fricciones  $F(q')$  depende exclusivamente de las velocidades, y el vector de esfuerzos centrífugos y de Coriolis  $V(q, q')$  se ha escrito en la forma factorizada (ver capítulo 5).

Obsérvese que con el manipulador parado  $q'' = q' = 0$  la ecuación (8.165) se convierte en:

$$\tau = G(q) + \tau_e \quad (8.167)$$

Es decir, los pares de control se emplean exclusivamente en vencer la gravedad y actuar sobre el entorno.

\* \* \*

### 8.7.2 Control de rigidez

En Lewis y otros (1993) se propone un *controlador de rigidez* de tipo PD que genera los pares de las articulaciones mediante:

$$\tau = \hat{J}^T(q)(-K_v x' + K_p \tilde{x}) + \hat{G}(q) + \hat{F}(q') \quad (8.168)$$

siendo  $K_p$  y  $K_v$  matrices  $N \times N$  diagonales definidas positivas y

$$\tilde{x} = x_d - x \quad (8.169)$$

el error en el espacio cartesiano.

Conviene poner de manifiesto que en control de esfuerzos normalmente se tendrá  $K_e \gg K_p$ ; es decir, la rigidez del entorno es mucho mayor que la ganancia que se introduce con el controlador.

Suponiendo  $\hat{J} = J$ ,  $\hat{G}(q) = G(q)$ ,  $\hat{F}(q') = F(q)$  y sustituyendo (8.168) y (8.164) en la ecuación dinámica del manipulador (8.165)-(8.166) se llega a:

$$M(q)q'' + V_m(q, q')q' = J^T(q)[-K_v x' + K_p \tilde{x} - K_e(x - x_e)] \quad (8.170)$$

Para estudiar la estabilidad del sistema en bucle cerrado, considérese la función:

$$\mathcal{V} = \frac{1}{2}q'^T M(q)q' + \frac{1}{2}\tilde{x}^T K_p \tilde{x} + \frac{1}{2}(x - x_e)^T K_e(x - x_e) \quad (8.171)$$

Derivando con respecto al tiempo y teniendo en cuenta que  $\tilde{x}' = J\dot{q}'$  se obtiene:

$$\mathcal{V}' = \frac{1}{2}q'^T M'(q)q' + q'^T M(q)q'' - q'^T J^T(q)K_p \tilde{x} + q'^T J^T(\theta)K_e(x - x_e) \quad (8.172)$$

Sustituyendo ahora  $M(q)q''$  por el resultado de despejar en (8.170) se obtiene:

$$\mathcal{V}' = \frac{1}{2}q'^T M'(q)q' - q'^T V_m(q, q')q' - q'^T J^T(q)K_v J(q)q' \quad (8.173)$$

Puede demostrarse que la matriz  $S = M' - 2V_m$  satisface la propiedad  $z^T S z = 0$  para cualquier  $z \in \mathfrak{R}^n$ . Por tanto, se tendrá:

$$\mathcal{V}' = -q'^T J^T(q)K_v J(q)q' \quad (8.174)$$

que es no positiva.

Si  $J(q)$  es no singular, también será no singular la  $J^T(q)K_v J(q)$ . Por tanto,  $\mathcal{V}'$  sólo puede ser cero en las trayectorias en que  $q' = 0$  y, por tanto,  $q'' = 0$ . Sustituyendo en (8.170) se obtiene en régimen permanente:

$$\lim_{t \rightarrow \infty} [K_p \tilde{x} - K_e(x - x_e)] = 0 \quad (8.175)$$

Teniendo en cuenta que  $\tilde{x} = x_d - x$  en el límite se verifica  $K_p x_d + K_e x_e = (K_p + K_e)x$ . Por consiguiente, se tendrá:

$$\lim_{t \rightarrow \infty} (x_i) = (K_{p_i} + K_{e_i})^{-1} (K_{p_i} x_{d_i} + K_{e_i} x_{e_i}) \quad (8.176)$$

en cada una de las componentes del espacio cartesiano.

La expresión (8.176) define el límite del movimiento en la componente  $i$  del espacio cartesiano. La fuerza correspondiente a este límite se obtiene sustituyendo (8.176) en (8.164), con lo cual en cada componente se tiene:

$$\lim_{t \rightarrow \infty} \psi_i = K_{e_i} (K_{p_i} + K_{e_i})^{-1} K_{p_i} x_{d_i} + K_{e_i} [(K_{p_i} + K_{e_i})^{-1} K_{e_i} - 1] x_{e_i} \quad (8.177)$$

Con lo cual, teniendo en cuenta que  $(K_{p_i} + K_{e_i})$  es un escalar se llega a:

$$\lim_{t \rightarrow \infty} \psi_i = K_{e_i} (K_{p_i} + K_{e_i})^{-1} K_{p_i} (x_{d_i} - x_{e_i}) \quad (8.178)$$

Teniendo en cuenta que en control de esfuerzos se hace  $K_{p_i} \ll K_{e_i}$ , la expresión (8.178) puede escribirse como:

$$\lim_{t \rightarrow \infty} \psi_i \approx K_{p_i} (x_{d_i} - x_{e_i}) \quad (8.179)$$

con lo cual las ganancias  $K_{p_i}$  pueden interpretarse como las especificaciones de la rigidez del manipulador en las direcciones del espacio cartesiano. Es decir, estas ganancias se emplean para ajustar la rigidez del manipulador.

Si el movimiento del manipulador no está restringido mediante esfuerzos en alguna dirección del espacio cartesiano, la correspondiente componente del vector  $K_{e_i}$  es nula. Sustituyendo en (8.176) se obtiene:

$$\lim_{t \rightarrow \infty} (x_i) = x_{d_i} \quad (8.180)$$

con lo cual se alcanza, en régimen permanente, el valor deseado.

En el espacio de dimensión  $N$  la expresión (8.179) puede escribirse:

$$\lim_{t \rightarrow \infty} \psi \equiv K_p (x_d - x_e) \quad (8.181)$$

Las expresiones anteriores permiten definir una estrategia de control de esfuerzos.

### Ejemplo 8.10

Considérese un manipulador plano con dos articulaciones prismáticas que actúa sobre una superficie  $S$  tal como se muestra en la figura 8.42 (Lewis y otros, 1993).

El modelo dinámico de dicho manipulador viene dado por:

$$\tau = M q'' + G + F(q') + J^T \psi \quad (8.182)$$

siendo

$$M = \begin{bmatrix} m_1 & 0 \\ 0 & m_1 + m_2 \end{bmatrix}; \quad G = \begin{bmatrix} 0 \\ (m_1 + m_2)g \end{bmatrix}; \quad F(q') = \begin{bmatrix} F_1(d_1') \\ F_2(d_2') \end{bmatrix}; \quad \psi = \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} \quad (8.183)$$

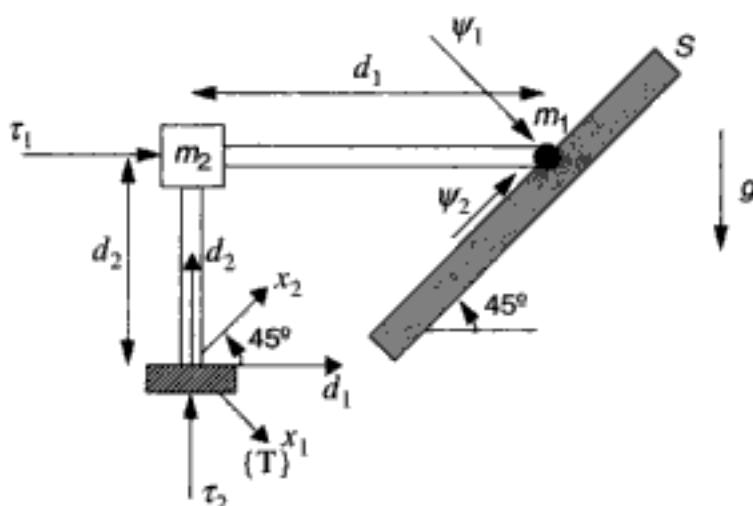


Figura 8.42: Manipulador plano moviéndose a lo largo de la superficie plana  $S$ .

Para estudiar la interacción entre el manipulador y la superficie se define un sistema de coordenadas  $\{T\}$  asociado al espacio de la tarea. En este sistema de referencia,  $x_1$  representa la distancia a la superficie y  $x_2$  la distancia a lo largo de la superficie. El vector del espacio de la tarea es:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (8.184)$$

Mediante relaciones geométricas, en la figura 8.42 se comprueba fácilmente que la relación entre este vector y el vector de variables articulares viene dada por:

$$\boldsymbol{x} = \boldsymbol{\varphi}(q) = \frac{1}{\sqrt{2}} \begin{bmatrix} d_1 - d_2 \\ d_1 + d_2 \end{bmatrix} \quad (8.185)$$

Por tanto, la matriz jacobiana del espacio de trabajo es:

$$J = \frac{\partial}{\partial q} \boldsymbol{\varphi}(q) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad (8.186)$$

Supóngase que se desea mover el efecto final a una posición definida por  $x_{d_2} = 2$  m (desplazamiento de 2 m a lo largo de la superficie), tratando de ejercer una fuerza  $\psi_{d_1} = 3$  N normal a la superficie. Se desprecia la fricción de la superficie (y, por tanto,  $\psi_2$ ) y las fricciones articulares del manipulador. Teniendo en cuenta (8.164) se considera que la fuerza normal a la superficie satisface la relación:

$$\psi_1 = K_{e_1} (x_1 - x_{e_1}) \quad (8.187)$$

donde  $x_{e_1} = 2$  m y  $K_{e_1} = 1200$  N/m. La posición inicial del efecto final viene dada por:

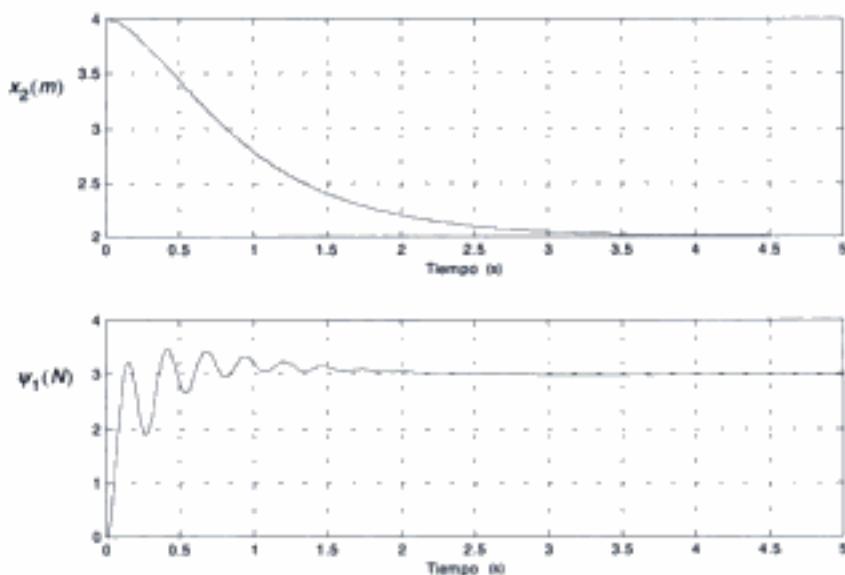


Figura 8.43: Simulación del control de esfuerzos.

$$x_1(0) = 2 \text{ m}; x_2(0) = 4 \text{ m} \quad (8.188)$$

Si se emplea un *controlador de rigidez* de tipo PD, el par que es necesario aplicar sería el correspondiente a la ecuación (8.168) que, cuando se desprecian las fricciones, queda:

$$\tau = \hat{J}^T(q)(-K_v x' + K_p \ddot{x}) + \hat{G}(q) \quad (8.189)$$

Se selecciona,

$$K_v = K_p = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

con lo cual se garantiza que  $K_{p_1} \ll K_{e_1}$ .

Para satisfacer el objetivo  $\psi_{d_1} = 3 \text{ N}$ , se emplea (8.179), con objeto de determinar la distancia deseada a la superficie, obteniéndose  $x_{d_1} = (\psi_{d_1}/K_{p_1}) + x_{e_1} = 2,3 \text{ m}$ .

Suponiendo unos valores  $m_1 = 2 \text{ kg}$  y  $m_2 = 0,35 \text{ kg}$  para las masas del manipulador, los resultados de la simulación son los que se muestran en la figura 8.43.

En las gráficas se observa que la posición a lo largo de la superficie y la fuerza normal a la misma especificadas se logran al cabo de unos 4 s.

### 8.7.3 Control híbrido esfuerzos/posición

En el apartado anterior, el objetivo consistía en aplicar una fuerza constante  $\psi_d$  a la que corresponde una posición constante  $x_d$ . El controlador sólo puede utilizarse para mantener una fuerza y posición de referencia. Sin embargo, este controlador no es

apropiado si se pretende seguir en el tiempo una determinada señal de referencia de fuerzas.

En general, el problema que se plantea es el del seguimiento de trayectorias de fuerza y posición para lo cual se aplican estrategias de *control híbrido esfuerzos/posición*. En Raibert y Craig (1981) se presentó un método original de control de esfuerzos. Este método se basa en los conceptos de restricciones naturales y artificiales de la tarea (Craig, 1986). La formulación que sigue es de Lewis y otros (1993).

Se parte de los modelos

$$\begin{aligned} x &= \varphi(q) \\ x' &= J(q)q' \\ x'' &= J(q)q'' + J'(q)q' \end{aligned} \quad (8.190)$$

Despejando en la tercera de las ecuaciones, se obtiene:

$$q'' = J^{-1}(q)[x'' - J'(q)q'] \quad (8.191)$$

Sustituyendo (8.191) en (8.165)-(8.166) se obtiene:

$$\tau = M(q)J^{-1}(q)[x'' - J'(q)q'] + V_m(q, q')q' + G(q) + F(q') + J^T(q)\psi_e \quad (8.192)$$

La ley de control que permite linealizar el comportamiento dinámico es:

$$\tau = \hat{M}(q)\hat{J}^{-1}(q)[p - \hat{J}(q)q'] + \hat{V}_m(q, q')q' + \hat{G}(q) + \hat{F}(q') + \hat{J}^T(q)\psi_e \quad (8.193)$$

siendo  $p$  un vector  $N \times 1$  que representa el comportamiento deseado. En efecto, sustituyendo (8.193) en (8.192) para  $\hat{M} = M$ ,  $\hat{V}_m = V$ ,  $\hat{G} = G$ ,  $\hat{F} = F$ ,  $\hat{J} = J$  se obtiene:

$$x'' = p \quad (8.194)$$

De las  $N$  componentes de este vector, algunas se emplearán para especificar posiciones deseadas mientras que otras se utilizarán para especificar esfuerzos. El control de posición se contempla en las direcciones tangenciales al movimiento, mientras que el control de esfuerzos se refiere a las direcciones normales. Por ello, la expresión anterior puede escribirse como:

$$x''_{Ti} = p_{Ti} \quad (8.195)$$

$$x''_{Nj} = p_{Nj} \quad (8.196)$$

expresiones en las cuales  $Ti$  indica la componente  $i$  de la dirección tangencial, y  $Nj$  la componente  $j$  de la dirección normal.

De acuerdo con lo expuesto anteriormente, es posible aplicar un controlador lineal en el cual se tenga:

$$p_{Ti} = K_{Tvr}\tilde{x}_{Ti} + K_{Tp_i}\tilde{x}_{Ti} + x''_{Tdi} \quad (8.197)$$

siendo

$$\ddot{x}_{Ti} = x_{Tdi} - x_{Ti} \quad (8.198)$$

y  $K_{Tp_i}$  y  $K_{Tv_i}$  ganancias positivas.

Sustituyendo (8.197)-(8.198) en (8.195) se obtiene:

$$\ddot{x}_{Ti}'' + K_{Tv_i} \dot{x}_{Ti} + K_{Tp_i} \ddot{x}_{Ti} = 0 \quad (8.199)$$

expresión que puede utilizarse para ajustar las ganancias. Si estas ganancias son positivas se tendrá:

$$\lim_{t \rightarrow \infty} \ddot{x}_{Ti} = 0 \quad (8.200)$$

Por tanto, se asegura que en régimen permanente se consigue el seguimiento de la posición en las direcciones tangenciales a la superficie.

Por lo que respecta a las componentes normales, las fuerzas ejercidas son:

$$\psi_{Nj} = K_{ej}(x_{Nj} - x_{ej}) \quad (8.201)$$

siendo  $K_{ej}$  la componente  $j$  de la rigidez del entorno, y  $x_{ej}$  la posición estática del entorno en la dirección normal  $x_{Nj}$ .

Tomando la segunda derivada con respecto al tiempo en (8.201) se obtiene:

$$x_{Nj}'' = \frac{1}{K_{ej}} \psi_{Nj}'' \quad (8.202)$$

Sustituyendo en (8.196) se obtiene:

$$\frac{1}{K_{ej}} \psi_{Nj}'' = p_{Nj} \quad (8.203)$$

Si se pretende ejercer fuerzas  $\psi_{Ndj}$  es posible definir bucles de control de esfuerzos en los cuales los errores serán:

$$\psi_{Nj} = \psi_{Ndj} - \psi_{Nj} \quad (8.204)$$

El controlador puede formularse análogamente al de posición, con lo cual se tendrá:

$$p_{Nj} = \frac{1}{K_{ej}} (\psi_{Ndj}'' + K_{Nvj} \dot{\psi}_{Nj} + K_{Npj} \ddot{\psi}_{Nj}) \quad (8.205)$$

siendo  $K_{Nvj}$  y  $K_{Npj}$  ganancias positivas.

Sustituyendo (8.205) en (8.203) se obtiene:

$$\psi_{Nj}'' + K_{Nvj} \dot{\psi}_{Nj} + K_{Npj} \ddot{\psi}_{Nj} = 0 \quad (8.206)$$

Para ganancias positivas se cumplirá:

$$\lim_{t \rightarrow \infty} \psi_{Nj} = 0 \quad (8.207)$$

Por tanto, se asegura que en régimen permanente se consigue el seguimiento de las fuerzas en las direcciones normales a la superficie.

### Ejemplo 8.11

Considérese el manipulador de la figura 8.42. Se pretende diseñar un controlador híbrido esfuerzos/posición con objeto de que dicho manipulador mueva el efecto final sobre la superficie  $S$  siguiendo la trayectoria dada por  $x_{d_2} = 2 + \cos(t)$  m. Durante dicho recorrido deberá ejercer una fuerza normal a dicha superficie dada por  $\psi_{d_1} = 1 - e^{-2t}$  N. Se desprecian las fricciones en las articulaciones y se supone que la fuerza normal verifica:

$$\psi_1 = K_{e_1}(x_1 - x_{e_1}) \quad (8.208)$$

donde  $x_{e_1} = 2$  m y  $K_{e_1} = 1200$  N/m. La posición inicial del efecto final viene dada por:

$$x_1(0) = 2 \text{ m}; x_2(0) = 4 \text{ m} \quad (8.209)$$

y se suponen unas masas  $m_1 = 2$  kg y  $m_2 = 0,35$  kg. Teniendo en cuenta que  $J$  es constante [ver (8.186)] y que  $V = 0$ , la ley de control de la ecuación (8.193) se reduce en este caso a:

$$\tau = \hat{M}(q)\hat{J}^{-1}(q)p + \hat{G}(q) + \hat{J}'(q)\psi \quad (8.210)$$

donde  $p$  es un vector  $2 \times 1$  que representa el comportamiento deseado, y  $\tau$ ,  $J$ ,  $G$  y  $\psi$  son los mismos que se definieron en el ejemplo 8.10.

Con esta ley de control debe llegar al comportamiento definido por (8.194). En este ejemplo las expresiones (8.194), (8.195) y (8.196) pueden representarse como:

$$\begin{bmatrix} x''_1 \\ x''_2 \end{bmatrix} = \begin{bmatrix} x''_{N1} \\ x''_{T1} \end{bmatrix} = \begin{bmatrix} p_{N1} \\ p_{T1} \end{bmatrix} \quad (8.211)$$

Las leyes de control lineal para el control de posición y de esfuerzos son:

$$\begin{aligned} p_{T1} &= x''_{Td1} + K_{Tv1}\tilde{x}_{T1} + K_{Tp1}\tilde{x}_{T1} \\ p_{N1} &= \frac{1}{K_{e1}}(\psi''_{Nd1} + K_{Nv1}\tilde{\psi}_{N1} + K_{Np1}\tilde{\psi}_{N1}) \end{aligned} \quad (8.212)$$

donde  $x_{Td1} = 2 + \cos(t)$ ,  $\psi_{Nd1} = 1 - e^{-2t}$  y  $K_{e1} = 1200$ .

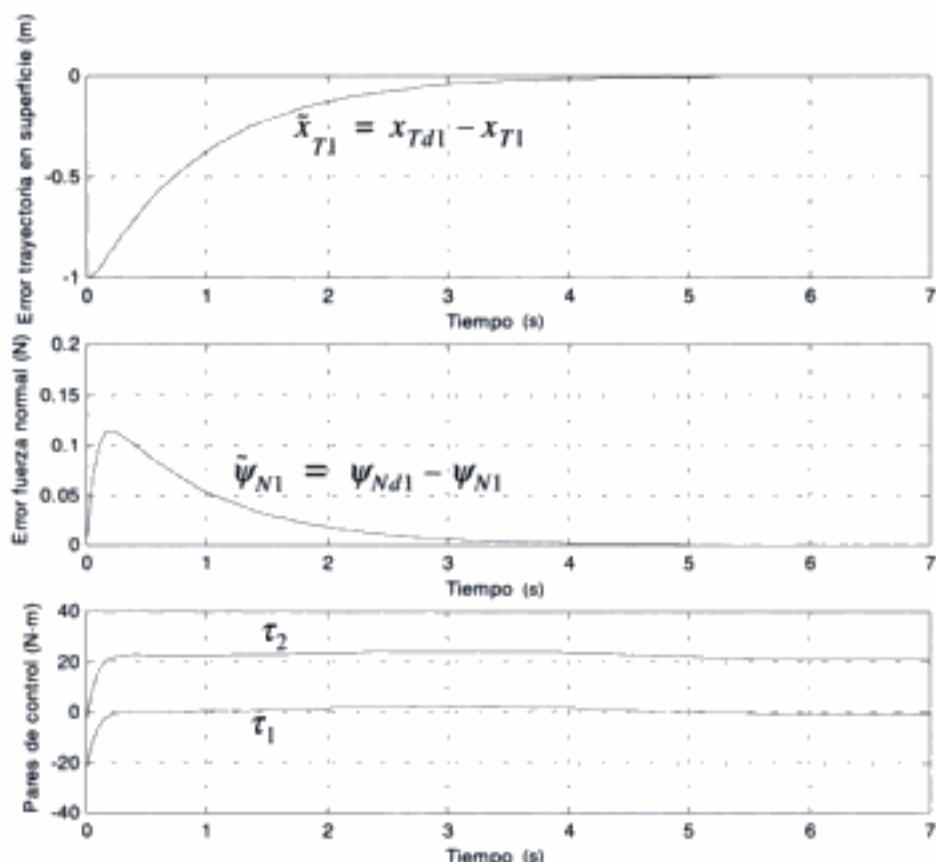
El comportamiento del sistema de control del manipulador ha sido simulado para unos valores de las ganancias:

$$K_{Tv1} = K_{Tp1} = K_{Nv1} = K_{Np1} = 15 \quad (8.213)$$

Sustituyendo todos los valores en (8.212), las ecuaciones que es necesario resolver para simular el comportamiento del brazo son:

$$\begin{aligned}x''_{T1} &= -\cos(t) + 15[-\sin(t) - x'_{T1}] + 15[2 + \cos(t) - x_{T1}] \\ \psi''_{N1} &= \frac{1}{1200}[-4e^{-2t} + 15(2e^{-2t} - 1200\psi'_{N1}) + 15(1 - e^{-2t} - 1200(\psi_{N1} - 2))]\end{aligned}\quad (8.214)$$

Las gráficas resultantes se muestran en la figura 8.44.



**Figura 8.44:** Resultados de la simulación del control híbrido posición/esfuerzos.

Obsérvese como el error de posición inicial es  $x_{2d} - x_2(0) = [2 + \cos(0)] - 4 = -1$ . Tanto los errores de seguimiento en posición como los de esfuerzo se anulan en aproximadamente 5 s.

## 8.8 CONCLUSIONES

En este capítulo se han estudiado estrategias de control de robots manipuladores. Se trata de generar señales de control a las articulaciones de un robot manipulador para que éste describa una trayectoria previamente especificada. El problema involucra esencialmente la consideración del comportamiento dinámico del robot. Para ello se trata de utilizar el modelo dinámico tal como se presentaba en el capítulo 5 de este libro. El conocimiento de este modelo permite formular la estrategia de control denominada de par computado, que no es más que una estrategia basada en el modelo in-

verso que se combina con una estrategia simple de control PD o PID desacoplado de cada una de las articulaciones. La principal dificultad es la necesidad de identificar de forma precisa el modelo del robot manipulador. Existen otros métodos que sólo suponen el conocimiento de ciertos términos del modelo dinámico, tal como los gravitatorios o la matriz de masas. Si el modelo es completamente desconocido y se aplican controladores PID en cada una de las articulaciones, el comportamiento dinámico se deteriora de forma importante. Incluso en el caso en que el modelo dinámico sea conocido, si se producen variaciones significativas en las condiciones de trabajo, tales como el cambio de la carga, el comportamiento dinámico puede verse afectado de forma importante, como se pone de manifiesto en los ejemplos de simulación que se han incluido en el capítulo.

Existen técnicas que tratan de ajustar las constantes del PID dependiendo de la carga o de la región de trabajo, consiguiendo mejorar dicho comportamiento. En general, las técnicas de control adaptativo, que se han tratado de forma introductoria en este capítulo, pretenden tener en cuenta los cambios en el comportamiento dinámico, estableciendo bucles supplementarios de control que permiten considerar la adaptación del modelo y la ley de control para tener en cuenta dichos cambios. Asimismo, se emplean técnicas de control con aprendizaje en las cuales se trata de mejorar la estrategia de control en repeticiones sucesivas del mismo movimiento.

En el capítulo se han tratado, también de forma introductoria, estrategias de control en las cuales los bucles de control se definen en el espacio cartesiano o de la tarea. Estas formulaciones son de interés para considerar realimentación de información de sensores del entorno. Se han tratado, en particular, el control de rigidez que permite mantener fuerzas y posiciones constantes en régimen permanente, y se introducen las estrategias de control híbrido esfuerzos/posición que permiten el seguimiento de fuerzas y posiciones de referencia.

---

## 8.9 REFERENCIAS

---

- Craig J.J., 1986. "Introduction to robotics. Mechanics and control". Addison Wesley.
- Craig J.J., 1988. "Adaptive control of mechanical manipulators". Addison Wesley.
- Canudas de Wit C., B. Siciliano y G. Bastin, 1996. "Theory of robot control". Springer.
- Khatib O., 1985. "The operational space formulation in robot manipulator control". Proc. of the 15th ISIR, Tokio, Japón, septiembre 11-15.
- Klein C.A. y C.H. Huang, 1983. "Review of pseudoinverse control for use with kinematically redundant manipulators". IEEE Trans. on Systems, Man, and Cybernetics. Vol. 13, pp. 245-250.
- Lewis F.L., 1993. "Control of robots manipulators". Macmillan, Inc.
- Nakamura Y. y H. Hanafusa, 1986. "Inverse kinematic solutions with singularity robustness for robot manipulator control". ASME J. of Dynamic Systems, Measurement and Control. Vol. 108, pp. 163-171.
- Ollero A., J. Vicente, J. González y A. Corral, 1990. "A parametric control method with applications to robot manipulators". Proc. IEEE Intl. Conference on Systems Engineering., pp. 164-167, Pittsburgh.
- Ortega R. y M.W. Spong, 1989. "Adaptive motion control of rigid robots: a tutorial". Automática. Vol. 25, pp. 877-888.

- Raibert M.y J. Craig, 1981. "Hybrid position/force control of manipulators". ASME Journal of Dynamic Systems Measurement and Control, junio.
- Slotine J. y W. Li , 1988. "Adaptive manipulator control: a case study". IEEE Trans. on Automatic Control. Vol. 33, pp. 995-1003.
- Spong M.W. y M. Vidyasagar, 1989. "Robot dynamics and control". Wiley.
- Wampler C.W., 1986. "Manipulator inverse kinematic solution based on vector formulations and damped least-squares methods". IEEE Trans. on Systems, Man, and Cybernetics. Vol. 16, pp. 93-101.

# Control de robots móviles

En este capítulo se tratan problemas específicos en robótica móvil. El objetivo es controlar un robot móvil a partir de la información procedente de los sensores existentes con objeto de ejecutar un plan establecido o realizar una determinada tarea reaccionando ante cambios en el entorno. En el capítulo se estudian métodos geométricos para el seguimiento de caminos explícitos. Se exponen también diversos conceptos y métodos basados en la teoría de control aplicándolos al seguimiento de trayectorias y al de caminos explícitos. En particular, se presenta el método del seguimiento predictivo generalizado. Por último, se introducen brevemente métodos de control reactivos basados en la reacción directa a sensores de percepción del entorno. En el capítulo se presentan tanto simulaciones, empleando para ello Simulink, como experimentos reales con vehículos autónomos.

## 9.1 CONTROL DE MOVIMIENTOS DE VEHÍCULOS AUTÓNOMOS

El objetivo es que el vehículo ejecute de forma autónoma movimientos previamente planificados o los que permiten reaccionar de forma apropiada a la percepción del entorno. En este capítulo se considera exclusivamente el control de movimientos en el plano.

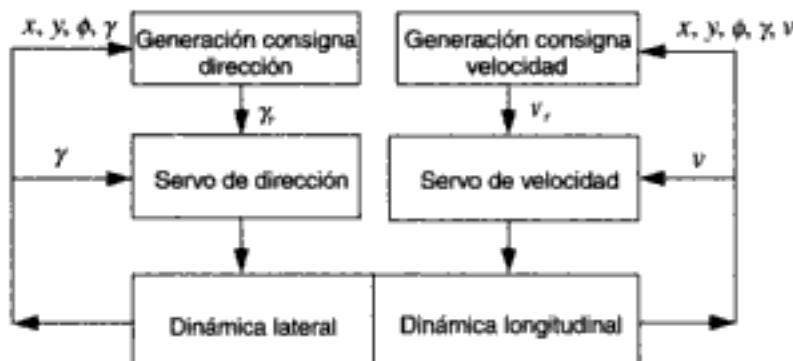
El problema del control de vehículos autónomos ha sido estudiado en teoría de control (ver, por ejemplo, Canudas y otros, 1997). Este problema puede formularse como la obtención de leyes de control que permitan estabilizar el vehículo sobre un punto de trabajo (condiciones nominales de funcionamiento), anulando el efecto de las perturbaciones (problema de regulación), o bien hacer que el vehículo siga de forma autónoma una trayectoria de referencia. En este segundo caso, se distingue entre seguimiento de postura, en el cual se debe seguir la trayectoria en el tiempo de una postura de referencia  $\rho_{ref} = (x, y, \phi)_{ref}$ , constituida por la posición y la orientación, o bien sólo el seguimiento en el tiempo de una posición de referencia  $\rho_{ref} = (x, y)_{ref}$ .

En el problema de seguimiento se pretende que el error  $\rho_{ref}(t) - \rho(t)$  tienda a cero manteniendo acotadas las señales de control.

Nótese que en el problema del seguimiento de  $\rho_{ref}(t)$  se involucra tanto el control de la dirección como el de la velocidad del vehículo autónomo. Este problema se presenta, por ejemplo, cuando se pretende seguir a otro móvil, determinándose  $\rho_{ref}(t)$  mediante el sistema de percepción. Es posible también plantearse un problema similar al del control de un manipulador robótico intentando seguir una trayectoria temporal. Sin embargo, cuando se pretende seguir un camino previamente definido sin considerar la velocidad, se involucra exclusivamente el control de la dirección.

Las características del robot móvil son importantes. Así, las técnicas para controlar pequeños robots omnidireccionales son diferentes de las utilizadas para controlar vehículos automóviles convencionales adaptados para su funcionamiento autónomo.

En la figura 9.1 se muestra un esquema de control de vehículos autónomos en el que se identifican dos bucles. El de la izquierda es el del control de la dirección del vehículo. Se trata de generar las consignas a los actuadores que hacen que el vehículo tome en cada instante la dirección apropiada. El bucle de la derecha es el de control de velocidad. En este caso, se trata de generar la consigna a los actuadores que realizan la propulsión del vehículo.



**Figura 9.1:** Control de velocidad y desplazamiento lateral. La posición del vehículo viene dada por  $x, y$ ;  $\phi$  es la orientación,  $\gamma$  la curvatura y  $v$  la velocidad.

En este punto conviene mencionar que, empleando vehículos convencionales con locomoción tipo triciclo convencional o Ackerman (ver capítulo 2), los bucles de control están fuertemente acoplados.

En el próximo apartado se estudia el problema del seguimiento de caminos explícitos. En el tercer apartado se estudian métodos de seguimiento de caminos que consideran exclusivamente modelos geométricos.

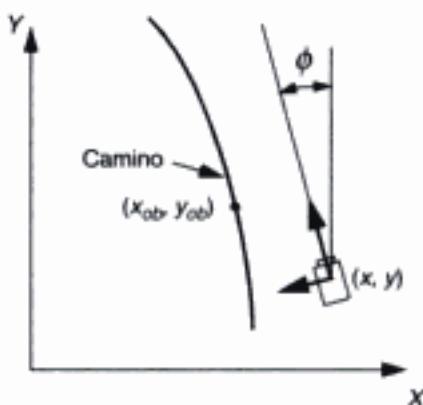
En el cuarto apartado se introduce la aplicación de la teoría de control para el control de robots móviles, y en el quinto la aplicación del método del control predictivo generalizado.

Un planteamiento diferente del problema de control de robots móviles es la del control reactivo. En este caso las actuaciones se obtienen empleando directamente la percepción sensorial del entorno. Este problema se aborda en el sexto apartado.

## 9.2 SEGUIMIENTO DE CAMINOS EXPLÍCITOS

En la figura 9.2 se muestra un vehículo robótico, o robot móvil, y un camino que se pretende que el robot siga de forma autónoma.

En muchos casos, se supone que la velocidad se mantiene constante. En este punto conviene mencionar que la velocidad es un parámetro que interviene en el modelo del bucle de control de la dirección (ver capítulo 4). Por tanto, si se emplea una estrategia de control basada en el modelo, cuando se modifica la velocidad cambia también el modelo y, por tanto, debe modificarse de forma apropiada la ley de control. Para ello pueden aplicarse estrategias tales como las de control adaptativo.



**Figura 9.2:** Seguimiento de caminos explícitos.

El camino que se pretende seguir puede especificarse de diversas formas entre las que cabe mencionar:

- 1) Especificación previa en coordenadas absolutas a partir de planos o mapas, o bien con el propio vehículo dotado de un sistema de GPS diferencial (ver capítulo 7) recorriendo un camino que se graba en memoria para su posterior reproducción.
- 2) Especificación interactiva desde un terminal empleando técnicas de telerrobótica (ver capítulo 13) o instrucciones específicas.
- 3) En una arquitectura de control inteligente empleando métodos de planificación de caminos, tal como se describirá en el capítulo 12. Esta es la técnica que se utiliza en numerosos sistemas de navegación autónoma tales como los del Nav-Lab (Thorpe, 1990), RAM y Romeo (ver capítulo 2).
- 4) Mediante el sistema de percepción, tal como sucede en el seguimiento de líneas obtenidas mediante un sistema de visión sobre el vehículo, o bien en el seguimiento de carreteras empleando también el sistema de visión, tal como se describe en Dickmanns y Zapp (1987) donde se presentaron experiencias pioneras de visión en tiempo real y control de vehículos por autopista.

En cualquier caso, se supondrá que la trayectoria deseada viene dada por una función explícita en coordenadas globales, o bien mediante una secuencia de posturas  $(x, y, \phi, \gamma)$  siendo  $(x, y)$  las coordenadas,  $\phi$  la orientación y  $\gamma$  la curvatura.

Las especificaciones para el diseño de este bucle de control son normalmente las siguientes:

- Precisión en el seguimiento: error de seguimiento lo menor posible.
- Comportamiento dinámico: estabilidad relativa y rapidez de los transitorios.
- Menor esfuerzo de control: minimización de actuaciones.

Para controlar el vehículo sobre la trayectoria se necesita estimar su posición. Como se sabe, el método más simple es mediante odometría empleando las medidas de los codificadores ópticos en los ejes de las ruedas y un cierto modelo. Asimismo, es posible emplear sensores de navegación, tales como los giróscopos, para estimar los ángulos del vehículo, y los compases, para estimar la dirección del vehículo con respecto al Norte.

Se han empleado también otros sensores, tales como los acelerómetros, que permiten obtener la posición del vehículo mediante doble integración de la aceleración medida. No obstante, es necesario poner de manifiesto que, en la mayor parte de los robots móviles, la relación señal/ruido suele ser pequeña (aceleraciones bajas) lo que hace que el error de estimación sea grande.

Los sistemas de navegación inercial (Inertial Navigation Systems, INS) integran diversas medidas de ángulos y aplican técnicas básicas de fusión sensorial para mejorar las estimaciones, añadiendo también otros sensores tales como acelerómetros. No obstante, su coste suele ser elevado.

Por último, en robótica de exteriores, es necesario mencionar también los sistemas de posicionamiento global por satélite (Global Positioning Systems, GPS). En la actualidad existen sistemas diferenciales de posicionamiento global por satélite (DGPS) que permiten llegar a estimar la posición con errores de pocos centímetros (ver capítulo 7).

En general, es posible mejorar la estimación de la posición del vehículo mediante la integración de diferentes sensores, para lo cual se emplean técnicas tales como las del filtro de Kalman.

## 9.3 SEGUIMIENTO DE CAMINOS EMPLEANDO MÉTODOS GEOMÉTRICOS

### 9.3.1 Seguimiento mediante persecución pura

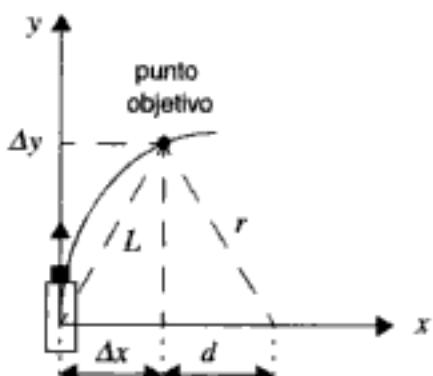
Considérese un sistema de referencia local asociado al movimiento del vehículo, tal como se muestra en la figura 9.3. Se supone que, en el intervalo de control, la curvatura es constante, describiendo el vehículo un arco de circunferencia.

Del análisis de la figura se deducen las relaciones:

$$r = \Delta x + d \quad (9.1)$$

$$d^2 + (\Delta y)^2 = r^2 \quad (9.2)$$

siendo  $r$  el radio de curvatura del vehículo.



**Figura 9.3:** Seguimiento de caminos mediante persecución pura.

Despejando  $d$  en la primera y sustituyendo en la segunda se obtiene:

$$(r - \Delta x)^2 + (\Delta y)^2 = r^2 \quad (9.3)$$

de donde el radio de curvatura necesario para que el vehículo se desplace  $\Delta x$ ,  $\Delta y$  es:

$$r = \frac{(\Delta x)^2 + (\Delta y)^2}{2\Delta x} \quad (9.4)$$

Por tanto, la curvatura que es necesario suministrar al vehículo es:

$$\gamma_r = \frac{1}{r} = -\frac{2(\Delta x)}{L^2} \quad (9.5)$$

donde el signo viene dado por el sentido de giro necesario para alcanzar el punto objetivo en la figura 9.3,  $L$  es la distancia a la que se encuentra el punto objetivo y  $\Delta x$  es el desplazamiento lateral.

La expresión anterior constituye la ley de control de persecución pura ("pure pursuit"). Obsérvese como es una ley de control proporcional al error lateral ( $\Delta x$ ) con respecto al punto objetivo. La constante de proporcionalidad (ganancia) varía con la inversa del cuadrado de  $L$ .

En la figura puede verse también que la curvatura de la persecución pura es la inversa del radio de una de las circunferencias que pasa por la posición actual del vehículo y por el punto objetivo.

El programa para la aplicación de esta ley de control es muy sencillo. Basta determinar el punto del camino que se encuentra a una distancia previamente definida  $L$ , y calcular el error lateral ( $\Delta x$ ) con respecto a la posición actual del centro de guiado del vehículo. Si las coordenadas están en un sistema global, es necesario tener en cuenta la orientación del vehículo para obtener ( $\Delta x$ ). En efecto, si el vehículo está en las coordenadas globales ( $x$ ,  $y$ ) con orientación  $\phi$  (figura 9.2), y el punto objetivo sobre el camino está en las coordenadas globales ( $x_{ob}$ ,  $y_{ob}$ ) entonces se tendrá:

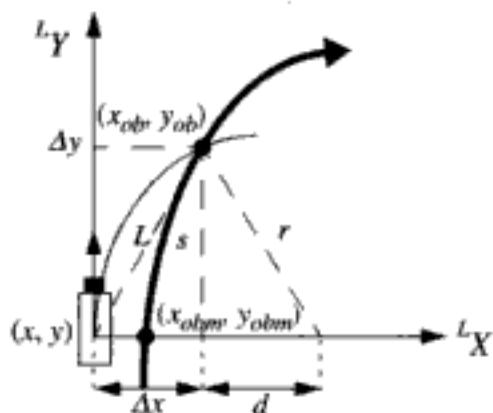
$$\Delta x = (x_{ob} - x) \cos \phi + (y_{ob} - y) \sin \phi \quad (9.6)$$

Un método práctico para aplicar la ley de control consiste en obtener, en cada periodo de control, el punto  $(x_{obm}, y_{obm})$  del camino objetivo que está más próximo al vehículo  $(x, y)$  y elegir el punto objetivo  $(x_{ob}, y_{ob})$  a una distancia fija  $s$  sobre el camino tomada en el sentido de avance a partir de  $(x_{obm}, y_{obm})$ , tal como se ilustra en la figura 9.4. A continuación se calculan:

$$L = \sqrt{(x_{ob} - x)^2 + (y_{ob} - y)^2} \quad (9.7)$$

y (9.6), y se aplica la ley de control (9.5).

En cualquier caso, la elección del parámetro de distancia al punto objetivo es crítica para la eficiencia del controlador de persecución pura. En Ollero y otros (1994) se presenta un método que ajusta automáticamente este parámetro en función de la curvatura del camino, velocidad y situación del vehículo con respecto al camino.



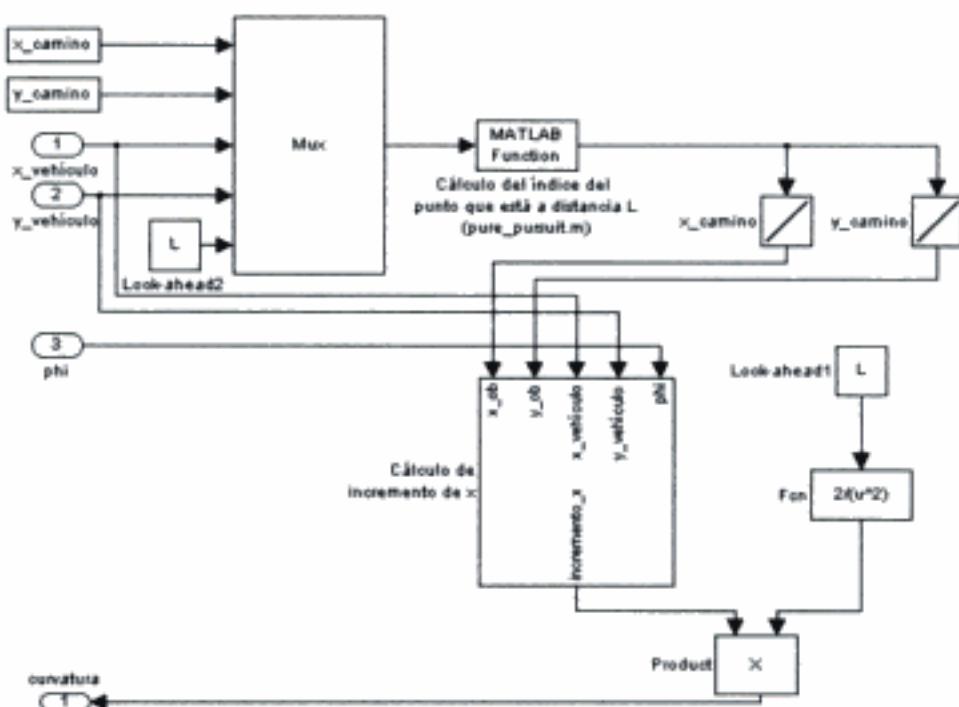
**Figura 9.4:** Aplicación del método de persecución pura con distancia fija sobre camino.

Existen circunstancias en las que la simple aplicación de la ley de control anterior presenta problemas. Así, cuando el punto objetivo se encuentra muy alejado ( $L$  es muy grande), la actuación suministrada por la ley de control puede resultar demasiado pequeña. En este caso, debe sustituirse la distancia  $L$  al punto objetivo por una distancia máxima.

Otro caso patológico es cuando el vehículo se encuentra sobre el camino ( $\Delta x$  muy pequeño), pero orientado en dirección contraria al camino, situación en la cual se obtendría un valor muy pequeño de la curvatura que no permitiría girar el vehículo hasta orientarse según el camino. Si se detecta esta condición, es necesario sustituir la curvatura obtenida por otro valor mayor.

### Ejemplo 9.1

El algoritmo de persecución pura se puede implementar en Simulink mediante el diagrama de la figura 9.5. La utilización de dicho esquema permite efectuar de modo sencillo una serie de simulaciones que tienen por objeto poner de manifiesto el efecto de modificar el parámetro  $L$  en el método de persecución pura.



**Figura 9.5:** Esquema de Simulink para el cálculo de la curvatura mediante el método de persecución pura.

Se considera un robot móvil para el que se supone válido el modelo de la bicicleta (ver capítulo 4):

$$\begin{aligned}x' &= -v \operatorname{sen} \phi \\y' &= v \cos \phi \\ \dot{\phi} &= v \gamma\end{aligned}\quad (9.8)$$

Se pretende que siga, con una velocidad constante de 1 m/s, una trayectoria recta mediante el método de persecución pura. El robot sale del punto de coordenadas (0, 0,5) con una orientación inicial de 0 rad. En la figura 9.6 se muestran los resultados de las simulaciones durante 10 segundos para diferentes valores de  $L$ .

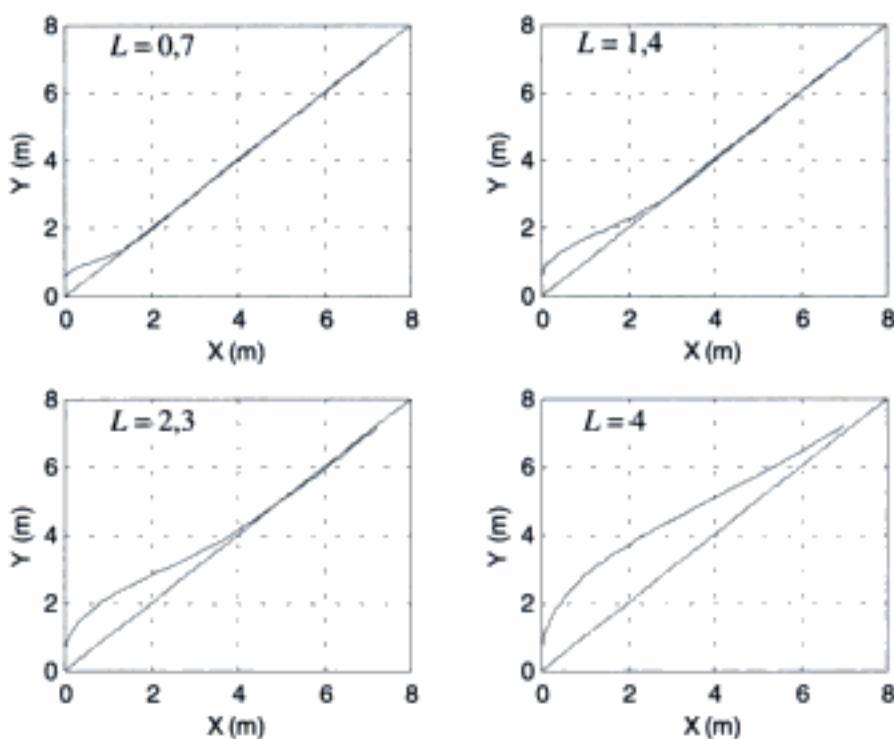
Tal como se indicó en el capítulo 5, es frecuente considerar una dinámica del actuador según un sistema de primer orden, por lo que a las ecuaciones anteriores se le añade:

$$\gamma' = -\frac{\gamma}{\tau} + \frac{K}{\tau} u \quad (9.9)$$

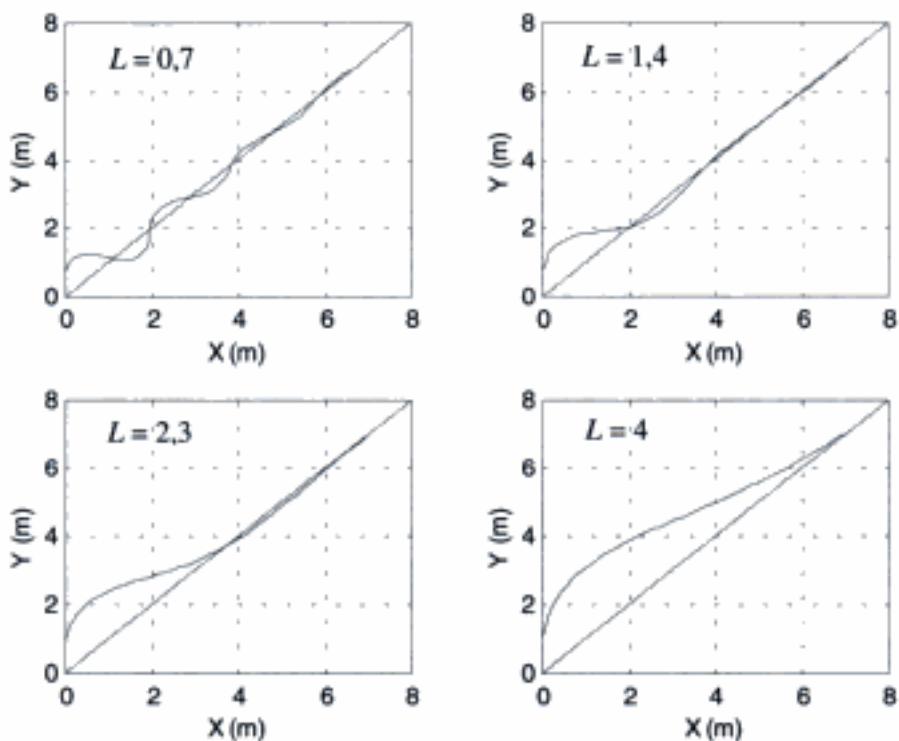
siendo  $u$  la variable de control,  $K$  la ganancia, y  $\tau$  la constante de tiempo. En este ejemplo se han tomado los siguientes valores:

$$K = 1; \quad \tau = 0,5 \text{ s}$$

Añadiendo el sistema de primer orden al diagrama Simulink de la figura 9.5 y volviendo a simular los cuatro casos de la figura 9.6, se obtienen los resultados que se muestran en la figura 9.7



**Figura 9.6:** Trayectorias resultantes de aplicar el método de persecución pura con diferentes valores del parámetro  $L$ .



**Figura 9.7:** Resultados obtenidos con el método de persecución pura modelando el actuador como un sistema de primer orden.

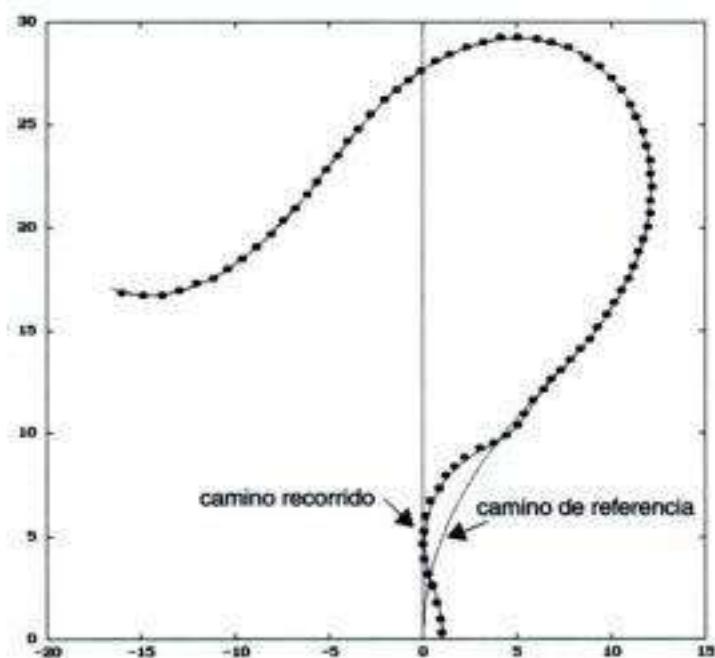


**Figura 9.8:** Vehículo autónomo Romeo 4R desarrollado en los laboratorios de la Escuela Superior de Ingenieros de la Universidad de Sevilla mediante la adaptación de un vehículo eléctrico de 4 ruedas.

Obsérvese como un valor de  $L$  pequeño (ganancia alta) origina oscilaciones en el transitorio, mientras que un valor demasiado alto (ganancia pequeña) hace que el vehículo tarde mucho en alcanzar el camino con la orientación deseada.

### Ejemplo 9.2

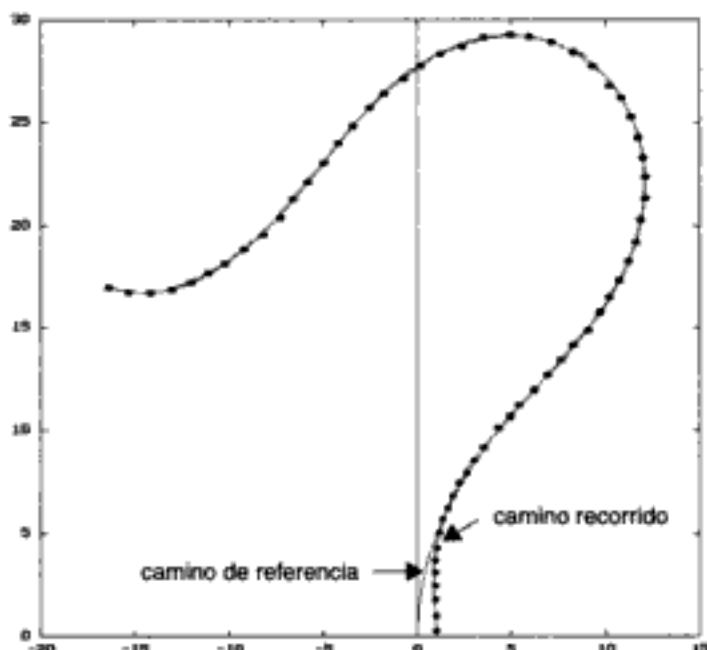
La técnica de seguimiento puro se ha aplicado en el vehículo autónomo Romeo 4R que se muestra en la figura 9.8. En la figura 9.9 se presentan los resultados del seguimiento de un camino de referencia a partir de una posición inicial separada 1 metro del camino. El experimento se realizó a una velocidad de 1,38 m/s. El parámetro de



**Figura 9.9:** Resultados obtenidos mediante la aplicación del método de seguimiento puro con distancia al punto objetivo  $L$  constante.

distancia al punto objetivo, o "look-ahead" es de  $L = 0,45$  m. Obsérvese que después de una fase en la cual el vehículo oscila sobre el camino, el seguimiento es bueno.

Si se pretende mejorar el comportamiento inicial es necesario modificar  $L$ . Sin embargo, esto empeora el comportamiento posterior en el seguimiento. En la práctica los mejores resultados se obtienen con un ajuste automático de  $L$ , en función del error actual y la curvatura del camino, según el método que se presenta en Ollero y otros (1994). Estos resultados se muestran en la figura 9.10.



**Figura 9.10:** Resultados obtenidos mediante la aplicación del método de seguimiento puro con ajuste automático de la distancia al punto objetivo.

Conviene poner de manifiesto que el ajuste automático de  $L$  es imprescindible si se pretende realizar un guiado eficiente a elevadas velocidades. En Rodríguez-Castaño y otros (1999, 2000) se presenta un sistema de guiado autónomo de camiones en el cual la posición del vehículo se estima combinando GPS diferencial (ver capítulo 7), odometría y sensores de navegación. Este sistema ha permitido realizar el guiado autónomo a velocidades de hasta 100 km/h empleando un método de persecución pura con ajuste automático de  $L$  en función de la velocidad, la curvatura del camino y los errores actuales de seguimiento.

### 9.3.2 Control geométrico mediante generación de polinomios de orden cinco

Supuesto elegido un punto objetivo sobre el camino que se pretende seguir, definido mediante la postura  $(x_{ob}, y_{ob}, \phi_{ob}, \gamma_{ob})$  se trata de encontrar una curva que enlaza la posición actual del vehículo caracterizada mediante la postura  $(x_{act}, y_{act}, \phi_{act}, \gamma_{act})$  con el punto objetivo (Shin, 1990).

Suponiendo que los datos están en coordenadas globales, si se transforman a un sistema de referencia asociado al vehículo en su misma dirección, el punto inicial de la curva puede escribirse como:

$$x_{in} = 0; \quad y_{in} = 0; \quad \phi_{in} = 0; \quad \gamma_{in} = \gamma_{act} \quad (9.10)$$

y el punto final en estas mismas coordenadas asociadas al vehículo puede calcularse según:

$$x_f = (x_{ob} - x_{act}) \cos(\phi_{act}) + (y_{ob} - y_{act}) \sin(\phi_{act}) \quad (9.11)$$

$$y_f = (y_{ob} - y_{act}) \cos(\phi_{act}) - (x_{ob} - x_{act}) \sin(\phi_{act}) \quad (9.12)$$

$$\phi_f = \phi_{ob} - \phi_{act} \quad (9.13)$$

$$\gamma_f = \gamma_{ob} \quad (9.14)$$

Suponiendo que la curva se escribe en la forma  $x = f(y)$ , si se adopta para el citado vehículo el modelo de la bicicleta (ver capítulo 4), la primera derivada sería

$$x' = f'(y) = \frac{\Delta x}{\Delta y} = -\operatorname{tg}(\phi) \quad (9.15)$$

La segunda derivada de la curva se calcula al aplicar la regla de la cadena sobre la ecuación anterior:

$$x'' = f''(y) = [-\operatorname{tg}(\phi)]' = \frac{-1}{[\cos(\phi)]^2} \frac{\Delta \phi}{\Delta y} \quad (9.16)$$

Mediante lo cual se obtiene que:

$$x'' = \frac{-1}{[\cos(\phi)]^2} \frac{\gamma \Delta s}{\cos(\phi) \Delta s} = \frac{-\gamma}{[\cos(\phi)]^3} \quad (9.17)$$

Las condiciones iniciales para la determinación de la curva son:

$$f(y_{in}) = f(0) = x_{in} = 0 \quad (9.18)$$

$$f'(y_{in}) = f'(0) = -\operatorname{tg}(\phi_{in}) = -\operatorname{tg}(0) = 0 \quad (9.19)$$

$$f''(y_{in}) = f''(0) = -\frac{\gamma_{in}}{[\cos(\phi_{in})]^3} = -\frac{\gamma_{in}}{[\cos(0)]^3} = -\gamma_{in} \quad (9.20)$$

y las condiciones finales:

$$f(y_f) = x_f \quad (9.21)$$

$$f'(y_f) = -\operatorname{tg}(\phi_f) \quad (9.22)$$

$$f''(y_f) = -\frac{\gamma_f}{[\cos(\phi_f)]^3} = k \quad (9.23)$$

donde  $k$  corresponde a un valor constante que se utilizará más adelante para simplificar la notación empleada.

Nótese que existen seis condiciones. Por tanto, se escoge un polinomio de quinto grado:

$$x = f(y) = a_0 + a_1 y + a_2 y^2 + a_3 y^3 + a_4 y^4 + a_5 y^5 \quad (9.24)$$

en el cual será necesario determinar seis coeficientes  $a_i$ .

Las expresiones de las dos primeras derivadas son:

$$x' = f'(y) = a_1 + 2a_2 y + 3a_3 y^2 + 4a_4 y^3 + 5a_5 y^4 \quad (9.25)$$

$$x'' = f''(y) = 2a_2 + 6a_3 y + 12a_4 y^2 + 20a_5 y^3 \quad (9.26)$$

Con lo cual se obtiene, para las condiciones iniciales:

$$a_0 = 0; \quad (9.27)$$

$$a_1 = 0; \quad (9.28)$$

$$a_2 = -\frac{\gamma_{ini}}{2} \quad (9.29)$$

y para las condiciones finales:

$$x_f = a_0 + a_1 y_f + a_2 y_f^2 + a_3 y_f^3 + a_4 y_f^4 + a_5 y_f^5 \quad (9.30)$$

$$-\operatorname{tg}(\phi_f) = a_1 + 2a_2 y_f + 3a_3 y_f^2 + 4a_4 y_f^3 + 5a_5 y_f^4 \quad (9.31)$$

$$k = 2a_2 + 6a_3 y_f + 12a_4 y_f^2 + 20a_5 y_f^3 \quad (9.32)$$

Nótese que, en la ecuación (9.31), la orientación del punto final  $\phi_f$ , en el sistema de coordenadas del punto inicial, debe pertenecer al intervalo  $[-\pi/2, \pi/2]$ , lo cual se debe a que  $\operatorname{tg}(\alpha) = \operatorname{tg}(\alpha + n\pi)$  siendo  $\alpha$  un cierto ángulo y  $n$  un número entero. Asimismo, para poder calcular la constante  $k$  según (9.23), se necesita que el coseno de  $\phi_f$  difiera de cero, de lo que se deduce que la orientación  $\phi_f$  debe ser distinta de múltiplos enteros de  $\pi/2$ , por lo que resulta que la condición necesaria y suficiente para que el polinomio quíntico pueda calcularse es que  $\phi_f$  pertenezca al intervalo abierto  $(-\pi/2, \pi/2)$ .

Sustituyendo los tres primeros coeficientes en las ecuaciones anteriores, se obtiene:

$$x_f + \frac{\gamma_i}{2} y_f^2 = a_3 y_f^3 + a_4 y_f^4 + a_5 y_f^5 \quad (9.33)$$

$$\gamma_i y_f - \operatorname{tg}(\phi_f) = 3a_3 y_f^2 + 4a_4 y_f^3 + 5a_5 y_f^4 \quad (9.34)$$

$$k + \gamma_i = 6a_3 y_f + 12a_4 y_f^2 + 20a_5 y_f^3 \quad (9.35)$$

Despejando  $a_3$  en la primera:

$$a_3 = \frac{x_f + \frac{\gamma_i y_f^2}{2} - a_4 y_f^4 - a_5 y_f^5}{y_f^3} \quad (9.36)$$

y sustituyendo en la segunda y la tercera:

$$-\left(3x_f + y_f \operatorname{tg}(\phi_f) + \frac{\gamma_i y_f^2}{2}\right) = a_4 y_f^4 + 2a_5 y_f^5 \quad (9.37)$$

$$\frac{ky_f^2}{2} - 3x_f - \gamma_i y_f^2 = 3a_4 y_f^4 + 7a_5 y_f^5 \quad (9.38)$$

Despejando  $a_4$  en la primera:

$$a_4 = -\frac{\frac{\gamma_i y_f^2}{2} + 3x_f + y_f \operatorname{tg}(\phi_f) + 2a_5 y_f^5}{y_f^4} \quad (9.39)$$

y sustituyendo en la segunda se obtiene:

$$a_5 = \frac{6y_f \operatorname{tg}(\phi_f) + 12x_f + (k + \gamma_i)y_f^2}{2y_f^3} \quad (9.40)$$

Sustituyendo  $a_5$  en la (9.39):

$$a_4 = -\frac{2ky_f^2 + 14y_f \operatorname{tg}(\phi_f) + 3\gamma_i y_f^2 + 30x_f}{2y_f^4} \quad (9.41)$$

Por último, basta sustituir estas expresiones de  $a_4$  y  $a_5$  en la (9.36) para obtener:

$$a_3 = \frac{20x_f + ky_f^2 + 3\gamma_i y_f^2 + 8y_f \operatorname{tg}(\phi_f)}{2y_f^3} \quad (9.42)$$

Para aplicar el método, la postura actual del vehículo y la del punto objetivo en coordenadas globales se convierten a coordenadas locales y, a continuación, se calculan los coeficientes del polinomio según las ecuaciones anteriores. El resultado de evaluar la curvatura del polinomio de aproximación en un punto proporciona la nueva curvatura deseada  $\gamma_r$ .

Una variación del método consiste en tener en cuenta la dinámica del sistema de actuación en la aplicación de la curvatura deseada. Para ello, se calcula la curvatura del punto del polinomio donde se prevé que se encontrará el vehículo de acuerdo con el modelo dinámico considerado, típicamente de orden 1. Por tanto, se calcula la curvatura que corresponde a un sistema de primer orden con constante de tiempo previamente determinada de direccionamiento del vehículo. Esta estrategia de adelanto resulta la mejor forma de minimizar los errores provocados por el comportamiento dinámico del robot móvil (Shin, 1990).

Existen situaciones en las que el polinomio de orden 5 resultante da lugar a situaciones patológicas (Martínez, 1995). En particular, se ha puesto de manifiesto que si el punto objetivo está, con respecto a la dirección del vehículo, a un ángulo suficientemente grande (menor de  $\pi/2$ ), la curvatura cambia de forma abrupta en los extremos. Asimismo, si el valor absoluto de la orientación del punto objetivo en coordenadas locales  $\phi_r$  es mayor que un determinado ángulo, se genera una trayectoria en la cual el vehículo debe marchar en dirección contraria al camino y, a continuación, debe girarse con brusquedad para retomar la dirección correcta.

En cualquier caso, nótese que la aplicación de la estrategia es en bucle abierto. Por tanto, el problema que se presenta es cuánto debe viajar el vehículo sobre la trayectoria deseada hasta que se vuelve a determinar un nuevo punto objetivo sobre dicha trayectoria y se vuelve a calcular un nuevo polinomio. Una posible estrategia consiste en volver a calcular un polinomio con un punto objetivo diferente en cada ciclo de control.

## 9.4 APLICACIÓN DE LA TEORÍA DE CONTROL

### 9.4.1 Controlabilidad y estabilización

De acuerdo con lo que se indica en el capítulo 4, los modelos cinemáticos de los robots móviles con ruedas pueden escribirse en la forma:

$$\dot{z} = B(z)u \quad (9.43)$$

siendo  $z$  un vector de variables de estado cuyo número de componentes depende de la configuración y  $u$  el vector de velocidades de control. Dicha configuración está caracterizada por el par  $(\delta_m, \delta_d)$  siendo  $\delta_m$  el grado de movilidad, y  $\delta_d$  el grado de direccionamiento (Campion y otros, 1996). El primero viene dado por  $\delta_m = 3 - nr$ , siendo  $nr$  el número de restricciones de movilidad. Así,  $\delta_m = 3$  corresponde a los robots omnidireccionales. El grado de direccionamiento está determinado por el número de ruedas direccionables. Si no existe ninguna rueda direccionable, como en el caso de los vehículos con locomoción diferencial pura, se tiene  $\delta_d = 0$ . Si existe una rueda o un eje

direccional, como en los triciclos o vehículos automóviles con tracción Ackerman, se tiene  $\delta_d = 1$ .

De esta forma, el robot de locomoción diferencial que se consideraba en el capítulo 4, está caracterizado por el par  $(2, 0)$  mientras que los vehículos tipo triciclo o Ackerman tienen una configuración  $(1, 1)$ .

Si el vehículo no tiene ruedas orientables, entonces el vector de estado  $z$  coincide con el vector de postura:

$$z = \rho = [x \ y \ \phi]^T \quad (9.44)$$

Sin embargo, si existen ruedas orientables, se amplía el vector con  $\delta_d$  componentes correspondientes a los ángulos de las ruedas direccionales.

Si no existen ruedas orientables, la dimensión de  $u$  coincide con el grado de movilidad. Para el caso en que existan estas ruedas direccionales, el vector se amplía con las velocidades de direccionamiento de dichas ruedas.

Sea  $z^*$  un estado de equilibrio en el cual el robot está en reposo con una posición y orientación constantes, y una orientación también constante de las ruedas direccionales. En dicho estado de equilibrio, las velocidades de control son nulas,  $u = 0$ . Linearizando en este estado de equilibrio, se obtiene:

$$\frac{d}{dt}(z - z^*) = B(z^*) \quad (9.45)$$

En Campion y otros (1996) se demuestra que la matriz de controlabilidad se reduce a  $B(z^*)$ , que tiene como rango  $\delta_m + \delta_d$ . Un robot omnidireccional se caracteriza por el par  $(3, 0)$  y la dimensión del vector de estado es  $\delta_m = 3$ . Por tanto, la aproximación lineal es completamente controlable. Sin embargo, si existen restricciones en la movilidad, tal como sucede en la locomoción diferencial  $(2, 0)$  o en la de los triciclos o automóviles  $(1, 1)$ , la aproximación no es completamente controlable.

Así, por ejemplo, el modelo  $(2, 0)$  que se consideraba en el capítulo 4:

$$\begin{aligned} x' &= -v \sin \phi \\ y' &= v \cos \phi \\ \phi' &= \omega \end{aligned} \quad (9.46)$$

puede escribirse en la forma  $z' = B(z)u$ , siendo  $z = [x \ y \ \phi]^T$ ,  $u = [v \ \omega]^T$  las variables de control correspondientes a las velocidades, y

$$B = \begin{bmatrix} -\sin \phi & 0 \\ \cos \phi & 0 \\ 0 & 1 \end{bmatrix} \quad (9.47)$$

Nótese que en esta configuración no existen ruedas direccionales y que, por tanto,  $z = \rho = [x \ y \ \phi]^T$ . Obsérvese también que  $B$  coincide con el jacobiano.

Linearizando en  $z = 0$ ;  $u = 0$ , se obtiene:

$$z' = B(0) \begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \quad (9.48)$$

El sistema no es controlable ya que el rango de la matriz de controlabilidad:

$$C = B(0) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (9.49)$$

es sólo 2.

Sin embargo, lo anterior se refiere sólo a la linealización, en la cual hay involucrada una pérdida de información. En efecto, de forma intuitiva puede afirmarse que el robot móvil es controlable, ya que puede llevase desde una postura inicial  $\rho_0$  a una final  $\rho_f$  en un tiempo finito, manipulando convenientemente las entradas.

Si un robot móvil es omnidireccional, es posible encontrar entradas de control que linealizan el modelo y lo estabilizan en un estado  $z^*$ . En efecto, en este caso es posible aplicar una ley de control:

$$u(z) = B(z)^{-1} A(z - z^*) \quad (9.50)$$

siendo  $A$  una matriz de Hurwitz, tal que el sistema en bucle cerrado viene dado por:

$$\frac{d}{dt}(z - z^*) = A(z - z^*) \quad (9.51)$$

En general, en los robots omnidireccionales siempre es posible diseñar leyes de control continuas que lleven al robot desde un estado inicial hasta un estado de equilibrio estable.

Sin embargo, los robots móviles con restricciones en la locomoción (no omnidireccionales) son sólo parcialmente linealizables por realimentación (Marino, 1986). Asimismo, en los robots de movilidad restringida, el modelo cinemático no es estabilizable mediante una realimentación continua del vector estado invariante en el tiempo  $u(z)$  (Brockett, 1983). Sin embargo, puede encontrarse una ley de control variable en el tiempo  $u(z, t)$  que lo estabilice.

Si en el sistema (9.46) se realiza el cambio de variables:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ -\sin \phi & \cos \phi & 0 \\ \cos \phi & \sin \phi & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ \phi \end{pmatrix} \quad (9.52)$$

y se deriva en (9.52), se obtiene:

$$\begin{aligned} x'_1 &= \omega \\ x'_2 &= -x' \sin \phi - \omega x \cos \phi + y' \cos \phi - \omega y \sin \phi \\ x'_3 &= x' \cos \phi - \omega x \sin \phi + y' \sin \phi + \omega y \cos \phi \end{aligned} \quad (9.53)$$

Sustituyendo las expresiones de  $x'$  e  $y'$  de (9.46), se obtiene:

$$\begin{aligned}x'_1 &= \omega \\x'_2 &= v - \omega x_3 \\x'_3 &= \omega x_2\end{aligned}\tag{9.54}$$

Si ahora se hace,

$$\begin{aligned}u_1 &= \omega \\u_2 &= v - \omega x_3\end{aligned}\tag{9.55}$$

el modelo puede escribirse en la forma:

$$\begin{aligned}x'_1 &= u_1 \\x'_2 &= u_2 \\x'_3 &= x_2 u_1\end{aligned}\tag{9.56}$$

Este modelo pertenece a la clase que se denomina "en cadena", caracterizada por ecuaciones de estado de la forma:

$$\begin{aligned}x'_1 &= u_1 \\x'_2 &= u_2 \\x'_3 &= x_2 u_1 \\&\dots \\x'_n &= x_{n-1} u_1\end{aligned}\tag{9.57}$$

Estos sistemas son de particular interés para el modelado y control de robots móviles con restricciones no holónomas (Samson, 1995).

Nótese que, aunque el sistema es no lineal, si  $u_1$  es una función del tiempo y no una variable de control, puede escribirse:

$$\begin{aligned}\vec{x}_1 &= 0 \\ \begin{bmatrix} x'_2 \\ x'_3 \\ \vdots \\ x'_n \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ u_1(t) & 0 & 0 & 0 \\ 0 & 0 & 0 & u_1(t) \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u_2\end{aligned}\tag{9.58}$$

siendo

$$\tilde{x}_1 = x_1 - \int_0^t u_1(\delta) d\delta$$

Por tanto, en este caso, se tiene un sistema de una sola entrada, lineal y de coeficientes variables en el tiempo. Es decir, si se escribe un sistema no lineal con dos entradas en la forma de cadena, es posible linealizarlo con respecto a una de sus entradas. En Murray (1993) se presenta una condición necesaria y suficiente de controlabilidad. Cuando  $u_1$  se toma como una función del tiempo, el sistema pierde la controlabilidad debido a la primera ecuación de estado. No obstante, bajo ciertas condiciones en la elección de  $u_1(t)$ , el sistema formado por las variables de estado  $[x_2, \dots, x_n]$  permanece controlable. Esta propiedad puede utilizarse para diseñar leyes de control en bucle cerrado que estabilicen asintóticamente el sistema en un punto  $x = 0$ , y para determinar entradas que conduzcan en bucle abierto el vehículo hasta llevarlo a una configuración deseada, que normalmente se escoge como el origen del espacio de estados (Samson, 1995). Este método consiste, básicamente, en dos pasos: 1) elegir una función integrable que asegure la controlabilidad de la segunda parte  $[x_2, \dots, x_n]$ , y determinar un control  $u_2(t)$  que lleve esta segunda parte al origen en tiempo finito (normalmente integrando las ecuaciones del sistema en un intervalo de tiempo y resolviendo un conjunto de ecuaciones algebraicas); y 2) manteniendo  $u_2$  igual a cero para dejar  $[x_2, \dots, x_n]$  en el origen, determinar  $u_1(t)$  para conducir  $x_1(t)$  al origen en un tiempo finito.

El método de dos pasos se ha empleado en Mónaco y Normand-Cyrot (1991) con  $u_1(t)$  y  $u_2(t)$  siendo entradas constantes a intervalos y en Murray y Sastry (1993) con  $u_1(t)$  y  $u_2(t)$  sinusoides.

Nótese que cuando  $u_1$  es constante y diferente de cero, el sistema (9.58) se convierte en invariante en el tiempo y  $[x_2, \dots, x_n]$  es controlable. Por tanto, aplicando métodos clásicos de control lineal, es posible encontrar leyes de control en bucle cerrado sobre el sistema  $[x_2, \dots, x_n]$ .

## 9.4.2 Seguimiento de trayectorias

### 9.4.2.1 Trayectorias de referencia

Considérese que la trayectoria que se pretende seguir viene dada por el modelo de referencia:

$$\begin{aligned} x'_{ref} &= -v_{ref} \sin \phi_{ref} \\ y'_{ref} &= v_{ref} \cos \phi_{ref} \\ \phi'_{ref} &= \omega_{ref} \end{aligned} \quad (9.59)$$

Se supone que  $v_{ref}$  y  $\omega_{ref} = \phi'_{ref}$  y sus derivadas están acotadas.

Se supondrá también que:

$$\lim_{t \rightarrow \infty} v_{ref}(t) \neq 0; \quad \lim_{t \rightarrow \infty} \omega_{ref}(t) \neq 0 \quad (9.60)$$

lo que significa que se pretende seguir una trayectoria, ya que en el caso en que fueran nulas, el problema consistiría en alcanzar una postura de referencia fija.

El problema de seguimiento consiste en encontrar una ley de control:

$$u = \begin{pmatrix} v \\ \omega \end{pmatrix} = \varphi(\rho, \rho_{ref}, v_{ref}, \omega_{ref}) \quad (9.61)$$

tal que:

$$\lim_{t \rightarrow \infty} [\rho(t) - \rho_{ref}(t)] = 0 \quad (9.62)$$

Este problema de seguimiento involucra una ecuación dinámica del error  $\rho - \rho_{ref}$ . Considérese el siguiente cambio de coordenadas:

$$\begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} -\sin \phi & \cos \phi & 0 \\ -\cos \phi & -\sin \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{ref} - x \\ y_{ref} - y \\ \phi_{ref} - \phi \end{pmatrix} \quad (9.63)$$

Derivando con respecto al tiempo, se obtiene:

$$\begin{aligned} e'_1 &= -\omega(x_{ref} - x) \cos \phi - (x'_{ref} - x') \sin \phi - \omega(y_{ref} - y) \sin \phi + (y'_{ref} - y') \cos \phi \\ e'_2 &= \omega(x_{ref} - x) \sin \phi - (x'_{ref} - x') \cos \phi - \omega(y_{ref} - y) \cos \phi - (y'_{ref} - y') \sin \phi \\ e'_3 &= \omega_{ref} - \omega \end{aligned} \quad (9.64)$$

Sustituyendo las expresiones de  $x'_{ref}$ ,  $x'$ ,  $y'_{ref}$ ,  $y'$  de (9.59) y (9.46) se obtiene

$$\begin{aligned} e'_1 &= \omega[(x - x_{ref}) \cos \phi + (y - y_{ref}) \sin \phi] - v(\sin^2 \phi + \cos^2 \phi) + v_{ref}(\sin \phi_{ref} \sin \phi + \cos \phi_{ref} \cos \phi) \\ e'_2 &= \omega[(x_{ref} - x) \sin \phi + (y_{ref} - y) \cos \phi] + v_{ref}(\sin \phi_{ref} \cos \phi - \cos \phi_{ref} \sin \phi) \\ e'_3 &= \omega_{ref} - \omega \end{aligned} \quad (9.65)$$

Sustituyendo (9.63) en (9.65) y haciendo:

$$\begin{aligned} u_1 &= -v + v_{ref} \cos e_3 \\ u_2 &= \omega_{ref} - \omega \end{aligned} \quad (9.66)$$

se llega a

$$e' = \begin{pmatrix} 0 & \omega & 0 \\ -\omega & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} e + \begin{pmatrix} 0 \\ \sin e_3 \\ 0 \end{pmatrix} v_{ref} + \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (9.67)$$

Linealizando el sistema (9.67) alrededor del equilibrio ( $e = 0$ ,  $u = 0$ ), para  $e_3$  suficientemente pequeño, puede emplearse el siguiente sistema lineal variante en el tiempo:

$$e' = \begin{pmatrix} 0 & \omega_{ref}(t) & 0 \\ -\omega_{ref}(t) & 0 & v_{ref}(t) \\ 0 & 0 & 0 \end{pmatrix} e + \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (9.68)$$

Si  $v_{ref}$  y  $\omega_{ref}$  son constantes, se tiene un sistema lineal e invariante en el tiempo cuya matriz de controlabilidad es:

$$C = \begin{bmatrix} B & AB & A^2B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & -\omega_{ref}^2 v_{ref} \omega_{ref} \\ 0 & 0 & -\omega_{ref} & v_{ref} & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (9.69)$$

Si  $v_{ref}$  y  $\omega_{ref}$  son diferentes de cero, el sistema es controlable. Nótese que si  $v_{ref} = \omega_{ref} = 0$  (sistema de referencia en reposo), se pierde la controlabilidad.

#### 9.4.2.2 Ley de control lineal

En Canudas y otros (1997) se propone la ley de control:

$$\begin{aligned} u_1 &= -k_1 e_1 \\ u_2 &= -k_2 \operatorname{sgn}(v_{ref}) e_2 - k_3 e_3 \end{aligned} \quad (9.70)$$

Sustituyendo en la ecuación de error (9.68), se obtiene:

$$e' = Ae = \begin{pmatrix} -k_1 & \omega_{ref} & 0 \\ -\omega_{ref} & 0 & v_{ref} \\ 0 & -k_2 \operatorname{sgn}(v_{ref}) & -k_3 \end{pmatrix} e \quad (9.71)$$

El polinomio característico viene dado por:

$$\det(sI - A) = s(s + k_1)(s + k_3) + v_{ref} k_2 \operatorname{sgn}(v_{ref})(s + k_1) + (s + k_3)\omega_{ref}^2 \quad (9.72)$$

Si se hace  $k_3 = k_1$  la ecuación característica es:

$$(s + k_1)[s(s + k_1) + v_{ref} k_2 \operatorname{sgn}(v_{ref}) + \omega_{ref}^2] = 0 \quad (9.73)$$

Identificando con una ecuación característica:

$$(s + 2\delta b)(s^2 + 2\delta b s + b^2) = 0 \quad (9.74)$$

y escogiendo los valores del coeficiente de amortiguamiento  $\delta$  y frecuencia natural no amortiguada  $b$  para que los polos estén situados donde se desee, los valores de las ganancias de realimentación serían:

$$\begin{aligned} k_1 &= 2\delta b \\ k_2 &= \frac{b^2 - \omega_{ref}^2}{|v_{ref}|} \\ k_3 &= 2\delta b \end{aligned} \quad (9.75)$$

Nótese que la ganancia  $k_2$  aumenta cuando  $|v_{ref}|$  se hace pequeña. Este problema puede evitarse mediante un procedimiento de variación de los polos deseados con la velocidad, al cual se denomina de "escalado de velocidad". De esta forma si, por ejemplo, se hace:

$$b = (\omega_{ref}^2 + \beta v_{ref}^2)^{1/2} \quad (9.76)$$

con  $\beta > 0$ , las ganancias de control son:

$$\begin{aligned} k_1 &= 2\delta(\omega_{ref}^2 + \beta v_{ref}^2)^{1/2} \\ k_2 &= \beta |v_{ref}| \\ k_3 &= 2\delta(\omega_{ref}^2 + \beta v_{ref}^2)^{1/2} \end{aligned} \quad (9.77)$$

Cuando  $v_{ref} = \omega_{ref} = 0$  no se realiza ninguna acción de control, lo cual es coherente con la pérdida de controlabilidad antes mencionada.

### Ejemplo 9.3

Se pretende que un robot móvil siga una trayectoria definida mediante las ecuaciones (9.59), tomando como condiciones iniciales una orientación de 0 rad y las coordenadas (0, 0). Se supondrá que la velocidad lineal de la referencia tiene un escalón de 0,2 m/s en  $t = 0$  s, y en  $t = 20$  s pasa a valer 0,1 m/s. Asimismo, la velocidad angular de la referencia tiene un escalón de 0,3 rad/s en  $t = 0$  s, y pasa a valer 0,2 rad/s en  $t = 10$  s.

Se supondrá que el robot móvil parte de las coordenadas (1, 0,5) con una orientación de  $\pi/6$  rad.

Considérese la aplicación de la ley de control lineal (9.70). En la figura 9.11 se muestra el diagrama Simulink correspondiente.

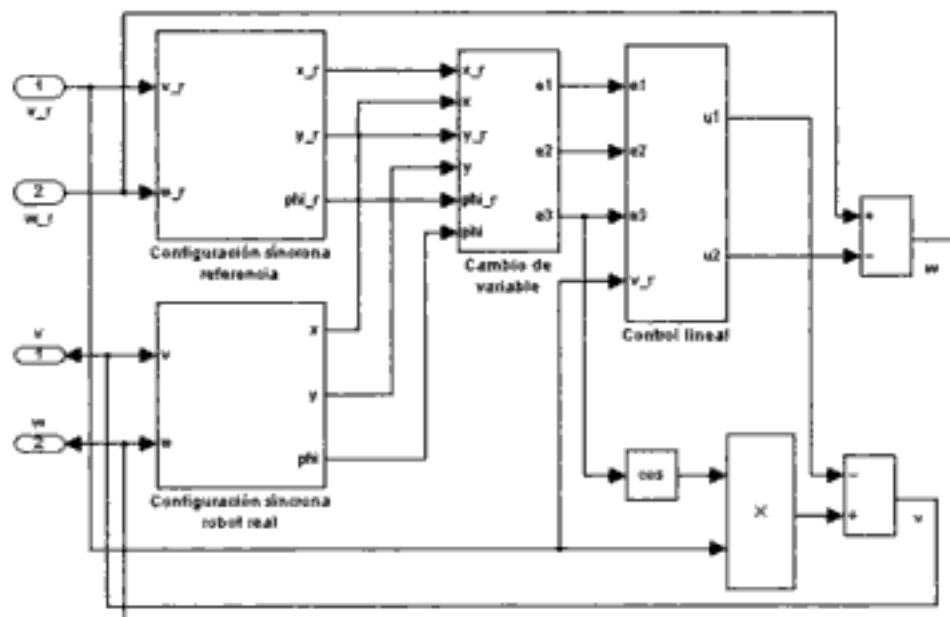
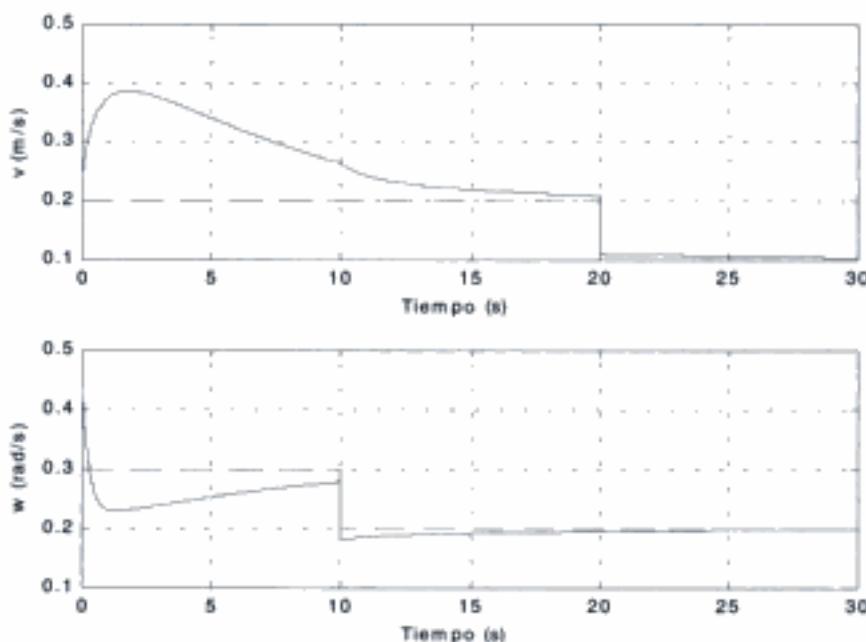


Figura 9.11: Esquema Simulink del control lineal.



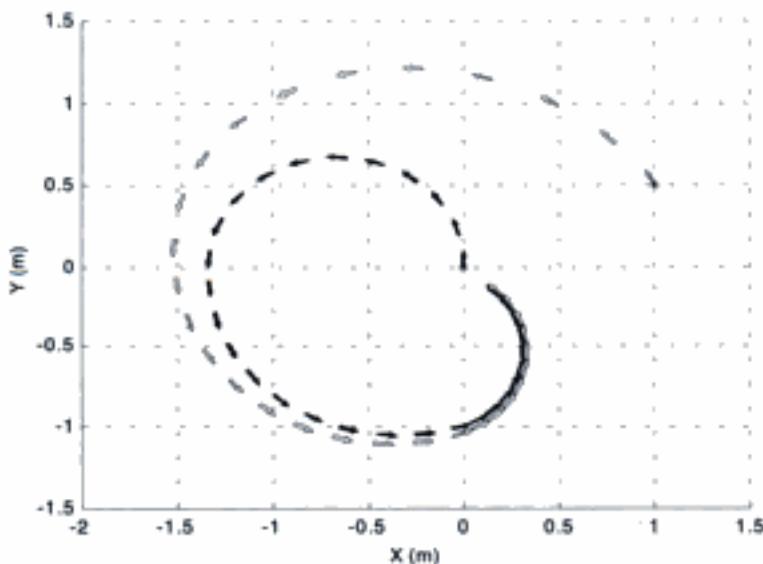
**Figura 9.12:** Velocidades lineal y angular obtenidas.

Se efectúa la simulación para unos valores arbitrarios de las constantes:

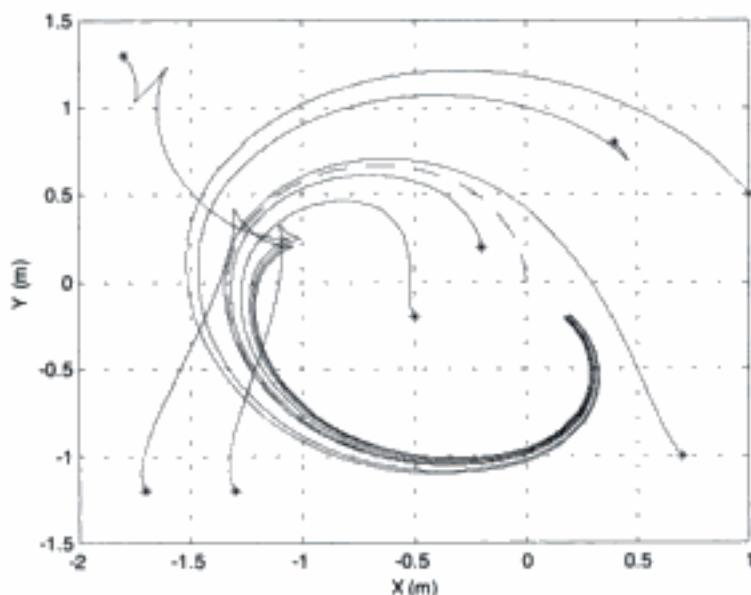
$$k_1 = 1; \quad k_2 = 2; \quad k_3 = 4$$

En la figura 9.12 se muestran los resultados de las velocidades lineal y angular obtenidas (línea continua) junto con la velocidades lineal y angular de referencia (línea discontinua).

En la figura 9.13 se representan la trayectoria seguida por el robot y la de referencia. Los puntos de partida se marcan mediante asteriscos. Se representa una flecha por



**Figura 9.13:** Representación de la trayectoria y orientación de la referencia (flechas negras), junto con las del robot (flechas blancas).

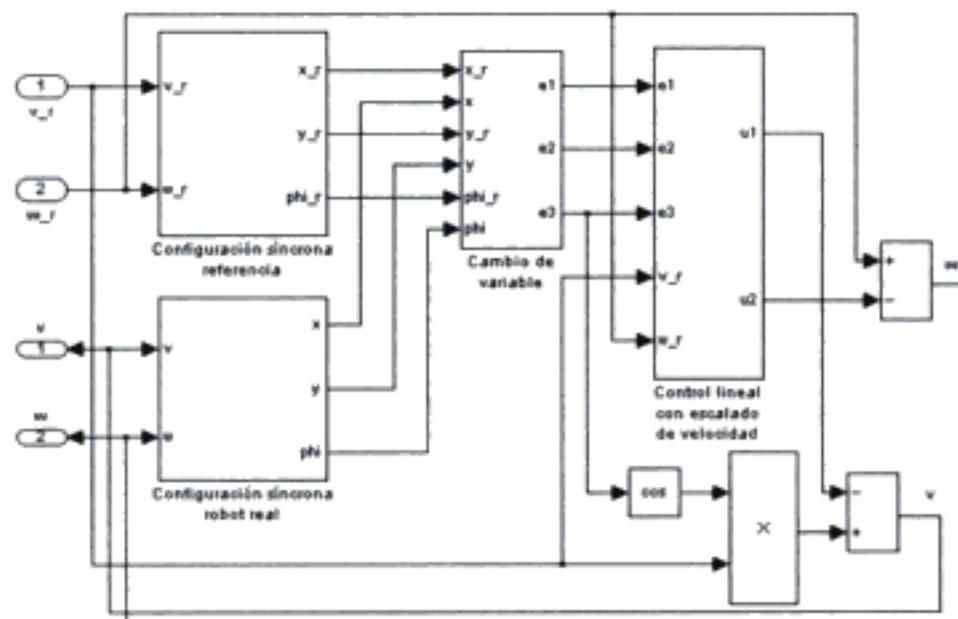


**Figura 9.14:** Trayectorias correspondientes a diversas coordenadas iniciales (todas con una misma orientación inicial de  $\pi/6$  rad).

cada segundo de simulación, por lo que, cuando en los últimos 10 segundos la velocidad lineal desciende a 0,1 m/s, las flechas aparecen más juntas.

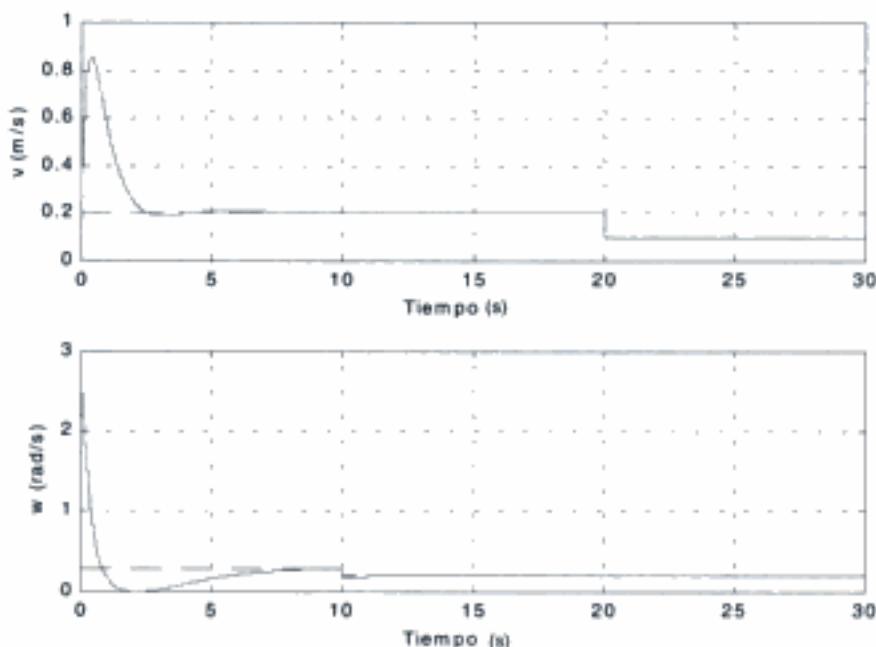
Por último, en la figura 9.14, se muestran las trayectorias seguidas para diferentes puntos iniciales.

Considérese ahora la aplicación de la técnica de “escalado de velocidad” empleando para ello las ganancias de control (9.77). En la figura 9.15 se presenta el diagrama Simulink correspondiente.

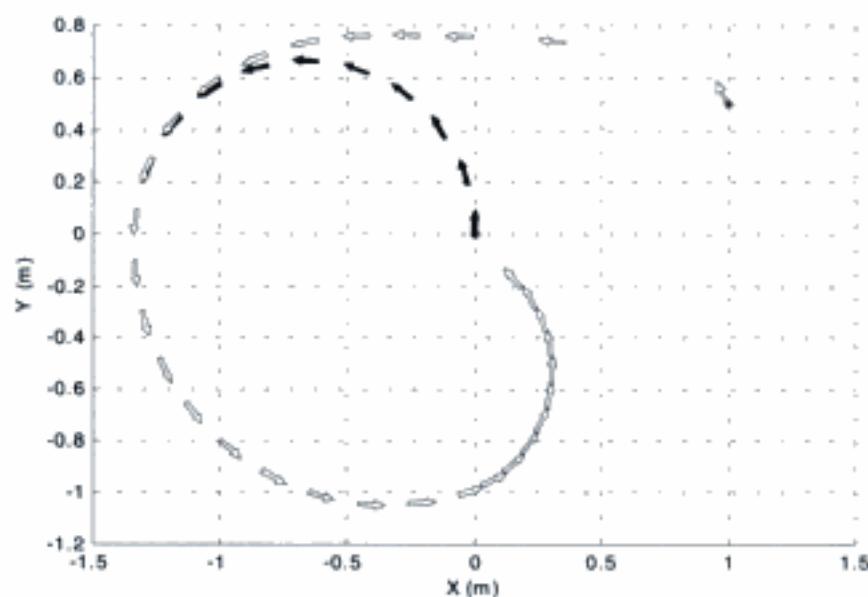


**Figura 9.15:** Esquema Simulink para simular el seguimiento de trayectorias con control lineal y “escalado de velocidad”.

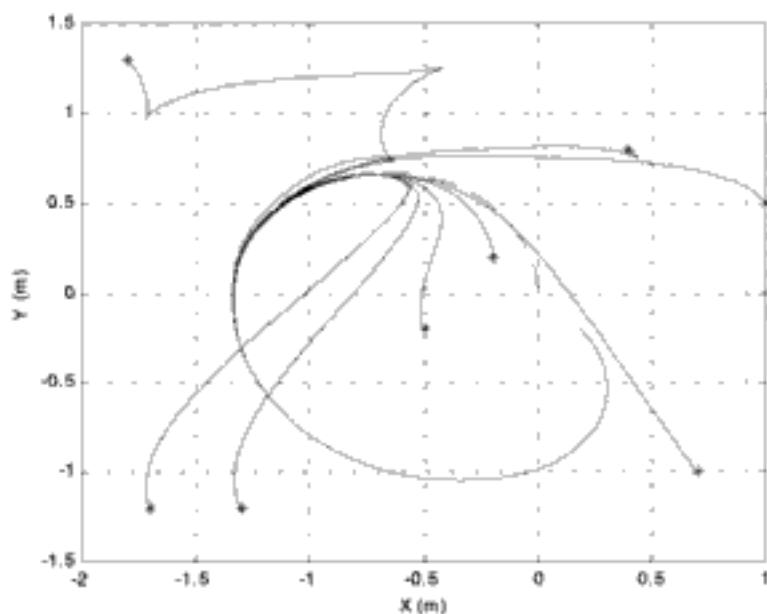
En las figuras 9.16 y 9.17 se muestran los resultados obtenidos de la simulación, con  $\beta = 13,75$  y  $\delta = 0,9$  y el mismo punto inicial que en el caso anterior. Finalmente, en la figura 9.18, se presentan los resultados desde los mismos puntos iniciales que en la figura 9.14. Obsérvese como el empleo del escalado de velocidad mejora claramente los resultados obtenidos.



**Figura 9.16:** Velocidades lineal y angular obtenidas con la técnica de “escalado de velocidad”.



**Figura 9.17:** Posiciones y orientaciones del móvil (flechas blancas) que se obtienen con la técnica de “escalado de velocidad”.



**Figura 9.18:** Trayectorias correspondientes a diversas coordenadas iniciales (todas con la misma orientación inicial de  $\pi/6$  rad) para el caso de “escalado de velocidad”.

#### 9.4.2.3 Ley de control no lineal

Es posible también diseñar leyes de control no lineales particulares, cuya estabilidad se demuestra mediante el método de Liapunov.

En Canudas y otros (1997) se propone la ley:

$$\begin{aligned} u_1 &= -k_1(v_{ref}, \omega_{ref}) e_1 \\ u_2 &= -k_2 v_{ref} \frac{\operatorname{sen} e_3}{e_3} e_2 - k_3(v_{ref}, \omega_{ref}) e_3 \end{aligned} \quad (9.78)$$

siendo  $k_2$  una constante positiva y  $k_1(\cdot)$ ,  $k_3(\cdot)$  funciones continuas estrictamente positivas en  $\mathfrak{R} \times \mathfrak{R} - (0, 0)$ .

Obsérvese la similitud de (9.78) con la realimentación lineal (9.70). Si se eligen  $k_1(v_{ref}, \omega_{ref}) = k_3(v_{ref}, \omega_{ref})$  según (9.77), la ley de control se comporta de forma análoga cerca del origen.

Para demostrar que la ley de control (9.78) estabiliza global y asintóticamente el sistema en el origen  $e = 0$ , se considera la función de Liapunov:

$$V(e) = \frac{k_2}{2}(e_1^2 + e_2^2) + \frac{e_3^2}{2} \quad (9.79)$$

Derivando en (9.79) con respecto al tiempo, se obtiene:

$$V' = k_2(e_1 e'_1 + e_2 e'_2) + e_3 e'_3$$

Sustituyendo las expresiones de  $e'_1$ ,  $e'_2$  y  $e'_3$  de (9.67):

$$V' = k_2[e_1(\omega e_2 + u_1) + e_2(-\omega e_1 + v_{ref} \operatorname{sen} e_3)] + e_3 u_2 \quad (9.80)$$

Sustituyendo ahora las expresiones de  $u_1$  y  $u_2$  definidas en (9.78) y en la (9.80) y simplificando, se obtiene:

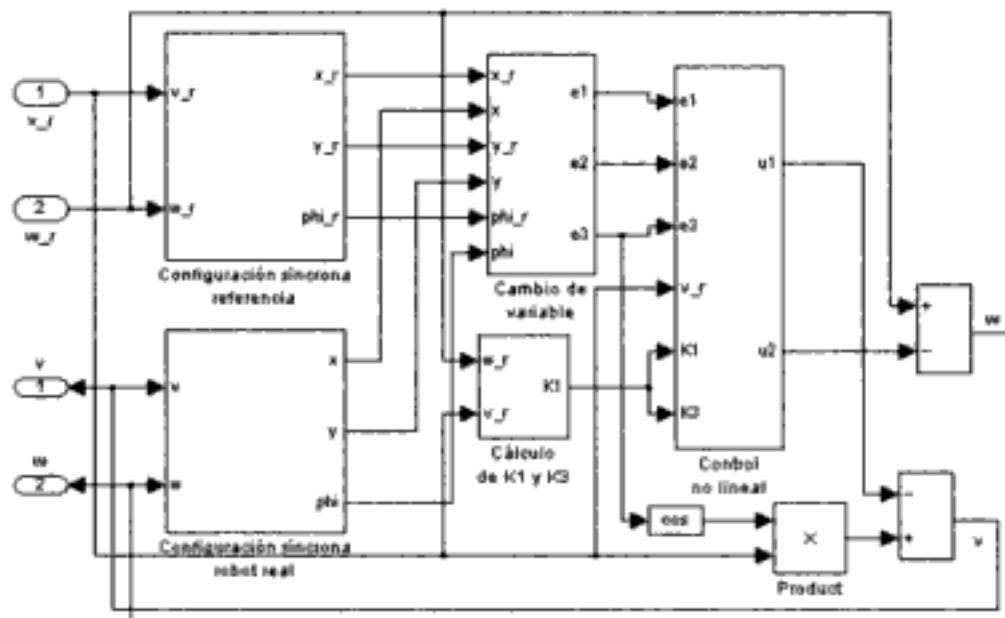
$$V' = -k_1 k_2 e_1^2 - k_3 e_3^2 \quad (9.81)$$

Aplicando estas expresiones en Canudas y otros (1997) se demuestra que, con la ley de control (9.78),  $e_1$ ,  $e_3$ ,  $v_{ref}$ ,  $e_2$  y  $V$  tienden a cero.

#### Ejemplo 9.4

Se pretende aplicar un controlador no lineal en el caso que se presenta en el ejemplo 9.3. Para ello se utilizarán las leyes de control (9.78) con:

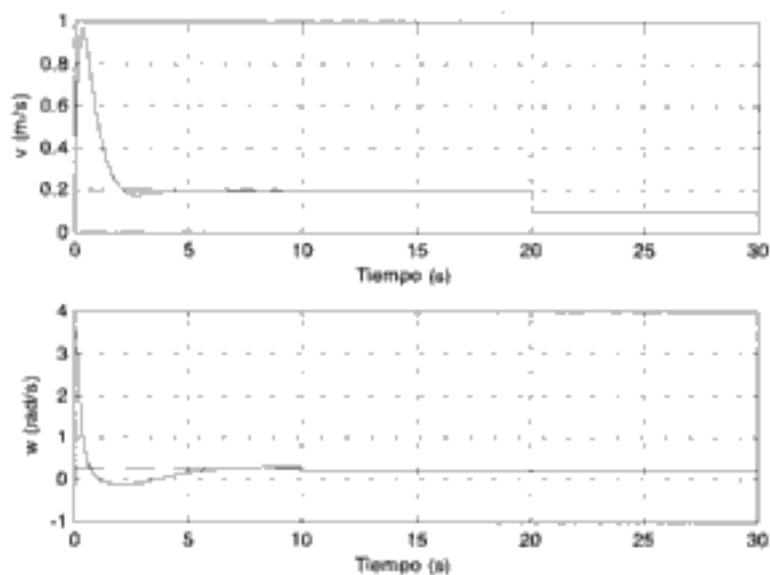
$$k_1(v_{ref}, \omega_{ref}) = k_3(v_{ref}, \omega_{ref}) = 2\delta(\omega_{ref}^2 + \beta v_{ref}^2)^{1/2} \quad (9.82)$$



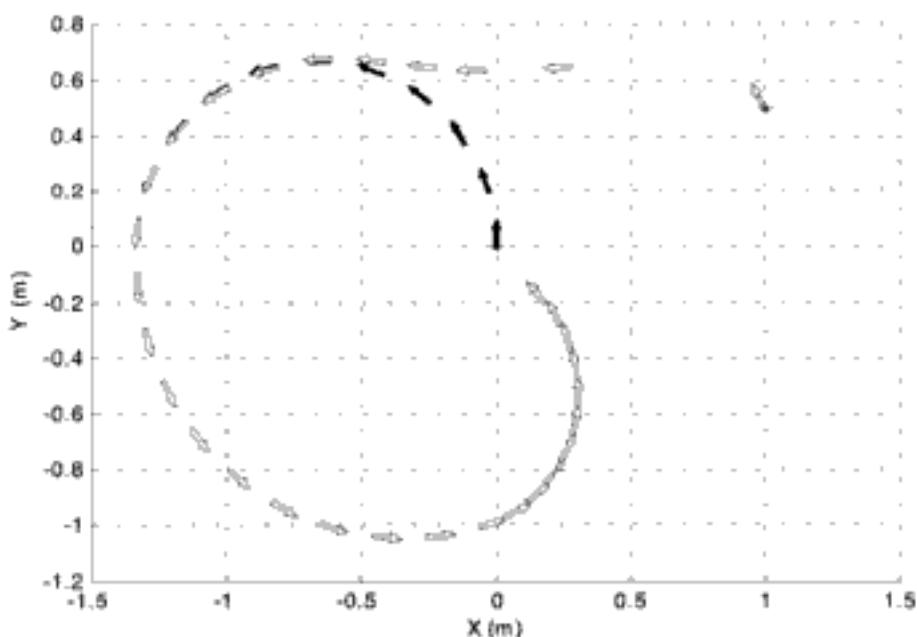
**Figura 9.19:** Esquema Simulink que se emplea para simular el control no lineal en el seguimiento de trayectorias.

El diagrama Simulink que se emplea para la simulación de dicho controlador es el mostrado en la figura 9.19. Los resultados que se representan en las figuras 9.20, 9.21 y 9.22, son los correspondientes a la siguiente elección de constantes:

$$\beta = 13,75; \quad \delta = 0,9; \quad k_2 = 19$$



**Figura 9.20:** Velocidades lineal y angular que se obtienen mediante el control no lineal.

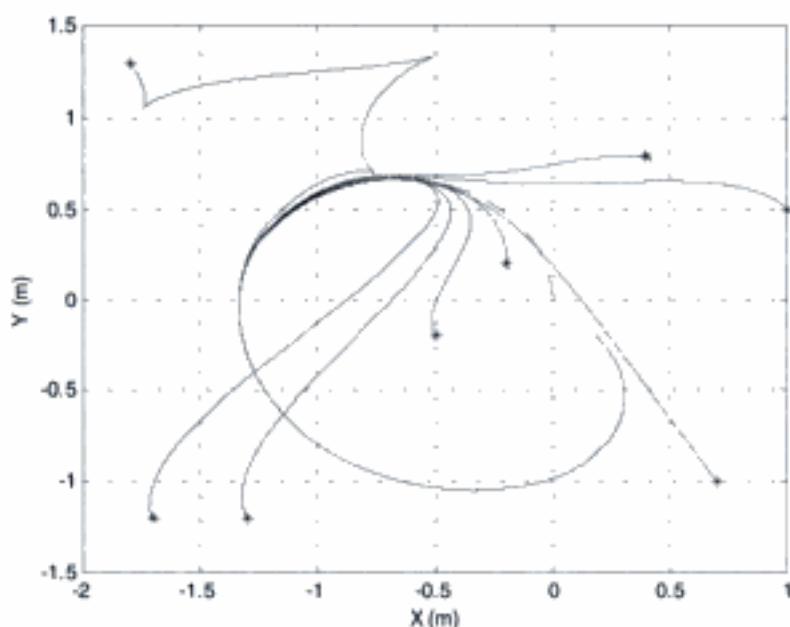


**Figura 9.21:** Posiciones y orientaciones del móvil (flechas blancas) al partir de las coordenadas  $(1, 0.5)$  con orientación  $\pi/6$  rad y aplicársele la ley de control no lineal.

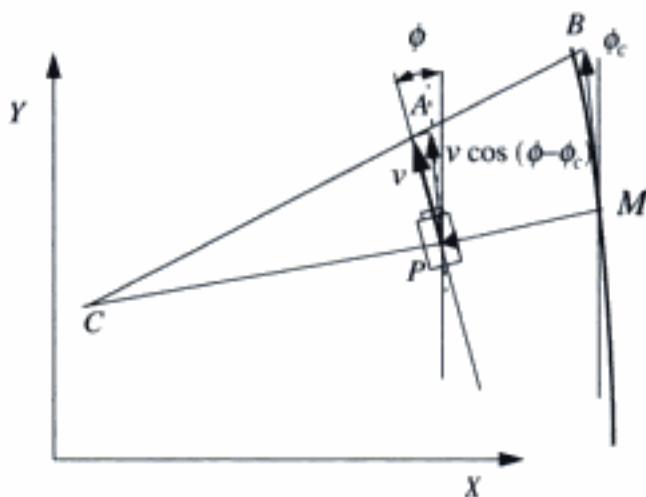
### 9.4.3 Seguimiento de caminos

#### 9.4.3.1 Planteamiento del problema

Considérese el problema de seguimiento de un camino conocido que se ilustra en la figura 9.23, en la cual  $M$  es la proyección del punto de guiado  $P$  sobre el camino. Se emplea un sistema móvil de coordenadas que tiene centro en  $M$  y ejes según la tan-



**Figura 9.22:** Trayectorias obtenidas empleando control no lineal cuando se varía el punto de partida del móvil manteniendo la misma orientación inicial ( $\pi/6$  rad).



**Figura 9.23:** Seguimiento de caminos utilizando teoría de control.

gente y la normal al camino en este punto. Sea  $\xi$  la distancia con signo entre  $M$  y  $P$ , o coordenada según la normal al camino en  $M$ . En este punto, la curvatura del camino es  $\gamma_c$  y el radio de curvatura  $R_c = 1/\gamma_c$  (distancia  $CM$  en la figura).

Sea  $\phi$  el ángulo de orientación del vehículo,  $\phi_c$  el ángulo de orientación del camino,  $\bar{\phi} = \phi - \phi_c$ , y  $V \cos \bar{\phi}$  la proyección de la velocidad lineal del vehículo sobre la tangente al camino en  $M$ . Si  $ds$  es la distancia recorrida sobre el camino desde un punto de referencia, aplicando semejanza de los triángulos  $CAP$  y  $CBM$  en la figura 9.23, se obtiene la siguiente ecuación, que iguala el movimiento que se realiza sobre la tangente al camino en  $M$  y el del vehículo:

$$\frac{s'}{R_c} = \frac{v \cos \tilde{\phi}}{R_c - \xi} \quad (9.83)$$

siendo  $R_c$  el radio de curvatura del camino en el punto  $M$ . La curvatura en dicho punto viene dada por  $\gamma_c(s) = 1/R_c$ . Por tanto, puede escribirse:

$$s' = \frac{1}{\gamma_c(s)} \frac{v \cos \tilde{\phi}}{\frac{1}{\gamma_c(s)} - \xi} \quad (9.84)$$

Por otra parte, si se considera el movimiento del vehículo en la dirección normal al camino en  $M$ , puede escribirse:

$$\xi' = v \operatorname{sen} \phi \quad (9.85)$$

Finalmente, derivando en  $\tilde{\phi} = \phi - \phi_c$ , se obtiene:

$$\tilde{\phi}' = \omega - \frac{d\phi_c}{dt} \quad (9.86)$$

siendo  $\omega$  la velocidad angular del vehículo. Teniendo en cuenta que  $d\phi_c = \gamma_c(s)ds$ , la expresión (9.86) puede escribirse como:

$$\tilde{\phi}' = \omega - \gamma_c(s)s'$$

Por tanto, se llega a:

$$\begin{aligned} s' &= v \cos \tilde{\phi} \frac{1}{1 - \gamma_c(s)\xi} \\ \xi' &= v \operatorname{sen} \tilde{\phi} \\ \tilde{\phi}' &= \omega - v \cos(\tilde{\phi}) \frac{\gamma_c(s)}{1 - \gamma_c(s)\xi} \end{aligned} \quad (9.87)$$

Supuesto conocido un camino, y la velocidad del vehículo  $v(t)$ , estando acotadas tanto la velocidad como su derivada  $v'(t)$ , se trata de encontrar una ley de control, suficientemente suave:

$$\omega = k[s, \xi, \tilde{\phi}, v(t)] \quad (9.88)$$

tal que el vehículo siga el camino o, lo que es lo mismo, que se cumpla:

$$\lim_{t \rightarrow \infty} \xi(t) = 0 \quad \lim_{t \rightarrow \infty} \tilde{\phi}(t) = 0 \quad (9.89)$$

Obsérvese que, en este caso, se trata de anular los errores de seguimiento de dos variables (distancia al camino  $\xi$  y diferencia de orientación  $\tilde{\phi}$ ) mientras que en el problema del seguimiento de trayectorias se involucraban tres. Por otra parte, ahora se

considera una única variable de control  $\omega$ , mientras que en seguimiento de trayectorias se involucraban dos. En efecto, en el seguimiento de caminos se considera exclusivamente la separación del robot del camino. Sin embargo, la posición del robot a lo largo del camino (caracterizada por el valor de la distancia con signo sobre el camino  $s$ ) depende de la velocidad  $v$ , la cual no se utiliza como variable de control.

Para resolver este problema se considera la variable auxiliar:

$$u = \omega - v \cos(\tilde{\phi}) \frac{\gamma_c(s)}{1 - \gamma_c(s)\xi} \quad (9.90)$$

con lo cual el sistema (9.87) puede escribirse como:

$$\begin{aligned} s' &= v \cos \tilde{\phi} \frac{1}{1 - \gamma_c(s)\xi} \\ \xi' &= v \sin \tilde{\phi} \\ \tilde{\phi}' &= u \end{aligned} \quad (9.91)$$

#### 9.4.3.2 Ley de control lineal

Se considera, en primer lugar, un método de control basado en la linealización de las ecuaciones (9.91) según la tangente en el punto ( $\xi = 0, \tilde{\phi} = 0$ ). Obsérvese que, en este caso, para controlar lateralmente el vehículo sólo se necesita considerar la segunda y tercera ecuación de (9.91). Si  $\tilde{\phi}$  es suficientemente pequeño puede escribirse:

$$\begin{aligned} \xi' &= v\tilde{\phi} \\ \tilde{\phi}' &= u \end{aligned} \quad (9.92)$$

Este sistema es controlable y estabilizable utilizando realimentación lineal del vector de estado. En Canudas y otros (1997) la ley de control se escribe en la forma:

$$u = -k_1 v \xi - k_2 |v| \tilde{\phi} \quad (9.93)$$

siendo  $k_1 > 0, k_2 > 0$ . La ecuación característica en bucle cerrado correspondiente a la variable de estado  $\xi$  es:

$$\xi'' + k_2 |v| \xi' + k_1 v^2 \xi = 0 \quad (9.94)$$

Este comportamiento dinámico puede estudiarse escalando en velocidad mediante la variable:

$$\eta' = \frac{\partial \xi}{\partial \rho} \quad (9.95)$$

siendo  $\rho$  la distancia recorrida:

$$\rho = \int_0^t |v| d\tau \quad (9.96)$$

En este caso, se tiene:

$$\eta'' + k_2 \eta' + k_1 \eta = 0 \quad (9.97)$$

Las constantes  $k_1$ ,  $k_2$  se eligen para conseguir un comportamiento dinámico determinado. Se emplean las expresiones:

$$\begin{aligned} k_1 &= \omega_n^2 \\ k_2 &= 2\delta b \end{aligned} \quad (9.98)$$

siendo  $b$  una frecuencia natural no amortiguada relacionada con la "distancia de seguimiento" y  $\delta$  un coeficiente de amortiguamiento (amortiguamiento crítico cuando  $\delta = 1$ ).

### Ejemplo 9.5

Considérese un robot situado inicialmente en las coordenadas  $(0, 0)$  con una orientación de 0 rad. Se aplica la ley de control (9.93) con el objetivo de seguir trayectorias rectilíneas y parabólicas. En la figura 9.24 se muestra el diagrama Simulink correspondiente.

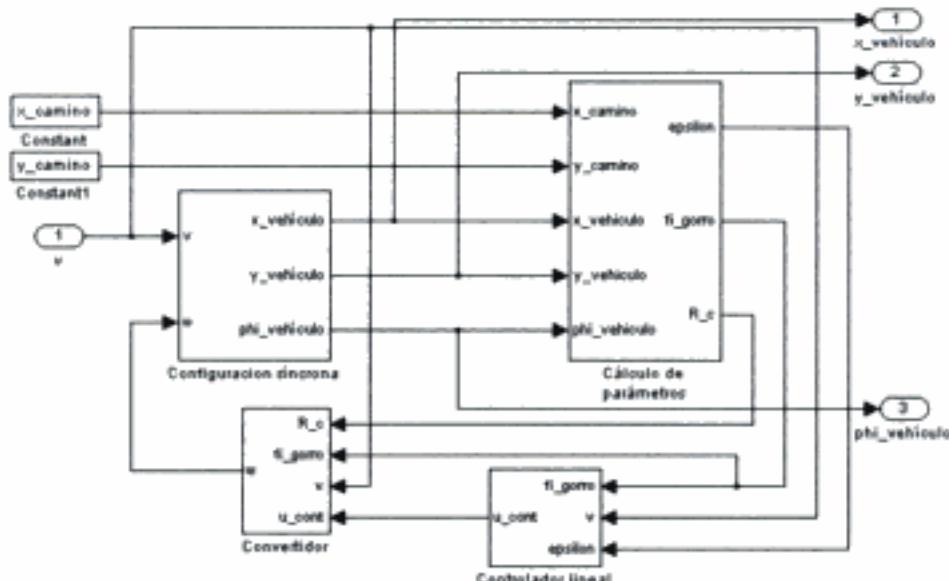
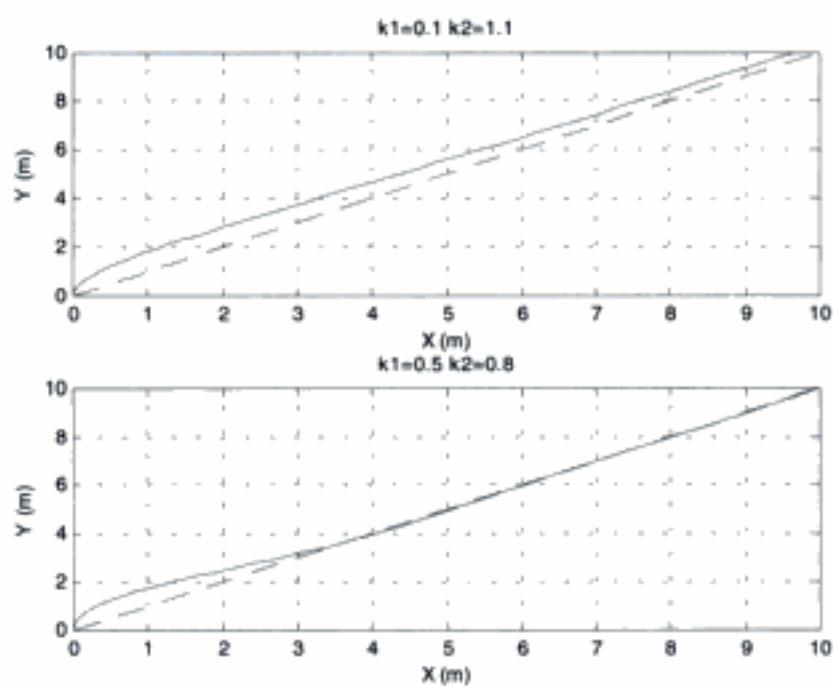


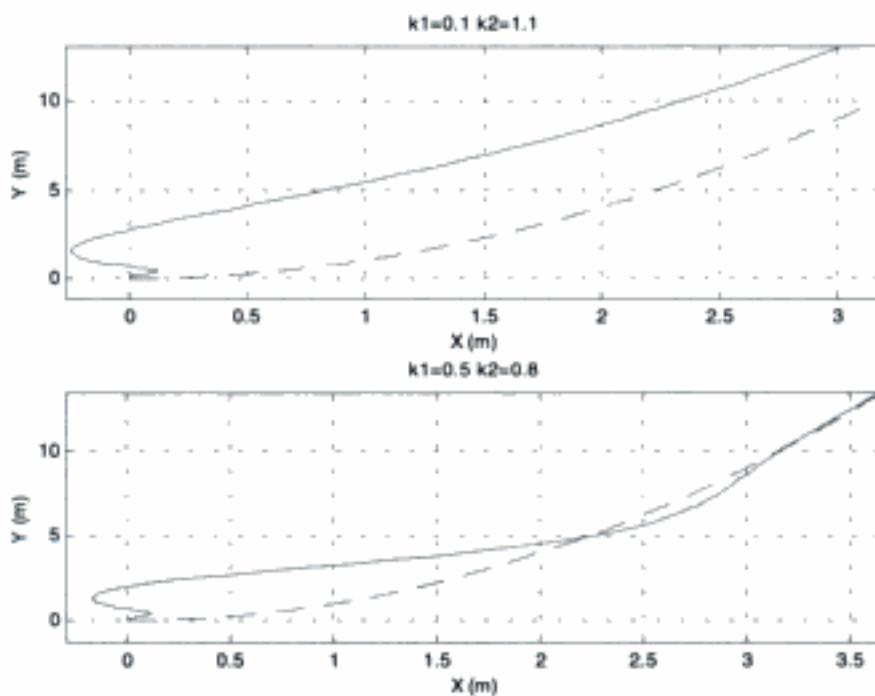
Figura 9.24: Diagrama Simulink para seguimiento de trayectorias mediante control lineal.

Se supone que el robot se desplaza con una velocidad constante de 1 m/s. En la figura 9.25 se muestran los resultados obtenidos en el seguimiento de una trayectoria rectilínea para dos pares de valores diferentes de  $k_1$  y  $k_2$ .

Asimismo, en la figura 9.26 se muestran los resultados en el seguimiento de una trayectoria parabólica para los mismos valores de las constantes.



**Figura 9.25:** Seguimiento de una trayectoria rectilínea mediante control lineal con  $v = 1 \text{ m/s}$ .



**Figura 9.26:** Seguimiento de trayectoria parabólica mediante control lineal con  $v = 1 \text{ m/s}$ .

### 9.4.3.3 Ley de control no lineal

Considérese otra vez el sistema sin linealizar (9.91) y la ley de control no lineal:

$$u = -k_1 v \xi \frac{\operatorname{sen} \tilde{\phi}}{\tilde{\phi}} - k(v) \tilde{\phi} \quad (9.99)$$

siendo  $k_1$  una constante positiva, y  $k(v)$ , una función estrictamente positiva cuando  $v \neq 0$ . La elección de esta ley de control está determinada por la propiedad de conseguir la estabilidad que se estudia más abajo.

Puede observarse también que la ley de control (9.99) proporciona resultados similares a la lineal (9.93) en el punto ( $\xi = 0, \phi = 0$ ), cuando  $k(v) = k_2|v|$ , con  $k_1$  y  $k_2$  según la (9.98).

En Canudas y otros (1997) se demuestra que si:

$$\lim_{t \rightarrow \infty} v(t) \neq 0 \quad (9.100)$$

y el vehículo robótico en su posición inicial satisface la restricción:

$$\xi(0) + \frac{1}{k_1} \tilde{\phi}(0)^2 < \frac{1}{\limsup[\gamma_c^2(s)]} \quad (9.101)$$

siendo  $\gamma_c(s)$  la curvatura del camino en el punto  $M$  (figura 9.23), entonces la ley de control (9.99) hace que el sistema sea asintóticamente estable en el punto ( $\xi = 0, \tilde{\phi} = 0$ ).

Para ello se emplea la función de Liapunov:

$$V = k_2 \frac{\xi^2}{2} + \frac{\tilde{\phi}^2}{2} \quad (9.102)$$

cuya derivada es:

$$V' = k_1 \xi \xi' + \tilde{\phi} \phi' = k_1 v \xi \operatorname{sen} \tilde{\phi} + \tilde{\phi} u = -k(v) \tilde{\phi}^2$$

### 9.4.4 Seguimiento de caminos utilizando el modelo de la bicicleta

En todo lo anterior se ha supuesto que las variables de control eran las velocidades lineales y angulares del vehículo. Como se ponía de manifiesto en el capítulo 4, en algunas configuraciones la relación entre las variables de control reales y dichas velocidades no es directa e involucra a las propias variables del modelo. Así, por ejemplo, para establecer la velocidad angular es necesario actuar sobre la dirección del vehículo y la velocidad angular que alcance dependerá también de la velocidad lineal. Por otra parte, el efecto de la dinámica del sistema de actuación puede ser importante, por lo cual debe tenerse en cuenta en el modelo (ver capítulo 5).

Considérese el modelo de locomoción denominado de la bicicleta:

$$x' = -v \operatorname{sen} \phi \quad (9.103)$$

$$y' = v \cos \phi \quad (9.104)$$

$$\phi' = v\gamma \quad (9.105)$$

siendo  $\gamma$  la curvatura del vehículo y  $v$  la velocidad lineal. La curvatura del vehículo viene dada por el ángulo de dirección  $\alpha$  (ver capítulo 4). Así, si  $l$  es la longitud del vehículo y el sistema de referencia está en el eje trasero, se tendrá  $\gamma = (\operatorname{tg} \alpha)/l$ .

Se estudia ahora el problema del seguimiento de un camino previamente definido. Este problema suele tratarse considerando la velocidad lineal  $v$  constante. Nótese que, para linealizar el sistema, debe suponerse además que el ángulo  $\phi$  es suficientemente pequeño, lo cual en coordenadas globales ( $X, Y$ ) sólo es cierto si inicialmente el robot está en el origen y el camino se dirige aproximadamente en la dirección del eje  $Y$ , y con baja curvatura.

Considérese ahora el modelo escrito en un sistema de coordenadas móvil asociado al vehículo o al camino que se pretende seguir (Fenton y otros, 1976; Cormier y Fenton, 1980). En particular, puede tomarse un sistema de coordenadas con un eje  $\eta$  paralelo a la tangente al camino en un punto de referencia y estudiar el comportamiento lateral según el eje  $\xi$  perpendicular al camino, como se muestra en la figura 9.27. En este caso, se trata de minimizar las coordenadas de los puntos del camino que se pretende seguir, con respecto al sistema de referencia asociado al vehículo. El modelo viene dado por:

$$\xi' = -v \operatorname{sen}(\phi) \quad (9.106)$$

$$\eta' = v \cos(\phi) \quad (9.107)$$

$$\phi' = \gamma v \quad (9.108)$$

siendo

$$\xi = d\xi/dt, \eta = d\eta/dt, \phi = d\phi/dt$$

En el sistema móvil de coordenadas, el modelo puede linealizarse fácilmente si se consideran pequeñas variaciones del ángulo en el intervalo de control. En efecto, suponiendo que el ángulo de orientación del vehículo en el sistema mencionado ante-

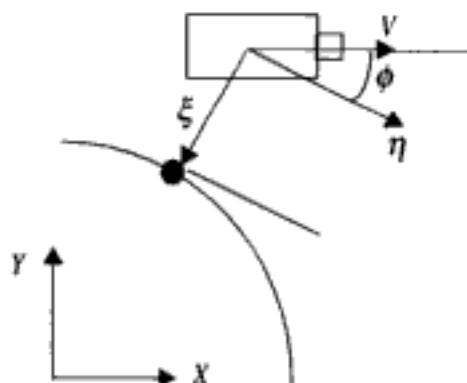


Figura 9.27: Sistemas de coordenadas en el seguimiento de caminos.

riamente es suficientemente pequeño (desviación en ángulo con respecto al camino que se pretende seguir es pequeña), es posible aproximar el seno por el ángulo, con lo cual el movimiento lateral, según la coordenada  $\xi$ , con respecto al camino que se pretende seguir, viene dado por:

$$\xi' = -v\phi \quad (9.109)$$

$$\phi' = \gamma v \quad (9.110)$$

que puede considerarse como un modelo lineal del seguimiento, válido como aproximación en torno al punto de trabajo.

A estas ecuaciones hay que añadir la correspondiente a la dinámica del actuador de dirección (ver capítulo 5) que relaciona la curvatura requerida  $u_z$ ,  $\gamma$ , con la curvatura real del vehículo. Es frecuente emplear un sistema de primer o segundo orden. En el primer caso, se tendría:

$$\gamma' = -\frac{\gamma}{\tau} + \frac{1}{\tau} u$$

siendo  $\tau$  la constante de tiempo.

Las ecuaciones anteriores pueden ponerse en la forma:

$$x' = Ax + Bu \quad (9.111)$$

siendo

$$A = \begin{bmatrix} 0 & -v & 0 \\ 0 & 0 & v \\ 0 & 0 & -\frac{1}{\tau} \end{bmatrix} \quad (9.112)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{bmatrix} \quad (9.113)$$

Si se aplica la ley de control por realimentación lineal del vector de estados:

$$u = -k_1 x_1 - k_2 x_2 - k_3 x_3 \quad (9.114)$$

se tendrá:

$$x' = \begin{bmatrix} 0 & -v & 0 \\ 0 & 0 & v \\ \frac{k_1}{\tau} & \frac{k_2}{\tau} & -\left(\frac{k_3 + 1}{\tau}\right) \end{bmatrix} x \quad (9.115)$$

Por tanto, la ecuación característica será:

$$\tau s^3 + (k_3 + 1)s^2 + \nu k_2 s - \nu^2 k_1 = 0 \quad (9.116)$$

El comportamiento dinámico vendrá dado por las ganancias de realimentación. El sistema será estable si se cumplen las condiciones:

$$\frac{k_3 + 1}{\tau} > 0; \quad \frac{\nu k_2}{\tau} > 0; \quad \frac{k_1}{\tau} < 0; \quad \frac{(1 + k_3)k_2}{\nu \tau} > -k_1 \quad (9.117)$$

En Aracil, Heredia y Ollero (2000) se consideran también controladores no lineales y se estudia la estabilidad global.

## 9.5 SEGUIMIENTO PREDICTIVO GENERALIZADO

Es un método basado en la aplicación al seguimiento de caminos de la teoría del control predictivo generalizado (Ollero y Amidi, 1991). En particular, se emplea el modelo de la bicicleta de la sección anterior.

Adicionalmente, es posible considerar la influencia de los ruidos en la descripción de entrada/salida del sistema. Formulando el problema en tiempo discreto, se llega al modelo CARIMA (Controlled Autoregressive and Moving Average) utilizado en la formulación original del problema del control predictivo generalizado (Clark y otros, 1987), que puede escribirse como:

$$A(q^{-1})y(k) = B(q^{-1})u(k-1) + \zeta(k)/\Delta \quad (9.118)$$

siendo  $u(k)$  la señal de control,  $y(k)$  la salida (variable que se pretende controlar) y  $\zeta(k)$  una secuencia aleatoria no correlacionada en el instante  $k$ . El operador  $q^{-1}$  corresponde a un retardo unitario y  $\Delta$  es el operador definido mediante  $\Delta = 1 - q^{-1}$ .

$A(q^{-1})$ ,  $B(q^{-1})$  son polinomios definidos con el operador retardo:

$$A(q^{-1}) = 1 + a_1 q^{-1} + \dots + a_{na} q^{-n_a} \quad (9.119)$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \dots + b_{nb} q^{-n_b} \quad (9.120)$$

La entrada de control  $u = \gamma$ , es la curvatura requerida, la salida  $y = \xi$  la distancia del camino al vehículo, y  $\zeta$  representa a los ruidos.

El control predictivo está basado en la minimización de la función:

$$J(N_1, N_2) = E \left\{ \sum_{j=N_1}^{N_2} [\hat{y}(k+j) - y_d(k+j)]^2 + \sum_{j=N_1}^{N_2} \lambda(j)[\Delta u(k+j-1)]^2 \right\} \quad (9.121)$$

donde:

- $E \{ \cdot \}$  representa la esperanza matemática.
- $N_1$  y  $N_2$  indican, respectivamente, el punto más cercano y el más lejano en el camino que se pretende seguir (ventana de seguimiento sobre el camino).
- $\hat{y}(k+j)$  e  $y_d(k+j)$  son, respectivamente, la predicción de la salida y los valores deseados de la salida en el intervalo  $[N_1, N_2]$ .
- $\lambda(j)$  es una secuencia de ponderación del control que se emplea para evitar excesivos cambios en el sistema de direccionamiento del vehículo.

Por tanto, lo que se pretende es la minimización de la distancia al camino en coordenadas locales (teniendo en cuenta la orientación del vehículo). Si el camino está expresado en coordenadas globales, es necesario transformarlo a este sistema de coordenadas local. De esta forma, si  $(x^*, y^*)$  y  $(x_d^*, y_d^*)$  representan, respectivamente, las coordenadas globales del vehículo y del punto de referencia sobre el camino, se tendrá:

$$\eta(k) = [x_d^*(k) - x^*(k)] \cos \phi_d(0) - [y_d^*(k) - y^*(k)] \sin \phi_d(0) \quad (9.122)$$

$$\xi(k) = [x_d^*(k) - x^*(k)] \sin \phi_d(0) + [y_d^*(k) - y^*(k)] \cos \phi_d(0) \quad (9.123)$$

siendo  $\phi_d(0)$  la orientación del camino en el punto de referencia, con respecto al vehículo, en el instante  $k = 0$ .

Obsérvese que el objetivo de la minimización del índice es hacer que las futuras salidas  $y(k+j)$  estén cerca de las salidas deseadas  $y_d(k+j)$ , teniendo en cuenta la actividad de control requerida. Si  $N_1 = 1$  y  $N_2 = N$ , el vector de predicción:

$$\hat{y} = [\hat{y}(k+1|k) \ \hat{y}(k+2|k) \ \dots \ \hat{y}(k+N|k)]^T \quad (9.124)$$

viene dado por:

$$\hat{y} = G\Delta u + f \quad (9.125)$$

donde  $\Delta u = [\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+N-1)]^T$  es el vector de incrementos de control proyectados (incrementos a la dirección del vehículo),  $G$  es una matriz  $N \times N$  triangular inferior, y:

$$f = [f(k+1), f(k+2), \dots, f(k+N)]^T \quad (9.126)$$

son las predicciones de la salida suponiendo que los futuros incrementos de control son todos nulos.

La ley de control predictivo generalizado viene dada por (Clarke y otros, 1987):

$$\Delta u = (G^T G + \lambda I)^{-1} G^T (y_d - f) \quad (9.127)$$

siendo  $y_d = [y_d(k+1), y_d(k+2), \dots, y_d(k+N)]^T$  la secuencia deseada para la variable de salida.

Si se supone que después de un cierto número de pasos  $NU < N_2$ , los incrementos de control proyectados son cero:

$$\Delta u(k+j-1) = 0, \quad j > NU \quad (9.128)$$

la ley de control puede escribirse como:

$$\Delta u = (G_1^T G_1 + \lambda I)^{-1} G_1^T (y_d - f) \quad (9.129)$$

siendo

$$G_1 = \begin{bmatrix} g_0 & 0 & \dots & 0 \\ g_1 & g_0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ g_{N-1} & g_{N-2} & \dots & g_{N-NU} \end{bmatrix} \quad (9.130)$$

La matriz  $(G_1^T G_1 + \lambda I)$  tiene como dimensiones  $NU \times NU$ . Por tanto, para el caso particular en que se suponga que  $NU = 1$ , sólo se considera un incremento en la variable de control  $\Delta u(k)$  y no es necesario invertir ninguna matriz para obtener la ley de control.

Los coeficientes de la matriz  $G_1$  en las expresiones anteriores, o de  $G$  en la más general, pueden obtenerse a partir de los polinomios  $G_j$  definidos mediante:

$$G_j(q^{-1}) = E_j(q^{-1})B(q^{-1}) \quad (9.131)$$

expresión en la cual  $E_j(q^{-1})$  se obtiene resolviendo de forma recurrente la ecuación diofántica:

$$1 + E_j(q^{-1})A\Delta + q^{-j}F_jq^{-1} \quad (9.132)$$

donde  $E_j(q^{-1})$  y  $F_j(q^{-1})$  son polinomios que se determinan únicamente conociendo  $A(q^{-1})$  y  $j$ .

En todo lo anterior se ha supuesto como única variable de salida la distancia al camino según uno de los ejes de coordenadas. Es posible emplear también como salidas la orientación del vehículo y su curvatura. Asimismo, el problema puede formularse con más de una variable de control, tal como, por ejemplo, las velocidades lineales y las rotacionales (capítulo 4). Se tiene así una formulación multivariable del problema del control predictivo generalizado.

Sean  $m$  y  $p$  el número de salidas y variables de control, respectivamente. En este caso,  $A(q^{-1})$  y  $B(q^{-1})$  son matrices polinomiales de  $m \times m$  y  $m \times p$ , respectivamente:

$$A(q^{-1}) = I_{m \times m} + A_1q^{-1} + \dots + A_nq^{-n} \quad (9.133)$$

$$B(q^{-1}) = B_0 + B_1q^{-1} + \dots + B_nq^{-n} \quad (9.134)$$

La ecuación diofántica es ahora una ecuación matricial  $m \times m$  que puede también resolverse recursivamente, y los polinomios  $G_j$  en (9.131) son ahora matrices polinomiales dadas por:

$$G_j(q^{-1}) = (\varepsilon_0 + \varepsilon_1 q^{-1} + \dots + \varepsilon_{nb} q^{-(j-1)})(B_0 + B_1 q^{-1} + \dots + B_{nb} q^{-nb}) \quad (9.135)$$

siendo  $\varepsilon_i$  matrices  $m \times m$ . El método descrito anteriormente puede extenderse al caso multivariable, calculándose los vectores proyectados  $\Delta u_j (j = 1 \dots p)$  para cada variable de control a partir de los vectores de predicciones  $f_i (i = 1 \dots m)$  de cada variable de salida. Este método se ha aplicado  $m = 3$  salidas (distancia al camino, error de orientación y error de curvatura), y una variable de control (curvatura requerida).

En la figura 9.28 se muestra la implantación de este controlador. Nótese que la ley de control (9.130) puede escribirse en la forma:

$$\Delta u = M(y_d - f) \quad (9.136)$$

siendo  $M$  la matriz de ganancias:

$$M = (G^T G + \lambda I)^{-1} G^T \quad (9.137)$$

o bien, si se emplea (9.131):

$$\Delta u = M_1(y_d - f) \quad (9.138)$$

siendo

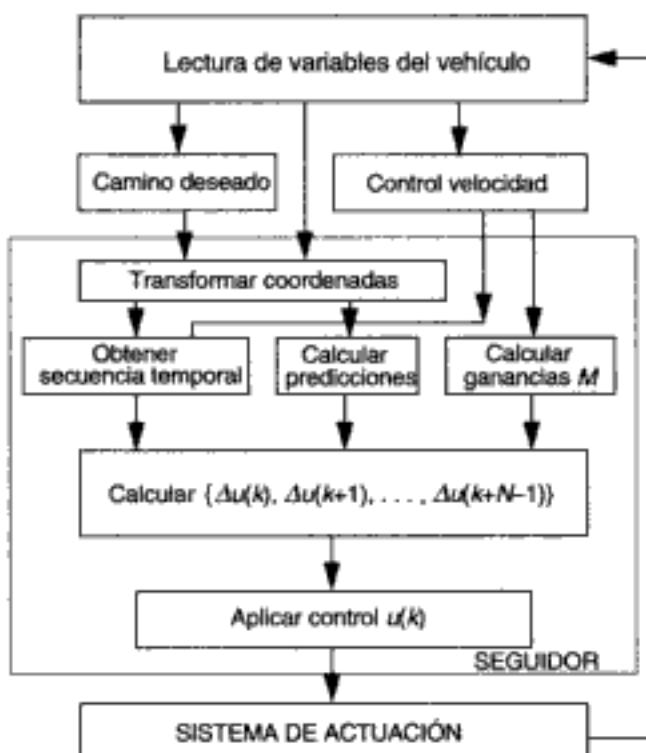
$$M_1 = (G_1^T G_1 + \lambda I)^{-1} G_1^T \quad (9.139)$$

El procedimiento para calcular estas ganancias es el siguiente (Ollero y Amidi, 1991):

- Calcular los parámetros del modelo linealizado  $a_i (i = 1 \dots na)$  y  $b_i (i = 1 \dots nb)$  para la velocidad actual del vehículo.
- Resolver la ecuación diofántica de forma recurrente.
- Calcular los polinomios  $G_j$  y formar la matriz  $G$  o la  $G_1$ .
- Calcular  $M$  o  $M_1$ .

Obsérvese que, si se aplica toda la secuencia proyectada de actuaciones de control, se tendría una estrategia de control en bucle abierto durante el periodo completo  $[N_1, N_2]$ . En lugar de esto, de acuerdo con el mecanismo del control predictivo generalizado, se aplica sólo el primer incremento  $\Delta u(k)$  y, en el siguiente instante de control, se vuelve a calcular una secuencia completa de variables de control, a partir de la secuencia de puntos sobre el segmento de camino que se pretende seguir. Este mecanismo de realimentación cabe considerarlo similar al que emplea una persona cuando conduce un vehículo. En la figura 9.28 se presenta un diagrama que ilustra la implantación del seguimiento predictivo generalizado.

Si se elige como primer punto de control sobre el camino uno relativamente próximo y el sistema tiene un error inicial apreciable, la aplicación del método da lugar a grandes incrementos no deseados de la actuación. Para evitarlo, cabe pensar en sustituir la trayectoria deseada (que resulta en las circunstancias mencionadas poco realista) por una trayectoria de aproximación a la deseada. En Hess y Jung (1989) y Ollero



**Figura 9.28:** Implantación del seguimiento predictivo generalizado.

y Amidi (1991) se emplea, en vez de la trayectoria deseada  $y_d(k+j)$ , la  $y_c(k+j)$  definida por la exponencial:

$$y_c(k+j) = y_d(k+j) - \exp(-\tau_e j)[y_d(k+j) - y(k+j)] \quad (9.140)$$

Obsérvese que, si  $\tau_e$  toma un valor muy grande,  $y_c(k+j)$  se approxima a  $y_d(k+j)$ , mientras que, si el valor es muy pequeño, se requiere una actuación de control muy pequeña.

Otra posibilidad es la determinación de un polinomio o función que enlace la posición del vehículo con un punto objetivo elegido sobre el camino. En particular, es posible emplear curvas de evaluación simple y variaciones suficientemente suaves, tales como las  $\beta$  splines, asegurando la continuidad en posición, orientación y curvatura.

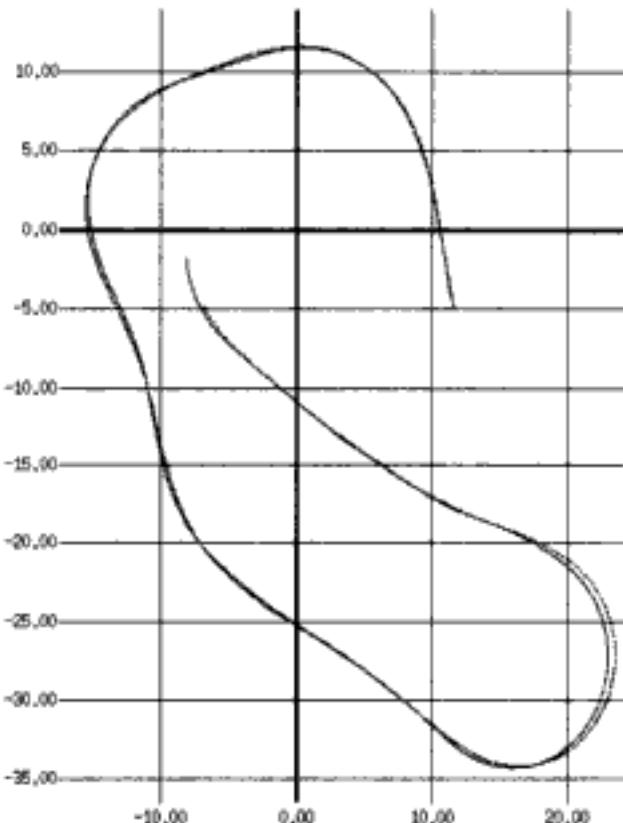
En la figura 9.29 se muestra el resultado de la implantación del control predictivo generalizado en su formulación multivariable (las salidas son la distancia al camino, la diferencia de orientación y la diferencia de curvatura) sobre el vehículo autónomo NavLab I de la Carnegie Mellon University (Ollero y Amidi, 1991), que se mencionaba en el capítulo 2. Se empleó un modelo muy simple obtenido mediante linealización local de las ecuaciones del sistema odometrónico para velocidad constante y una dinámica de primer orden en la actuación con constante de tiempo 0,4 s. En el experimento se empleó una secuencia temporal de 18 puntos, y  $NU = 1$  para una velocidad de 3,4 m/s.

El camino de la figura 9.29 es difícil de seguir debido a sus curvas, especialmente la que aparece en la parte inferior derecha que tiene una curvatura que es prácticamente la máxima del vehículo. Sin embargo, el seguimiento es bueno y se minimizan las

oscilaciones de la dirección del vehículo, tal como se muestra en la figura 9.30. Obsérvese que la variable de dirección del vehículo (variable de control) toma valores próximos a la curvatura del camino.

En el experimento se tomó un valor  $N_1 = 1$  en el índice, escogiendo como primer punto el más próximo sobre el camino. El horizonte  $N = N_2$  se ajustó automáticamente teniendo en cuenta la velocidad del vehículo.

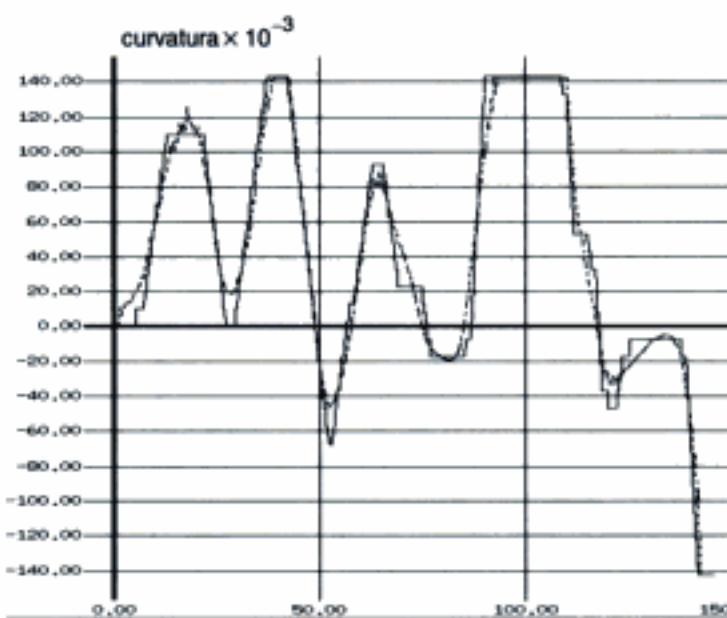
El aumento de  $N$  tiene un efecto anticipativo en el seguimiento de curvas, y lo opuesto ocurre cuando se disminuye  $N$ , caso en el cual tarda en reaccionar ante la aparición de curvas sobre el camino. Este efecto se ilustra en la figura 9.31.



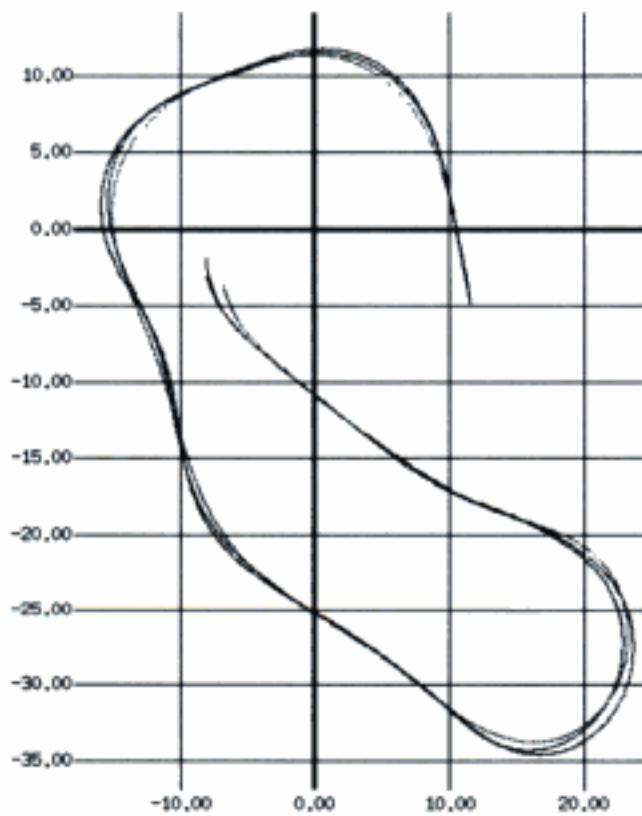
**Figura 9.29:** Seguimiento predictivo generalizado con el NavLab I (Ollero y Amidi, 1991). Camino de referencia en línea continua y camino realizado en línea de trazos.

Es también posible emplear  $NU > 1$ . Sin embargo, esto incrementa el esfuerzo computacional de la implantación ya que es necesario realizar la inversión de una matriz cuando se cambia la velocidad, situación en la cual se requiere volver a calcular las ganancias  $M$ . Asimismo, se ha observado cómo el incremento de  $NU$  aumenta la actividad de control produciendo oscilaciones no deseadas en la dirección.

En Ollero y otros (1994), se presenta un método que permite el ajuste automático de los parámetros del controlador predictivo en función de las características del camino, velocidad y situación relativa del vehículo con respecto al camino.



**Figura 9.30:** Curvatura del camino de referencia (línea continua) y curvatura real del vehículo (línea de trazos) en el seguimiento de la figura 9.29.



**Figura 9.31:** Modificación del horizonte  $N = N_2$  en el seguimiento de la figura 9.29 manteniendo  $N_1 = 1$ . En línea continua el camino de referencia, en línea de trazos el camino seguido por el NavLab con  $N = 10$ , y en línea de trazos y puntos el camino seguido cuando  $N = 25$ .

## 9.6 CONTROL REACTIVO

Corresponde a las arquitecturas de control reactivo que se mencionaban en el capítulo 6, en las cuales las actuaciones de control del vehículo se obtienen *directamente* de la percepción sensorial del entorno. En este caso, pueden contemplarse dos situaciones diferentes:

- Acoplamiento directo entre los sensores externos y los actuadores, de forma que ante determinadas lecturas de los sensores se generen consignas a los actuadores de dirección y velocidad.
- Generación de maniobras cuya ejecución es necesario controlar.

El primer caso puede representarse también mediante el esquema de la figura 9.1, pero sustituyendo los dos bloques superiores por sistemas que tienen como entradas las señales de los sensores externos. Así, por ejemplo, se dispondría de las lecturas de sensores de proximetría que indican la distancia de cada uno de ellos a los objetos del entorno del vehículo. Se trataría de generar la consigna de dirección dependiendo de esas lecturas y, eventualmente, de una dirección global deseada para alcanzar cierto objetivo. Asimismo, se generaría la consigna de velocidad a los actuadores dependiendo de la distancia a la que se encuentren los obstáculos.

La principal dificultad del enfoque del párrafo anterior es la complejidad del diseño del sistema de control a que da origen el normalmente alto número de entradas. Para reducir esta complejidad puede efectuarse una primera etapa de percepción del entorno. Mediante dicha percepción se trata de establecer una descripción que integre el resultado de las diferentes medidas de los sensores y que sea utilizable para el control del vehículo. De esta forma, cabe hablar de la detección de un obstáculo en un determinado ángulo con respecto a la dirección del vehículo o, incluso, de la obtención del camino local más apropiado para el vehículo teniendo en cuenta todos los obstáculos existentes detectados por los sensores. En función de esta percepción se generarían las órdenes de dirección y velocidad del vehículo.

En la generación de maniobras suele distinguirse entre la percepción del entorno y la propia generación de la maniobra. En la percepción se obtiene una descripción del entorno mediante la cual se trata de establecer las maniobras apropiadas. Es también posible asociar "*comportamientos*" (ver capítulo 6) a estas maniobras. Las maniobras pueden ser de diferente grado de complejidad: cambiar a línea paralela a una determinada distancia, girar esquina a la derecha, seguir pared a un lado del vehículo, seguir pasillo, etc. Nótese que estos problemas de seguimiento pueden tratarse, desde el punto de vista del algoritmo de control, de forma parecida a la del seguimiento de caminos tratado en apartados precedentes. La diferencia básica es que ahora no se dispone del camino explícitamente antes de comenzar el seguimiento. Es decir, se emplea realimentación de los sensores de percepción del entorno durante la propia ejecución de la trayectoria.

Otro problema de características similares al comentado en el párrafo precedente es el del seguimiento de objetos móviles. Se trata de que el vehículo siga a un objeto y, en particular, a otro vehículo, por el mismo camino y a la misma velocidad. El problema requiere también el empleo del sistema de percepción externa. Para ello se ha

utilizado, por ejemplo, una realimentación visual y/o sensores de proximetría. Desde el punto de vista del control, el problema del seguimiento visual puede tratarse también mediante el esquema que se mostraba en la figura 9.1. El bucle de control de la izquierda está relacionado con la percepción de movimientos laterales, mientras que el de la derecha lo está con la percepción de distancia al objeto que se está siguiendo. En este caso, pueden aplicarse también los conceptos de teoría de control planteando el problema como el seguimiento de  $\rho_{ref}$  obtenido mediante sensores de percepción del entorno.

## 9.7 CONCLUSIONES

El control de robots móviles involucra métodos diversos, desde los que emplean fundamentalmente conceptos de la teoría de control hasta los basados en técnicas de inteligencia artificial.

Desde un punto de vista práctico, cabe distinguir también entre el control de pequeños robots con ruedas, frecuentemente con locomoción omnidireccional o de tipo diferencial, y las aplicaciones en las que se trabaja con la adaptación de un vehículo convencional, típicamente con locomoción tipo triciclo o Ackerman.

Por otra parte, dentro del control de robots móviles, puede distinguirse entre el seguimiento de caminos explícitos, previamente definidos, y la realización de maniobras reaccionando según la información suministrada por los sensores de percepción del entorno, tales como los sensores de proximetría o el sistema de visión.

En este capítulo se ha prestado mayor atención a los métodos de seguimiento de caminos, presentando tanto fundamentos teóricos del problema como aplicaciones prácticas.

## 9.8 REFERENCIAS

- Aracil J., G. Heredia y A. Ollero, 2000. "Global stability analysis of fuzzy path tracking using frequency response". *Engineering Applications of Artificial Intelligence*. Pergamon. Vol. 13, núm. 2, pp. 109-119.
- Brockett R.W., 1983. "Asymptotic stability and feedback stabilization". *Differential geometric control theory*, R.W. Brockett, R.S. Millmann y H.J. Sussmann, Eds., Birkhauser, Boston, USA, pp. 181-191.
- Campion G., G. Bastin y B. D'Andréa-Novel, 1996. "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots". *IEEE Trans. on Robotics and Automation*. Vol. 12, núm. 1, pp. 47-62.
- Canudas de Wit, B. Siciliano y G. Bastin, 1997. "Theory of Robot Control". Springer.
- Clarke D.W., C. Mohtadi y P.F. Tuffs, 1987. "Generalized predictive control, parte I y II". *Automatica*. Vol. 23, núm. 2, pp. 137-160.
- Cormier W.H. y R. Fenton, 1980. "On the steering of automated vehicles -a velocity-adaptive controller". *IEEE Trans. on Automatic Control*, pp. 375-385.
- Dickmanns E.D. y A. Zapp, 1987. "Autonomous high speed road vehicle guidance by computer vision". Prepr. 10th IFAC World Congress, Munich. Vol. 4, pp. 232-237.
- Fenton R., G. Melocik y K. Olson, 1976. "On the steering of automated vehicles: theory and experiment". *IEEE Trans. on Automatic Control*, pp. 306-315.

- Hess R. y Y. Jung, 1989. "An application of generalized predictive control to rotorcraft terrain following flight". IEEE Trans. on SMC. Vol.19, núm. 5, pp. 955-960.
- Marino R., 1986. "On the largest feedback linearizable subsystem". Syst. Contr. Let. Vol. 6, pp. 345-351.
- Martínez J.L., 1995. "Seguimiento autónomo de caminos en robots móviles". Tesis Doctoral. Universidad de Málaga.
- Monaco S. y D. Normand-Cyrot, 1991. "An introduction to motion planning under multirate digital control". Proc. 31st Conference on Decision and Control, Tucson, AZ, 1991, pp. 1780-1785.
- Murray R.M., 1993. "Applications and extensions of goursat normal form to control of nonlinear systems". Proc. 32nd IEEE Conference on Decision and Control, San Antonio, TX.
- Murray R.M. y S.S. Sastry, 1993. "Nonholonomic motion planning: Steering using sinusoids". IEEE Trans. Automat. Contr. Vol. 38, núm. 5, pp. 700-716.
- Ollero A. y O. Amidi, 1991. "Predictive Path Tracking of Mobile Robots. Applications to the CMU NavLab". Proc. of the Fifth International Conference on Advanced Robotics, Pisa. Vol. II, pp. 1081-1086.
- Ollero A., A. García-Cerezo y J. Martínez, 1994. "Fuzzy Supervisory Path Tracking of Mobile Robots". Control Engineering Practice. Vol.2, núm. 2, pp. 313-319, Pergamon Press.
- Rodríguez Castaño A., G. Heredia y A. Ollero (1999). "Analysis of a GPS-based fuzzy supervised path tracking system for large unmanned vehicles". Preprints of the 4th IFAC International Symposium on Intelligent Components and instruments for Control Applications". SICICA 2000, pp. 141-146. Buenos Aires, Argentina, 13-15.
- Rodríguez Castaño A., G. Heredia y A. Ollero (1999). "Fuzzy path tracking of autonomous vehicles using GPS". Proceedings of the 1999 Eusflat-Estyif Joint Conference, pp. 111-114. Palma de Mallorca.
- Samson C., 1995. "Control of chained systems. Application of path following and time-varying point stabilization of mobile robots". IEEE Trans. on Automatic Control. Vol. 40, pp. 64-77.
- Shin D., 1990. "High performance tracking of explicit paths by roadworthy mobile robots". Ph. D. Dissertation. Carnegie Mellon University.
- Thorpe C., 1990. "Vision and Navigation: The CMU NavLab". Kluwer Academic Publishers.

# Generación de trayectorias

---

En los capítulos 8 y 9 se ha supuesto que se conocía la trayectoria que debía describir el robot en su movimiento y se estudiaba el problema del control del robot para que siguiera la trayectoria especificada. En este capítulo se estudia la generación de dichas trayectorias. Partiendo de una especificación del movimiento que debe seguir el robot, se trata de definir de forma precisa una trayectoria espacial y temporal del robot. En el capítulo se estudia la generación de trayectorias en el espacio cartesiano y en el de las articulaciones. Se consideran tanto robots manipuladores como móviles. Se incluyen ejemplos en los cuales se utilizan funciones MATLAB específicas.

## 10.1 PLANTEAMIENTO DEL PROBLEMA

---

Supóngase que el usuario especifica un movimiento en el espacio de trabajo del robot, normalmente espacio cartesiano. Para ello puede emplearse un terminal interactivo indicando simplemente el punto de destino. El movimiento puede especificarse también mediante una sentencia de programa o bien, automáticamente, como resultado de la ejecución de un programa de planificación.

En los lenguajes de programación de robots existen sentencias para especificar el movimiento de un robot indicando la duración del movimiento o la velocidad. Así, podría emplearse una sentencia tal como:

mover BRAZO a C con duración 3 segundos

En caso de no especificarse una duración, el movimiento se realiza con una velocidad previamente establecida en el sistema (velocidad por defecto). Es también posible especificar movimientos con puntos de paso mediante sentencias tales como:

mover BRAZO a C vía B con duración 6 segundos

La duración se refiere al movimiento completo. El procedimiento de generación de trayectorias debe repartir la duración total entre los segmentos. Puede especificarse también la duración de un segmento simple tal como, por ejemplo:

mover BRAZO a C vía B donde duración 3 segundos

con lo cual se indica que el primer segmento que conduce a B tiene una duración de 3 s.

Por último, es posible también especificar diversos puntos de paso con una sentencia tal como:

mover BRAZO a C vía B, A, D

en la cual no se ha indicado la duración (velocidad por defecto).

Se recuerda que, tal como se ha mencionado en el capítulo 3, la posición y orientación se indican mediante la posición del origen del sistema y la orientación de sus ejes. En los robots móviles se indicaría la trayectoria que debe describir un sistema sólidario al centro de guiado del robot. En los robots manipuladores, la trayectoria en el espacio de trabajo suele especificarse por la posición y orientación de un sistema asociado al efecto final.

De forma más concreta, existen dos formas básicas para especificar el movimiento:

- 1) Suministrando puntos consecutivos e ignorando la trayectoria espacial que describe el robot entre cada dos puntos.
- 2) Especificando el camino que debe unir los puntos mediante una determinada trayectoria, tal como una línea recta o un círculo, que debe describir el robot en el espacio de trabajo.

La primera alternativa, denominada tradicionalmente *control punto a punto*, sólo tiene interés práctico cuando los puntos están suficientemente separados ya que, en caso contrario, la especificación sería muy laboriosa. Por otra parte, los puntos tampoco pueden estar muy separados ya que entonces el riesgo de que se generen movimientos imprevisibles o no controlados es grande. En este control punto a punto el sistema de control automático del robot debe realizar la interpolación entre los puntos especificados, de forma tal que, posteriormente, sea posible realizar el control de movimientos para que el robot pase por dichos puntos.

La segunda estrategia se denomina tradicionalmente *control de trayectoria continua*. En este caso, el sistema de control debe hacer que el robot reproduzca lo más fielmente posible la trayectoria especificada.

Como se ha indicado anteriormente, la especificación del movimiento puede involucrar o no restricciones temporales. En los robots manipuladores esta especificación temporal es muy usual, dando lugar a condiciones sobre la velocidad y aceleración con las que deben ejecutarse los movimientos. Sin embargo, tal como se ha indicado en el capítulo 9, en los robots móviles el control de la dirección del vehículo y el de la velocidad, aunque están acoplados, suelen tratarse de forma independiente, con lo cual suele hablarse del problema de la generación de caminos sin hacer alusión al tiempo en recorrer dicho camino. No obstante, la velocidad debe tenerse en cuenta para la generación de caminos apropiados, ya que existe una relación entre la curvatura del camino y la velocidad máxima a la que puede recorrerse dicho camino.

Supuesto que la trayectoria que debe seguir el robot se especifica en el espacio cartesiano, existen dos alternativas para su ejecución:

- 1) Definir los bucles de control directamente en el espacio cartesiano y controlar al robot para que se anule el error de seguimiento de la trayectoria en este espacio.
- 2) Transformar la trayectoria del espacio cartesiano al espacio de las variables articulares y controlar la evolución de cada una de las variables articulares definiendo los bucles de control en este espacio.

El primer caso es el más habitual en robots móviles. Como se ha puesto de manifiesto en el capítulo 9, la curvatura del camino generado en el espacio cartesiano está directamente relacionada con la variable de control que se emplea para el seguimiento de caminos por parte de diversas configuraciones de vehículos.

En robots manipuladores con configuración cartesiana (ver capítulo 2) las coordenadas del camino son directamente las referencias de los bucles de control de las articulaciones, con lo cual resulta evidente que la generación de caminos debe realizarse en el espacio cartesiano ya que los bucles de control se plantean directamente en este espacio. Sin embargo, este no es el caso de otras configuraciones, en las cuales es necesario acudir al modelo cinemático para obtener las variables articulares que son las que definen los bucles de control de bajo nivel.

Por tanto, los sistemas de control de los manipuladores robóticos industriales suelen convertir las especificaciones en el espacio de trabajo a un conjunto de valores deseados para las variables articulares, empleando para ello la cinemática inversa. De esta forma, el problema de generación de trayectorias se plantea normalmente en el espacio articular, en cuyo caso se trata de especificar la posición, velocidad y aceleración para cada una de las articulaciones. Nótese como las referencias de los bucles de control en el capítulo 8 son las posiciones, velocidades y aceleraciones de las articulaciones.

No debe olvidarse que las trayectorias generadas deben tener buenas propiedades con vistas a su ejecución por el robot. Por consiguiente, es necesario tener en cuenta, directa o indirectamente, las restricciones cinemáticas y dinámicas involucradas. En general, las trayectorias deben ser suaves, lo que implica restricciones sobre las derivadas. Normalmente se exige que al menos la primera derivada sea continua, pudiendo exigirse también la continuidad en derivadas de orden superior.

Se recuerda que el problema de la generación de trayectorias debe resolverse en tiempo real. Por tanto, se trata también de que la generación de las trayectorias sea computacionalmente eficiente. En robots móviles, normalmente, se requiere que las trayectorias se generen en el espacio cartesiano en pocas décimas de segundo. En robots manipuladores, la generación de trayectorias articulares suele realizarse en tiempos del orden de los milisegundos o decenas de milisegundos.

## 10.2 DEFINICIÓN PARAMÉTRICA DE LAS CURVAS

En general, interesaría emplear una definición paramétrica de la curva que permita contemplar fácilmente la generación de movimientos tanto en el espacio de trabajo como en el espacio de las variables articulares. Las curvas paramétricas se caracterizan por la representación funcional de cada una de sus dimensiones. Las coordenadas de un punto de la curva en el plano se expresan mediante el vector:

$$p(\lambda) = [x(\lambda), y(\lambda)] \quad (10.1)$$

donde  $\lambda$  es el parámetro, y  $x(\lambda)$  e  $y(\lambda)$  son las funciones asociadas a cada una de las dimensiones. Esta definición paramétrica ofrece también la posibilidad de representar curvas cerradas.

La derivada con respecto al parámetro se indica como:

$$p' = \frac{d}{d\lambda} p(\lambda) = \left[ \frac{d}{d\lambda} x(\lambda), \frac{d}{d\lambda} y(\lambda) \right] \quad (10.2)$$

La derivada para un valor del parámetro  $\lambda_1$  se representa como:

$$p'(\lambda_1) = \frac{d}{d\lambda} p(\lambda)|_{\lambda=\lambda_1} \quad (10.3)$$

La misma notación se empleará para las derivadas de orden superior tal como  $p''(\lambda_1)$  para la derivada de orden segundo.

En robótica móvil es muy usual considerar las variaciones de curvatura del camino. Se recuerda que la curvatura está directamente relacionada con la variable típica de control de la dirección del vehículo, que suele ser también una curvatura requerida. Para ello es necesario acudir a conceptos de geometría diferencial.

El vector tangente unitario  $T(\lambda)$  es un vector normalizado con la misma dirección y sentido que el de la primera derivada paramétrica:

$$T(\lambda) = \frac{p'(\lambda)}{|p'(\lambda)|} \quad (10.4)$$

siendo  $|p'(\lambda)|$  el módulo del vector  $p'(\lambda)$ .

Resulta también interesante considerar la longitud de arco, que puede definirse como:

$$s(\lambda) = \int_0^\lambda |p'(\lambda)| d\lambda \quad (10.5)$$

o bien:

$$s'(\lambda) = |p'(\lambda)| \quad (10.6)$$

que es una expresión de la velocidad.

Se verifican las relaciones:

$$\frac{dp}{ds} = \frac{p'(\lambda)}{s'(\lambda)} = T(s) \quad (10.7)$$

El vector tangente  $T$  tiene magnitud constante y, por tanto, su derivada con respecto a  $\lambda$  es el vector cero o un vector no cero perpendicular a  $T$ . Nótese que si  $|T(\lambda)|$  es constante, entonces  $T(\lambda) \cdot T(\lambda)$  es también constante y al diferenciar respecto a  $\lambda$  se tendrá:

$$\frac{dT}{d\lambda} \cdot T + T \frac{dT}{d\lambda} = 0 \quad (10.8)$$

Por tanto,

$$2T \frac{dT}{d\lambda} = 0 \quad (10.9)$$

Por tanto, el producto escalar de  $T$  con  $dT/d\lambda$  es cero, lo cual sucede si ambos vectores son perpendiculares o si uno de ellos es el vector cero. Por otra parte, nótese que en la figura 10.1 se pone de manifiesto que si  $\Delta T$  es pequeño, entonces  $\Delta T$  es aproximadamente perpendicular a  $T$ .

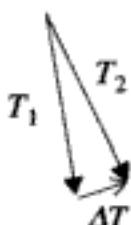


Figura 10.1: Aproximación de vector perpendicular.

El vector normal es un vector perpendicular a  $T(\lambda)$ . Por tanto, de los párrafos anteriores se deduce que el vector normal unitario viene dado por:

$$N(\lambda) = \frac{T'(\lambda)}{|T'(\lambda)|} \quad (10.10)$$

Nótese también que, si se aplica la regla de la cadena al numerador y denominador de (10.10), se obtiene:

$$N = \frac{(dT/ds) (ds/d\lambda)}{|(dT/ds) (ds/d\lambda)|} = \frac{(dT/ds)}{|dT/ds|} \quad (10.11)$$

La curvatura  $\gamma$  de la curva en cualquier punto se define como la magnitud del vector  $dT/ds$  en ese punto:

$$\gamma = \left| \frac{dT}{ds} \right| \quad (10.12)$$

Por tanto, teniendo en cuenta (10.11) se tendrá:

$$\frac{dT}{ds} = \gamma N \quad (10.13)$$

lo que puede interpretarse diciendo que  $T$  da vuelta en la dirección  $N$  a la razón  $k$  con respecto a la longitud del arco.

Los vectores  $T(\lambda)$  y  $N(\lambda)$  forman un sistema local de coordenadas en cada punto de la curva.

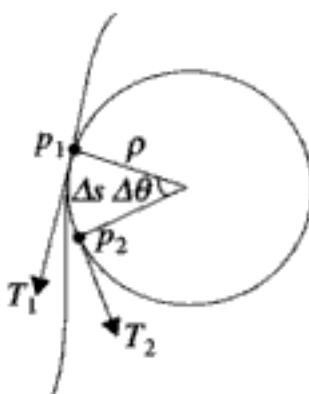


Figura 10.2: Círculo de osculación y radio de curvatura.

El círculo de osculación en un punto de la curva  $p_1$ , es el que más se aproxima a la curva en dicho punto  $p_1$ , tal como se ilustra en la figura 10.2. La tangente de la curva debe coincidir con la tangente en el punto. Si el radio de este círculo es  $\rho$  diferente de cero, la curvatura se define también como la inversa del radio:

$$\gamma = \frac{1}{\rho} \quad (10.14)$$

El radio  $\rho$  se denomina radio de curvatura.

Nótese que la longitud del arco entre  $p_1$  y  $p_2$  en la figura 10.2 es  $\Delta s = \rho \Delta \theta$ . Los vectores unitarios tangentes  $T_1$  y  $T_2$  forman también un ángulo  $\Delta \theta$ , y el cambio de la tangente unitaria al pasar de  $p_1$  a  $p_2$  es  $\Delta T = T_2 - T_1$ . Teniendo en cuenta que las magnitudes de  $T_2$  y  $T_1$  son la unidad, la magnitud de  $\Delta T$  es, aproximadamente,  $\Delta \theta$  (figura 10.1).

Por tanto,

$$\left| \frac{\Delta T}{\Delta s} \right| = \frac{\Delta \theta}{\Delta s} = \frac{1}{\rho}$$

Esta aproximación se vuelve exacta a medida que  $\Delta s$  se acerca a cero, por lo cual puede escribirse:

$$\left| \frac{dT}{ds} \right| = \frac{1}{\rho}$$

El vector de curvatura  $vk(\lambda)$ , es un vector que tiene como módulo la curvatura  $\gamma(\lambda)$  y como dirección y sentido los del vector normal  $N(\lambda)$ :

$$vk(\lambda) = k(\lambda)N(\lambda) \quad (10.15)$$

El vector curvatura se define también como:

$$vk(s) = \frac{d^2}{ds^2} p(s) \quad (10.16)$$

Teniendo en cuenta (10.4) y (10.6), puede escribirse:

$$p'(\lambda) = |p'(\lambda)|T(\lambda) = s'(\lambda)T(\lambda) \quad (10.17)$$

Derivando con respecto a  $\lambda$  (10.17), se obtiene:

$$p''(\lambda) = s''(\lambda)T(\lambda) + s'(\lambda)\frac{dT(\lambda)}{d\lambda} \quad (10.18)$$

Utilizando (10.13)

$$\frac{dT(\lambda)}{d\lambda} = \frac{dT}{ds} \frac{ds}{d\lambda} = \gamma N(\lambda) s'(\lambda) \quad (10.19)$$

Sustituyendo (10.19) en (10.18):

$$p''(\lambda) = s''(\lambda)T(\lambda) + [s'(\lambda)]^2 \gamma N(\lambda) \quad (10.20)$$

El producto vectorial de  $p'\lambda$  y  $p''(\lambda)$  es:

$$p'(\lambda) \times p''(\lambda) = [s'(\lambda)T(\lambda)] \times [s''(\lambda)T(\lambda) + [s'(\lambda)]^2 \gamma N(\lambda)] \quad (10.21)$$

Teniendo en cuenta que  $T \times T = 0$  y que  $T$  y  $N$  son perpendiculares, por lo cual su producto vectorial  $B = T \times N$  es un vector unitario, al cual se denomina vector binormal, la (10.21) puede escribirse como:

$$p'(\lambda) \times p''(\lambda) = \gamma [s'(\lambda)]^3 B \quad (10.22)$$

Por tanto, la curvatura puede obtenerse también como:

$$\gamma = \frac{|p'(\lambda) \times p''(\lambda)|}{|s'(\lambda)|^3} \quad (10.23)$$

Otra posible expresión es:

$$\gamma = \frac{|p'(\lambda) \times p''(\lambda)|}{[p'(\lambda)p''(\lambda)]^{3/2}} \quad (10.24)$$

## 10.3 TÉCNICAS DE INTERPOLACIÓN

### 10.3.1 Técnicas básicas de interpolación

La generación de trayectorias hace necesario considerar al menos la continuidad de la primera derivada. Por consiguiente, las restricciones son:

$$p(\lambda_0) = p_0 \quad (10.25)$$

$$p(\lambda_f) = p_f \quad (10.26)$$

$$p'(\lambda_0) = p'_0 \quad (10.27)$$

$$p''(\lambda_f) = p''_f \quad (10.28)$$

Para que se cumplan estas restricciones, se necesita un polinomio de grado al menos tres (cúbico):

$$p(\lambda) = a_0 + a_1\lambda + a_2\lambda^2 + a_3\lambda^3 \quad (10.29)$$

$$p'(\lambda) = a_1 + 2a_2\lambda + 3a_3\lambda^2 \quad (10.30)$$

$$p''(\lambda) = 2a_2 + 6a_3\lambda \quad (10.31)$$

Por tanto, se tienen cuatro ecuaciones con cuatro incógnitas que son los parámetros  $a_0, a_1, a_2, a_3$ :

Suponiendo  $\lambda_0 = 0$ , las ecuaciones son:

$$p_0 = a_0 \quad (10.32)$$

$$p_f = a_0 + a_1\lambda_f + a_2\lambda_f^2 + a_3\lambda_f^3 \quad (10.33)$$

$$p'_0 = a_1 \quad (10.34)$$

$$p'_f = a_1 + 2a_2\lambda_f + 3a_3\lambda_f^2 \quad (10.35)$$

Resolviendo, se obtiene:

$$a_0 = p_0 \quad (10.36)$$

$$a_1 = p'_0 \quad (10.37)$$

$$a_2 = \frac{3}{\lambda_f^2}(p_f - p_0) - \frac{2}{\lambda_f}p'_0 - \frac{1}{\lambda_f}p'_f \quad (10.38)$$

$$a_3 = -\frac{2}{\lambda_f^3}(p_f - p_0) + \frac{1}{\lambda_f^2}(p'_f + p'_0) \quad (10.39)$$

Si se desean tener en cuenta restricciones en derivadas de orden superior a uno, es necesario incrementar el orden del polinomio. En particular, se emplean polinomios quínticos para especificar también la segunda derivada:

$$p(\lambda) = a_0 + a_1\lambda + a_2\lambda^2 + a_3\lambda^3 + a_4\lambda^4 + a_5\lambda^5 \quad (10.40)$$

En este caso, suponiendo  $\lambda_0 = 0$ , las condiciones son:

$$p_0 = a_0 \quad (10.41)$$

$$p_f = a_0 + a_1 \lambda_f + a_2 \lambda_f^2 + a_3 \lambda_f^3 + a_4 \lambda_f^4 + a_5 \lambda_f^5 \quad (10.42)$$

$$p'_0 = a_1 \quad (10.43)$$

$$p'_f = a_1 + 2a_2 \lambda_f + 3a_3 \lambda_f^2 + 4a_4 \lambda_f^3 + 5a_5 \lambda_f^4 \quad (10.44)$$

$$p''_0 = 2a_2 \quad (10.45)$$

$$p''_f = 2a_2 + 6a_3 \lambda_f + 12a_4 \lambda_f^2 + 20a_5 \lambda_f^3 \quad (10.46)$$

Estas restricciones especifican un sistema lineal de seis ecuaciones con seis incógnitas, cuya solución es:

$$a_0 = p_0 \quad (10.47)$$

$$a_1 = p'_0 \quad (10.48)$$

$$a_2 = \frac{p''_0}{2} \quad (10.49)$$

$$a_3 = \frac{20p_f - 20p_0 - (8p'_f + 12p'_0)\lambda_f - (3p''_0 - p''_f)\lambda_f^2}{2\lambda_f^3} \quad (10.50)$$

$$a_4 = \frac{30p_0 - 30p_f + (14p'_f + 16p'_0)\lambda_f + (3p''_0 - 2p''_f)\lambda_f^2}{2\lambda_f^4} \quad (10.51)$$

$$a_5 = \frac{12p_f - 12p_0 - (6p'_f + 6p'_0)\lambda_f - (p''_0 - p''_f)\lambda_f^2}{2\lambda_f^5} \quad (10.52)$$

Las ecuaciones anteriores pueden aplicarse de forma recurrente para definir una curva por tramos considerando puntos de paso.

Si el parámetro  $\lambda$  es el tiempo, las derivadas primera y segunda corresponden a la velocidad y aceleración. Estas restricciones sobre velocidad y aceleración son frecuentes en la generación de trayectorias para manipuladores robóticos, como se verá en apartados posteriores.

### 10.3.2 Empleo de funciones spline

La consideración de las variaciones de la curvatura empleando las técnicas básicas de interpolación mencionadas más arriba es una tarea difícil. Por ello, cuando la cur-

vatura es importante, resulta más conveniente el empleo de funciones spline. La curva spline más sencilla es la spline cúbica (Rogers y Adams, 1989).

Supóngase que el parámetro  $\lambda$  en el polinomio cúbico del apartado 10.3.1 varía entre 0 y 1, es decir:  $\lambda_0 = 0$  y  $\lambda_f = 1$ . Por tanto, los coeficientes (10.36)-(10.39) con  $p_f = p_1$ ,  $p'_f = p'_1$  son:

$$a_0 = p_0 \quad (10.53)$$

$$a_1 = p'_0 \quad (10.54)$$

$$a_2 = 3(p_1 - p_0) - 2p'_0 - p'_1 \quad (10.55)$$

$$a_3 = -2(p_1 - p_0) - (p'_1 + p'_0) \quad (10.56)$$

Sustituyendo en (10.40) y agrupando términos se obtiene:

$$p(\lambda) = (2\lambda^3 - 3\lambda^2 + 1)p_0 + (-2\lambda^3 + 3\lambda^2)p_1 + (\lambda^3 - 2\lambda^2 + \lambda)p'_0 + (\lambda^3 - \lambda^2)p'_1 \quad (10.57)$$

que puede escribirse en la forma:

$$p(\lambda) = F_1(\lambda)p_0 + F_2(\lambda)p_1 + F_3(\lambda)p'_0 + F_4(\lambda)p'_1 = F^T P \quad (10.58)$$

siendo  $F^T = [F_1 \ F_2 \ F_3 \ F_4]$  un vector de funciones de ponderación o mezclado y  $P = [p_0 \ p_1 \ p'_0 \ p'_1]^T$  una matriz de coeficientes geométricos o condiciones de contorno.

Derivando en la ecuación anterior, se obtiene:

$$p'(\lambda) = F'_1(\lambda)p_0 + F'_2(\lambda)p_1 + F'_3(\lambda)p'_0 + F'_4(\lambda)p'_1 = (F')^T P \quad (10.59)$$

siendo:

$$(F')^T = [(6\lambda^2 - 6\lambda)(-6\lambda^2 + 6\lambda)(3\lambda^2 - 4\lambda + 1)(3\lambda^2 - 2\lambda)] \quad (10.60)$$

Derivando otra vez:

$$p''(\lambda) = F''_1(\lambda)p_0 + F''_2(\lambda)p_1 + F''_3(\lambda)p'_0 + F''_4(\lambda)p'_1 = (F'')^T P \quad (10.61)$$

siendo:

$$(F'')^T = [(12\lambda - 6)(-12\lambda + 6)(6\lambda - 4)(6\lambda - 2)] \quad (10.62)$$

Empleando las expresiones anteriores es posible calcular fácilmente los valores de las coordenadas y su primera y segunda derivada en cualquier punto en el intervalo  $\lambda \in [0, 1]$ .

Considérense las expresiones de la segunda derivada. Particularizando para  $\lambda = 0$  se obtiene:

$$p''_0 = -6p_0 + 6p_1 - 4p'_0 - 2p'_1 \quad (10.63)$$

y para  $\lambda = 1$ :

$$p''_1 = 6p_0 - 6p_1 + 2p'_0 + 4p'_1 \quad (10.64)$$

Despejando en las últimas dos ecuaciones:

$$p'_0 = -p_0 + p_1 - \frac{1}{6}(2p''_0 + p''_1) \quad (10.65)$$

$$p'_1 = -p_0 + p_1 + \frac{1}{6}(p''_0 + 2p''_1) \quad (10.66)$$

Sustituyendo en (10.58) y agrupando términos se obtiene:

$$p(\lambda) = (1-\lambda)p_0 + \lambda p_1 + \frac{1}{6}(-\lambda^3 + 3\lambda^2 - 2\lambda)p''_0 + \frac{1}{6}(\lambda^3 - \lambda)p''_1 \quad (10.67)$$

que se conoce como forma en derivada segunda de una curva paramétrica. Esta forma puede utilizarse para expresar las ecuaciones de deformación en el sentido vertical de una barra horizontal suspendida por sus extremos. En este caso, la deformación viene dada por la coordenada vertical  $y$ ; su derivada  $y'$  coincide con la pendiente de la curva, y su segunda derivada  $y''$  con la curvatura. Las ecuaciones son las mismas que las de curvas elásticas, o de mínima energía, a las que se conoce con el nombre de "splines". Esta denominación se utiliza en inglés para designar a tiras plásticas o de otro material flexible que se utilizaban para dibujar una curva suave que pasara a través de ciertos puntos de control.

Considérese ahora que se pretende diseñar una curva que pase por una serie de puntos de control  $\{p_1, p_2, \dots, p_n\}$ . La curva entre cada par de puntos consecutivos es una spline de las características de los párrafos anteriores. Así, en el segmento entre  $p_i$  y  $p_{i+1}$  se asigna a  $p_i$  el valor del parámetro  $\lambda = 0$  y a  $p_{i+1}$  el valor  $\lambda = 1$ . Si se conocen las pendientes en cada punto, es posible calcular los coeficientes de las splines. Teniendo en cuenta que la curva que interpola todos los puntos debe ser continua y suave, cada dos segmentos de curva que se encuentran en uno de los puntos de paso  $p_i$  deben tener una tangente común. Asimismo, debe cumplirse la condición de continuidad en la curvatura.

Si  $\beta_{i-1}$  es el ángulo que forma con la horizontal ( $x = 0$ ) la línea de longitud  $L_{i-1}$  que une el punto  $p_{i-1}$  con el  $p_i$  (figura 10.2), se demuestra (Mortenson, 1985) que en la unión entre el segmento  $i-1$  y el  $i$  debe verificarse la relación:

$$6 \operatorname{tg}(\beta_i - \beta_{i-1}) = \left(\frac{1}{L_{i-1}}\right)y''_{i-1} + \left(\frac{2}{L_{i-1}} + \frac{2}{L_i}\right)y''_i + \left(\frac{1}{L_i}\right)y''_{i+1} \quad (10.68)$$

siendo  $y_i, y'_i, y''_i$  la coordenada y sus derivadas con respecto al parámetro en el punto  $p_i$ . Por tanto, si se conocen los ángulos  $\beta_i$  y las longitudes  $L_i$  en cada uno de los  $n$  puntos, pueden plantearse  $n$  ecuaciones con  $n$  incógnitas para  $i = 1, \dots, n$ . Nótese que, de esta forma, en las ecuaciones aparecen  $y''_0$  e  $y''_{n+1}$ , así como  $L_0$  y  $\beta_0$  sin sentido

aparente. Para resolver estas ambigüedades, puede actuarse de formas diferentes. Una primera opción es suponer que las splines no están restringidas en los puntos 1 y  $n$ . En este caso puede tomarse  $y''_1 = y''_n = 0$ . Otra alternativa es elegir las pendientes en  $i = 1$  e  $i = n$ . Finalmente, es posible también suponer una curvatura constante en el primer y último segmento, con lo cual  $y''_0 = y''_1$ ,  $y''_n = y''_{n+1}$ . Con cualquiera de las tres opciones es posible resolver el sistema de  $n$  ecuaciones con  $n$  incógnitas obteniéndose finalmente  $y''_1, \dots, y''_n$ , que son los coeficientes que permiten formular las ecuaciones explícitas de las splines en la forma en derivada segunda.

La figura 10.3 representa una curva spline cúbica  $p(\lambda)$ , formada por dos segmentos  $p_{i-1}(\lambda)$  y  $p_i(\lambda)$  que mantienen su continuidad hasta la segunda derivada en el punto de unión  $p_i$ . Esta continuidad se debe a que la orientación al final del primer tramo coincide con la inicial en el segundo, ambas definidas por el vector tangente  $p'_i$ . Por tanto, las orientaciones en los puntos intermedios no pueden elegirse libremente y sólo se pueden imponer las orientaciones inicial y final en el primer y último punto del polígono de control respectivamente. Este hecho limita el empleo de las splines cúbicas en la generación de caminos.

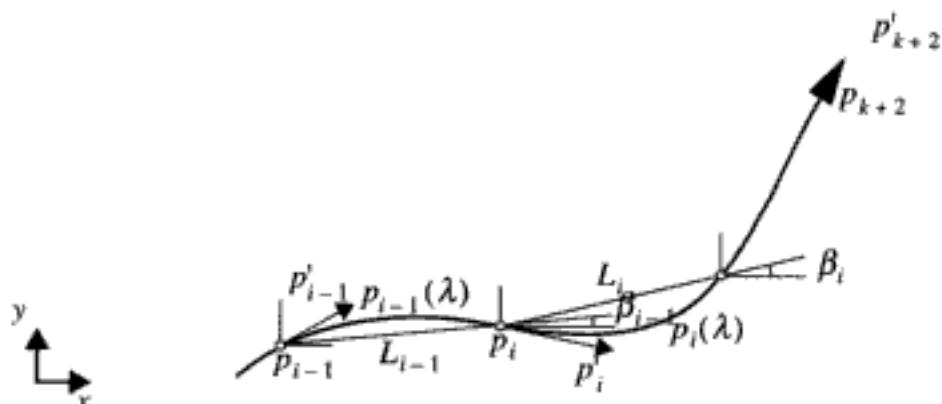


Figura 10.3: Spline cónica.

Por ello se utilizan normalmente funciones más complejas que ofrecen mayor flexibilidad y generan curvas más suaves al precio de no tener que pasar necesariamente por todos los puntos intermedios que definen el camino, sino tan sólo aproximarse a ellos. Los puntos definen los vértices de un polígono de control. Existen funciones de ponderación de los puntos que definen un índice de atracción de la curva sobre el vértice de control al cual ponderan. Este índice da lugar a una mayor o menor proximidad de la curva a un determinado vértice del polígono de control. Por tanto, la curva no interpola el polígono de control como en el caso anterior, sino que se trata de una curva de aproximación. Entre estas curvas cabe destacar las denominadas B-splines (de Boor, 1972) o splines básicas, que pueden describirse mediante:

$$p_i(\lambda) = \sum_{i=0}^{N-1} V_i \cdot B_{i,k}(\lambda) \quad (10.69)$$

siendo  $\{V_0, V_1, \dots, V_{N-2}, V_{N-1}\}$  los vértices del polígono de control, y  $B_{i,k}$  funciones de ponderación o mezclado de orden  $k$  definidas de forma recurrente como:

$$B_{i,1}(\lambda) = 1 \quad \text{para } \lambda_i \leq \lambda \leq \lambda_{i+1} \quad (10.70)$$

$$B_{i,1}(\lambda) = 0 \quad \text{en otro caso} \quad (10.71)$$

$$B_{i,k}(\lambda) = \frac{(\lambda - \lambda_i)B_{i,k-1}(\lambda)}{\lambda_{i+k-1} - \lambda_i} + \frac{(\lambda_{i+k} - \lambda)B_{i+1,k-1}(\lambda)}{\lambda_{i+k} - \lambda_i} \quad (10.72)$$

Las funciones base  $B_{i,k}(\lambda)$  son polinomios de grado  $k-1$ , los cuales garantizan la continuidad  $C^{k-2}$  para la curva completa. El orden mínimo para garantizar la continuidad de la orientación y la curvatura es  $k=4$  lo cual corresponde a las B-splines cúbicas.

Los  $\lambda_i$  en las ecuaciones (10.70)-(10.72) son los nudos de la curva B-spline. Un caso especialmente útil es cuando los nudos están uniformemente espaciados con el primero y el último repetidos  $k$  veces:

$$\lambda_i = 0 \quad \text{si } 0 \leq i < k \quad (10.73)$$

$$\lambda_i = i - k + 1 \quad \text{si } k \leq i \leq N \quad (10.74)$$

$$\lambda_i = N - k + 2 \quad \text{si } i > N \quad (10.75)$$

La ecuación (10.69) se aplica tanto para la coordenada  $x$  como para la  $y$ . El cálculo de la curva para valores particulares de los parámetros  $\lambda \in (\lambda_j, \lambda_{j+1})$  sólo involucra el cálculo de  $k$  números  $B_{i,k}(\lambda)$ ,  $i = j-k+1, \dots, j$ ; por tanto,  $p_k(\lambda)$  viene dada por:

$$p_k(\lambda) = \sum_{i=j-k+1}^j V_i \cdot B_{i,k}(\lambda) \quad (10.76)$$

La diferenciación de  $p(\lambda)$  puede realizarse a partir de las ecuaciones:

$$V_i^{(0)} = V_i \quad (10.77)$$

$$V_i^{(j)} = (V_i^{(j-1)} - V_{i-1}^{(j-1)}) / (\lambda_{i+k-j} - \lambda_i) \quad \text{para } j > 0 \quad (10.78)$$

$$p^{(j)}(\lambda) = (k-1)\dots(k-j) \sum_{i=j-k+1}^j V_i^{(j)} \cdot B_{i,k-j}(\lambda) \quad (10.79)$$

donde  $(j)$  representa el orden de la derivada. Se han desarrollado algoritmos eficientes para evaluar la función y sus derivadas. La orientación y la curvatura para un valor particular del parámetro se calculan mediante las expresiones:

$$\theta(\lambda) = \arctg \left( \frac{y^{(1)}(\lambda)}{x^{(1)}(\lambda)} \right) \quad (10.80)$$

$$\kappa(\lambda) = \frac{[x^{(1)}(\lambda) \cdot y^{(2)}(\lambda)] - [x^{(2)}(\lambda) \cdot y^{(1)}(\lambda)]}{\{[x^{(1)}(\lambda)]^2 + [y^{(1)}(\lambda)]^2\}^{3/2}} \quad (10.81)$$

donde  $[x^{(j)}(\lambda)][y^{(j)}(\lambda)]$  para  $j = 1, 2$  se calculan usando (10.69) particularizada para  $x$  e  $y$ .

Empleando las ecuaciones anteriores es fácil generar una curva que comience en  $V_0$  y llegue hasta  $V_{N-1}$ . En general, la curva no pasa por los puntos interiores de control  $V_1, \dots, V_{n-2}$ . Sin embargo, puede asegurarse que la curva está situada en el "hull-convex" definido por los vértices del polígono, tal como se ilustra en la figura 10.4.

Los B-splines son curvas paramétricas. Por tanto, las coordenadas deben generarse para los diferentes valores del parámetro  $\lambda$  desde  $\lambda = 0$  hasta  $\lambda_{max}$ , con  $\lambda_{max}$  dado por el número de puntos de control y el orden de la curva. Cuando no hay puntos de control duplicados,  $\lambda_{max}$  viene dado por  $N - k + 2$ .

Las curvas de Bezier (Farin, 1992) tienen funciones peso basadas en los polinomios de Bernstein y el grado de la curva resulta igual al número de vértices del polígono de control. Esta curva de aproximación interpola al primer y último punto del polígono de control, mientras que sobre el resto realiza una aproximación. Las B-splines tienen un comportamiento idéntico al de las curvas de Bezier cuando el orden coincide con el número de vértices de control y los nudos están uniformemente espaciados.

En general, la clase de spline queda determinada por las funciones peso, que confieren a la curva ciertas características en relación a su polígono de control. El tipo de función de peso que se emplee influye también de forma importante en el tiempo de cálculo. Otra característica es la relación entre el parámetro de la curva y la longitud de arco. Interesa que esta relación sea lo más lineal posible para poder predecir dónde se encuentra el robot cuando recorre la curva.

A diferencia de los splines cúbicos, tanto en las B-splines como en las curvas de Bezier el aumento del número de vértices hace que existan variables libres que pueden ser utilizadas para la imposición de la posición, orientación y curvatura en la postura inicial o final. Gracias a esta característica y a la suavidad de variación de la curvatura, se utiliza en muchos métodos de generación de caminos (Muñoz y otros, 1993; Segovia y Rombaut, 1993). Sin embargo, el aumento de vértices de control incrementa la complejidad computacional y, por tanto, la dificultad para la ejecución en tiempo real de los programas de generación de caminos. Por otro lado, la modificación de un solo vértice de control implica la reconstrucción de toda la curva.

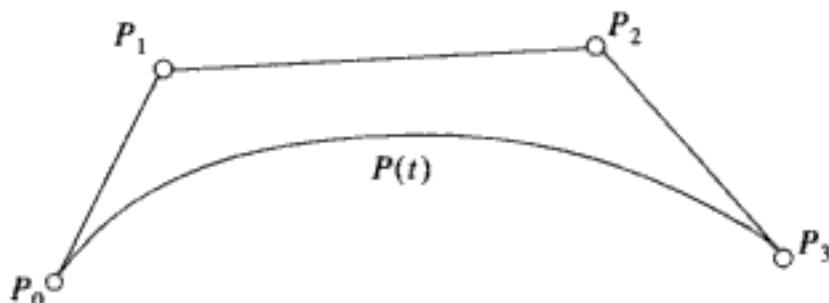


Figura 10.4: B-spline de cuarto grado y polígono de control.

Una clase de curvas que aúna las ventajas de los tipos comentados anteriormente permitiendo la construcción del camino con polinomios de orden reducido, así como la posibilidad de introducir vértices de control adicionales para la imposición de condiciones iniciales, es la denominada  $\beta$ -spline.

Las curvas  $\beta$ -spline se clasifican dentro de las funciones definidas por secciones cúbicas que garantizan la continuidad en sus uniones hasta la segunda derivada. El segmento  $i$ -ésimo de curva se define como:

$$P_i(\lambda) = \sum_{r=-2}^1 B_r(\beta_1, \beta_2, \lambda) V_{i+r}, \quad 0 \leq \lambda \leq 1 \quad (10.82)$$

Las funciones peso  $B_r$  dependen del parámetro de la curva  $\lambda$  y de dos valores adicionales  $\beta_1$  y  $\beta_2$ , denominados parámetros de forma, por afectar la topología de la curva. La continuidad en las uniones entre dos segmentos consecutivos se expresa mediante las siguientes relaciones:

$$\begin{aligned} P_{i+1}(0) &= P_i(1) \\ P'_{i+1}(0) &= \beta_1 P'_i(1) \\ P''_{i+1}(0) &= \beta_1 P''_i(1) + \beta_2 P'_i(1) \end{aligned} \quad (10.83)$$

La especificación de las funciones peso se realiza mediante polinomios cúbicos:

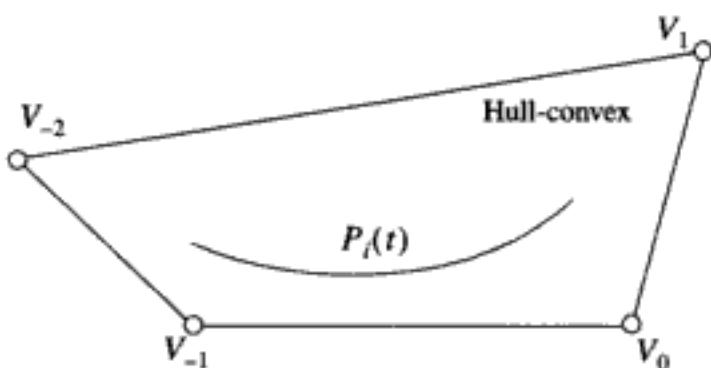
$$B_r(\beta_1, \beta_2, \lambda) = \sum_{g=0}^3 c_{gr}(\beta_1, \beta_2) \lambda^g \quad 0 \leq \lambda \leq 1 \quad r = -2, 1, 0, 1 \quad (10.84)$$

Los coeficientes  $c_{gr}$  de cada polinomio se calculan con objeto de que los segmentos de la curva  $\beta$ -spline garanticen en sus uniones las relaciones expresadas en (10.83). Existen algoritmos que permiten evaluar la curva, de forma computacionalmente eficiente, en los puntos intermedios (Barsky, 1988).

Una propiedad importante de las curvas  $\beta$ -splines para su empleo en la generación de caminos, es la posibilidad de controlar fácilmente la separación del polígono de control empleando para ello los parámetros  $\beta_1$  y  $\beta_2$ . En efecto, en otras curvas sólo es posible asegurar que estará contenida en la zona convexa de su polígono de control o "Hull-convex" (figura 10.5).

Por otra parte, la modificación en el polígono de control de la curva sólo requiere una reconstrucción de la curva local en los segmentos afectados (control local). Asimismo, el orden de la curva se mantiene constante e independiente del número de vértices de control. Esto permite el empleo de un mayor número de ellos en la construcción de la curva, permitiendo que el camino comience o termine con posiciones, orientaciones y curvaturas especificadas, sin incrementar de manera significativa el tiempo de cálculo.

Las propiedades mencionadas de las  $\beta$ -splines se deben a la sencillez y flexibilidad de la definición de las funciones peso, modeladas como polinomios de tercer orden (Barsky, 1988). Es muy frecuente utilizar  $\beta$ -splines cúbicas con valores de los pa-

Figura 10.5: Curva  $\beta$ -spline y propiedad "Hull-convex".

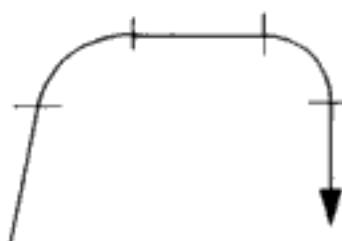
rámetros  $\beta_1 = 1$ ,  $\beta_2 = 0$ . Las curvas resultantes son fáciles de generar y la curvatura tiene una variación local aproximadamente lineal. La definición de las funciones peso de un segmento de curva  $\beta$ -spline, para estos valores de los parámetros, se muestra en la siguiente expresión matricial:

$$\begin{bmatrix} B_{-2}(\lambda) \\ B_{-1}(\lambda) \\ B_0(\lambda) \\ B_1(\lambda) \end{bmatrix} = \begin{bmatrix} \frac{1}{6} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{6} \\ \frac{2}{3} & 0 & -1 & \frac{1}{2} \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{6} \end{bmatrix} \times \begin{bmatrix} 1 \\ \lambda \\ \lambda^2 \\ \lambda^3 \end{bmatrix} \quad (10.85)$$

## 10.4 GENERACIÓN DE CAMINOS EN EL ESPACIO CARTESIANO

De acuerdo con lo indicado en el apartado 10.1, cabe distinguir entre la generación punto a punto y la de trayectoria continua.

En el primer caso se plantea directamente un problema de interpolación entre los puntos, mientras que en el segundo es necesario especificar qué tipo de trayectorias continuas deben emplearse. Las más frecuentes son las trayectorias en líneas rectas o las trayectorias circulares. Nótese como, en los robots móviles en el plano, en ambos casos se mantiene constante la curvatura del robot, siendo esta curvatura nula en el caso de la línea recta. Por consiguiente, si no se produjeran perturbaciones durante la ejecución de la trayectoria, no sería necesario modificar la variable de control. En la práctica siempre existirán ciertas perturbaciones con lo cual, si se pretende que un vehículo robótico siga una línea recta o un círculo, será necesario efectuar pequeñas correcciones en su dirección. El problema se plantea en los extremos de los tramos rectos o los círculos. En efecto, es frecuente que el camino que describe el robot esté



**Figura 10.6:** Camino en el plano formado con tramos rectos y circulares.

compuesto de una secuencia de tramos rectos y circulares tal como se muestra en la figura 10.6.

En el caso de puntos de unión de tramos rectos, no existe continuidad en orientación. En el caso de uniones entre tramos rectos y circulares, puede existir continuidad en orientación pero no en curvatura. Existen sistemas de control de robots móviles que se basan en la generación de trayectorias mediante concatenación de tramos rectos y circulares (Nelson y Cox, 1988). No obstante, la discontinuidad en la curvatura hace necesario un cambio brusco de la variable de control, que afecta la ejecución de la trayectoria. Este cambio es más aparente a medida que la velocidad es mayor.

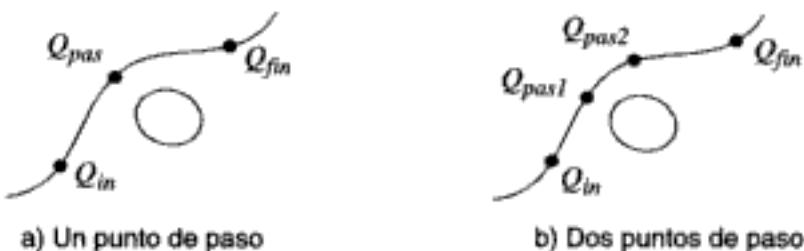
En efecto, nótese que si el robot no es omnidireccional y no realiza paradas en las cuales cambie de orientación y curvatura, el problema de la generación de trayectorias es necesario plantearlo de manera que se garantice que, en toda la curva, el camino es continuo en posición, orientación y curvatura. Asimismo, la curvatura debe estar limitada por el mínimo radio de curvatura del vehículo.

Para evitar estos problemas, que pueden ser importantes cuando se emplean vehículos convencionales, en los cuales las limitaciones de radio de curvatura son significativas, interesa emplear caminos con propiedades de continuidad en curvatura.

Considérese la generación de un camino que, partiendo de un punto inicial  $Q_{in}$ , llegue a otro punto final  $Q_{fin}$  pasando por un punto  $Q_{pas}$ . Este problema podría plantearse cuando se trata de que un robot móvil situado en el punto inicial, llegue al final evitando un obstáculo detectado mediante el sistema sensorial para lo cual se ha determinado el punto de paso, tal como se ilustra en la figura 10.7.

El problema se plantea estando fijadas normalmente la posición, orientación y curvatura en el punto inicial y final. Asimismo, en el punto de paso, además de la posición puede fijarse la orientación.

Un caso más general es cuando hay más de un punto de paso  $Q_{pas1}, Q_{pas2}, \dots, Q_{pasm}$  tal como se muestra en la figura 10.7 para dos puntos de paso.



**Figura 10.7:** Generación de trayectoria para evitación de obstáculo.

Los puntos de paso, normalmente, se definen con una cierta tolerancia con respecto a los obstáculos, con lo cual es frecuente que no se exija que el robot pase exactamente por el punto de paso, sino suficientemente próximo.

Por otra parte, en la generación de trayectorias para robots móviles, aunque no se esté considerando explícitamente la evolución temporal, es necesario tener en cuenta las restricciones cinemáticas del vehículo. Asimismo, si se considera la navegación de vehículos con velocidad elevada y masas importantes, resulta necesario tener en cuenta también restricciones impuestas por la dinámica del vehículo.

Las curvas splines presentadas en el apartado 10.3 son útiles para la generación eficiente de caminos de robots móviles. En particular, se han utilizado curvas de Bezier (Segovia y Rombaut, 1993), B-splines (Muñoz y otros, 1993) y  $\beta$ -splines (Muñoz, 1995) que permiten establecer condiciones de continuidad en la posición, orientación y curvatura. Los vértices se escogen de modo que la curva satisfaga las restricciones de los puntos inicial, final y de paso. Estas restricciones dan origen a un sistema de ecuaciones que debe resolverse para obtener las coordenadas de los vértices. Este sistema normalmente no está totalmente determinado, por lo que deben añadirse determinadas condiciones relacionadas con la orientación y curvatura en los puntos de paso o con la elección del punto final. Es posible también calcular los vértices para que la separación del polígono de control esté acotada y el valor de la curvatura sea inferior a un máximo (Muñoz, 1995).

Las B-splines y  $\beta$ -splines presentadas en el apartado 10.3 permiten hacer la curvatura continua a lo largo del camino. Es posible también conseguir variaciones de las curvaturas que se aproximan a las lineales por tramos. Las curvas clotoidales son una familia de curvas en las cuales la curvatura es continua y lineal. Esta propiedad es muy interesante para la generación de caminos (Shin, 1990). Sin embargo, la definición de estas curvas involucra integrales de Fresnel, y su generación en tiempo real es más compleja que las de las splines mencionadas.

## 10.5 GENERACIÓN DE TRAYECTORIAS PARA MANIPULADORES

---

Como se ha mencionado en la Introducción, el problema de la generación de trayectorias puede plantearse tanto en el espacio cartesiano como en el espacio de las articulaciones.

En el espacio cartesiano la especificación de caminos se realiza en términos de las posiciones y orientaciones de la herramienta  $\{H\}$  con respecto a un sistema de referencia  $\{P\}$  sin tener que realizar previamente la transformación cinemática inversa. Es decir, se trata de estudiar la evolución de las variables cartesianas en el tiempo.

Los métodos de generación en el espacio cartesiano tienen mayor complejidad ya que es necesario resolver en tiempo real el modelo cinemático inverso. Se consideran, en primer lugar, los movimientos rectilíneos ya que son los de especificación más usual.

Si se especifica la orientación mediante matrices en los puntos de paso, no es posible interpolar linealmente sus elementos ya que se generarían matrices de rotación no válidas. En efecto, las matrices de rotación deben tener columnas ortonormales, condición que no se garantiza si se interpolan linealmente elementos de dos matrices. Co-

mo se sabe, la posición y orientación suelen representarse mediante un vector de  $6 \times 1$ , con tres números especificando la posición y otros tres la orientación. Considérese un punto de paso definido como:

$$\begin{array}{c|c} {}^P_A T = \left[ \begin{array}{c|c} {}^P_A R & {}^P P_{ORG A} \\ \hline 000 & 1 \end{array} \right] \end{array} \quad (10.86)$$

transformación en la cual  ${}^P P_{ORG A}$  da la posición del origen del sistema  $\{A\}$  con respecto a  $\{P\}$  y  ${}^P_A R$  es una matriz de rotación que define la orientación del sistema  $\{A\}$  con respecto a  $\{P\}$ .

El vector  $6 \times 1$  correspondiente es:

$$\begin{array}{c} {}^P \chi_A = \left[ \begin{array}{c} {}^P P_{ORG A} \\ {}^P K_A \end{array} \right] \end{array} \quad (10.87)$$

expresión en la cual el vector  ${}^P K_A$  se forma escalando el vector unidad  ${}^P \hat{K}_A$  con los ángulos de orientación  $\theta_{PA}$ . La dificultad de estos métodos es la posible aparición de singularidades en el espacio cartesiano.

El problema es ahora emplear funciones de interpolación suave de movimientos entre puntos en el espacio de seis dimensiones definidas por  ${}^P \chi_A$ .

Una dificultad adicional es que la representación del ángulo no es única (Craig, 1986):

$${}^P K_A = ({}^P \hat{K}_A, \theta_{PA}) = ({}^P \hat{K}_A, \theta_{PA} + n360) \quad (10.88)$$

siendo  $n$  un número entero positivo o negativo.

Normalmente se elige la que haga más pequeña la rotación total. Si se desea ir desde el punto  $\{A\}$  al punto  $\{B\}$ , suponiendo que la orientación en  $A$  viene dada por  ${}^P K_A$ , se elige  ${}^P K_B$  para que se minimice  $|{}^P K_B - {}^P K_A|$ .

Existen diferentes problemas geométricos que limitan la aplicación de los métodos en espacio cartesiano. Entre estos casos cabe mencionar (Craig, 1986):

### I) Puntos intermedios inalcanzables

La situación se ilustra en la figura 10.8 para un manipulador con dos articulaciones. El problema es que el enlace 2 es más corto que el 1. Por tanto, el espacio de trabajo tiene un agujero en el centro cuyo radio es la diferencia de radios.

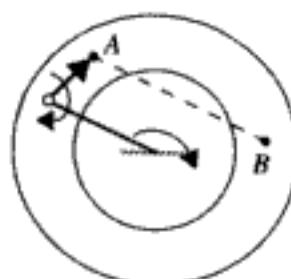


Figura 10.8: Camino con puntos intermedios no alcanzables.

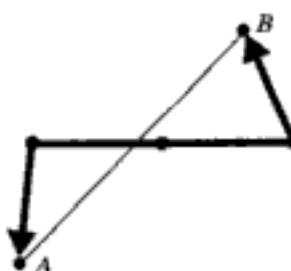


Figura 10.9: Camino con singularidades.

El movimiento de  $A$  a  $B$  es posible en el espacio articular. Sin embargo, la línea recta en el espacio cartesiano contiene puntos que no son alcanzables. Existen sistemas de control que notifican al operador esta circunstancia. En otros, comienza el movimiento hasta que se alcanza el límite de una articulación, momento en el cual se para.

### 2) Velocidades articulares altas cerca de las singularidades

La situación se ilustra en la figura 10.9. El manipulador está siguiendo una línea recta y aproximándose a una singularidad. En este caso, una o más velocidades crecen hacia infinito. Obviamente, existen limitaciones físicas que hacen que el manipulador se separe del camino deseado.

En efecto, en la figura 10.9 se trata de mover el extremo final del manipulador a velocidad lineal constante a lo largo de la línea recta. Todos los puntos son alcanzables pero, cuando el robot pasa por la mitad, la velocidad de la primera articulación es muy alta. Esta velocidad es tanto más alta cuanto más se aproxima el camino al eje de la primera articulación. Para resolver el problema puede limitarse la velocidad del camino al impuesto por el límite de velocidad de las articulaciones. Obviamente, las propiedades temporales se pierden.

### 3) Puntos alcanzables sólo con diferentes orientaciones

Considérese que se especifica un punto alcanzable en el interior del espacio de trabajo al cual sólo puede accederse con determinadas orientaciones. Si se especifica una orientación distinta, el planificador podría detectar el problema evitando que el controlador intente ejecutarlo.

Debido a todos los problemas mencionados en los párrafos anteriores, la mayor parte de manipuladores industriales emplea, en tiempo real, métodos de generación en el espacio articular.

---

## 10.6 TRAYECTORIAS ARTICULARES PARA MANIPULADORES ROBÓTICOS

---

En los métodos de generación de trayectorias en el espacio de las articulaciones no existe un control de posiciones/orientaciones cartesianas entre cada dos puntos de paso consecutivos.

Se trata de determinar una función suave de interpolación para cada articulación. Nótese que todas las articulaciones deben alcanzar los puntos de paso al mismo tiempo.

A continuación, se presentan diversos métodos para generación de trayectorias en el espacio de las articulaciones (Craig, 1986).

### 10.6.1 Empleo de polinomios cúbicos

Se trata de obtener un polinomio para cada articulación de forma que su valor para  $t_0$  sea el valor inicial  $q_0$  de la variable articular y su valor para  $t_f$  el valor final  $q_f$ . Se impone también la continuidad en la velocidad.

Por consiguiente, las restricciones son:

$$\begin{aligned} q(0) &= q_0 \\ q(t_f) &= q_f \\ q'(0) &= 0 \\ q'(t_f) &= 0 \end{aligned} \tag{10.89}$$

Obsérvese que se trata de un movimiento que termina parándose en  $t_f$  (velocidad final nula). Este problema se resuelve en el apartado 10.2. Los coeficientes del polinomio vienen dados por:

$$\begin{aligned} a_0 &= q_0 \\ a_1 &= 0 \\ a_2 &= \frac{3}{t_f^2}(q_f - q_0) \\ a_3 &= -\frac{2}{t_f^3}(q_f - q_0) \end{aligned} \tag{10.90}$$

#### Ejemplo 10.1

Se considera una única articulación rotacional y se trata de realizar un movimiento desde  $q_0 = 10$  hasta  $q_f = 90$  en 4 s. Se desea que las velocidades inicial y final sean nulas, por lo que es posible aplicar las expresiones anteriores para obtener:

$$\begin{aligned} a_0 &= 10, \quad a_1 = 0, \quad a_2 = 15, \quad a_3 = 2,5 \\ q(t) &= 10 + 15t^2 - 2,5t^3; \quad q'(t) = 30t - 7,5t^2; \quad q''(t) = 30 - 15t \end{aligned}$$

Empleando la herramienta MATLAB que se presenta en el apéndice, bastaría con escribir las siguientes líneas:

```
q = [10 90];
qd = [0 0];
```

```
t0 = 0; tij = 4;
```

```
[Qcoef, time, q_t, qd_t, qdd_t] = pol3 (q, qd, t0, tij, 0.001);
```

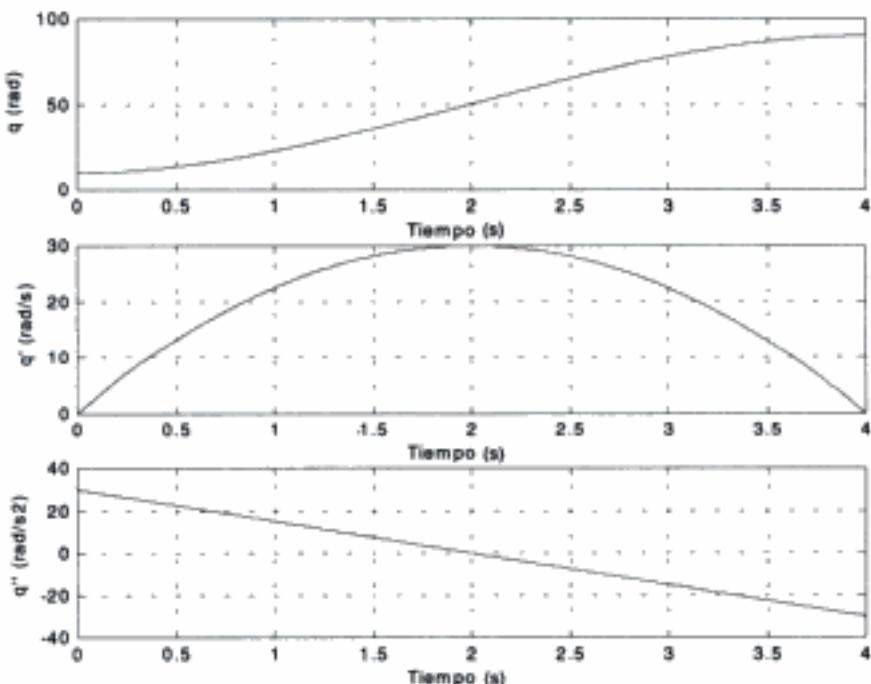
```
subplot (3,1,1),plot (time,q_t,'k'), xlabel ('Tiempo(seg)'), ylabel ('q(rad)'), grid
```

```
subplot (3,1,2),plot (time,qd_t,'k'), xlabel ('Tiempo(seg)'), ylabel ('q''(rad)'), grid
```

```
subplot (3,1,3),plot (time,qdd_t,'k'), xlabel ('Tiempo(seg)'), ylabel ('q'''(rad)'), grid
```

para obtener la representación de la figura 10.10, en la que se muestran las trayectorias correspondientes. La función `pol3` se incluye en la herramienta MATLAB que se describe en el apéndice. Las tres últimas líneas corresponden a la representación gráfica y son idénticas en todos los ejemplos posteriores, por lo que se omitirán en estos ejemplos.

En general, las funciones cúbicas tienen como perfil de velocidad una parábola y su perfil de aceleración es lineal (figura 10.10).



**Figura 10.10:** Representación gráfica de trayectorias de posición, velocidad y aceleración empleando una función cúbica en el ejemplo 10.1.

### 10.6.2 Empleo de polinomios cúbicos con puntos de paso

Se trata de un método que permite especificar movimientos, sin necesidad de pararse. Al igual que en el caso anterior, la especificación se efectuaría empleando posiciones y orientaciones de la herramienta. Las coordenadas articulares correspondientes se obtienen empleando la cinemática inversa.

En los puntos de paso se tendrán velocidades conocidas (no nulas) que corresponden a un paso sin parar:

$$q'(0) = q'_0, \quad q'(t_f) = q'_f \quad (10.91)$$

Es decir, igual que en el caso anterior pero con velocidades no nulas. Las ecuaciones son las mismas que se plantean en el apartado 10.3. Por tanto, la solución viene dada por:

$$\begin{aligned} a_0 &= q_0 \\ a_1 &= q'_0 \\ a_2 &= \frac{3}{t_f^2}(q_f - q_0) - \frac{2}{t_f}q'_0 - \frac{1}{t_f}q'_f \\ a_3 &= -\frac{2}{t_f^3}(q_f - q_0) + \frac{1}{t_f^2}(q'_f + q'_0) \end{aligned} \quad (10.92)$$

### Ejemplo 10.2

Se considera de nuevo una única articulación rotacional y se trata de realizar un movimiento desde  $q_0 = 10$  hasta  $q_2 = 20$  en 6 s, pasando por  $q_1 = 5$  en  $t = 2$  s. Asimismo, se desea que las velocidades inicial, de paso y final sean:  $q'_0 = 10$ ,  $q'_1 = 5$  y  $q'_2 = 20$ . Aplicando las ecuaciones (10.92) en cada uno de los tramos se obtiene:

$$\begin{aligned} a_{10} &= 10, \quad a_{11} = 2, \quad a_{12} = -(7,75), \quad a_{13} = 2,75 \\ a_{20} &= 5, \quad a_{21} = 4, \quad a_{22} = 0,5625, \quad a_{23} = -(0,15625) \end{aligned}$$

Para resolver el problema empleando MATLAB, habría que escribir:

```
q = [10 5 20];
qd = [2 4 1];
t0 = 0; tij = [2 4];
[Qcoef, time, q_t, qd_t, qdd_t] = pol3 (q, qd, t0, tij, 0.001);
```

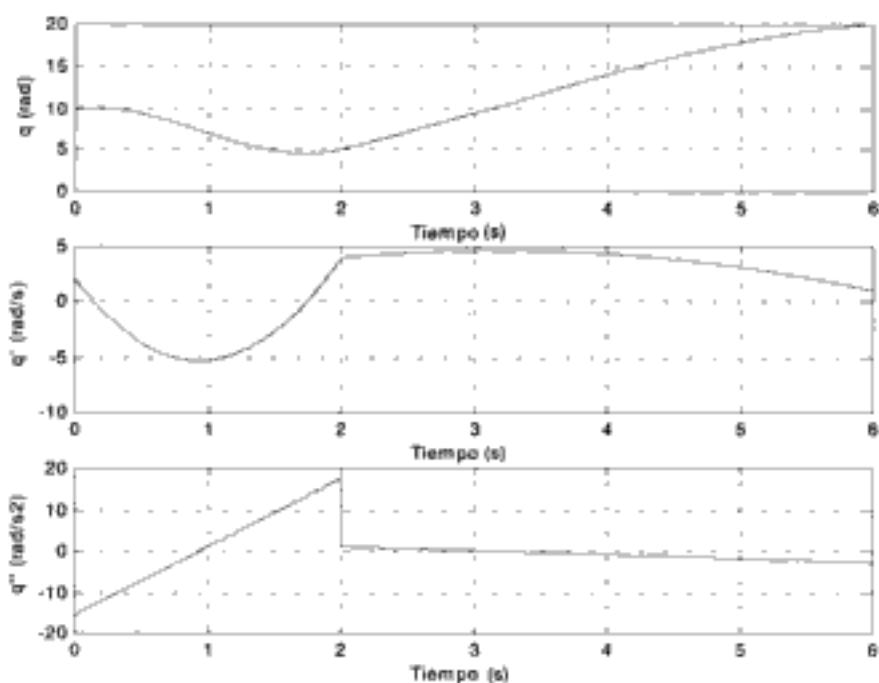
El resultado de la ejecución de estas líneas es el que se muestra en la figura 10.11.

Obsérvese que, cuando se concatenan diferentes tramos,  $q_f$  se convierte en intermedio. Por consiguiente, en este caso, el problema que se plantea es el de la elección de las velocidades deseadas en estos puntos intermedios.

Para ello existen diferentes técnicas alternativas entre las que cabe mencionar:

- 1) A partir de especificaciones del usuario de las velocidades cartesianas (lineales y angulares) de la herramienta en el punto intermedio.

En este caso, se emplea la inversa del jacobiano para calcular la velocidad articular en el punto de paso. Sin embargo, nótense que en los puntos en los que el jacobiano es singular, el usuario no puede realizar libremente la asignación de velocidades. Por tanto, el sistema debe incluir también alguno de los métodos siguientes.



**Figura 10.11:** Resultado del empleo de polinomios cúbicos con puntos de paso.

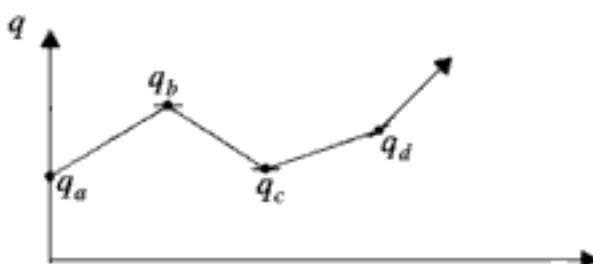
- 2) Mediante consideraciones heurísticas en el espacio de las articulaciones o en el espacio cartesiano. Así, si la pendiente de las líneas que unen los puntos cambia de signo, se elige la velocidad nula. En caso de que la pendiente no cambie de signo, se elige como velocidad la media de las dos pendientes. El método se ilustra en la figura 10.12.
- 3) De forma que la aceleración en los puntos de paso cumpla determinadas propiedades (continua, suave). En este caso, se cambian las restricciones de velocidad en la conexión por restricciones de velocidad continua y aceleración continua.

Considérese una trayectoria articular compuesta por dos tramos, con ángulo inicial  $q_0$ , ángulo de paso  $q_b$ , y ángulo final  $q_d$ . El primer tramo cúbico viene dado por:

$$q = a_{10} + a_{11}t + a_{12}t^2 + a_{13}t^3 \quad (10.93)$$

y el segundo por:

$$q = a_{20} + a_{21}t + a_{22}t^2 + a_{23}t^3 \quad (10.94)$$



**Figura 10.12:** Elección de pendientes.

Sean  $t_{f1}$  y  $t_{f2}$  los tiempos de finalización de los dos tramos. En este caso, pueden plantearse las ecuaciones:

$$\begin{aligned} q_0 &= a_{10} \\ q_v &= a_{10} + a_{11}t_{f1} + a_{12}t_{f1}^2 + a_{13}t_{f1}^3 \\ q_v &= a_{20} \\ q_f &= a_{20} + a_{21}t_{f2} + a_{22}t_{f2}^2 + a_{23}t_{f2}^3 \end{aligned} \quad (10.95)$$

Además, la velocidad inicial en el primer tramo y la final en el segundo son nulas y, por tanto:

$$\begin{aligned} 0 &= a_{11} \\ 0 &= a_{21} + 2a_{22}t_{f2} + 3a_{23}t_{f2}^2 \end{aligned} \quad (10.96)$$

Por otra parte, la velocidad final del primer tramo debe ser igual que la inicial del segundo:

$$a_{11} + 2a_{12}t_{f1} + 3a_{13}t_{f1}^2 = a_{21} \quad (10.97)$$

Asimismo, se considera la continuidad de la aceleración. Por tanto, la aceleración final del primer tramo debe ser igual que la aceleración inicial del segundo:

$$2a_{12} + 6a_{13}t_{f1} = 2a_{22} \quad (10.98)$$

Se tiene un problema con ocho ecuaciones y ocho incógnitas cuya solución para  $t_{f1} = t_{f2} = t_f$  es:

$$\begin{aligned} a_{10} &= q_0; \quad a_{11} = 0 \\ a_{12} &= \frac{12q_v - 3q_f - 9q_0}{4t_f^2}; \quad a_{13} = \frac{-8q_v + 3q_f + 5q_0}{4t_f^3} \\ a_{20} &= q_v; \quad a_{21} = \frac{3q_f - 3q_0}{4t_f} \\ a_{22} &= \frac{-12q_v + 6q_f + 6q_0}{4t_f^2}; \quad a_{23} = \frac{8q_v - 5q_f - 3q_0}{4t_f^3} \end{aligned} \quad (10.99)$$

### 10.6.3 Empleo de polinomios de orden superior

Si se desean tener en cuenta derivadas de orden superior, es necesario incrementar el orden del polinomio. En particular, se emplean polinomios quínticos para especificar posiciones, velocidades y aceleraciones al comienzo y final de cada segmento:

$$q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (10.100)$$

En este caso, las condiciones son:

$$\begin{aligned} q_0 &= a_0; \quad q_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \\ q'_0 &= q'(0) = a_1 \\ q'_f &= a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \\ q''_0 &= q''(0) = 2a_2; \quad q''_f = 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3 \end{aligned} \quad (10.101)$$

Estas restricciones especifican un sistema lineal de seis ecuaciones con seis incógnitas cuya solución es:

$$\begin{aligned} a_0 &= q_0; \quad a_1 = q'_0; \quad a_2 = \frac{q''_0}{2} \\ a_3 &= \frac{20q_f - 20q_0 - (8q'_f + 12q'_0)t_f - (3q''_0 - q''_f)t_f^2}{2t_f^3} \\ a_4 &= \frac{30q_0 - 30q_f + (14q'_f + 16q'_0)t_f + (3q''_0 - 2q''_f)t_f^2}{2t_f^4} \\ a_5 &= \frac{12q_f - 12q_0 - (6q'_f + 6q'_0)t_f - (q''_0 - q''_f)t_f^2}{2t_f^5} \end{aligned} \quad (10.102)$$

### Ejemplo 10.3

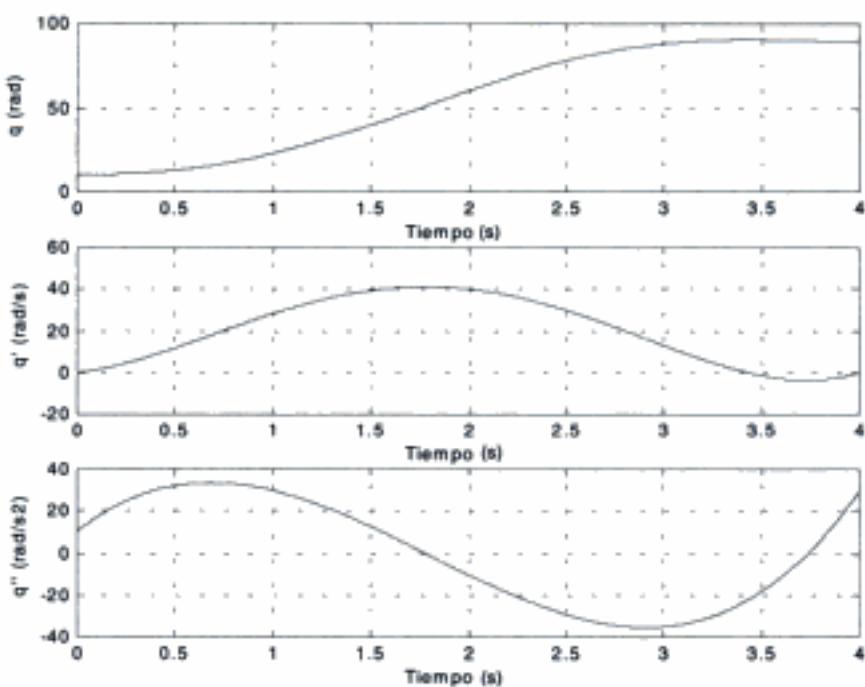
Se pretende utilizar un polinomio de orden cinco para cumplir las mismas especificaciones del ejemplo 10.1. Adicionalmente se desea que en los puntos inicial y final las aceleraciones sean  $q''_0 = 10$  y  $q''_f = 30$ .

Los coeficientes del polinomio se calculan mediante las expresiones de (10.102), resultando:

$$a_0 = 10, \quad a_1 = 0, \quad a_2 = 5, \quad a_3 = 12,5, \quad a_4 = -(5,625), \quad a_5 = 0,625$$

Para obtener la figura 10.13 con los resultados de la posición, velocidad y aceleración, se utilizan las siguientes líneas de MATLAB:

```
q = [10 90];
qd = [0 0];
qdd = [10 30];
t0 = 0; tij = 4;
[Qcoef, time, q_t, qd_t, qdd_t] = pol5 (q, qd, qdd, t0, tij, 0.001);
```



**Figura 10.13:** Empleo de polinomios de orden cinco para la obtención de trayectorias articulares en el ejemplo 10.3.

La función `pol5` se incluye en la herramienta MATLAB que se describe en el apéndice.

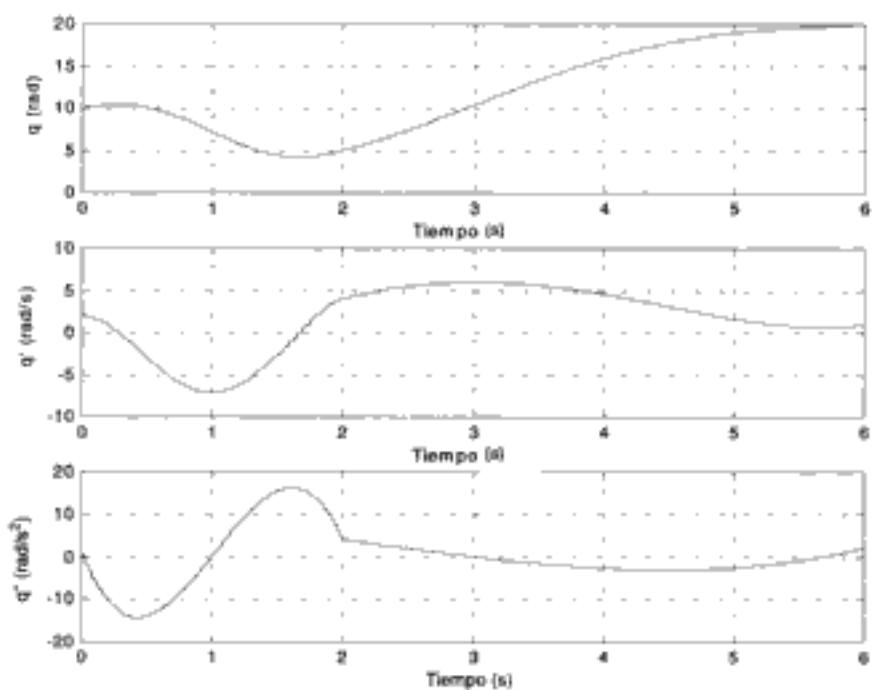
#### Ejemplo 10.4

En el ejemplo 10.2, se podía observar que la velocidad no era una curva suave y que la aceleración era discontinua (figura 10.11). Si se emplean polinomios de orden cinco, se pueden cumplir las especificaciones del ejemplo 10.2, y al mismo tiempo tener una curva suave para la velocidad y una aceleración continua. Asimismo, es posible especificar también las aceleraciones en los puntos inicial, final y de paso. Por ejemplo, para  $q''_0 = 1$ ,  $q''_1 = 4$  y  $q''_2 = 2$ , los coeficientes de los polinomios resultan ser:

$$\begin{aligned} a_{10} &= 10, \quad a_{11} = 2, \quad a_{12} = 0,5, \quad a_{13} = -13, \quad a_{14} = 9,5625, \quad a_{15} = -(1,875) \\ a_{20} &= 5, \quad a_{21} = 4, \quad a_{22} = 2, \quad a_{23} = -(0,6563), \quad a_{24} = -(0,0195), \quad a_{25} = 0,0137 \end{aligned}$$

Para hacer los cálculos con MATLAB, basta con escribir las siguientes líneas:

```
q = [10 5 20];
qd = [2 4 1];
qdd = [1 4 2];
t0 = 0; tij = [2 4];
[Qcoef, time, q_t, qd_t, qdd_t] = pol5 (q, qd, qdd, t0, tij, 0.001);
```



**Figura 10.14:** Utilización de polinomios de orden cinco para el cálculo de trayectorias articulares con puntos de paso en el ejemplo 10.4.

En las gráficas resultantes (figura 10.14), se pueden observar las características de suavidad y continuidad aludidas anteriormente.

Se emplean también técnicas para calcular funciones con variaciones suaves definidas por un cierto número de puntos. Entre estas funciones cabe mencionar las splines presentadas en el apartado 10.3.

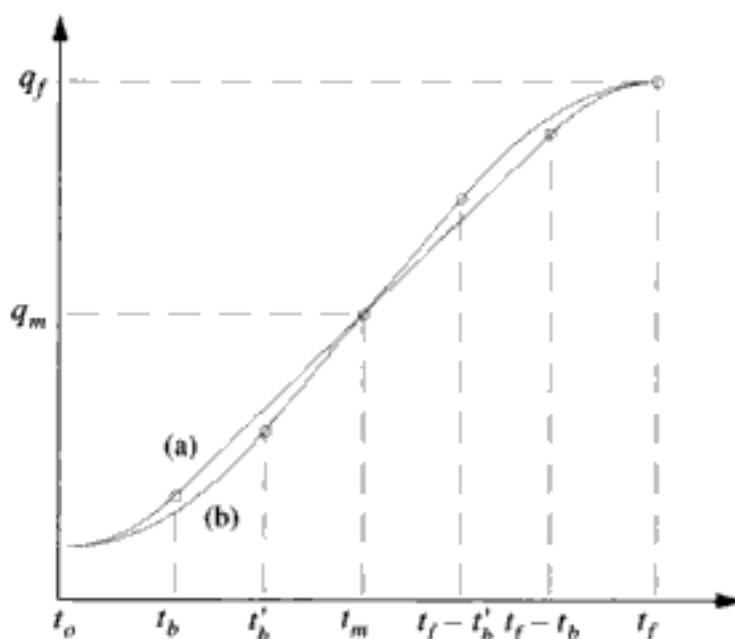
Otra técnica para lograr propiedades de continuidad es combinar polinomios de distinto grado. En el apartado siguiente se adopta este enfoque.

#### 10.6.4 Empleo de funciones lineales con enlace parabólico

La interpolación más simple del movimiento entre dos puntos del espacio articular es la lineal. Sin embargo, en este caso, la velocidad es discontinua en el comienzo y el final del movimiento. Para conseguir la continuidad en la velocidad y una transición más suave en la posición, se emplean enlaces parabólicos. En la figura 10.15 se muestran interpolaciones lineales con enlaces parabólicos en los extremos.

En el caso de la figura, se tiene velocidad inicial y final nula, y un enlace parabólico. El enlace parabólico da lugar a una aceleración constante. Existen diferentes soluciones de enlaces parabólicos, todas ellas simétricas con respecto al punto medio de la interpolación, que en la figura viene dado por  $t_m$ ,  $q_m$ . En la figura 10.15 se muestran dos de estas soluciones con duraciones del enlace parabólico  $t_b$  y  $t'_b$ .

Para resolver el problema se plantean las restricciones del enlace. Obsérvese en la figura que la velocidad al final de la región de enlace debe ser igual a la velocidad de la sección lineal. Por tanto, debe verificarse:



**Figura 10.15:** Interpolaciones lineales con enlaces parabólicos.

$$q'' t_b = \frac{q_m - q_b}{t_m - t_b} \quad (10.103)$$

siendo  $(t_b, q_b)$  el punto de enlace de la primera parábola con el tramo lineal y  $q''$  la aceleración que actúa en el tramo parabólico.

La ecuación del enlace parabólico es:

$$q_b = q_0 + \frac{1}{2} q'' t_b^2 \quad (10.104)$$

Para un tiempo total  $t_f = 2t_m$ , sustituyendo  $t_m = t_f/2$  y (10.104) en (10.103), y teniendo en cuenta que, por la simetría anteriormente mencionada, se verifica  $q_m - q_0 = (q_f - q_0)/2$ , se llega a:

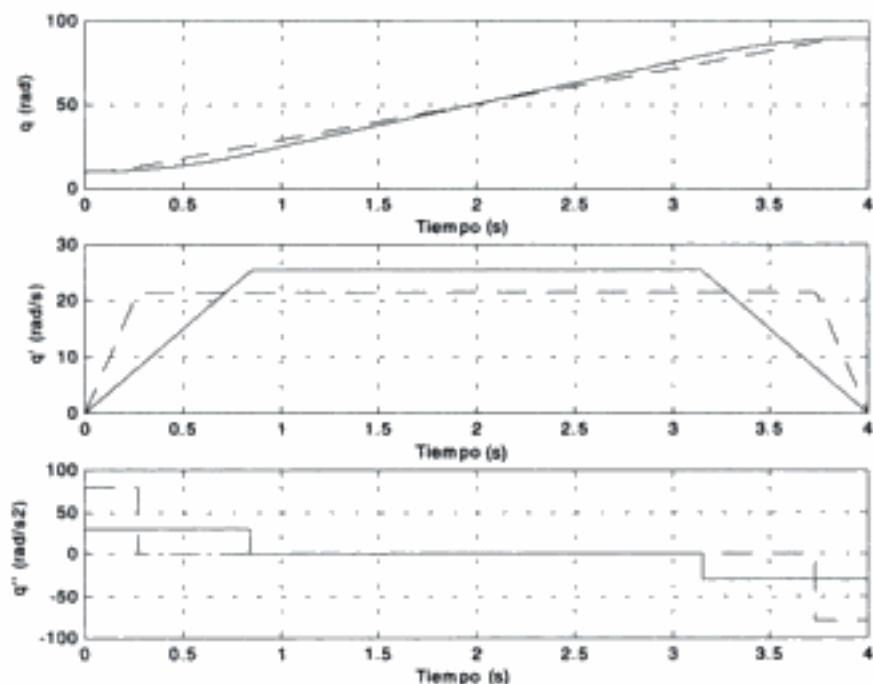
$$q'' t_b^2 - q'' t_f t_b + (q_f - q_0) = 0 \quad (10.105)$$

Dados  $q_f$ ,  $q_0$ , y  $t$  (duración deseada del movimiento) deben escogerse  $q''$  y  $t_b$  para satisfacer la ecuación. Normalmente se elige  $q''$  y se obtiene  $t_b$  mediante:

$$t_b = \frac{t_f - \sqrt{q''^2 t_f^2 - 4 q'' (q_f - q_0)}}{2 q''} \quad (10.106)$$

Para escoger la aceleración es necesario tener en cuenta la restricción:

$$q'' \geq \frac{4(q_f - q_0)}{t_f^2} \quad (10.107)$$



**Figura 10.16:** Enlaces parabólicos con aceleración 30 (línea continua) y 80 (línea discontinua).

Para el caso de la igualdad en (10.107), la porción lineal tiene longitud cero y el camino se compone de dos enlaces parabólicos que se conectan entre sí.

### Ejemplo 10.5

Para valores inicial y final de la variable articular  $q_0 = 10$  y  $q_f = 90$  respectivamente y una duración total de  $t_f = 4$  s, escogiendo una aceleración de  $q'' = 30$ , sustituyendo en (10.106) se obtiene un punto de enlace entre la parábola y la zona lineal en  $t_b = 0,845$  s. Esta zona lineal se mantendría hasta  $t_f - t_b = 3,155$  s y los últimos 0,845 s serían otra vez de parábola (figura 10.17).

Cuando la aceleración crece, el enlace parabólico disminuye. Así, por ejemplo, si la aceleración es de  $q'' = 80$ , sustituyendo en (10.106) se obtiene  $t_b = 0,268$ . Por tanto, el tramo lineal comienza antes (figura 10.17). Cuando la aceleración tiende a infinito, se tiende al caso de interpolación lineal simple sin enlace, con lo cual se presenta el problema de la discontinuidad.

Las líneas de MATLAB que permiten efectuar los cálculos para el primer caso son:

```

q = [10 90];
ac = [30 30];
t0 = 0; tij = 4;
[Qcoef, time, q_t, qd_t, qdd_t, interv] = parab (q, ac, t0, tij, 0.001);

```

La función `parab` se incluye en la herramienta MATLAB que se describe en el apéndice.

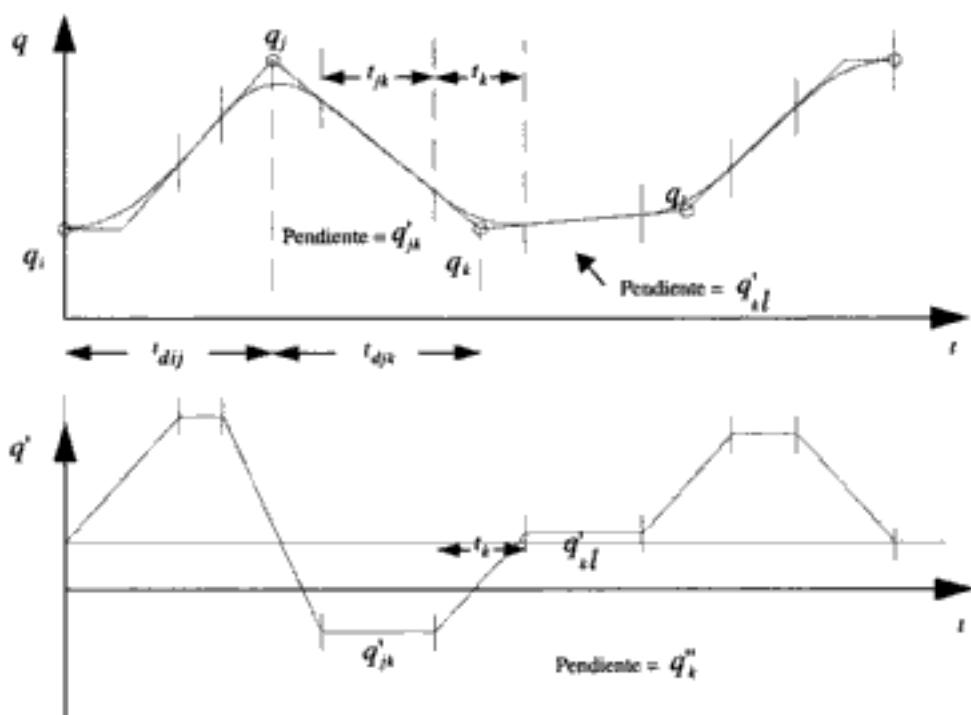


Figura 10.17: Segmentos lineales con enlaces parabólicos y múltiples puntos de paso.

Considérese ahora el mismo caso de funciones lineales con enlaces parabólicos, pero cuando se especifica un número arbitrario de puntos de paso. En la figura 10.17 existen tres puntos de paso consecutivos  $j, k, l$ . Sea  $t_k$  la duración de la región de enlace en el punto  $k$ . Sea  $t_{jk}$  la duración de la porción lineal entre los puntos  $j$  y  $k$ , y  $t_{djk}$  la duración total del segmento conectando  $j$  y  $k$ . La velocidad en la porción lineal es  $q'_{jk}$  y la aceleración en el enlace  $j$  la  $q''_k$ .

El problema tiene distintas soluciones dependiendo del valor de la aceleración en cada enlace. Dado un punto de paso  $q_k$ , la duración deseada del movimiento desde el anterior punto de paso  $t_{djk}$ , y la magnitud de la aceleración  $|q''_k|$ , es posible calcular los tiempos de enlace  $t_k$ . En puntos interiores del segmento lineal se tiene:

$$q'_{jk} = \frac{q_k - q_j}{t_{djk}} \quad (10.108)$$

La aceleración constante durante el enlace parabólico  $k$  es:

$$q''_k = |q''_k| \operatorname{sgn}(q'_{kl} - q'_{jk}) \quad (10.109)$$

y la duración de este enlace parabólico es:

$$t_k = \frac{q'_{kl} - q'_{jk}}{|q''_k|}. \quad (10.110)$$

Por tanto, la duración del segmento lineal  $jk$  será:

$$t_{jk} = t_{djk} - \frac{1}{2}t_j - \frac{1}{2}t_k \quad (10.111)$$

siendo  $t_j$  la duración del enlace parabólico en el punto de paso  $j$ .

Para determinar completamente la trayectoria se comenzaría con el primer segmento. Suponiendo conocidos  $q_1$  y  $q_2$ , la duración  $t_{d12}$  y el valor absoluto de la aceleración  $q''_1$ , es posible obtener:

$$\frac{\frac{q_2 - q_1}{1}}{t_{d12} - \frac{1}{2}t_1} = q''_1 t_1 \quad (10.112)$$

En esta expresión se ha tenido en cuenta que en el primer punto la línea recta no empieza en  $t = 0$ , sino en  $t = 1/2 t_1$  (figura 10.18).

De la expresión (10.112) se obtiene:

$$t_1^2 - 2t_{d12}t_1 = \frac{-2(q_2 - q_1)}{q''_1}$$

Sumando y restando  $t_{d12}^2$  en el lado izquierdo se llega a:

$$(t_1 - t_{d12})^2 = t_{d12}^2 - \frac{2(q_2 - q_1)}{q''_1}$$

de donde:

$$t_1 = t_{d12} - \sqrt{t_{d12}^2 - \frac{2(q_2 - q_1)}{q''_1}}, \quad (10.113)$$

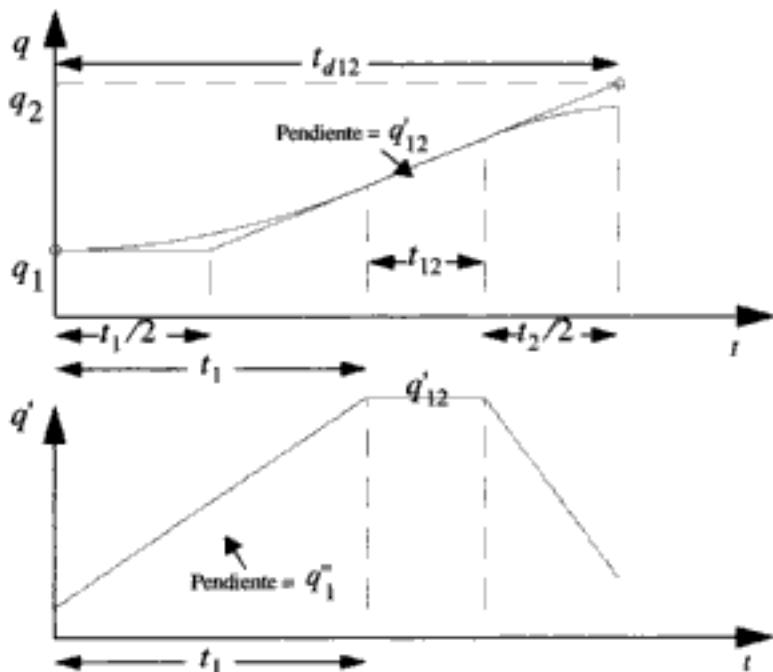


Figura 10.18: Primer tramo con enlace parabólico.

siendo:

$$q''_1 = |q''_1| \operatorname{sgn}(q'_{12} - 0) = |q''_1| \operatorname{sgn}(q_2 - q_1) \quad (10.114)$$

ya que la velocidad inicial es nula.

Asimismo, es posible determinar:

$$q'_{12} = \frac{\frac{q_2 - q_1}{1}}{t_{d12} - \frac{1}{2}t_1}, \quad (10.115)$$

y también:

$$t_{12} = t_{d12} - t_1 - \frac{1}{2}t_2 \quad (10.116)$$

A continuación se aplicaría (10.108) con  $j = 2$  y  $k = 3$  para obtener  $q'_{23}$  y a continuación (10.109) (el valor absoluto de la aceleración se supone conocido), con lo cual se obtendría  $q''_2$  y posteriormente se aplicaría (10.110) y (10.111) para obtener  $t_2$  y  $t_{12}$  respectivamente ( $t_{d12}$  se supone conocido). De esta forma se continúa hasta el último segmento, en el cual:

$$\frac{\frac{q_{n-1} - q_n}{1}}{t_{d(n-1)n} - \frac{1}{2}t_n} = q''_n t_n \quad (10.117)$$

En esta expresión hay que tener en cuenta que en el último punto la línea recta no termina en el punto final, sino cuando falta  $1/2 t_n$  (figura 10.19).

Asimismo, puede escribirse

$$q'_{(n-1)n} = \frac{\frac{q_n - q_{n-1}}{1}}{t_{d(n-1)n} - \frac{1}{2}t_n} \quad (10.118)$$

Teniendo en cuenta que la velocidad final es nula:

$$q''_n = |q''_n| \operatorname{sgn}(0 - q'_{(n-1)n}) = |q''_n| \operatorname{sgn}(q_{n-1} - q_n), \quad (10.119)$$

Por otra parte, desarrollando en (10.117) se obtiene:

$$t_n^2 - 2t_{d(n-1)n}t_n = 2 \frac{(q_n - q_{n-1})}{q''_n} \quad (10.120)$$

Sumando en cada lado  $t_{d(n-1)n}$  se llega a:

$$t_n = t_{d(n-1)n} - \sqrt{t_{d(n-1)n}^2 + \frac{2(q_n - q_{n-1})}{q''_n}} \quad (10.121)$$

Asimismo puede escribirse:

$$q'_{(n-1)n} = \frac{q_n - q_{n-1}}{t_{d(n-1)n} - \frac{1}{2}t_n} \quad (10.122)$$

ya que la última parábola, que acaba en  $q_n$ , tiene de duración  $t_n$  (figura 10.19). Normalmente el usuario especifica sólo los puntos de paso y la duración deseada de los segmentos. En este caso, se usan valores por defecto de las aceleraciones.

Finalmente conviene poner de manifiesto que la curva sólo se aproxima a los puntos "de paso" no pasando por ellos. Si las aceleraciones son altas, la aproximación es mejor. Obviamente, siempre es posible asegurar que se pasa exactamente por el punto escogiéndolo como si fuera el último, lo cual provocaría paradas intermedias.

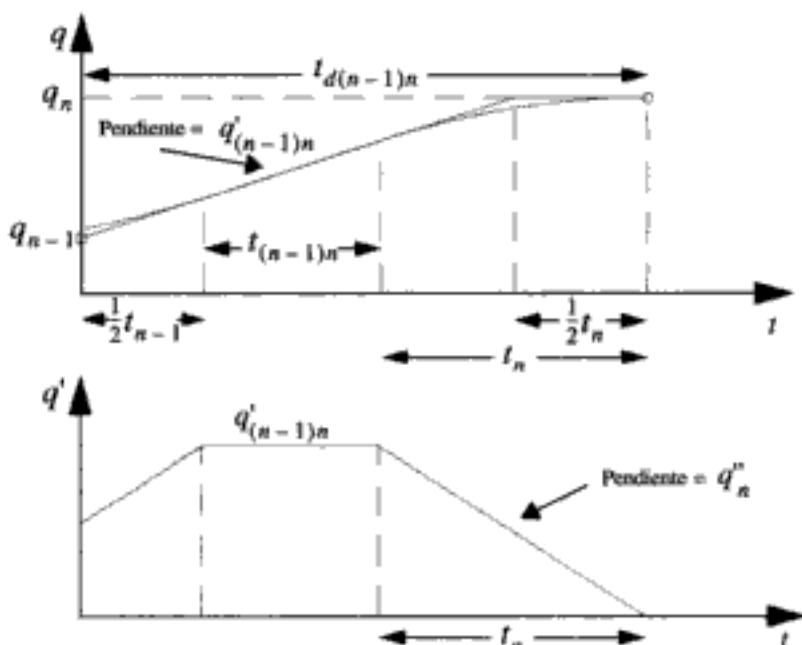


Figura 10.19: Último tramo con enlace parabólico.

### Ejemplo 10.6

Se trata de generar una trayectoria lineal con enlaces parabólicos que salga del punto de coordenada articular  $q_1 = 0$  grados y llegue al punto  $q_3 = 5$  grados pasando por el punto  $q_2 = 10$  grados. La duración especificada para cada uno de los dos tramos es de 1 segundo y la aceleración en los tres enlaces es de  $40$  grados/ $s^2$ .

Aplicando la expresión (10.114) se tiene:

$$q''_1 = 40 \operatorname{sgn}(q_2 - q_1) = 40 \operatorname{sgn}(10 - 0) = 40$$

La duración del tramo parabólico de arranque viene dada por (10.113):

$$t_1 = 1 - \sqrt{1 - \frac{20}{40}} = 0,293$$

La velocidad en el primer segmento lineal (pendiente) es (10.115):

$$q'_{12} = \frac{10}{1 - \frac{1}{2} 0,293} = 11,72$$

y su duración (10.116):

$$t_{12} = 1 - 0,293 - \frac{1}{2} t_2$$

que no puede resolverse todavía ya que no se conoce  $t_2$ .

En el último segmento se tiene (10.119):

$$q''_3 = 40 \operatorname{sgn}(10 - 5) = 40$$

La duración del último tramo parabólico es (10.121):

$$t_3 = 1 - \sqrt{1 + \frac{2(5-10)}{40}} = 0,134$$

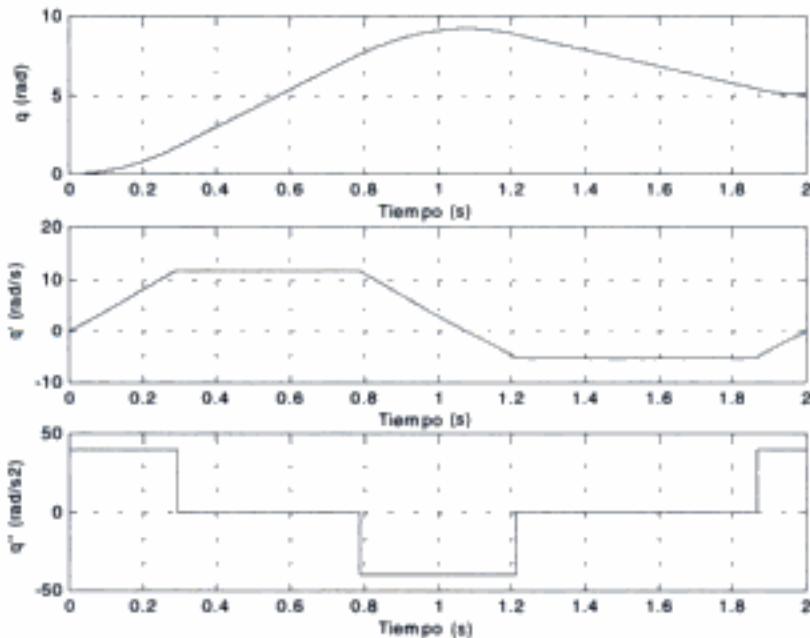
y la pendiente del último tramo lineal (10.122) es:

$$q'_{23} = \frac{-5}{1 - \frac{0,134}{2}} = -5,36$$

Ahora es posible terminar de resolver el tramo intermedio obteniendo (10.109):

$$q''_2 = 40 \operatorname{sgn}(-5,36 - 11,72) = -40$$

y finalmente (10.110):



**Figura 10.20:** Trayectorias de posición, velocidad y aceleración en el ejemplo 10.6.

$$t_2 = \frac{-5,36 - 11,72}{-40} = 0,427$$

Ahora ya es posible terminar de calcular la duración del primer y segundo segmento lineal:

$$t_{12} = 1 - 0,293 - \frac{0,427}{2} = 0,4935$$

$$t_{23} = t_{d23} - t_3 - \frac{1}{2}t_2 = 1 - 0,134 - \frac{0,437}{2} = 0,652$$

Para resolver el problema mediante MATLAB basta con escribir:

```
q = [0 10 5];
ac = [40 40 40];
t0 = 0; tij = [1 1];
[Qcoef, time, q_t, qd_t, qdd_t, intervals] = parab(q, ac, t0, tij, 0.001);
```

En la figura 10.20 se muestran las trayectorias correspondientes.

## 10.7 GENERACIÓN EN TIEMPO REAL

El resultado de la planificación en el espacio articular es un conjunto de datos de cada segmento de la trayectoria. Un generador de caminos articulares utiliza estos datos en tiempo real para calcular  $q$ ,  $q'$ , y  $q''$  enviando sus valores al control de las articulaciones.

En el caso en que se utilicen diferentes polinomios, es necesario tener en cuenta el tramo que se está ejecutando. Así, por ejemplo, cuando se emplean funciones lineales con enlace parabólico, en cada ciclo de cálculo el generador debe comprobar el tiempo transcurrido para determinar si se encuentra en el tramo lineal o en el parabólico. En el tramo lineal  $jk$  la trayectoria de cada articulación se calcula mediante:

$$q = q_j + q'_{jk} t, \quad q' = q'_{jk}, \quad q'' = 0 \quad (10.123)$$

siendo  $t$  el tiempo desde el paso por  $q_j$ , y  $q'_{jk}$  la velocidad calculada previamente mediante (10.108):

$$q'_{jk} = \frac{q_k - q_j}{t_{djk}} \quad (10.124)$$

En la región de enlace parabólico para llegar al punto de paso  $q_k$ , la trayectoria de cada articulación se calcula mediante:

$$t_{ent} = t - \left( \frac{1}{2}t_j + t_{jk} \right) \quad (10.125)$$

$$q = q_j + q'_{jk}(t - t_{ent}) + \frac{1}{2}q''_k t_{ent}^2 \quad (10.126)$$

$$q' = q'_{jk} + q''_k t_{ent} \quad (10.127)$$

$$q'' = q''_k \quad (10.128)$$

Expresiones en las cuales  $q'_{jk}$ ,  $q''_k$ ,  $t_j$  y  $t_{jk}$  se calculan mediante las ecuaciones de funciones lineales con enlaces parabólicos.

El algoritmo debe poner el tiempo  $t$  a  $1/2 t_j$  cada vez que se introduce un nuevo segmento lineal.

Los métodos de generación presentados anteriormente también pueden aplicarse para la generación de trayectorias en el espacio cartesiano (Craig, 1986). Así, cuando se emplea el método de las funciones lineales con enlaces parabólicos, se tendría una variación lineal de las componentes de la posición en el tramo lineal.

Para la generación en tiempo real podrían emplearse las mismas ecuaciones anteriores sustituyendo las variables articulares por las componentes del vector que indica la posición y orientación en el espacio cartesiano. En este caso, una vez determinada la trayectoria cartesiana, debe convertirse al espacio articular empleando modelos inversos, lo cual puede representar una considerable complejidad computacional, por lo cual se introducen simplificaciones.

## 10.8 CONCLUSIONES

La generación de trayectorias es una importante función del sistema de control de un robot. Estas trayectorias deben generarse a partir de especificaciones de un programador, o de un operador que actúa empleando un dispositivo de control. Es también posible que la generación de trayectorias se realice como última fase de un sistema de planificación automática de trayectorias.

En este capítulo se han presentado métodos de generación de trayectorias tanto para robots manipuladores como para robots móviles, aunque se pone un mayor énfasis en los robots manipuladores. En primer lugar, se han introducido conceptos básicos de geometría diferencial y de interpolación, incluyendo técnicas básicas de interpolación así como el empleo de funciones spline.

A continuación se han presentado nociones básicas sobre generación de trayectorias en el espacio cartesiano, considerando tanto robots manipuladores como móviles.

La última parte del capítulo se ha dedicado a la generación de trayectorias articulares para manipuladores robóticos incluyendo métodos de generación mediante polinomios cúbicos con puntos de paso y empleo de funciones lineales con enlace parabólico. El capítulo termina considerando aspectos relacionados con la generación en tiempo real empleando estos métodos.

## 10.9 REFERENCIAS

- Barsky B.A., 1988. "Computer graphics and geometric modelling using beta-Splines". Springer Verlag.  
 Craig J.J., 1986. "Introduction to robotics". Addison Wesley.

- de Boor C., 1978. "Computer graphics and geometric modelling using beta-splines". Springer-Verlag.
- Martínez J.L., A. Corral, y A. Ollero, 1991. "MICROVAL: Simulador del sistema y lenguaje VAL II del robot PUMA-500". Jornadas sobre Control y Robótica. Sevilla, enero de 1991.
- Mortenson M.E., 1985. "Geometric Modelling". Wiley.
- Muñoz V., 1995. "Planificación de trayectorias para robots móviles". Tesis Doctoral. Universidad de Málaga.
- Muñoz V., J. L. Martínez y A. Ollero, 1993. "New Continuous-Curvature Local Path Generators for Mobile Robots". Intelligent Components and Instruments for Control Applications", Editores: A. Ollero y E. F. Camacho, pp. 455-460. Pergamon Press, 1993.
- Nelson L.W. y I.J. Cox, 1988. "Local path control for an autonomous vehicle". Autonomous robot vehicles. Editores I. J. Cox y G.T. Wilfong. Springer-Verlag, pp. 38-44.
- A. Ollero, V. Muñoz, M. Prado y A. Simón, 1994. "Dynamic path planning and trajectory generation for mobile robots". Proceedings of the International Conference on Robotics and Automation". San Diego (California).
- Roger D.F. y J.A. Adams, 1989. "Mathematical elements for computer graphics". McGraw-Hill.
- Segovia A. y M. Rombaut, 1993. "Continuous curvature path-finding for a non-holonomic mobile robot". First International Workshop on Intelligent Autonomous Vehicles" (IAV'93), pp. 481-486.
- Shin D.H., 1990. "High performance tracking of explicit paths by roadworthy mobile robots". Ph.D. Thesis. Carnegie Mellon University.

# Programación de robots

---

En este capítulo se presentan diversos conceptos, métodos y técnicas sobre programación de robots, con especial énfasis en robots manipuladores industriales. En primer lugar, se trata la técnica de programación por guiado. A continuación, se consideran los lenguajes de programación textual con mayor atención a las instrucciones de especificación de movimientos, estructuras de datos, localización e interacción con el entorno. Como ejemplo de lenguaje de programación textual se emplea el VAL II. En la última parte del capítulo se considera brevemente la programación de vehículos robóticos.

## 11.1 SISTEMAS DE PROGRAMACIÓN DE ROBOTS

---

La programación de robots puede contemplarse en el marco de las arquitecturas de control que se presentaron en el capítulo 6. De esta forma, cabe hablar de programación a diferentes niveles que corresponden a los de la arquitectura de control de un robot.

Así, cuando existe un nivel superior de control con funciones que realizan de forma automática la planificación de caminos, la programación se reduce a especificar la tarea que es necesario ejecutar llamando a dichas funciones.

Sin embargo, si no se cuenta con la planificación automática de caminos, la programación debe realizarse indicando explícitamente la trayectoria que debe seguir el robot. De forma más precisa, si se pretende programar un robot manipulador, se indicaría el camino o la trayectoria que debe describir el sistema de referencia asociado al efecto final del robot. Asimismo, si se pretende programar un vehículo robótico, se indicaría la trayectoria que debe seguir el sistema de referencia situado en el punto de guiado del robot.

Nótese que, en el párrafo anterior, se ha mencionado la programación de trayectorias en el espacio de trabajo, normalmente espacio cartesiano. En este caso, se debe

contar con técnicas que generen automáticamente las trayectorias de las articulaciones. Es decir, deben existir métodos que implanten técnicas tales como las que se presentaron en el capítulo 10. Si el sistema de control no dispusiera de estas funciones, sería necesario que el programador especificara las trayectorias articulares, lo cual es notablemente más difícil para él.

Desde un punto de vista funcional, el sistema de programación del robot forma parte de un sistema informático que debe realizar un cierto número de tareas tales como: programación, control automático en tiempo real, comunicación con el operador, comunicación con periféricos y comunicación con otros equipos.

Normalmente, la programación del robot se realiza empleando un sistema informático para el desarrollo de programas. En el caso más simple es un editor de programas y un traductor. No obstante, en muchos sistemas, se incluyen también procedimientos de depuración de programas y tratamiento de errores de ejecución. De esta forma, si el programa origina que, en el movimiento entre un punto y otro especificados en el mismo, una articulación alcance un valor superior a su límite, se pararía el robot y se generaría una señal de error que pudiera ser leída por el operador. Esta condición de error puede ser corregida volviendo al modo editor y modificando el programa, o cambiando el punto al que se dirigía el robot.

De forma más precisa, existen sistemas para la edición y puesta a punto de programas que permiten:

- Ejecución paso a paso o/y en tiempo “ralentizado”.
- “Ejecución inversa” para retorno a condiciones previas y repetición de ejecución parcial.
- Gestión de trazas de ejecución y registro de la evolución de la información captada por sensores.
- Modificación de programas y variables en línea con continuación de la ejecución sin necesidad de reiniciar después de cada cambio (“hot editing”), incluyendo la definición interactiva de posiciones en el espacio empleando para ello gráficos y dispositivos especiales de guiado.
- Acceso simbólico al modelo del universo (si existe).

Como se sabe, la traducción de un lenguaje de programación de computadores puede realizarse mediante compiladores o intérpretes. Muchos de los lenguajes de programación son interpretados, con lo cual se facilita la depuración interactiva. De hecho, para conseguir las propiedades de edición y puesta a punto mencionadas anteriormente, es necesario utilizar traducción interpretada. Sin embargo, esto reduce las prestaciones durante la ejecución. Por ello, a veces se emplea un intérprete para escritura y puesta a punto y, posteriormente, un compilador para traducir el código ya verificado.

### 11.2 PROGRAMACIÓN POR GUIADO

---

La programación por guiado consiste en desplazar un sistema de referencia asociado al robot (en el efecto final de los manipuladores, o en el centro de guiado de los vehículos autónomos), de forma que se alcancen las configuraciones deseadas a la vez que se registran sus valores.

En otros términos, el robot actúa como un digitalizador de posiciones en un espacio con tantas dimensiones como articulaciones tiene, por las que se desea que pase en la fase de ejecución.

Otros autores utilizan términos diferentes para designar a la programación por guiado, tal como se ha definido anteriormente, o con ciertas modificaciones. Entre estos términos cabe mencionar los de "programación no textual", "programación por aprendizaje", "programación en línea" y "programación en modo enseñanza".

En los primeros tiempos de la robótica se distinguía entre "guiado pasivo" y "guiado activo". En el primer caso, el programador posicionaba el robot con sus propias manos y sólo se registraban las coordenadas articulares que posteriormente se convertían a tensiones de mando de los servomotores que controlan el robot. En el guiado activo, el robot se posiciona desde un puesto de mando, empleando para ello distintos dispositivos, registrándose durante la enseñanza las posiciones. Entre los dispositivos de guiado cabe mencionar teclado, botonera de programación, joystick y dispositivos especiales de guiado.

El dispositivo más utilizado en la programación por guiado de los robots manipuladores industriales es la botonera de programación. En la figura 11.1 se muestran algunas de ellas.

Existe también una técnica de guiado, que algunos autores incluyen dentro de la del guiado pasivo, que es el movimiento de una réplica del robot, de idéntica configuración pero más ligero y, por tanto, más fácil de mover.

Nótese que la réplica puede considerarse también como un dispositivo especial de guiado activo. En general, el dispositivo no tiene que tener la misma configuración que



**Figura 11.1:** Botoneras de programación de robots industriales.

el robot que se pretende programar. Se trata con ello de que el operador realice el guiado de la forma más cómoda posible. Obviamente, en este caso, es necesario disponer de los modelos cinemáticos de ambos para realizar las transformaciones oportunas.

Obsérvese también que, en vez de un dispositivo físico, es posible mover un robot virtual que se visualiza en la pantalla de un computador.

Los párrafos anteriores establecen una relación con los dispositivos que se emplean en teleoperación, y que se considerarán con más detalle en el capítulo dedicado a ésta.

Existen dos formas básicas de registrar los movimientos:

- Registro en puntos de paso. El robot se dirige hacia puntos consecutivos de paso en los que se registran las posiciones empleando pulsadores especiales. El sistema de control, durante la ejecución, interpola trayectorias en el espacio articular o en el cartesiano. Esta técnica se emplea en la programación de muchos robots industriales tales como los de ABB, empleando para ello la botonera de programación. En los robots Puma de Stäubli (antes Unimate) se emplea el pulsador RECORD de la botonera (figura 11.1) para registrar los puntos de paso.
- Registro continuo. Se muestran con una frecuencia fija los movimientos de guiado. La frecuencia depende de la precisión requerida en la aplicación. De esta forma, en aplicaciones de pintura, suele utilizarse una frecuencia elevada para obtener puntos muy próximos.

La programación por guiado es conceptualmente simple y no requiere el estudio de lenguajes de programación. Sin embargo, si no se utilizan réplicas o robots virtuales, es necesario ocupar a un robot durante el tiempo de programación.

Por otra parte, en la programación por guiado, existen problemas para incorporar la percepción del entorno y las estructuras de decisión. Asimismo, en algunas aplicaciones, es difícil que un operador posicione el robot con la suficiente precisión.

Para evitar estos problemas se han concebido sistemas de guiado avanzado o guiado extendido (Silva, 1984). En estos sistemas es posible definir saltos condicionales al estado de un sensor. De esta forma, por ejemplo, es posible la realización de movimientos precavidos (en inglés "guarded moves") en los que la condición de fin del movimiento es una señal que indica el contacto con un objeto.

En estos sistemas es posible también considerar movimientos relativos a una cierta referencia que puede ser modificada para la ejecución del programa. Asimismo, es posible definir subprogramas elementales, tal como sucede en la programación de los mencionados robots de ABB.

Otra técnica de programación consiste en que, tras la sesión de enseñanza, se genera un texto en un lenguaje de programación, tal como los que se consideran en apartados posteriores. De esta forma, es posible editar posteriormente el programa modificándolo o ampliándolo según la aplicación, introduciéndose saltos condicionales dependiendo de información sensorial, o realizando cálculos para obtener una posición de forma precisa. Este es el caso de lenguajes de programación de IBM tales como el Funky y el XPROBE.

Por último, cabe mencionar a los sistemas de guiado fuera de línea basados en la utilización de un sistema CAD que dispone del modelo de la tarea y del robot, y con el

que se simula la respuesta del robot a unas entradas procedentes de teclado o de dispositivos especiales. Estos sistemas son interesantes cuando es necesario involucrar elementos caros, de difícil utilización, o que todavía no existen físicamente. Como ejemplo cabe mencionar el sistema GRASP ('Graphical Robot Applications Simulation Package) desarrollado por BYG systems, o la herramienta MICROVAL (Martínez y otros, 1991) para el desarrollo de aplicaciones simples en VAL II con el PUMA. En Ollero y otros (1995) se presenta un sistema de teleoperación de manipuladores que permite generar automáticamente programas en VAL II empleando distintos dispositivos de guiado, observando los movimientos en un sistema de gráficos en 3D implantado en una estación de trabajo Silicon Graphics. El sistema incluye la verificación de los programas empleando modelos del manipulador y su entorno, e incorpora también planificación automática de caminos libres de obstáculos.

### 11.3 PROGRAMACIÓN TEXTUAL

En la programación textual se define la tarea mediante lenguajes que formalmente recuerdan a lenguajes de programación de computadores.

La definición de la tarea conduce a un texto de un programa en el que existen órdenes especiales para especificar los movimientos del robot, acceder a la información de los sensores, actuar sobre los objetos del entorno, etc.

La mayor parte de los lenguajes de programación de robots se definieron en los años setenta y ochenta y su sintaxis es similar a lenguajes tales como el BASIC o el Pascal.

La ventaja principal de la programación textual es que permite expresar la adaptación del comportamiento del robot ante imprecisiones y variaciones del entorno de trabajo medidas mediante sensores. Existe una gran variedad de lenguajes de programación de robots con diferentes características en lo que respecta a:

- Percepción: desde entrada binaria a visión artificial.
- Modelo: desde modelo del universo a no disponer siquiera de modelo cinemático, con lo cual es necesario expresar los movimientos en coordenadas articulares.
- Definición y generación de trayectorias: desde programación punto a punto a trayectorias interpoladas en el espacio cartesiano con propiedades de continuidad (capítulo 10).

De acuerdo con las arquitecturas de control de robots que se presentaron en el capítulo 6, existen diferentes niveles de programación textual. Una primera clasificación es en lenguajes de nivel accionador o articular, en los que la tarea se describe en términos del comportamiento de los accionadores, y nivel efector, en los cuales la tarea se describe en términos de los desplazamientos cartesianos y operaciones de los efectores, de forma independiente de las articulaciones. La práctica totalidad de los lenguajes actuales son del segundo tipo.

La clasificación más frecuente es en lenguajes de *nivel robot, objeto y tarea*, dependiendo de que las órdenes se refieran a movimientos a realizar por el robot, al esta-

do en que deben ir quedando los objetos o al objetivo que se pretende conseguir (Barrantes y otros, 1997).

La mayor parte de los lenguajes son de nivel robot, a los que también se denomina lenguajes de programación textual explícita, haciendo alusión a la necesidad de definir explícitamente las diferentes acciones.

La técnica de programación habitual consiste en que el programador establece la secuencia de operaciones preparando un texto y dejando para un posterior ajuste ciertos parámetros que dependen de información sensorial, de datos que se reciben por el sistema de comunicaciones e incluso de programación mediante guiado.

Normalmente, se emplean bibliotecas de funciones de nivel inferior, pero es el programador quien debe seleccionar las funciones necesarias y su secuencia, de acuerdo con la tarea a ejecutar y las restricciones del entorno.

Nótese que, para poder utilizar información sensorial, es necesario disponer de mecanismos generales de entrada/salida y de estructuras de control del flujo del programa. El programador puede hacer uso de esta información para iniciar y terminar movimientos, involucrando sincronización con los elementos del entorno, escoger entre acciones alternativas, identificar objetos y sus posiciones, y acomodarse a condiciones externas de forma activa (Silva, 1983).

Existen diferentes taxonomías de lenguajes de programación de *nivel robot* (Silva, 1984), tal como la propuesta por Bonner y Shin (1982) que distingue entre lenguajes de movimientos primitivos y lenguajes de programación estructurada. Los primeros tienen un nivel de potencia próxima al de la programación por guiado avanzado, de la que se distingue por tener características tales como: bifurcaciones simples, subprogramas sin transferencia de parámetros, capacidad sensorial limitada y paralelismo elemental. Estos lenguajes suelen disponer de bastantes órdenes, algunas demasiado similares, pero pocas estructuras de control. Entre éstos se encuentran AIVORAD, EMILY, RCL, RPL, SIGLA, VAL y MAL. El más conocido es el VAL que se empleó en los robots Unimate (posteriormente Stäubli).

Los lenguajes de programación estructurada se diferencian de los primitivos en que fueron desarrollados sobre una base de definición formal que les proporciona características tales como primitivas estructuradas, uso extensivo de transformaciones de coordenadas, definición de estructuras complejas de datos, definición por el usuario de subprogramas con parámetros y proceso paralelo estructurado y sincronizado (Silva, 1984). El empleo de múltiples sistemas de coordenadas y composición de transformaciones lleva a la utilización de formas embrionarias de modelo de universo que llegan hasta la definición de árboles. Sin embargo, el empleo generalizado de transformaciones hace más difícil la interpretación.

Entre los lenguajes de programación estructurada cabe mencionar HELP, MAPLE, MCL, PAL, LM, AL y AML. El lenguaje VAL II (Unimate, 1986), incorpora mayores capacidades de interacción con el entorno, cálculo, primitivas estructuradas, etc., que los lenguajes de movimientos primitivos, pero no reúne todas las características de los estructurados como el AL o el AML.

El lenguaje AL fue desarrollado en la Universidad de Stanford. Como lenguaje de programación tiene características de ALGOL y Pascal. Un programa en AL se traduce a otro lenguaje de bajo nivel que posteriormente se interpreta en tiempo real en el controlador. El AL permite definir especificaciones de *nivel robot*. Asimismo, es posible definir movimientos haciendo referencia a los objetos, aplicándose automáticamente

mente las transformaciones necesarias. Por tanto, tiene también características de los denominados lenguajes de *nivel objeto*.

El lenguaje AML (A Manufacturing Language) fue diseñado como un lenguaje estructurado para su ejecución en computadores IBM controlando robots de IBM. Desde el punto de vista de lenguaje de programación de computadores incorpora características de APL, Pascal y Lisp. Está compuesto de módulos para la interacción con el sistema, el control del manipulador y tareas en tiempo real. Una característica singular de AML es su capacidad para manejar agregaciones de datos, lo que facilita el tratamiento de vectores y diferentes sistemas de referencia.

El lenguaje V+ (Stäubli, 1997) fue desarrollado por la firma Adept Technologies y se emplea en la actualidad, además de en los robots Adepts, en algunos de los Stäubli, tales como en los Rx 60 y Rx 90. Este lenguaje incorpora las ventajas del VAL II presentando, además, mejores propiedades de fiabilidad, modificación de programas y transportabilidad.

Otro lenguaje moderno para la programación de robots industriales es el RAPID desarrollado para los robots de ABB. Es un lenguaje estructurado que permite utilizar fácilmente funciones y procedimientos. Los programas se dividen en módulos, uno de los cuales es el principal. Los módulos incluyen submódulos de datos y rutinas de ejecución (Barrientos y otros, 1997).

En los cuatro próximos apartados se indican las características principales de los lenguajes de programación de robots manipuladores. En estos apartados se utiliza como ejemplo el lenguaje VAL II. No obstante, muchos de los conceptos se expresan de forma parecida en otros lenguajes de programación. Conviene poner de manifiesto que no se pretende dar un curso del lenguaje, para lo cual se debe acudir a manuales apropiados. Por tanto, no se pretende ser exhaustivo ni se trata de forma detallada la sintaxis del lenguaje.

## 11.4 SISTEMAS DE REFERENCIA

El desarrollo de programas a nivel robot hace necesaria la consideración de sistemas de referencia. Normalmente existirá un sistema de referencia asociado a la base que está relacionado a través del modelo cinemático con el sistema de referencia asociado a la muñeca del manipulador sobre la cual se coloca la herramienta que se vaya a utilizar.

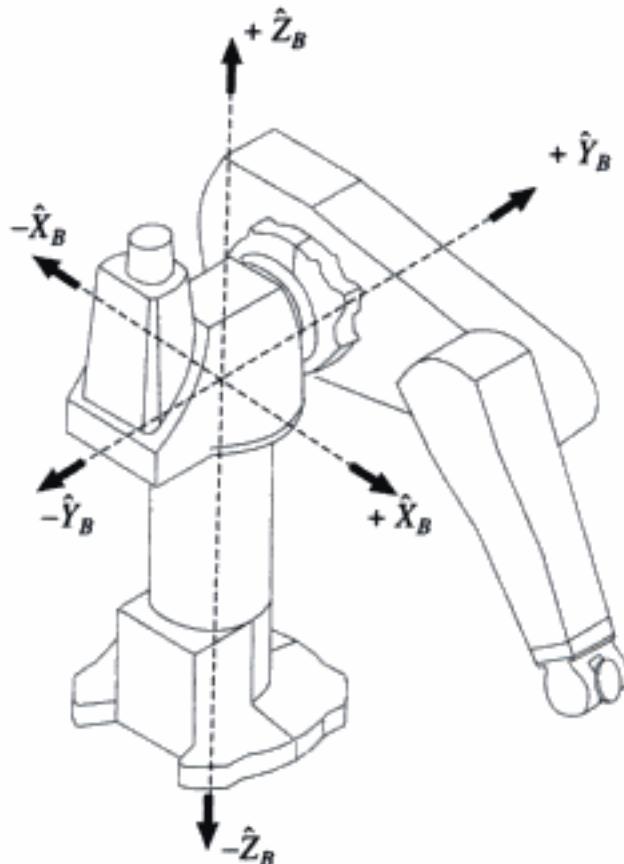
En los robots manipuladores Puma, los sistemas de referencia del brazo y del efecto final son los que se muestran en las figuras 11.2 (Staubli Unimation, 1990) y 11.3 respectivamente. Nótese que el sistema de referencia del robot se sitúa en la intersección de las articulaciones primera y segunda (figura 11.2). Este es el sistema de referencia que se adopta cuando se inicializa VAL II.

Sean  $\hat{X}_B - \hat{Y}_B - \hat{Z}_B$  los ejes del sistema de referencia asociado a la base y  $\hat{X}_M - \hat{Y}_M - \hat{Z}_M$  los ejes del sistema asociado a la muñeca o herramienta. En el lenguaje VAL II del robot Puma se emplean los ángulos denominados Orientación (O), Altitud (A) y Herramienta (T) definidos como sigue:

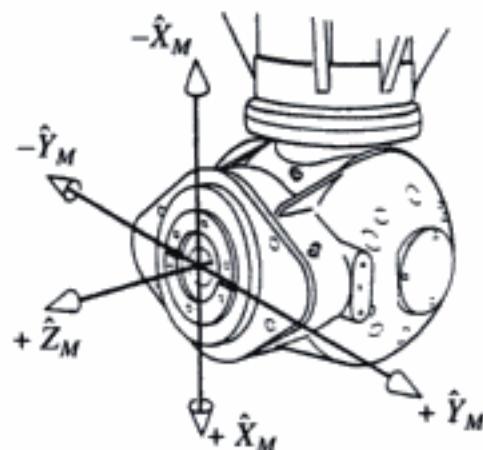
- El ángulo O es el formado por el eje  $\hat{X}_B$  con la proyección del eje  $\hat{Z}_M$  sobre el plano  $\hat{X}_B - \hat{Y}_B$ .

- El ángulo A es el que existe entre el eje  $\hat{Z}_M$  y un plano paralelo al  $\hat{X}_B - \hat{Y}_B$ .
- El ángulo T es el formado por el eje  $\hat{Y}_M$  de la herramienta con un plano paralelo al  $\hat{X}_B - \hat{Y}_B$ .

En la figura 11.4 (Staubli Unimation, 1990) se ilustran estos ángulos.



**Figura 11.2:** Sistema de referencia base en el Puma.



**Figura 11.3:** Sistema de referencia asociado al efecto final del Puma.

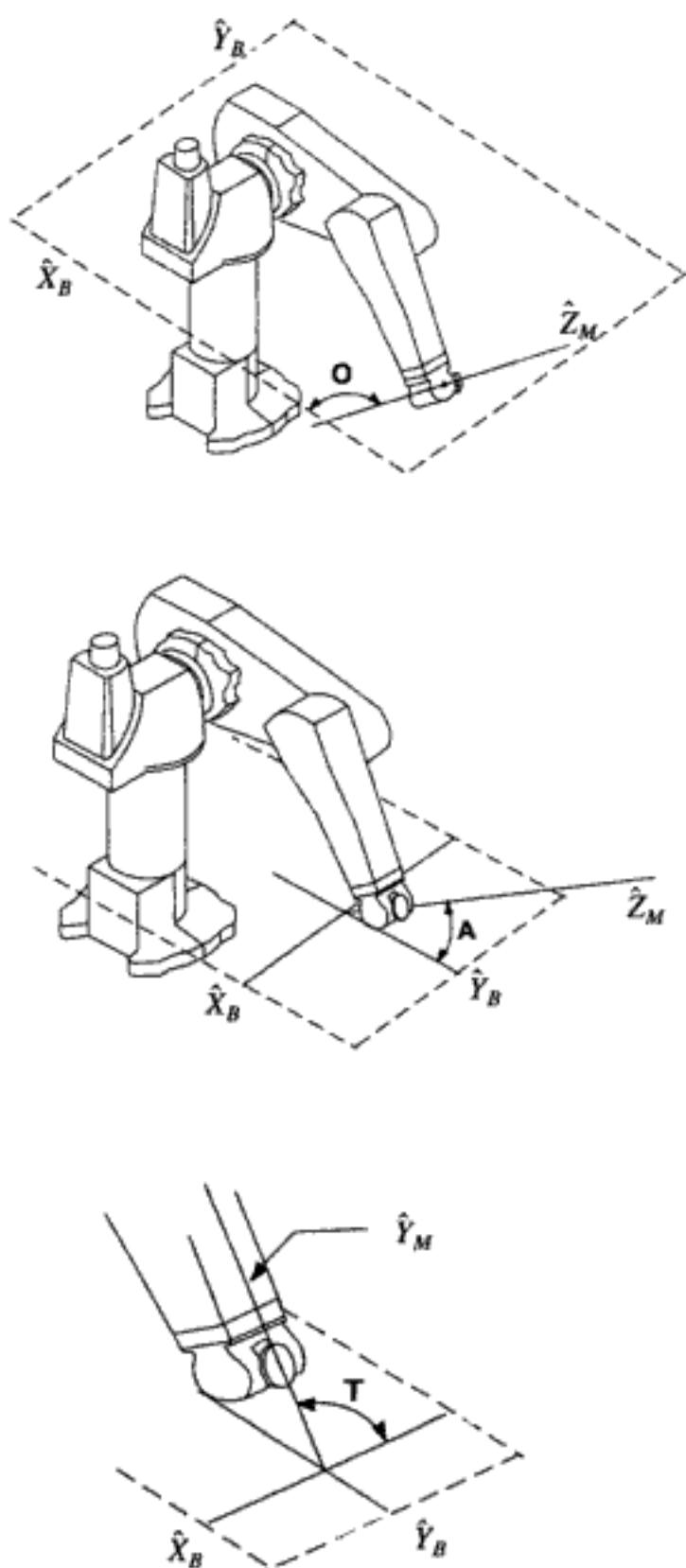


Figura 11.4: Definición de ángulos O-A-T.

En otros lenguajes la definición de los ángulos es diferente. Así, en el lenguaje V+ se definen los ángulos R, P, Y con relación al sistema de referencia asociado al efecto final.

## 11.5 ESPECIFICACIÓN DE MOVIMIENTOS EN ROBOTS MANIPULADORES

---

La sentencia más utilizada en la mayoría de los lenguajes es la de tipo MOVE. Cuando se escribe:

**MOVE A1**

se está especificando un movimiento del efecto final desde la localización actual a la previamente definida mediante la etiqueta A1. La denominación localización se empleará para considerar tanto posición como orientación. En todo lo que sigue se emplearán las mayúsculas para indicar las palabras clave empleadas en el lenguaje.

Normalmente MOVE origina un movimiento con interpolación articular. Existen otras posibilidades. Por ejemplo, en VAL II el movimiento en línea recta se consigue con:

**MOVES A1**

donde el carácter final "S" indica "straight line": movimiento en línea recta en el espacio cartesiano desde la localización actual a la A1. Durante el movimiento, la pinza de un robot manipulador se mueve en la dirección de la línea recta y es suavemente rotada a su orientación final.

Otro movimiento de interés específico es la aproximación y la salida de un punto. Se trata de efectuar un movimiento hasta llegar a una cierta distancia del punto. Ejemplo típico es la aproximación a una pieza antes de agarrarla. Así, en VAL II, para coger un objeto, suponiendo que la pinza está inicialmente abierta, se escribe:

**APPRO A1, 50**

indicando moverse hasta 50 mm encima de A1 (dirección negativa Z del sistema de referencia en el efecto final, que se muestra en la figura 11.3).

A continuación se especificaría:

**MOVES A1**

Posteriormente se cerraría la pinza (la instrucción se mencionará posteriormente), y a continuación se indicaría:

**DEPART 50**

que es una instrucción que indica la salida según el eje Z asociado al efecto final (pinza) también con valor del parámetro 50 indicando la distancia.

APPRO y DEPART se realizan con interpolación articular. Si se desea un movimiento en línea recta en el espacio cartesiano se utilizarían APPROS y DEPARTS.

Los movimientos anteriores se refieren a puntos definidos en forma absoluta en el espacio de trabajo. En diversos lenguajes es posible también especificar movimientos incrementales en los cuales es necesario definir la dirección y la distancia del movimiento.

Por otra parte, existen instrucciones para mover las articulaciones indicando magnitudes de movimiento lineal o angular. En VAL II se emplea la instrucción DRIVE. Así, por ejemplo,

**DRIVE 4 , -62.5, 75**

indicaría mover la articulación 4 un ángulo de giro -62,5 grados a una velocidad del 75% de la especificada en el monitor. Estas velocidades se establecen mediante instrucciones tales como la SPEED del VAL, mediante la cual se indica la velocidad a la que deben ejecutarse los movimientos posteriores, excepto que en la instrucción correspondiente al movimiento se indique otra cosa, tal como sucedía en la instrucción DRIVE anterior.

Desde el programa puede emplearse la instrucción SPEED con el formato:

**SPEED valor unidad [ALWAYS]**

donde *valor* puede ser una constante, una variable real, o una expresión aritmética. La *unidad* puede ser MMPS (milímetros por segundo) o IPS (pulgadas por segundo). En caso de no especificarse *unidad* se está indicando un porcentaje de la velocidad normal. De esta forma

**SPEED 20**

especifica una velocidad del 20% de la normal.

Cuando se especifica ALWAYS, la velocidad se fija hasta que no se cambie por otra instrucción. En caso contrario, la velocidad especificada sólo afecta al movimiento siguiente.

La instrucción

**COARSE [ALWAYS]**

se utiliza para realizar movimientos rápidos con menor precisión. Su ejecución hace que en los bucles de control de las articulaciones se permita un mayor error en régimen permanente. Para restaurar el movimiento preciso que es el que se emplea normalmente (por defecto), es necesario utilizar la instrucción FINE.

Por último, cabe mencionar las instrucciones que afectan al efecto final. En los lenguajes actuales existen órdenes, tales como las OPEN y CLOSE del VAL II, que abren y cierran durante la ejecución del próximo movimiento, mientras que OPENI y CLOSEI abren y cierran inmediatamente, tan sólo después de un pequeño tiempo de retraso que puede ser definido previamente como parámetro del sistema.

Si la pinza tiene servocontrol, pueden emplearse órdenes del tipo

**CLOSE 40 MM**

**CLOSE 1.575 IN**

que cierran la pinza hasta conseguir la separación especificada en milímetros o en pulgadas.

En algunas pinzas existen sensores de fuerza que permiten controlar el agarre. Así: CLOSE 3.0 LB

se utiliza para aplicar una fuerza de 3 libras sobre el objeto agarrado.

En VAL II existe una instrucción:

GRASP 40 *eti*

que además de la función de la CLOSE comprueba que la distancia es la especificada y, en caso contrario, transfiere el control a la instrucción con etiqueta *eti* en el programa.

## 11.6 ESTRUCTURAS DE DATOS

---

Son diferentes según los lenguajes. Las variables lógicas permiten tratar fácilmente operaciones de entradas y salidas binarias. Es bastante usual emplear operadores lógicos tales como AND, NOT, OR con los que se construyen expresiones. Las constantes octales y ASCII son de interés para trabajar con entradas y salidas binarias.

Algunos lenguajes, tales como el VAL, sólo permiten expresar variables numéricas enteras, mientras que en VAL II y en la mayoría de los lenguajes actuales también se permiten reales. Los operadores son los usuales: suma, resta, multiplicación, división, exponenciación y módulo. Existen también funciones matemáticas: trigonométricas, logarítmicas, exponenciales, valor absoluto, valor entero y raíces cuadradas.

En diversos lenguajes de programación es posible emplear estructuras ARRAY. En VAL II se pueden definir variables de tipo vector, con sólo una dimensión, para lo cual basta con añadir al final del nombre un índice entre corchetes.

En VAL II las variables se declaran al utilizarlas por primera vez, asignándoles un dato del tipo correspondiente. Se verá posteriormente que las variables pueden definirse tanto desde el terminal del sistema como durante la ejecución del programa.

Las estructuras más específicas de los lenguajes de programación de robots son las que se emplean para indicar localizaciones (posiciones y orientaciones) de objetos.

Las localizaciones pueden representarse en dos formas diferentes en VAL II. La primera es mediante los denominados *puntos de precisión*, que consisten en un conjunto de seis valores correspondientes a los ángulos en las seis articulaciones de un robot PUMA ( $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$ ). La ventaja de los puntos de precisión es que permiten conseguir una precisión alta y no existe ambigüedad sobre la configuración. Se recuerda que esta ambigüedad se presenta cuando se especifica en el espacio cartesiano y se resuelve el modelo inverso para obtener automáticamente las variables articulares. La desventaja es que estos puntos sólo sirven para un determinado robot. Por otra parte, cuando se producen cambios en el espacio de trabajo, su modificación en el programa es laboriosa.

La segunda representación consiste en indicar mediante seis valores la posición y la orientación en el espacio cartesiano. Para ello se emplean las denominadas transformaciones, que permiten una representación independiente del robot. La posición de la herramienta se define por las coordenadas x, y, z del origen de un sistema de referencia asociado a la herramienta y la orientación mediante tres ángulos ( $\phi, \theta, \psi$ ) que indi-

can la orientación de este sistema, todo ello expresado con referencia a los ejes de la base.

Las transformaciones son fáciles de modificar teniendo en cuenta consideraciones sobre el espacio de trabajo. De esta forma, por ejemplo, para expresar un desplazamiento según el eje X, sólo hace falta modificar la componente x. Un aspecto muy importante es la posibilidad de definir transformaciones como combinaciones de transformaciones, lo cual permite definir la posición de piezas con respecto a un soporte o una cinta, de forma que un cambio en la definición de la situación del soporte no exige la modificación en la posición de la pieza.

Entre los inconvenientes del empleo de las transformaciones cabe mencionar los problemas de control que plantea (modelo cinemático inverso con múltiples soluciones), tal como se trató en el capítulo 4. Por ejemplo, en el PUMA, una localización puede alcanzarse tanto desde un lado como desde el otro (brazo derecho o izquierdo) o desde arriba o abajo. Cuando se controla el robot desde VAL II, normalmente el sistema elige una configuración que podría no coincidir con la deseada.

Otra desventaja es la menor precisión resultante de los cálculos que realiza el sistema para obtener configuraciones.

Para manejar localizaciones (puntos de precisión o transformaciones) se emplean nombres simbólicos de variables. Existen nombres reservados por el sistema que no pueden utilizarse, tal como, por ejemplo en VAL II, los nombres BASE, DEST, EXCEPTION, TOOL,...

### Ejemplo 11.1

Considérese la transferencia de piezas situadas en un pallet de entrada a un pallet de salida. Parece conveniente definir el vector *pallet.ent[]* para representar la localización de las piezas en el pallet de entrada, y el vector *pallet.sal[]* para representar la localización de las piezas en el pallet de salida. Si *contador.ent* y *contador.sal* son las variables enteras que permiten indexar los vectores anteriores, y *altura.ent*, *altura.sal* son las alturas para aproximación y salida en los movimientos de agarre, podría escribirse el siguiente segmento de programa en VAL II:

```

OPEN
APPROS pallet.ent[contador.ent], altura.ent
SPEED 30
MOVES pallet.ent[contador.ent]
CLOSEI
DEPARTS 10
contador.ent= contador.ent -1

APPROS pallet.sal[contador.sal], altura.sal
SPEED 30

```

MOVES *pallet.sal[contador.sal]*

OPENI

DEPARTS 10

*contador.sal= contador.sal -1*

En VAL II las transformaciones compuestas se especifican mediante sus nombres separados mediante dos puntos (:). Así, por ejemplo, para agarrar un objeto en la localización *agarre* definida con respecto a *objeto*, el cual se define a su vez con respecto a *mesa*, se hace:

mesa: objeto: agarre

Obviamente, el orden es importante.

## 11.7 ESPECIFICACIÓN DE LOCALIZACIONES

---

Las localizaciones (posiciones y orientaciones) se definen utilizando las estructuras de datos específicas mencionadas en el apartado precedente.

En VAL, las localizaciones pueden especificarse tanto desde el monitor como desde el programa. En el primer modo, es posible utilizar órdenes en teclado en conjunción con enseñanza por guiado mediante, por ejemplo, una botonera de programación como las que se muestran en la figura 11.1. De esta forma, si se guía el robot hasta una determinada localización y se introduce la orden:

HERE *tr*

o bien:

HERE *pp*

siendo *tr* y *pp* nombres simbólicos, se define una transformación o un punto de precisión, asignándole el valor asociado a la posición actual del robot. Un punto de precisión se distingue de una transformación en que tiene el símbolo # antepuesto al nombre que lo identifica: HERE # *punto*.

Si se especifica una transformación compuesta, sólo se define la transformación más a la derecha. Así:

HERE *mesa: objeto: agarre*.

define *agarre*, produciéndose un error si *mesa* u *objeto* no están definidos previamente.

La función HERE se implanta leyendo los codificadores ópticos situados en los ejes de las articulaciones del robot. Si a su derecha se sitúa el nombre de una transformación, se emplea el modelo cinemático directo del manipulador para obtener las correspondientes coordenadas y ángulos.

Asimismo, en VAL II, es posible utilizar la instrucción:

TEACH *varpos*

para registrar un conjunto de localizaciones con el botón RECORD de la botonera. Cada vez que se oprime el botón se define una localización. De esta forma, es posible enseñar un camino completo posicionando el robot y presionando RECORD. Al igual que con HERE, si se especifica una transformación compuesta, sólo se suministran valores al nombre más a la derecha.

Si el nombre especificado no es un ARRAY, cada nombre de variable de localización da como resultado la concatenación del nombre de la base especificado en la instrucción con los incrementos sucesivos. Por ejemplo, si la orden es:

**TEACH P1**

el primer registro es P1, el segundo P2 y así sucesivamente. Si se especifica en la orden una transformación compuesta, sólo le afecta la que está más a la derecha.

Si en la instrucción se emplea un ARRAY:

**TEACH camino [ ]**

al ir presionando el botón se van creando *camino* [0], *camino* [1], y así sucesivamente.

Asimismo, para especificar posiciones en un programa en VAL II, puede emplearse la instrucción:

**SET coge = HERE: desplaza**

con la cual se hace que el valor de la transformación *coge* sea igual a la definida como la transformación *desplaza* relativa a la situación del robot (HERE).

Si a la izquierda se especifica una transformación compuesta, el valor sólo se asigna a la transformación más a la derecha.

Asimismo, si a la izquierda se indica una variable de transformación, a la derecha puede especificarse una transformación, una transformación compuesta, o una función de transformación.

La instrucción:

**SET # lugar = # post**

hace que el valor del punto de precisión *# lugar* sea igual al del punto de precisión *# post*.

Para especificar un punto de precisión como resultado de evaluar ciertas expresiones, es posible emplear la función PPOINT (*expr1, expr2, expr3, expr4, expr5, expr6*). De esta forma, puede escribirse:

**# lugar = PPOINT (x[0], x[1]+a, x[2]-a/2, x[3]+2, x[4], x[5])**

VAL II dispone de una serie de funciones de localización que facilitan el trabajo con las transformaciones. Entre éstas se encuentran las que se mencionan a continuación.

La función BASE devuelve un valor de transformación representando la traslación y rotación de la última orden o instrucción BASE. Como se ha mencionado anteriormente, cuando el sistema VAL II se inicializa, el sistema de referencia del robot se sitúa como se indica en la figura 11.2. La instrucción:

**BASE dx, dy, dz, rotación\_z**

permite modificar la referencia con un desplazamiento del origen (*dx, dy, dz*) y una ro-

tación alrededor de  $z$  con los grados especificados. Mediante esta instrucción se cambia la posición del robot sin tener que cambiar programas.

La función TOOL devuelve el valor de la transformación especificada en la última orden o instrucción TOOL. Asimismo, la orden:

**TOOL *trans***

permite modificar el sistema asociado a la herramienta. Si no se especifica ningún valor, la transformación es NULL, que tiene su centro en un sistema de referencia asociado al efecto final con el origen en la superficie en que se monta el efecto y sus ejes paralelos a los de la última articulación del robot. Se representa por (0, 0, 0, 90, -90, 0).

Si se emplea la sentencia:

**TOOL *transformación\_compuesta***

se hace que el valor de la transformación de herramienta sea igual que la especificada.

Por ejemplo,

**TOOL *pinza: llave***

reemplaza la transformación actual de la herramienta por el valor de la transformación compuesta *pinza: llave*.

La función DEST devuelve un valor de transformación representando la localización planificada para el próximo movimiento del robot.

La función FRAME ( $cp_1, cp_2, cp_3, cp_4$ ) devuelve un valor de transformación representando unos ejes de coordenadas en el cual el origen está definido por el punto  $cp_4$ , se toma como eje X positivo una línea paralela a la que pasa por los puntos  $cp_1$  y  $cp_2$  en la dirección de  $cp_1$  a  $cp_2$ , el eje Y es una recta perpendicular al eje X estando el sentido positivo definido por el punto  $cp_3$ , y el plano ( $x, y$ ) está definido por los puntos  $cp_1, cp_2, cp_3$ . Las  $cp_1, cp_2, cp_3, cp_4$  pueden ser a su vez transformaciones compuestas.

La función INVERSE ( $tr$ ) devuelve la transformación inversa de  $tr$ . Por ejemplo, si  $tr$  es una transformación que define la localización de un objeto relativo a las coordenadas del robot, INVERSE ( $tr$ ) define la localización del robot con referencia al objeto.

La función NULL devuelve un valor de transformación que corresponde a ( $x, y, z, O, A, T$ ) = (0, 0, 0, 90, -90, 0), vector nulo y sin rotación.

La función SCALE ( $tr$  BY *expresión*). Devuelve un valor de transformación en el cual el vector de posición está escalado por el valor de *expresión* y las componentes de rotación no se modifican. Por ejemplo, SCALE ( $tr$  BY 1.5), si  $tr$  vale (100, 200, 150, 90, -90, 0), devolvería (150, 300, 225, 90, -90, 0).  $tr$  puede ser también una transformación compuesta.

La función SHIFT ( $tr$  BY *expresión, expresión, expresión*) devuelve una transformación desplazada. Los argumentos omitidos se toman como nulos. Así, SHIFT (NULL BY 150, -100); devuelve (150, 0, -100, 90, -90, 0).

La función TRANS (*expr1, expr2, expr3, expr4, expr5, expr6*) crea una transformación a partir de la localización ( $x, y, z, O, A, T$ ). Un ejemplo de su empleo puede encontrarse en el segmento de programa VAL II:

```

FOR angle = start TO last+start STEP angle.step
  x= radius * COS (angle)+x.center
  y= radius*SIN(angle)+y.center
  MOVE cframe:TRANS(x,y,0,90,-90,0)
END

```

En este segmento de programa se obtienen las coordenadas de los puntos de un círculo y se crea con ellas una transformación que se aplica con relación al sistema de referencia cframe.

Nótese que la utilización de sentencias con las funciones BASE, DEST, FRAME, INVERSE, NULL, SHIFT, SCALE, TRANS, PPOINT permite establecer un modelo del entorno en el cual quedan determinadas la posición y orientación de los objetos.

Por último, cabe reiterar que la utilización de sistemas de coordenadas relacionados mediante transformaciones, aunque muy extendido, tiene dificultades tales como el que la configuración del robot no quede determinada de forma única mediante puntos en el espacio cartesiano (resolución no única del modelo cinemático inverso). Por otra parte, cuando el número de objetos se hace grande, las relaciones entre los sistemas de referencia se hacen complicadas, creciendo el número de cálculos.

## 11.8 INTERACCIÓN CON EL ENTORNO Y CARACTERÍSTICAS TIEMPO REAL

El control posicional puro sin percepción del entorno requiere:

- Elevada precisión y repetibilidad del manipulador, que en caso contrario podría superar la tolerancia de las tareas a realizar.
- Ubicación de objetos perfectamente conocida: entorno muy estructurado y alimentadores con cadencia conocida.

Las entradas más simples son señales lógicas correspondientes a contactos fin de carrera, o a sensores que indican, por ejemplo, la presencia o ausencia de un objeto en el alimentador. Así, el movimiento de aproximación de un manipulador a un objeto puede definirse mediante una trayectoria del órgano terminal y una condición binaria de parada. Los denominados *movimientos precavidos* ("guarded moves") son aquellos que están definidos por una condición sensorial de parada.

Por otra parte, la interacción con el entorno requiere que el sistema de control del robot pueda suministrar señales a dispositivos para tareas tales como poner en marcha o parar un alimentador. Los sistemas de control de robots disponen típicamente de entradas y salidas digitales. En el lenguaje existen instrucciones que permiten tratar las correspondientes señales lógicas.

De esta forma, en VAL, existen las instrucciones SIGNAL para activar o desactivar líneas de salida, e instrucciones WAIT para esperar hasta que se produzca un evento detectado por una señal de entrada.

En general, es posible, con una única SIGNAL, activar o desactivar diferentes líneas de entrada. En VAL se emplea:

SIGNAL *canal 1, canal 2, ..., canal n*

afectando los parámetros con el signo + o con el signo - según se pretenda activar o desactivar el canal. Así, podría escribirse:

SIGNAL 1, -2

para poner el canal 1 en ON y el canal 2 en OFF. Existen también sistemas con entradas y salidas digitales en las que pueden aplicarse también instrucciones tipo SIGNAL tales como:

SIGNAL 105, 4.5

que especifica un valor de 4.5 unidades (típicamente voltios) en el canal de salida 105.

La instrucción:

WAIT *expresión*

hace que el programa entre en un bucle de espera hasta que el resultado de la expresión sea TRUE (diferente de cero).

La instrucción SIG del VAL II se emplea para comprobar el estado de señales binarias, devolviendo un valor TRUE (numéricamente 1) o FALSE (numéricamente 0) de la señal que se indique. Por ejemplo, la instrucción:

SIG (1001)

devolverá el valor TRUE si la señal 1001 está en ON, y el valor FALSE si está en OFF. La instrucción SIG admite también como argumentos valores negados. Así, la instrucción:

SIG (-1004)

devuelve el valor TRUE si 1004 está en OFF. Si en vez de una señal se indican varias separadas por comas, se devuelve el valor lógico del resultado de efectuar un AND de todas las señales. Por ejemplo, la instrucción:

SIG (1001, 1004)

devuelve el valor TRUE si tanto 1001 como 1004 están en ON.

La instrucción:

WAIT SIG (1001)

hace que se produzca una espera hasta que la señal de entrada 1001 esté en ON, y la instrucción:

WAIT SIG (1001,-1004)

producirá una espera hasta que la señal 1001 esté en ON y la señal 1004 en OFF.

En VAL II pueden emplearse también instrucciones como:

DAC *ch = val*

que asigna al canal analógico *ch* el valor definido por *val*.

Asimismo, en vez de dar el número del canal de salida, puede ser conveniente emplear variables, tal como se realiza en las instrucciones:

**DEFINE MOTOR1 = OUTPORT 5**

**DEFINE SENS3 = INPORT 15**

...

**SIGNAL MOTOR1, ON**

**WAIT SENS3, ON**

...

**SIGNAL MOTOR1, OFF**

**WAIT SENS3, OFF.**

Otro ejemplo sería el conjunto de instrucciones:

**DEFINE VOLT1=OUTPORT 105**

...

**SIGNAL VOLT1**

entre las cuales podría realizarse el cálculo de VOLT1. Asimismo, las instrucciones:

**DEFINE VOLT3=INPORT 115**

...

**WAIT VOLT3**

podrían utilizarse para esperar a que la entrada tenga un valor superior a VOLT3.

En VAL II existen también instrucciones que permiten poner a 1 o a 0 una o varias señales binarias. RESET se emplea para poner a cero todas las señales de salida:

**RUNSIG *sif***

se emplea para poner a 1 la señal de salida *sif* mientras dure la ejecución del programa en curso. Cuando finalice se pone a cero automáticamente.

Además del tratamiento de entradas y salidas, es necesario contemplar otras características específicas de *tiempo real* tales como la sincronización con el entorno y la concurrencia.

Obsérvese cómo determinadas instrucciones de entradas y salidas tales como las:

**WAIT *canal***

del VAL II permiten realizar sincronización con el entorno deteniendo el flujo del programa hasta que en la línea digital de entrada se adopte el valor indicado en la especificación del canal (OFF si se afecta el canal con signo - y ON si es +).

Asimismo, en VAL II puede emplearse la instrucción:

**DELAY *t***

para detener la ejecución del programa durante *t* segundos.

En general, en los lenguajes de programación de robots, el tratamiento de interrupciones y concurrencia es bastante pobre, disponiéndose tan sólo de un sistema de gestión de rutinas asíncronas para desencadenar programas de acuerdo con el estado de señales externas.

No obstante, en algunos lenguajes, tales como VAL II, es posible definir prioridad relativa entre las diferentes rutinas. La instrucción:

**LOCK <pr>**

permite establecer la prioridad del segmento de programa al cual pertenece esta instrucción como igual al valor pr en el rango [0,127].

En la mayoría de los lenguajes existen instrucciones tipo STOP y PAUSE. En VAL la instrucción:

**STOP mensaje**

detiene la ejecución y muestra por pantalla el mensaje; si hubiera ejecuciones pendientes el control pasa a la siguiente. Existen también lenguajes con posibilidad de diversas instrucciones de parada según sea parada inmediata, parada del manipulador pero continuación con el programa para realizar otras operaciones con objetos del entorno, o parada después del movimiento.

En VAL II puede emplearse también la instrucción BREAK que tiene dos efectos: interrumpir la ejecución del programa hasta que el robot alcance el destino que se le ha ordenado, e interrumpir la transición continua entre el movimiento actual y el que se ha ordenado con la siguiente instrucción de movimiento.

La instrucción BRAKE hace que el movimiento del robot pare de forma inmediata.

En VAL es posible utilizar para las pausas la instrucción:

**PAUSE mensaje**

que detiene la ejecución del programa y muestra un mensaje por consola siendo posible reactivar el programa mediante una orden en el monitor.

Otro aspecto de interés es el tratamiento de las excepciones: rotura o deformación de herramientas, objetos defectuosos,...

En los lenguajes de programación de robots, la gestión de excepciones está poco desarrollada. No obstante, a veces se pueden definir procesos de monitorización de eventos a los que se asocia un nombre de procedimiento para su manipulación.

En VAL II existen las instrucciones:

**REACT canal, [nsubprog] [ALWAYS]**

que monitoriza el estado del canal digital de entrada. Si *canal* es positivo y se detecta una transición desde OFF hasta ON, se transfiere el control al subprograma *nsubprog*. En caso de que *canal* tenga signo negativo, se busca la transición de ON a OFF. La instrucción se ejecutará en el transcurso del próximo movimiento. De no estipularse ALWAYS, la instrucción sólo es válida en el transcurso del próximo movimiento.

La instrucción:

**REACTI canal, [nsubprog] [ALWAYS]**

difiere de la anterior en que el control se transfiere inmediatamente y no en el próximo movimiento.

Las instrucciones REACT y REACTI asociadas a un determinado canal pueden inhibirse mediante la:

**IGNORE canal [ALWAYS]**

De no estipularse ALWAYS, la instrucción será válida sólo en el transcurso del próximo movimiento que ejecute el robot.

Otra instrucción del VAL II es:

**REACTE [nprog]**

En caso de que se produzca un error (error de hardware, aritmético, etc.), se ejecuta el programa *nprog*. El error puede identificarse mediante la función ERROR.

Existen lenguajes en los que es posible emplear REACT con niveles de prioridad. VAL no permite concurrencia explícita en el programa. Sin embargo, se permite la utilización concurrente de la consola con el control del robot.

### Ejemplo 11.2

En un estudio de grabación se tiene una cámara de televisión montada en el extremo de un brazo robótico programable en VAL II. El brazo robótico está situado en una plataforma que puede desplazarse sobre unas guías. Existen tres posiciones predeterminadas de la plataforma en las que la cámara está enfocando a tres locutores. Tanto la orientación de la cámara como el movimiento de la plataforma se pueden operar desde una mesa de control conectada al brazo robótico. Las funciones de control son las siguientes:

- giro de la cámara según el eje vertical (articulación 1 del brazo) a izquierda y derecha,
- desplazamiento de la plataforma a cada una de las tres posiciones predeterminadas.

Las líneas de entrada y salida del brazo robótico son las siguientes:

– Líneas de entrada:

- 1001 - GIRO+. Cuando está activada, orden de girar la cámara hacia la derecha.
- 1002 - GIRO-. Cuando está activada, orden de girar la cámara hacia la izquierda.
- 1003 - POS1. Cuando está activada, orden de mover la plataforma a posición 1.
- 1004 - POS2. Cuando está activada, orden de mover la plataforma a posición 2.
- 1005 - POS3. Cuando está activada, orden de mover la plataforma a posición 3.
- 1010 - Detecta si la plataforma está en la posición 1.
- 1011 - Detecta si la plataforma está en la posición 2.
- 1012 - Detecta si la plataforma está en la posición 3.

– Líneas de salida:

10 - Conectada a una señal luminosa.

12 - Cuando está activada, conecta el motor que mueve la plataforma (dirección de POS1 a POS3).

13 - Cuando está activada, conecta el motor que mueve la plataforma (dirección de POS3 a POS1).

Las tres posiciones de la plataforma están situadas de forma consecutiva (POS1, POS2 y POS3) sobre las guías. Es decir, POS1 y POS3 están situadas en los extremos de las guías, mientras que POS2 está situada entre POS1 y POS3.

A continuación se presenta un programa en VAL II que realiza las siguientes funciones:

- Al empezar el programa, la plataforma irá a la posición POS1 y el brazo robótico a la posición posinicial (se supone conocida). Cuando reciba una orden por las líneas de entrada, la ejecutará y seguirá esperando más órdenes.
- Al recibir las órdenes GIRO+ y GIRO-, girará el brazo robótico según el eje vertical (articulación 1) en sentido positivo o negativo en incrementos de 1 grado.
- Al activarse alguna de las líneas POS1, POS2 o POS3, moverá la plataforma a la posición correspondiente.
- Siempre que la plataforma esté en movimiento, la señal luminosa debe estar encendida.

El programa es el siguiente:

```
.PROGRAM plataforma  
MOVE posinicial ; posición inicial  
SIGNAL 13, 10  
WAIT SIG (1010)  
SIGNAL -13, -10  
pos = 1; flag con la posición de la plataforma  
WHILE (TRUE) DO
```

```
IF SIG (1001) THEN  
DRIVE 1, 1, 100; giro unidad a la derecha  
END  
  
IF SIG (1002) THEN  
DRIVE 1, -1, 100; giro unidad a la izquierda  
END
```

IF SIG (1003) THEN

SIGNAL 13, 10; desplazamiento hacia POS1

WAIT SIG (1010)

SIGNAL -13, -10

pos = 1

END

IF SIG (1005) THEN

SIGNAL 12, 10; desplazamiento hacia POS 3

WAIT SIG (1012)

SIGNAL -12, -10

pos = 3

END

IF SIG (1004) THEN

IF (pos == 1) THEN

SIGNAL 12, 10; desplazamiento hacia POS2 desde POS1

ELSE

SIGNAL 13, 10; desplazamiento hacia POS2 desde POS3

END

WAIT SIG (1011)

IF (pos == 1) THEN

SIGNAL -12, -10

ELSE

SIGNAL -13, -10

END

pos = 2

END

END; del WHILE

.END

### Ejemplo 11.3

Se trata de programar la operación de recogida de tuercas situadas sobre una cinta transportadora, atornillándolas en cuatro tornillos situados sobre una pieza en determinadas posiciones previamente establecidas en el espacio de trabajo del robot según un rectángulo de lados 150 y 315. El movimiento de la cinta transportadora se contro-

la mediante la señal 1. La cinta debe pararse cuando un sensor detecte el paso de una tuerca, moviéndose entonces el manipulador para realizar el agarre de la tuerca, transportarla hacia el siguiente tornillo libre y atornillarla.

El programa en VAL II es el siguiente:

```
.PROGRAM tornillo; programa principal
SET posalim = TRANS (0, 500, -487, 0, 90, 0)
SET postorn = TRANS (300, 157.5, -470, 90, 90, 0)
FINE
FOR ntorn=1 to 4
CASE ntorn OF
  VALUE 1:
    SET postorn = SHIFT (postorn BY 0, 315, 0)
  VALUE 2:
    SET postorn = SHIFT (postorn BY 150, 0, 0)
  VALUE 3:
    SET postorn = SHIFT (postorn BY 0, -315, 0)
  VALUE 4:
    SET postorn = SHIFT (postorn BY -150, 0, 0)
END
APPRO posalim, 25
OPENI
SIGNAL 1; activar cinta transportadora
WAIT SIG(1001); esperar hasta que se active el sensor
SIGNAL -1; parar cinta transportadora
MOVES posalim
CLOSEI
DEPARTS 50
APPRO postorn, 100
MOVES postorn
CALL aprietatorn
DEPARTS 100
END
READY; mover robot a posición READY
CLOSEI
.END

.PROGRAM aprietatorn; aprieta la tuerca en el tornillo
npasos=24; selección numero pasos
```

```
avance=8/npasos; avance 8mm por paso  
giro=180/npasos; giro 180 grados por paso  
SET rosca = postorn  
FOR i = 1 TO 4  
FOR j = 1 TO npasos  
MOVE rosca: TRANS (0, 0, avance, 90, -90, giro)  
BREAK  
HERE rosca  
END  
OPENI  
SET rosca = rosca: TRANS (0, 0, 0, 90, -90, -180)  
IF (i<>4) THEN  
MOVE rosca  
BREAK  
CLOSEI  
END  
END  
.END
```

Obsérvese que en el ejemplo se utiliza un programa principal y subrutinas. Cuando se emplea una instrucción con CALL, el programa que la ejecuta queda suspendido hasta que el programa llamado (considerado como una subrutina del que le llama) le devuelva el control. Una subrutina puede, a su vez, llamar a otra subrutina. Las subrutinas devuelven el control al programa o subrutina que le ha llamado de dos formas diferentes: ejecutando una instrucción RETURN o alcanzando el final de la subrutina. En el ejemplo, el control se devuelve según el segundo caso.

## 11.9 PROGRAMACIÓN DE VEHÍCULOS ROBÓTICOS

Los apartados anteriores tratan fundamentalmente la programación de robots manipuladores. Los lenguajes de programación de vehículos robóticos han recibido una atención menor.

Nótese que, en la robótica móvil, el énfasis se pone fundamentalmente en los aspectos de interacción con el entorno, sin suponer que éste es conocido y estático. La estimación de la posición del sistema asociado al efecto del robot manipulador suele realizarse empleando las lecturas de los codificadores en las articulaciones y el modelo cinemático. En muchos casos, la precisión es suficiente, no siendo necesario acudir a sensores de percepción del entorno para confirmar que el efecto final está en la posición a la que se le ha ordenado dirigirse. Sin embargo, la estimación de posición de los vehículos robóticos es más difícil ya que normalmente no basta con la odometría y

es necesario acudir a sensores de percepción del entorno, o sensores específicos para posicionamiento.

Conviene poner de manifiesto que las técnicas indicadas en apartados anteriores pueden ser aplicadas para la programación de las operaciones de robots móviles en los que existe un manipulador situado sobre el vehículo. Por tanto, en este apartado, se tratan más específicamente los problemas asociados a la programación de los movimientos del vehículo robótico.

Normalmente, la robótica móvil no está asociada a tareas repetitivas, en un entorno industrial, que resultan más fácilmente programables. Desde un punto de vista práctico, la teleoperación juega un importante papel en las aplicaciones de los vehículos robóticos. Por tanto, no es extraño que muchos sistemas de programación de robots móviles involucren lenguajes que pueden considerarse también como lenguajes de teloperación (Cuesta y otros, 1998).

Por otra parte, exceptuando vehículos especiales que se mueven en tres dimensiones (capítulo 2), la mayor parte de los vehículos robóticos actuales son vehículos con ruedas que se mueven en el plano.

Por todo ello, la programación de vehículos robóticos suele realizarse con un juego de instrucciones relativamente pequeño en las que se involucran movimientos en el plano y con una relativamente elevada interacción con el entorno.

De esta forma, se combinan órdenes básicas a la dirección o velocidad del vehículo con otras cuya ejecución hace necesario el empleo del sistema de percepción. En el sistema de programación de los vehículos robóticos denominados Romeo (Ollero y otros, 1999) que se presentaron en el capítulo 2, entre las órdenes básicas se cuenta con instrucciones tales como las siguientes:

LEFT 5

RIGHT 5

HOME

que indican, respectivamente, giro a la izquierda o a la derecha de 5 grados en la dirección (similares a giros en el volante del vehículo) y volver posteriormente a la dirección cero (centrar el volante), o bien la:

STRAIGHT FWD

STRAIGHT BWD

que indican moverse en línea recta hacia adelante o hacia atrás, o bien:

SPEED 1

para indicar velocidad lineal en metros por segundo.

Entre las que involucran percepción del entorno cabe mencionar las de control reactivo, tales como:

LEFTWALL [20]

RIGHTWALL [20]

CORRIDOR

Las dos primeras especifican un seguimiento reactivo de pared izquierda o derecha, respectivamente. El parámetro puede utilizarse para indicar la distancia a la que hay que realizar el seguimiento de la pared. La tercera especifica el seguimiento de un corredor tratando de circular por el centro del mismo.

Nótese que la implantación de estas instrucciones podría realizarse mediante sensores de proximetría en arquitecturas de control reactivo, tales como las que se mencionan en el capítulo 6.

Otras instrucciones que pueden involucrar o no el empleo de sensores de percepción del entorno son las:

LEFTTURN

RIGHTTURN

CHANGELANE 50

Las dos primeras indican giros de 90 grados en la dirección del vehículo. La tercera especifica un cambio a una línea paralela a la distancia especificada. Otra instrucción que también puede emplearse en la programación de los vehículos Romeo (Olleto y otros, 1999) es la:

PATHTRACKING *f*

Mediante ella se trata de que el vehículo siga automáticamente el camino indicado en el fichero *f*. Los registros de este fichero están formados por la posición, orientación y curvatura de un conjunto de puntos consecutivos que forman el camino que se pretende seguir.

Hay también un conjunto de instrucciones dedicadas específicamente al sistema de percepción. Entre ellas existen algunas muy simples tales como:

PAN

TILT

HOMECAM

que especifican giros elementales en el sistema de orientación de una cámara sobre el vehículo y su vuelta a la posición origen, o bien de ejecución más compleja tal como:

VISEST

para la estimación de la posición empleando el sistema de visión, o incluso:

VISTRK

para el seguimiento visual de un objeto situado en el campo de visión de una cámara.

Además del empleo de instrucciones específicas, la programación puede realizarse mediante botonera, tal como se indica en el teclado funcional del robot Aurora (Mandow y otros, 1996), que se ilustra en la figura 11.5 (Gómez de Gabriel y otros, 1996).

En este teclado funcional pueden identificarse gráficamente algunas de las instrucciones indicadas en los párrafos precedentes tales como RIGHTWALL, LEFTWALL, CORRIDOR, LEFTTURN, RIGHTTURN.



Figura 11.5: Teclado funcional para programación del robot Aurora.

## 11.10 CONCLUSIONES

---

La programación de robots puede realizarse a muy diferentes niveles, desde un nivel inferior en el cual se especifican movimientos de cada una de las articulaciones, a niveles superiores en los que la programación se realiza a nivel objeto; es decir, especificando acciones sobre objetos; o incluso a nivel tarea. La programación a niveles superiores hace necesario disponer de modelos suficientemente fiables. Así, cuando se programa un robot especificando sus movimientos en el espacio cartesiano, es necesario disponer de un modelo que relacione los movimientos de las articulaciones con los movimientos del robot en el espacio. Análogamente, para poder programar a nivel objeto es necesario disponer de modelos tanto del robot como de todos los objetos que están en su entorno. En principio, se requiere también que el entorno del robot sea estático.

En este capítulo se ha estudiado la programación de robots, con especial énfasis en la programación de robots manipuladores industriales. Tanto la técnica de guiado como los lenguajes considerados son de nivel robot. De hecho, la gran mayoría de las aplicaciones actuales se programan empleando instrucciones en las que se especifican los movimientos de los robots. En los ejemplos se ha utilizado el lenguaje VAL II, debido a su gran difusión.

En la última parte del capítulo se ha considerado la programación de vehículos robóticos, poniendo de manifiesto las diferencias con respecto a la programación de manipuladores e indicando algunas de las instrucciones que se utilizan para programar los vehículos Romeo.

## 11.11 REFERENCIAS

---

ABB Robotics, 1994. RAPID Programming Manual.

Barrientos A. y otros, 1997. "Fundamentos de robótica". McGraw-Hill.

Bonner S. y K.G. Shin, 1982. "A Comparative Study of Robot Languages". Computer, 12, pp. 82-96.

- Cuesta F., B.C. Arrue y A. Ollero, 1998. "A new system for intelligent teleoperation of vehicles". Proceeding of the Intelligent Autonomous Vehicles - IAV'98, pp. 297-302. Madrid.
- Gómez de Gabriel J., J.L. Martínez, A. Ollero, A. Mandow y V.F. Muñoz, 1996. "Autonomous and Teleoperated Control of the Aurora Mobile Robot". Proceeding of the 13th World Congress of IFAC International Federation of Automatic Control, pp. 181-186. San Francisco, Calif., USA.
- Mandow A., J. Gómez de Gabriel, J.L. Martínez, V.F. Muñoz, A. Ollero y A. García-Cerezo, 1996. "The Autonomous Mobile Robot AURORA for Greenhouse Operation". IEEE Robotics and Automation Magazine. Vol. 3, núm. 4, pp. 18-28.
- Martínez J.L., A. Corral y A. Ollero, 1991. "MICROVAL: Simulador del sistema y lenguaje VAL II del robot PUMA-500". Jornadas sobre control inteligente y robótica. Sevilla, enero.
- Ollero A., B.C. Arrue, J. Ferruz, G. Heredia, F. Cuesta, F. López-Pichaco y C. Nogales, 1999. "Control and Perception Components for Autonomous Vehicles Guidance. Application to the ROMEO Vehicle". Control Engineering Practice. Vol. 7, núm. 10, pp. 1291-1299.
- Ollero A., J. Gómez de Gabriel, E. Lozano, F. Cuesta y J. Gómez Elvira, 1995. "MINIMAN Telerobotic Station". Proceedings of the International Conference on Advanced Robotics. ICAR'95. Vol. II, pp. 591-599, Sant Feliu de Guíxols, Cataluña.
- Silva M., 1983. "Lenguajes de programación de robots industriales". Revista de Robótica, núm. 6, pp. 35-48.
- Silva M., 1984. "Lenguajes de programación para robots industriales (1ª parte)". Revista de Robótica. núm. 7, pp. 39-53.
- Stäubli, 1997. Lenguaje V+.
- Staubli Unimation Ltd., 1990. "Robot 560 C arm manual"
- Unimate Industrial Robot, 1986. User's guide to VAL II. Version 2.0, 1986.

# Detección de colisiones y planificación de caminos

---

En el capítulo anterior se estudiaba la programación de robots suponiendo que el programador conoce las trayectorias que debe describir un sistema de referencia asociado al robot. No obstante, si se producen modificaciones en el entorno del robot, la ejecución de las trayectorias programadas puede originar colisiones que es necesario detectar y evitar; se aborda ahora este problema de interés obvio en robótica. Por otra parte, nótese que si se pretende programar el robot a un nivel superior de abstracción, sin necesidad de especificar las trayectorias que debe seguir el robot para realizar una determinada tarea, es necesario que el sistema informático de control incorpore funciones que permitan determinar automáticamente caminos libres de colisión. El estudio de este problema es también objeto del capítulo. Teniendo en cuenta la complejidad de los problemas involucrados, tan sólo se pretende realizar una introducción en la cual se ponen de manifiesto los aspectos principales de dichos problemas y diversas técnicas existentes en la actualidad para resolverlos. Los conceptos y métodos que se introducen son de interés tanto para robots manipuladores como para vehículos robóticos.

## **12.1 DETECCIÓN, EVITACIÓN Y PLANIFICACIÓN**

---

La planificación y ejecución de caminos libres de obstáculos es una función de interés obvio en robótica debido a los requerimientos de seguridad de funcionamiento y fiabilidad. Antes de profundizar en los conceptos y métodos involucrados conviene precisar la terminología utilizada.

Mediante la *detección de colisiones* se trata de prever la posibilidad de una colisión basándose en información sobre la posición del robot y su carga, así como de los objetos en su entorno. Para ello, desde un punto de vista práctico, cabe distinguir entre la detección de colisión en una determinada dirección de movimiento, a partir de una medida de proximidad suministrada en tiempo real por un sensor, y la detección de

una intersección entre los espacios ocupados por el manipulador, su carga y los objetos de su entorno empleando modelos apropiados del manipulador, la carga y su entorno. En este segundo caso, cabe hablar también de tiempo real dependiendo de la complejidad de los modelos y de su forma de obtención o actualización empleando medidas de sensores.

La *evitación de colisiones* es típicamente una función que se implanta en tiempo real durante la ejecución del movimiento. Una vez que se ha detectado una posible colisión, se trata de generar acciones que permitan evitarla. En el caso más simple, consiste en disminuir la velocidad y realizar una parada, o generar un movimiento seguro, seguido en ambos casos de una llamada al sistema de supervisión para que el operador tome las acciones oportunas. En el caso más complejo, se involucra una *planificación local* de movimientos: se trata de prever los efectos de realizar un determinado movimiento y realizar una planificación del camino más apropiado disponiendo de un modelo obtenido a partir de información previa del robot y su entorno, adquirida mediante el sistema sensorial y/o por el conocimiento previo del entorno, supuesto éste invariante.

El problema de la *planificación de caminos libres de obstáculos* suele formularse como sigue: dadas unas posiciones/orientaciones inicial ( $S$ ) y objetivo ( $G$ ) del robot y su carga, se trata de generar un camino desde  $S$  a  $G$  libre de obstáculos y, eventualmente, óptimo respecto a ciertos criterios de funcionamiento.

La planificación autónoma involucra siempre la función de detección de colisiones. En efecto, es necesario saber si la trayectoria del robot que se está considerando producirá colisión en un determinado espacio de trabajo.

En este capítulo se consideran tanto las funciones de detección y evitación de colisiones, como la de planificación de caminos libres de obstáculos. En el próximo apartado se trata su implantación distinguiendo entre las que deben ejecutarse en tiempo real y las que pueden abordarse fuera de línea, durante la fase de programación de la aplicación.

En el tercer apartado se consideran los modelos que pueden utilizarse en la detección de obstáculos y planificación de caminos libres de obstáculos. El cuarto está dedicado específicamente a la detección y evitación de colisiones empleando modelos. En el quinto se estudia la planificación de caminos empleando modelos.

Finalmente, en el sexto apartado, se introducen métodos reactivos que permiten generar movimientos como respuesta directa a la información suministrada por sensores.

## 12.2 FUNCIONES DE DETECCIÓN, EVITACIÓN Y PLANIFICACIÓN

Con objeto de precisar más los conceptos involucrados en las funciones de detección, evitación y planificación, hay que distinguir entre: funciones en tiempo real durante la ejecución de movimientos, funciones que se realizan antes de iniciar el movimiento y funciones de ayuda a la programación.

Las primeras utilizan en tiempo real información de sensores; deben realizarse con importantes restricciones temporales y de forma integrada en el controlador tiempo

real de movimientos, de forma que los aspectos de sincronización y coordinación en tiempo real son importantes.

En las segundas se necesita una cierta integración con el sistema de control de movimientos en tiempo real, según una determinada arquitectura del sistema completo de control. Esta integración es más débil que en el caso anterior. Nótese que si el tiempo de procesamiento de esta función es inferior al de la ejecución del movimiento y se utilizan distintos procesadores o sistemas para planificación y control de ejecución, cabe pensar en la ejecución continua de movimientos, sin necesidad de esperas para planificación entre la ejecución de dos caminos consecutivos. En este caso, se supone que el movimiento que está siendo ejecutado se realizará de forma perfecta y, por tanto, es posible planificar a partir de la posición a la que llegará el robot en el movimiento actual. En esta estrategia también cabe hablar de planificación en tiempo real.

Por último, mediante las funciones de ayuda, se trata de establecer una herramienta para la programación de las operaciones del robot que facilite la tarea del programador mediante la detección de una colisión potencial de un movimiento programado, o incluso mediante la generación de caminos en entornos complejos que requieren mucho tiempo de programación. Otra aplicación de interés es la detección de posibles transiciones bruscas entre configuraciones, originadas por una programación aparentemente correcta. Por ejemplo, en un manipulador, la programación de trayectorias cartesianas rectilíneas puede dar origen a que, entre dos configuraciones consecutivas del manipulador, la última articulación tenga que realizar un giro de casi 180 grados para orientar el órgano terminal con la dificultad práctica y el riesgo que ello supone. Sin embargo, estas transiciones no se producirían con caminos generados mediante un método apropiado de planificación. Esta función se realiza típicamente fuera de línea. Asimismo, esta función puede utilizarse en un sistema de teleoperación con objeto de facilitar la tarea del operador, tal como se indicará en el capítulo 13.

Por último, es necesario poner de manifiesto que la *complejidad* de las funciones mencionadas anteriormente es muy dependiente de los siguientes factores:

- Complejidad del entorno, que depende a su vez de la disposición, número y forma de obstáculos, así como de su tamaño con relación al del robot.
- Número de articulaciones.
- Conocimiento del entorno: presencia de obstáculos no previstos.
- Movilidad de obstáculos.
- Requerimientos sobre el camino planificado, facilidad de ejecución por un determinado robot, distancia mínima a los obstáculos para mantener condiciones de seguridad, longitud y potencia necesaria para ejecutar el camino.
- Tiempo de ejecución del planificador con unos determinados recursos "hardware" y "software" disponibles.

En la evitación de obstáculos y planificación de caminos libres de obstáculos deben contemplarse las relaciones con los subsistemas de percepción y control de movimientos del manipulador. Estas relaciones se tratan brevemente en los párrafos siguientes.

Con respecto a la percepción, debe contemplarse la incorporación de sensores para detección de colisiones, construcción de mapas y estimación de posición del robot. Estos sensores pueden ser pasivos (por ejemplo, cámaras de vídeo) o activos (por ejemplo, emisión láser o ultrasonidos). Es claro que las características de la información suministrada tienen una gran influencia en los métodos que se empleen para la evitación de colisiones y planificación. De esta forma, mientras que para detectar la presencia de un obstáculo en una determinada dirección o direcciones de movimiento, en tiempo real, se necesita relativamente poca información, la construcción de modelos geométricos para planificación de caminos en tres dimensiones necesita un volumen importante y una gran cantidad de procesamiento. Como se pondrá de manifiesto posteriormente, las estructuras de datos para representar la información geométrica del entorno tienen también una notable influencia en los algoritmos de planificación.

En sistemas robóticos móviles, es frecuente que el controlador a bordo carezca de un modelo completo del entorno y, por consiguiente, no pueda determinar caminos libres de obstáculos utilizando el modelo. Sin embargo, es posible que exista un sistema telerrobotico, tal como se estudiará en el capítulo 13, dotado de un potente computador y modelos apropiados. Este sistema podría realizar las tareas de planificación y enviar al sistema robótico móvil la trayectoria deseada. Alternativamente, un operador de una estación de teleoperación podría enviar también esta trayectoria. Nótese que si, por ejemplo, al controlador de un manipulador remoto se le especifica, desde una estación de teleoperación, un camino en coordenadas cartesianas, dicho controlador debería comprobar que en la resolución del modelo inverso en el controlador local del manipulador no se adoptarán soluciones en variables articulares que produzcan colisión. Sin embargo, este problema no sería necesario resolverlo si se envía desde la estación de teleoperación las coordenadas articulares del camino.

Para poder realizar las tareas de planificación, el sistema telerrobotico necesita tener actualizada la posición del robot, por lo cual el robot móvil debe enviar la información captada por sus sensores. En cualquier caso, el controlador situado sobre el robot móvil debe mantener un cierto número de funciones tales como el control de movimientos y la detección de obstáculos, que requieren una respuesta rápida.

Aunque en los apartados posteriores se considera fundamentalmente la detección de obstáculos y planificación de caminos, suponiendo conocida la información procedente de los sensores y suponiendo también que los movimientos planificados se ejecutan mediante el controlador de forma eficiente, la consideración de las inevitables imprecisiones es muy importante para asegurar la fiabilidad del sistema. En la práctica, las funciones de percepción, planificación y control de movimientos deben integrarse en una determinada arquitectura del sistema informático de control.

---

### **12.3 DETECCIÓN Y EVITACIÓN DE COLISIONES MEDIANTE REACCIÓN DIRECTA A INFORMACIÓN DE SENsoRES**

---

Como se ha mencionado anteriormente, la detección de una colisión puede efectuarse sin necesidad de emplear un modelo del entorno de las características citadas en la sección precedente. Para ello es necesario contar con información apropiada de sensores en tiempo real durante el movimiento del robot.

En los párrafos siguientes se mencionan algunas características de sensores utilizados para detección de obstáculos, sin pretender realizar un estudio comparativo riguroso.

Los métodos de percepción activa han sido quizás los más utilizados desde el punto de vista práctico en detección de colisiones. En particular, los sensores de ultrasonidos han sido muy utilizados en vehículos robóticos para interiores. Como se indicó en el capítulo 7, se emplea el principio del tiempo de vuelo o tiempo de eco, mediante el cual la distancia viene dada por el tiempo transcurrido entre la emisión de un pulso y su recepción después de ser reflejado por una superficie. No obstante, estos sensores tienen importantes limitaciones de alcance en la transmisión del sonido por el aire, por lo cual su empleo en vehículos para exteriores es escasa. Sin embargo, tienen gran interés para vehículos submarinos debido a las mejores propiedades de la transmisión de los ultrasonidos en este medio.

La detección de obstáculos mediante láser es cada vez más empleada en robótica móvil debido a su mayor fiabilidad. En el capítulo 7 se presentaron diferentes técnicas entre las que está también el tiempo de vuelo. En este caso el alcance es mucho mayor que el de los sensores de ultrasonidos, pero existen limitaciones en la detección de obstáculos próximos. Nótese como, por ejemplo, para una distancia de 0,3 m con la velocidad de la luz de  $3 \cdot 10^8$  m/s el tiempo de vuelo es de 2 ns lo que exige una electrónica de detección compleja y cara. Existen también sistemas láser basados en otros principios tales como las medidas de desfase entre la onda emitida y la reflejada.

Para la determinación de la distancia a objetos muy próximos se emplean también sensores láser que utilizan el principio de la triangulación. Así existen sensores comerciales que emplean un emisor y una línea de receptores determinando la distancia según el receptor que reciba mayor ganancia en la reflexión. Estos sensores, de alcance reducido, se instalan en efectores finales de manipuladores.

Los sensores de contacto han sido empleados también en la detección de colisiones e incluso reconocimiento de objetos. Obviamente, suministran información cuando el contacto ya se ha producido, pero pueden utilizarse en movimientos lentos para minimizar los efectos de la colisión. La dificultad práctica es su colocación sobre el robot, carga y objetos del entorno. En este mismo contexto resultan útiles los sensores de esfuerzos colocados sobre el manipulador que tienen la ventaja adicional de permitir la aplicación de métodos de control de esfuerzos en movimientos que implican contacto.

Se han empleado también técnicas de procesamiento de imágenes para detección de colisiones. Para ello es imprescindible tener la cámara o cámaras convenientemente localizadas para suministrar condiciones apropiadas de visión en el movimiento del robot con respecto a los objetos del entorno con los que puede colisionar, lo cual puede no ser fácil. El empleo de cámaras para detección de obstáculos presenta diversos inconvenientes entre los que cabe mencionar las condiciones de iluminación, brillos de las superficies, sombras, etc. Nótese también la posibilidad de emplear técnicas de triangulación con emisión de láser en el espectro visual y adquisición de la imagen resultante de la intersección del haz láser con los objetos. En cualquier caso, los problemas de procesamiento de imágenes son complejos y, aunque en los últimos años se han realizado progresos notables y hardware especializado, resulta difícil su aplicación en tiempo real durante el movimiento del manipulador.

Para la determinación de colisiones a distancias relativamente elevadas, de interés en vehículos robóticos para exteriores, circulando a velocidad elevada, existen técnicas de detección empleando emisiones de radiofrecuencia que utilizan también el principio del tiempo de eco. Las dificultades son las mismas que las mencionadas anteriormente a las que hay que añadir la posibilidad de interferencias con las transmisiones de radiofrecuencia, así como el tamaño y potencia consumida.

Una vez detectada la posibilidad de una colisión en tiempo real, caben distintas posibilidades entre las que se encuentran:

- Disminución inmediata de velocidad y realización de una parada: el sistema de control de movimientos deberá realizar la generación de las órdenes oportunas para la realización de la parada teniendo en cuenta la información de los sensores (direcciones de movimientos en las que se ha detectado obstáculo).
- Reacción autónoma. Desde un punto de vista amplio pueden incluirse desde los métodos simples para la generación de movimientos seguros hasta las técnicas de planificación reactiva que se consideran en un apartado posterior de este capítulo. En estos métodos la planificación es evidentemente local.

## 12.4 MODELOS

Como se ha mencionado anteriormente, las características del modelo geométrico disponible del robot y su entorno tienen una importante influencia en la eficiencia de la planificación y detección de obstáculos. En este apartado se resumen distintas técnicas utilizadas para la construcción de estos modelos.

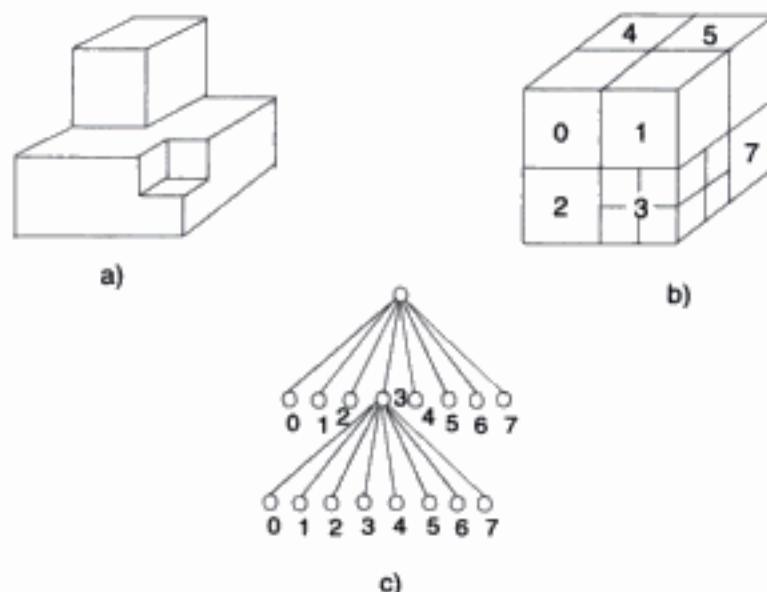
### 12.4.1 Modelado mediante ocupación de celdas

Se trata de dividir el volumen de trabajo en celdas de una determinada dimensión y marcar cada celda como ocupada o libre. La actualización del modelo es relativamente simple pero su consulta puede ser muy ineficiente si la resolución es alta, lo cual puede ser necesario en las proximidades de obstáculos y posición final. Para hacer más eficiente el modelo puede acudirse a técnicas de tamaño de celda variable. Se han propuesto también modelos que tienen en cuenta la incertidumbre de las medidas asociando a cada rejilla una probabilidad de estar ocupada, y actualizando la probabilidad cuando se toman nuevas medidas (Moravec y Elfes, 1985; Elfes, 1987).

### 12.4.2 Empleo de estructuras jerárquicas

Se trata de emplear estructuras de datos eficientes que proporcionen una resolución elevada sólo en las regiones del espacio de trabajo en que sea necesaria.

En particular, los quadtrees y octrees (Samet y Tamminen, 1985; Samet, 1989) son estructuras de datos de amplia utilización en informática gráfica. El octree es una estructura jerárquica que subdivide recursivamente un volumen cúbico en ocho volúmenes más pequeños, denominados octantes, hasta que se satisface un cierto criterio al



**Figura 12.1:** Representación mediante octree-región.

que se denomina regla de descomposición. El proceso de descomposición suele representarse mediante un árbol de grado 8 (“octree”), como se ilustra en la figura 12.1 (Shaffer y Herb, 1992). Dependiendo de la regla de subdivisión, existen diversas variedades de octrees.

El octree-región es una estructura en la cual se comienza con un cubo que incluye todo el espacio de trabajo. El proceso de subdivisión se realiza hasta que cada octante esté completamente dentro de un objeto o completamente vacío (figura 12.1). La aplicación del octree-región en algoritmos de evitación de colisiones y planificación de caminos ha sido estudiada por numerosos investigadores (Hayward, 1986; Herman, 1986).

Los octrees  $N$ -objetos también subdividen el espacio en octantes, pero en este caso se almacena una lista de los objetos asociados a cada nodo del árbol. El proceso de subdivisión acaba cuando en cada hoja del árbol no existen más de  $N$  objetos. Estos octrees han sido también empleados para la evitación de obstáculos en tiempo real (Shaffer y Herb, 1992). En la figura 12.2 se ilustra la representación mediante un quadtree 5-objetos. Como propiedad importante cabe señalar que sólo los objetos que comparten un nodo pueden colisionar de forma inminente.

Existen también autores que han empleado octrees, y en particular octrees-región, con cuatro dimensiones, añadiendo el tiempo como una nueva dimensión, para planificación de movimientos (Fujimura y Sammet, 1988).

#### 12.4.3 Modelos del entorno basado en primitivas 3-D de sólidos

Se trata de modelar el robot y su entorno empleando objetos geométricos simples que en términos informáticos corresponden a primitivas. Entre las primitivas más utilizadas se encuentran: esferas, cilindros, paralelepípedos, conos y planos. Se emplean también otras estructuras que resultan de la combinación de las anteriores. Entre éstas

cabe mencionar las cilindro-esferas o "cylspheres", que consisten en un cilindro con una esfera en cada uno de sus extremos, mediante las cuales se modela el enlace de un manipulador con una articulación en cada extremo (figura 12.2).

Los objetos se representan mediante la unión de primitivas. La elección de las primitivas es básica para conseguir una representación suficientemente precisa y fiable y, a su vez, facilitar las operaciones de intersección de primitivas en las cuales se basan los algoritmos de detección de colisiones. Nótese, por ejemplo, que las esferas son objetos que pueden ser descritos por un solo parámetro (el radio), y la descripción de su movimiento se reduce a la descripción del movimiento del centro. Existen también algoritmos eficientes para intersección de esferas (Hopcroft y otros, 1983). Sin embargo, el número de esferas necesarias para conseguir una aproximación aceptable puede ser muy grande.

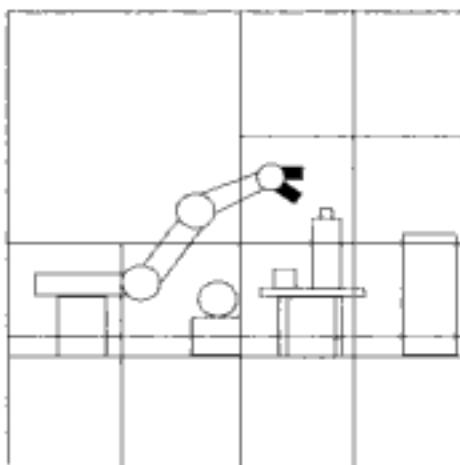


Figura 12.2: Quadtree  $N$ -objetos.

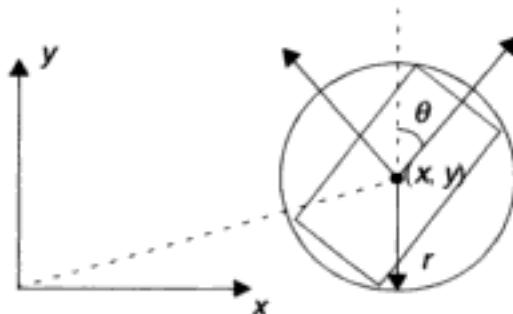
En general, la elección de las primitivas debe realizarse para conseguir un balance entre la eficiencia de las operaciones de intersección entre primitivas y el número de primitivas necesario para representar adecuadamente el espacio de trabajo. A su vez, el número de primitivas necesario afecta también al número total de intersecciones entre primitivas que hay que realizar.

En la detección de colisiones en tiempo real, la dificultad para comprobar la intersección de todos los enlaces del robot y su carga con los posibles obstáculos en cada punto de control resulta obvia. En estos casos, puede resultar aconsejable el empleo de procedimientos, tales como los de búsquedas en tablas precalculadas, para acelerar los cálculos en tiempo real.

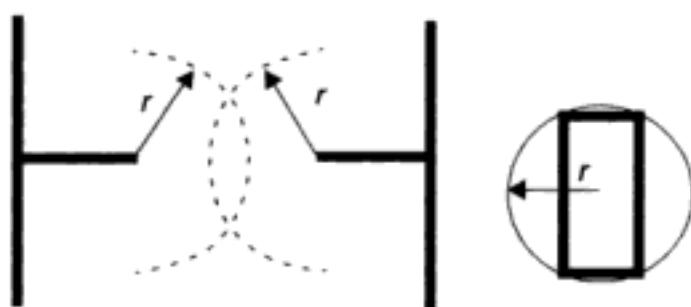
Por otra parte, la representación mediante primitivas puede combinarse también con las estructuras de datos jerárquicas. Por ejemplo, el entorno supuesto estático puede representarse mediante octree-región y el robot en movimiento mediante primitivas cuya posición sea fácilmente actualizable. En Shaffer y Herb (1992) se modela el manipulador mediante cilindro-esferas, empleándose un octree  $N$ -objetos para minimizar el número de intersecciones (intersección de primitivas sólo si comparten un nodo de octree  $N$ -objetos).

#### 12.4.4 Expansión de obstáculos

Existen numerosos algoritmos que permiten determinar los caminos más cortos libres de obstáculos entre dos puntos, suponiendo que el objeto que se mueve es un punto. Para que estos algoritmos sean aplicables cuando lo que se mueve es un objeto de determinadas dimensiones, es necesario adoptar modelos apropiados. Suponiendo que el problema se plantea en el plano, el modelo más simple del robot consiste en emplear un círculo de un determinado radio  $r$ , tal como se ilustra en la figura 12.3. De esta forma, si todos los posibles obstáculos se expanden según arcos circulares de radio  $r$ , pueden aplicarse los mismos algoritmos que suponen que el objeto que se mueve es un punto. Sin embargo, esta técnica presenta los problemas que se ilustran en la figura 12.4. Nótese que un robot de contorno rectangular podría pasar entre los obstáculos, pero la expansión de dichos obstáculos mediante arcos circulares de radio  $r$  haría que no existiera ningún camino que permitiera el paso del robot.



**Figura 12.3:** Representación de un robot móvil en el plano mediante un círculo de radio  $r$ .



**Figura 12.4:** Expansión de obstáculos mediante arcos circulares de radio  $r$ .

En el párrafo anterior se pone de manifiesto que la simplicidad y conservadurismo implícitos en la elección del modelo circular del robot pueden hacer que no exista ninguna solución al problema de planificación de caminos. Sin embargo, si se adoptara un contorno poligonal como modelo del robot, podría llegar a determinarse una trayectoria para pasar entre los obstáculos. Obsérvese que, en este caso, no basta con conocer las coordenadas del centro de guiado del robot; se necesita también la orientación (ángulo  $\theta$ ), tal como se indica en la figura 12.3.

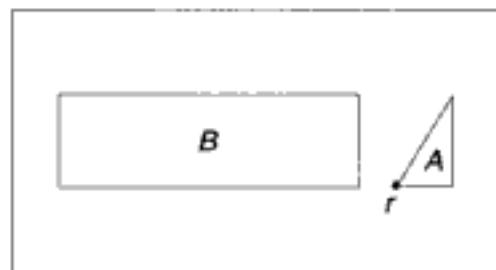
El problema que se trataba en los párrafos anteriores se planteaba en el plano. Nótese que en 2D se necesitan tres parámetros para especificar la configuración del robot: las coordenadas  $x$  e  $y$  y su orientación. Si se trabajara en un espacio 3D, se emplearían poliedros en vez de contornos poligonales para representar al robot, y su posición y orientación vendrían dadas por seis parámetros (tres coordenadas y tres ángulos de orientación), tal como se indicaba en el capítulo 3.

En lo que sigue, se considera un método que permite expandir los obstáculos teniendo en cuenta de forma más apropiada las características geométricas del robot.

#### 12.4.5 Modelado en el espacio de configuraciones

El modelado en el espacio de configuraciones es uno de los más empleados para planificación de caminos. Suponiendo un móvil en un espacio con obstáculos, la idea básica consiste en transformar los obstáculos y los límites del espacio de forma tal que el problema se reduzca a encontrar el camino de un punto de referencia en el móvil desde la posición inicial a la final; es decir, los obstáculos crecen en una cantidad tal que sus límites definen el lugar de las regiones prohibidas (Lozano Pérez y Wesley, 1979, 1980; Lozano Pérez, 1980, 1981).

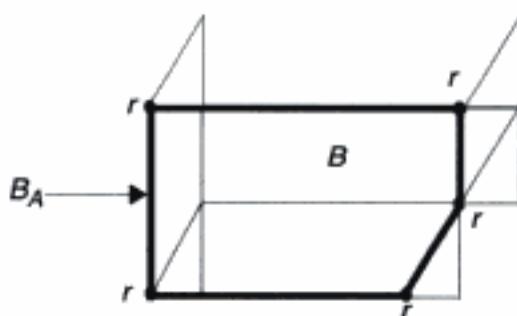
Para introducir la representación en el espacio de configuraciones, conviene considerar el caso de objetos poligonales en un plano, tal como se muestra en la figura 12.5 (Schilling, 1990). Supóngase que  $A$  es el objeto móvil, en el cual se elige el punto de referencia  $r$ , y que  $B$  es un obstáculo fijo. El espacio de configuraciones será de dimensión 2, o de dimensión 3 si se permite que el objeto  $A$  rote.



**Figura 12.5:** *Objetos poligonales en un plano.*

Supóngase, en primer lugar, que el objeto  $A$  no puede rotar. El estudio en el espacio de configuraciones se realiza considerando el movimiento del punto  $r$  en vez de todo el objeto  $A$  para lo cual se aumentan los límites del obstáculo  $B$  de forma apropiada. Estos límites se obtienen deslizando el objeto  $A$  alrededor de  $B$  manteniendo fija su orientación (se supone que  $A$  no gira). De esta forma, se obtiene el polígono de cinco vértices  $B_A$  que se ilustra en la figura 12.6.  $B_A$  es un obstáculo en el espacio de configuraciones inducido por la traslación de  $A$ .

Para terminar de crear el espacio de configuraciones es necesario modificar también los límites del espacio de trabajo. De esta forma se llega al espacio que se muestra en la figura 12.7. Los movimientos de  $r$  que no producen colisión en este espacio



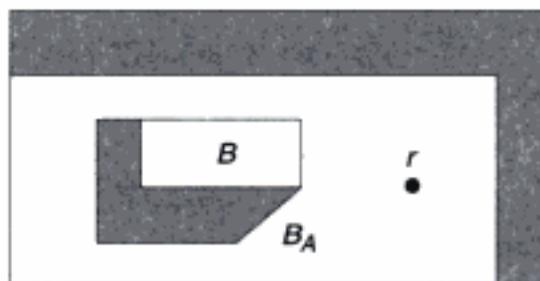
**Figura 12.6:** Crecimiento del obstáculo  $B$  inducido por la traslación de  $A$  dando lugar a  $B_A$ .

harán que tampoco se produzcan colisiones en el espacio de trabajo representado en la figura 12.5 cuando se traslada el objeto  $A$ .

Supuesto que  $A$  y  $B$  son polígonos convexos, si  $n_A, n_B, n_{BA}$  son, respectivamente, el número de vértices de  $A, B$  y  $B_A$ , entonces se verifica (Schilling, 1990):

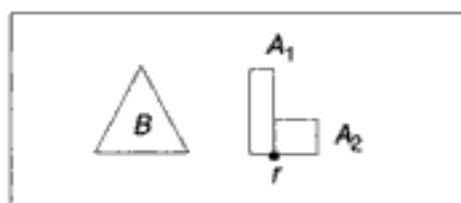
- $n_B \leq n_{BA} \leq n_A + n_B$ .
- $B_A$  es convexo.
- Si  $r \in A$ , entonces  $B \subseteq B_A$ .

Obsérvese que, en el ejemplo de las figuras, se cumple la relación entre el número de vértices. El hecho de que los polígonos sean convexos es importante ya que, como se verá posteriormente, los algoritmos de búsqueda de caminos adoptan normalmente esta hipótesis.

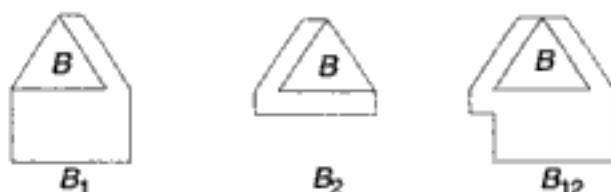


**Figura 12.7:** Espacio de configuraciones inducido por el movimiento del objeto  $A$ .

En cualquier caso, hay que poner de manifiesto que cuando los objetos son polígonos no convexos pueden considerarse formados por la unión de polígonos convexos, con lo cual los resultados anteriores son generales si, para objetos móviles no convexos, se considera como referencia un punto en la línea de unión de los polígonos convexos (punto de referencia común a los objetos convexos). Para ilustrar este concepto considérese el espacio de trabajo que se muestra en la figura 12.8, en la cual el objeto móvil  $A$  está formado por los polígonos  $A_1$  y  $A_2$ . Los obstáculos correspondientes en el espacio de configuraciones son los  $B_1$  y  $B_2$ , que se ilustran en la figura 12.9.



**Figura 12.8:** Objeto móvil formado por polígonos no convexos.

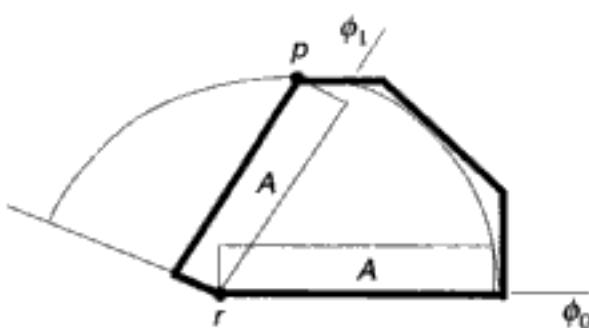


**Figura 12.9:** Obstáculos generados por polígonos no convexos en el espacio de configuraciones.

Teniendo en cuenta que el punto  $r$  es común, el espacio de configuraciones  $B_{12}$  inducido por la unión de  $A_1$  y  $A_2$  es la unión de los obstáculos  $B_1$  y  $B_2$  generados en dicho espacio por  $A_1$  y por  $A_2$ , respectivamente.

Existen programas que permiten obtener automáticamente el espacio de configuraciones (Schilling, 1990).

Cuando los objetos pueden rotar, la obtención del espacio de configuraciones es más complejo ya que los obstáculos generados ya no son poligonales sino que tienen líneas curvas. Una posible solución del problema consiste en agrandar los obstáculos, teniendo en cuenta que el objeto móvil puede cambiar de orientación en un cierto rango. Supóngase que el objeto  $A$  de la figura 12.10 puede rotar con respecto a un eje perpendicular al plano un cierto ángulo en un rango  $[\phi_0, \phi_1]$ . Si la distancia desde el vértice de referencia al punto más alejado  $p$  en el polígono  $A$  es  $d$ , el polígono rotado estará contenido en un sector circular de radio  $d$  y ángulo  $\Delta\phi \geq \phi_1 - \phi_0$ . El sector puede ser truncado teniendo en cuenta los límites de  $A$  y el sector truncado puede circunscribirse por un polígono, tal como se muestra en la figura 12.10.



**Figura 12.10:** Polígono resultante de considerar el giro de un obstáculo.

El empleo del espacio de configuraciones cuando puede variar la orientación es más complejo y requiere la utilización de técnicas de búsqueda de mayor coste computacional, como se pondrá de manifiesto en el apartado 12.5.

En muchos métodos no se considera explícitamente el caso de robots articulados, de forma que, por ejemplo, su aplicación directa a la planificación de caminos para brazos manipuladores resulta difícil ya que el espacio resultante de manipuladores con  $n$  articulaciones de rotación es un  $n$ -toro (Lumelsky y Sun, 1987).

Existen otros trabajos (Lozano-Pérez, 1987) dedicados expresamente a manipuladores, con articulaciones de rotación. El aspecto clave, al menos en lo que respecta a la aplicación a estos manipuladores, es la transformación de los objetos en el espacio de trabajo al espacio de configuraciones definido por las variables asociadas a la configuración del robot. De esta forma, en el espacio de configuraciones, los obstáculos se representan con relación a un determinado robot mediante regiones prohibidas en el espacio de variables articulares del robot. Cada punto en este espacio corresponde a una configuración del robot. En general, no todos los puntos del espacio de las variables articulares pueden ser alcanzados por el robot debido a los límites físicos de las variables articulares del robot.

En cualquier caso, es evidente que la complejidad computacional es creciente con el número de grados de libertad, por lo que su aplicación directa en tiempo real en entornos dinámicos resulta muy difícil. Sin embargo, se ha realizado un importante esfuerzo para mejorar su eficiencia. Estas mejoras se refieren tanto al propio cálculo del espacio de configuraciones (Branicky y Newman, 1990; Newman y Branicky, 1991) como a la aplicación de métodos de búsqueda considerando secuencialmente subespacios de dimensión más reducida del espacio de configuraciones.

## 12.5 DETECCIÓN Y EVITACIÓN DE COLISIONES EMPLEANDO MODELOS

---

En este apartado se consideran diversas técnicas que se emplean en sistemas robóticos tanto en tiempo real, durante la ejecución de movimientos, como antes de iniciar los movimientos, o incluso en funciones de ayuda al operador en teleoperación.

Se trata de detectar una posible colisión debido a los movimientos que se producen en el entorno del robot. En vehículos robóticos el problema es normalmente la detección de obstáculos en la dirección de movimiento del robot. El problema se complica cuando el robot tiene un cuerpo con articulaciones. Precisando más, para los robots manipuladores cabe distinguir colisiones entre:

- a) Distintos enlaces del manipulador.
- b) Enlaces y efecto final-carga.
- c) Enlaces y objetos del entorno.
- d) Efecto final-carga y objetos del entorno.

Para el caso a), las colisiones pueden preverse utilizando el modelo cinemático del manipulador. Los resultados pueden extenderse al caso b) disponiendo de un modelo geométrico de la carga. Con relación a los casos c) y d), es necesario considerar la complejidad del entorno y la movilidad de los posibles obstáculos. Aunque el entorno

sea esencialmente estático, es necesario tener en cuenta que las operaciones previas del robot pueden producir alteraciones en dicho entorno y que estas alteraciones deben ser consideradas para la planificación y ejecución de los movimientos posteriores. Por tanto, es necesario disponer de un modelo del entorno convenientemente actualizado. Por otra parte, conviene poner de manifiesto la necesidad de tener en cuenta las posibles imprecisiones en la ejecución de los movimientos, lo cual puede hacer necesario el empleo del sistema de percepción del entorno.

Es necesario también disponer de alguna técnica para detectar colisiones debidas a errores o imprecisiones en la ejecución de los caminos planificados.

Asimismo, se consideran como requisitos necesarios en la detección de obstáculos en tiempo real el evitar en lo posible falsas alarmas debidas a problemas relacionados con los sensores o con modelos demasiado conservadores, manteniendo la seguridad (paradas o/y avisos al operador) y la implantación consistente con requerimientos informáticos en tiempo real.

Los métodos de detección y evitación de colisiones que emplean los modelos se basan en la determinación de la intersección entre los espacios ocupados por el robot, su carga, y los objetos de su entorno.

Para su utilización en tiempo real, interesan modelos que permitan determinar de forma rápida la posibilidad de colisión y sean fácilmente actualizables para contemplar el movimiento del robot y, eventualmente, de objetos del entorno. Conviene indicar también que las representaciones para aplicación en tiempo real deben ser fiables pero no necesariamente muy precisas.

En Cameron (1985) se diferencian tres posibles enfoques para resolver el problema:

- 1) Muestreo del movimiento del robot un número finito de veces con detección de interferencias de objetos.
- 2) Construcción de modelos de los objetos en espacio y tiempo con determinación de intersecciones en el tiempo.
- 3) Modelado de volúmenes barridos por objetos móviles y búsqueda de posibles intersecciones.

Los enfoques 2) y 3) han sido utilizados en combinación con técnicas de planificación de caminos, pero, en principio, no parecen apropiados para su aplicación en tiempo real debido a su complejidad computacional. El primer enfoque implica una detección estática de obstáculos a intervalos de tiempo. Las consideraciones mencionadas anteriormente sobre el modelo son claves para la eficiencia del algoritmo de detección. En Preparata y Shamos (1985) se estudian diversos métodos alternativos para el cálculo de las intersecciones tanto en dos como en tres dimensiones. En general, la detección de colisiones mediante intersección de sólidos es computacionalmente compleja, por lo cual se hace necesario emplear estructuras eficientes y métodos para acelerar la búsqueda. En particular, pueden utilizarse estructuras tales como octrees o modelos aproximados basados en primitivas de sólidos, como los que se mencionaban en la apartado 12.4, de forma que los cálculos de intersección de primitivas sean relativamente fáciles. Si el movimiento es lento, y el tiempo del ciclo para detección de una colisión es pequeño, puede suponerse que durante el ciclo el modelo no varía. Este método ha sido empleado también en planificación de movimientos libres de obstáculos. Suponiendo que el algoritmo de planificación ha generado un camino, se trata

de comprobar si este camino está libre de colisión muestreando el movimiento planificado cada cierto intervalo de tiempo. En Roach y Boaz (1987) se sigue esta técnica construyendo a priori una estructura octree región para los objetos estáticos. En cada muestreo se construye un octree-región para cada objeto móvil y se realiza la determinación de intersecciones empleando ambas estructuras.

En Shafer y Herb (1992) se presenta un sistema para detección de colisiones en tiempo real en manipuladores. Se emplea un modelo con primitivas de sólidos (cilindro-esferas) para representar al manipulador y primitivas de sólidos rectangulares para el entorno. El cálculo de intersecciones se realiza empleando también una estructura de octree-región que permite una búsqueda jerarquizada. De esta forma, se consiguió que el ciclo completo de detección de una colisión para una configuración determinada del manipulador (procesamiento de valores de variables articulares, actualización del octree para reflejar movimiento y detección de la colisión) fuera del orden de unas pocas decenas de milisegundos, con un valor medio de 30 ms en un sistema UNIX con una CPU I80386 a 20 MHz. Los experimentos se realizaron simulando un robot con dos brazos en movimiento que podrían colisionar entre sí y con los objetos del entorno.

En detección de colisiones se emplean también algoritmos de cálculo de mínima distancia entre objetos. Cada vez que un objeto se mueve, se calcula la mínima distancia a cualquier otro objeto del entorno para averiguar si se va a producir una colisión. La dificultad principal es que los algoritmos de mínima distancia son computacionalmente costosos (Hurteau y Stewart, 1988). Para reducir el número de comparaciones, se encierran los objetos en cajas y se aplica el algoritmo de mínima distancia sólo a los pares cuyas cajas se supone que están suficientemente próximas.

Otra técnica para detección de colisiones inminentes consiste en la utilización de características del movimiento y, en particular, la velocidad. En Cheung y Lumelsky (1990) se estudia la velocidad de aproximación de dos sólidos en movimiento y la mínima distancia entre dos puntos de sus superficies. Si previamente se tienen definidos los límites máximos de la velocidad y distancia, estos valores pueden utilizarse para calcular el tiempo que transcurrirá antes de la posible colisión. Cuando este tiempo tiende a cero, la colisión es inminente. Para un espacio de trabajo con  $N$  objetos incluyendo el robot y los objetos de su entorno, en el caso más general, sería necesario considerar  $N^2$  posibles colisiones entre las primitivas que representan los objetos. En Foisy y otros (1990) se mejora el algoritmo básico organizando la búsqueda de colisiones empleando una lista en la cual se inserta en cada instante de muestreo el par de primitivas que tienen más posibilidad de colisionar.

Para acelerar la detección de colisiones, pueden emplearse también tablas de búsqueda precalculadas fuera de línea. Si se consideran fijos los objetos del entorno del robot, un posible método consiste en representar el espacio libre mediante una tabla resultante de la discretización del espacio de las variables articulares. Esta tabla puede utilizarse para seleccionar los objetos del entorno que están más próximos, reduciendo a éstos las intersecciones entre primitivas.

En Ollero y Gómez de Gabriel (1993) se presenta un método de detección de colisiones en manipuladores basado en la aproximación de los elementos del manipulador mediante cubos transformados. Se realiza una comprobación de las intersecciones de los poliedros que modelan el entorno y el modelo aproximado mediante cubos del manipulador. En Gómez de Gabriel (2000) se presenta un test rápido de colisión basado

en envolver el manipulador con una malla de puntos, realizando las intersecciones entre esta malla y los objetos del entorno modelados mediante poliedros convexos. En esta misma referencia se presenta también un test de colisiones que incorpora las ventajas de los anteriores acelerando su ejecución mediante una primera fase basada en la técnica de las cajas-frontera, que permiten descartar rápidamente objetos que no se encuentran próximos al robot.

En Cohen y otros (1995) se estudia la detección de colisiones entre objetos en movimiento. En el test de colisiones es posible tener en cuenta consideraciones dinámicas para realizar un preprocesamiento que elimina las caras de los objetos que no pueden colisionar debido a la velocidad actual y las limitaciones de aceleración.

## 12.6 PLANIFICACIÓN BASADA EN MODELO GEOMÉTRICO

### 12.6.1 Planteamiento del problema

Como se ha mencionado anteriormente, mediante estos métodos se trata de encontrar un camino desde una posición/orientación inicial  $S$  del robot hasta una posición/orientación final  $G$  (objetivo), de tal forma que no se produzcan colisiones y se optimice un cierto criterio. El criterio más usual es la minimización de la longitud del camino.

El problema, en su planteamiento general, es muy complejo, por lo que suelen adoptarse una serie de hipótesis simplificadoras entre las cuales las más comunes son:

- entorno conocido (modelo previo)
- entorno estático
- obstáculos modelados como polígonos o mediante segmentos lineales
- robot modelado como un punto
- robot desplazándose lentamente
- robot ejecutando perfectamente los caminos.

Las dos primeras hipótesis son muy frecuentes en los métodos de planificación que emplean modelos. Sin embargo, existen también métodos de planificación de robots en entornos dinámicos, y en particular en sistemas con múltiples robots que se mueven en el mismo espacio. Una posible técnica consiste en efectuar replanificaciones periódicas considerando el movimiento de los objetos del entorno. Si se pretende aplicar esta técnica en tiempo real, las replanificaciones deben ser lo suficientemente rápidas. Otra técnica alternativa, consiste en emplear métodos reactivos tales como los que se consideran en el apartado 12.7 (Hamilton y Dodds, 1998). En Mediavilla (2000) se revisan los métodos existentes para la planificación de trayectorias de múltiples manipuladores articulados y se presenta un método que combina una etapa fuera de línea, basada en modelos en el espacio de configuraciones que involucra una importante carga computacional (Mediavilla y otros, 1998), y una etapa en línea que utiliza un método reactivo.

La tercera hipótesis implica un planteamiento del problema en dos dimensiones. Este planteamiento es válido para robots móviles navegando en el plano pero, en ge-

neral, no es admisible para manipuladores, por lo que es necesario acudir a extensiones y generalizaciones de los métodos básicos.

De igual forma, el modelado del robot como un punto es una condición que sólo sirve como una referencia para obtener resultados que después pueden ser generalizados mediante el empleo de modelos apropiados del robot y su entorno. En particular, se emplean expansiones de los obstáculos tales como las que se mencionaban en el apartado 12.4.

Las hipótesis de movimientos lentos y de ejecución perfecta de los caminos transfieren los problemas de comportamiento dinámico y precisión a los niveles inferiores de control del robot y, obviamente, su aproximación a la realidad depende de las condiciones de operación y prestaciones exigidas al sistema de control de movimientos. En cualquier caso, resulta imprescindible tenerlas en cuenta para establecer condiciones seguras en la planificación.

### 12.6.2 Métodos en espacio cartesiano

Un método clásico de planificación de trayectorias que permite obtener los caminos más cortos libres de obstáculos es el de los *grafos de vértices o líneas de visibilidad*. Suponiendo que se trabaja en 2D, que el objeto móvil es un punto y que los obstáculos son convexos, se elige el camino más corto, formado por segmentos que conectan los vértices de los objetos.

En el método se definen todos los segmentos que conectan los vértices con  $S$ , con  $G$  y entre sí, eliminando las conexiones que intersectan obstáculos (se dejan exclusivamente las líneas de visibilidad, tal como se ilustra en la figura 12.11). Entre los segmentos que quedan se busca la secuencia que, empezando en  $S$  y terminando en  $G$ , define el camino más corto.

El método fue formulado inicialmente en 2D con el objeto móvil reducido a un punto y con obstáculos definidos mediante polígonos convexos. Expandingiendo los obstáculos puede aplicarse en robots no puntuales. Con algunas modificaciones simples, se aplica también en el caso de polígonos no convexos. Asimismo, existen extensiones a espacios en 3D pero, en este caso, no se garantiza que el camino sea el más corto.

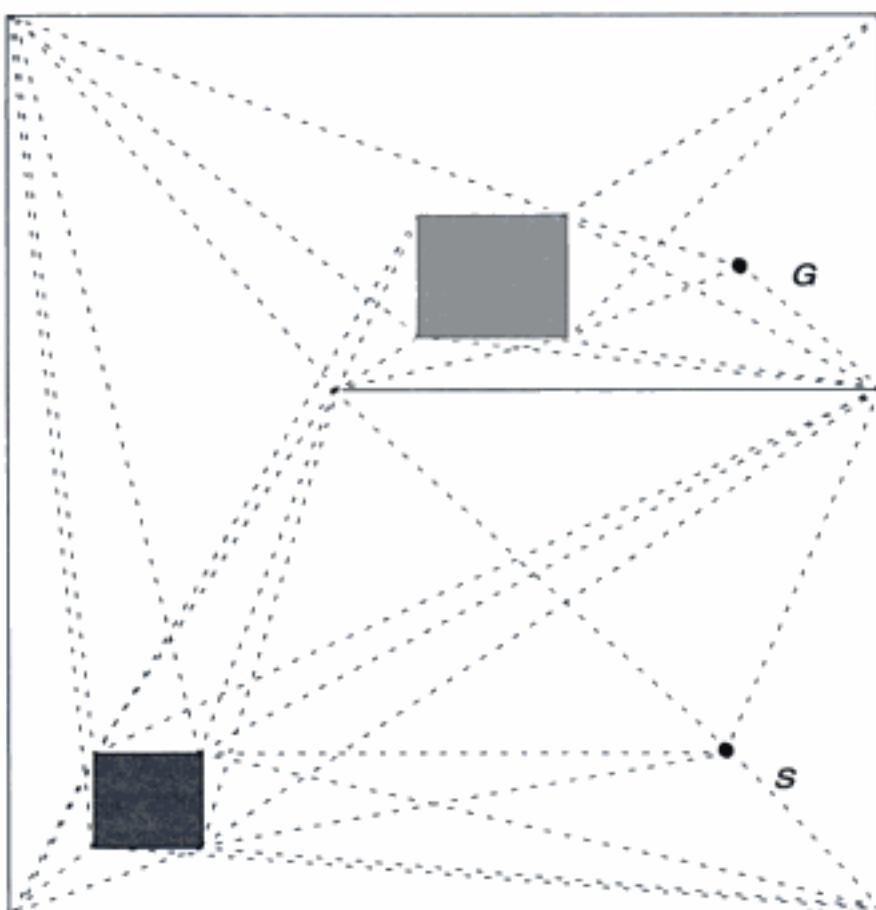
Por otra parte, el camino pasa muy cerca de los obstáculos, por lo cual resulta poco seguro si no se utilizan márgenes suficientemente amplios en la definición de los obstáculos.

Otro método muy utilizado es el de *búsqueda heurística* en el espacio cartesiano. Se trata de establecer un camino que optimice una función de evaluación que, suponiendo al objeto móvil en un estado  $x$  (posición/orientación), viene dada por

$$f(x) = g(x) + h(x) \quad (12.1)$$

siendo:

- $x$  el estado (posición/orientación del robot).
- $f$  la estimación del coste del camino desde el estado inicial pasando por  $x$  hasta el estado final.
- $g$  el coste del camino desde el estado inicial a  $x$ .
- $h$  la estimación del coste del camino desde  $x$  al estado final.



**Figura 12.11:** Gráficos de visibilidad.

El método se define inicialmente para un punto o una forma arbitraria móvil que mantiene en su movimiento una orientación fija. Puede extenderse a 3D sin modificar sus características.

El espacio se considera dividido en rejillas (o celdas en 3D). Comenzando en  $S$  se van explorando todos los vecinos  $n$ -conectados, típicamente 8 o 16 en 2D (18 o 26 en 3D), definiéndose una estructura de datos tipo árbol en cuyos nodos se evalúa la función. Se sitúan todos los puntos conectados en una lista abierta de acuerdo con el valor de la función de coste. La operación se repite para otro punto conectado que presente un valor mínimo de  $f$  y así hasta que se alcance  $G$  mediante una serie de puntos conectados. Cuando se exploran todos los caminos se elige el de mínima distancia. Si  $LA$  es una lista de estados candidatos para expansión y  $LC$  es una lista de estados expandidos, un posible algoritmo de búsqueda sería:

1. Calcular  $f(S)$  y situar  $S$  en  $LA$ .
2. Extraer de  $LA$  el  $x$  con menor valor de  $f$ . Llevar  $x$  a  $LC$ .
3. Terminar si  $x$  es el estado final.
4. Para cada estado  $x_i$  conectado a  $x$ :
  - a) Si  $x_i$  está en  $LA$ , entonces si nuevo  $g(x_i)$  es menor que antiguo  $g(x_i)$ , actualizar  $g(x_i)$  y  $f(x_i)$  y redireccionar puntero.

- b) Si  $x_i$  está en LC, entonces si nuevo  $g(x_i)$  es menor que antiguo  $g(x_i)$ , entonces redireccionar puntero.
  - c) Si  $x_i$  no está en ninguna lista, entonces calcular  $f(x_i)$  y situar  $x_i$  en LA.
- 5) Si LA está vacía, terminar sin ningún camino.
  - 6) Volver al paso 2.

El método puede aplicarse también considerando modelos geométricos mediante primitivas y realizando pruebas de intersección para determinar posibles colisiones en los caminos. Es decir, sólo se expanden los nodos cuando no existe colisión.

Como se ha mencionado anteriormente, el método se aplica directamente suponiendo que la orientación no varía. Sin embargo, si se pretende planificar un movimiento entre  $S$  y  $G$  con orientaciones diferentes, es necesario considerar también una cuarta dimensión. Para ello puede discretizarse un ángulo de orientación en intervalos y proceder de igual forma con celdas en 4D. Obviamente, la complejidad computacional aumenta de forma importante. Si se pretende considerar los tres ángulos de orientación en el espacio, se necesita trabajar en un espacio en 6D, con lo cual la búsqueda se convierte en prohibitiva.

Es posible también introducir consideraciones heurísticas de acuerdo con el entorno de trabajo e imponer restricciones en el movimiento tales como, por ejemplo, imponer que el cambio de orientación se realice antes de dejar  $S$  o después de llegar a  $G$ , verificándose en ambos casos la ausencia de colisión en la rotación.

Otro enfoque diferente del problema consiste en razonar en términos del espacio libre y no de los obstáculos como se ha venido haciendo. Este enfoque fue propuesto por Brooks (1983) modelando el espacio libre mediante la superposición de los denominados "conos generalizados" (formas generadas moviendo una sección bidimensional a lo largo de una curva en el espacio). Se trata de buscar cilindros conectados de forma que el robot se mantenga centrado dentro de los cilindros. Para construir los cilindros generalizados se procede determinando, en primer lugar, el contorno del cilindro y, a continuación, los rangos de orientación para mantener el robot en el cilindro. Después, se establecen correspondencias entre rangos de orientación y cilindros superpuestos. En 2D el tiempo de cálculo es proporcional al cubo del número de aristas de los obstáculos. El método presenta dificultades en espacios estrechos. Su generalización a 3D es difícil y, en este caso, la complejidad es proporcional al número de caras de los obstáculos elevado a cuatro.

Los conceptos de búsqueda del espacio libre de Brooks también se emplean en otros métodos. En particular, se utilizan representaciones explícitas del espacio libre basándose en diagramas de Voronoi generalizados (Lee y Drysdale, 1981). Un diagrama de Voronoi generalizado del espacio libre es el lugar de los puntos que son equidistantes de dos o más límites de obstáculos. Una vez que se obtiene este diagrama, se busca el camino más corto empleando métodos de la teoría de grafos. En Preparata y Shamos (1985) se estudian las propiedades de los diagramas de Voronoi.

La complejidad de estos métodos es función del número de segmentos que delimitan los contornos del robot y los obstáculos, así como de los grados de libertad en los movimientos del robot. Esta complejidad puede evaluarse de forma aproximada mediante:

$$(C_r C_o)^r \quad (12.2)$$

siendo

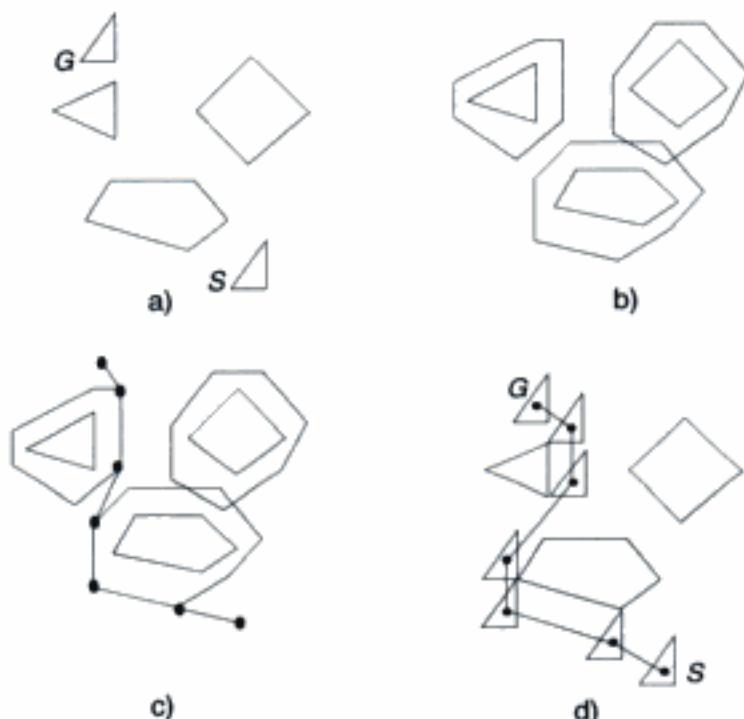
- $C_r$  el número de segmentos delimitando el contorno del robot.
- $C_o$  el número de segmentos delimitando el obstáculo.
- $r$  los grados de libertad del robot.

### 12.6.3 Métodos en espacio de configuraciones

En el apartado 12.4 se expusieron diversos conceptos relacionados con el modelado en el espacio de configuraciones. Se trata ahora de emplear estos conceptos para la planificación de caminos. Sea  $A$  un objeto móvil representado mediante un polígono convexo con un punto de referencia  $r$ . Sean  $B^k$ ,  $1 \leq k \leq m$  obstáculos poligonales convexos. Se trata de mover el objeto  $A$  desde una posición en la que  $r$  coincide con un punto origen  $S$ , hasta otra en la que  $r$  coincide con el punto destino  $G$  (figura 12.12).

Supóngase que el polígono  $A$  no puede rotar. En este caso, el camino más corto de  $S$  hasta  $G$  es el formado por los segmentos rectos  $r^k$ ,  $1 \leq k \leq m$ , cuyos puntos interiores ( $1 < k < n$ ) son vértices de los obstáculos en el espacio de configuración inducidos por  $A$ :  $B_A^k$ ,  $1 \leq k \leq m$ . Por tanto, el método de planificación consiste en ir explorando el conjunto finito de vértices hasta encontrar la secuencia de menor longitud. Para ello pueden emplearse métodos tales como los basados en la teoría de grafos (Gondran y otros, 1984).

Considérese el ejemplo de la figura 12.12 (Schilling, 1990) en la cual, a la representación de la figura a) le corresponde la b) en el espacio de configuraciones induci-

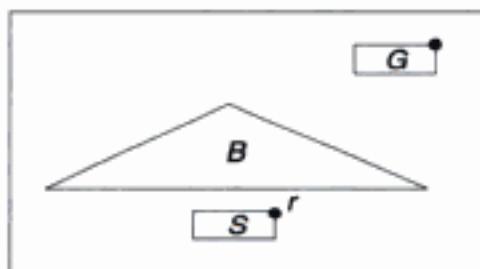


**Figura 12.12:** Planificación de caminos en el espacio de configuraciones con orientación constante.

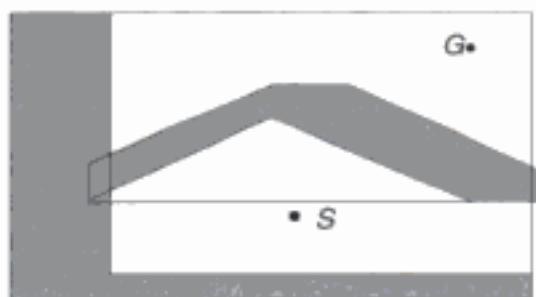
do por A. En la c) se muestra el camino más corto en el espacio de configuraciones. El correspondiente camino en el espacio original se muestra en la figura d).

Nótese como el camino óptimo (distancia recorrida mínima) hace que el móvil se desplace por el borde de los obstáculos. Evidentemente, es necesario aumentar los límites de los obstáculos más de lo que le correspondería con objeto de evitar colisiones que pueden producirse por incertidumbres en la estimación de la posición del móvil o en la ejecución del camino si los límites no se aumentan lo suficiente.

Cuando se tiene en cuenta también la rotación del móvil, el problema se complica. Considérese el problema mostrado en la figura 12.13 (Schilling, 1990), en el cual se pretende desplazar el objeto rectangular desde la posición origen  $S$  a la posición destino  $G$ . El espacio de configuraciones generado con  $\phi = 0$  se muestra en la figura 12.14 y el generado con  $\phi = \pi/2$  se muestra en la figura 12.15. Obsérvese que no existe ninguna solución del problema con  $\phi = 0$ , mientras que, por el contrario, es posible alcanzar la solución si la orientación es  $\phi = \pi/2$ . Para resolver en un caso general estos problemas, es necesario realizar una búsqueda en el espacio de configuraciones  $\{x, y, \phi\}$ .



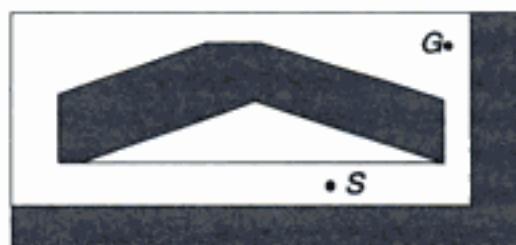
**Figura 12.13:** Ejemplo para la determinación del espacio de configuraciones con rotación de objetos.



**Figura 12.14:** Espacio de configuraciones con  $\phi = 0$ .

Este espacio está constituido por trozos de superficies curvas rotantes generadas por líneas de translación y rotación. El espacio con orientación fija puede considerarse como una sección del espacio completo para un valor determinado de  $\phi$ .

El método de resolución consiste en la aproximación de obstáculos mediante proyecciones para distintos intervalos de valores de  $\phi$ , realizando la búsqueda de un camino seguro para cada intervalo.



**Figura 12.15:** Espacio de configuraciones con  $\phi = \pi/2$ .

El área barrida al variar  $\phi$  en la sección se aproxima mediante unión de polígonos. Las proyecciones de secciones forman el espacio de configuraciones inducido por  $A$  en el intervalo de valores de  $\phi$  que se está considerando.

La exploración consiste en una búsqueda secuencial para todos los intervalos. Nótese que la magnitud del intervalo (precisión) no tiene que ser la misma en todo el rango de variación de  $\phi$ .

Cuando  $A$  es un sólido tridimensional que puede rotar, el espacio de configuraciones inducido por  $A$  es un objeto complejo con seis dimensiones. En este caso, puede aplicarse también el método con proyecciones en subespacios de dimensión menor para buscar una solución del problema original en un espacio de mayor dimensión en el que no todas las soluciones pueden ser encontradas. En general, no se garantiza que se encuentre una solución.

Existen numerosos trabajos en los cuales los modelos permiten considerar polígonos móviéndose entre polígonos, poliedros moviéndose entre poliedros, o elipses moviéndose entre elipses. En estos trabajos se formulan métodos de planificación de caminos globales y óptimos que encuentran un camino, si existe, en el espacio de configuraciones.

En los robots manipuladores el espacio de configuraciones está definido por las variables articulares. Cada punto en el espacio articular corresponde a una única configuración. Nótese que, debido a los límites en las articulaciones, no pueden alcanzarse todos los puntos en espacio articular.

Las dificultades que se presentan son la dimensionalidad alta del espacio de configuraciones y la complejidad de los obstáculos en este espacio. El límite de tiempo para el peor caso crece exponencialmente con la dimensionalidad del espacio de configuración.

Por ello, es necesario emplear estrategias simplificadoras para la construcción y búsqueda en subespacios de dimensión inferior. Entre estas estrategias pueden contemplarse:

- 1) Planificar un camino para las tres primeras articulaciones y utilizar una aproximación conservadora para las restantes (Lozano-Pérez, 1987). Los puntos origen y destino de este camino son las proyecciones en el subespacio tridimensional de las posiciones/orientaciones actuales origen y destino. Una vez que se ha encontrado el camino con esta técnica, se realiza una búsqueda completa de dimensión 6 en las regiones próximas al origen y destino. La dificultad es que la aproximación conservadora puede hacer que se olviden caminos libres de obstáculos.

- 2) Construcción secuencial de un subespacio del espacio libre (Lozano-Pérez, 1987). Se construye el espacio de configuraciones (C-espacio) de obstáculos para los dos primeros enlaces encontrando una secuencia de regiones del espacio libre que contiene un camino desde origen a destino, y se expanden sucesivamente las porciones del espacio libre para incluir la siguiente articulación. Para un manipulador con  $n$  enlaces se construye, en primer lugar, un C-espacio bidimensional, a continuación un subconjunto tridimensional para el tercer enlace, hasta llegar a un subconjunto  $n$  dimensional para el enlace  $n$ . El tiempo de búsqueda se reduce pero, en cualquier caso, hay que realizar una búsqueda de dimensión 6, aunque ésta sea en un subconjunto del espacio  $n$  dimensional completo.
- 3) Planificación secuencial con empleo de parametrización. Consiste en los pasos: planificación para los tres primeros enlaces ( $q_1, q_2, q_3$ ), parametrización del camino con un parámetro  $p$ , planificación del camino para los dos enlaces siguientes con  $(p, q_4, q_5)$  y así sucesivamente. Alternativamente, existen formulaciones (Gupta, 1990) que permiten reducir a un espacio de una dimensión y  $n-1$  espacios bidimensionales.

En cualquier caso, interesa considerar con detalle la construcción del espacio de configuraciones debido a su relevancia en la eficiencia de los algoritmos de planificación. Hay que señalar que, aunque se emplee un método de planificación en el espacio de configuraciones, existen técnicas con las que no es necesario construir explícitamente el espacio de configuraciones. Sin embargo, el cálculo previo de dicho espacio permite organizar sus regiones libres en un grafo que posteriormente se explora para generar los caminos.

El espacio de configuraciones puede construirse empleando algoritmos de interacción de objetos y detección, los cuales pueden ser ejecutados también como un paso del algoritmo de búsqueda de caminos. Estos algoritmos tratan con polígonos moviéndose entre polígonos en el plano, o poliedros moviéndose entre poliedros en el espacio, o también elipses entre elipses.

Muchas de las técnicas presentadas no son aplicables a espacios de configuración de brazos manipuladores con articulaciones de rotación. Como se mencionaba en el apartado 12.3, el espacio de uno de estos manipuladores con  $n$  grados de libertad es un  $n$ -toro. Con objeto de simplificar el problema de planificación de caminos para estos manipuladores, se siguen estrategias tales como la determinación del espacio de configuraciones en una dimensión más baja obteniendo soluciones aproximadas.

Se han estudiado también propiedades del espacio de configuraciones de manipuladores con articulaciones de rotación para facilitar la generación de este espacio (Newman y Branicky, 1991).

## 12.7 MÉTODOS REACTIVOS

Se trata de métodos que generan una respuesta rápida a información de sensores de proximetría sin considerar modelos geométricos del entorno. Sin embargo, para diversos autores, constituyen también métodos de planificación en los cuales se prima la reactividad, integrándose las funciones de planificación con las de control de los movimientos.

### 12.7.1 Campos potenciales

El *método de campos potenciales* se basa en considerar el movimiento en un campo de fuerzas con los obstáculos generando fuerzas repulsoras y la posición objetivo una fuerza atractora (Khatib, 1986, 1987). El método se ha formulado tanto para robots móviles como para manipuladores, aunque ha tenido más difusión en su aplicación a robots móviles, para los cuales se ha aplicado también en planificación de caminos fuera de línea (Thorpe, 1984).

El método de los campos potenciales generalizados permite considerar también la velocidad del robot en la proximidad de un obstáculo (Krogh, 1984). Aunque la aplicación típica es en planificación local, este método ha sido utilizado combinando planificación local y global (Krogh y Thorpe, 1986).

Considérese un robot en la posición  $\rho$  que circula con una velocidad  $v$ , tal como se muestra en la figura 12.16. Sea  $O_i$  un obstáculo en el entorno del robot y  $p_i$  el punto del obstáculo más cercano al robot (único si  $O_i$  es convexo). Sea  $\hat{N}_i$  el vector unitario dirigido de  $p_i$  al centro de guiado del robot.

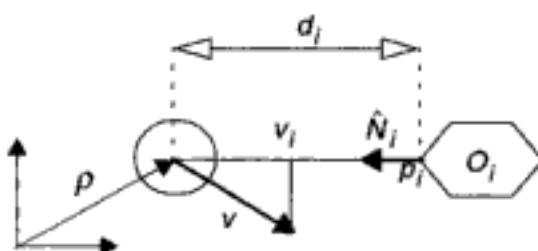


Figura 12.16: Robot aproximándose a obstáculo.

Si  $d_i$  es la distancia actual a un obstáculo,  $v_i$  es la proyección del vector velocidad actual del robot sobre  $\hat{N}_i$ , y  $\alpha$  es la deceleración máxima del robot, el tiempo mínimo para evitar el obstáculo será:

$$t_{min} = \frac{v_i}{\alpha} \quad (12.3)$$

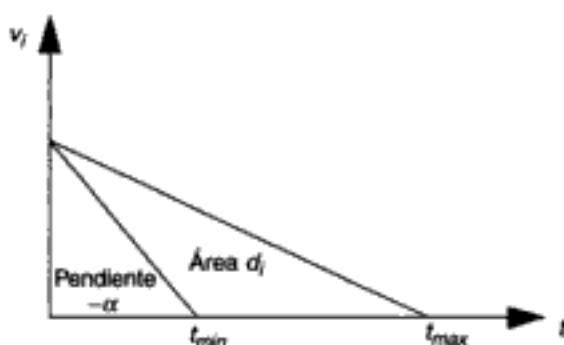
Puede comprobarse que este es el valor de  $t$  para el cual el área bajo la línea de  $v_i$ , que se muestra en la figura 12.17, es igual a  $d_i$ .

Por otra parte, si se aplica una deceleración constante inferior a  $\alpha$ , la colisión se evita si  $v_i$  se hace cero antes de que el robot recorra una distancia  $d_i$  hacia el obstáculo. Por tanto, el tiempo máximo para evitar la colisión (tiempo que tarda en pararse) aplicando una deceleración constante es:

$$t_{max} = \frac{2d_i}{v_i} \quad (12.4)$$

La función de potencial generalizada con respecto al obstáculo  $O_i$  se define como:

$$P_i(\rho, v) = 0 \text{ si } v_i \leq 0 \quad (12.5)$$

Figura 12.17: Ilustración de  $t_{min}$  y  $t_{max}$ .

$$P_i(\rho, v) = \frac{1}{t_{max} - t_{min}} = \frac{\alpha v_i}{2d_i\alpha - v_i^2} \text{ si } v_i > 0 \quad (12.6)$$

Obsérvese como el valor de la función tiende a infinito cuando el tiempo de reserva hasta la colisión ( $t_{max} - t_{min}$ ) tiende a cero, y tiende a cero cuando la velocidad tiende a cero. La función no está definida cuando  $t_{max} \leq t_{min}$  indicando que, en este caso, la colisión no puede evitarse.

El vector de evitación de un obstáculo  $c_{o,i}$  se define como:

$$c_{o,i} = -\nabla_\rho P_i = 0 \text{ si } v_i \leq 0 \quad (12.7)$$

$$c_{o,i} = -\nabla_\rho P_i = \frac{\partial}{\partial \rho} P_i = \frac{\partial P_i}{\partial d_i} \hat{N}_i = \frac{2\alpha^2 v_i}{(2d_i\alpha - v_i^2)^2} \hat{N}_i = \frac{2P_i^2}{v_i} \hat{N}_i \text{ si } v_i > 0 \quad (12.8)$$

Este vector dirige al robot en dirección contraria al obstáculo (dirección  $\hat{N}_i$ ).

La planificación mediante el método de los campos potenciales se realiza según los pasos:

- 1) Considerar que cada obstáculo genera una fuerza repulsora normal a su superficie, y la posición objetivo una fuerza atractora.
- 2) Calcular el vector de fuerzas resultante  $R$ .
- 3) Calcular una nueva posición para el robot resultante de aplicar una fuerza aceleradora  $R$  y vuelta al paso 1).

Alternativamente, es posible considerar que cada lectura de un sensor de proximidad genera una fuerza repulsora proporcional (Brooks, 1986; Brooks y Flynn, 1989). Si la magnitud de la suma de las fuerzas repulsoras excede un cierto umbral, el robot se para, vuelve según la dirección de la fuerza resultante, y comienza otro movimiento.

Otro método de planificación reactiva es el de las *fuerzas virtuales*. Desarrollado también para robots móviles (Borenstein y Koren, 1989), está igualmente concebido para evitación rápida de obstáculos. En este método se combinan las técnicas de las rejillas de certidumbre propuestas por Moravec y Elfes (1985) con los campos poten-

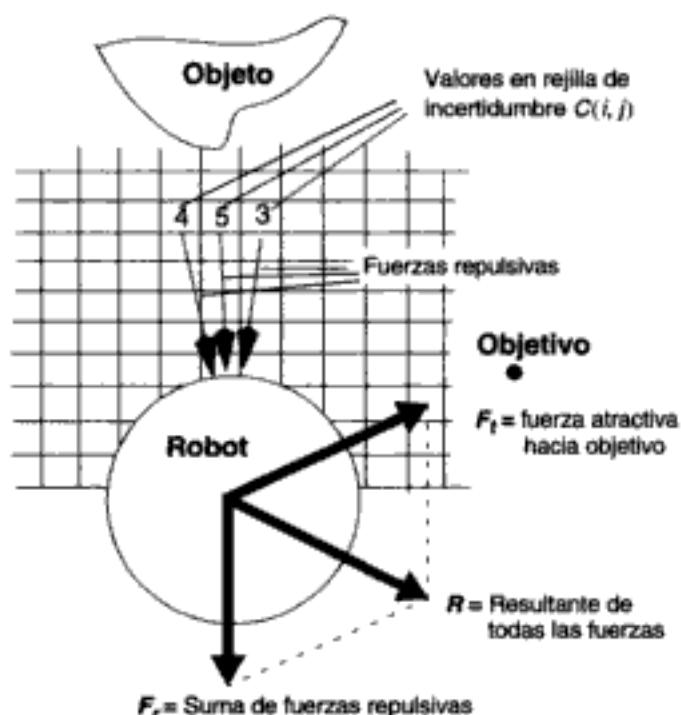


Figura 12.18: Método de las fuerzas virtuales.

ciales. El método se ilustra en la figura 12.18 (Borenstein y Koren, 1989). Se considera el espacio de trabajo del robot, en dos dimensiones, dividido en celdas (rejilla). Se asocia a cada celda una probabilidad de estar ocupada por obstáculos. Estas probabilidades se obtienen a partir de los datos de sensores, típicamente sensores de ultrasonidos, teniendo en cuenta la incertidumbre asociada a las medidas. La probabilidad se actualiza mediante nuevas lecturas de los sensores cuando el robot navega.

Cada celda ocupada aplica una fuerza repulsiva sobre el robot. La magnitud de esta fuerza es proporcional al contenido de la celda  $C(i, j)$  [nivel de certidumbre de la celda  $(i, j)$ ] e inversamente proporcional al cuadrado de la distancia  $d(i, j)$  entre el robot y la celda  $(i, j)$ :

$$F(i, j) = \frac{F_{cr} C(i, j)}{d^2(i, j)} \left[ \frac{(x_i - x_a)}{d(i, j)} \hat{X} + \frac{(y_i - y_a)}{d(i, j)} \hat{Y} \right] \quad (12.9)$$

siendo  $F_{cr}$  una fuerza constante de repulsión,  $(x_a, y_a)$  las coordenadas actuales del robot,  $(x_i, y_i)$  las coordenadas de la celda  $(i, j)$ , y  $\hat{X}$  e  $\hat{Y}$  vectores unitarios en las direcciones  $x$  e  $y$  respectivamente. Obsérvese que la división por el cuadrado de la distancia genera fuerzas repulsivas de intensidad elevada cuando el robot está muy próximo a una celda ocupada. La fuerza repulsiva resultante  $F_r$  es la suma vectorial de las fuerzas individuales generadas por todas las celdas:

$$F_r = \sum_{i, j} F(i, j) \quad (12.10)$$

El robot es atraído a la posición objetivo ( $x_o, y_o$ ) mediante una fuerza atractora  $F_o$  de magnitud constante:

$$F_o = F_{co} \left[ \frac{(x_o - x_r)}{d(o)} \hat{X} + \frac{(y_o - y_r)}{d(o)} \hat{Y} \right] \quad (12.11)$$

siendo  $F_{co}$  una constante y  $d(o)$  la distancia entre el robot y el objetivo.

La suma vectorial de todas las fuerzas repulsivas de las celdas ocupadas y la fuerza de atracción a la posición objetivo produce el vector de fuerza resultante:

$$R = F_o + F_r \quad (12.12)$$

El ángulo de dirección del vector unitario  $R/|R|$ , expresado en grados, se utiliza como referencia de dirección (incremento de dirección) en el servo de dirección del robot. En la implantación práctica se utilizan también filtros de paso bajo para suavizar el movimiento y amortiguamientos para evitar oscilaciones y poder realizar movimientos paralelos a obstáculos (Borenstein y Koren, 1989). La velocidad del robot  $v$  se controla también mediante las expresiones:

$$v = V_{max} \text{ para } |F_r| = 0 \text{ (ausencia de obstáculos)} \quad (12.13)$$

$$v = V_{max}(1 - |\cos \theta|) \text{ para } |F_r| > 0 \quad (12.14)$$

siendo  $\cos \theta$  el coseno direccional del ángulo entre los vectores suma de fuerzas repulsoras  $F_r$  y velocidad  $v$ , que se calcula según:

$$\cos \theta = \frac{v_x F_{rx} + v_y F_{ry}}{|v| |F_r|} \quad (12.15)$$

donde  $v_x, v_y$  son las componentes  $x$  e  $y$  del vector  $v$ , y  $F_{rx}, F_{ry}$  las componentes  $x$  e  $y$  del vector  $F_r$ .

El método también puede aplicarse en teleoperación sustituyendo la fuerza hacia el objetivo por una fuerza virtual cuya dirección y magnitud la establece el operador con el dispositivo de teleoperación.

En general, los métodos de reacción directa comentados son típicamente locales (utilizan únicamente información local) y, empleados sin ninguna técnica adicional de planificación, dan origen a una menor eficiencia con posibilidad de comportamientos oscilantes del robot. Otro problema típico es que el robot queda atrapado en mínimos locales. Nótese también que los pares y fuerzas que puede generar el sistema de potencia del robot son limitados, por lo que en la práctica pueden no garantizar que se evite el obstáculo o conducir a una estrategia muy conservadora.

Posteriormente, estos métodos se han combinado con métodos de planificación local como los comentados en apartados precedentes. En Horsch y Nolzen (1992) se realiza una planificación global para las tres primeras articulaciones de un manipulador en el espacio de las configuraciones y movimientos reactivos en las tres últimas.

Por último, hay que señalar que la aplicación de estos métodos reactivos, así como los más recientes basados en el comportamiento, tiene implicaciones en la arquitectu-

ra del sistema de control inteligente (capítulo 6). Como se puso de manifiesto en dicho capítulo, estos métodos pueden aplicarse también en arquitecturas jerarquizadas más convencionales.

## 12.8 CONCLUSIONES

La detección de colisiones y planificación de caminos son funciones de interés obvio en robótica. En este capítulo se introducen conceptos básicos sobre detección y evitación de colisiones y planificación de caminos óptimos libres de colisión, distinguiendo entre diferentes posibilidades de aplicación: en tiempo real, ayuda a los operadores, o fuera de línea como ayuda a la programación del robot.

Las características del modelo geométrico del robot y su entorno tienen una notable influencia en la eficiencia de los métodos de detección de colisiones y planificación de caminos. En el capítulo se han presentado distintos tipos de modelos incluyendo los que emplean estructuras jerarquizadas, que facilitan la búsqueda, y los modelos en el espacio de configuraciones, que tienen como objetivo la aplicación de métodos de planificación de caminos.

Por lo que respecta a los problemas de detección y evitación de colisiones, se distingue entre los métodos de reacción directa a información de sensores y los métodos basados en modelos geométricos del entorno.

Se presentan métodos de detección y evitación de colisiones basados en modelos geométricos y se aborda también la planificación empleando modelos introduciendo métodos de búsqueda en el espacio cartesiano y métodos en el espacio de configuraciones. Por último, se presentan métodos reactivos que pueden generar directamente órdenes al controlador de movimientos a partir de las medidas obtenidas con sensores, típicamente sensores de rango.

## 12.9 REFERENCIAS

- Borestein J. y Y. Koren, 1989. "Real-time obstacle avoidance for fast mobile robots". IEEE Transactions on Systems Man and Cybernetics. Vol. 19, pp. 1179-1187.
- Branicky M.S. y W.S. Newman, 1990. "Rapid computation of configuration space approaches". Proc. IEEE Intl. Conference on Robotics and Automation, pp. 304-310.
- Brooks R.A., 1983. "Solving the find-path problem by good representation of free space". IEEE Transactions on System Man and Cybernetics. Vol. 13, pp. 190-197.
- Brooks R. A., 1986. "A robust layered control system for a mobile robot". IEEE Journal of Robotics and Automation, RA-2, pp. 14-23.
- Brooks R.A. y M.A. Flynn, 1989. "Robot Beings". Proc. of IEEE Int. Workshop on Intelligent Robots and Systems, pp. 2-9, Tsukuba (Japón).
- Cameron S., 1985. "A study of the clash detection problem in robotics". Proc. IEEE International Conference on Robotics and Automation. St. Louis, Mo. (USA). pp. 488-493.
- Cheung E. y V. Lumelsky, 1990. "Motion planning for a whole-sensitive robot arm". Proc. of the IEEE International Conference on Robotics and Automation, Cincinnati, O. (USA). pp. 344-349.

- Cohen C., M.C. Lin, D. Manocha y M. Ponangi, 1995. "I Collide: An interactive and exact collision detection system for large-scale environments". Proc. of the 1995 Symposium on Interactive 3D Graphics.
- Elfes A., 1987. "Sonar based real world mapping and navigation". IEEE J. of Robotics and Automation. Vol. RA-3, núm. 3.
- Foisy A., V. Hayward y S. Aubry, 1990. Proc. IEEE International Conference on Robotics and Automation, pp. 338-343. Cincinnati, O. (USA).
- Fujimura K. y H. Sammet, 1988. "Path planning among moving obstacles using spatial indexing". Proc. IEEE International Conference on Robotics and Automation. Philadelphia, Pa. (USA), pp. 1662-1667.
- Gómez de Gabriel, J.M., 2000. "Contribuciones a la teleoperación de robots con retardos de comunicación". Tesis doctoral. Universidad de Málaga.
- Gondran M., M. Minoux y S. Vajda, 1984. "Graphs and Algorithms", Wiley.
- Gupta K.K., 1990. "Fast collision avoidance for manipulator arms: a sequential search strategy". IEEE Trans. Robotics and Automation. Vol. 6, núm. 5, pp. 522-532.
- Hamilton K. y G.I. Dodds (1998). "Reactive planning of robot arms in single and cooperative tasks". Proc. of the IEEE International Conference on robotics and Automation, pp. 336-340.
- Hayward V., 1986. "Fast collision detection scheme by recursive decomposition of a manipulator workspace". Proc. IEEE Intl. Conference on Robotics and Automation, San Francisco, Ca. (USA). Vol. 2, pp. 1056-1063.
- Herman M., 1986. "Fast three-dimensional, collision-free motion planning". Proc. IEEE International Conference on Robotics and Automation, San Francisco, Ca., pp. 1056-1063.
- Hopcroft J.E., J. Schwartz y M. Sharir, 1983. "Efficient detection of intersections among spheres". Intl. J. Robotics Research. Vol. 2, No. 4, pp. 77-80.
- Horsch Th. y H. Nolzen, 1992. "Local motion planning avoiding obstacles with dual quaternions". Proc. IEEE Conference on Robotics and Automation. Niza (Francia), pp. 241-245.
- Hurteau G. y N.F. Stewart, 1988. "Distance calculation for imminent collision indication in a robot system simulation". Robotica. Vol. 6, núm. 1, pp. 47-51.
- Khatib O., 1986. "Real time obstacle avoidance for manipulators and mobile robots". IEEE Journal of Robotics and Automation. Vol 2, pp. 90-98.
- Khatib O., 1987. "A unified approach for motion and force control of robot manipulators: the operational space formulation". IEEE Robotics and Automation. Vol. 3, pp. 43-53.
- Krogh B.H., 1984. "A generalized potential field approach to obstacle avoidance control". Intl. J. Robotic Research. Bethlehem, Pa. (USA).
- Krogh B.H. y C. Thorpe, 1986. "Integrated path planning and dynamic steering control for autonomous vehicles". Proc. IEEE Conference Robotics and Automation", San Francisco, Calif. (USA), pp. 1664-1669.
- Lee D. T. y R.L. Drysdale, 1981. "Generalized Voronoi diagrams in the plane". SIAM J. Computers. Vol. 10, núm. 1, pp. 73-87.
- Lozano-Pérez T. y M.A. Wesley, 1979. "An algorithm for planning collision free path among polyhedral obstacles". Communications of ACM. Vol. 22, pp. 560-570.
- Lozano-Pérez T., 1980. "Spatial planning: A configuration space approach". IEEE Trans. on Computers, Vol. C-32, pp. 108-120.
- Lozano-Pérez T., 1981. "Automatic planning of manipulator transfer movements". IEEE Trans. Syst., Man, Cyber. Vol. SMC-11, pp. 681-689.
- Lozano-Pérez T., 1987. "A simple motion-planning algorithm for general robot manipulators". IEEE J. of Robotics and Automation. Vol. RA-3, núm. 3, pp. 224-238.

- Lumelsky V. y K. Sun, 1987. "Gross motion planning for a simple 3D articulated robot arm moving amidst unknown arbitrarily shaped obstacles". Proc. IEEE Intl. Conference on Robotics and Automation, pp. 1929-1934.
- Mediavilla M., J.C. Fraile, J.R. Perán, G.I. Dodds (1998). "Optimization of collision free trajectories in multi-robot systems". IEEE International Conference on Robotics and Automation, ICRA'98, pp. 2910-2915, Leuven, Bélgica.
- Mediavilla M., 2000. "Planificación de trayectorias para sistemas multirrobot en entornos dinámicos". Tesis Doctoral, diciembre. Universidad de Valladolid.
- Moravec H. y A. Elfes, 1985. "High resolution maps from a wide-angle sonar". Proc. IEEE Conference on Robotics and Automation.
- Newman W.S. y M.S. Branicky, 1991. "Real-time configuration space transform for obstacle avoidance". Intl. J. Robotics Research, pp. 650-667.
- Ollero A. y J. Gómez de Gabriel, 1993. "Detección de colisiones en un sistema telerobótico". Tercer Congreso de la AER, Zaragoza, pp. 259-264.
- Preparata F.P. y M.I. Shamos, 1985. "Computational geometry. An introduction". Springer-Verlag.
- Roach J.W. y M.N. Boaz, 1987. "Coordinating the motions of robot arms in a common workspace". IEEE J. Robotics and Automation. Vol. 3, núm. 5, pp. 437-444.
- Samet H., 1989. "Application of spatial data structures". Computer Graphics, Image Processing and GIS. Addison Wesley.
- Samet H. y M. Tamminen, 1985. "Bintrees, CSG trees, and time". Computer Graphics. Vol. 19, núm.3, pp. 121-130.
- Schilling R.J., 1990. "Fundamentals of Robotics. Analysis and control". Prentice Hall.
- Shaffer C.A. y G.M. Herb, 1992. "A real-time robot arm collision avoidance system". IEEE Trans. Robotics and Automation". Vol. 8, núm. 2, pp. 149-160.
- Thorpe C., 1984. "FIDO: Vision and navigation for a robot rover". Ph. D. Thesis, Carnegie Mellon University.

En este capítulo se presentan conceptos, métodos y tecnologías sobre teleoperación y telerrobótica. Se estudian sistemas de control en telerrobótica, con especial énfasis en los sistemas bilaterales maestro-esclavo para la manipulación. Se consideran también las técnicas para reducir el efecto de los retardos de transmisión de información. Asimismo, se estudian las técnicas de visualización y las interfaces persona-máquina en teleoperación. En el capítulo se introducen también nuevas tecnologías de telepresencia relacionadas con las que se emplean en realidad virtual.

## 13.1 INTRODUCCIÓN Y CONCEPTOS BÁSICOS

La intervención del operador humano es imprescindible en un gran número de las aplicaciones de la robótica, especialmente en entornos no estructurados y dinámicos en los cuales los problemas de percepción y planificación automática son muy complejos. En muchos casos, el operador está físicamente separado del robot, existiendo un sistema de telecomunicaciones entre los dispositivos que utiliza directamente el operador y el sistema de control local del robot.

La intervención del operador puede producirse en muchas formas diferentes, desde la teleoperación directa de los actuadores de las articulaciones, hasta la simple especificación de movimientos, o incluso de tareas, que se realizan de forma automática.

Se entiende por *teleoperación* la extensión de capacidades sensoriales y destreza humanas a una localización remota. Se utilizan también los términos de *teleactuación* para referirse a los aspectos específicos de generación de órdenes a los actuadores y de *telesensorización* para la captación y visualización de información sensorial.

La *telerrobótica* puede considerarse como una forma evolucionada de teleoperación, caracterizada por un aumento de autonomía (capacidad de decisión y actuación) en el sistema remoto manteniendo una intervención significativa del operador huma-

no para supervisión o teleoperación directa. En telerrobótica la comunicación entre el operador humano y el computador de control se realiza a un nivel más elevado que en la teleoperación convencional (Vertut y Coiffet, 1985; Sheridan, 1989).

Por lo que respecta a las funciones específicas de control, en telerrobótica se emplean los términos de *control supervisado* ("supervisory"), *compartido* ("shared") e *intercambiado* o *negociado* ("traded"). En control supervisado el hombre dirige y monitoriza las actividades de un sistema semiautónomo de control. El control compartido consiste en una combinación de autonomía y telecontrol para controlar una función dada. El control intercambiado es una selección de autonomía o telecontrol para controlar una determinada función. El término *control de supervisión* tiene un sentido más amplio incluyendo métodos de simulación, planificación y ayuda a la toma de decisiones del operador empleando sistemas informáticos.

Otro concepto específico de la teleoperación de manipuladores es el de *sistemas bilaterales*, compuestos por un manipulador maestro y un manipulador esclavo que debe reproducir los movimientos que se generan moviendo el maestro. Normalmente, el *sistema de control bilateral* emplea realimentación a las articulaciones del manipulador maestro de los esfuerzos del esclavo, con el objetivo de conseguir un acoplamiento entre ambos manipuladores.

Existen diversos investigadores que han trabajado en técnicas de *cinetesia* y "percepción propia" con el objetivo de suministrar al teleoperador un sentido de movimiento y conciencia de miembros, diseñando mecanismos antropomorfos de manipulación y estructuras de esqueletos que se colocan sobre el brazo del operador.

El término *telepresencia* se refiere a una situación ideal en la que el operador se "siente" presente en la localización remota. En telepresencia es fundamental el empleo de mecanismos, tales como los comentados en el parrafo anterior, y sistemas de visualización que crean una "ilusión" de presencia al operador humano en la localización remota. El aspecto más específico de la telepresencia es la explotación de la interacción entre aspectos sensoriales y motores. Esta interacción se realiza mediante la integración de sensores y tecnologías de efectores.

Otro campo relacionado con la telepresencia, pero con una identidad propia fuera del marco de la robótica, es el de la *realidad virtual*. De forma general, puede considerarse que las tecnologías de realidad virtual permiten que el sistema informático y los procesos que controla sean transparentes al usuario.

## 13.2 TELEACTUACIÓN

Como se ha mencionado en el apartado anterior, existen diversos dispositivos para que el operador genere sus actuaciones de control. Es frecuente el empleo de los sistemas maestro-esclavo en los cuales el operador actúa moviendo el efecto final de un manipulador maestro cuyos movimientos deben ser reproducidos por el efecto final del manipulador esclavo. Se emplean manipuladores réplica del manipulador esclavo estableciéndose sistemas de control bilateral en los servos de las articulaciones correspondientes (Bejczy y Salisbury, 1983).

Se utilizan también manipuladores maestros diferentes del esclavo no sólo en su tamaño sino también en su configuración cinemática. El problema es que la diferencia

entre las cinemáticas debe ser compensada por el sistema informático de control que debe realizar las transformaciones correspondientes. Este problema se tratará de forma más detallada en un próximo apartado dedicado a sistemas bilaterales.

Asimismo, se emplean dispositivos de entrada tales como "joysticks" de tres grados de libertad o bolas de control con seis grados de libertad (tres traslaciones y tres rotaciones) que miden las fuerzas y pares generados por el operador en cada eje de un sistema con centro en el centro de la bola. En la figura 13.1 se muestra una de estas bolas de control.

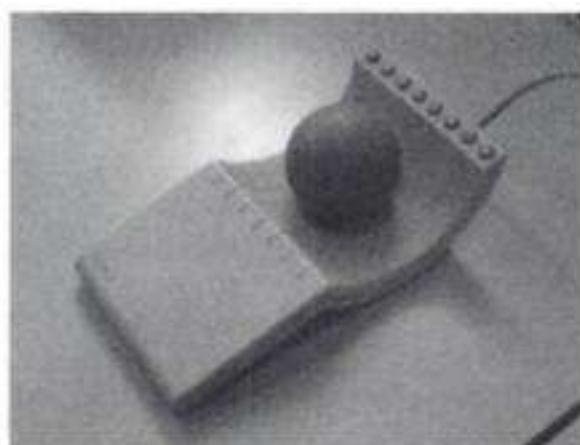


Figura 13.1: Bola de control.

Se han desarrollado también controladores de mano con seis grados de libertad y reflexión de los esfuerzos medidos en el manipulador remoto. En todos estos casos es necesario aplicar el modelo del manipulador para generar las consignas. Para resolver el problema de control planteado se aplican métodos tales como el control de impedancia, en el cual se emplean diferentes características de impedancia para diferentes tareas, y se ajustan las impedancias maestro/operador y tarea/esclavo.

Como se mencionó en el apartado de introducción, existen también sistemas de teleoperación avanzada que explotan conceptos de antropomorfismos para facilitar la teleoperación.

En el concepto de telepresencia se involucran no sólo los sensores sino sus acoplamientos con los actuadores que utiliza el operador humano para generar las órdenes, integrándose aspectos sensoriales y motores. Así, existen estructuras de brazos que se colocan sobre el operador, e incluso guantes que se utilizan para generar automáticamente las consignas necesarias para el manipulador esclavo y su efecto final.

En este punto conviene mencionar la utilización de dispositivos concebidos en el campo de la realidad virtual, tales como guantes, que consisten en una combinación de sistemas de seguimiento electromagnéticos y fibras ópticas para monitorizar movimientos de dedos y la posición y orientación de la mano en el espacio. Existen también dispositivos de teleoperación basados en el empleo de sensores de efecto Hall para registrar los movimientos de los dedos.

### 13.3 DISEÑO DE SISTEMAS DE CONTROL DE TELEOPERACIÓN

Los objetivos de diseño de un sistema de control de teleoperación son:

- 1) Hacer que el control manual del operador humano sea robusto ante retardos, saturación de los actuadores y otras no linealidades, e incluso ante los propios errores del operador.
- 2) Permitir elevadas prestaciones en la teleoperación (precisión, tiempo de teleoperación, comodidad) haciendo que los bucles de control tengan un comportamiento dinámico apropiado y reduciendo el trabajo del operador para lo cual se llega a realizar eventualmente de forma automática la acomodación y el control de esfuerzos en esquemas de control compartido. Nótese que la reducción del tiempo de teleoperación es importante ya que la ventana temporal de comunicaciones con la localización remota puede ser limitada, lo cual puede llegar a impedir que se complete la operación.

Un aspecto particularmente importante es la determinación de la información que se suministra al operador o, en otros términos, la definición de los bucles de teleoperación. En muchos sistemas, la información sensorial fundamental es la visual. Existen también numerosos sistemas que emplean información de proximetría.

Se han realizado experiencias que demuestran que la realimentación de esfuerzos al operador permite disminuir las fuerzas de contacto, el tiempo de teleoperación y el número de errores del operador (Hannaford, 1989). La reflexión de esfuerzos a las articulaciones del manipulador maestro permite acoplar el operador cinestésicamente al entorno. Sin embargo, cuando existen retardos en la comunicación, la realimentación de esfuerzos puede afectar de forma negativa a la teleoperación. En Ferrell (1966) se estudió por primera vez este problema, poniendo de manifiesto que retardos del orden de una décima de segundo eran suficientes para desestabilizar el sistema. En Vertut y otros (1981) se describen resultados de experimentos, poniendo de manifiesto que para que el sistema mantuviera la estabilidad sólo podía teleoperarse a velocidades muy bajas (sólo 10 cm/s). En Whitney (1985) se pone también de manifiesto que los retardos hacen impracticable la teleoperación con realimentación de esfuerzos de los manipuladores espaciales desde tierra.

### 13.4 SISTEMAS BILATERALES MAESTRO-ESCLAVO

Considérese el sistema de teleoperación compuesto por el manipulador remoto (manipulador esclavo) y su sistema de control, un manipulador maestro para la generación de movimientos, también con su propio sistema de control, el operador y el enlace de comunicación. Sean  $(\theta_s, \dot{\theta}_s)$ ,  $(\theta_m, \dot{\theta}_m)$  las variables articulares (posiciones y velocidades) en el manipulador esclavo y maestro, respectivamente. Suponiendo que  $\tau_{sl}(\theta_s, \dot{\theta}_s)$  son los pares generados por un controlador local en el manipulador esclavo, se aplica en este manipulador esclavo la ley de control:

$$\tau_s = \tau_{sl}(\theta_s, \dot{\theta}_s) + \tau_{sc}(e_s, \dot{e}_s) \quad (13.1)$$

siendo  $\tau_{sc}(e_s, e'_s)$  los pares de acoplamiento con el maestro, que son funciones de los errores en las variables articulares del manipulador esclavo  $\theta_s$  con respecto a los valores deseados  $\theta_{sd}$ :

$$e_s = \theta_{sd} - \theta_s \quad (13.2)$$

y de los correspondientes errores de velocidad con respecto a las velocidades deseadas:

$$e'_s = \dot{\theta}_{sd} - \dot{\theta}_s \quad (13.3)$$

En teleoperación resulta natural tomar los errores en el esclavo con respecto a las posiciones y velocidades del manipulador maestro:

$$\theta_{sd} = \theta_m, \theta'_{sd} = \theta'_m \quad (13.4)$$

En Hirzinger y otros (1991) se emplean estos errores para aplicar en el manipulador esclavo un par  $\tau_{sc}$  correspondiente al acoplamiento con el maestro. Para ello se utiliza el concepto de rigidez (Salisbury y Craig, 1980) que se introdujo en el capítulo 8, definiéndose:

$$\tau_{sc} = K_{s\theta} e_s + D e'_s \quad (13.5)$$

siendo  $K_{s\theta}$  una matriz de rigidez en el espacio de las articulaciones y  $D$  una matriz que juega el mismo papel que la fricción. Los errores  $e_s$  y  $e'_s$ , y las matrices de rigidez y de fricciones, se definen en el espacio de las articulaciones. Si se desean contemplar los errores  $\Delta x$  y esfuerzos  $\psi$  en el espacio cartesiano (capítulo 8), se tendrá:

$$\psi = K_{xx} \Delta x \quad (13.6)$$

siendo  $K_{xx}$  la matriz de rigidez en el espacio cartesiano. Teniendo en cuenta las relaciones puestas de manifiesto en el capítulo 8, los esfuerzos pueden aplicarse generando en las articulaciones unos pares:

$$J_s^T \psi = J_s^T K_{xx} \Delta x = J_s^T K_{xx} J_s e_s = K_{s\theta} e_s \quad (13.7)$$

siendo la relación entre las matrices de rigidez en el espacio de las articulaciones y en el espacio cartesiano  $K_{s\theta} = J_s^T K_{xx} J_s$ , y  $J_s$  el jacobiano del manipulador esclavo. Nótese que el lado derecho de (13.7) corresponde al primer sumando del lado derecho de (13.5), que representa al par de acoplamiento proporcional a los errores.

Por otra parte, interesa que el operador que acciona el manipulador maestro perciba el esfuerzo que está realizando el esclavo ya que, en caso contrario, podría generar pares que cuando se aplicaran en el esclavo podrían dañarlo. Para tener en cuenta este acoplamiento en el manipulador maestro se aplica la ley de control general:

$$\tau_m = \tau_{mi}(\theta_m, \theta'_m) + \tau_{mc}(e_s, e'_s) \quad (13.8)$$

donde el primer término del lado derecho corresponde a un controlador local y el segundo al acoplamiento con el manipulador esclavo. En el caso más simple se hace:

$$\tau_{mc} = K_{m\theta} e_s \quad (13.9)$$

siendo  $K_{m\theta}$  la matriz de rigidez del manipulador maestro en el espacio de las articulaciones.

El esquema de control que corresponde a la aplicación de (13.1) y (13.8)-(13.9) se ilustra en la figura 13.2.

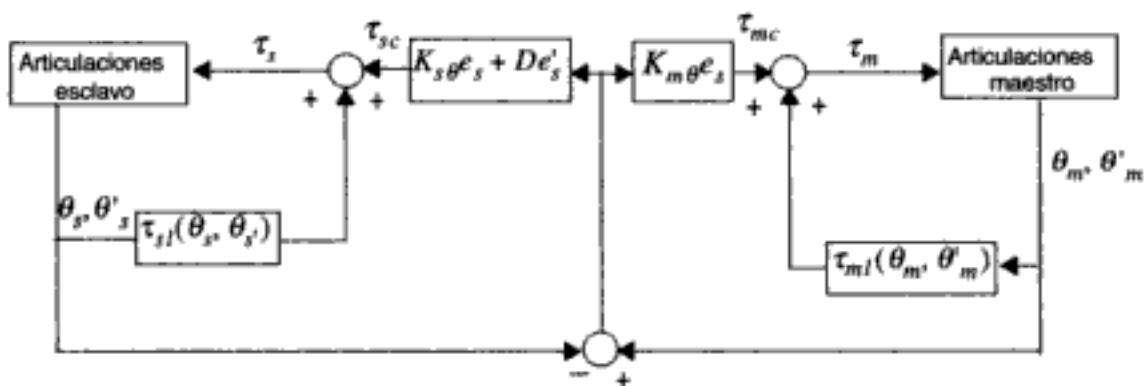


Figura 13.2: Sistema de control bilateral maestro-esclavo con realimentación de posición.

En el esquema de la figura 13.2, el operador percibe unos esfuerzos en el esclavo que se calculan a partir de los desplazamientos. Por tanto, parece evidente que el sistema puede mejorarse midiendo los esfuerzos  $\psi_s$  que realmente se producen en el esclavo y transmitiéndolos al maestro para generar un par de acoplamiento  $\tau_{mc}(\psi_s)$  apropiado. Son los denominados sistemas maestro-esclavo bilaterales con reflexión de esfuerzos, en los cuales la ley de control en el manipulador maestro viene dada por:

$$\tau_m = \tau_{ml}(\theta_m, \theta'_m) + \tau_{mc}(\psi_s) \quad (13.10)$$

siendo, como antes,  $\tau_{ml}(\theta_m, \theta'_m)$  el par que corresponde al controlador local del manipulador maestro. Nótese que:

$$\tau_{mc} = J_m^T \psi_s \quad (13.11)$$

siendo  $J_m$  el jacobiano del manipulador maestro. El esquema de control correspondiente se muestra en la figura 13.3.

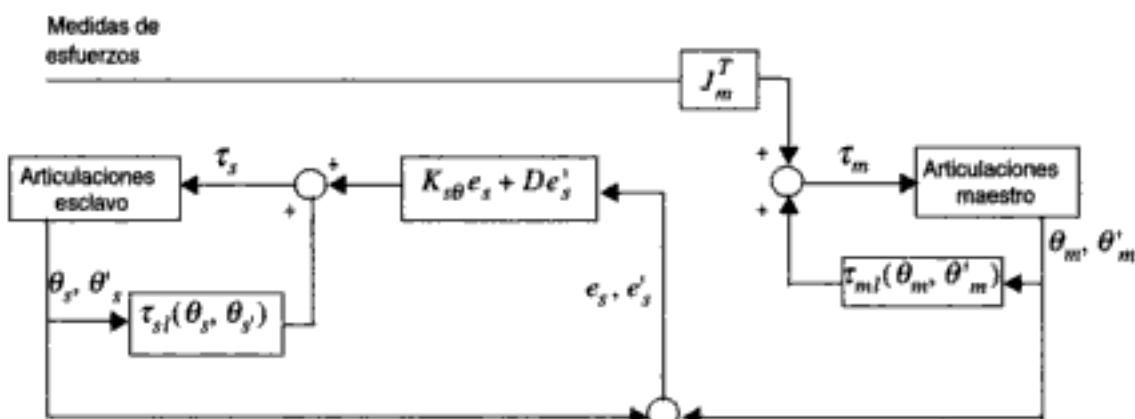


Figura 13.3: Sistema de control bilateral maestro-esclavo con realimentación de esfuerzos.

En los esquemas de control anteriores se ha supuesto que los manipuladores maestro y esclavo tienen la misma configuración cinemática. En la mayor parte de los casos esto no será así. Lo mismo ocurre cuando, en vez de un manipulador maestro, se emplean dispositivos de entrada tales como "joysticks" o bolas de control, como la que se muestra en la figura 13.1, u otros controladores de mano con seis grados de libertad. En este caso, en los bucles de control deben utilizarse las posiciones y orientaciones en el espacio cartesiano  $x_s$  y sus velocidades correspondientes  $x'_s$  comparándolas con las procedentes del maestro o dispositivo de entrada  $x_m, x'_m$ . Los errores cartesianos obtenidos deben transformarse a las articulaciones empleando la cinemática inversa, lo cual obviamente complica el problema. En la figura 13.4 se muestra un esquema de control bilateral con manipulador maestro de cinemática diferente.

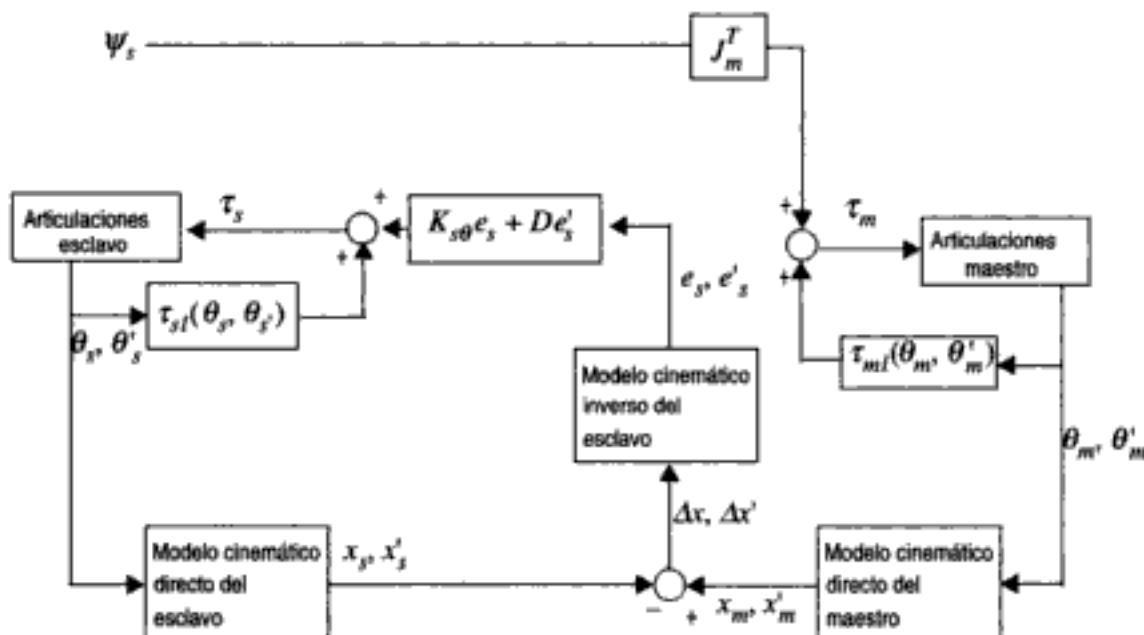


Figura 13.4: Sistema de control bilateral maestro-esclavo con realimentación de esfuerzos y cinemáticas diferentes en maestro y esclavo.

Los parámetros de los controladores en los esquemas anteriores deben elegirse cuidadosamente debido a la interacción entre los bucles de control automático y teleoperado. En efecto, cuando el manipulador esclavo realiza automáticamente el control de esfuerzos (Whitney, 1985), la realimentación directa al operador de las fuerzas medidas puede confundirlo en la evaluación de la orden de control de posición, ya que el operador percibe los esfuerzos sin comprender la respuesta del controlador automático de esfuerzos, con lo cual el error de posición puede incrementarse. En cualquier caso, los resultados de esta ley de control son críticamente dependientes del efecto de los retardos.

Para evitar los mencionados problemas, existen métodos que no utilizan la realimentación de esfuerzos suministrando otras indicaciones al operador, tales como tonos auditivos o vibraciones táctiles, consiguiendo de esta forma mejorar el rendimiento de la teleoperación.

Otra técnica consiste en el empleo de conceptos de control de impedancia según los cuales, cuando se aplica a un manipulador una fuerza externa  $\psi$  reacciona como una impedancia según la ecuación:

$$M_x \ddot{x} + D_x \dot{x} + K_x x = \psi \quad (13.12)$$

siendo  $M_x$ ,  $D_x$ ,  $K_x$  respectivamente matrices de masa, fricción y rigidez en el espacio cartesiano, e  $\Delta x$ ,  $\Delta x'$  los errores entre la posición y velocidad actuales y las deseadas en el espacio cartesano.

Los métodos de control de impedancia pueden aplicarse exista o no realimentación de esfuerzos medidos, utilizando en el segundo caso exclusivamente realimentación de posición al operador para generar una acomodación artificial.

Considérese el modelo dinámico del manipulador expresado en la forma:

$$\tau + J^T \psi = J^T [M_x \ddot{x} + \tau_x(\theta, \theta')] \quad (13.13)$$

En el lado izquierdo se suman los pares que se aplican a las articulaciones y los que resultan de aplicar esfuerzos externos. En el lado derecho se suman los pares provocados por las inercias en el espacio cartesano y una función  $\tau_x(\theta, \theta')$  que expresa los pares en el espacio cartesano que resultan de sumar los efectos centrífugos, de Coriolis, gravitatorios y de fricción. Nótese que en el lado derecho se premultiplica por la traspuesta del jacobiano para obtener los pares resultantes en el espacio articular. La expresión (13.13) es la misma que se consideraba en el capítulo 8 con el vector de fuerzas externas  $\psi$  cambiado de signo. Sustituyendo (13.12) en (13.13), se compensa el término de la inercia y resulta la ley de control:

$$\tau = J^T [\tau_x(\theta, \theta') - (D_x \dot{x} + K_x x)] \quad (13.14)$$

que, cuando se aplica al manipulador esclavo, puede interpretarse también como la suma de dos términos:  $\tau_x(\theta, \theta')$ , que puede asimilarse a una realimentación local, y  $D_x \dot{x} + K_x x$ , que correspondería al efecto del maestro en el cual se indican posiciones y velocidades deseadas que, cuando se comparan con las del manipulador esclavo, dan origen a los errores  $\Delta x$ ,  $\Delta x'$ . En la figura 13.5 se representa este esquema de con-

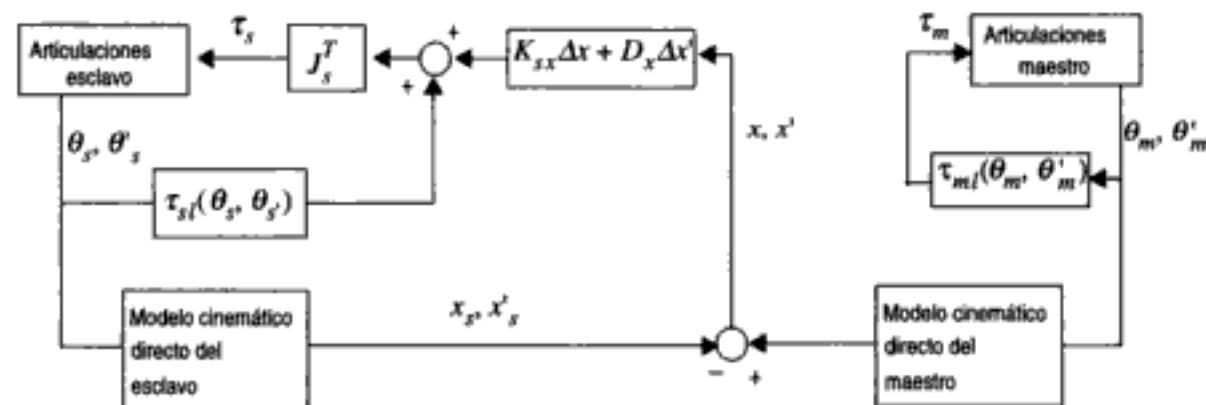


Figura 13.5: Sistema bilateral maestro-esclavo con control de impedancia.

trol (Hirzinger y otros, 1991, 1992). El operador observa en un monitor la posición del manipulador esclavo y genera, mediante el manipulador maestro, posiciones y velocidades deseadas que se comparan con las del manipulador esclavo.

El esquema anterior de control de impedancia se ha aplicado también cuando en el manipulador esclavo existe una adaptación de esfuerzos, ya sea mediante dispositivos dotados de una cierta acomodación, o bien mediante realimentación local de sensores de esfuerzo (capítulo 7) empleando estrategias locales de control de esfuerzos (sin realimentar al operador los esfuerzos medidos).

Los conceptos indicados en los párrafos precedentes se han aplicado también cuando, en vez de manipuladores maestros, se emplean dispositivos de actuación, tales como los mencionados en el apartado 13.2. En particular, se ha utilizado el dispositivo de bola de control que se muestra en la figura 13.1. Estas bolas de control son, en realidad, sensores de esfuerzos en el espacio cartesiano, empleándose los esfuerzos medidos para generar los incrementos de posiciones  $\Delta x$  en el espacio cartesiano, utilizando una cierta relación de rigidez artificial. En la práctica, estas órdenes admiten una doble interpretación. Cuando el manipulador esclavo se mueve libremente, se consideran como órdenes de traslación y/o rotación, mientras que cuando está en contacto con un objeto pueden interpretarse también como consignas tanto de posición como de esfuerzos.

Existen también métodos de impedancia generalizada (Lee y Lee, 1991, 1992) que tienen en cuenta la presencia de retardos e incluso errores en las actuaciones del operador.

Otro enfoque (Raju y otros, 1989) es el modelado del sistema de teleoperación en control bilateral como un circuito con dos puertas, con la interfase operador-maestro como la puerta maestra y la interfase tarea-esclavo como la puerta esclava. El análisis del sistema de control se realiza en términos de la energía intercambiada.

Asimismo, diversos investigadores (Anderson y Spong, 1989, 1992; Niemeyer y Slotine, 1991; Andriot y otros, 1991) aplican el concepto de pasividad de la teoría de control de sistemas no lineales para asegurar la estabilidad de sistemas de teleoperación con reflexión de fuerzas y retardos.

Finalmente, hay que mencionar los trabajos que consideran modelos de la respuesta del operador humano en el bucle de control de teleoperación (Chapell, 1989, Lee y Lee, 1992), utilizando los citados modelos para el diseño de leyes de control robustas.

En cualquier caso, el principal problema de los métodos mencionados en los párrafos precedentes es la necesidad de emplear un modelo dinámico suficientemente bueno. La bondad del modelo se convierte en crítica cuando se pretende compensar los retardos de transmisión. En efecto, el procedimiento más habitual para considerar el efecto de los retardos en métodos basados en modelos, consiste en incorporar los retardos en dichos modelos.

La técnica de teleoperación más simple es la denominada "mover y esperar" (Ferrall, 1966). En presencia de retardos, esta técnica obliga a que el operador desarrolle un considerable esfuerzo, el rendimiento es bajo y no garantiza una teleoperación segura, especialmente si los retardos son importantes. Por otra parte, cuando, debido a las comunicaciones, existen restricciones en el tiempo de teleoperación, esta estrategia puede hacer que la teleoperación sea impracticable.

Por ello, se han desarrollado otras técnicas cuya motivación básica, aunque no única, es la presencia de retardos en el bucle de teleoperación. Estas técnicas se estudian a continuación.

### 13.5 EMPLEO DE GRÁFICOS PREDICTIVOS

Se trata de que el operador perciba, mediante gráficos, el resultado de sus actuaciones antes de que se transmitan para su ejecución. De esta forma puede corregir dichas actuaciones en función de las diferencias existentes entre el movimiento que se predice y el que se pretende conseguir. Los métodos de control predictivo o, de forma más precisa, de *visualización predictiva* en teleoperación se deben a Noyes y Sheridan (1984). La motivación básica es la compensación de los retardos de comunicación entre el sistema local y el remoto.

Para su aplicación se superpone a la imagen obtenida de la señal de vídeo (con retraso), procedente del sistema remoto, un gráfico, típicamente en alambre, de la predicción de la posición del manipulador obtenida mediante un modelo en el computador de la estación de teleoperación. El esquema se muestra en la figura 13.6.

En el sistema de control predictivo, el operador puede actuar y visualizar de forma inmediata el efecto de su actuación en el simulador, sin tener que esperar el retorno de la señal de vídeo. La predicción permite que el operador introduzca un efecto anticipativo que compensa el retraso de la transmisión.

En la figura 13.7 (Noyes y Sheridan, 1984) se muestra una comparación experimental de tiempos de teleoperación con y sin simulación predictiva. Obsérvese que la reducción del tiempo de teleoperación es mayor cuando aumenta el retraso de transmisión. Para retardos de aproximadamente 2 s, el empleo de gráficos predictivos originaría una disminución de casi el 40% del tiempo de teleoperación.

El interés del control predictivo no se restringe a la compensación de retardos sino que, en general, puede permitir una mejora de prestaciones de la teleoperación. Con esta técnica, el operador puede reducir considerablemente el tiempo necesario para la teleoperación, eliminando la necesidad de la sincronización local-remoto, lo cual es particularmente ventajoso cuando los tiempos de ejecución de las órdenes en el sistema remoto son altos. El operador generaría una orden, visualizaría su efecto en el si-

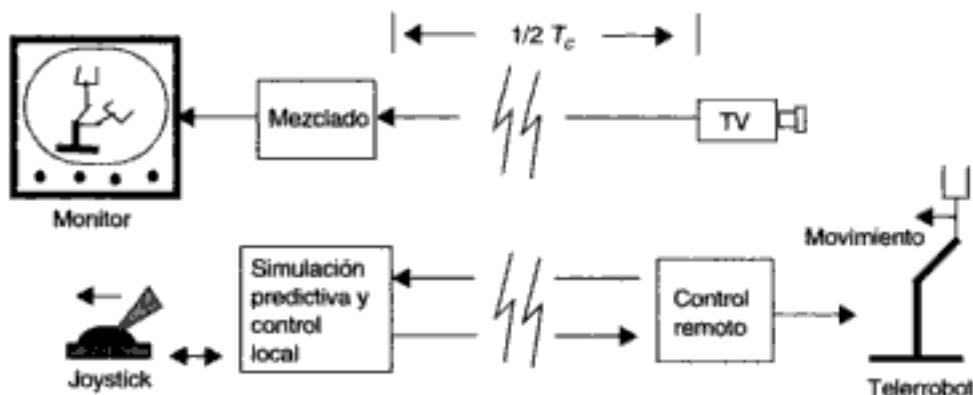
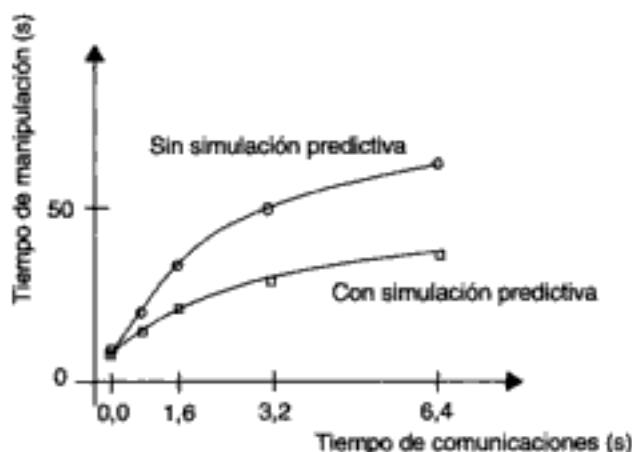


Figura 13.6: Simulación con gráficos predictivos.



**Figura 13.7: Comparación de tiempos de teleoperación.**

mulador y, en caso de que éste sea el apropiado, validaría la orden y pasaría a generar otra sin necesidad de esperar a que la orden se ejecutara realmente. Otra aplicación interesante es la prevención de posibles errores de operación que se detectarían mediante el simulador antes de su ejecución.

Existen otros métodos que generalizan el desacople de las actuaciones del operador con su ejecución en el sistema remoto para mejorar las prestaciones de la teleoperación (Conway y otros, 1987). En estos métodos de relajación de sincronía, cabe distinguir entre la relajación de sincronía temporal y espacial. De esta forma, se dispone de un "embrague" de tiempo para permitir desacoplo de sincronía temporal en planificación de caminos, y un embrague de posición para permitir simulación hacia adelante y ensayos de posicionamiento sin generar caminos. En la figura 13.8 se muestra un esquema de control en el que pueden introducirse ambos desacoplos.

El embrague de tiempo permite que el operador se desacople del tiempo real y manipule hacia adelante a partir de la imagen mostrada en un monitor de vídeo del manipulador real, con la superposición sobre esta imagen del gráfico en alambre del manipulador. El interés es evidente en movimientos lentos en los cuales el ahorro de tiempo de operación es significativo. El operador puede observar resultados y repetirlos a velocidad más lenta en maniobras difíciles. Cuando el sistema está embragado, la posición (o el cambio de posición) del "joystick" se muestrea y se utiliza directamente para controlar los movimientos en el modelo de simulación, teniendo en cuenta las restricciones reales de velocidades y aceleraciones del robot.

El "buffer" entre el simulador y el control directo del robot mantiene una cadena de incrementos de posiciones. De esta forma, cuando el simulador está embragado, el "buffer" presenta una cadena de puntos a una frecuencia fija y el robot simplemente va incrementando sus posiciones para seguir la secuencia.

Cuando se desembraza, la distancia entre posiciones del camino puede ser mayor que lo que el robot puede moverse en el tiempo en que se generan las posiciones por el operador. Se emplea entonces un interpolador para generar puntos intermedios.

En sistemas de control con realimentación de fuerzas o tacto al operador, el desembraque del tiempo debe desconectar también esta realimentación y sustituirla por la generada por el simulador.

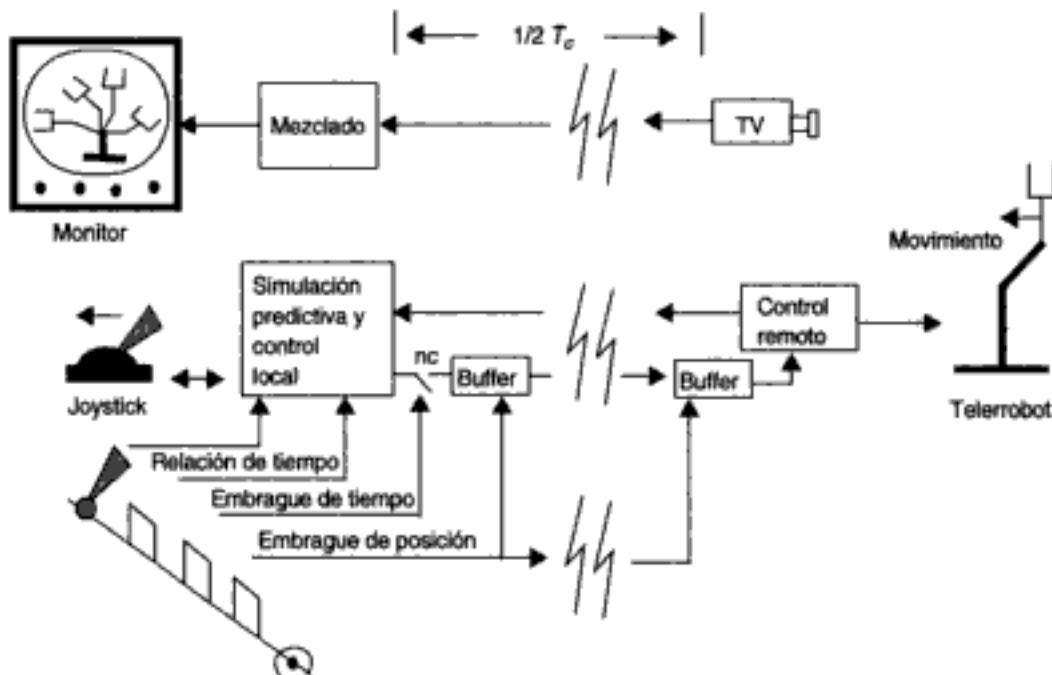


Figura 13.8: Desacoplamiento de posición y de tiempo.

El desembrague de la sincronía de posición es interesante cuando se desea experimentar una nueva maniobra o ensayar una maniobra compleja. Para ello, se lanza la simulación hacia adelante comprobando el efecto del movimiento generado, enviándose sólo cuando el resultado de la simulación es satisfactorio.

Tanto los métodos de gráficos predictivos como, en general, los de relajación de sincronía temporal o espacial, exigen que el modelo de predicción sea suficientemente fiable y eficiente para la simulación en tiempo real. Por otra parte, estos métodos no suministran al operador sentido de acoplamiento cinestésico con el entorno.

## 13.6 TELEPROGRAMACIÓN E INTERACCIÓN CON EL ENTORNO

En un sistema de teleprogramación (Fundá y Paul, 1991) las acciones del operador en un entorno simulado se trasladan a una secuencia de instrucciones simbólicas para el robot remoto, que se transmiten para su ejecución retrasada en el tiempo en el sistema de control remoto. Las instrucciones simbólicas se diseñan teniendo en cuenta las posibles discrepancias entre el entorno y el modelo disponible en la estación de teleoperación.

Cuando el operador especifica una tarea en el entorno virtual, el sistema de teleprogramación debe interpretar las acciones del operador y producir la secuencia de instrucciones simbólicas. Para ello, se analizan las trayectorias de posición y fuerza, suministradas por el operador con un dispositivo apropiado de entrada, y las diferencias de estado entre la simulación del manipulador y el entorno.

Para que los resultados no sean críticamente dependientes de la bondad del modelo, es posible emplear un sistema de monitorización en el controlador remoto, de tal forma que cuando se producen errores (no se alcanza condición esperada de final de movimiento, contacto con obstáculo, pares excesivos en articulaciones...) se informe a la estación de teleoperación. En este caso, la estación debe parar la simulación del modelo y arrancarla desde el estado comunicado por el sistema de control remoto.

Hirzinger y otros (1992), en el proyecto de telerobot espacial ROTEX, propusieron la integración de gráficos predictivos con información del sistema de percepción sensorial para tareas de ensamblado. Asimismo, propusieron la utilización de técnicas de aprendizaje a partir de información sensorial utilizando también simulación gráfica. El robot se guía gráficamente desde Tierra fuera de línea, almacenándose no sólo las posiciones y orientaciones en las que debe estar el efecto final, sino también los correspondientes patrones sensoriales (simulados gráficamente) para su utilización posterior en la fase de ejecución en línea. Mediante esta técnica, denominada "tele-sensor-programming", se suministra al robot información sensorial simulada sobre las posiciones relativas entre el efecto final y el entorno, con lo cual se trata de compensar las posibles imprecisiones en las posiciones absolutas del robot y el entorno, evitando los problemas que surgen cuando el modelo simulado no coincide con el real. Teniendo en cuenta la posibilidad de errores en la simulación de los sensores, el sistema de simulación no sólo recibe las órdenes del dispositivo de actuación del operador (bola de control en el ROTEX), sino también los datos sensoriales reales procedentes del robot y su entorno, los cuales se emplean para mejorar los modelos.

### 13.7 CONTROL DE SUPERVISIÓN

---

Dentro del control por supervisión se engloban un conjunto de métodos de simulación, planificación y ayuda a la toma de decisiones, soportados por sistemas informáticos de ayuda al operador en la teleoperación (Sheridan, 1989). Dentro de estas funciones cabe distinguir entre las siguientes:

- Funciones fuera de línea: Simulación interactiva en tiempo real del teleoperador con realimentación de esfuerzos y realimentación visual. Estas tareas son normalmente complejas y requieren importantes recursos computacionales.
- Funciones de simulación en línea: Se emplean para previsión de maniobras y asignación a control manual o autónomo.
- Funciones de simulación en línea para compensación de retardos temporales. Se trata de que el operador pueda prever el efecto de los retardos para compensarlos de forma oportuna.
- Funciones de medida, estimación y visualización de estado: Entre estas funciones se encuentran las dedicadas a ofrecer una visualización más comprensible para el operador del estado de la operación. Se emplean también métodos para combinar de forma apropiada medidas de diversos sensores, información de estados precedentes y utilización de bases de datos con medidas y parámetros de teleoperación.

- Funciones de procesamiento de órdenes del operador: Combinación de entradas analógicas y simbólicas según lenguajes específicos de control supervisado.
- Funciones de ajuste del sistema de control: Entre estas funciones se encuentran los ajustes de ganancias y otros parámetros del sistema de control automático, tales como relaciones de fuerzas. Otras funciones que pueden incluirse en este grupo son las de guiado de realimentación sensorial, tales como la orientación de cámara, ajuste del zoom, etc. que se consideran en un apartado posterior.
- Ayudas para detección de fallos, identificación y respuestas de emergencia: Se involucran funciones de prevención, detección de comienzo de fallo, identificación de fallos, respuestas automáticas de emergencia, y recomendaciones para restauración de condiciones normales de operación.
- Monitorización de rendimiento: Se trata de obtener medidas de rendimiento, velocidad, precisión, resolución, tamaño, etc. Asimismo, es interesante establecer procedimientos de comparación de rendimiento manual y automático. Estas medidas son de interés en sistemas de aprendizaje monitorizado, que pueden emplearse como transición a modos de control automático.

## 13.8 TELESENSORIZACIÓN

Se trata de extender la capacidad sensorial humana a una localidad remota empleando sensores y sistemas de presentación de la información.

En la telesensorización es necesario considerar tanto los sistemas sensoriales involucrados como la forma en que se presenta al operador la información.

Por lo que respecta a las funciones sensoriales, es necesario poner de manifiesto el importante papel que en la teleoperación juega la visión, a la que se dedica el próximo apartado de este capítulo. En cualquier caso, la teleoperación involucra otras funciones sensoriales que se mencionan a continuación.

Así, la información suministrada por los sensores de proximetría puede mostrarse al operador en un monitor de teleoperación. Es aconsejable que la presentación de esta información sea tanto numérica como analógica, empleando una representación gráfica a color en una ventana del terminal del monitor.

El operador puede percibir también la proximidad empleando el sistema de visión. Como se comentará posteriormente, la facilidad de percepción está relacionada con el diseño del sistema de visión para teleoperación. Por otra parte, el sistema de visión autónoma involucra funciones de procesamiento de imágenes mediante las cuales es posible obtener medidas de proximidad. La dificultad obvia es la complejidad del procesamiento necesario.

Asimismo, es interesante la transmisión y visualización de la información de los sensores de contacto en un monitor de teleoperación (ver, por ejemplo, Backes y otros, 1991). En numerosos trabajos de manipulación se recomienda reflejar los esfuerzos en los motores del manipulador maestro. En todo caso, cuando se disponga de sensores de esfuerzos, se considera interesante prever la transmisión de esta información y la visualización de los pares y fuerzas mediante técnicas similares a las comentadas para los sensores de proximetría.

Se emplean también distintas técnicas para simular la percepción de tacto incluyendo la visualización, para lo cual existen métodos ópticos. No obstante, la aportación del hombre es inferior a la que puede conseguirse con los sensores mencionados anteriormente debido, fundamentalmente, a la insuficiencia de la percepción de las presiones diferenciales, la cual puede ser realizada de forma más eficiente por un computador.

## 13.9 SISTEMAS DE VISIÓN EN TELEOPERACIÓN

---

En este apartado se utiliza fundamentalmente el informe de Brooks y otros (1991) y algunas experiencias realizadas en los proyectos Miniman (Gómez Elvira y Ollero, 1994; Ollero y otros, 1995) y Aurora (Mandow y otros, 1996).

### 13.9.1 Cámaras

En los primeros sistemas de teleoperación se emplearon cámaras de tubos de vacío. En la actualidad se utilizan cámaras CCD, consiguiéndose cada vez una mayor resolución en el detector (cámaras de varios miles de elementos en cada una de las dos dimensiones) lo que resulta difícil de aprovechar utilizando tecnologías convencionales de visualización, debido a las limitaciones de muestreo y transmisión (limitación en el número de líneas horizontales de televisión y número de imágenes por segundo). Las cámaras actuales admiten disparo electrónico y ganancia de señal controlada remota y electrónicamente. Existen también cámaras que permiten resolver problemas de deslumbramiento y agudeza visual del operador, tales como las que emplean tecnología de cristal líquido para suprimir rayos de luz directa, o las que permiten tomar una vista de muy alta resolución y vista periférica de baja resolución, lo cual hace posible conseguir alta resolución sin tener que aumentar el ancho de banda en zonas periféricas sin interés específico.

Otro aspecto que debe ser analizado es la necesidad o no de visión en color. Para ello es necesario estudiar de forma detallada la tarea ya que, en caso de no ser relevante para dicha tarea, cuando existen limitaciones en la transmisión digital de imágenes se recomienda la visión monocroma.

El diseño de un sistema de teleoperación hace necesario tener en cuenta las características de las lentes, incluyendo el zoom, apertura y campo de vista. Aunque existen sistemas de teleoperación que emplean distancia focal fija, en general, se considera conveniente disponer de *zoom*. Éste debe tener suficiente rango para suministrar una perspectiva global del área de trabajo a la menor distancia focal y, simultáneamente, suministrar suficientes detalles a la mayor distancia focal, realizando la transición lo suficientemente rápido para que el operador no olvide las características que tenía la escena antes de aplicar el zoom.

Con respecto a la *apertura*, hay que señalar que el control del iris de la cámara es interesante para conseguir imágenes con un número elevado de niveles de grises. Existen estudios que han puesto de manifiesto la posibilidad de teleoperar con sólo 16 niveles de grises, aunque, en general, las condiciones ideales son las de la visión humana directa que involucra, aproximadamente, 158. El empleo de autoiris es muy fre-

cuente en teleoperación en entornos con iluminación variable, con objeto de compensar dichas variaciones. En cualquier caso, el control de la iluminación del entorno de teleoperación es muy importante para la visualización, ya que los deslubramientos, por acción directa de la fuente de luz o debido a la reflexión de superficies pulimentadas, puede llegar a hacer muy difícil la teleoperación.

Un *campo de vista angular* amplio suministra perspectiva, lo cual es de gran interés para realizar movimientos gruesos, pero la resolución espacial resultante es menor, lo que dificulta, o incluso puede impedir, los movimientos finos.

Nótese que el comentario anterior involucra criterios de diseño conflictivos entre sí. En general, no es posible satisfacer con una única cámara todas las necesidades de la teleoperación. Así, se recurre a cámaras panorámicas para obtener una perspectiva global y cámaras para obtener visión de detalle, tales como las que se instalan sobre el efecto final de un manipulador. El número de cámaras viene también determinado por la complejidad del entorno y las tareas que se realicen (Miller y Mitchell, 1990; Spain, 1990). Para tareas de manipulación, los operadores prefieren disponer de una cámara con el eje en una elevación de aproximadamente 45 grados debajo de la horizontal. Por otra parte, en muchos casos, la percepción de profundidad y, en general, la visualización de la tarea en un espacio de tres dimensiones, es crítica. Así, el empleo de una cámara ortogonal al movimiento suministra indicaciones visuales precisas sobre la distancia entre dos objetos que se aproximan, lo que, obviamente, es fundamental en la realización de tareas de acoplamiento, agarre, etc.

Por último, en la ubicación y movimiento de las cámaras hay que tener en cuenta la posibilidad de confusión del operador cuando se emplean cámaras invertidas o de puntos de vista en sentido contrario, por problemas de correspondencia entre las acciones que genera debido a su percepción visual de orientaciones relativas entre objetos y las que debería generar por las orientaciones relativas reales.

Otro aspecto de interés en teleoperación es el *control de las cámaras*, tarea en la que los teleoperadores suelen emplear un tiempo significativo. Típicamente, se controlan dos ángulos (orientación y elevación), pero también se utilizan dispositivos especiales e incluso montajes sobre manipuladores controlándose seis grados de libertad. La técnica de control más usual es el empleo de "joysticks" u otros dispositivos de entrada en el panel de control. Existen también técnicas de control mediante voz. Su ventaja principal es que permite que el operador pueda manejar a la vez otro controlador de mano. El principal problema del control mediante voz es la naturaleza discreta de las órdenes, lo cual presenta dificultades para las funciones que requieren un control de tipo analógico. Por ello, los mejores resultados se obtienen para las tareas de selección. Existen también sistemas de seguimiento automático de objetivos ("autotracking") que emplean procesamiento de imágenes en tiempo real.

### 13.9.2 Visualizadores

Los visualizadores más utilizados en teleoperación son los tubos de rayos catódicos (CRT) debido al mayor rendimiento de visualización (resolución alta, brillo, escala de grises y rendimiento de color). Como se sabe, existen diversas tecnologías de CRT, dependiendo del tipo de cañón y la máscara. Otra alternativa es el empleo de visualizadores planos. Estos visualizadores tienen como ventajas el menor peso, volu-

men, consumo y mantenimiento. En la actualidad, los visualizadores planos más utilizados son los LCD que emplean tecnología de transistores de película fina, "thin-film transistor" (TFT), debido a su mejor tiempo de respuesta, capacidad de colores y brillo.

La situación de los visualizadores en zonas próximas a la línea de vista (entre 10 y 15 grados debajo de la horizontal) evita los movimientos de cabeza haciendo más rápida y segura la teleoperación. Las dimensiones del visualizador están relacionadas con la distancia a la que se sitúa el observador y las características del circuito cerrado de televisión (resolución, campo de vista).

Se emplean también visualizadores montados en casco que son interesantes cuando existen limitaciones de espacio o cuando se requiere un acoplamiento entre el seguimiento del movimiento de la cabeza y el control de las cámaras. Existen dos posibilidades dependiendo de que el vídeo se superponga al entorno del usuario, o reemplace totalmente el entorno real (diseño opaco), tal como se muestra en la figura 13.9. La primera tiene la ventaja de hacer que el operador continúe sintiéndose en un entorno que le es familiar, permitiéndole continuar monitorizando u operando diversos dispositivos de control situados en su entorno. La segunda es más apropiada para visualizaciones de anchura de campo, pero puede llegar a desorientar al operador. Con las tecnologías actuales existen todavía limitaciones en la resolución y campo de vista en los visualizadores montados en casco, por lo que la obtención en estos sistemas de una buena agudeza visual es difícil a un precio razonable.



Figura 13.9: Visualizador opaco.

Para terminar este apartado es necesario dedicar comentarios específicos a la *visualización estereoscópica*. Se trata de crear el sentido de profundidad presentando dos imágenes dispares. El cerebro humano funde las imágenes para crear la percepción de profundidad. La visualización estéreo suele requerir que el operador utilice dispositivos especiales, típicamente gafas, para la observación de la escena en el visualizador. Las técnicas básicas son la de paralelismo en el tiempo y la de multiplexación en el tiempo.

En la técnica de paralelismo en el tiempo, las imágenes del ojo izquierdo y derecho se observan de forma simultánea. Existen sistemas binoculares montados en casco que la utilizan empleando lentes y espejos para mostrar simultáneamente a los ojos dos imágenes separadas espacialmente. Otra posibilidad de paralelismo en el tiempo es la presentación de imágenes izquierda y derecha a diferentes longitudes de onda de la luz. Para observar las dos imágenes simultáneamente el operador debe utilizar gafas equipadas con filtros apropiados. La técnica de "anaglyph", basada en la separación de colores mostrando dos imágenes de diferentes colores y visualizando mediante filtros de color, es también de tipo paralelismo en el tiempo. En estos sistemas, la visualización mediante filtros polarizados para los canales izquierdo y derecho parece de mayor interés para teleoperación.

El estereó por multiplexación en el tiempo consiste en la visualización secuencial de las imágenes izquierda y derecha a una velocidad suficientemente rápida de comutación. La visualización puede conseguirse empleando gafas con lentes polarizadas a diferentes orientaciones. Un sistema de disparo se sincroniza con el generador del visualizador para controlar la polarización, de tal forma que cuando se muestra la imagen derecha sea sólo visible por el ojo derecho. Para el ciclo alternativo se invierte el disparo para permitir que el ojo izquierdo vea la imagen correspondiente.

Por último, cabe mencionar que existen sistemas de visualización que no requieren dispositivos adicionales para observar el monitor. A estos sistemas se les ha denominado autoestereoscópicos y entre ellos se incluyen las holografías, y las técnicas de visualización que muestran a los ojos izquierdo y derecho las columnas impares y pares de la misma imagen.

Existen diferentes estudios (Cole y Parker, 1989; Brooks y otros, 1991; Kim y otros, 1987a) que han puesto de manifiesto que la visualización estereó permite mejorar de forma importante la teleoperación especialmente cuando es necesario identificar objetos en entornos complejos y poco familiares. Cuando se emplea visión estereó la sensibilidad a la situación de las cámaras es menor. Se ha puesto también de manifiesto que las mejoras conseguidas con la visión estereó son más importantes cuando la visibilidad en la estación remota es más reducida y cuando no existe realimentación de esfuerzos en tareas de manipulación.

Sin embargo, cuando se necesita una elevada precisión en la estimación de distancias los resultados son mejores utilizando, en vez de visión estereó, visión mono con cámaras ortogonales. Asimismo, el aprendizaje del operador utilizando realimentación visual mono es más rápido que con estereó, aunque esta ventaja disminuye después de un cierto número de repeticiones del experimento. En cualquier caso, las principales dificultades de la visión estereó son los desajustes de convergencia y acomodación visual, que causan la fatiga del operador, los problemas de acomodación cuando el operador mueve la cabeza, los problemas de implantación deficiente del estereó, tales como la falta de correspondencia entre imágenes (objetos o cámaras en movimiento, falta de alineación entre ejes de cámaras) y las pérdidas de intensidad debidas a los polarizadores.

Existen también sistemas de visualización estereó, con imágenes generadas en la estación de teleoperación empleando para ello simulación y datos procedentes de sensores. Estos trabajos están relacionados con las técnicas de teleprogramación sensorial que se mencionaban anteriormente.

### 13.9.3 Visualización de imágenes virtuales

El empleo de modelos del entorno en tres dimensiones permite la generación de imágenes virtuales tal como se verían desde una cámara situada en una posición y orientación idóneas para la operación. La visualización puede realizarse en monitores fijos o empleando visualizadores montados en casco, tal como se indicó en el apartado anterior (Chung y otros, 1989).

La visualización puede mejorarse mediante la aplicación de software gráfico. Entre las funciones gráficas se encuentran las de destacar áreas de la imagen u obstáculos próximos que pueden producir una colisión, para lo cual pueden tenerse en cuenta no sólo las posiciones y orientaciones sino también la dirección del movimiento e incluso la velocidad del manipulador a partir de la información suministrada por sensores. Asimismo, puede ser conveniente marcar el área de trabajo actual del robot teniendo en cuenta las limitaciones en el movimiento de las articulaciones, así como las restricciones en el controlador de mano. Es posible también realizar transformaciones de sistemas de referencia que permitan visualizar el movimiento en la forma más apropiada para su control, evitando los problemas de falta de acoplamiento entre estímulos y respuestas que se mencionaron al tratar sobre la situación de las cámaras.

Existen también otras técnicas para suministrar información de profundidad (Kim y otros, 1987b; Wickens, 1990) tales como la representación de rejillas que se ilustra en la figura 13.10 (Brooks, 1991) sobre las que pueden indicarse puntos de control para el operador o símbolos que indiquen los objetivos.

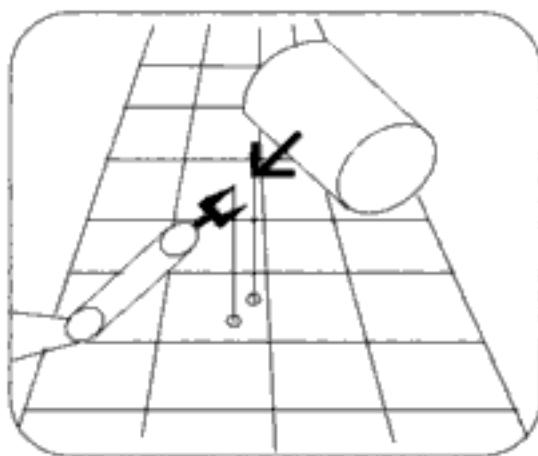


Figura 13.10: Empleo de rejillas y proyecciones.

Las imágenes virtuales pueden incorporar también información de sensores, tales como medidas de proximidad o de esfuerzos.

Asimismo, cabe pensar en el empleo conjunto, y eventual superposición, de imágenes virtuales generadas por una cámara situada en la misma posición y orientación que la cámara real con las imágenes procedentes de esta cámara real, con objeto de corregir las diferencias entre la posición prevista para el manipulador y la posición

real (diferencias entre donde debe estar y donde está realmente). Es posible también simplificar el problema generando únicamente representaciones virtuales simples o incluso rasgos característicos (ejes, marcas) que puedan ser comparados por el operador con los existentes en la imagen real. Estas técnicas, así como la incorporación de información sensorial mencionada en los párrafos precedentes, están relacionadas con los métodos de teleprogramación sensorial (Hirzinger y otros, 1991) que se mencionaban anteriormente. Otro punto relacionado con la visualización es el empleo de *gráficos predictivos* (temporales y espaciales) que también se discutieron en este capítulo. En este caso, es necesario realizar funciones gráficas que permitan superponer resultados de un modelo de simulación (típicamente modelos de alambre) con imágenes reales o gráficos obtenidos a partir de valores de las variables articulares.

#### 13.9.4 Transmisión de imágenes para teleoperación

La calidad de visualización de los objetos en movimiento es muy dependiente de la frecuencia a la que se muestren las imágenes, lo cual depende no sólo del dispositivo de visualización sino de las comunicaciones con la localización remota en la que se sitúan las cámaras. Así, la transmisión de imágenes es, en muchos casos, la mayor limitación para la visión estéreo y, en general, para el empleo simultáneo de múltiples cámaras (panorámicas, de visión de detalle, ortogonales, etc.) tal como se comentaba anteriormente.

En general, para la teleoperación ideal no debería bajarse de forma importante de los 25-30 Hz. Se aconseja el empleo de técnicas de compresión de imágenes para casos en los que existen importantes limitaciones en el ancho de banda del canal de comunicaciones.

Se han realizado experimentos de teleoperación simulando restricciones en las comunicaciones que han puesto de manifiesto que las velocidades de transmisión inferiores a 5,6 cuadros por segundo degradan considerablemente el rendimiento e incrementan los errores, ya que a esta velocidad el operador debe emplear una estrategia de mover y esperar. Un factor de reducción de 4 pasando de 28 a 7 cuadros por segundo originaba una disminución del 20% del rendimiento de la teleoperación, mientras que una reducción a la mitad (14 cuadros por segundo) permitía mantener prácticamente las condiciones de la teleoperación (Ranadive, 1981).

Sin embargo, existen también otros efectos laterales de la disminución de la velocidad de transmisión, tales como la fatiga del operador y la percepción de la velocidad del movimiento, los cuales dan origen en la práctica a una disminución del rendimiento de la teleoperación. Asimismo, a velocidades de transmisión menores, el efecto del ruido es mayor.

### 13.10 CONCLUSIONES

Existe un buen número de aplicaciones en robótica que sólo pueden resolverse mediante la teleoperación de un operador. En este capítulo se han presentado de forma resumida conceptos y tecnologías de teleoperación y telerrobótica, que se considera como una forma evolucionada de teleoperación caracterizada por un aumento de la

autonomía en el sistema remoto (robot) manteniendo una intervención significativa del operador humano.

En la actualidad se utilizan numerosas tecnologías para la teleactuación, desde los tradicionales "joysticks" hasta sofisticados dispositivos que explotan conceptos antropomórficos para facilitar la teleoperación. Asimismo, las posibilidades en visualización son muy diversas, empleándose monitores de diversas tecnologías y visualizadores de realidad virtual que, en combinación con los dispositivos de teleactuación, permiten la aplicación de técnicas de telepresencia.

Los sistemas de control de teleoperación utilizan modelos y gráficos predictivos que permiten paliar el efecto de los retardos de comunicación. Existen también sistemas maestro-esclavo para la telemanipulación, con posibilidad de cinemáticas diferentes en el maestro y esclavo.

La aplicación de tecnologías de realidad virtual y la compensación de retardos para la realización de una teleoperación eficiente, siguen siendo temas de investigación y desarrollo que se incorporarán en los próximos años a los sistemas comerciales de teleoperación. Estos sistemas explotarán los progresos en las tecnologías de las comunicaciones y la robótica para realizar una gran cantidad de nuevas aplicaciones en campos tales como la cirugía, la intervención en los fondos marinos, la minería, la construcción y la exploración espacial.

### 13.11 REFERENCIAS

---

- Anderson R.J. y M.W. Spong, 1989. "Bilateral control of teleoperator with time delay". IEEE Trans. on Automatic Control. Vol. 34, núm. 5, pp. 494-501.
- Anderson R.J. y M.W. Spong, 1992. "Asymptotic Stability for force reflecting teleoperators with time delay". International Journal of Robotic Research. Vol. 11, núm. 2, pp. 135-149.
- Andriot C., R. Fournier y J. Vuillemey, 1991. "On the bilateral control of teleoperators with flexible joints and time delay by the passive approach". Proc. of the fifth International Conference on Advanced Robotics. Pisa (Italia), pp. 231-236.
- Backes P.G., K.S. Thomas, S. Lee y S. Hayati, 1991. "A local-remote telerobot system for time delayed traded and shared control". Proc. of the fifth International Conference on Advanced Robotics. Pisa (Italia), pp. 243-248.
- Bejczy A. y M.W. Salisbury, 1983. "Controlling remote manipulators through kinesthetic coupling". Comput. Mech. Engineer, 2, pp. 48-60.
- Brooks T., I. Ince y G. Lee, 1991. "Vision issues for teleoperation assembly and servicing. A survey of visual requirements for teleoperation by the FTS". Rep. núm. STX/ROB/91-01. STX Robotics, Maryland (USA).
- Chapel J.J., 1989. "Performance limitations of bilateral force reflection imposed by operator dynamic characteristics". NASA Conf. on Space Telerobotics.
- Chung J. y otros, 1989. "Exploring virtual worlds with head mounted displays". SPIE. Vol. 1083. "Three dimensional visualization and display technologies".
- Cole R. y D. Parker, 1989. "Stereo TV improves manipulator performance". SPIE. Vol. 1083. "Three-dimensional visualization and display technologies".
- Conway L., R. Volz y M. Walker, 1987. "Tele-autonomous systems: methods and architectures for intermingling autonomous and telerobotic technology". Proc. IEEE Conference on Robotics and Automation.
- Ferrell W.R., 1966. "Delayed force feedback". IEEE Trans. Human Factors. Vol. 8, pp. 449-455.

- Funda J. y R.P. Paul, 1991. "Efficient control of a robotic system for time-delayed environments". Proc. of the fifth International Conference on Advanced Robotics. Pisa (Italia), pp. 219-224.
- Gómez Elvira J. y A. Ollero, 1994. "MINIMAN Project. A Space Telerobotic Demonstrator". Proceedings of the IARP, Second Workshop on Robotics in Space. Montreal, Canada. Julio, 1994.
- Hannaford B., 1989. "A design framework for teleoperators with kinesthetic feedback". IEEE Trans. on Robotics and Automation. Vol. 5, núm. 4, pp. 426-434.
- Hirzinger G., J. Heindl y K. Landzettel, 1991. "Control Structures in sensor-based telerobotic-systems". Proc. of the fifth International Conference on Advanced Robotics. Pisa (Italia), pp. 267-276.
- Hirzinger G. y otros, 1992. "The sensory and telerobotic aspects of the space robot technology experiment ROTEX". DLR Report.
- Kim W. y otros, 1987a. "Quantitative evaluation of perspective and stereoscopic displays in three axis manual tracking tasks". IEEE Trans. on System, Man and Cybernetics. Vol. SMC-1.
- Kim W. y otros, 1987b. "Visual enhancements in pick-and-place tasks: human operators controlling a simulated cylindrical manipulator". IEEE J. of Robotics and Automation, RA-3.
- Lee S. y H.S. Lee, 1991. "A Kinesthetically coupled teleoperation: Its modeling and control". Proc. of the fifth International Conference on Advanced Robotics. Pisa (Italia), pp. 255-260.
- Lee S. y H.S. Lee, 1992. "An advanced teleoperator control systems: design and evaluation". Proc. IEEE Robotics and Automation Conference. Niza (Francia), pp. 859-864.
- Mandow A., J. Gómez de Gabriel, J.L. Martínez, V.F. Muñoz, A. Ollero y A. García-Cerezo, 1996. "The autonomous mobile robot AURORA for greenhouse operation". IEEE Robotics and Automation Magazine. Vol. 3, núm. 4, pp. 18-28.
- Miller D. y P. Mitchell, 1990. "2D/3D comparison using commercial vision products". SPIE Stereoscopic Displays and Applications.
- Niemeyer G. y J.J. E. Slotine, 1991. "Transient shaping in force reflecting teleoperation". Proc. of the fifth International Conference on Advanced Robotics. Pisa (Italia), pp. 261-266.
- Noyes M. y T.B. Sheridan, 1984. "A novel predictor for telemanipulation through a time delay". Proc. Annual Conference on Manual Control. NASA Ames Research Center, CA.
- Ollero A., J. Gómez de Gabriel, E. Lozano, F. Cuesta y J. Gómez Elvira, 1995. "MINIMAN Telerobotic Station". Proceedings of the International Conference on Advanced Robotics. ICAR'95. Vol. II, pp. 591-599, Sant Feliu de Guixols, Cataluña, España.
- Raju G.J., G.C. Verghese y T.B. Sheridan, 1989. "Design issues in 2-port network models of bilateral remote manipulation". Proc. IEEE Conference on Robotics and Automation, pp. 1316-1321.
- Ranadive V., 1981. "Video framerate, resolution and tradeoffs under limited bandwidth for underwater telemanipulator control". 17th Annual Conference on Manual Control, JPL.
- Sheridan T., 1989. "Telerobotics". Automatica. Vol. 25, núm. 4, pp. 487-506.
- Spain E., 1990. "Stereo advantage for a peg-in-hole task using a force-feedback manipulator". Stereoscopic Displays and Applications, SPIE. Vol. 1256.
- Vertut J., R. Fournier y B. Espiau, 1981. "Short transmission delay on a force reflective bilateral manipulator". 4th Robotics-Man-Systems, Varsovia (Polonia), pp. 269-295.
- Vertut J. y P Coiffet, 1985. "Teleoperations and Robotics". Vol. 1 y 2. Kogan Page, Prentice Hall.
- Whitney D.E., 1985. "Historical perspective and state of the art in robot force control". Proc. IEEE Conference on Robotics and Automation, pp. 262-268.
- Wickens C., 1990. "Three-dimensional stereoscopic display implementation: guidelines derived from human visual capabilities". SPIE. Vol. 1256. Stereoscopic display and applications.

# Introducción a la herramienta HEMERO

---

En este apéndice se resume la herramienta MATLAB-Simulink que se ha empleado a lo largo de todo el texto para el análisis y simulación de los ejemplos. Esta herramienta, a la que se ha denominado HEMERO (HErramienta Matlab-simulink para la Enseñanza de la RObótica), puede ser empleada tanto para los manipuladores robóticos como para los robots móviles y está compuesta por funciones MATLAB y por bloques de Simulink que se han definido para facilitar la simulaciones de los modelos cinemáticos y dinámicos. En este apéndice se introduce la herramienta sin enumerar y describir todas las funciones MATLAB y bloques de Simulink existentes. El lector debe utilizar el CD-ROM que acompaña al libro para encontrar una descripción detallada. Se ha preparado también un sitio Web (<http://www.esi.us/hetero>) que podrá utilizarse para el uso de parte de la funcionalidad de la herramienta a través de Internet, así como para obtener versiones actualizadas.

## A.1 INTRODUCCIÓN

---

En la actualidad existen numerosos programas que pueden utilizarse para el diseño de robots. Algunos son programas generales de análisis y diseño de mecanismos articulados que también pueden aplicarse para el diseño de robots manipuladores o de sistemas de locomoción de robots móviles, permitiendo simular sus comportamientos cinemáticos y dinámicos. En otros, el énfasis se pone en las interfaces gráficas consiguiéndose simulaciones realistas del robot y su entorno.

El interés de MATLAB para el tratamiento matricial de los problemas involucrados en la cinemática y dinámica de los robots es evidente y ha sido puesto de manifiesto por numerosos autores. MATLAB permite una escritura casi directa y de fácil comprensión de las expresiones matriciales involucradas en los modelos cinemáticos y dinámicos de los robots. La herramienta MATLAB más conocida en robótica es posiblemente la de Corke (1996). Sin embargo, el interés práctico de esta herramienta se

restringe fundamentalmente al análisis cinemático y dinámico de los manipuladores robóticos, presentando diversos inconvenientes para su uso generalizado en el control, ya que solo puede utilizarse directamente la técnica del par computado, necesitándose un esfuerzo significativo de los usuarios para la programación de otras estrategias de control, así como para la simulación dinámica de los manipuladores con el sistema de control. Asimismo, en esta herramienta, tampoco se consideran robots móviles.

Por otra parte, existen también librerías de Simulink que permiten simular estrategias de control de robots manipuladores, tales como la que se presenta en Valera y otros (1999). Esta herramienta tiene interés para la simulación de una determinada estrategia de control de la que se dispone de su expresión simbólica para un determinado manipulador. Asimismo, las matrices de la dinámica han de ser calculadas previamente por el usuario.

En este apéndice se presenta la herramienta denominada HEMERO (HErramienta Matlab-simulink para el Estudio de manipuladores y RObots móviles, que se describe de forma detallada en el CD-ROM que acompaña a este texto (Maza y Ollero, 2001). En esta herramienta informática se ha optado por combinar MATLAB y Simulink con objeto de facilitar el estudio de la cinemática y dinámica de robots manipuladores y móviles, así como para la simulación de las estrategias de control. En Maza y Ollero (2000) se presenta una primera versión de la herramienta.

En el próximo apartado se ponen de manifiesto algunas características generales sobre la utilización de la herramienta para el análisis cinemático de los robots. Se presenta como ejemplo el Puma 560, y se consideran también los modelos cinemáticos de distintas configuraciones de robots móviles. En el tercer apartado se estudia el modelo dinámico, presentándose también como ejemplo el Puma 560. El cuarto apartado está dedicado al control, incluyéndose la aplicación del método del par computado y de un método de control con aprendizaje al Puma 560. Asimismo, en este cuarto apartado, se considera el control de un robot móvil con locomoción tipo triciclo con un manipulador Puma 560 sobre él, aplicándose el método de la persecución pura para el control de la plataforma móvil. En el quinto apartado se mencionan los métodos de generación de trayectorias disponibles en la herramienta. El sexto trata sobre la interfase gráfica y el séptimo se dedica a las conclusiones.

## A.2 CINEMÁTICA

HEMERO dispone de un conjunto de funciones MATLAB que facilitan la definición cinemática de los robots manipuladores, así como representaciones gráficas simples que permiten la visualización esquemática de los manipuladores y sistemas de referencia involucrados.

Las posiciones y orientaciones se definen empleando vectores de posición, matrices de rotación, y matrices de transformación homogéneas tal como se introduce en el capítulo 3. Se ha decidido utilizar los vectores de posición aumentados para obtener expresiones coherentes con las matrices de transformación correspondientes a las articulaciones. Así, la posición de un punto en el sistema de referencia {A} se expresa mediante el vector extendido de cuatro componentes  $\text{PenA} = [x \ y \ z \ 1]$ , cuyas tres primeras componentes son las coordenadas del punto en {A}.

Las transformaciones entre sistemas de referencia se expresan mediante matrices  $4 \times 4$ . En la transformación que define el sistema de referencia  $\{B\}$  con respecto al  $\{A\}$ , la submatriz  $3 \times 3$  superior izquierda es la correspondiente matriz de rotación (las columnas son los vectores unitarios del sistema  $\{B\}$  expresado en el  $\{A\}$ ), y los tres primeros elementos de la última columna son las coordenadas del origen del sistema de referencia  $\{B\}$  en el  $\{A\}$ , tal como se indica en el capítulo 3. Dada una matriz  $T$ , se comprueba si es una matriz de transformación mediante la función `ishomog(T)`.

Existe una función `T = rotvec(v, theta)` para obtener la transformación resultante de la rotación de un ángulo `theta` en torno a un vector `v`, y funciones específicas para realizar rotaciones de un ángulo `theta` alrededor de los ejes  $\hat{X}$ ,  $\hat{Y}$  y  $\hat{Z}$ : `rotx(theta)`, `roty(theta)`, `rotz(theta)`.

Análogamente, la función `T = transl(x, y, z)` devuelve una matriz de transformación  $T$  que representa una traslación dada por los tres escalares `x, y, z`; y la `T = transl(v)` devuelve la matriz de transformación resultado de la traslación por el vector `v`.

El sistema de referencia con una posición y una orientación dadas se representa mediante la función `frame(TT, color, tam, opt)`, siendo `TT` la matriz de transformación, `color` especifica el color en que se representará gráficamente el sistema de referencia, y `tam` indica el tamaño deseado para las flechas que representan el sistema de coordenadas. Con el parámetro `opt` se especifica si se desea mantener sin borrar los sistemas de referencia, con lo cual es posible representar o no todos los sistemas a lo largo de una trayectoria de transformaciones. La transformación inversa de una dada se obtiene mediante la función `trinv`.

Conociendo los ángulos de Euler Z-Y-Z, o los ángulos RPY, es posible crear la matriz de transformación homogénea correspondiente mediante las funciones `T = eul2tr(alfa, beta, gamma)`, o `T = rpy2tr(r,p,y)`, respectivamente.

Para tratar los modelos cinemáticos de los robots manipuladores se emplea fundamentalmente MATLAB. En esta herramienta se adopta la notación de Craig (1989). Las funciones MATLAB emplean como argumento una matriz con los parámetros de Denavit-Hartenberg (Hartenberg y Denavit, 1955) según la mencionada notación de Craig. Cada fila  $i$  de esta matriz tiene cinco elementos:  $\alpha_{i-1}$ ,  $a_{i-1}$ ,  $\theta_i$ ,  $d_i$ , y otro parámetro adicional  $\sigma_i$  que indica si la articulación es de rotación (valor 0) o prismática (valor 1) tal como se indica en el capítulo 4.

Como ejemplo se presentan las líneas de MATLAB que permiten definir la matriz de parámetros de Denavit-Hartenberg del robot manipulador PUMA 560:

```

dhp560 = [ 0 0 0 0 0;
            -pi/2 0 0 0.2435 0;
            0 0.4318 0 -0.0934 0;
            pi/2 -0.0203 0 0.4331 0;
            -pi/2 0 0 0 0;
            pi/2 0 0 0 0];
syms t1 t2 t3 t4 t5 t6 real; % Se definen simbólicamente las variables articulares.
q = [ t1 t2 t3 t4 t5 t6 ]; % Se define el vector de variables articulares.
T = simple( fkine( dhp560,q ) ); % Se calcula el modelo directo.

```

La función `fkin` permite obtener las expresiones simbólicas que proporcionan el modelo directo. Nótese como en la última línea se ha empleado también la función `simple`, que tiene como objetivo simplificar la expresión simbólica resultante de aplicar `fkin`.

Es posible, también, obtener el modelo directo multiplicando las transformaciones asociadas a cada uno de los enlaces del manipulador. Dichas transformaciones se calculan mediante la función `linktrans` cuyo primer argumento es ahora un vector fila con los parámetros de Denavit-Hartenberg del enlace en estudio y cuyo segundo argumento es el valor de la variable articular de dicho enlace. En el Puma 560 se emplearían las líneas de MATLAB:

```
T1 = linktrans (dhp560(:,1),t1);
T2 = linktrans (dhp560(:,2),t2);
T3 = linktrans (dhp560(:,3),t3);
T4 = linktrans (dhp560(:,4),t4);
T5 = linktrans (dhp560(:,5),t5);
T6 = linktrans (dhp560(:,6),t6);
T= T1*T2*T3*T4*T5*T6
```

donde  $T_i$  representa  $T_{i-1}^{-1}T_i$ .

Las expresiones MATLAB se emplean también para implementar las expresiones simbólicas que resuelven el modelo inverso y representar los resultados. Alternativamente, es posible también emplear la función `ikin` que permite resolver el modelo inverso mediante un método numérico. Así, por ejemplo, si se pretende que un manipulador Puma 560 siga una trayectoria rectilínea con una orientación constante, podrían emplearse las siguientes líneas de MATLAB:

```
x = [ 0.4: 0.001: 0.45 ];
y = 0.1*ones (1, length(x));
z = y; % Creación de la trayectoria de transformaciones
correspondiente

for k=1:length(x)
    TG (:, :, k) = [1 0 0 x(k);
                    0 1 0 y(k);
                    0 0 1 z(k);
                    0 0 0 1];
end
stol = 0.1; ilimit = 1000; % Se establecen la tolerancia y el número
máximo de iteraciones.

dhp560 = [ 0 0 0 0 0;
            -pi/2 0 0 0.2435 0;
            0 0.4318 0 -0.0934 0;
```

```

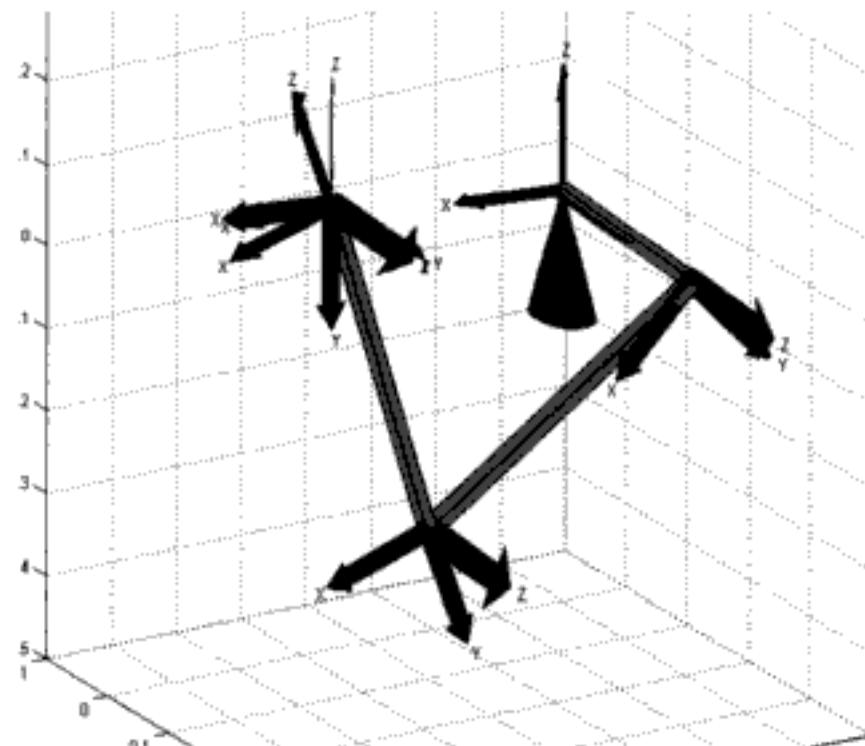
pi/2 -0.0203 0 0.4331 0;
-pi/2 0 0 0 0;
pi/2 0 0 0 0];
q = ikine (dhp560, stol, ilimit, TG);

```

Como resultado se obtiene una matriz  $q$  que contiene en cada fila los valores de las variables articulares correspondientes a cada punto de la trayectoria. Para representar el esquema del brazo siguiendo esa trayectoria se puede escribir:

```
plotbot (dhp560, q, 'fw')
```

El resultado para el último punto de la trayectoria se muestra en la figura A.1. En dicha figura se representan los sistemas de referencia asociados a las seis articulaciones del robot Puma 560. Obsérvese que los sistemas de las tres últimas articulaciones tienen un origen común.



**Figura A.1:** Representación esquemática del brazo con los cuadros de referencia correspondientes. El cono oscuro representa la base del manipulador.

Por lo que respecta al jacobiano, se dispone de las funciones `jacobn(dh, q)` y `jacob0(dh, q)` que permiten obtener, mediante el método de Paul (1981), la matriz del jacobiano con respecto a la articulación  $n$  y a la cero respectivamente. Así, por ejemplo, puede escribirse:

`J = simple(jacobn(dh,q))`

Alternativamente, es posible emplear el método de propagación de velocidades del capítulo 4 mediante la función `velprop(dh,q,qd,v0,w0)` que propaga las velocidades lineales y angulares desde la articulación 0 a la  $n$ .

Dado un vector de movimiento diferencial  $D = [d_x \ d_y \ d_z \ \delta_x \ \delta_y \ \delta_z]^T$  (Paul, 1981), la correspondiente transformación homogénea se obtiene mediante la función: `T=diff2tr(D)`. Asimismo, la función `D=tr2diff(T1,T2)`, calcula la diferencia de dos transformaciones y crea la correspondiente transformación homogénea.

Para el estudio de la cinemática de los robots móviles se emplea fundamentalmente Simulink. De esta forma, conociendo las velocidades lineales y angulares que se suministran como entradas a la plataforma móvil, es posible estudiar su evolución en posición y orientación. Para ello se han desarrollado bloques Simulink que corresponden a los modelos directos e inversos de diversas configuraciones del sistema de locomoción, incluyendo todas las que se presentan en el capítulo 4. En las capturas de la figura A.2 se muestran las entradas y salidas de algunos de estos bloques. Cada bloque tiene un cuadro de diálogo que permite introducir los parámetros requeridos.

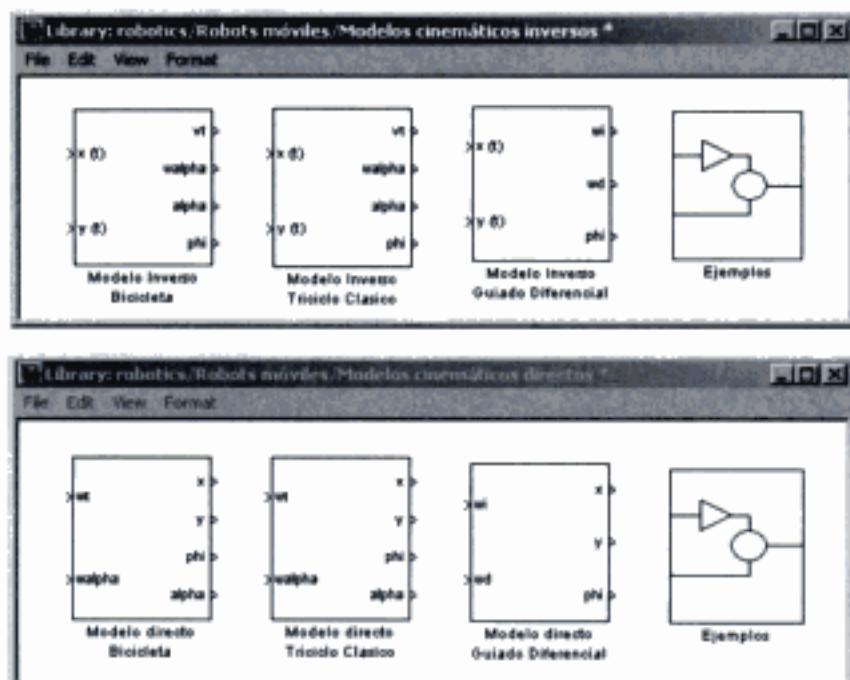


Figura A.2: Modelos cinemáticos directos e inversos de diversos robots móviles.

Obsérvese que se han incluido tanto bloques de modelos directos, como inversos. En los primeros las entradas son las variables de control y las salidas la posición y la orientación en el plano; en los bloques de modelos inversos las entradas corresponden a la trayectoria en el plano X-Y y las salidas son las variables de control. Así, por ejemplo, en el capítulo 4 se mostraba el diagrama detallado del bloque del modelo inverso del triciclo convencional que se indica en la figura A.2.

Obsérvese también que el empleo de los modelos directos, conocidas las entradas de control, permite realizar la simulación del robot móvil estimando su posición y orientación tal como se pone de manifiesto en el último ejemplo del capítulo 4.

### A.3 DINÁMICA

Se emplean tanto funciones MATLAB como Simulink. Las funciones MATLAB permiten obtener expresiones simbólicas de interés, tanto para el análisis del comportamiento dinámico como para el diseño del sistema de control.

Para ello es necesario suministrar una matriz denominada *dyn* de parámetros cinemáticos y dinámicos del manipulador. La matriz tiene tantas filas como articulaciones y 20 columnas. Las cinco primeras son las mismas que las de la matriz de parámetros de Denavit-Hartenberg. La sexta fila es la masa del enlace. Las filas 7 a 9 definen la posición del centro de masas del enlace con respecto al sistema de referencia del enlace. Las filas 10 a 15 son los elementos del tensor de inercia referido al centro de masas del enlace. La fila 16 corresponde al momento de inercia en el eje del motor de la articulación, la 17 es el coeficiente de reducción velocidad de articulación/velocidad del enlace, la columna 18 el coeficiente de fricción viscosa y las columnas 19 y 20 corresponden a las fricciones de Coulomb (rotaciones positiva y negativa referidas al eje del motor).

La herramienta desarrollada permite calcular todos los términos del modelo dinámico que se presenta en el capítulo 5:

$$\tau = M(q)q'' + G(q) + V(q, q') + F(q') \quad (\text{A.1})$$

siendo  $M(q)$  la matriz de masas,  $V(q, q')$  el vector de términos centrífugos y de Coriolis,  $G(q)$  el vector de términos gravitatorios y  $F(q')$  el vector que contiene los términos de fricción. La función *inertia(dyn,q)* permite obtener la matriz de masas del manipulador. Las funciones *gravity(dyn,q)*, *coriolis(dyn,q)* y *friction(dyn,q)* permiten obtener los pares correspondientes a los tres últimos términos de la expresión anterior.

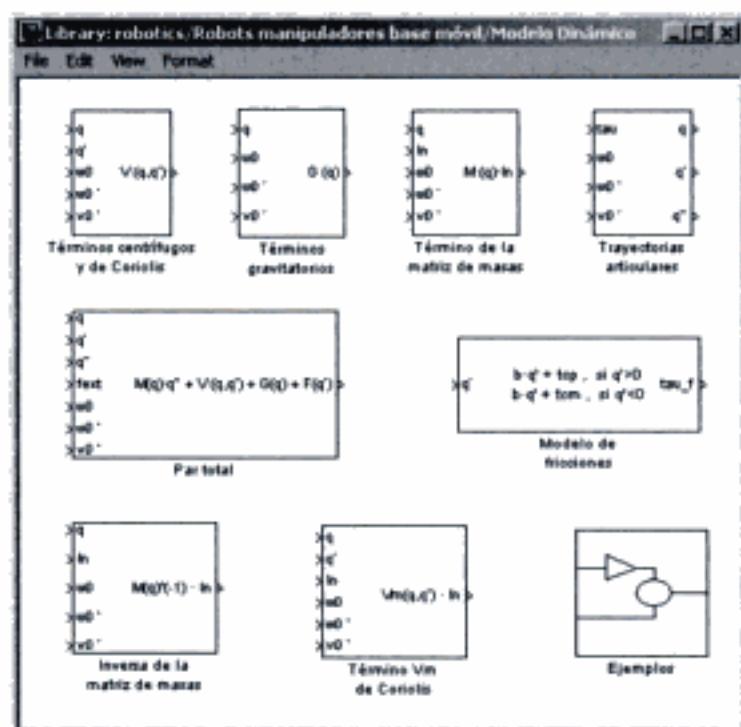
La función *rne(dyn, q, qd, qdd)* calcula el modelo dinámico completo de forma analítica empleando para ello el método de Newton-Euler recurrente, tal como se describe en el capítulo 5.

La función *fdyn* integra la expresión A.1 mediante la función de integración numérica *ode45* de MATLAB, devolviendo como resultado un vector de tiempo y dos matrices  $q$  y  $qd$  correspondientes a las trayectorias de las variables articulares y sus derivadas.

Alternativamente, si se desea integrar el modelo dinámico, es posible emplear Simulink, para lo cual se han previsto bloques que permiten implantar cada uno de los términos del modelo tal como se muestra en la figura A.3. El último bloque que se observa en dicha figura permite utilizar los diagramas Simulink de todos los ejemplos del capítulo 5.

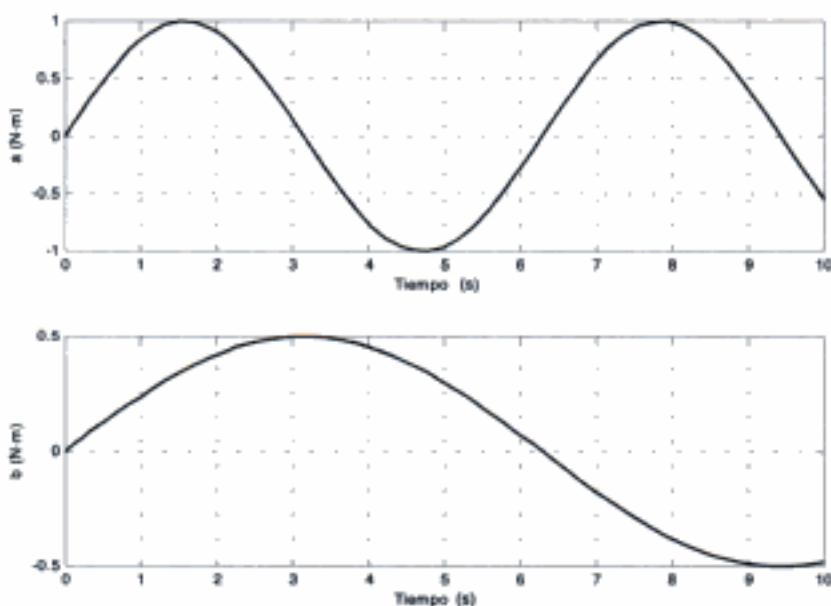
Los parámetros de cada uno de los bloques se introducen mediante los correspondientes cuadros de diálogo.

Se ha previsto que el robot manipulador pueda situarse sobre una plataforma móvil con un modelo cinemático tal como los que se mencionan en la sección anterior. Por



**Figura A.3:** Bloques de Simulink relacionados con la dinámica de los robots manipuladores.

tanto, se dispone de bloques correspondientes a la dinámica que tienen como entradas las variables articulares del manipulador  $q$  y sus derivadas  $q'$  y  $q''$ , así como  $\omega$ ,  $\omega'$  y  $v'$ , que son las velocidades y aceleraciones correspondientes a la base del robot manipulador.

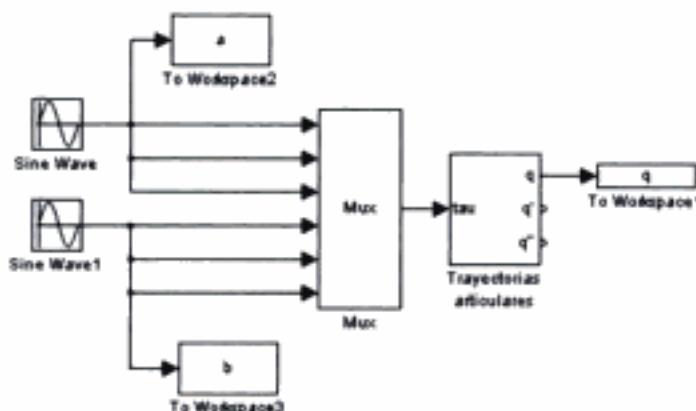


**Figura A.4:** Pares que se aplican a las articulaciones.

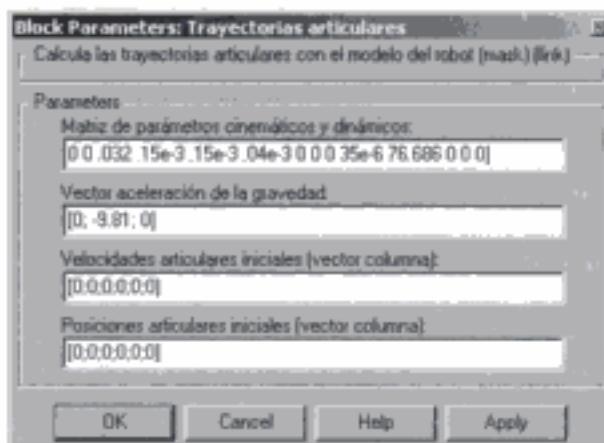
Supóngase que se aplica a las tres primeras articulaciones del robot Puma 560 el par senoidal **a** de la figura A.4 y a las articulaciones cuarta, quinta y sexta, el par senoidal **b** de dicha figura.

Para simular el comportamiento del robot Puma con estas entradas puede aplicarse el diagrama Simulink de la figura A.5.

El cuadro de diálogo correspondiente al bloque “Trayectorias articulares” aparece en la figura A.6.



**Figura A.5:** Esquema Simulink para calcular las trayectorias articulares que resultan de aplicar unos determinados pares a las articulaciones.



**Figura A.6:** Cuadro de diálogo correspondiente al bloque “Trayectorias articulares”.

La matriz de parámetros cinemáticos y dinámicos se muestra en la figura A.7. Los valores se han tomado de Armstrong y otros (1986).

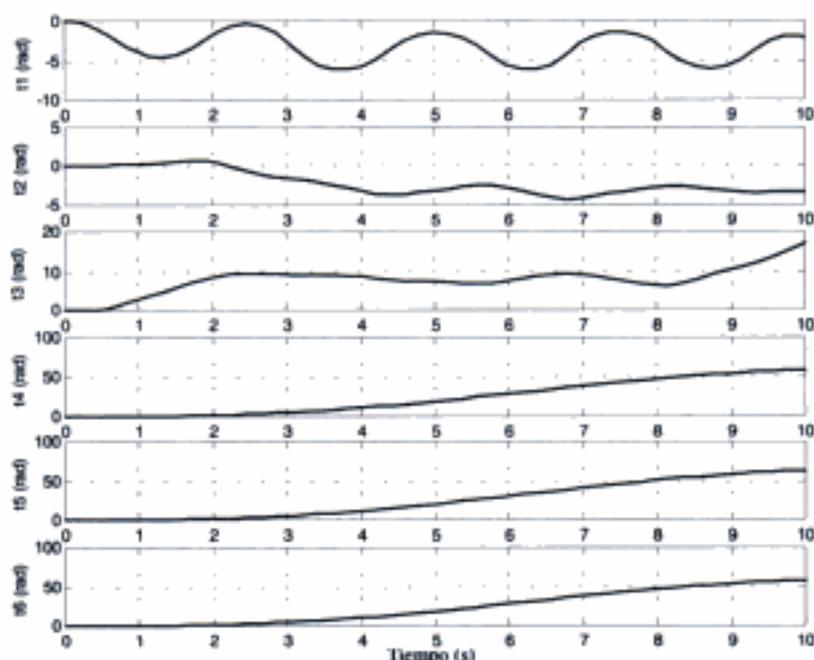
El resultado de la simulación se muestra en la figura A.8.

Se dispone también de un bloque que permite calcular los pares en las articulaciones a partir de las trayectorias articulares. Por tanto, se puede emplear ese bloque para comprobar que los resultados que se obtienen son los correctos. El diagrama Simulink de este bloque es el que se indica en la figura A.9.

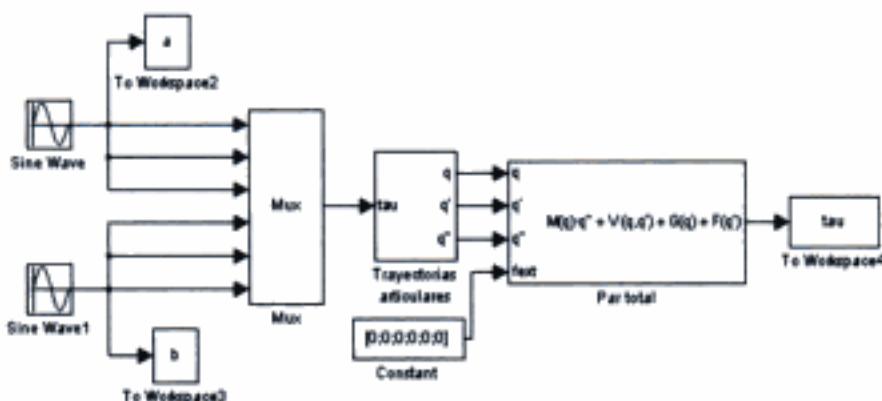
Matriz de parámetros dinámicos:

```
[0 0 0 0 0 0 0 0 0 0 0.35 0 0 0 291e-6 -62.6111 0 0 0;
-pi/2 0 0 0.2435 0 17.4 0.068 0.006 -0.016 .13 .524 .539 0 0 0 409e-6 107.815 0 0 0;
0 0.4318 0 -0.0934 0 4.8 0 -0.070 0.014 .066 .0125 .066 0 0 0 299e-6 -53.7063 0 0 0;
pi/2 -0.0203 0 .4331 0 0.82 0 0 -0.019 1.8e-3 1.8e-3 1.3e-3 0 0 0 35e-6 76.0364 0 0 0;
-pi/2 0 0 0 0 0.34 0 0 0 .3e-3 .3e-3 .4e-3 0 0 0 35e-6 71.923 0 0 0;
pi/2 0 0 0 0 0.09 0 0 .032 .15e-3 .15e-3 .04e-3 0 0 0 35e-6 76.686 0 0 0]
```

**Figura A.7:** Matriz de parámetros cinemáticos y dinámicos del robot Puma 560.



**Figura A.8:** Trayectorias articulares obtenidas en la simulación del modelo dinámico del Puma 560 con las entradas de la figura A.4. El gráfico que aparece más arriba corresponde a la primera articulación.



**Figura A.9:** Diagrama de bloques para comprobar el correcto cálculo de las variables articulares a partir de los pares aplicados.

## A.4 CONTROL

Las estrategias de control de las articulaciones de robots manipuladores, conociendo las trayectorias que deben seguir dichas articulaciones, han sido desarrolladas en Simulink. Para ello, además de los bloques correspondientes al modelo dinámico, se han desarrollado bloques que facilitan la implantación de diversas estrategias de control, entre las que se incluyen las mencionadas en el capítulo 8. De esta forma pueden implantarse las estrategias de par calculado y sus diferentes casos particulares, incluyendo el control PID desacoplado de las articulaciones.

Supóngase que se pretende simular la estrategia de control del par computado con un PD para el Puma 560. El esquema que se puede emplear para ello, es el que aparece en la figura A.10.

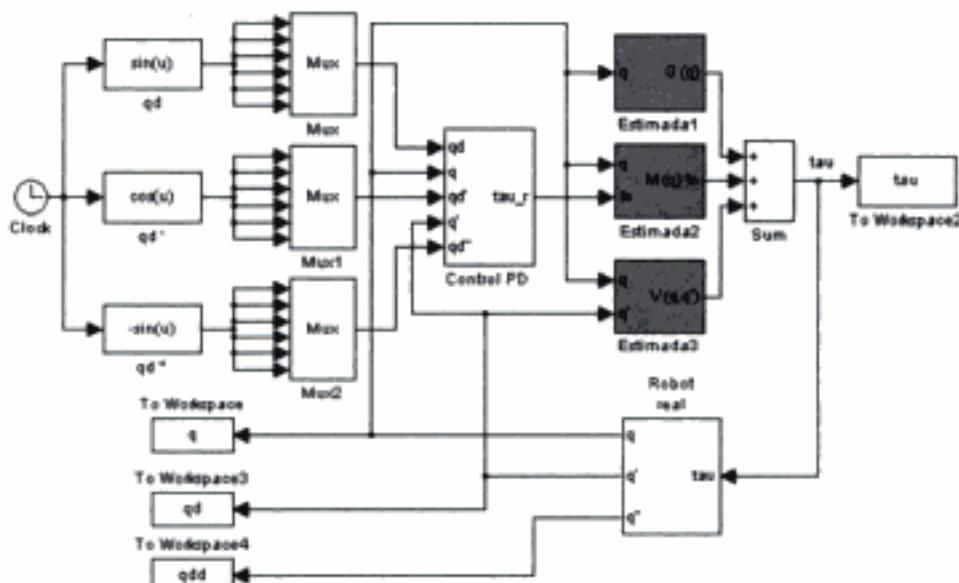


Figura A.10: Diagrama Simulink para la simulación de la técnica del par computado.

Se pretende que el robot se mueva según unas trayectorias articulares dadas por:

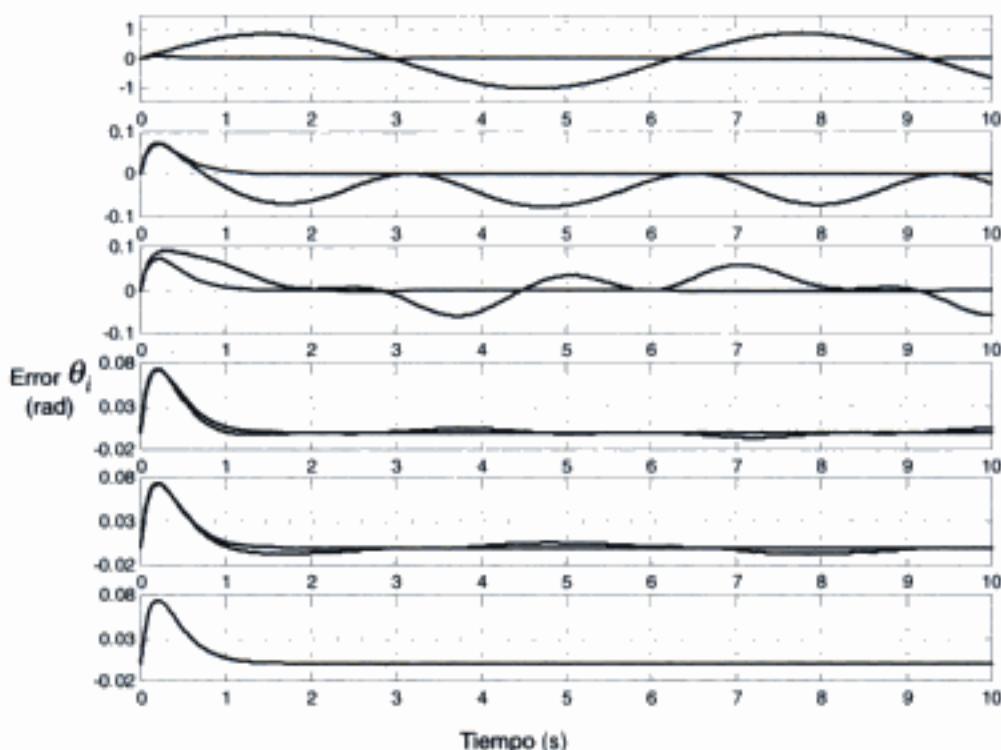
$$\theta_i(t) = \operatorname{sen} t \quad i = 1, \dots, 6 \quad (\text{A.2})$$

Se toman los siguientes valores para las constantes del PD:

$$K_p = 25I_6 \quad K_v = 10I_6 \quad (\text{A.3})$$

donde  $I_6$  es la matriz unidad de dimensiones  $6 \times 6$ .

En primer lugar se efectúa una simulación considerando que los parámetros del brazo son perfectamente conocidos y, a continuación, se supone que aparece una fricción viscosa de 0,1 N·m s en la primera articulación del Puma no considerada en el modelo.



**Figura A.11:** Comparación de los errores de posición en las articulaciones cuando existe una fricción viscosa no modelada frente al caso ideal de estimación perfecta de todos los parámetros.

En la figura A.11 se presentan gráficas que comparan las respuestas de las articulaciones en uno y otro caso. Obsérvese como en el caso en que la estimación es perfecta, el error en las articulaciones tiende a cero después de un corto transitorio. Sin embargo, cuando aparece la fricción viscosa no modelada, el comportamiento se deteriora significativamente en las primeras articulaciones.

Los pares que se aplican a las articulaciones en el caso en que aparece la fricción no modelada, se muestran en la figura A.12.

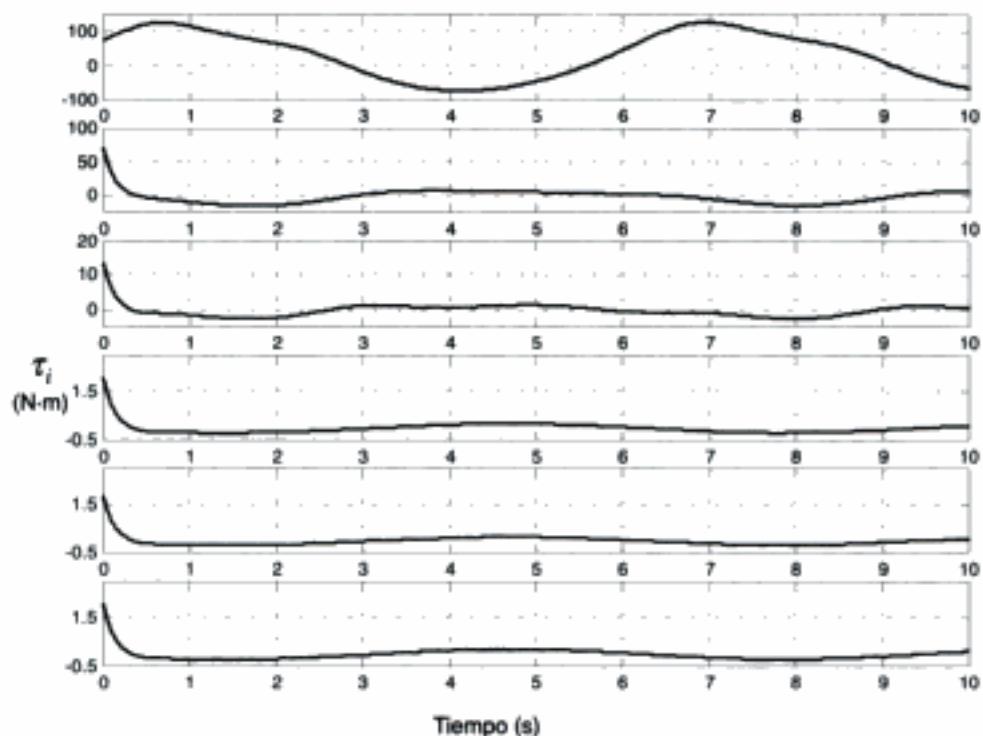
Es posible también la aplicación de las técnicas de control adaptativo (Slotine y Lee, 1988) y control con aprendizaje (Craig, 1988) que se introducen en el capítulo 8, para lo cual se han previsto bloques Simulink específicos. Se han desarrollado funciones dentro de dichos bloques que facilitan la tarea del diseñador al permitir obtener automáticamente las matrices y vectores de las leyes de control.

En la figura A.13 se muestra el esquema Simulink para la aplicación al Puma 560 del método de control con aprendizaje que se expuso en el capítulo 8.

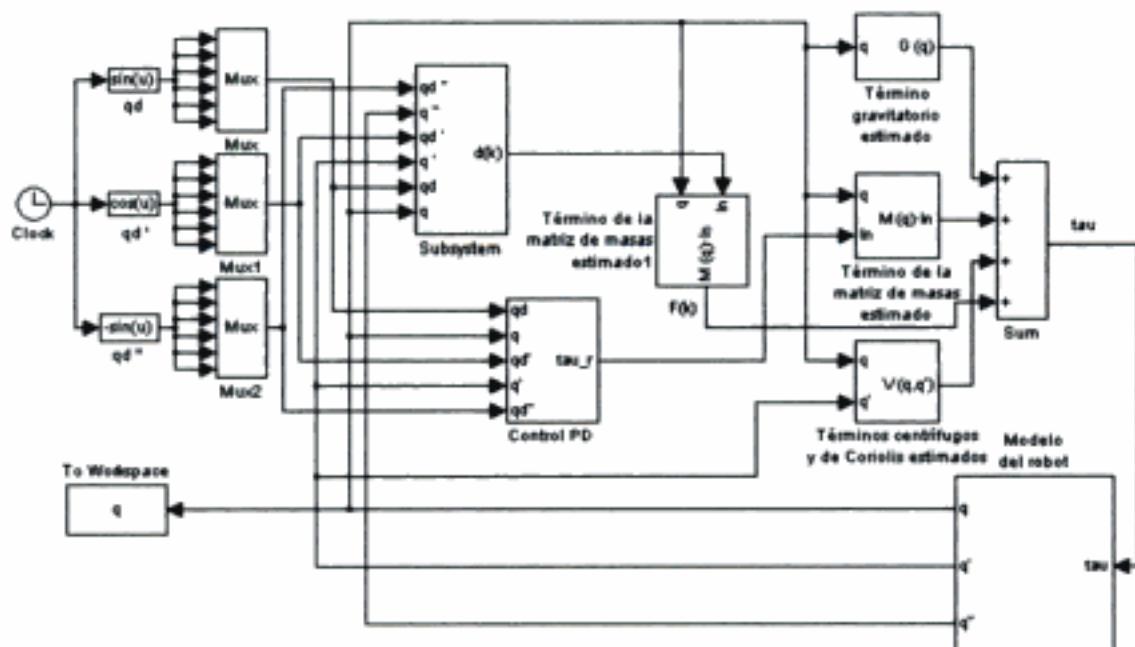
Se supone que aparece una fricción no modelada en la primera articulación de 0,1 N·m s (al igual que en el caso anterior). Los parámetros que se eligen para la ley de control son:

$$K_p = 125 I_6 \quad K_v = 50 I_6 \quad \mu = 2 I_6 \quad (\text{A.4})$$

donde  $I_6$  es la matriz unidad  $6 \times 6$ .



**Figura A.12:** Pares aplicados a las articulaciones cuando se emplea la técnica del par computado en el caso de una fricción viscosa no modelada en la primera articulación.  
El gráfico que aparece más arriba corresponde a la primera articulación.

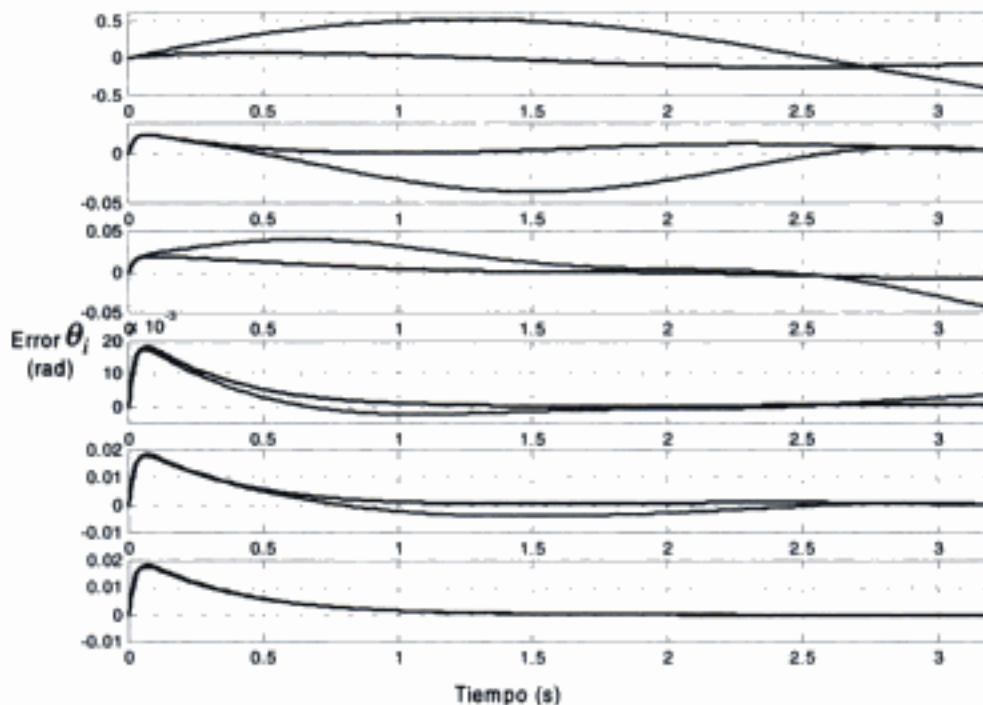


**Figura A.13:** Diagrama Simulink que permite simular el control con aprendizaje.

En la figura A.14 se pueden apreciar los errores en posición de las diferentes articulaciones para las iteraciones número 1 y 5 del método que se presenta en el capítulo 8. Obsérvese como en cinco iteraciones el comportamiento mejora de forma importante.

La herramienta MATLAB/Simulink incluye también diversas funciones y bloques Simulink diseñadas para la implantación de los métodos de control de vehículos robóticos que se incluyen en el capítulo 9.

En particular, una vez obtenido el camino que debe seguir el robot mediante los métodos de generación de trayectorias, es posible implantar métodos geométricos de seguimiento de caminos.



**Figura A.14:** Errores en posición en las iteraciones 1 y 5 del método de control por aprendizaje.

Supóngase que se quiere simular el comportamiento dinámico del Puma 560 cuando su base va fijada al centro de guiado de un triciclo, que sigue una determinada trayectoria mediante el método de persecución pura que se presenta en el capítulo 9. En primer lugar, se puede calcular la trayectoria que seguiría el triciclo y almacenar su aceleración lineal, velocidad angular y aceleración angular a lo largo del trayecto (que serán las mismas que las de la base del Puma 560). El esquema que permite efectuar dichos cálculos es el que se muestra en la figura A.15. La trayectoria que se pretende seguir es una línea recta y el triciclo parte de las coordenadas  $(0, 0, 0)$  con una orientación inicial de 0 rad. Tomando un valor de 0,5 para el parámetro de distancia al punto objetivo ("look-ahead"), el resultado es el que aparece en la figura A.16.

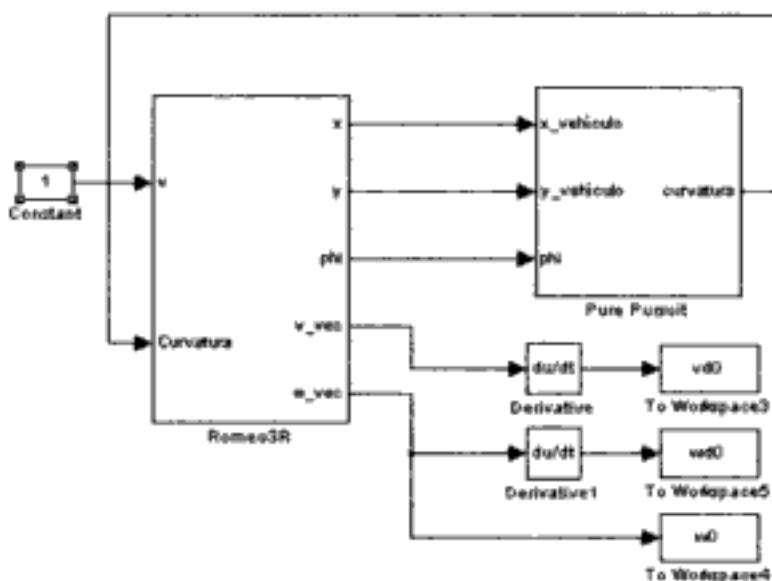


Figura A.15: Esquema Simulink para la técnica de persecución pura.

Para simular el comportamiento dinámico del Puma 560 sobre la plataforma móvil se emplea el esquema de la figura A.17. Se supone que no se cometen errores en la estimación de los parámetros del manipulador, y que se utilizan las mismas constantes para el PD que en el ejemplo del par computado. Asimismo, se considera que el Puma se pone en marcha en el instante  $t = 0,05$  s (es decir, una vez que el triciclo ya está en movimiento).

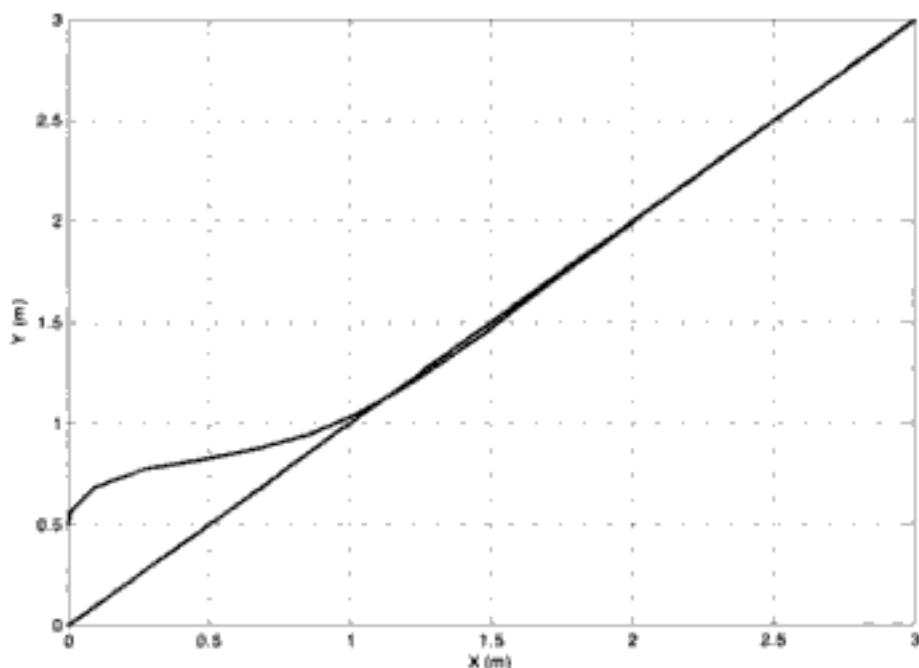
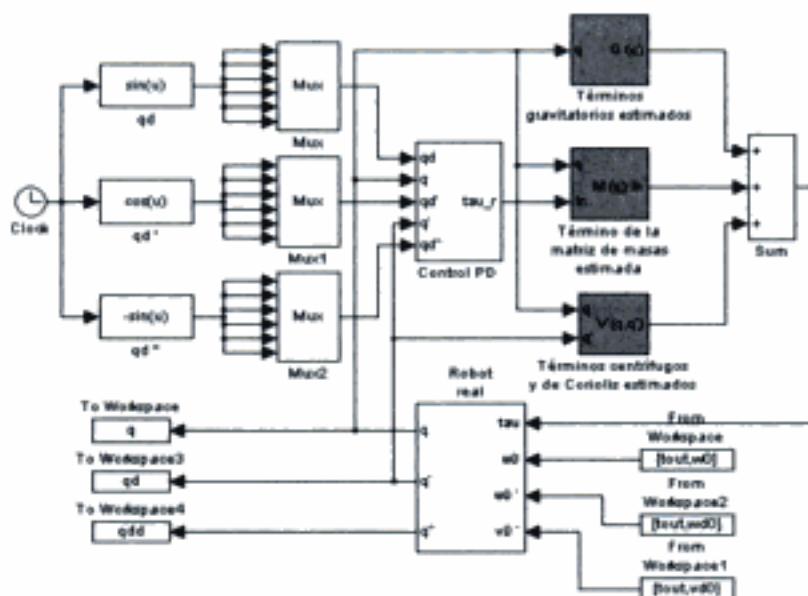
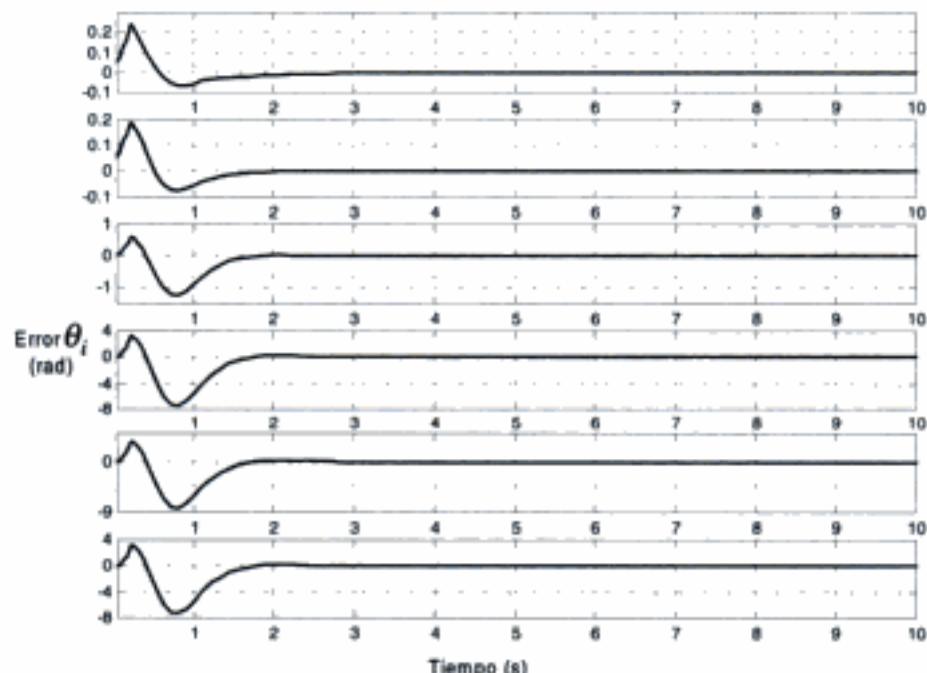


Figura A.16: Resultado de la persecución pura con  $L = 0,5$ .



**Figura A.17:** Esquema para simular el comportamiento dinámico del Puma 560 montado sobre una plataforma con locomoción tipo triciclo. La técnica de control empleada para el manipulador es la del par computado.

En la figura A.18, se muestra el error en posición para cada una de las articulaciones del Puma 560. Se puede observar que aparecen unas perturbaciones debidas al movimiento de la base del robot que no existían en el caso en que el robot tenía la base fija (trayectorias de la figura A.11 en caso de estimación perfecta).



**Figura A.18:** Errores en posición de las articulaciones del Puma 560 cuando su base va fijada al centro de guiado de un triciclo que sigue una determinada trayectoria.

Es posible también la aplicación de diversos métodos basados en la Teoría de Control mediante la aplicación de leyes de control lineales o no lineales (Canudas y otros, 1997). Asimismo, puede implantarse el método de control predictivo generalizado basado en la linealización del modelo de la bicicleta tal como se indicó en el capítulo 9.

## A.5 GENERACIÓN DE TRAYECTORIAS

---

Se han desarrollado una serie de funciones de MATLAB que permiten generar trayectorias empleando diversos métodos de interpolación. De esta forma, existen funciones específicas de generación de trayectorias mediante polinomios de orden tres y orden cinco a partir de las condiciones iniciales y finales, así como la implantación del método de enlaces parabólicos del capítulo 10.

Se ha previsto también la generación de trayectorias con aproximación a puntos de paso empleando para ello tanto los mencionados enlaces parabólicos como curvas B-spline, polinomios de Bezier y curvas  $\beta$ -spline (Barsky, 1988).

Existen funciones para generar trayectorias tanto en el espacio articular como en el cartesiano. Asimismo, es posible aplicarlas tanto para robots manipuladores como para vehículos robóticos.

Se ha previsto también la próxima incorporación de funciones que permiten resolver problemas específicos tales como la generación de trayectorias de aparcamiento en paralelo y en diagonal de vehículos robóticos (Gomez-Bravo y otros, 2001).

## A.6 INTERFASE GRÁFICA

---

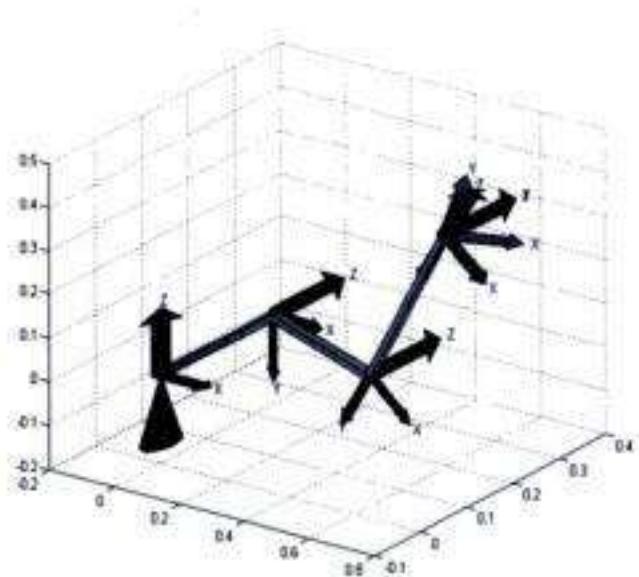
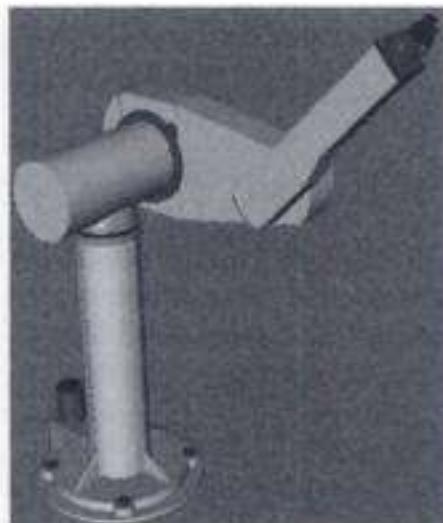
Obsérvese que algunas de las funciones MATLAB mencionadas en apartados anteriores, tales como la función `plotbot`, permiten la representación gráfica empleando para ello órdenes específicas de MATLAB.



**Figura A.19:** Representaciones tridimensionales en RoboWorks de los robots de los ejemplos del capítulo 4.

Si se desea una visualización de mayor realismo puede acudirse a otros programas específicos. En particular se puede utilizar el programa RoboWorks (2001) que facilita el diseño gráfico en 3D y la visualización de los movimientos resultantes. En la figura A.19 se muestran representaciones gráficas de los robots correspondientes a los ejemplos 4.1 y 4.2 del capítulo 4. En la figura A.20 se compara la representación del Puma 560 en RoboWorks con el resultado de la ejecución de la función plotbot para unos mismos valores de las variables articulares.

Los ficheros con las trayectorias generadas con MATLAB y Simulink mediante HEMERO son adquiridos por RoboWorks para generar las animaciones correspondientes a las trayectorias.



**Figura A.20:** Representación del Puma 560 en RoboWorks junto con el gráfico resultante de la ejecución de la función plotbot para unos mismos valores de las variables articulares.

## A.7 CONCLUSIONES

En este apéndice se ha introducido la herramienta informática denominada HEMERO para el estudio de los robots manipuladores y móviles. La herramienta combina MATLAB y Simulink para la representación y modelado cinemático y dinámico de los robots. Asimismo, incluye diversos métodos de control tanto de manipuladores robóticos como de robots móviles.

La herramienta permite simular todos los ejemplos incluidos en el texto. Asimismo, el lector puede crear sus propios ejemplos y aplicarles diferentes métodos de control añadiéndole nuevos métodos mediante las funciones oportunas. En este apéndice se utiliza como ejemplo el robot manipulador Puma 560 sobre el cual se ha aplicado el método de control del par computado y el del control con aprendizaje. Asimismo, se ha simulado un robot móvil sobre el cual se ha colocado el robot Puma 560, con la base situada en el centro de guiado de dicho robot móvil, observándose el efecto del mo-

vimiento de la base (controlada mediante el método de la persecución pura) sobre las articulaciones del Puma.

Los resultados numéricos se ilustran mediante representaciones gráficas simplificadas. Asimismo, es posible generar ficheros que pueden ser empleados para simular en tres dimensiones los robots utilizando para ello la herramienta gráfica RoboWorks. Pueden observarse representaciones realizadas mediante RoboWorks de los robots de los ejemplos del capítulo 4 y del Puma 560.

Para poder ejecutar la herramienta, el lector necesita disponer de MATLAB 5.2, o versiones posteriores, y Simulink. Asimismo, si desea realizar representaciones gráficas realistas, es posible emplear RoboWorks 2.0 u otro programa gráfico que pueda leer los ficheros generados por MATLAB o Simulink. Si no dispone de estos productos, algunas funcionalidades de la herramienta pueden utilizarse también a través de Internet (<http://www.esi.us.es/hetero>). En este sitio Web se encontrarán también futuras actualizaciones de la herramienta.

## A.8 REFERENCIAS

---

- Armstrong B., O. Khatib, y J. Burdick, "The explicit dynamic model and inertial parameters of the Puma 560 arm," en Proc. IEEE Int. Conf. Robotics and Automation, vol. 1, (Washington, USA), pp. 510-18, 1986.
- Barsky B.A. (1988). "Computer graphics and geometric modelling using  $\beta$ -Splines". Springer.
- Canudas de Wit C., B. Siciliano, y G. Bastin (1997). "Theory of Robot Control". Springer.
- Corke, P.I., (1996), "A Robotics Toolbox for MATLAB", IEEE Robotics and Automation Magazine, vol. 3, núm. 1, pp. 24-32.
- Craig, J.J., (1988), "Adaptive control of mechanical manipulators", Addison Wesley.
- Craig J.J. (1989). "Introduction to robotics". Addison Wesley. Segunda edición.
- Gómez-Bravo F., F. Cuesta y A. Ollero (2001). "Parallel and diagonal parking in nonholonomic autonomous vehicles". Engineering Applications of Artificial Intelligence. Pergamon. Núm. 4.
- Hartenberg, R.S. y J. Denavit, (1955), "A kinematic notation for lower pair mechanisms based on matrices", Journal of Applied Mechanics, vol. 77, pp. 215-221.
- Maza J.I. y A. Ollero (2000). "Herramienta MATLAB-Simulink para la simulación y el control de robots manipuladores y móviles". Actas de las XXI Jornadas de Automática. Sevilla.
- Maza J.I. y A. Ollero (2001). "HEMERO. Una herramienta MATLAB-Simulink para la robótica". Publicada en CD-ROM. Marcombo.
- Paul R.P., 1981. "Robot manipulators: mathematics, programming and control". MIT Press, Massachusetts.
- RoboWorks (2001). <http://www.newtonium.com/>
- Slotine J., Li, W. (1988). "Adaptive manipulator control: a case study", IEEE Trans. on Automatic Control, Vol. 33, pp. 995-1003.
- Valera, A., Tornero, J., Salt, J.J., (1999) "Desarrollo de una Librería de Control de Robots para MATLAB/Simulink", Actas de las XX Jornadas de Automática, Salamanca, pp. 131-136.

---

# Índice alfabético

---

## A

abrir pinza, 351  
accesibilidad, 147  
aceleración angular, 121  
– lineal, 120  
acelerómetro, 191  
activación línea de salida, 357  
actuador, 5  
adaptabilidad, 150  
AGV, 194  
algoritmo de mínima distancia, 384  
ángulos de Euler Z-Y-X, 60  
-- Z-Y-Z, 61, 424  
ángulos O-A-T, 349  
– RPY, 57, 424  
aproximación, 350  
arquitectura centralizada, 157  
– descentralizada, 157  
– híbrida, 152  
– reactiva, 151  
arquitecturas PC, 162  
articulación, 5, 16  
– de rotación, 203  
asignación de ganancias, 221, 224  
AURIGA, 31  
AURORA, 33  
autocalibración, 168  
autoiris, 414  
automatización industrial, 2

## B

B spline, 314-316  
beta spline, 297, 312  
bípedo, 35  
bola de control, 402, 406, 408  
botonera de programación, 343, 354  
brazo articulado, 16  
búsqueda del espacio libre, 388  
– heurística, 386

## C

cámara de vídeo, 11  
– panorámica, 415  
camino más corto, 386  
campo de vista angular, 415  
– potencial, 393  
– generalizado, 393  
capacidad de carga, 147  
– de evolución, 149  
carga y descarga de máquinas, 6  
cerrar pinza, 351  
cilindro-esfera, 377  
cinemática, 64  
cinestesia, 401  
círculo de osculación, 308  
codificador lineal, 171  
– óptico, 170, 171  
– absoluto, 172

codificador óptico relativo, 173  
coeficiente de amortiguamiento, 204  
compás de efecto Hall, 188  
– giromagnético, 189  
– magnético, 186  
complejidad de planificación de caminos, 372  
comportamiento, 151, 300  
comprobación de señales binarias, 358  
compromiso planes/reacciones, 155  
comunicación bidireccional, 168  
condición de rodadura, 99  
configuración angular, 21  
– cartesiana, 18  
– cilíndrica, 20  
– esférica, 21  
– rómbica, 109  
– Scara, 22  
cono generalizado, 388  
control adaptativo, 224, 433  
– automático, 2  
– basado en el modelo, 204  
– bilateral, 405  
– compartido, 401  
– de articulación, 159  
– de AURORA, 162-163  
– de efecto final, 154  
– de esfuerzo, 247  
– de impedancia, 402, 407  
– de la dirección, 259  
– de las cámaras, 415  
– de movimientos, 4  
– de RAM-1, 160  
– de rigidez, 248, 404  
– de Romeo, 162  
– de supervisión, 401, 412  
– de teleoperación, 403  
– de trayectoria continua, 304  
– de una articulación simple, 199  
– de vehículos autónomos, 258, 435  
– de velocidad, 259  
– del iris, 414  
– del robot Puma, 159  
– en bucle cerrado, 200  
– en espacio cartesiano, 243  
– híbrido esfuerzos/posición, 251  
– inteligente, 146  
– intercambiado, 401  
– mediante voz, 415  
– PD, 201  
– PID, 198, 201

control predictivo generalizado, 293  
– punto a punto, 304  
– reactivo, 300, 366  
– supervisado, 401  
controlabilidad, 271  
controlador de inercia adaptativa, 231  
– de rigidez PD, 251  
creación de transformación, 356  
cuaternion, 62  
cubo transformado, 384  
curva clotoidal, 320  
– de Bezier, 316  
– elástica, 313  
– paramétrica, 305  
curvatura, 101, 106, 308  
– continua, 313, 320  
– requerida, 293

## D

definición de prioridades en arquitectura de control, 156  
Denavit-Hartenberg, 66  
detección de colisión, 370  
– – empleando modelo, 382  
– de obstáculos mediante láser, 374  
diagrama de Voronoi, 388  
dinámica del actuador, 264  
– en espacio cartesiano, 246  
discontinuidad en curvatura, 319  
diseño funcional, 153  
dispositivo de guiado, 343  
distancia recorrida mínima, 390

## E

ecuación de Liapunov, 229  
– diofántica, 295  
efector final, 18, 24  
eficiencia, 149  
ejecución inversa, 342  
– paso a paso, 342  
embrague de posición, 410  
– de tiempo, 410  
energía cinética, 119  
– potencial, 119  
entorno dinámico, 385  
errores de estimación, 230, 233  
– en la estimación de las masas, 213

escalado de velocidad, 278  
 espacio de configuraciones, 379, 389  
 – de trabajo alcanzable, 74  
 – de manipulación con destreza, 75  
 especificación de localización, 354  
 espera de un evento, 357  
 esquema, 152  
 estabilidad de Liapunov, 227, 228, 248  
 estimación de la orientación, 111, 185  
 – de la posición, 111, 185, 191  
 estructura jerárquica en planificación, 375  
 Ethernet, 160, 162  
 evitación de colisión, 371  
 expansión de obstáculos, 378

**F**

factorización de fuerzas centrífugas y de Coriolis, 227, 232  
 fiabilidad, 150  
 filtrado, 168  
 final de carrera, 178  
 forma de cadena, 275  
 frecuencia natural no amortiguada, 204  
 fricción, 118, 119, 131, 202  
 fuerza atractora, 394  
 – centrífuga, 119, 131  
 – de Coriolis, 119, 131  
 – gravitatoria, 119, 131  
 – normal, 250, 253  
 – repulsora, 394  
 – tangencial, 253  
 – virtual, 394  
 función cognitiva, 155  
 – de Liapunov, 227, 232, 282, 290  
 – de ponderación, 312  
 – de potencial generalizada, 393  
 – estratégica, 155  
 – inteligente, 146  
 – lagrangiana, 119  
 – lineal con enlace parabólico, 330, 338  
 funciones de ayuda a la programación, 372  
 fusión multisensorial, 168

**G**

gain scheduling, 221, 224  
 galga extensiometrífica, 182  
 generación de trayectoria, 6, 303, 438

generación de trayectoria articular, 154, 322, 338  
 generador de caminos articulares, 338  
 gestión de ejecución, 156  
 – de excepciones, 360  
 – de trazas de ejecución, 342  
 gimbal, 186  
 girocompás, 189  
 giróscopo, 188  
 – electrónico, 190  
 – mecánico, 188  
 – óptico, 190  
 goal driven, 152  
 GPS, 191  
 – diferencial, 193  
 grado de autonomía, 150  
 – de direccionamiento, 271  
 – de libertad, 18  
 – de movilidad, 271  
 gráfico predictivo, 409, 419  
 grafo de vértices, 386  
 gravedad, 118  
 guante, 402  
 guiado manual, 6  
 – mediante balizas, 195  
 – químico, 195

**H**

HEMERO, 422  
 hipótesis para planificación de caminos, 385  
 histéresis, 168  
 hot editing, 342  
 hull-convex, 316, 317

**I**

implantación, 156  
 incremento de control proyectado, 294  
 INS, 191  
 integración numérica, 133, 428  
 inteligencia artificial, 7, 12  
 interfase persona-máquina, 12  
 intersección entre primitivas, 377  
 inversa de mínimos cuadrados amortiguadores, 244  
 iteración hacia dentro, 125, 129  
 – hacia fuera, 125, 127

### J

- jacobiano, 426
  - de robots móviles, 102
  - del manipulador, 89
- joystick, 343, [402](#), 406

### L

- Lagrange-Euler, 119
- láser, 11
- lenguaje AL, 346
  - de movimiento primitivo, 346
  - de nivel objeto, 345
  - – robot, 345
  - – tarea, 345
  - de programación estructurada, 346
  - – textual explícita, 346
  - V+, 347
  - VAL II, 345-347, 350-365
- ley de adaptación, 229
  - – de parámetros, 231
  - de aprendizaje, 238
- límites del espacio de trabajo, 379
- línea de visibilidad, 386
- linealidad, 168
- localización, 350
- locomoción Ackerman, 28, 29, 108
  - diferencial, 30, 104, 111
  - síncrona, 32, 104
- Locomotion Emulator, 32
- longitud de arco, 306

### M

- maestro-esclavo, [401](#), [403](#)
- magnetómetro, 187
  - magnetoinductivo, 187
- malla de puntos, 385
- maniobrabilidad, 5
- manipulador industrial, 16
  - redundante, 5
- mano, 27
- máquina eléctrica síncrona, 175
- matriz de controlabilidad, 272
  - de masas, 119, 131
  - de parámetros cinemáticos y dinámicos, 428
  - – de Denavit-Hartenberg, 424

- matriz de rotación, 41, 46, 424
  - de transformación homogénea, 48, 424
- método de Liapunov, 282
- de Newton-Euler, 120
- – iterativo, 122, 133
- geométrico de seguimiento de caminos, [261](#)

- reactivo, 385, 392
- microprocesador, 159
- mínimo local, [396](#)
- radio de curvatura, 319
- modelado mediante ocupación de celdas, 375
- modelo cinemático de robot móvil, 97
  - de la bicicleta, 102, [106](#), 143, 264, 290
  - de la respuesta del operador humano, [408](#)
  - de referencia, 225, [275](#)
  - dinámico, 116, 428
    - – de un manipulador, 119, 131
    - – de un vehículo robótico, 142
    - directo de manipulador, 66, [425](#)
      - – con  $n$  articulaciones, 68
    - geométrico, 64
    - inverso del manipulador, 74, [425](#)
  - lineal, 117
- momento angular, 122
- momento de inercia, 117
- motor controlado por el inducido, 199
- mover y esperar, [408](#)
- movimiento con interpolación articular, 350
  - de articulación, 351
  - en línea recta, 350, 366
  - precavido, [344](#), 357
- múltiples soluciones del problema inverso, 79

### N

- navegación autónoma, 9
  - hiperbólica, 191
  - inercial, 10
- niveles de jerarquización, 153
  - de arquitectura de control, 154

### O

- octree, 375, 383
  - $N$ -objetos, 376
  - – región, 376

odometría, 10, 261  
 operador de rotación, 50  
 – de translación, 49  
 optoguiado, 195  
 órgano terminal, 18  
 orientación del efecto final, 22  
 – en el espacio, 44  
 – en el plano, 39

## P

par compensador, 202  
 – computado, 204, 432  
 – adaptativo, 226  
 – de control, 202  
 – de perturbación, 205  
 – de rotación, 61  
 parada de ejecución, 360  
 – de movimiento, 360  
 parámetro de Denavit-Hartenberg, 66, 79  
 pasividad, 408  
 pata, 34  
 péndulo invertido, 118, 203  
 percepción, 4, 373  
 – activa, 374  
 – sensorial del entorno, 300  
 persecución pura, 261, 263  
 PD, 206  
 PID, 198, 206, 208  
 pinza, 18  
 pista de deslizamiento, 31  
 planificación, 4  
 – de camino, 7, 9  
 – libre de obstáculo, 371  
 – de movimiento, 154  
 – en sistema telerrobotico, 373  
 – estratégica, 12, 151  
 – reactiva, 13  
 – secuencial, 392  
 polígono convexo, 380  
 – de control, 314  
 polinomio cúbico, 310, 323  
 – con punto de paso, 324  
 – quíntico, 267, 310, 327  
 posición en el espacio, 44  
 – en el plano, 39  
 posicionamiento de un eje, 117  
 postura, 258  
 potenciómetro, 168  
 – lineal, 169

potenciómetro rotatorio, 170  
 precisión, 5, 148  
 predicción de salida, 294  
 primitivas de sólidos, 376, 383, 388  
 prioridad, 360  
 procesamiento de imagen, 374  
 programabilidad, 149  
 programación a diferentes niveles, 341  
 – de robots, 341  
 – móviles, 365  
 – en línea, 343  
 – en modo enseñanza, 343  
 – no textual, 343  
 – por aprendizaje, 343  
 – por guiado, 342  
 propagación de velocidad, 87, 427  
 puente de Wheatstone, 183  
 PUMA, 22, 347-349, 352, 424-426, 432, 435  
 punto de control, 313  
 – de paso, 303, 319, 321  
 – de precisión, 352  
 – intermedio inalcanzable, 321

## R

radio de curvatura, 308  
 – de giro, 101, 106  
 RAM1, 33  
 rapidez de respuesta, 147  
 realidad virtual, 401, 402  
 realimentación de esfuerzos, 402-403, 405-406  
 – visual, 8  
 recursos hardware, 156  
 – software, 156  
 registro continuo, 344  
 – en puntos de paso, 344  
 regulación automática, 2  
 rejilla, 375, 387  
 – de certidumbre, 375, 394  
 relajación de sincronía, 410  
 repetibilidad, 148, 168  
 réplica del robot, 343  
 representación de rejillas, 418  
 resolución, 148, 168  
 – del modelo inverso, 81  
 resolvedor, 176  
 restricción cinemática, 97  
 – holónoma, 98  
 – no holónoma, 10, 98, 99

- retardo, [403, 406, 408-409](#)  
retraso de transmisión de información, [12](#)  
rigidez, [247](#)  
robot, [1-8, 11-12](#)  
– aéreo, [36](#)  
– autónomo, [11, 12](#)  
– de funcionamiento repetitivo, [12](#)  
– de locomoción diferencial, [272](#)  
– flexible, [5, 27](#)  
– industrial, [3](#)  
– manipulador, [5, 16](#)  
– móvil, [8-11, 28](#)  
– – articulado, [36](#)  
– – omnidireccional, [273](#)  
– redundante, [26](#)  
– serpiente, [26](#)  
– submarino, [36](#)  
– teleoperado, [11](#)  
– trepador, [35](#)  
robótica, [1, 2, 5](#)  
– espacial, [158](#)  
Robotics Toolbox, [39](#)  
RoboWorks, [439](#)  
Romeo 3R, [30](#)  
Romeo 4R, [266](#)  
rotación de un ángulo, [424](#)  
ruedas de direccionamiento, [100](#)  
– orientables, [272](#)  
– suecas, [34, 100](#)  
ruido, [168](#)
- S**
- salida según eje Z, [350](#)  
seguimiento de caminos, [284, 367, 435](#)  
– de objetos móviles, [300](#)  
– de trayectorias, [275](#)  
– visual, [367](#)  
sensibilidad, [168](#)  
sensor, [6, 7, 166](#)  
– capacitivo, [179](#)  
– de contacto, [374](#)  
– de efecto Hall, [179](#)  
– de esfuerzo, [408](#)  
– de infrarrojo, [179](#)  
– de matriz de agujas, [180](#)  
– de muñequera, [184](#)  
– de percepción del entorno, [1](#)  
– de proximetría, [300](#)
- sensor de proximidad, [179](#)  
– de silicio, [168](#)  
– de tacto, [180](#)  
– de ultrasonido, [11, 180](#)  
– de visión, [7](#)  
– incremental, [167](#)  
– inductivo, [178](#)  
– láser de tiempo de vuelo, [180](#)  
– – de medida de desfase, [180](#)  
– óptico, [179](#)  
– – tipo barrera, [179](#)  
– táctil óptico, [181](#)  
señal auxiliar de error, [232](#)  
servocontrol, [6, 154](#)  
servomecanismo, [6](#)  
servosistemas, [2](#)  
seudoinversa, [103, 244](#)  
sincronización con el entorno, [359](#)  
– local-remoto, [409](#)  
singularidad, [89, 322](#)  
sistema asociado a la herramienta, [356](#)  
– bilateral, [403](#)  
– centralizado, [156](#)  
– con múltiples robots, [385](#)  
– de control, [4, 6](#)  
– de guiado avanzado, [344](#)  
– de navegación inercial, [191](#)  
– de percepción, [10](#)  
– de posicionamiento global, [192](#)  
– de programación, [341](#)  
– de referencia, [64, 424](#)  
– – base, [347](#)  
– – efecto final, [347](#)  
– lineal de segundo orden, [204](#)  
– – variante en el tiempo, [276](#)  
– mecánico, [5](#)  
– móvil de coordenadas, [291](#)  
– robot, [3](#)  
skid steer, [31](#)  
smart sensors, [168](#)  
soluciones hardware, [158](#)  
– software, [158](#)  
spline, [311](#)  
– básica, [314](#)  
– cúbica, [312](#)  
subespacio del espacio libre, [392](#)  
subsumption, [151, 163](#)  
supervisión de control de movimientos, [155](#)  
– de planificación, [155](#)  
synchro drive, [32](#)

**T**

teclado funcional de robot, [367](#)  
 técnicas de interpolación, [309](#)  
 teleactuación, [400](#)  
 teleoperación, [146, 400](#)  
 teleoperador, [3](#)  
 telepresencia, [401, 402](#)  
 teleprogramación, [411](#)  
 telerrobótica, [11, 400](#)  
 telesensor-programming, [412](#)  
 telesensorización, [400, 413](#)  
 tensor de inercia, [117, 122, 126, 138](#)  
 teorema de Frobenius, [99](#)  
 teoría de control, [271](#)  
 término integral, [206](#)  
 Terregator, [31](#)  
 tipos de articulaciones, [16](#)  
 transformación, [352](#)  
 – compuesta, [54, 354](#)  
 – desplazada, [356](#)  
 – inversa, [52, 356](#)  
 transformador diferencial variable, [176](#)  
 – lineal, [176](#)  
 – rotatorio, [176](#)  
 transmisión de imagen, [419](#)  
 transmisor/receptor de tres fases, [175](#)  
 traslación, [47, 424](#)  
 trayectoria articular, [133, 322](#)  
 – de aproximación, [296](#)  
 – deseada, [260](#)  
 triciclo clásico, [29, 105, 111](#)

**U**

Unix, [160-161](#)

Uranus, [34](#)

utilización de un sistema CAD, [344](#)

**V**

VAL II, [345-347, 350-365](#)  
 variable de tipo vector, [352](#)  
 variación de curvatura, [306, 308](#)  
 vector de fuerzas y pares, [245](#)  
 – de movimiento diferencial, [427](#)  
 – normal, [307](#)  
 – tangente unitario, [306](#)  
 vehículo con ruedas, [28](#)  
 – filoguiado, [195](#)  
 – robótico, [64, 97, 260](#)  
 – – con manipulador, [64, 113](#)  
 velocidad angular, [86](#)  
 – de aproximación, [384](#)  
 – de ejecución, [351](#)  
 – de rotación, [86](#)  
 – de translación, [85](#)  
 – lineal, [85](#)  
 visión en teleoperación, [414](#)  
 visualización de imágenes virtuales, [418](#)  
 – estéreo, [416, 417](#)  
 – predictiva, [409](#)  
 visualizador, [415](#)  
 – montado en casco, [416](#)  
 VME, [160-161](#)

**Z**

zoom, [414](#)

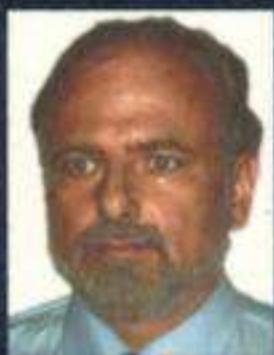




**Texto dedicado a los fundamentos de la robótica y a las tecnologías involucradas en su desarrollo. No se restringe al estudio de los manipuladores robóticos industriales con una base fija, como es habitual en la mayor parte de libros de robótica, sino que contempla también la robótica móvil y los vehículos autónomos.**

Incluye capítulos sobre morfología de los robots, modelos cinemáticos y dinámicos, arquitecturas de control, sensores, métodos de control, generación de trayectorias y programación. Asimismo, contiene capítulos que tratan de forma introductoria la detección de colisiones, planificación de caminos y telerrobótica. El libro incluye 43 ejemplos sobre manipuladores, robots móviles y técnicas de control y programación. Le acompaña un CD-ROM con el cual se suministra una herramienta MATLAB-Simulink que el lector podrá utilizar para realizar esos ejemplos, cambiar sus parámetros, o generar ejemplos nuevos involucrando diferentes robots y técnicas de control. Existe un sitio Web desde el que se podrán obtener versiones actualizadas de la herramienta a través de Internet. Este sitio podrá utilizarse también para que el lector ejecute ciertos ejemplos del libro y funciones de la herramienta sin necesidad de disponer de ningún software específico en su ordenador.

El libro puede emplearse en asignaturas de robótica de distintas titulaciones, así como en cursos para profesionales que deseen introducirse en la robótica o actualizar conocimientos con vistas a sus aplicaciones.



#### EL AUTOR

**Aníbal Ollero Baturone** es Catedrático de Ingeniería de Sistemas y Automática de la Universidad de Sevilla, Escuela Superior de Ingenieros, donde imparte asignaturas de Robótica y Sistemas de Percepción. Ha sido Catedrático de las Universidades de Málaga y Santiago de Compostela en Vigo, "research scientist" en el "Robotics Institute" de la Carnegie Mellon University (Pittsburgh, USA) y becario en el "Laboratoire d'Architecture et d'Analyse des Systèmes" del CNRS en Toulouse (Francia).

Ha participado en más de 50 proyectos y contratos de Investigación y Desarrollo sobre robótica, visión artificial, y sistemas inteligentes para aplicaciones de control, monitorización, fabricación y prueba, temas en los cuales ha realizado también trabajos, asesorías y cursos para organismos y empresas. Ha dirigido o participado en el diseño y construcción de robots y sistemas robóticos para aplicaciones en el espacio, agricultura, forestales, minería, protección del medio ambiente, inspección y fabricación industrial. Es autor de dos libros, editor o coeditor de otros seis, y autor o co-autor de más de 200 artículos en revistas, capítulos de libros y publicaciones de comunicaciones a congresos. Es "chairman" del "Coordinating Committee on Manufacturing and Instrumentation" de la "International Federation of Automatic Control" (IFAC). Ha ocupado diversos cargos académicos incluyendo los de Director de Escuela (Málaga), Subdirector de Escuela (Sevilla y Vigo), y Director de tres Departamentos Universitarios.

ISBN 84-267-1313-0



9 788426 713131