



Universidad
de La Laguna

Escuela de Doctorado
y Estudios de Posgrado

TÍTULO DE LA TESIS DOCTORAL

Control de un Robot Autónomo Móvil para la Recogida de objetos (Mejora del Rendimiento en el Control de Manipuladores Robóticos Mediante Combinación de Técnicas de Control Robusto y Predictivo)

AUTOR/A

SANTIAGO

TORRES

ALVAREZ

DIRECTOR/A

JUAN ALBINO

MENDEZ

PEREZ

CODIRECTOR/A

DEPARTAMENTO O INSTITUTO UNIVERSITARIO

FECHA DE LECTURA

07/03/08



Universidad
de La Laguna

FACULTAD DE FÍSICA

**DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA
Y ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES**

CONTROL DE UN ROBOT AUTÓNOMO MÓVIL PARA LA RECOGIDA DE OBJETOS

Autor: Santiago Torres Álvarez

D. JUAN ALBINO MÉNDEZ PÉREZ, Doctor en Informática y Profesor Titular de Ingeniería de Sistemas y Automática del Departamento de Ingeniería de Sistemas y Automática y Arquitectura y Tecnología de Computadores de la Universidad de La Laguna

CERTIFICA:

que D. Santiago Torres Álvarez, Licenciado en Física e Ingeniero en Electrónica, ha realizado bajo mi dirección la presente Tesis, titulada "Control de un robot autónomo móvil para la recogida de objetos", para optar al grado de Doctor por la Universidad de la Laguna.

Con esta fecha, autorizo la presentación de la misma.

La Laguna, a de de .

El Director,

Juan A. Méndez Pérez

"[...] Tu vara y tu cayado me sosiegan."

(Salmo 22, 4)

*A Sandra, mi mujer,
mi compañera, mi amiga,
mi razón de ser.*

*A mis padres y hermanos,
Santiago, Ignacia,
María Ignacia y Jorge.*

*A mi segunda familia,
Juan, María Jesús,
Juanjo, Ana, Saúl,
Mary Lux, Fran y Juan.*

Agradecimientos

En primer lugar quiero agradecer al Dr. D. Juan Albino Méndez Pérez su apoyo total, incondicional y constante durante toda la realización de este trabajo, el cual ha salido adelante gracias, principalmente, a su inestimable ayuda. Sus aportaciones, siempre ricas e innovadoras, han supuesto el espaldarazo definitivo para que finalmente haya salido a la luz esta tesis. Por todo ello le doy las gracias de todo corazón.

En segundo lugar quiero agradecer al Dr. D. Leopoldo Acosta Sánchez su acompañamiento durante todo mi periodo universitario y, en especial, en esta última fase como doctorando y docente. Quiero agradecerle su confianza en mí y en mis posibilidades como investigador, aparte de todo el cariño que me transmite.

Quiero agradecer al Dr. D. Lorenzo Moreno Ruiz la posibilidad que me brindó al abrirme las puertas de la carrera universitaria como docente ofreciéndome un puesto en el grupo de Computadoras y Control, hoy Departamento de Ingeniería de Sistemas y Automática y Arquitectura y Tecnología de Computadores.

También me gustaría dar las gracias, de una forma especial, al Dr. D. Victor M. Becerra, profesor de la Universidad de Reading. Fue él quien me dio la posibilidad de realizar una estancia como investigador en su universidad. Su acogida, seguimiento y confianza fueron todo un detalle que, tal vez, nunca podré agradecerle como merece.

Agradecer también a los miembros del Departamento de Ingeniería de Sistemas y Automática y Arquitectura y Tecnología de Computadores de la Universidad de La Laguna su ayuda y compañerismo: Dr. D. José Luis Sánchez De la Rosa, Dr. D. José Demetrio Piñeiro Vera, Dra. Rosa M^a Aguilar China, D. Juan Julián Merino Rubio, Dr. D. Alberto Francisco Hamilton Castro, Dr. D. Graciliano Nicolás Marichal Plasencia, Dr. D. José Ignacio Estévez Damas, Dr. D. José Francisco Sigut Saavedra, Dra. Dña. Marta Sigut Saavedra, Dr. D. Roberto Luis Marichal Plasencia, Dra. Dña. Carina Soledad González González, Dr. D. Evelio González González, Dra. Dña. Silvia Alayón Miranda, Dra. Dña. Vanessa Muñoz Cruz, D. Jonay Tomás Toledo Carrillo, D. Jesús Miguel Torres Jorge, D. Germán Carlos González Rodríguez, D. Carlos Alberto Martín Galán, D. Héctor Rebozo Morales, D. Ginés Coll Barbuzano, D. Eladio Hernández Díaz, D. Iván Castilla Rodríguez, D. Pedro Antonio Toledo Delgado, Dña. Hilda Mercedes Sanabria Medina, D. Jonatán Felipe García y D. Roberto Muñoz González.

Y no puedo terminar sin tener unas palabras de agradecimiento con mis seres más queridos.

Gracias, en primer lugar, a Sandra, que me regala día y a día todo su amor y su apoyo. Ella ha sido y es pieza fundamental para que esta tesis haya llegado a buen término. Aparte de contribuir en el formato del trabajo, es la persona que ha estado a mi lado en todo el proceso, alentándome y ayudándome en los malos momentos, en esos en los que más hace falta.

En segundo lugar, a mis padres, que me manifiestan siempre su amor y cariño, y a quienes les debo todo lo que soy. Gracias, porque en la vida también son importantes los ejemplos de nobleza, de seriedad, de solidaridad, de esfuerzo y de amor desmedido, y todo eso lo he aprendido de ellos.

En tercer lugar, a mis hermanos, por su cariño y su interés constante por mi trabajo. Particularmente a mi hermana, quien tuvo el detalle de leer toda la tesis y que aportó toda su experiencia como filóloga en los aspectos de redacción de la misma.

También quiero darle las gracias a mi segunda familia, la familia de mi mujer, que me ha aportado en todos estos años unas dosis extra de amor y aliento inigualables.

Y, por último, a mis amigos. Los amigos con mayúsculas. Aquellos, que pese a las circunstancias o las distancias, siempre están a tu lado. Muchas gracias a todos por su interés y su cariño.

Introducción

Esta tesis lleva por título general *Control de un robot autónomo móvil para la recogida de objetos*. Esto se debe a que, en sus inicios, el resultado de esta investigación nació de un proyecto denominado *Diseño e implementación hardware y software de un robot autónomo inteligente para la recogida de objetos. Aplicación a la recogida de fruta* [PI1999/140], subvencionado por el Gobierno de Canarias. Este proyecto perseguía la implementación de un robot móvil autónomo, compuesto por una estructura móvil para su movimiento libre a través de espacios no estructurados, y de un manipulador en su parte superior para efectuar las tareas de recogida de objetos (frutas, en una primera aplicación, y enseres de restauración –platos, cubiertos, etc.- en una segunda y posterior aplicación igualmente subvencionada en un nuevo proyecto). El objetivo de esta parte del proyecto fue el desarrollo de técnicas de control para el seguimiento efectivo de trayectorias por parte del manipulador.

Por esta razón y bajo el auspicio del primer proyecto de investigación se presentó el proyecto de tesis inicial con el título antes indicado. Con el objeto de mantener el proyecto de tesis inicial inalterado, no se ha cambiado su título, aunque realmente la investigación realizada se centra en el desarrollo de estrategias de control robustas, adaptativas y predictivas para el seguimiento de trayectorias en robots afectados por perturbaciones, incertidumbres y ligaduras. De ahí el subtítulo elegido para esta tesis que engloba los objetivos conseguidos en la misma:

“Mejora del rendimiento en el control de manipuladores robóticos mediante combinación de técnicas de control robusto y predictivo”.

Las técnicas novedosas que aquí se proponen son una mejora de técnicas clásicas de control robusto, adaptativo y predictivo que han sido ampliamente utilizadas en el campo de la Robótica. Ya en la década de los 80, numerosas técnicas de control adaptativo y control robusto fueron propuestas para conferir al robot un seguimiento de trayectoria eficiente. Problemas como la existencia de incertidumbres, rechazo a perturbaciones, variación de condiciones dinámicas de la estructura, ligaduras en las entradas o en el movimiento del manipulador, etc. fueron abordados con este tipo de técnicas. Además, fueron estudiadas las condiciones de estabilidad del sistema completo (controlador + robot) para estos controladores.

Sin embargo, no hay en la literatura trabajos que aporten una solución global a todos los problemas que luego se presentan en la práctica y que han sido mencionados con anterioridad. El problema del control de manipuladores

sigue siendo, pues, un tema de actualidad y que sigue generando técnicas y desarrollos entre los investigadores de la materia: [Alo07],[Fer07],[Has07], etc.

La Robótica es una disciplina de la Ingeniería que ha experimentado un gran avance en las últimas décadas. Dada la complejidad de funcionamiento de los robots, se deben emplear conocimientos de una amplia diversidad de campos en aras de conseguir una aplicación satisfactoria de los robots en las tareas a las que son encomendados. Así, se requieren conocimientos de mecánica, electrotecnia, automática, sistemas, programación, etc. para dar solución a los problemas que se presentan en la Robótica: cinemática, dinámica, control, programación, visión artificial, agarre y manipulación, planificación, etc.

En esta tesis se desarrollan técnicas de control para robots industriales o manipuladores, los cuales han sido diseñados para tareas esencialmente de manipulación: soldadura, manejo de material de manufactura, pintado, etc. El otro gran campo de aplicación de la Robótica se centra en los robots móviles, de dinámica más simple, y cuyo principal objetivo es la ejecución de movimientos y su localización en el espacio, en interacción constante con un entorno amplio y muy variable.

Por tanto, los objetivos que se plantean en este trabajo son:

- ✓ el desarrollo de técnicas de control eficientes para el seguimiento satisfactorio de trayectorias en el espacio de trabajo de un manipulador,
- ✓ proporcionar una mejora en los diseños de control que tienen en cuenta los aspectos prácticos que pueden alterar el comportamiento del sistema respecto a su modelo teórico equivalente, como son las incertidumbres presentes en el sistema y las perturbaciones en el movimiento del mismo,
- ✓ dar una solución eficiente al problema de la no violación de las ligaduras a las que está sometido el robot,
- ✓ diseñar herramientas de control con un bajo coste computacional para que puedan ser implementadas en cualquier robot real con un controlador digital lo suficientemente rápido.

Haciendo un repaso por la literatura relativa al control de manipuladores, se puede considerar el seguimiento de trayectoria como el principal problema que tienen que resolver las herramientas de control, las cuales deben proveer velocidades de movimiento lo suficientemente elevadas y gran precisión de ejecución de trayectorias.

Los primeros controladores ya acometieron este problema con soluciones que pasaban desde el control PD y PID monoarticular [Luh83] hasta el control con compensación [Kho88] o el control de par calculado [Mid88],

[Kre89]. El control PD y PID monoarticular considera que el modelo teórico del sistema es aproximadamente lineal en un margen más o menos cercano al punto de operación del robot y no tiene en cuenta los acoplamientos no lineales que se producen entre los distintos elementos que componen la estructura del robot. Por esta razón un controlador de estas características sólo es adecuado en el caso de velocidades de movimiento lentas y con requerimientos de precisión no muy elevados.

Sin embargo, la dinámica del sistema varía en rangos de operación diferentes al punto de operación actual, con masas de cargas diferentes, además de con la velocidad de movimiento de la estructura. Entonces el modelo lineal supuesto para el robot deja de ser válido si no se compensan las no linealidades inherentes a la estructura compleja y acoplada del manipulador. Por esta razón surgieron las otras dos estrategias, control compensado y de par calculado, que tratan de eliminar las no linealidades del modelo y conseguir un sistema linealizado que permita un mejor recorrido de las trayectorias en diferentes puntos de operación.

Estos primeros planteamientos de control tienen varios inconvenientes que se irán exponiendo a continuación.

El primer problema es el rechazo a perturbaciones. Las técnicas habituales de control realimentado ofrecen un rechazo a perturbaciones limitado dado por el propio bucle de realimentación. Esto es debido a que, para reducir apreciablemente los errores de seguimiento en presencia de perturbaciones, se deben escoger ganancias de realimentación elevadas, con lo que los requerimientos de ancho de banda del sistema se ven aumentados. Sin embargo, no se pueden conseguir anchos de banda elevados debido a las limitaciones en el diseño mecánico de los robots.

Para paliar este efecto, se han diseñado estrategias de control basadas en control robusto. Este tipo de controladores tienen una estructura fija, a diferencia de los controladores adaptativos que se presentan más adelante, y su acción de control trata de minimizar las perturbaciones que se puedan dar en el manipulador. Tienen como ventaja su fácil implementación y que, en principio, no precisan una sintonización previa de sus parámetros para su funcionamiento. Sin embargo, el rango de perturbaciones para los que presenta un buen comportamiento es bastante limitado.

El segundo problema que se presenta para un seguimiento eficaz de la trayectoria es la presencia de incertidumbres. Estas incertidumbres se dan por la inexactitud en el conocimiento del modelo dinámico del sistema, ya sea por el desconocimiento de los parámetros del mismo (masas, longitudes, inercias, etc.) o por el desconocimiento de las masas de carga que manipula (su peso, tamaño o la distribución de masa de la misma). Las técnicas de control robusto mencionadas también son aplicables para resolver este problema. La acción robusta que se añade a la acción de control clásica (PID, compensación, par calculado) trata de compensar la desviación del modelo estimado del

manipulador respecto al sistema real debido a la existencia de estas incertidumbres.

Sin embargo, tal y como se demuestra en esta tesis, esta acción robusta no es suficiente cuando el rango de incertidumbre se desconoce. En la literatura existen soluciones ([Fre82], [Spo87a], [Tar87], [Slo88], [Fu90], [Su93], etc.) para determinar la ley de control robusta en el caso de incertidumbres acotadas, pero no existen soluciones eficientes para el caso de variabilidad en las condiciones de carga o no acotación de las incertidumbres paramétricas de la dinámica del manipulador.

Otra forma de atacar el problema de las incertidumbres es mediante el uso de técnicas adaptativas. Este tipo de estrategias se basan en la modificación de un controlador ajustable de acuerdo a un criterio de adaptación basado en el comportamiento del robot. La gran mayoría de estos controladores se basan en la propiedad de parametrización lineal de la dinámica de los manipuladores ([Ara93], [Whi93], [Spo93], [Liu94], [Koo94], [Jar96], [Sic96], etc). El rechazo a perturbaciones, ya sean debidas a incertidumbres o a otras causas, es más efectivo con este tipo de controladores. Sin embargo, precisan de una sintonización previa que dificulta su aplicación práctica.

El tercer problema que está presente en el control de manipuladores son las ligaduras a las que está sujeto. En la práctica, las entradas a los actuadores están acotadas, en el caso más común, por los valores máximos de tensión que permiten los motores que mueven las articulaciones. Además, las articulaciones están restringidas a un rango de actuación en cuanto a su posición y velocidad. Si se aumentan las ganancias de realimentación para proveer al sistema de un buen rechazo a perturbaciones, se estará en mayor disposición de violar las ligaduras del sistema. Por otro lado, si se violan las ligaduras en el estado del sistema (posiciones o velocidades de las distintas articulaciones), el comportamiento de los controladores es impredecible, pues éstas no se tienen en cuenta en el modelo dinámico estimado.

En este trabajo se presenta una técnica basada en control predictivo para evitar la violación de las ligaduras del sistema, con lo que se aumenta la eficacia del controlador y la precisión en el seguimiento de las trayectorias de consigna.

Las técnicas que aquí se presentan, además, combinan las soluciones que aportan las distintas estrategias, de forma que finalmente se obtiene una técnica robusta (para acometer el rechazo a perturbaciones e incertidumbres), autoadaptativa (para acometer la variabilidad de las condiciones dinámicas del manipulador) y eficiente en el cumplimiento de las ligaduras del sistema. Además, el costo computacional de las mismas no es elevado. Éste es un problema importante que deben acometer los distintos controladores, debido a los reducidos períodos de muestreo que exige la complejidad dinámica del sistema. Se deben diseñar estrategias lo suficientemente rápidas como para ser acometidas en tiempos inferiores a la centésima de segundo, como norma

general. Sólo en ese caso, las aproximaciones discretas del modelo continuo del manipulador son lo suficientemente exactas como para llevar a cabo con éxito la implementación práctica de los mismos utilizando control digital.

Una manera de reducir el coste computacional de las estrategias de control es reemplazando los modelos matemáticos del manipulador por modelos empíricos, de más fácil evaluación. Los modelos matemáticos del manipulador son ecuaciones altamente no lineales y acopladas, con una gran cantidad de funciones sinusoidales que se deben evaluar, lo que conlleva realizar un alto número de operaciones. Una red neuronal es una herramienta matemática que puede aprender la dinámica del sistema, de forma que puede reemplazar al modelo matemático para su uso en los controladores propuestos. Para ello, es importante que la red identifique con gran precisión el comportamiento dinámico del manipulador, en sus diferentes situaciones de carga y punto de operación. Una ventaja importante de obtener un modelo empírico, basado en medidas del propio sistema, es que dicho modelo ya incluye las incertidumbres que presenta el modelo matemático al que reemplaza, por lo que las estrategias de rechazo a perturbaciones serán menos exigentes.

En esta tesis se propone la identificación de un sistema robótico mediante el uso de redes neuronales dinámicas, presentando un esquema de identificación y de validación que asegura un buen comportamiento de la red neuronal. Este método está avalado por pruebas reales sobre un robot. Además, el modelo neuronal así obtenido se utiliza en los algoritmos de control propuestos, para rebajar la carga computacional de los mismos. Por ejemplo, en los algoritmos predictivos se emplea el modelo para realizar las predicciones que se hagan en cada etapa.

Recapitulando todo lo dicho anteriormente, y para un mejor entendimiento de las distintas estrategias propuestas en este trabajo, la memoria de tesis se presenta dividida en tres partes fundamentales, cada una, a su vez, compuesta de capítulos. Estas tres partes se refieren a los tres resultados principales presentados en la tesis: estrategias de control robusto adaptativo para el rechazo de perturbaciones, estrategias de control predictivo para abordar el problema de ligaduras y métodos eficientes de identificación de la dinámica de manipuladores basados en redes neuronales dinámicas, así como su aplicación en las estrategias de control anteriormente propuestas para mejorar su eficiencia computacional.

En el primer capítulo se introduce el problema de control en manipuladores y la descripción de las técnicas de control habituales para acometer los mismos, realizando una amplia descripción del estado actual de la investigación en este tema.

A continuación se expone la primera parte de la memoria, dedicada al control robusto de manipuladores. En la misma se diseña una nueva técnica robusta autoadaptativa (ARC) que mejora las prestaciones de los controladores robustos clásicos. Se presentan resultados obtenidos en un manipulador PUMA

real que muestran las ventajas de esta técnica respecto a otras técnicas de control robusto.

En la segunda parte de la memoria se presentan estrategias basadas en control predictivo para la inclusión de las ligaduras del sistema en las estrategias de control. Las técnicas que se presentan, IAPC y RIAPC, están basadas en el control predictivo mediante interpolación, con lo cual dan como resultado técnicas con un costo computacional bajo y de alta eficiencia en el cumplimiento de las ligaduras presentes en el sistema.

En la tercera y última parte de la memoria se expone un método para la identificación de un sistema robótico basado en el entrenamiento de redes neuronales dinámicas, a la que se aplica una técnica novedosa para la inicialización de las neuronas que asegura una mejor identificación del sistema. El modelo de identificación obtenido se emplea en los controladores anteriores para mejorar la eficiencia computacional de los mismos, reduciendo el tiempo que se emplea en realizar las predicciones por parte de los algoritmos IAPC y RIAPC.

Finalmente se exponen las conclusiones del trabajo y las principales líneas abiertas de la tesis.

Índice de contenidos

CAPÍTULO 1

EL PROBLEMA DEL CONTROL DE MANIPULADORES	1
1.1 REVISIÓN DE CONCEPTOS BÁSICOS DE ROBÓTICA DE MANIPULACIÓN	1
1.1.1 ROBOTS INDUSTRIALES	2
1.1.1.1 Definición de robot industrial	2
1.1.1.2 Aplicaciones de los robots industriales	3
1.1.2 CINEMÁTICA	5
1.1.3 DINÁMICA	7
1.1.3.1 Ecuaciones dinámicas de un robot manipulador	8
1.1.3.2 Manipulador de dos grados de libertad	9
1.1.3.3 Manipulador PUMA	10
1.1.4 PLANIFICACIÓN DE TRAYECTORIAS.	11
1.2 PROBLEMAS EN EL CONTROL DE MANIPULADORES	12
1.2.1 PROBLEMAS CLÁSICOS DEL CONTROL DE MANIPULADORES	12
1.2.2 PROBLEMAS ABIERTOS EN EL CONTROL DE MANIPULADORES	13
1.2.2.1 Problemas abiertos en el control de fuerza	14
1.2.2.2 Robots cooperativos	15
1.2.2.3 Manos robóticas de múltiples dedos	15
1.2.2.4 Sistemas subactuados	15
1.2.2.5 Teleoperación	16
1.2.2.6 Visión por computador	16
1.3 ELECCIÓN DE LA ESTRUCTURA DE CONTROL	17
1.3.1 INFLUENCIA DE LAS REDUCTORAS	17
1.3.2 VENTAJAS E INCONVENIENTES DEL CONTROL MONOARTICULAR	18
1.4 CONTROL DE POSICIÓN	18
1.4.1 CONTROL DE POSICIÓN EN EL CASO MONOARTICULAR	19
1.4.1.1 Control PID	20
1.4.1.2 Control I-PD	20
1.4.2 CONTROL DE POSICIÓN EN EL CASO MULTIARTICULAR	21
1.4.2.1 Desacoplamiento por inversión del modelo	22

1.4.2.2 Algoritmos de control en el caso multiarticular	22
1.5 CONTROL DE MOVIMIENTO	23
1.5.1 CONTROL PID CON PREALIMENTACIÓN	23
1.5.2 CONTROL PD CON COMPENSACIÓN DE GRAVEDAD	24
1.5.3 CONTROL PD CON COMPENSACIÓN DE GRAVEDAD PRECALCULADA	25
1.6 INCERTIDUMBRES EN EL MODELO	26
1.6.1 CONTROL ADAPTATIVO	26
1.6.1.1 Control adaptativo por planificación de ganancias	26
1.6.1.2 Control adaptativo por modelo de referencia	27
1.6.1.3 Control adaptativo de par calculado	28
1.6.1.4 Otras técnicas de control adaptativo	28
1.6.2 CONTROL ROBUSTO	31
1.6.2.1 Control robusto basado en linealización por realimentación	32
1.6.2.2 Control robusto basado en la propiedad de pasividad	34
1.6.2.3 Control de estructura variable	35
1.6.2.4 Control robusto por saturación	36
1.7 COTAS DE INCERTIDUMBRE	37
1.8 VARIABILIDAD EN LOS PARÁMETROS DINÁMICOS Y MASAS DE TRANSPORTE	38
1.9 RESTRICCIONES DE MOVIMIENTO DE LAS ARTICULACIONES	38
1.10 ALGORITMOS DE BAJO COSTO COMPUTACIONAL	39
1.11 PORTABILIDAD DE LOS ALGORITMOS HACIA OTRO TIPO DE ROBOTS	39
1.12 CONCLUSIONES	40

PARTE I: RECHAZO A PERTURBACIONES EN ROBOTS MANIPULADORES

CAPÍTULO 2

TÉCNICAS DE CONTROL ROBUSTO PARA EL RECHAZO DE PERTURBACIONES	43
2.1 INTRODUCCIÓN	43
2.2 LINEALIZACIÓN POR REALIMENTACIÓN.	46
2.3 LEY DE CONTROL.	47

2.3.1 LEY DE CONTROL PD ESTABILIZANTE	47
2.3.2 ACCIÓN ROBUSTA	48
2.3.2.1 Aplicación del método directo de Lyapunov para la obtención del término de acción robusta	49
2.3.2.2 Consideraciones prácticas del término de acción robusta.	52
2.4 ELECCIÓN DEL PARÁMETRO DE COTA DE INCERTIDUMBRE	53
2.4.1 ALGORITMO DE SPONG	55
2.4.2 OTROS CONTROLADORES ROBUSTOS	57

CAPÍTULO 3

CONTROLADOR ROBUSTO ADAPTATIVO (ARC)	59
3.1 INTRODUCCIÓN	59
3.2 LEY AUTOADAPTATIVA PARA EL PARÁMETRO DE COTA DE INCERTIDUMBRE	61
3.2.1 ALGORITMO DEL CONTROLADOR ROBUSTO AUTO-ADAPTATIVO PROPUESTO	64
3.3 CONSIDERACIONES SOBRE ESTABILIDAD	64
3.3.1 DINÁMICA DEL ERROR	65
3.3.2 ESTABILIDAD DEL SISTEMA EN LAZO CERRADO	66
3.3.2.1 Discretización de la ecuación dinámica del error	66
3.3.2.2 Aplicación de la teoría de estabilidad de Lyapunov	66
3.4 DESCRIPCIÓN DEL ROBOT MANIPULADOR	71
3.5 ESTRATEGIAS DE CONTROL IMPLEMENTADAS	73
3.6 RESULTADOS DE CONTROL	74

PARTE II: INCLUSIÓN DE LIGADURAS MEDIANTE TÉCNICAS COMPUTACIONALMENTE EFICIENTES

CAPÍTULO 4

ALGORITMOS PREDICTIVOS PARA LA MEJORA DE PRESTACIONES	89
4.1 INTRODUCCIÓN AL CONTROL PREDICTIVO	89

4.2 CONTROL PREDICTIVO BASADO EN MODELO, MPC	91
4.2.1 IDEAS BÁSICAS DEL CONTROLADOR MPC	91
4.2.2 ESQUEMA DE CONTROL DEL MPC	92
4.2.3 ALGORITMO DEL CONTROLADOR MPC	93
4.2.4 ESTABILIDAD EN EL MPC	94
4.2.5 MODELO PREDICTIVO DEL MPC	96
4.2.5.1 Modelos de predicción	97
4.2.5.2 Modelo de perturbaciones	99
4.2.6 LIGADURAS EN EL MPC	102
4.2.7 ELECCIÓN DE LA FUNCIÓN DE COSTO EN MPC	103
4.2.7.1 Posibilidades de formulación de la función de costo	104
4.2.7.2 Problema de optimización de tiempo infinito sin restricciones	105
4.2.7.3 Problema de optimización de tiempo finito sin restricciones	106
4.2.7.4 Problema de optimización de tiempo finito con restricciones	106
4.2.7.5 Problema de optimización de tiempo infinito con restricciones	108
4.3 CONTROL PREDICTIVO CON INTERPOLACIÓN	109
4.3.1 CONTROL LQ O LINEAL CUADRÁTICO.	110
4.3.2 CONTROL ML O DE NIVEL MEDIO	111
4.3.3 INTERPOLACIÓN LQ+ML	111
4.3.4 OBTENCIÓN DE LA COLA EN EL CONTROL PREDICTIVO	112
4.3.5 INTERPOLACIÓN LQ+ML+COLA	113
4.4 PRINCIPALES DIFICULTADES DEL MPC	114
4.4.1 FACTIBILIDAD DE LA SOLUCIÓN EN MPC	115
4.4.2 COSTE COMPUTACIONAL	116
4.4.2.1 Algoritmos de optimización en MPC	116
4.4.2.2 Algoritmos de optimización en MPC de bajo coste computacional	117

CAPÍTULO 5

CONTROL DE PERTURBACIÓN ADAPTATIVA BASADO EN INTERPOLACIÓN (IAPC)	121
5.1 INTRODUCCIÓN	121

5.2 ESQUEMA DE CONTROL IAPC	123
5.2.1 OBTENCIÓN DEL MODELO MEDIANTE <i>FEEDBACK LINEARISATION</i>	123
5.2.2 LINEALIZACIÓN EXACTA	126
5.3 MODELO DE PREDICCIONES	127
5.4 PROBLEMA DE OPTIMIZACIÓN	128
5.5 RESULTADOS CON EL CONTROLADOR IAPC	131
5.5.1 MODELO SIN INCERTIDUMBRES	132
5.5.2 MODELO CON INCERTIDUMBRES	135
5.6 AÑADIENDO ROBUSTEZ PARA RECHAZO A PERTURBACIONES: CONTROLADOR RIAPC	138
5.6.1 ESQUEMA DE CONTROL RIAPC	138
5.6.2 RESULTADOS CON EL CONTROLADOR RIAPC	139

PARTE III: MEJORA DEL COSTO COMPUTACIONAL DEL LOS CONTROLADORES USANDO MODELOS NEURONALES

CAPÍTULO 6

TÉCNICAS BASADAS EN REDES NEURONALES PARA LA IDENTIFICACIÓN	147
6.1 IDENTIFICACIÓN DE LA DINÁMICA DEL MANIPULADOR	147
6.1.1 IDENTIFICACIÓN MEDIANTE REDES NEURONALES	148
6.1.2 MEJORAS EN LA IDENTIFICACIÓN: INICIALIZACIÓN DE LA DNN	149
6.2 REDES NEURONALES DINÁMICAS	150
6.3 ENTRENAMIENTO DE LA DNN	152
6.4 INICIALIZACIÓN DE LA DNN	153
6.5 EJEMPLO DE IDENTIFICACIÓN DE UN SISTEMA ROBÓTICO	154
6.5.1 DESCRIPCIÓN DEL SISTEMA	154
6.5.2 RESULTADOS DE IDENTIFICACIÓN	155

CAPÍTULO 7

REDUCCIÓN DEL COSTO COMPUTACIONAL DE LOS CONTROLADORES USANDO DNN	161
7.1 INTRODUCCIÓN	161
7.2 IDENTIFICACIÓN DEL MANIPULADOR PUMA 560	162
7.3 INCORPORACIÓN DEL MODELO NEURONAL AL CONTROLADOR RIAPC: ESTRATEGIA NRIAPC.	166
7.4 RESULTADOS CON EL CONTROLADOR NRIAPC	167
<u>ANEXO A: REPRESENTACIÓN DE DENVAIT-HARTENBERG</u>	<u>179</u>
<u>ANEXO B: CINEMÁTICA DIRECTA</u>	<u>183</u>
<u>ANEXO C: CINEMÁTICA INVERSA</u>	<u>185</u>
<u>ANEXO D: FORMULACIÓN DE LAGRANGE-EULER</u>	<u>191</u>
<u>ANEXO E: FORMULACIÓN DE NEWTON-EULER</u>	<u>195</u>
<u>ANEXO F: MOMENTOS DE INERCIA</u>	<u>199</u>
<u>ANEXO G: LINEALIZACIÓN EXACTA</u>	<u>201</u>
<u>BIBLIOGRAFÍA</u>	<u>209</u>

El Problema del Control de Manipuladores

El problema de control de robots manipuladores ha sido un t3pico ampliamente estudiado en la historia reciente de la rob3tica. Ya desde la d3cada de los ochenta fue investigado en profundidad, dise1andose estrategias que consiguen un buen comportamiento de los manipuladores para solucionar problemas de regulaci3n y de seguimiento de trayectorias. Las primeras estrategias dise1adas asumen un conocimiento exacto de la din3mica del manipulador y las condiciones de trabajo. Posteriormente, surgen otras estrategias de control, basadas en t3cnicas de control robusto y adaptativo, que incorporan las incertidumbres presentes en el modelo estimado del sistema real. Sin embargo, estas t3cnicas no son f3cilmente implementables en un manipulador real, o requieren de una sintonizaci3n previa mediante un estudio en simulaci3n. Con todo, el problema del control de manipuladores en presencia de incertidumbres no est3 completamente resuelto. Por otro lado, la exigencia cada vez mayor de optimizar prestaciones y la presencia de nuevas estructuras de manipulaci3n flexibles y de bajo peso, obligan a introducir algoritmos de control m3s avanzados para garantizar un rendimiento adecuado. En este cap3tulo se hace una breve revisi3n de los conceptos b3sicos de la rob3tica de manipulaci3n, se definen las estrategias b3sicas de control y se describen los problemas abiertos que surgen de la definici3n de estos controladores.

1.1 Revisi3n de conceptos b3sicos de rob3tica de manipulaci3n

La Rob3tica, tal y como la conocemos hoy en d3a, no es simplemente el estudio mec3nico-din3mico de los robots, sino que comprende un conjunto de 3reas interdisciplinarias tales como la cinem3tica, la din3mica, el control, los sensores y actuadores, la planificaci3n de sistemas, los lenguajes de programaci3n y la inteligencia artificial.

En esta primera sección, veremos una breve revisión sobre conceptos básicos de robótica de manipulación. Para una revisión completa de los mismos, se puede consultar bibliografía básica como [Pau81], [Cra86], [Fu88], [Spo89], [Gro89], [Ang89]. Títulos más actualizados son [Bar97], [Ark98], [Sci00], [Oll01], [Tor02] y [Kel03].

Dentro de esta breve descripción, veremos en primer lugar las definiciones de robot y robot industrial, así como sus principales aplicaciones. Luego se introducen someramente aspectos como la cinemática, la dinámica y la planificación de trayectorias. Esta parte es de especial interés para la robótica, y para este trabajo en particular, pues permite la descripción posterior de las técnicas de control en manipuladores.

1.1.1 Robots industriales

La palabra robot proviene del término checo *robota*, que significa *trabajo*, ya que, originalmente, los robots se idearon como máquinas que sustituirían al hombre en el desempeño de determinadas tareas. Una definición más acorde con lo que hoy se conoce como robot es la que proporciona el RIA (*Robot Institute of America*): “Un robot es cualquier estructura mecánica que opera con un cierto grado de autonomía, bajo el control de un computador, y que dispone de un sistema sensorial más o menos evolucionado para obtener información de su entorno”.

A continuación se introduce una definición de robot industrial, se describen algunos ejemplos de robots habituales en la industria y se detallan algunas de sus aplicaciones principales.

1.1.1.1 Definición de robot industrial

Los *robots industriales* son robots autorregulados, programables por aprendizaje o por lenguaje simbólico, el cual incorpora instrucciones para el manejo de la información sensorial. En una pequeña proporción, pero cada vez de forma más frecuente, estos robots incorporan técnicas de inteligencia artificial para decidir su modo de actuación a través de su capacidad sensorial y sus algoritmos de reconocimiento.

Uno de los robots industriales más empleados es el manipulador PUMA (*Programmable Universal Machine for Assembly*), el cual se puede observar en la figura 1-1.

Originariamente, los robots industriales fueron concebidos para realizar tareas que normalmente harían los humanos, y poder insertarlos en cadenas de producción industrial, con la consiguiente ganancia en rapidez y precisión en la ejecución de dichas tareas. Así, un robot industrial se define como “un manipulador reprogramable y multifuncional, diseñado para mover materiales, piezas o dispositivos especializados, a través de movimientos programados variables para la realización de una diversidad de tareas”.

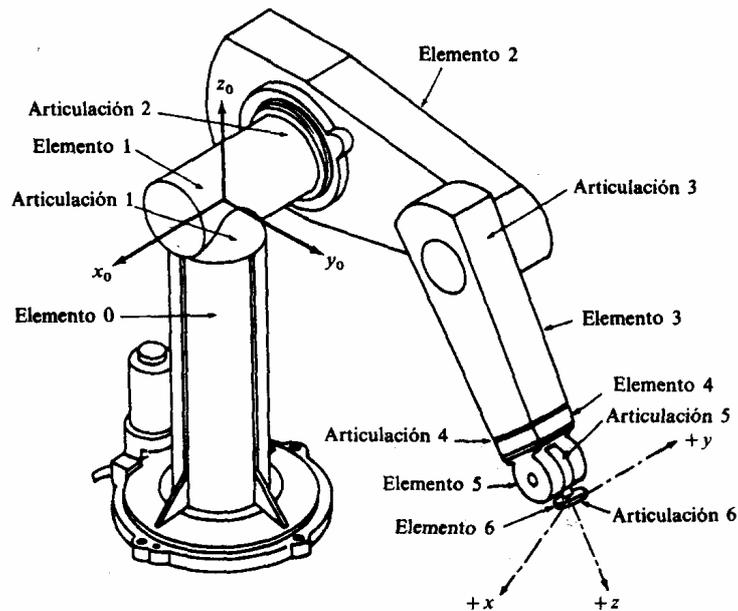


Figura 1-1: Representación del robot PUMA-560, donde se enumeran todas sus articulaciones y elementos, así como los sistemas de referencia de la base y del efector final. Fuente: [Fu88].

Este tipo de robots consiste en cadenas poliarticuladas de elementos rígidos conectados en serie, estando el inicio de la cadena fijo a una base o soporte. El otro extremo es el que se emplea para alcanzar una posición determinada dentro de su región de trabajo. Es por esa razón que se introduce, al final de la cadena, una *herramienta* con la que es capaz de manipular objetos o realizar tareas de ensamblaje.

Mecánicamente, un robot industrial se compone de la *base*, el *brazo* y la *muñeca* (o herramienta). El brazo se puede mover habitualmente con tres grados de libertad. Para facilitar la recogida de un elemento por parte de la muñeca, ésta generalmente posee tres movimientos giratorios, lo que confiere a la estructura total seis grados de libertad. El brazo es el encargado de posicionar la muñeca, y la muñeca es la encargada de la orientación de la estructura completa. La muñeca puede ser intercambiada con facilidad por otra de distinto tipo, sin alterar la estructura del brazo, lo que confiere multifuncionalidad al robot.

1.1.1.2 Aplicaciones de los robots industriales

Las aplicaciones de los robots industriales en la actualidad corresponden a un amplio conjunto de sectores, y cada vez el nivel de introducción de robots en tareas industriales es mayor, como se muestra en la figura 1-2. Este crecimiento es desigual por continentes, tal y como se observa en la figura 1-3, donde se aprecia un crecimiento continuo en Europa y América, y una estabilización en Japón, país que posee una densidad altísima de robots en referencia a la población o número de industrias.

Los sectores que cuentan con un mayor nivel de robotización son los correspondientes a la manufactura de productos: automóvil, textil, siderurgia, etc. Sin embargo, existen importantes sectores productivos que cuentan con un bajo nivel de automatización, como son el de servicios, construcción, alimentación, agricultura y otros, que ofrecen enormes posibilidades de expansión.

Robots Industriales OPERATIVOS

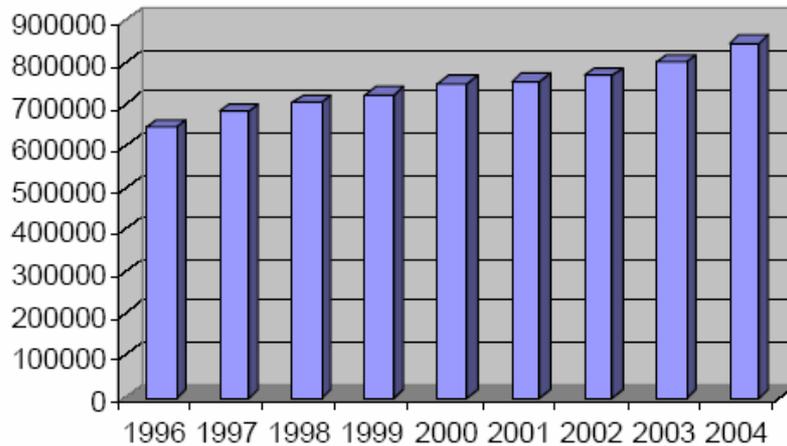


Figura 1-2: Implantación de robots industriales a nivel mundial en los últimos nueve años. Fuente: Libro Blanco de Robótica.

Otra de las aplicaciones de los robots en los últimos tiempos es en la medicina, donde se emplean como máquinas auxiliares para superar las limitaciones físicas de personas con algún tipo de discapacidad; o en telemedicina, en la realización de operaciones quirúrgicas a distancia. La industria farmacéutica es otro novedoso campo de aplicación donde la robótica ofrece un excelente compromiso entre productividad y flexibilidad, además de una calidad uniforme de los productos.

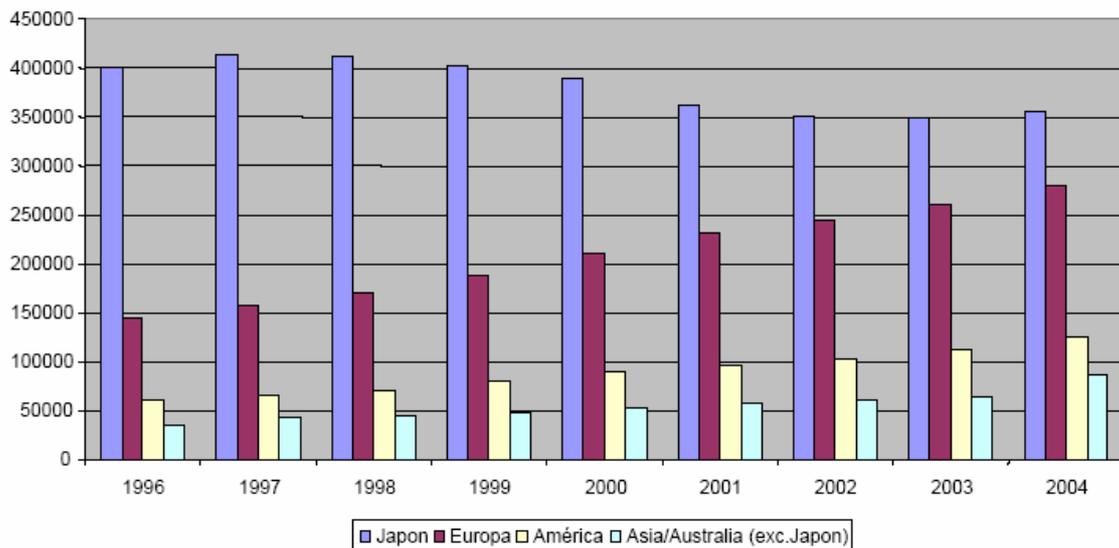


Figura 1-3: Número de robots industriales instalados por continentes en los últimos nueve años. Fuente: Libro Blanco de Robótica.

Uno de los principales puntos de introducción de los robots en las industrias ha sido su inserción en sistemas de fabricación flexible. Este tipo de sistemas corresponde a un grupo de máquinas-herramientas de control numérico, enlazadas entre sí mediante un sistema de transporte de piezas común y un sistema de control centralizado. Su objetivo es la fabricación de piezas, para cada una de las cuales se dispone de programas de piezas comprobados y memorizados en una estación de datos central, de manera que el proceso de fabricación tiene lugar de modo automático. Estos sistemas reducen la intervención del personal a misiones de vigilancia. Los robots manipuladores han sido incluidos con éxito en este tipo de sistemas, y su papel principal es el transporte, manejo de materiales, maquinado, carga y descarga, etc. Gracias a la eficiencia en el comportamiento de los manipuladores, estos sistemas ven incrementadas sus características de automatización, productividad, calidad del producto y fiabilidad del proceso.

1.1.2 Cinemática

El estudio de la cinemática del robot, así como de su dinámica, es fundamental en los pasos posteriores que se den dentro del campo de la robótica. El objetivo de estas dos disciplinas es la obtención de ecuaciones matemáticas que expresen, por un lado, la posición y orientación finales del robot dada una configuración determinada para las articulaciones; y por otro lado, el movimiento del robot ante la aplicación de unas determinadas fuerzas/momentos. Dichas ecuaciones son las que se utilizarán posteriormente en los algoritmos de control aplicados al robot.

La cinemática trata el estudio del movimiento del robot sin atender a las causas (fuerzas y momentos) que originan ese movimiento. Se trata, pues, de un análisis puramente geométrico del sistema.

Un robot se puede modelar como una estructura mecánica formada por una serie concatenada de cuerpos rígidos, que llamaremos *elementos*, los cuales se unen mediante articulaciones. Dichas articulaciones pueden ser de *revolución*, cuando el movimiento de un elemento respecto al que le antecede es de rotación, o *prismáticas*, cuando ese movimiento es un desplazamiento. Uno de los extremos de la cadena está unido a una base o soporte, mientras que el otro está libre y unido a una herramienta o muñeca. Ésta puede a su vez ser descompuesta en una o varias cadenas poliarticuladas.

En la figura 1-4 se pueden observar distintos tipos de robots industriales, dependiendo de la combinación de articulaciones con las que se diseñan. Ejemplos típicos de robots de revolución son el PUMA de Unimation, con seis articulaciones de revolución, y el T3 de Cincinnati Milacron. El robot Stanford es del tipo esférico, con dos articulaciones de revolución, perpendiculares entre sí, y una prismática. El robot SCARA se asemeja a un robot cilíndrico, salvo que posee dos articulaciones de revolución en un plano (en vez de una) y una articulación prismática normal a ese plano. Un esquema del robot PUMA y del SCARA puede verse en la figura 1-5.

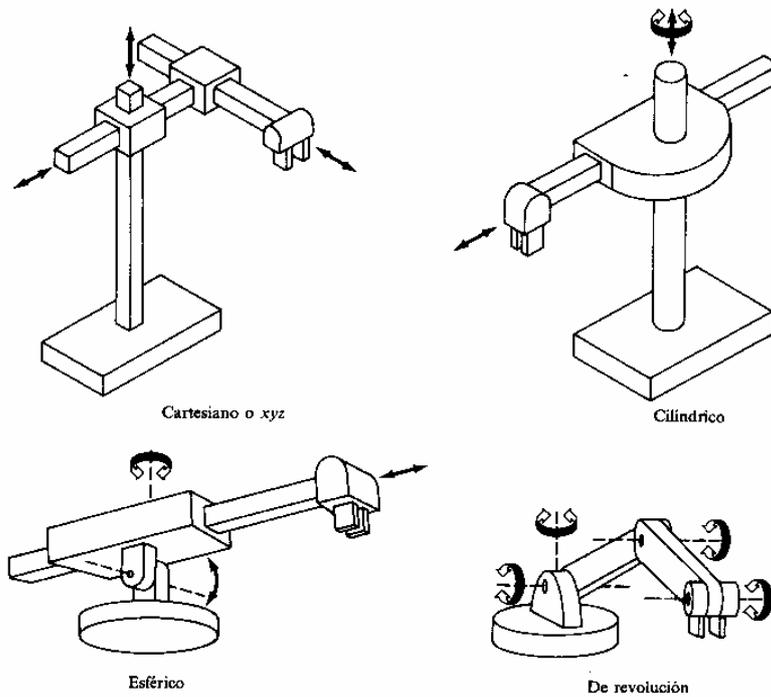


Figura 1-4 : Tipos de robots industriales, dependiendo de la combinación de articulaciones elegida en el diseño de los mismos.
Fuente: [Fu88].

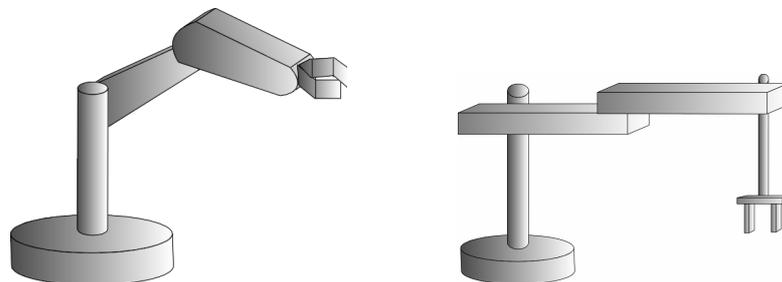


Figura 1-5 : Representación esquemática de los robots PUMA y SCARA.

Las coordenadas homogéneas o generalizadas incluyen los ángulos (en el caso de articulaciones de revolución) o distancias (en el caso de articulaciones prismáticas) que caracterizan la configuración de cada articulación. Cualquier configuración del robot se especifica mediante el vector de coordenadas:

$$\theta(t) = (\theta_1(t), \theta_2(t), \dots, \theta_n(t))^T \quad (1-1)$$

donde $\theta_i(t)$ es un ángulo o una distancia, y n es el número de grados de libertad de la estructura. Dicha cantidad viene determinada por el número de articulaciones que posea la estructura.

La cinemática, en la mayoría de las aplicaciones, trata de resolver las relaciones entre las coordenadas generalizadas y la posición y orientación final del brazo en función de un sistema de referencia fijo. Según se trate esa relación en un sentido u otro (figura 1-6), se tendrán los problemas cinemáticos directo (ver Anexo B) e inverso (ver Anexo C).

La resolución de los mismos se realiza utilizando un procedimiento formal que consiste en colocar un sistema de ejes en cada articulación. La posición y orientación del extremo libre del brazo (denominado *efector final*), puede expresarse en función de un sistema de referencia fijo, realizando sucesivas transformaciones de coordenadas, a medida que vamos recorriendo el brazo desde el punto final de la cadena hasta el punto fijo a la base. Se tendrá entonces una relación no lineal entre las variables cartesianas del efector final y las coordenadas generalizadas del robot. Esta relación se obtiene mediante la representación ideada por Denavit-Hartenberg (ver Anexo A).

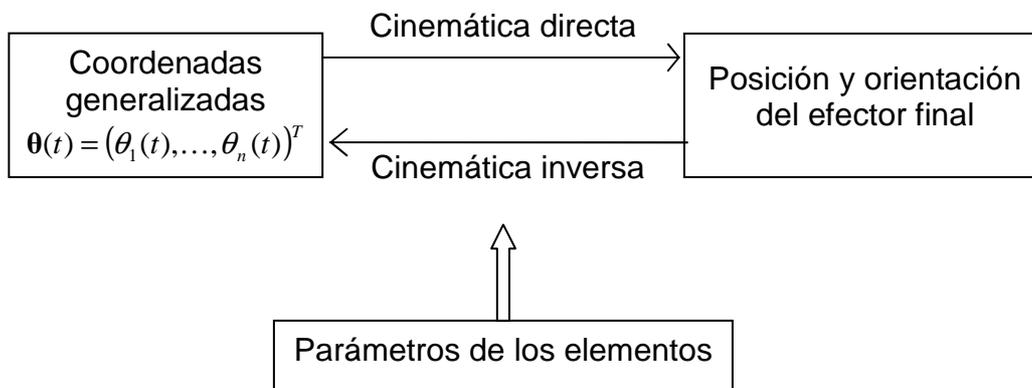


Figura 1-6 : Esquema de los problemas cinemáticos directo e inverso.

1.1.3 Dinámica

El estudio de las ecuaciones de movimiento del brazo de robot, atendiendo a las causas que lo producen, es el objetivo de la dinámica. Estas ecuaciones son de utilidad para la simulación del comportamiento dinámico del brazo de robot a través de un computador, así como para el diseño de algoritmos de control.

Las leyes físicas de la mecánica, como las leyes de Newton o de Lagrange, son las que originan el conjunto de ecuaciones que describen la dinámica del robot. Estas ecuaciones contendrán, por tanto, los parámetros geométricos de diseño del robot, así como las inercias de cada uno de los elementos. Las formulaciones de Lagrange-Euler y las de Newton-Euler son los métodos más recurridos a la hora de obtener las ecuaciones dinámicas de un robot. Sin embargo, existen otras formulaciones igualmente válidas como la recursiva de Lagrange o la generalizada de d'Alembert y Lee.

Todas estas formulaciones son distintas formas de afrontar la resolución de la dinámica de un manipulador, aunque el objetivo es el mismo: el cálculo de los pares de fuerza. Los resultados obtenidos son igualmente equivalentes. La ventaja de utilizar una formulación u otra reside en la parte del cómputo de la dinámica que queramos acelerar.

Por ejemplo, la formulación de Lagrange-Euler (ver Anexo D) es una forma simple y elegante (pues parte de la formulación lagrangiana) de las ecuaciones dinámicas, y se utilizan para definir eficientemente algoritmos de control en el espacio de las variables de articulación. La formulación de Newton-Euler (ver Anexo E) es más engorrosa y, aparentemente, menos elegante, pero se utiliza para calcular con rapidez los pares de fuerza generalizados. Éste es un método de aplicación en tiempo real, el cual desarrolla adecuadamente las ecuaciones de Newton para obtener una formulación recursiva computacionalmente eficiente (Una comparación entre ambas formulaciones puede encontrarse en el Anexo E).

1.1.3.1 Ecuaciones dinámicas de un robot manipulador

Supongamos un robot de n articulaciones. Cualquiera de las formulaciones anteriores nos lleva a obtener un conjunto de ecuaciones diferenciales, en general altamente no lineales y acopladas, que describen la dinámica de dicho robot. Así, por ejemplo, utilizando la formulación de L-E¹, las ecuaciones dinámicas que describen el movimiento del manipulador son:

$$\boldsymbol{\tau}(t) = \mathbf{B}(\boldsymbol{\theta}(t))\ddot{\boldsymbol{\theta}}(t) + \mathbf{h}(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t)) + \mathbf{c}(\boldsymbol{\theta}(t)) \quad (1-2)$$

En este conjunto de ecuaciones, $\boldsymbol{\tau}(t) = (\tau_1(t), \dots, \tau_n(t))^T$ es el momento aplicado en cada una de las articulaciones, $\boldsymbol{\theta}(t) = (\theta_1(t), \dots, \theta_n(t))^T$ es el vector de coordenadas generalizadas del brazo, tal y como se definió en (1-1), mientras que $\dot{\boldsymbol{\theta}}(t)$ y $\ddot{\boldsymbol{\theta}}(t)$ son los vectores velocidad y aceleración respectivamente.

$\mathbf{B}(\boldsymbol{\theta}(t))$ es una matriz de inercia simétrica, cuyos elementos B_{ij} están relacionados con los pares de reacción inducidos por la aceleración de la articulación j actuando en la articulación i .

$\mathbf{h}(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t))$ es el vector de fuerza de Coriolis y centrífuga. Sus elementos h_i están relacionados con la fuerza centrífuga y con la fuerza de Coriolis sobre la articulación i generada por la velocidad angular de las otras articulaciones.

Por último, $\mathbf{c}(\boldsymbol{\theta}(t))$ es un vector de fuerza gravitatoria, donde sus elementos c_i representan los términos de carga gravitatoria debido a cada una de las articulaciones.

¹ Utilizando las otras formulaciones (Newton-Euler, d'Alembert, etc.) también se pueden obtener las ecuaciones matriciales presentadas. Sin embargo, la notación empleada para la ecuación general (1-2), proviene directamente de la formulación lagrangiana.

El sistema de ecuaciones (1-2) está compuesto de ecuaciones diferenciales de segundo orden, acopladas y no lineales. Incluyen, como se ha descrito, todos los efectos inerciales, centrífugos, de Coriolis y gravitacionales de los elementos que componen el brazo de robot.

Igual que para el caso de la cinemática, se plantean dos problemas fundamentales. En primer lugar, dadas las fuerzas y momentos aplicados, $\tau_i(t)$, el cálculo de las aceleraciones en cada una de las articulaciones y, a partir de éstas, las velocidades y coordenadas generalizadas. Este es el *problema dinámico directo*.

En segundo lugar, dadas las coordenadas generalizadas que se quieren alcanzar (a través de un planificador de trayectoria), y sus derivadas, calcular las fuerzas y momentos que se deben aplicar a cada articulación para alcanzar dicha configuración. Se habla entonces del *problema dinámico inverso*. Un esquema de ambos problemas puede verse en la figura 1-7.

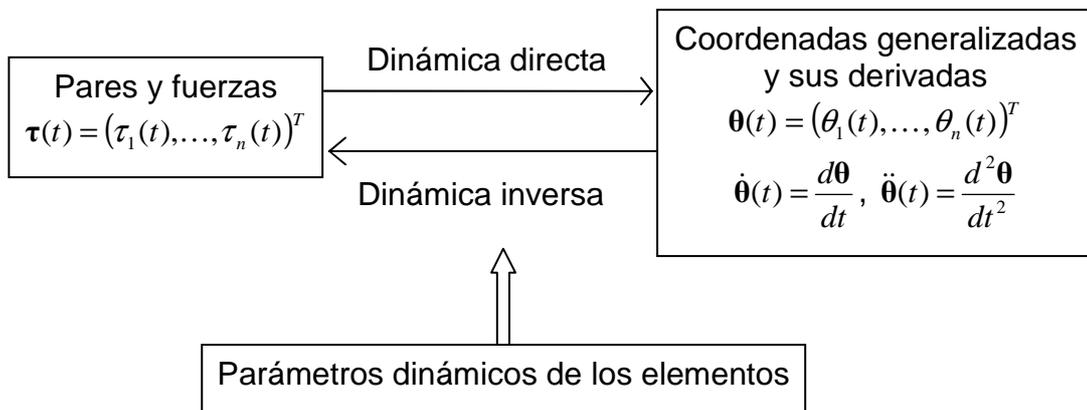


Figura 1-7: Esquema de los problemas dinámicos directo e inverso.

A continuación se presentan los modelos dinámicos de dos robots manipuladores empleados en el transcurso de este trabajo. Se trata de un modelo de dos articulaciones y del modelo completo de seis articulaciones del robot PUMA de Unimation.

1.1.3.2 Manipulador de dos grados de libertad

Uno de los ejemplos más sencillos de aplicación de estas ecuaciones es el manipulador plano de dos articulaciones de revolución, afectado por la gravedad. El sistema de ecuaciones que describe la dinámica del mismo viene dado por:

$$\begin{aligned}
 \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} &= \begin{bmatrix} \frac{1}{3}m_1l_1^2 + \frac{1}{3}m_2l_2^2 + m_2l_1^2 + m_2l_1l_2C_2 & \frac{1}{3}m_2l_2^2 + \frac{1}{2}m_2l_1l_2C_2 \\ \frac{1}{3}m_2l_2^2 + \frac{1}{2}m_2l_1l_2C_2 & \frac{1}{3}m_2l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \\
 &+ \begin{bmatrix} -m_2l_1l_2S_2\dot{\theta}_1\dot{\theta}_2 - \frac{1}{2}m_2l_1l_2S_2\dot{\theta}_2^2 \\ \frac{1}{2}m_2l_1l_2S_2\dot{\theta}_1^2 \end{bmatrix} + \begin{bmatrix} \frac{1}{2}m_1gl_1C_1 + \frac{1}{2}m_2gl_2C_{12} + m_2gl_1C_1 \\ \frac{1}{2}m_2gl_2C_{12} \end{bmatrix} \quad (1-3)
 \end{aligned}$$

donde :

l_i es la longitud del elemento i ; con $i=1,2$

m_i es la masa del elemento i

$$C_i = \cos(\theta_i)$$

$$S_i = \sin(\theta_i)$$

$$C_{ij} = \cos(\theta_i + \theta_j)$$

Para manipuladores con tres o más grados de libertad, la obtención de dichas ecuaciones es compleja, por el gran número de operaciones a realizar. En ese caso, se suele utilizar la formulación de N-E recursiva, unida a un método numérico de integración, con el fin de resolver el problema dinámico inverso.

1.1.3.3 Manipulador PUMA

El brazo manipulador PUMA es uno de los robots más habituales en tareas de manipulación. Dicho robot tiene seis grados de libertad, correspondientes a las seis articulaciones de revolución que posee (ver figura 1-1). La obtención del modelo dinámico del mismo es bastante compleja. En este trabajo se utiliza el modelo obtenido por Armstrong et al [Arm86] para el manipulador PUMA 560, el cual es empleado para comprobar los algoritmos de control expuestos.

Dicho manipulador se representa en la figura 1-8. Como se observa, posee tres articulaciones en el cuerpo y dos articulaciones en la muñeca, todas ellas de revolución. El último grado de libertad queda definido por la apertura o cierre de la pinza aprehensora, utilizada como efector final de esta estructura.

En el citado artículo [Arm86], se explicitan los parámetros dinámicos e inerciales de la estructura, el modelo cinemático con sus parámetros D-H y se obtienen cada uno de los términos de la matriz de inercia y de los vectores de fuerza de Coriolis, centrífuga y gravitatoria. Además, se obtiene una aproximación del modelo dinámico bastante realista, en la medida en que se desechan los términos con menos influencia en el resultado final de las distintas matrices, posibilitando unas ecuaciones dinámicas más sencillas de evaluar en tiempo real.

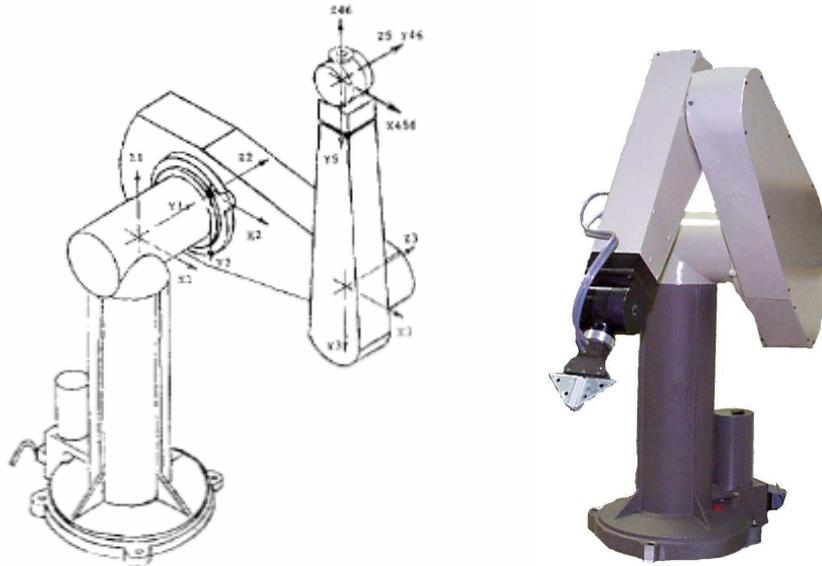


Figura 1-8 : Esquema del manipulador PUMA 560 de Unimation en su posición inicial [Arm86], donde se observa la disposición de los sistemas de referencia utilizados para cálculos cinemáticos y la obtención del modelo dinámico. A la derecha, una imagen de un manipulador real con el efector final incorporado.

1.1.4 Planificación de trayectorias.

En los apartados anteriores quedan definidos todos los aspectos y ecuaciones relevantes, en lo concerniente a la cinemática y dinámica de manipuladores. La cinemática directa e inversa es vital en el cálculo de las trayectorias a recorrer por el robot. Es el problema denominado *control cinemático*. Una vez obtenida la misma se procede a los cálculos de la dinámica, necesarios para resolver el problema denominado *control dinámico*, que consiste en diseñar controladores eficaces para la ejecución precisa por parte del manipulador de movimientos predefinidos.

La trayectoria del robot es la curva que sigue el manipulador desde una configuración (posición y orientación) inicial hasta una configuración final. El problema de *planificación de trayectoria* consiste en encontrar una curva próxima a una serie de puntos que queremos que recorra el brazo manipulador. El planificador tiene en cuenta las ecuaciones cinemáticas del robot y utiliza técnicas de interpolación de trayectorias para resolver el movimiento deseado. Dicho movimiento puede estar sujeto a *ligaduras de obstáculo*, si algún objeto se interpone en el ámbito de trabajo del robot, o a *ligaduras de camino*, si el manipulador ha de recorrer una trayectoria predefinida.

La planificación puede hacerse atendiendo a las coordenadas generalizadas de las articulaciones, o atendiendo a las coordenadas cartesianas de la trayectoria. La primera de ellas tiene ventajas sobre la segunda ya que, en general, se puede realizar en tiempo real y se trabaja directamente sobre las variables controladas.

En la planificación de una trayectoria de articulación son importantes cuatro posiciones: la posición inicial y final de la articulación, y dos puntos intermedios que se denominan posición de despegue y de asentamiento. Desde la posición inicial a la posición de despegue, la articulación adquiere la velocidad suficiente para avanzar hacia la posición final. A partir de la posición de asentamiento, la articulación debe reducir la velocidad para llegar a la posición final (por lo general) con velocidad y aceleración nulas.

Los algoritmos usuales de planificación utilizan funciones polinomiales de un cierto grado para aproximar la trayectoria. Uno de los métodos habituales [Fu88] es considerar una trayectoria dada por un polinomio de orden 7 para cada una de las articulaciones, calculando los parámetros de las mismas en base a los nodos especificados y a condiciones de continuidad. La complejidad inherente a ese método (en cuanto al elevado orden de los polinomios) se ve reducida si la trayectoria se divide en *segmentos* y se utilizan polinomios de menor orden en cada una de ellos. Así surgen algunos métodos como la trayectoria 4-3-4, la 3-5-3 y la trayectoria con 5 cúbicas, que difieren en el orden de los polinomios empleados en los segmentos.

1.2 Problemas en el control de manipuladores

Una vez expuestos los conceptos básicos de la robótica de manipulación, en esta sección se aborda el problema del control de manipuladores.

El orden seguido para el planteamiento de este tema es exponer cada uno de los problemas de control, los cuales se irán detallando en las secciones siguientes, y a continuación proponer el controlador que da respuesta a dicho problema. De esta forma se plantearán desde los controladores más sencillos, como el PID y sus variantes (compensación de gravedad, prealimentación), hasta los más complejos, como los controladores adaptativos, robustos o de estructura variable.

Más adelante se presentan los problemas abiertos en el control de manipuladores, indicando algunas de las soluciones propuestas hasta la fecha, pero que han de ser ampliamente desarrolladas en el futuro.

1.2.1 Problemas clásicos del control de manipuladores

El problema básico del control de manipuladores se puede sintetizar en encontrar los pares a aplicar a cada una de las articulaciones del manipulador (o, equivalentemente, los voltajes a aplicar a cada uno de los actuadores que comandan las articulaciones) para que el extremo del manipulador realice la tarea especificada.

El primer problema de control es el control de posición, en el cual simplemente se requiere que el extremo del manipulador alcance una posición

y orientación finales con unas especificaciones dadas, como pueden ser, tiempo de ejecución de la trayectoria, precisión del posicionamiento final de la estructura, oscilaciones en torno a ese punto, sobrepasamientos, etc.

Este problema se resuelve con técnicas de control PID. Pero, inicialmente, se debe plantear si se realiza control monoarticular o multiarticular, dependiendo de si la influencia del resto de la estructura sobre cada articulación es despreciable o no, respectivamente.

En el primer caso se diseña la estrategia para cada una de las articulaciones, sin tener en cuenta el resto. Las técnicas de control PID e I-PD resuelven eficazmente el problema de control de posición en este caso. En el caso de control multiarticular, se debe acudir a técnicas de par calculado o linealización, para luego aplicar las técnicas de control anteriores.

El segundo problema de control es el control de movimiento. El objetivo de control es que el extremo final del manipulador recorra una trayectoria especificada con la máxima precisión. Debido al aumento de la complejidad de este problema, se deben usar técnicas basadas en modelo. Esto asegura un comportamiento cercano al de la trayectoria de referencia. Surgen así las técnicas de control PID con prealimentación y PD con compensación de gravedad.

Sin embargo, las técnicas basadas en modelo presentan varias dificultades. La principal es la existencia de incertidumbres en el modelo, debido a parámetros desconocidos de la estructura (masas de los eslabones, inercias, etc.) o incluso variabilidad de los mismos. Para ello se emplean técnicas de control adaptativo, de control robusto y de control de estructura variable.

El tercer problema de control es el control de fuerza. En los dos anteriores, el problema a resolver no tiene en cuenta la interacción del robot con su entorno. Pero en tareas tales como el ensamblado, atornillado, corte de piezas, paletizado, etc., se interactúa con el entorno, por lo que se debe controlar el par y la fuerza aplicada por cada una de las articulaciones. Se precisa por tanto la utilización de sensores de fuerza. Los métodos de control más comunes en este caso son el control por rigidez, control por impedancia y control híbrido posición/fuerza.

Estos tres problemas básicos son explicados en las secciones que vienen a continuación. La forma de presentar las distintas técnicas de control que resuelven estos problemas es presentar el problema de control y las técnicas empleadas para solucionarlos. Pero antes, veamos algunos de los problemas que quedan por resolver dentro del control de manipuladores.

1.2.2 Problemas abiertos en el control de manipuladores

Aunque se han dado numerosos métodos para el control de manipuladores, incluso un método de clasificación de controladores basado en

tareas, arquitecturas de control y características de movimiento [Mil99], la gran mayoría van encaminados hacia los manipuladores rígidos de elevado peso, que son los que han dominado las aplicaciones industriales. Sin embargo, estos robots comienzan a ser reemplazados por robots más ligeros, de estructura flexible, más pequeños y a los que se les requieren las mismas prestaciones que los anteriores. Incluso las tareas a los que están dedicados requieren por parte del robot una mayor inteligencia [Sic98].

Por tanto, problemas que están totalmente cerrados para el caso de robots rígidos, como son el control de posición y movimiento, vuelven a ser problemas abiertos para el caso de robots ligeros y flexibles, cuyo comportamiento dinámico varía sustancialmente respecto a los primeros.

Hay también otro tipo de problemas que aún no están resueltos completamente hasta la fecha, los cuales se indican en las subsecciones siguientes.

1.2.2.1 Problemas abiertos en el control de fuerza

Tal y como se ha indicado anteriormente, se han propuesto diversos algoritmos de control para solucionar el problema del control de fuerza en manipuladores [Whi85],[Yos00]. De entre ellos destacan el control por impedancia [Hog85],[Sic97], el control de posición / fuerza de bucle interno / externo [Sch88], el control paralelo de posición / fuerza [Chi93] y el control híbrido de posición / fuerza [Rai81],[Zha85],[Wan88]. Para una descripción detallada de los mismos referirse a los textos citados, los cuales son los textos originarios de dichos esquemas de control.

El control por impedancia y el control híbrido son los que más éxito han tenido dentro de los esquemas de control de fuerza. El primero trata de controlar la fuerza aplicada sobre un objeto o superficie a través del ajuste de la impedancia mecánica del extremo del robot, que se define como la relación entre la fuerza aplicada y la velocidad lineal del extremo del robot. El segundo esquema de control se basa en la diferenciación de direcciones en el espacio de tarea del robot para realizar, por separado, control de posición y control de fuerza. Así, mientras en una dirección se realiza control de fuerza, en las otras se realiza control de posición. Estas direcciones pueden variar durante la ejecución de una tarea. Este esquema también ha sido aplicado a robots móviles [You00].

Una alternativa a estos métodos es la utilización del controlador de fuerza adaptativo presentado en [Sic96], en donde se plantea un control adaptativo para la posición (ver 1.6.1 y subsecciones) basado en el esquema de control paralelo de posición / fuerza mencionado anteriormente.

Dentro de este campo aún quedan por resolver varios aspectos de control como son:

- Estabilidad de los esquemas de control durante el paso entre las situaciones de no contacto a contacto entre el extremo del robot y el objeto o superficie sobre la que se aplica la fuerza,
- Control de posición / fuerza (o velocidad / fuerza) en el caso de manipuladores flexibles [Jat03],[Jat06].
- Modelado de entornos dinámicos,
- Etc.

1.2.2.2 Robots cooperativos

Algunas tareas precisan el movimiento coordinado dos o más robots, ya que uno sólo no es suficiente para realizarlas: ensamblado de piezas mecánicas, manipulación de objetos de grandes dimensiones o peso, etc. El problema de control de multi-robots ha sido resuelto con eficacia aplicando técnicas de maestro-esclavo [Luh87], formulación simétrica [Uch93] y control basado en espacio de tarea cooperativo [Chi96].

Sin embargo, problemas como el manejo de piezas flexibles o el control cooperativo de manipuladores con articulaciones o elementos flexibles están aún en estudio.

1.2.2.3 Manos robóticas de múltiples dedos

Una de las tareas en las que más se ha dedicado tiempo de investigación por parte de la comunidad científica internacional es en el control de manos robóticas de múltiples dedos [Mas85],[Gru89]. Este problema se puede ver como derivado del problema del control de multi-robots. Su principal cometido es la sujeción y manipulación con precisión extrema de objetos. Por esa razón, el diseño mecánico de las mismas es bastante complejo, lo que acarrea el problema de distribuir la acción de control entre los distintos dedos.

Por otra parte, debido al alto coste de estos dispositivos, se ha optado por dispositivos más simples y baratos, los cuales deben ofrecer buenas prestaciones en términos de sujeción y manipulación de objetos. Los diseños de control clásicos asumen estructuras de dedos rígidos, con uniones no rotatorias ni deslizantes. Si alguna de estas premisas no se cumple, el problema de control es bastante más complejo. Por ejemplo, si se permite una rotación en las uniones de la cadena cinemática de los dedos, el sistema resulta ser no holonómico.

1.2.2.4 Sistemas subactuados

Estos sistemas se caracterizan por disponer de menos entradas de control que grados de libertad. Suelen presentar propiedades, como dinámicas de fase no-mínima, restricciones no holonómicas, etc., que hacen que el problema de control en estos sistemas sea complejo. Ejemplos de este tipo de sistemas son los manipuladores flexibles, los robots móviles, los robots

caminadores y los de navegación aérea (vehículos voladores) o marítima (sumergibles).

Los algoritmos de control típicos, como la linealización por realimentación o técnicas basadas en la propiedad de pasividad, no son aplicables cuando el sistema es subactuado. Las técnicas de control probadas en ciertos sistemas subactuados son la de perturbaciones singulares [Sic88], *backstepping* [Krs95] y *forwarding* [Sep97]. Sin embargo, la robustez y adaptabilidad de las mismas no están aún probadas.

1.2.2.5 Teleoperación

Las tareas típicas dentro del campo de la teleoperación son la manipulación de objetos radioactivos, la robótica de servicios y la exploración submarina. El problema de control básico es el posicionamiento a distancia de la estructura, así como la manipulación de la herramienta para realizar las tareas específicas.

Uno de los aspectos de interés actuales dentro de este campo son las interfaces táctiles, las cuales incrementan la interacción entre el usuario maestro y el robot esclavo teleoperado. Estos dispositivos son ampliamente utilizados en operaciones médicas o interfaces de realimentación de fuerza. En este punto es en donde se concentra la investigación actual, ya que las capacidades sensoriales de fuerza y movimiento son difícilmente reproducibles en la interfaz maestra.

1.2.2.6 Visión por computador

Actualmente la visión por computador es uno de los campos de mayor aplicación dentro de la robótica de manipulación, ya que a través de la información extraída de las imágenes se puede obtener una gran cantidad de información del entorno del robot y del propio robot, permitiendo ahorrar la operación con múltiples sensores, además de propiciar un método de medida sin contacto con el entorno.

El control visual (*visual servo control*) [Hut96] se basa en el control del robot a través de la información obtenida mediante cámaras. Un problema de control común es el posicionamiento preciso de la muñeca o herramienta de trabajo del robot a través de las imágenes captadas por una cámara calibrada y colocada, bien en el extremo del robot, bien en una parte fija de su estructura o alrededores. El control visual extrae la información requerida de las imágenes y la realimenta hacia el controlador, el cual posicionará el robot de forma adecuada.

Problemas abiertos en este campo son relativos a la captación de imágenes en entornos complejos y dinámicos, por lo que se requieren técnicas robustas y adaptativas para realizar el control visual [Sha96].

1.3 Elección de la estructura de control

El primer problema que se plantea a la hora de controlar un manipulador es la elección entre un control descentralizado o monoarticular, y un control multiarticular. La diferencia entre ambas definiciones estriba en despreciar o no el efecto que tiene en una articulación el movimiento de las otras articulaciones. Así, en el control monoarticular se desprecia la interacción entre los grados de libertad del robot, teniéndose entonces un control de tipo descentralizado, donde cada articulación posee su controlador independiente. En el control multiarticular, sin embargo, se considera el manipulador como un sistema multivariable, en el que el movimiento de una articulación influye sobre el resto de articulaciones.

El utilizar un tipo de control u otro depende en gran medida de los coeficientes de reducción de las reductoras, utilizadas en las articulaciones para incrementar el par suministrado por el actuador.

1.3.1 Influencia de las reductoras

La elección entre técnicas de control monoarticulares o multiarticulares depende de las reductoras que se acoplan a las articulaciones. Para ver su influencia en la dinámica de los manipuladores consideremos la ecuación dinámica del conjunto formado por el manipulador y los actuadores de cada una de las articulaciones.

Para ello consideramos que cada actuador lleva acoplado una reductora de factor de reducción $k_i > 1$, con $i = 1, \dots, n$, siendo n el número de articulaciones. Entonces podemos poner:

$$\theta_a = \mathbf{K}\theta \quad (1-4)$$

donde θ_a es la posición del actuador, θ la posición de la articulación y \mathbf{K} la matriz diagonal de elementos k_i . De igual forma se tiene que:

$$\dot{\theta}_a = \mathbf{K}\dot{\theta} \quad (1-5)$$

$$\ddot{\theta}_a = \mathbf{K}\ddot{\theta} \quad (1-6)$$

$$\tau_a = \mathbf{K}^{-1}\tau \quad (1-7)$$

donde τ_a es el par ejercido por el actuador y τ el par desarrollado por la articulación. Sustituyendo estas expresiones en la ecuación (1-2) se tiene la siguiente expresión para la dinámica del conjunto manipulador, actuadores y reductoras:

$$\boldsymbol{\tau}_a = \mathbf{K}^{-1} \mathbf{D}_1 \mathbf{K}^{-1} \ddot{\boldsymbol{\theta}}_a + \boldsymbol{\tau}_p \quad (1-8)$$

donde se ha aplicado la siguiente propiedad para la matriz de inercia:

$$\mathbf{D}(\boldsymbol{\theta}) = \mathbf{D}_1 + \mathbf{D}_2(\boldsymbol{\theta}) \quad (1-9)$$

siendo \mathbf{D}_1 una matriz diagonal de elementos constantes que expresa la influencia de cada elemento en la inercia que soporta el actuador asociado al mismo, mientras que $\mathbf{D}_2(\boldsymbol{\theta})$ recoge los términos no constantes de la matriz de inercia. El par perturbador $\boldsymbol{\tau}_p$ recoge la influencia de unas articulaciones sobre otras (acoplamiento), y viene dado por:

$$\boldsymbol{\tau}_p = \mathbf{K}^{-1} \mathbf{D}_2 \mathbf{K}^{-1} \ddot{\boldsymbol{\theta}}_a + \mathbf{K}^{-1} \mathbf{h} + \mathbf{K}^{-1} \mathbf{c} \quad (1-10)$$

A la vista de (1-8) y (1-10) se deduce que si se dispone de coeficientes de reducción altos, se puede despreciar el par perturbador, por lo que es posible considerar un desacoplo entre las articulaciones, que se comportarán como sistemas lineales de segundo orden:

$$\boldsymbol{\tau}_a \approx \mathbf{J} \ddot{\boldsymbol{\theta}}_a \quad (1-11)$$

donde \mathbf{J} es una matriz diagonal que recoge los términos de inercia de cada elemento sobre su propio actuador. Es en este caso cuando se puede hablar de control monoarticular.

1.3.2 Ventajas e inconvenientes del control monoarticular

La principal ventaja del control monoarticular es el comportamiento del sistema como un sistema desacoplado de segundo orden, lo que conlleva a la facilidad del diseño de las estrategias de control. Esto ocurre con los manipuladores que disponen de reductoras con coeficientes de reducción elevados.

Sin embargo, tener coeficientes de reducción elevados impide que se ejecuten movimientos a velocidades elevadas, aumenta las holguras y elasticidades en los sistemas de transmisión y aumenta el rozamiento viscoso. Por esta razón no se utiliza este tipo de reductoras. Como contrapartida, la complejidad en el control del manipulador es mayor.

1.4 Control de posición

Las tareas más comunes de los robots manipuladores pueden clasificarse según si realizan movimientos sin interaccionar con su entorno o si interaccionan con él, por ejemplo, realizando una determinada presión sobre un objeto. Los robots encargados de realizar trabajos de pintado o aplicación de

puntos de soldadura pertenecen a la primera clase de robots, mientras que aquellos destinados al transporte de piezas o ensamblado pertenecen a la segunda clase.

En ambos casos, uno de los problemas de control fundamentales es el control de posición, que consiste en determinar los pares a aplicar en las articulaciones para que el extremo final del manipulador alcance un punto determinado en el espacio de trabajo del robot. Este es el llamado problema de regulación. Formalmente:

$$\tau(t) \quad \left| \quad \lim_{t \rightarrow \infty} \theta(t) = \theta_d \quad (1-12)$$

siendo θ_d un vector constante con las posiciones de referencia para cada articulación.

Para el caso más general, el controlador tendrá la forma:

$$\tau = \tau(\theta, \dot{\theta}, \ddot{\theta}, \theta_d, \mathbf{B}(\theta), \mathbf{h}(\theta, \dot{\theta}), \mathbf{c}(\theta)) \quad (1-13)$$

donde la matriz $\mathbf{B}(\theta)$ y los vectores $\mathbf{h}(\theta, \dot{\theta})$ y $\mathbf{c}(\theta)$ son los que aparecen en el modelo dinámico (1-2). La gran mayoría de manipuladores poseen sensores de posición, y muchos incorporan sensores de velocidad o bien la estiman a partir de la medida de la posición, por lo que los vectores θ y $\dot{\theta}$ son medibles. En la práctica, el controlador no va a depender de las aceleraciones articulares $\ddot{\theta}$.

Se dice que el controlador no está basado en modelo si la expresión (1-13) no depende de forma explícita de $\mathbf{B}(\theta)$, $\mathbf{h}(\theta, \dot{\theta})$ ni $\mathbf{c}(\theta)$. Los primeros controladores que se exponen en este capítulo (PID, PID con prealimentación, PD con compensación de gravedad) son de esta clase, y son los que se emplean habitualmente para el problema de control de posición.

Sin embargo, un control más adecuado incluye la utilización del modelo. De hecho, en algunos controladores PID los parámetros del mismo se pueden calcular utilizando el modelo del robot. Las ventajas de utilizar un controlador basado en modelo son más evidentes en el caso del control de movimiento del robot, también llamado problema de seguimiento o *tracking*.

En los apartados siguientes se exponen los algoritmos de control de posición en los casos monoarticular y multiarticular.

1.4.1 Control de posición en el caso monoarticular

En los siguientes apartados se exponen los principales controladores en el caso monoarticular. Dado que el sistema resulta, en este caso, totalmente desacoplado, el control del robot será tratado como el control de n ejes independientes. Los esquemas utilizados son el control PID y el control I-PD.

1.4.1.1 Control PID

El control PID es la técnica de control habitual para abordar el control de un manipulador. La expresión del controlador, en el dominio de Laplace, es la siguiente:

$$\mathbf{C}(s) = \begin{bmatrix} K_{p1} + \frac{K_{i1}}{s} + K_{d1}s & 0 & \cdots & 0 \\ 0 & K_{p2} + \frac{K_{i2}}{s} + K_{d2}s & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & K_{pn} + \frac{K_{in}}{s} + K_{dn}s \end{bmatrix} \quad (1-14)$$

Con esta definición, la ley de control queda:

$$\mathbf{T}(s) = \mathbf{C}(s)\tilde{\Theta}(s) \quad (1-15)$$

donde $\mathbf{T}(s)$ es la entrada del sistema y $\tilde{\Theta}(s)$ el error, expresados en el dominio de Laplace. Dicho error viene dado por la diferencia entre la trayectoria de referencia y la trayectoria real:

$$\tilde{\theta}(t) = \theta_d(t) - \theta(t) \quad (1-16)$$

Con este controlador se asegura, en ausencia de perturbaciones, un error nulo en el problema de regulación, pero no se tiene un seguimiento perfecto de la trayectoria de referencia para cada una de las articulaciones (problema de seguimiento). En presencia de perturbaciones, si estas tienen forma de escalón, se asegura un rechazo de las mismas en régimen permanente, siempre que se utilice una acción integral no nula [Bar97].

Para mejorar el comportamiento de este controlador, se introducen dos mejoras, ambas para corregir los problemas que no resuelve el PID original. Se trata de, en el primer caso, añadir un término *feedforward*, y en el segundo caso, de un término de compensación de la gravedad.

1.4.1.2 Control I-PD

La condición de rechazo a perturbaciones del PID está condicionada a que la acción integral sea no nula. Sin embargo, es sabido que la acción integral que anula el error en régimen permanente en la posición tiene un efecto negativo en el control pues empeora la respuesta dinámica de la articulación (por efecto de oscilaciones indeseadas).

Una manera de corregir este efecto negativo es no incluir el término integral en el controlador. En la práctica, anular la acción integral y seguir obteniendo las mismas prestaciones por parte del controlador, puede conseguirse proponiendo ganancias altas para el controlador PD resultante,

sobre todo si se trata con un manipulador con reductoras de alto coeficiente de reducción. Sin embargo, aumentar las ganancias puede conllevar la saturación de los actuadores.

Una segunda posibilidad es incluir un término de compensación gravitatorio que permita el uso de un controlador PD (acción integral nula), lo cual seguirá asegurando un error estacionario nulo. Este controlador se emplea en el caso del control de movimiento, y será explicado más adelante.

Pero hay un modo de corregir los efectos negativos de la acción integral, sin anularla como en el caso anterior, consistente en modificar el esquema PID de forma que las acciones proporcional y derivativa se pasen a la realimentación, dejando la acción integral original, tal y como se observa en la figura 1-9.

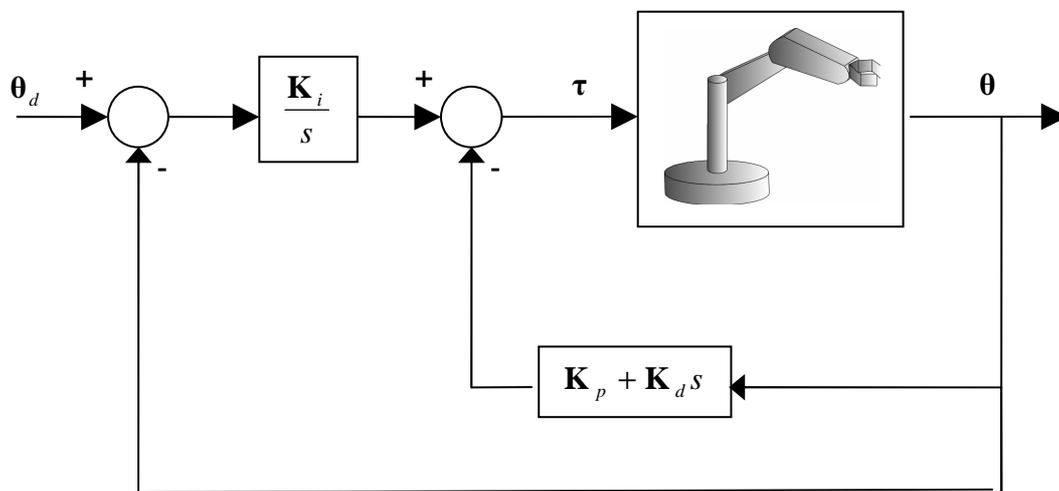


Figura 1-9: Esquema de control monoarticular I-PD.

Con este esquema de control se trata de reducir los efectos que se producen cuando existen cambios bruscos en las consignas o, en general, entre la posición real y la deseada. Dichos efectos son las fuertes acciones de control generadas, que pueden derivar en saturación de los actuadores, lo que implica tiempos de establecimientos en la consigna elevados e, incluso, inestabilidad del lazo cerrado. El controlador I-PD disminuye el problema de saturación pero a costa de ralentizar la respuesta del sistema.

1.4.2 Control de posición en el caso multiarticular

En esta sección nos centraremos en el problema de control multiarticular, donde las reductoras presentes en el sistema no tienen un coeficiente de reducción elevado, lo que supone un acoplamiento no despreciable entre las distintas articulaciones. Por otra parte, es éste el problema que plantea mayor dificultad a la hora del control y el más común en manipuladores industriales.

Cuando el control monoarticular no es posible, se utilizan técnicas de control multiarticular. Con objeto de utilizar técnicas semejantes a las de control monoarticular, se contempla el modelo dinámico real del robot completo, para realizar un desacoplamiento del mismo. Esta técnica es conocida como desacoplo por inversión de modelo o *feedback linearisation*.

1.4.2.1 Desacoplamiento por inversión del modelo

Para realizar el desacoplo [Kre89], se requiere el conocimiento exacto del modelo dinámico del sistema, dado por las ecuaciones (1-2), que recopilamos aquí sin incluir la dependencia temporal de las coordenadas generalizadas y sus derivadas:

$$\boldsymbol{\tau}(t) = \mathbf{B}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{c}(\boldsymbol{\theta}) \quad (1-17)$$

Para realizar la linealización de este modelo se define la entrada al modelo no lineal como:

$$\boldsymbol{\tau}(t) = \mathbf{B}(\boldsymbol{\theta})\mathbf{y}(t) + \mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{c}(\boldsymbol{\theta}) \quad (1-18)$$

donde $\mathbf{y}(t)$ es la nueva entrada al modelo lineal resultante. De esta forma se tiene una cancelación de los términos no lineales del sistema. Esto conduce al siguiente modelo de segundo orden, lineal y desacoplado con respecto a la nueva entrada:

$$\mathbf{y}(t) = \ddot{\boldsymbol{\theta}} \quad (1-19)$$

donde cada componente $y_i(t)$ influye sólo en la variable de articulación θ_i , independientemente del estado de las otras articulaciones. Con ello se tiene un modelo lineal del sistema sobre el cual aplicar técnicas de control convencionales.

La implementación del control dinámico inverso (1-18) requiere un conocimiento preciso del modelo dinámico del sistema. Esta condición es difícil de verificar debido al conocimiento impreciso de los parámetros mecánicos, la existencia de fricciones no modeladas (si éstas se tuvieran en cuenta) y el efecto de la masa de transporte en la dinámica del sistema.

1.4.2.2 Algoritmos de control en el caso multiarticular

El esquema de desacoplo por inversión del modelo permite obtener un sistema linealizado (1-19), que consiste en un sistema de segundo orden desacoplado, semejante a (1-11) pero respecto a la nueva entrada $\mathbf{y}(t)$.

Por esta razón, se pueden utilizar los mismos esquemas de control utilizados en el caso monoarticular: PID (ver sección 1.4.1.1) e I-PD (ver sección 1.4.1.2).

Sin embargo, estos controladores no aseguran un buen comportamiento en el problema del seguimiento. Para ese caso se emplea el control basado en modelo, tal y como se verá en la sección posterior.

1.5 Control de movimiento

A partir de este momento, dado que las técnicas de control multiarticular son las mismas que en el caso monoarticular, aplicando previamente la técnica de linealización por realimentación explicada en 1.4.2.1, nos referiremos sin distinción a un tipo de control u otro.

En el caso de control de movimiento de un manipulador, el problema a resolver es el de seguimiento de una trayectoria predefinida, también llamado problema de *tracking*. Formalmente, se puede expresar de la siguiente manera:

$$\boldsymbol{\tau}(t) \quad \left| \quad \lim_{t \rightarrow \infty} \boldsymbol{\theta}(t) = \boldsymbol{\theta}_d(t) \quad (1-20)$$

o, de forma equivalente,

$$\boldsymbol{\tau}(t) \quad \left| \quad \lim_{t \rightarrow \infty} \tilde{\boldsymbol{\theta}}(t) = 0 \quad (1-21)$$

donde, a diferencia de (1-12), el vector de posiciones deseadas no es constante.

En este caso se utiliza el control basado en modelo. Concretamente, se trata de añadir términos al controlador PID que aseguren un seguimiento adecuado de la trayectoria de referencia. Surgen así los controladores PID con prealimentación y PD con compensación de gravedad.

1.5.1 Control PID con prealimentación

Esta variante del PID se basa en añadir un término de prealimentación que incluya los efectos de la inercia de cada uno de los elementos de la estructura del manipulador. La ley de control en este caso viene dada por:

$$\mathbf{T}(s) = \mathbf{C}(s)\tilde{\boldsymbol{\Theta}}(s) + \mathbf{J}s^2\boldsymbol{\Theta}(s) \quad (1-22)$$

donde $\mathbf{C}(s)$ es el controlador PID de (1-14) y \mathbf{J} viene definida por (1-11).

En el caso, más general, de que se contemplen las fricciones viscosas que se producen en los actuadores, la expresión (1-22) se convierte en:

$$\mathbf{T}(s) = \mathbf{C}(s)\tilde{\boldsymbol{\Theta}}(s) + (\mathbf{J}s + \mathbf{F})s\boldsymbol{\Theta}(s) \quad (1-23)$$

donde

$$\mathbf{F} = \mathbf{K}^{-1} \mathbf{F}_v \mathbf{K}^{-1} \quad (1-24)$$

es una matriz que modela la fricción viscosa, dada por la matriz diagonal \mathbf{F}_v , y \mathbf{K} la matriz de factores de reducción provenientes de las reductoras acopladas a cada articulación.

Con este término se asegura solventar el problema del seguimiento de trayectoria. Un esquema de este controlador se puede ver en la Figura 1-10.

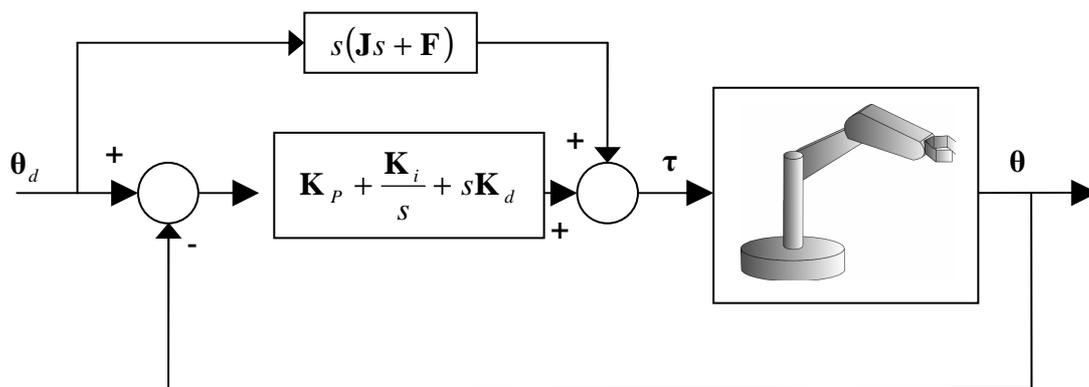


Figura 1-10: Controlador PID con prealimentación.

1.5.2 Control PD con compensación de gravedad

Otro tipo de controlador que da solución al problema de control de movimiento basado en modelo es el control PD con compensación de gravedad. Consiste en un controlador PD al que se añade un término de gravedad calculado con la expresión del vector de fuerzas gravitatorias de (1-2).

Este término evita la utilización del término integral del controlador PID, con los efectos negativos ya comentados previamente (ver 1.4.1.2). Además, este controlador sigue asegurando un error estacionario nulo.

En este caso, la ley de control adopta la forma:

$$\mathbf{T}(s) = \mathbf{C}_{K_i=0}(s) \tilde{\Theta}(s) + \mathbf{c}(\theta) \quad (1-25)$$

siendo $\mathbf{C}_{K_i=0}(s)$ el controlador PD dado por (1-14) cuando se anula la acción integral y $\mathbf{c}(\theta)$ el vector de fuerzas gravitatorio de (1-2). Un esquema de este tipo de control puede observarse en la figura 1-11, donde en el término de compensación se indica, con líneas de puntos, que las posiciones se utilizan para el cálculo del término gravitatorio.

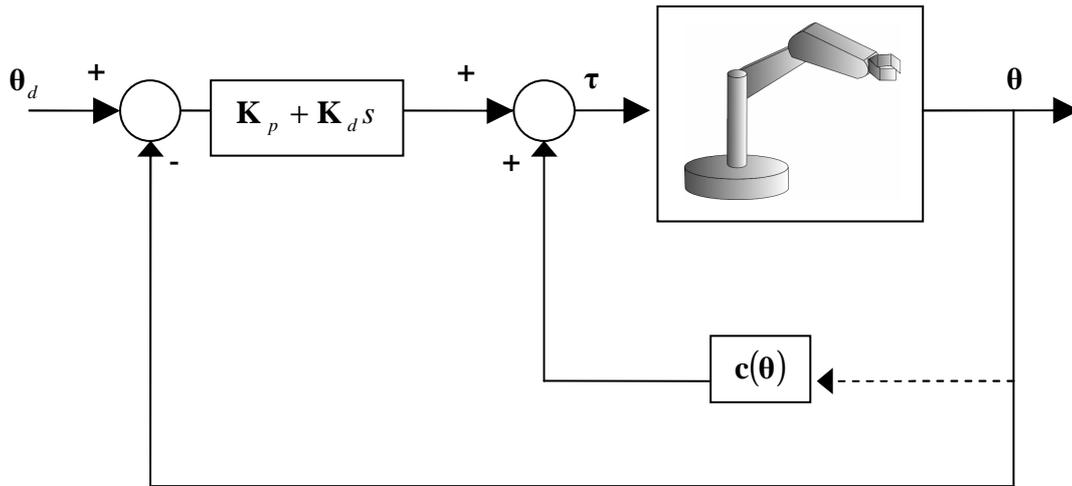


Figura 1-11: Esquema de control PD con compensación de gravedad.

1.5.3 Control PD con compensación de gravedad precalculada

Se trata del mismo controlador que el presentado en la sección anterior, pero con el término de compensación calculado a partir de las posiciones de referencia. En este caso, la ley de control adopta la forma:

$$\mathbf{T}(s) = \mathbf{C}_{K_i=0}(s) \tilde{\Theta}(s) + \mathbf{c}(\theta_d) \quad (1-26)$$

Un esquema de este tipo de control puede observarse en la figura 1-12. Este controlador da resultados análogos al anterior, siempre que los errores en la posición debido a perturbaciones no sean elevados. Se evita así, además, el cálculo *on-line* del término gravitatorio.

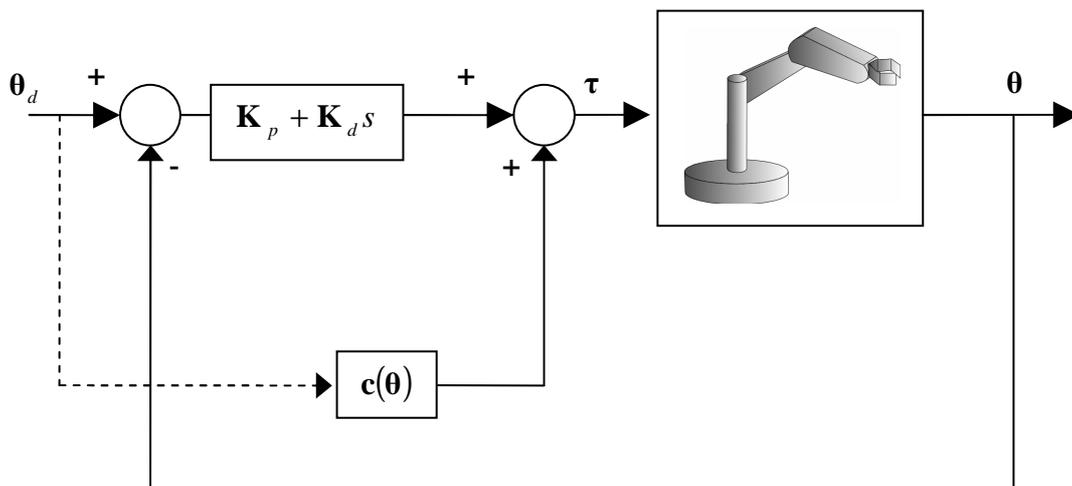


Figura 1-12: Esquema de control PD con compensación de gravedad precalculada.

1.6 Incertidumbres en el modelo

El problema más común en el control de manipulador es el conocimiento impreciso de los parámetros dinámicos del sistema, tales como inercias, rozamientos, masa de carga, etc. Pero también es común el caso en el que el modelo cambia debido a cambios en las condiciones de funcionamiento, como puede ser la variación de masas de transporte.

Este problema suele ser abordado mediante técnicas de control robusto y técnicas adaptativas [Abd91]. Estas técnicas son explicadas a continuación.

1.6.1 Control adaptativo

El controlador adaptativo se basa en la modificación de los parámetros de un controlador ajustable de acuerdo a un criterio de adaptación basado en el comportamiento del sistema a controlar. Esta situación es la más común en el caso de un robot, ya que un controlador fijo ajustado para una zona de trabajo y una masa determinada no será válido cuando esas condiciones varíen.

Hay varios esquemas de control adaptativo que se suelen aplicar en el caso de un manipulador [Hsi86],[Tos88]. El primero de ellos se refiere al controlador por planificación de ganancias, el segundo al esquema MRAC (del inglés *model reference adaptive controller*) y el tercero al controlador de par calculado adaptativo, el cual es una particularización del regulador autosintonizable (*STR* del inglés *self-tuning regulator*).

1.6.1.1 Control adaptativo por planificación de ganancias

Este tipo de controlador se basa en un controlador ajustable, el cual varía sus parámetros de acuerdo a una tabla prediseñada. La adaptación de los parámetros se realiza en base a la medida de ciertas variables del sistema o de su entorno.

En el caso de un manipulador, esta técnica se emplea tomando un controlador PID y realizando la optimización del mismo para diferentes puntos de operación y con distintas masas de transporte [Bar97]. Esto se debe a que la variación de estas magnitudes es la que implica un cambio significativo en la dinámica del manipulador.

Dado el alto número de variables (en un manipulador de 5 grados de libertad, por ejemplo, se tendrían 6 variables: las variables de articulación y la masa de transporte) cuya modificación implica un cambio en la dinámica del manipulador, y la discretización que se haga de las mismas, se puede tener un número elevadísimo de puntos de funcionamiento. Esto conlleva al diseño de una tabla de un número elevado de entradas, inviable de evaluar. Para disminuir el costo de diseño de la tabla, se realizan ciertas aproximaciones, como son el desprestigiar los movimientos de los últimos grados de libertad (los correspondientes al efector final), combinar los elementos consecutivos a partir

de un elemento determinado utilizando las definiciones de masa reducida y radio de giro, etc. Con estas aproximaciones, el número de entradas se reduce considerablemente.

Por otra parte, la estabilidad del controlador está asegurada en los puntos de operación considerados en la tabla, pero no en los puntos intermedios.

1.6.1.2 Control adaptativo por modelo de referencia

El esquema de este tipo de controlador se puede observar en la figura 1-13. El objetivo de control es que el lazo cerrado se comporte según un modelo de referencia predefinido. El controlador adapta sus parámetros en función del error entre la salida real del sistema en lazo cerrado y la salida del modelo de referencia, según un criterio de estimación dado [Ast95].

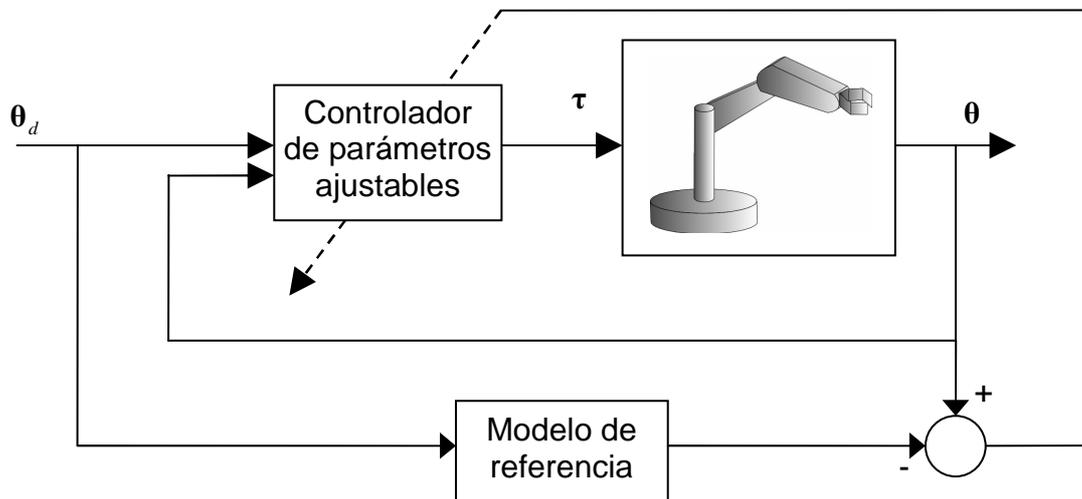


Figura 1-13: Controlador adaptativo por modelo de referencia (MRAC).

Una dificultad de este método radica precisamente en la elección del criterio de adaptación de los parámetros para asegurar la estabilidad y un buen funcionamiento del sistema. Los métodos más habituales son el de la función de sensibilidad, el de estabilidad de Lyapunov y el de hiperestabilidad de Popov [Rod96].

Otra dificultad radica en la elección del modelo de referencia. Una forma habitual [Tos88] es considerar cada articulación por separado y modelarla como un sistema SISO de segundo orden, sin tener en cuenta el acoplamiento con el resto de articulaciones. En ese caso las ganancias de realimentación para la posición y velocidad se calculan, para cada articulación, mediante la minimización de una función de costo que depende del error en cada grado de libertad del robot.

Este tipo de control ha sido ampliamente estudiado, tanto en su formulación continua como discreta. Para este último caso, se puede acudir a [Akh05], donde se da una prueba de estabilidad en sentido Lyapunov del algoritmo de control.

1.6.1.3 Control adaptativo de par calculado

Este tipo de controlador, denominado también controlador por dinámica inversa o linealización por realimentación (*feedback linearisation*), consiste en la cancelación de la dinámica no lineal del modelo del manipulador a través de una entrada adecuada al mismo, según se ha descrito en el apartado 1.4.2.1.

Dicha cancelación se consigue si son conocidos los términos dinámicos del modelo. Este controlador consiste en el cálculo de estos términos a través de una identificación paramétrica. El inconveniente de esta técnica es que, para resolver eficazmente el problema de identificación, se precisa una trayectoria rica en frecuencias y las utilizadas en este caso son trayectorias suaves, obtenidas del planificador de trayectorias.

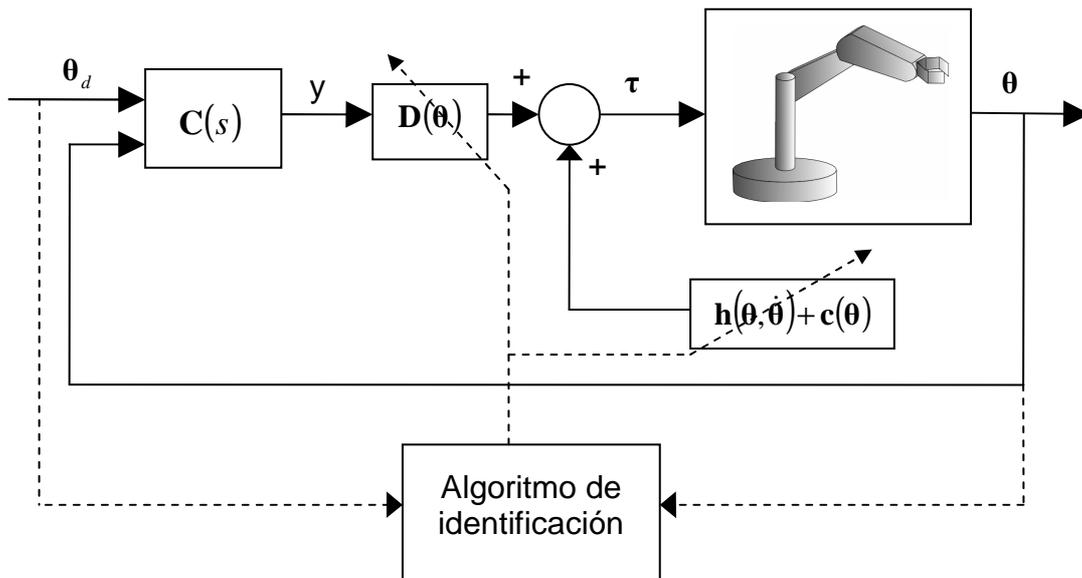


Figura 1-14: Control de par calculado adaptativo. El algoritmo de identificación estima las matrices dinámicas que aparecen en los recuadros atravesados por líneas discontinuas.

1.6.1.4 Otras técnicas de control adaptativo

La técnica de control anteriormente presentada hace uso del esquema de linealización por realimentación presentado en 1.4.2.1. Sin embargo, hay esquemas de control que hacen uso de la propiedad de parametrización lineal que presentan las ecuaciones dinámicas de un manipulador. Esta propiedad y las técnicas que se derivan de su aplicación son explicadas a continuación.

A) Propiedad de parametrización lineal

Esta propiedad establece que el conjunto de ecuaciones (1-2) es lineal en los parámetros de inercia, es decir, se puede definir un vector \mathbf{p} compuesto por parámetros de inercia tales como masas, longitudes de los elementos, momentos de inercia y una combinación de los mismos, así como una función $\mathbf{Y}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}})$, de forma que se cumple:

$$\boldsymbol{\tau} = \mathbf{B}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{c}(\boldsymbol{\theta}) = \mathbf{Y}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}})\mathbf{p} \quad (1-27)$$

B) Control adaptativo basado en la propiedad de parametrización lineal

Si se suponen incertidumbres en el modelo, que es el caso al que nos estamos refiriendo, o bien si los parámetros son completamente desconocidos, se puede plantear cualquiera de los esquemas vistos anteriormente basados en esta propiedad de parametrización [Spo03]. Por ejemplo, el esquema de control 1.6.1.3 se reduce a la estimación del vector de parámetros \mathbf{p} , habitualmente denominado $\hat{\mathbf{p}}$, a partir de las medidas del error $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}_d - \boldsymbol{\theta}$ en la posición del manipulador. Esta relación, en general, se puede expresar de la siguiente forma:

$$\dot{\hat{\mathbf{p}}} = f(\mathbf{B}(\boldsymbol{\theta}), \mathbf{Y}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}), \tilde{\boldsymbol{\theta}}) \quad (1-28)$$

que indica la actualización a realizar en el vector de parámetros desconocidos o que presentan incertidumbre.

Este esquema de control puede probarse que es asintóticamente estable (convergencia global a cero del error de seguimiento), permaneciendo todas las señales internas del bucle acotadas. Además, los parámetros estimados $\hat{\mathbf{p}}$ tienden a los parámetros reales \mathbf{p} siempre que la trayectoria de referencia satisfaga que sea persistentemente excitada. Esta propiedad puede expresarse de tal forma que para todo $t_0 > 0$, se debe cumplir:

$$\alpha \mathbf{I} \leq \int_0^{t_0+T} \mathbf{Y}^T(\boldsymbol{\theta}_d, \dot{\boldsymbol{\theta}}_d, \ddot{\boldsymbol{\theta}}_d) \mathbf{Y}(\boldsymbol{\theta}_d, \dot{\boldsymbol{\theta}}_d, \ddot{\boldsymbol{\theta}}_d) dt \leq \beta \mathbf{I} \quad (1-29)$$

siendo $\alpha > 0$, $\beta > 0$, $T > 0$ e \mathbf{I} la matriz identidad de tamaño adecuado.

Las desventajas de este método son que, por una parte, las aceleraciones del manipulador deben ser conocidas para evaluar la ley de control y, por otra, que la matriz de inercia $\mathbf{B}(\boldsymbol{\theta})$ debe estar bien definida (incluso, debe ser invertible) en todo instante.

Un esquema de control de este tipo, pero planteado en su versión discreta, puede verse en [Ara93], en el que el primer método planteado es un regulador autosintonizable basado en la propiedad de parametrización lineal. Asimismo se plantea el algoritmo de identificación del manipulador, el cual se basa en expresiones procedentes de la formulación del filtro de Kalman.

Otros controladores adaptativos están descritos en [Whi93], donde se comparan técnicas de control PD y basadas en linealización por realimentación con técnicas basadas en la propiedad de parametrización lineal. Esta comparación se realiza de forma empírica en un robot Scara y en un robot de diseño propio de accionamiento directo.

C) Control adaptativo basado en la propiedad de pasividad

La propiedad de pasividad de la dinámica de manipuladores es una alternativa a la propiedad de parametrización lineal, donde para evaluar la función $Y(\cdot)$ no hay que tener una medida de las aceleraciones en cada instante de la trayectoria, ni tampoco hay que calcular la inversa de la matriz de inercia $B(\theta)$.

Basados en esta propiedad, la ley de control, integrada por una acción PD y la acción de compensación no lineal de la dinámica, es [Spo93]:

$$\tau = \hat{B}(\theta)\mathbf{a} + \hat{h}'(\theta, \dot{\theta})\mathbf{v} + \hat{c}(\theta) - \mathbf{K}\mathbf{r} \quad (1-30)$$

donde se utiliza la siguiente propiedad:

$$\mathbf{h}(\theta, \dot{\theta}) = \mathbf{h}'(\theta, \dot{\theta})\dot{\theta} \quad (1-31)$$

además de la definición de las siguientes variables:

$$\begin{aligned} \mathbf{v} &= \dot{\theta}_d - \Lambda\tilde{\theta} \\ \mathbf{a} &= \dot{\mathbf{v}} = \ddot{\theta}_d - \Lambda\dot{\tilde{\theta}} \\ \mathbf{r} &= \tilde{\theta} + \Lambda\tilde{\theta} \end{aligned} \quad (1-32)$$

siendo \mathbf{K} y Λ matrices diagonales definidas positivas. Si se hace uso de la propiedad de parametrización lineal, la ley de control (1-30) queda:

$$\tau = \mathbf{Y}(\theta, \dot{\theta}, \mathbf{a}, \mathbf{v})\hat{\mathbf{p}} - \mathbf{K}\mathbf{r} \quad (1-33)$$

donde $\hat{\mathbf{p}}$ es la estimación del vector de parámetros de inercia \mathbf{p} .

Hay algunas propuestas de control que evitan la realimentación de la velocidad en este algoritmo. Así, en [Alo03] se dan las pautas para utilizar simplemente las medidas de la posición, y estimar de forma implícita la velocidad utilizando un filtro paso-bajo de primer orden:

$$\tau = \mathbf{Y}(\theta, \dot{\theta}, \mathbf{a}, \mathbf{v})\hat{\mathbf{p}} - \tilde{\mathbf{s}} \quad (1-34)$$

donde la actualización de los parámetros se hace utilizando:

$$\dot{\hat{\mathbf{p}}} = -\Gamma\mathbf{Y}(\theta, \dot{\theta}, \mathbf{a}, \mathbf{v})\tilde{\mathbf{s}} \quad (1-35)$$

Los vectores \tilde{s} y \hat{s} se obtiene como salida de los mencionados filtros aplicados sobre la medida de la posición.

1.6.2 Control robusto

En las técnicas de control adaptativo se pretende que el controlador identifique la dinámica del sistema a controlar, como es el caso del controlador de par calculado adaptativo (ver 1.6.1.3) o el basado en la propiedad de parametrización lineal (ver 0), o bien se adapte a cambios en la dinámica mediante la medición de variables auxiliares (ver 1.6.1.1) o asegurando un comportamiento en lazo cerrado predefinido (ver 1.6.1.2). De cualquier forma se consigue un controlador eficaz para un sistema en particular.

En las técnicas de control robusto el planteamiento es diferente. El controlador tiene una estructura fija, con un comportamiento aceptable para diferentes manipuladores que presenten incertidumbres en sus dinámicas. De hecho, este tipo de técnicas han sido ampliamente utilizadas en manipuladores con el objetivo de asegurar la estabilidad de los mismos en condiciones de incertidumbres en el modelo del mismo o situaciones de control desconocidas, la más común, masas de transporte desconocidas y variables.

En la literatura hay bastantes planteamientos de control robusto para manipuladores, los cuales, siguiendo a [Abd91] pueden clasificarse en cuatro clases:

La primera se refiere a las técnicas basadas en linealización por realimentación (o desacoplo por inversión del modelo, explicado ya en 1.4.2.1). Estas técnicas tratan de compensar la cancelación imperfecta de los términos no lineales de (1-17) a través de la entrada (1-18), debida a incertidumbres presentes en el modelo. Dicha compensación se lleva a cabo mediante entradas de control derivadas de la aplicación de la teoría de estabilidad de Lyapunov en sistemas no lineales.

Una segunda clase de controladores robustos aplican la propiedad de pasividad presente en las ecuaciones dinámicas del manipulador. Estas técnicas tratan de calcular la acción de control de forma que el conjunto controlador-planta mantenga la propiedad de pasividad ante incertidumbres en la dinámica del manipulador.

Una tercera clase de controladores robustos hacen uso de técnicas de control de estructura variable. Estas técnicas, en muchos de los casos, se reducen a las basadas en linealización por realimentación o las basadas en la propiedad de pasividad.

La cuarta clase de controladores robustos se basa en la aproximación de control por saturación, donde se emplea una ley de control discontinua para compensar las discrepancias entre el modelo estimado y el manipulador real.

Otro tipo de controladores combinan técnicas de control adaptativo y robusto [Sin85],[Slo87],[Ree88],[Alo04]. Básicamente estas técnicas tratan de dar robustez a entradas de control adaptativo basadas en la estimación de parámetros dinámicos del sistema.

Una de las ventajas del control robusto frente al adaptativo es que no se precisa tiempo para la sintonización de parámetros del controlador para hacer frente a las variaciones del sistema. Además, su implementación es mucho más simple. Sin embargo, el rango de aplicación respecto a las incertidumbres presentes en el manipulador es mucho más pequeño que en aquel caso.

1.6.2.1 Control robusto basado en linealización por realimentación

Aunque hay esquemas de control que utilizan linealización local para conseguir un sistema lineal [Hun83],[Des85],[Tor00], la mayoría de controladores se basan en las técnicas de linealización por realimentación y en la propiedad de parametrización lineal para conseguir un modelo lineal del manipulador. Se plantea en esta sección este método aplicado al control robusto.

A partir de la ecuación de la dinámica del manipulador (1-2), se puede obtener un sistema lineal en el error:

$$\dot{\xi} = \mathbf{H}\xi + \mathbf{D}\varpi \quad (1-36)$$

siendo:

$$\begin{aligned} \xi &= \begin{bmatrix} \tilde{\theta} & \tilde{\dot{\theta}} \end{bmatrix}^T \\ \mathbf{H} &= \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ \mathbf{D} &= \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \\ \varpi &= \ddot{\theta}_d - \mathbf{B}^{-1}(\theta) [\tau - \mathbf{h}(\theta, \dot{\theta}) - \mathbf{c}(\theta)] \end{aligned} \quad (1-37)$$

Desde este punto de vista, el problema de control se reduce a encontrar ϖ de forma que el sistema global cumpla las especificaciones de diseño deseadas. El controlador lineal que resuelve el problema habitualmente [Fre82] tiene la forma:

$$\varpi = \mathbf{K}\xi \quad (1-38)$$

Para evitar la no linealidad del sistema, dicha no linealidad se pasa al controlador (es el conocido esquema de linealización por realimentación), lo cual conlleva al siguiente controlador no lineal:

$$\boldsymbol{\tau} = \mathbf{B}(\boldsymbol{\theta})(\ddot{\boldsymbol{\theta}}_d - \boldsymbol{\varpi}) + \mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{c}(\boldsymbol{\theta}) \quad (1-39)$$

que, por comodidad, se expresa de la siguiente forma:

$$\boldsymbol{\tau} = \mathbf{B}(\boldsymbol{\theta})(\ddot{\boldsymbol{\theta}}_d - \boldsymbol{\varpi}) + \mathbf{n}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \quad (1-40)$$

donde los términos de aceleración centrífuga y de Coriolis y el término gravitatorio se aúnan en un único vector:

$$\mathbf{n}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{c}(\boldsymbol{\theta}) \quad (1-41)$$

De las expresiones (1-36) y (1-38), teniendo en cuenta (1-37) y aplicando el controlador no lineal (1-39), se obtiene la siguiente expresión de segundo orden para el lazo cerrado:

$$\ddot{\tilde{\boldsymbol{\theta}}} + \mathbf{K}_d \dot{\tilde{\boldsymbol{\theta}}} + \mathbf{K}_p \tilde{\boldsymbol{\theta}} = \mathbf{0} \quad (1-42)$$

donde se ha tenido en cuenta que:

$$\mathbf{K} = [\mathbf{K}_p \quad \mathbf{K}_d]^T \quad (1-43)$$

La ley de control robusto que se aplica a este sistema surge del hecho de que, en la práctica, debido a las incertidumbres presentes en el sistema o al desconocimiento de sus parámetros de inercia, sólo se puede dar una estimación de la matriz de inercia, $\hat{\mathbf{B}}(\boldsymbol{\theta})$, así como de los vectores de fuerza de Coriolis y centrífuga, $\hat{\mathbf{h}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$, y de fuerza gravitatoria, $\hat{\mathbf{c}}(\boldsymbol{\theta})$. De la misma forma que (1-41), $\hat{\mathbf{n}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \hat{\mathbf{h}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \hat{\mathbf{c}}(\boldsymbol{\theta})$, con lo que el controlador no lineal (1-39) queda de la forma:

$$\boldsymbol{\tau} = \hat{\mathbf{B}}(\boldsymbol{\theta})(\ddot{\boldsymbol{\theta}}_d - \boldsymbol{\varpi}) + \hat{\mathbf{n}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \quad (1-44)$$

Esta expresión lleva al sistema en lazo cerrado tener la siguiente expresión [Spo87a]:

$$\dot{\boldsymbol{\xi}} = \mathbf{H}\boldsymbol{\xi} + \mathbf{D}(\boldsymbol{\varpi} + \boldsymbol{\eta}) \quad (1-45)$$

siendo

$$\begin{aligned} \boldsymbol{\eta} &= (\mathbf{I} - \mathbf{B}^{-1}\hat{\mathbf{B}})(\ddot{\boldsymbol{\theta}}_d - \boldsymbol{\varpi}) - \mathbf{B}^{-1}\tilde{\mathbf{n}} \\ \tilde{\mathbf{n}} &= \hat{\mathbf{n}} - \mathbf{n} \end{aligned} \quad (1-46)$$

El término $\boldsymbol{\eta}$ representa una perturbación en el sistema en lazo cerrado debido a la incertidumbre presente en el mismo. Esta incertidumbre es consecuencia de dinámicas no modeladas, de incertidumbre o variación en los parámetros dinámicos del sistema e incluso del ruido de medida.

En este caso, el problema de control queda formulado de forma que se debe encontrar ϖ tal que el error sea estable. Para probar la estabilidad del error hay que hacer ciertas consideraciones sobre las incertidumbres, las cuales son:

$$\|\mathbf{I} - \mathbf{B}^{-1}(\boldsymbol{\theta})\hat{\mathbf{B}}(\boldsymbol{\theta})\| \leq \alpha \leq 1 \quad (1-47)$$

$$0 < B_m \leq \|\mathbf{B}^{-1}(\boldsymbol{\theta})\| \leq B_M \quad (1-48)$$

$$\|\tilde{\mathbf{n}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\| \leq \beta_0 + \beta_1 \|\xi\| + \beta_2 \|\xi\|^2 \quad (1-49)$$

donde α , B_m , B_M , β_0 , β_1 y β_2 son constantes positivas que dependen del tamaño de las incertidumbres. En caso de que hubiera en la estructura del robot articulaciones de tipo prismático, la asunción (1-49) debe variar. Escogiendo la forma habitual (1-38) para la ley de control, la ecuación (1-45) queda:

$$\dot{\xi} = (\mathbf{H} - \mathbf{DK})\xi + \mathbf{D}\eta = \tilde{\mathbf{H}}\xi + \mathbf{D}\eta \quad (1-50)$$

donde

$$\tilde{\mathbf{H}} = \mathbf{H} - \mathbf{DK} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K}_p & -\mathbf{K}_d \end{bmatrix} \quad (1-51)$$

Una de las formas de acometer este control es utilizando realimentación de variables de estado, de forma que se elige \mathbf{K} para ubicar los polos del lazo cerrado lo suficientemente lejos del origen, garantizando de esta forma [Fre82] la estabilidad en presencia de la incertidumbre $\boldsymbol{\eta}$.

Otra forma [Tar87] es añadir un bucle extra para corregir los efectos de la incertidumbre $\boldsymbol{\eta}$. Otros autores añaden una acción PI [Cai89] o PID [Ari83],[Gil87] para añadir robustez al controlador anterior.

1.6.2.2 Control robusto basado en la propiedad de pasividad

Retomando las ecuaciones dinámicas del manipulador (1-2), se puede encontrar una matriz $\mathbf{h}'(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ que cumpla (1-31) de forma que se puede demostrar [Kel03] que $\dot{\mathbf{B}}(\boldsymbol{\theta}) - 2\mathbf{h}'(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ es una matriz antisimétrica².

Con esta propiedad se puede demostrar que el modelo dinámico es un mapa pasivo desde los pares aplicados por los actuadores $\boldsymbol{\tau}$ hasta las velocidades $\dot{\boldsymbol{\theta}}$. De forma matemática, esta propiedad se expresa como que para todo $T > 0$ existe una constante positiva δ tal que se cumple:

² Una matriz \mathbf{A} es antisimétrica si $\mathbf{A} = -\mathbf{A}^T$.

$$\int_0^T \dot{\boldsymbol{\theta}}^T(t) \boldsymbol{\tau}(t) dt \geq -\delta \quad (1-52)$$

Basado en este teorema se puede asegurar la estabilidad del lazo cerrado, cuando se realiza un control de velocidad del motor. Para garantizar la estabilidad en el caso de la posición, se debe hacer uso del teorema desde el par $\boldsymbol{\tau}$ hasta una señal \mathbf{r} , versión filtrada de las velocidades [Ort88],[Slo88]. En ese caso, la ley de control viene dada por la siguiente expresión:

$$\boldsymbol{\tau} = \mathbf{B}(\boldsymbol{\theta})\mathbf{a} + \mathbf{h}'(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\mathbf{v} + \mathbf{c}(\boldsymbol{\theta}) - \mathbf{K}\mathbf{r} \quad (1-53)$$

donde

$$\begin{aligned} \mathbf{v} &= \dot{\boldsymbol{\theta}} - \mathbf{r} \\ \mathbf{a} &= \dot{\mathbf{v}} \\ \mathbf{r} &= \left(s\mathbf{I} + \frac{\mathbf{K}(s)}{s} \right) \tilde{\boldsymbol{\theta}} = \mathbf{F}(s)\tilde{\boldsymbol{\theta}} \end{aligned} \quad (1-54)$$

Sin embargo, esta ley de control no contempla las discrepancias entre el modelo estimado del manipulador y el sistema real. Para robustecer este lazo de control, se considera una versión modificada de la anterior:

$$\boldsymbol{\tau} = \boldsymbol{\Lambda}(s)\dot{\boldsymbol{\theta}} + \mathbf{u} \quad (1-55)$$

donde con una apropiada elección de $\boldsymbol{\Lambda}(s)$ y \mathbf{u} se puede aplicar el teorema de pasividad y demostrar la estabilidad del error [Abd91].

1.6.2.3 Control de estructura variable

El control de estructura variable ha sido aplicado a muchos procesos no lineales [Gao93]. Su principal objetivo [Hun93] es llevar al error hacia una superficie de cambio (*switching surface*), a partir de la cual el sistema se dice que permanece en un modo deslizante (*sliding mode*) en el que no se ve afectado por incertidumbres en el modelo o perturbaciones al sistema.

Hay numerosa bibliografía referente a la aplicación de este tipo de control en manipuladores [Fu90],[Leu91],[Su93],[Men02]. En una de las primeras aproximaciones de la aplicación de controladores de estructura variable a robots [You78], la ley de control propuesta tiene la forma:

$$\tau_i = \begin{cases} \tau_i^+ & , \text{ si } s_i(\tilde{\theta}_i, \dot{\theta}_i) > 0 \\ \tau_i^- & , \text{ si } s_i(\tilde{\theta}_i, \dot{\theta}_i) < 0 \end{cases} \quad (1-56)$$

donde las superficies de cambio s_i están definidas como:

$$s_i(\tilde{\theta}_i, \dot{\theta}_i) = k_i \tilde{\theta}_i + \dot{\theta}_i \quad (1-57)$$

donde k_i son constantes positivas y el índice i se refiere a cada uno de los grados de libertad del manipulador. Una vez establecidas las cotas en las incertidumbres, se pueden definir τ_i^+ y τ_i^- para llevar el error del sistema en lazo cerrado a cero.

Este controlador da solución al problema de posicionamiento, pero no al de seguimiento. En ese caso, se establecen superficies de cambio variables en el tiempo, de forma que, en un primer paso, se lleva al sistema al modo deslizante deseado y después, en un segundo paso, se elige entre un conjunto de superficies variables en el tiempo, con el objetivo que el sistema sea insensible a perturbaciones y dinámicas no modeladas o con incertidumbres.

En este caso, el controlador tiene la forma:

$$\boldsymbol{\tau} = \hat{\mathbf{B}}(\boldsymbol{\theta}) \left[\ddot{\boldsymbol{\theta}}_d + \mathbf{K}_d \dot{\boldsymbol{\theta}} + \mathbf{K}_p \tilde{\boldsymbol{\theta}} + \boldsymbol{\psi}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, t) \right] + \hat{\mathbf{h}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \hat{\mathbf{c}}(\boldsymbol{\theta}) \quad (1-58)$$

donde

$$\begin{aligned} \mathbf{K}_p &= \Lambda^2 \\ \mathbf{K}_d &= 2\Lambda \end{aligned} \quad (1-59)$$

siendo Λ una matriz diagonal de elementos positivos variables en el tiempo y $\boldsymbol{\psi}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, t)$ un término no lineal determinado por las cotas de incertidumbres presentes en el sistema.

1.6.2.4 Control robusto por saturación

Este tipo de controlador robusto ha sido ampliamente utilizado en el control de manipuladores. En esta tesis también se abordará el problema de dar robustez a los algoritmos de control propuestos en base a esta aproximación, por lo cual las expresiones serán definidas en detalle en posteriores capítulos.

A modo de reseña, comentar que la ley de control aplicada en este caso es [Spo87a]:

$$\boldsymbol{\tau} = \hat{\mathbf{B}}(\boldsymbol{\theta}) \left[\ddot{\boldsymbol{\theta}}_d + \mathbf{K}_d \dot{\boldsymbol{\theta}} + \mathbf{K}_p \tilde{\boldsymbol{\theta}} + \mathbf{y}_r \right] + \hat{\mathbf{h}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \hat{\mathbf{c}}(\boldsymbol{\theta}) \quad (1-60)$$

donde:

$$\mathbf{y}_r = \begin{cases} \frac{\rho}{\|\mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}\|} \mathbf{D}^T \mathbf{Q} \boldsymbol{\xi} & , \text{ si } \|\mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}\| \geq \varepsilon \\ \frac{\rho}{\varepsilon} \mathbf{D}^T \mathbf{Q} \boldsymbol{\xi} & , \text{ si } \|\mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}\| < \varepsilon \end{cases} \quad (1-61)$$

siendo ρ el parámetro de cota de incertidumbre, el cual viene dado tras dar ciertas estimaciones sobre las incertidumbres presentes en el sistema; \mathbf{Q} es una matriz definida positiva y ε un parámetro que delimita una banda de error donde se lleva el sistema.

En cierta medida, esta aproximación es muy parecida a la anterior, dado que se intenta llevar al sistema a un modo deslizante de error cero, aunque se opta por una ley de control discontinua para evitar el problema de *chattering* que aparece cuando el error se aproxima a cero, lo cual puede excitar dinámicas de altas frecuencias no modeladas. Esta aproximación hace que se pierda la estabilidad global o asintótica del lazo cerrado. En vez de eso, se asegura una estabilidad menos restrictiva, denominada *acotación uniforme ulterior* (del inglés *uniform ultimate boundedness*) [Cor81].

Otros tipos de controladores basados en esta aproximación pueden encontrarse en [Spo92],[Liu94],[Koo94] y [Jar96], donde se aplica la propiedad de parametrización lineal vista en 0.

1.7 Cotas de incertidumbre

En las secciones anteriores se han introducido las diversas técnicas de control empleadas en manipuladores rígidos. Así se han descrito desde las más sencillas, basadas en control PID descentralizado, cuando el problema de control se puede considerar monoarticular, hasta las técnicas más complejas, basadas en control adaptativo y robusto, que tienen en cuenta los diversos aspectos de control presentes en manipuladores reales, como son las incertidumbres en el modelo y la presencia de dinámicas no modeladas.

Sin embargo, surgen una serie de dificultades que todavía no han sido resueltas eficazmente en la literatura hasta la fecha, las cuales se exponen a continuación.

De la aplicación de técnicas de control robusto, surgen definiciones de controladores que aseguran la estabilidad del sistema en lazo cerrado, en presencia de incertidumbres, para un amplio conjunto de manipuladores.

Sin embargo, estas técnicas se basan en el conocimiento o estimación de las incertidumbres presentes en el modelo. Mediante estos parámetros de acotación se pueden estimar los diversos parámetros involucrados en la acción de control robusta.

Habitualmente, para estimar las incertidumbres, se hace uso de información del sistema que se puede conocer antes de controlar el mismo, como puede ser el modelo estimado del manipulador, las trayectorias de referencia para el mismo y ciertas restricciones derivadas de la mecánica del sistema, como pueden ser las velocidades máximas de las articulaciones o las aceleraciones máximas que pueden imprimir los actuadores. Con esta

información y el estudio en simulación del sistema, se obtienen finalmente los parámetros que se necesitan para diseñar completamente la ley de control.

En la mayoría de los casos, esto supone una desventaja, puesto que hay que realizar de forma pormenorizada este estudio en simulación, o bien estimar las cotas de incertidumbre.

1.8 Variabilidad en los parámetros dinámicos y masas de transporte

Asociado al problema anterior de la estimación de cotas de incertidumbres se encuentra el caso de manipuladores cuyos parámetros dinámicos pueden variar, por ejemplo, con el uso del mismo (por ejemplo, rozamientos y holguras mecánicas), siendo dificultosa la estimación de una cota para los mismos. Asimismo se tiene el caso de manipuladores que pueden operar con masas de transporte variables y totalmente desconocidas.

En ambos casos, las estrategias propuestas hasta el momento siempre hablan de una estimación de los parámetros dinámicos y sus desviaciones máximas. De esta forma se diseñan controladores robustos para asegurar el buen comportamiento dentro de esas zonas de variación.

Por ejemplo, en el caso de masas de transporte, algunos autores la suponen conocida o bien que puede ser calculada a través de la modificación del comportamiento dinámico del manipulador. Otros suponen situaciones de masa de transporte acotadas entre un valor mínimo y uno máximo. Otros, incluso, experimentan con varias situaciones de carga, de forma que el controlador realice una especie de interpolación entre su comportamiento para las distintas situaciones de masa previamente estudiadas [Car95].

En definitiva, estas situaciones de control no están completamente resueltas en la literatura, proponiéndose siempre soluciones con buen comportamiento dinámico dentro de un rango de variabilidad limitado, ya sea para los parámetros dinámicos como para la masa de transporte.

1.9 Restricciones de movimiento de las articulaciones

Todas estas técnicas de control no incluyen de forma sencilla las restricciones de movimiento de las articulaciones. La resolución del problema cinemático, o planificación de trayectorias, es la forma habitual de tener en cuenta las restricciones de movimiento del manipulador, ya sea en el posicionamiento de las articulaciones o bien en la velocidad máxima que puedan alcanzar las mismas.

Sin embargo, en presencia de incertidumbres, el comportamiento real del manipulador puede desviarse del comportamiento predefinido, por lo que dichas restricciones pueden superarse. En ese caso sería conveniente una incorporación sencilla de las restricciones del sistema al algoritmo de control.

1.10 Algoritmos de bajo costo computacional

Otra de las dificultades que se plantean en la implementación real de los distintos algoritmos de control es su eficiencia computacional. Esta dificultad surge de los períodos de muestreo empleados en el control dinámico de los manipuladores, como veremos a continuación.

En la resolución del problema de control multiarticular, se adoptan varias soluciones que pasan por una cancelación de la dinámica no lineal del manipulador. Para ello se aplican entradas basadas en el modelo dinámico del robot. Para tener un buen comportamiento de estas técnicas, se deben asegurar períodos de muestreo lo suficientemente pequeños. Así se puede considerar que las entradas discretas aplicadas se asemejan a las entradas para el modelo continuo, las cuales cancelan la dinámica no lineal del mismo.

De igual forma ocurre en el esquema de par calculado adaptativo, donde se requieren períodos de muestro pequeños para asegurar que la identificación de la dinámica no lineal del sistema es válida para su posterior cancelación.

Los períodos de muestreo utilizados varían según las velocidades de movimiento del manipulador. Lo más habitual es que se tengan períodos de muestreo del orden de centésimas de segundo, para los manipuladores más lentos, a milésimas de segundo, para los más rápidos y de uso industrial.

De esta forma, los algoritmos de control que a partir de este momento se planteen, deben tener en cuenta que su costo computacional debe ser bajo, para disponer del tiempo suficiente para llevar a cabo la cancelación de la dinámica no lineal de la planta. Es por ello que los algoritmos que se proponen en esta tesis son algoritmos de bajo costo computacional.

1.11 Portabilidad de los algoritmos hacia otro tipo de robots

Tal y como se comentaba en la sección dedicada a problemas abiertos en el control de manipuladores (ver 1.2.2), en la robótica de manipulación existen multitud de problemas abiertos que los algoritmos clásicos no dan una solución completa.

Uno de dichos problemas es el concerniente al control de manipuladores flexibles [Boo84],[Spo87b]. Estos sistemas subactuados son una alternativa que permite aumentar la relación carga/peso de los robots convencionales de

uso industrial. Asimismo, son capaces de efectuar manipulaciones de objetos a mayor velocidad que los robots rígidos. El principal problema de estos robots es que la exactitud de manipulación cuando operan a altas velocidades o cuando interactúan con fuerzas de contacto elevadas, es afectada en forma significativa por su elasticidad. Por tal motivo al llevar a cabo tareas complejas, debe considerarse la flexibilidad del manipulador en el diseño de los algoritmos de control. El requisito para esto es contar con un modelo efectivo de la dinámica del manipulador.

Por tanto, la portabilidad de los algoritmos de control presentados para los manipuladores rígidos hacia el caso de manipuladores flexibles no es un problema de fácil solución. Esta característica también sería deseable para el caso de manipuladores ligeros y otras estructuras de manipuladores, las cuales se empiezan a introducir en la industria.

En este trabajo se intenta dar una respuesta a este problema a través de algoritmos de control autosintonizables, los cuales se adaptan en cada caso a la dinámica del robot, sin tener en cuenta previamente sus características morfológicas, mecánicas o dinámicas.

1.12 Conclusiones

Como se ha descrito en este capítulo, las soluciones de control aportadas para los manipuladores rígidos son variadas, y van desde los controladores PID y sus variantes, hasta algoritmos basados en control adaptativo y robusto.

Estas técnicas de control robusto proveen un buen comportamiento del lazo cerrado para un conjunto amplio de manipuladores y en presencia de incertidumbres.

Sin embargo, hay problemas de control aún no completamente solucionados. Por ejemplo, las incertidumbres deben estar acotadas para que el controlador sea eficiente y/o se asegure la estabilidad del lazo cerrado. Con esas cotas se puede proceder al cálculo de la acción de control. Además, este tipo de control no tiene en cuenta las posibles restricciones de movimiento, a las que se puede llegar debido a esas incertidumbres.

En esta tesis se avanza sobre los algoritmos propuestos hasta el momento, proponiendo estrategias de control robustas, computacionalmente eficientes, que resuelven estos problemas y que evitan gran parte del cálculo previo a realizar para la definición de la acción de control sobre el manipulador. Estas técnicas, tal y como serán planteadas, son susceptibles de aplicarse a otras estructuras de manipulador, como robots ligeros o robots flexibles, dado que el controlador se adapta a las variaciones dinámicas del manipulador.

Parte I

**Rechazo a perturbaciones
en robots manipuladores**

Técnicas de control robusto para el rechazo de perturbaciones

Las técnicas de control robusto han sido utilizadas con éxito en numerosos trabajos relativos al control de manipuladores de robot, ya que incorporan a los lazos de control habituales términos que confieren robustez frente a perturbaciones no contempladas en el modelo. Dichas perturbaciones se deben principalmente a dos razones. La primera son las discrepancias entre el modelo estimado y la planta real. Esto se traduce en incertidumbres en los parámetros del modelo o la existencia de dinámicas no modeladas, términos de fricción, etc. La segunda es que el modelo es dependiente de las masas de carga en el efector final, las cuales no son siempre conocidas a priori. Esto hace que la cancelación de términos no lineales presentes en el modelo, para conseguir un modelo lineal del manipulador y aplicar técnicas de control habituales, sea imperfecta. Uno de los esquemas robustos de más éxito se basa en añadir un término a la entrada de control para conseguir propiedades de rechazo a las perturbaciones mencionadas. Este esquema incluye un parámetro de diseño, el parámetro de cota de incertidumbre ρ , que debe tomar un valor que cumpla con ciertos criterios de estabilidad del controlador definidos previamente. Dicha elección resulta fundamental en el comportamiento del controlador robusto. En este capítulo será introducida esta técnica de control, la cual será mejorada en el siguiente capítulo con una técnica autoadaptativa que consiga un rechazo más eficiente de las perturbaciones que afecten al manipulador.

2.1 Introducción

Las técnicas de control de manipuladores vistas en el capítulo anterior tienen como objetivo resolver eficientemente el problema del seguimiento de trayectorias, con un bajo coste computacional e incorporando las restricciones a las que está sometido el sistema. Estas técnicas utilizan el modelo del

sistema para obtener la ley de control final a aplicar. Sin embargo, no se han contemplado las desviaciones del modelo estimado para el robot respecto al sistema real. Este es un problema bastante común a la hora de abordar el control de manipuladores, sobre todo en lo concerniente a las incertidumbres paramétricas, a la incertidumbre en la masa de carga, así como a términos de rozamiento o fricción no contemplados en el modelo estimado.

Como es bien sabido, el problema de controlar el movimiento de robots manipuladores ha sido abordado tradicionalmente con controladores PID, controladores PD con compensación de gravedad, técnicas basadas en control dinámico inverso e incluso estrategias basadas en control difuso [Jin98],[Tor00],[Lon03].

Los controladores PID y difuso no tienen en cuenta las ecuaciones dinámicas del sistema, pero sólo son excepciones, porque son usados únicamente para resolver el problema de regulación, por ejemplo, el movimiento de un punto a otro punto. El resto de controladores, empleados para solucionar el problema del seguimiento de trayectoria, utilizan el modelo dinámico para calcular la acción de control, por ejemplo el término de gravedad en el caso de compensación de gravedad o el término no lineal en el caso de control dinámico inverso, denominado en robótica como linealización por realimentación.

El algoritmo de linealización por realimentación está basado en un conocimiento perfecto del modelo del robot y sus parámetros dinámicos. Se consigue una cancelación perfecta de la dinámica no lineal si se satisfacen estas dos premisas. Los controladores lineales presentan en ese caso un rendimiento satisfactorio [Fre82],[Kre89].

Otras técnicas de control no utilizan esta linealización exacta por realimentación sino una linealización local en torno a la trayectoria deseada, para así obtener un sistema linealizado y poder aplicar de nuevo un controlador lineal [Des85],[Tor02]. Otros algoritmos hacen uso de la propiedad de parametrización lineal del robot rígido de n articulaciones para obtener un modelo lineal del sistema y aplicar técnicas de control lineal [Ree88],[Spo92].

Las técnicas de control robusto que se emplean en este trabajo tratan de solucionar el problema de la cancelación imperfecta de la dinámica no lineal debida a incertidumbres. Se considera que existen varias clases de incertidumbres en un sistema robótico [Ish01], tal y como se describe a continuación:

- a) incertidumbre paramétrica, debida a variaciones de la carga.
 - b) error de modelado, tal como dinámicas no modeladas o términos de fricción (expresado como una función del ángulo de la articulación y su velocidad)
 - c) incertidumbres tales como el ruido de medida.
-

Las perturbaciones no contempladas en la clasificación anterior, tales como el ruido de proceso o de medida, son abordadas por el controlador lineal. Las incertidumbres, en cambio, están siempre presentes en el sistema, y son abordadas por los algoritmos de control propuestos, tal y como son las acciones robusta o adaptativa.

Se han propuesto un extenso número de trabajos relacionados con esquemas de control adaptativo [Ort89],[Slo87], esquemas de control robusto [Slo85],[Spo87a],[Daw92] e incluso esquemas de control híbrido [Su97] para tratar con estas incertidumbres.

Concretamente, la mayoría de los controladores robustos están basados en el método directo de Lyapunov [Slo87]. Estos esquemas añaden un término robusto a la entrada de control, el cual trata de compensar las discrepancias entre el modelo estimado y el modelo real del sistema. Para hacer esto se emplea un bucle interno basado en el esquema de linealización por realimentación. Se escoge además una ley de control estabilizante para el sistema linealizado. La clásica ley de control PD con un término lineal de prealimentación asegura un comportamiento satisfactorio para dar solución al problema del seguimiento.

Finalmente, se añade una acción robusta definida mediante el método directo de Lyapunov. Esto resulta en un nuevo término añadido a la entrada del sistema linealizado que corrige la compensación incorrecta derivada de las incertidumbres en el modelo. En la figura 2-1 se muestra un esquema del controlador robusto.

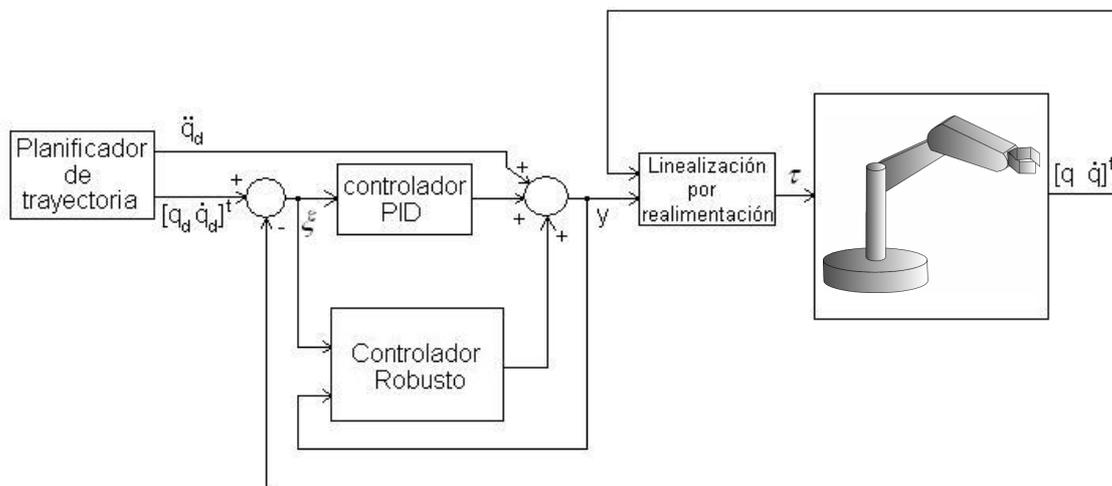


Figura 2-1: Esquema del control adaptativo robusto.

En los siguientes apartados se describirán los distintos términos de la acción de control. En el apartado 2.2 se describe el bucle interno de linealización por realimentación utilizado para obtener un modelo lineal del sistema, mientras que en el apartado 2.3 se introduce la ley de control robusta. Finalmente, en el apartado 2.4 se exponen algunas técnicas para la elección de

un parámetro fundamental en el diseño de la acción robusta, como es el parámetro de cota de incertidumbre.

2.2 Linealización por realimentación.

Para la obtención de una ley de control aplicable en el sistema real, se considera un tratamiento discreto del sistema. Para ello se tiene en cuenta que, con una elección del período de muestreo lo suficientemente pequeña, las ecuaciones dinámicas del manipulador se pueden expresar de la forma:

$$\mathbf{B}(\boldsymbol{\theta}_k)\ddot{\boldsymbol{\theta}}_k + \mathbf{h}(\boldsymbol{\theta}_k, \dot{\boldsymbol{\theta}}_k) + \mathbf{c}(\boldsymbol{\theta}_k) = \boldsymbol{\tau}_k \quad (2-1)$$

siendo $\mathbf{B}(\boldsymbol{\theta}_k)$ la matriz de inercia, $\mathbf{h}(\boldsymbol{\theta}_k, \dot{\boldsymbol{\theta}}_k)$ el vector de fuerza centrífuga y de Coriolis, $\mathbf{c}(\boldsymbol{\theta}_k)$ el vector de fuerza gravitacional y $\boldsymbol{\tau}_k$ el par aplicado a cada articulación. El subíndice k indica el instante de muestreo, mientras que $\boldsymbol{\theta}_k$ y $\dot{\boldsymbol{\theta}}_k$ se refieren a la posición y velocidad medidas en dicho instante.

En esta ecuación no se tiene en cuenta el término de fricciones, dependiente de las velocidades $\dot{\boldsymbol{\theta}}_k$, pero puede ser incluido sin más problemas en las ecuaciones, y luego tenerse en cuenta en la entrada de linealización propuesta más adelante. También puede considerarse que dicho término es una perturbación al sistema, la cual es corregida mediante el algoritmo de control propuesto.

El primer paso para el diseño de la ley de control es la obtención de un modelo lineal del sistema. Para ello se utiliza el conocido esquema de desacoplamiento por inversión del modelo (o linealización por realimentación), explicado en el capítulo 1. Dicha linealización se basa en considerar el sistema como un sistema multivariable no lineal.

Una vez obtenido el estado completo del robot en cada instante de tiempo, y conociendo su modelo teórico, se puede obtener un modelo lineal del sistema. Para hacer esto se define la entrada al modelo no lineal como:

$$\boldsymbol{\tau}_k = \mathbf{B}(\boldsymbol{\theta}_k)\mathbf{y}_k + \mathbf{h}(\boldsymbol{\theta}_k, \dot{\boldsymbol{\theta}}_k) + \mathbf{c}(\boldsymbol{\theta}_k) \quad (2-2)$$

donde \mathbf{y}_k es la nueva entrada al modelo lineal resultante. En general, se obtiene una cancelación satisfactoria de los términos no lineales cuando se tiene una buena estimación del modelo real del sistema y cuando se considera un periodo de muestreo lo suficientemente pequeño. Esto conduce al siguiente modelo de segundo orden, lineal y desacoplado con respecto a la nueva entrada:

$$\mathbf{y}_k = \ddot{\boldsymbol{\theta}}_k \quad (2-3)$$

donde cada componente $y_{i,k}$ influye sólo en la variable de articulación $\theta_{i,k}$, independientemente del estado de las otras articulaciones.

La implementación del desacoplo por inversión del modelo (2-2) requiere un conocimiento preciso del modelo dinámico del sistema. Esta condición es difícil de alcanzar debido al conocimiento impreciso de los parámetros mecánicos, la existencia de fricciones no modeladas y el efecto de la masa de transporte en la dinámica del sistema.

El problema de control se reduce a encontrar una ley de control para la entrada lineal \mathbf{y}_k que lleve al sistema a un seguimiento eficaz de las trayectorias de consigna. Dicha ley se propone en el siguiente apartado.

2.3 Ley de control.

Una vez se tiene el sistema linealizado, se define la ley de control que persigue el seguimiento de las trayectorias de referencia. Dado que consideramos incertidumbres en el modelo del sistema, debemos proveer una ley de control con los siguientes objetivos:

- seguimiento eficaz de las trayectorias de consigna,
- robustez del sistema en lazo cerrado frente a incertidumbres en el modelo de sistema considerado.

Para la consecución de estos dos objetivos, se propone una ley de control que consta de dos términos: un primer término estabilizante y una acción robusta que cancele los errores originados por la inexactitud en la cancelación de los términos no lineales del paso anterior (linealización por realimentación). Un esquema de esta ley de control se puede observar en la figura 2-2.

2.3.1 Ley de control PD estabilizante

Para conseguir un rechazo a perturbaciones del sistema linealizado, debemos establecer una ley de control apropiada que además le confiera al sistema la propiedad de estabilidad frente a dichas perturbaciones. Denominamos a esta ley de control *estabilizante* por dotar al lazo cerrado de control con esta característica.

La ley de control estabilizante, a su vez, está formada por dos términos: un controlador PD y un término *feedforward* que incluye las aceleraciones para la trayectoria de referencia, tal y como se expresa en la siguiente expresión:

$$\mathbf{y}_k = \ddot{\mathbf{0}}_{k,d} + \mathbf{K}_D \dot{\tilde{\mathbf{0}}}_k + \mathbf{K}_p \tilde{\mathbf{0}}_k \quad (2-4)$$

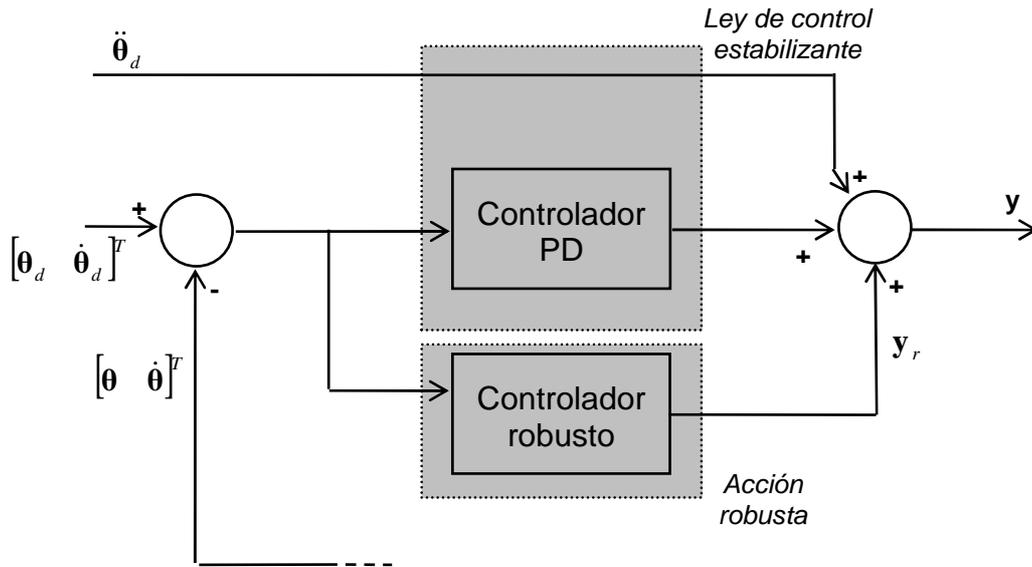


Figura 2-2 : Detalle de la ley de control propuesta para el sistema linealizado.

donde $\tilde{\theta}_k = \theta_{k,d} - \theta_k$, y de forma similar para sus derivadas temporales. El subíndice d indica que la variable a la que afecta es la referencia o consigna. Se dice que esta ley de control es estabilizante pues si sustituimos (2-4) en (2-3), obtenemos la ecuación dinámica del sistema en lazo cerrado, expresado en términos del error de seguimiento:

$$\ddot{\tilde{\theta}}_k + \mathbf{K}_D \dot{\tilde{\theta}}_k + \mathbf{K}_p \tilde{\theta}_k = 0 \quad (2-5)$$

Este sistema es asintóticamente estable si \mathbf{K}_p y \mathbf{K}_D son matrices definidas positivas. Además escogiendo una forma diagonal para estas matrices, el sistema resulta desacoplado. La forma habitual de elección de estas matrices es:

$$\mathbf{K}_p = \text{diag}\{w_{n1}^2, \dots, w_{nm}^2\}$$

$$\mathbf{K}_D = \text{diag}\{2\zeta_1 w_{n1}, \dots, 2\zeta_n w_{nm}\}$$

donde n es el número de articulaciones del manipulador. Cuando la posición y la velocidad inicial son nulas, \mathbf{K}_p y \mathbf{K}_D dan la velocidad de convergencia a cero del error de posición.

2.3.2 Acción robusta

El segundo término de la ley de control, la acción robusta, es añadido para corregir la compensación imperfecta del término no lineal en (2-1), dado por el control de dinámica inversa (2-2). En la asunción de que sólo pueden ser

obtenidas unas estimaciones de las matrices reales $\mathbf{B}(\boldsymbol{\theta}_k)$, $\mathbf{h}(\boldsymbol{\theta}_k, \dot{\boldsymbol{\theta}}_k)$ y $\mathbf{c}(\boldsymbol{\theta}_k)$, la ecuación (2-5) se convierte ahora en la siguiente expresión:

$$\ddot{\tilde{\boldsymbol{\theta}}}_k + \mathbf{K}_D \dot{\tilde{\boldsymbol{\theta}}}_k + \mathbf{K}_P \tilde{\boldsymbol{\theta}}_k = \boldsymbol{\eta}_k \quad (2-6)$$

donde $\boldsymbol{\eta}_k$ expresa las discrepancias entre los valores reales y los estimados de las matrices en el instante de tiempo k [Sci00]:

$$\boldsymbol{\eta}_k = (\mathbf{I} - \mathbf{B}_k^{-1} \hat{\mathbf{B}}_k) \mathbf{y}_k - \mathbf{B}_k^{-1} (\hat{\mathbf{n}}_k - \mathbf{n}_k) \quad (2-7)$$

donde

$$\mathbf{n}_k = \mathbf{n}(\boldsymbol{\theta}_k, \dot{\boldsymbol{\theta}}_k) = \mathbf{h}(\boldsymbol{\theta}_k, \dot{\boldsymbol{\theta}}_k) + \mathbf{c}(\boldsymbol{\theta}_k) \quad (2-8)$$

y las matrices con el símbolo \wedge indican que son estimaciones de las matrices presentes en el modelo real del sistema. A la vista de esto, para el sistema acoplado no lineal, el seguimiento con error nulo no está asegurado y la acción de control PD no es suficiente.

2.3.2.1 Aplicación del método directo de Lyapunov para la obtención del término de acción robusta

Siguiendo el conocido método directo de Lyapunov [Slo87], se puede diseñar un bucle de realimentación externo para el error que sea robusto a la incertidumbre $\boldsymbol{\eta}_k$, expresada por (2-7).

Dicho planteamiento se realiza para el caso continuo. Sin embargo, asumiendo que el período de muestreo es lo suficientemente pequeño, podemos obtener fácilmente una aproximación discreta de la ley de control obtenida.

A) Dinámica del error del sistema

El modelo continuo del sistema viene dado por:

$$\mathbf{B}(\boldsymbol{\theta}(t)) \ddot{\boldsymbol{\theta}}(t) + \mathbf{h}(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t)) + \mathbf{c}(\boldsymbol{\theta}(t)) = \boldsymbol{\tau}(t) \quad (2-9)$$

Este sistema tiene aplicada una entrada de linealización que viene dada por un conocimiento, más o menos aproximado, de las ecuaciones dinámicas del mismo. Dicha entrada de linealización viene dada por:

$$\boldsymbol{\tau}(t) = \hat{\mathbf{B}}(\boldsymbol{\theta}(t)) \mathbf{y}(t) + \hat{\mathbf{h}}(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t)) + \hat{\mathbf{c}}(\boldsymbol{\theta}(t)) \quad (2-10)$$

donde las variables con el símbolo \wedge designan la estimación a la correspondiente variable real. Si definimos:

$$\mathbf{n}(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t)) = \mathbf{h}(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t)) + \mathbf{c}(\boldsymbol{\theta}(t)) \quad (2-11)$$

y de igual forma para las variables estimadas, sustituyendo la entrada (2-10) en la ecuación (2-9), se tiene, obviando por comodidad en la formulación la dependencia de las coordenadas generalizadas, que:

$$\ddot{\boldsymbol{\theta}}(t) = \mathbf{B}^{-1} \hat{\mathbf{B}} \mathbf{y}(t) + \mathbf{B}^{-1} (\hat{\mathbf{n}} - \mathbf{n}) \quad (2-12)$$

De esta relación se llega, sumando y restando la entrada al sistema lineal $\mathbf{y}(t)$ al segundo término de (2-12), a la siguiente ecuación de entrada-salida del sistema linealizado:

$$\ddot{\boldsymbol{\theta}}(t) = \mathbf{y}(t) - \boldsymbol{\eta}(t) \quad (2-13)$$

donde:

$$\boldsymbol{\eta}(t) = (\mathbf{I} - \mathbf{B}^{-1} \hat{\mathbf{B}}) \mathbf{y}(t) - \mathbf{B}^{-1} (\hat{\mathbf{n}} - \mathbf{n}) \quad (2-14)$$

Este término mide las discrepancias entre el sistema real y el modelo estimado del mismo. Nótese que, con estimaciones exactas del modelo real, se tiene una contribución nula a la ecuación de entrada-salida del sistema.

Según el esquema propuesto, la entrada de control al sistema (2-10) viene dada por:

$$\mathbf{y}(t) = \ddot{\boldsymbol{\theta}}_d(t) + \mathbf{K}_D \dot{\boldsymbol{\theta}}(t) + \mathbf{K}_P \boldsymbol{\theta}(t) + \mathbf{y}_r(t) \quad (2-15)$$

con lo que se puede expresar el sistema de la siguiente manera:

$$\ddot{\boldsymbol{\theta}}(t) + \mathbf{K}_D \dot{\boldsymbol{\theta}}(t) + \mathbf{K}_P \boldsymbol{\theta}(t) + \mathbf{y}_r(t) = \boldsymbol{\eta}(t) \quad (2-16)$$

En términos del error $\boldsymbol{\xi}$, definido por:

$$\boldsymbol{\xi}_{2n \times 1} = \begin{bmatrix} \tilde{\boldsymbol{\theta}}_{n \times 1} & \dot{\tilde{\boldsymbol{\theta}}}_{n \times 1} \end{bmatrix}^T \quad (2-17)$$

(donde los subíndices indican la dimensión del vector) y obviando por claridad en la formulación la dependencia temporal, la ecuación (2-16) puede ser escrita de la siguiente forma:

$$\dot{\boldsymbol{\xi}} = \tilde{\mathbf{H}} \boldsymbol{\xi} + \mathbf{D}(\boldsymbol{\eta} - \mathbf{y}_r) \quad (2-18)$$

donde

$$\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K}_P & -\mathbf{K}_D \end{bmatrix}; \quad \mathbf{D} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \quad (2-19)$$

La ecuación (2-18) expresa la dinámica del error del sistema en lazo cerrado bajo la ley de control robusta propuesta. Nótese que la acción robusta se propone de esta forma para cancelar la incertidumbre producida por la estimación errónea de la planta real.

B) Obtención del término robusto

La aplicación del método directo de Lyapunov produce una expresión para la acción robusta y_r . Este método asegura que la solución será asintóticamente estable. Para ello se escoge la función candidata de Lyapunov dada por:

$$V(\xi) = \xi^T Q \xi > 0 \quad (2-20)$$

donde Q es una matriz simétrica definida positiva de dimensiones apropiadas. Para asegurar la estabilidad del sistema basta demostrar que su derivada evaluada a lo largo de las trayectorias para el error del sistema (2-18) es negativa. Dicha derivada viene dada por:

$$\dot{V} = \dot{\xi}^T Q \xi + \xi^T Q \dot{\xi} = \xi^T (\tilde{H}^T Q + Q \tilde{H}) \xi + 2\xi^T D Q (\eta - y_r) \quad (2-21)$$

Dado que \tilde{H} tiene todos sus autovalores con parte real negativa, es bien conocido el resultado que, para cualquier matriz simétrica definida positiva P , la ecuación:

$$\tilde{H}^T Q + Q \tilde{H} = -P \quad (2-22)$$

posee una solución única Q simétrica definida positiva. Luego:

$$\dot{V} = -\xi^T P \xi + 2\xi^T D Q (\eta - y_r) \quad (2-23)$$

El primer término de (2-23) es negativo. Para asegurar la estabilidad del sistema en el caso de que $z = D^T Q \xi \neq 0$, se debe escoger la acción robusta de tal forma que el segundo término de dicha ecuación sea igual o menor que cero. Para ello se adopta la solución:

$$y_r = \frac{\rho}{\|z\|} z \quad (2-24)$$

donde $\rho > 0$ es un parámetro de diseño, denominado parámetro de cota de incertidumbre (del inglés *uncertainty bound parameter*). El segundo término de (2-23) queda de la forma:

$$z^T \left(\eta - \frac{\rho}{\|z\|} z \right) = z^T \eta - \frac{\rho}{\|z\|} z^T z \leq \|z\| \|\eta\| - \rho \|z\| = \|z\| (\|\eta\| - \rho) \quad (2-25)$$

Con lo que eligiendo:

$$\rho > \|\boldsymbol{\eta}\| \quad (2-26)$$

se tiene que $\dot{V} < 0$ para todas las trayectorias del error del sistema en lazo cerrado.

2.3.2.2 Consideraciones prácticas del término de acción robusta.

Aplicando el método visto en el apartado anterior al caso discreto, se propone la siguiente expresión para el término de acción robusta:

$$\mathbf{y}_{k,r} = \frac{\rho}{\|\mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}_k\|} \mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}_k \quad (2-27)$$

donde $\mathbf{D}_{2n \times n} = [\mathbf{0}_{n \times n} \quad \mathbf{I}_{n \times n}]^T$, $\mathbf{Q}_{2n \times 2n}$ es una matriz definida positiva y ρ es el parámetro de cota de incertidumbre. De esta forma, la ley de control completa viene dada por:

$$\mathbf{y}_k = \ddot{\boldsymbol{\theta}}_{k,d} + \mathbf{K} \boldsymbol{\xi}_k + \mathbf{y}_{k,r} \quad (2-28)$$

siendo $\mathbf{K} = [\mathbf{K}_p \quad \mathbf{K}_D]$.

La ecuación (2-27) presenta problemas de divergencia cuando el error se aproxima a cero. Para evitarlo se utiliza la siguiente expresión:

$$\mathbf{y}_{k,r} = \begin{cases} \frac{\rho}{\|\mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}_k\|} \mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}_k & , \text{ si } \|\mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}_k\| \geq \varepsilon \\ \frac{\rho}{\varepsilon} \mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}_k & , \text{ si } \|\mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}_k\| < \varepsilon \end{cases} \quad (2-29)$$

donde ε es cierta constante que viene determinada por las pruebas finales con el sistema real. Esta ley de control robusta (2-29) no presenta las mismas características de robustez que (2-27). La ley original posee la propiedad de que todas las trayectorias del sistema bajo dicha ley de control alcanzan el subespacio $\mathbf{z}_k = \mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}_k = \mathbf{0}$, denominado *subespacio deslizante*, que depende de la matriz \mathbf{Q} de la función de Lyapunov escogida.

En este subespacio, la ley de control (2-27) cambia a una frecuencia idealmente infinita y el error tiende a cero con un transitorio que depende de \mathbf{Q} , \mathbf{K}_p y \mathbf{K}_D , y con una evolución temporal que depende de ρ .

Sin embargo, en el caso real, el controlador no conmutará su señal a una frecuencia infinita, sino que estará limitado por los actuadores y los elementos físicos involucrados en el control.

La ley de control modificada (2-29) elimina el problema de las altas frecuencias de conmutación de (2-27), a costa de dejar de garantizar convergencia del error a cero. Sin embargo sí asegura que el error está acotado, dado que las trayectorias tienden a un hiperplano en torno a $\mathbf{z}_k = \mathbf{0}$, cuyas dimensiones dependen de la elección de ε .

2.4 Elección del parámetro de cota de incertidumbre

El valor de la cota de incertidumbre ρ es importante para tener un buen comportamiento del sistema en lazo cerrado. Mediante un estudio de simulación se puede asegurar que un pequeño valor de ρ tiene como consecuencia que los resultados de seguimiento sobre las trayectorias deseadas para las articulaciones sean insatisfactorios, mientras que un valor muy grande de ρ lleva a una saturación de las entradas y consecuentemente a un mal comportamiento del sistema completo.

Considérese, como muestra, el experimento 2.1 de control de un robot manipulador de dos articulaciones, bajo la acción de un controlador robusto como el definido en el apartado 2.3.

La figura 2-3 muestra los resultados obtenidos con una acción robusta fija con $\rho=17$. La trayectoria real para la primera articulación es bastante similar a la trayectoria de referencia, pero los errores de posición son apreciables para la segunda articulación (cerca de 0.03 radianes en el peor caso) a partir del instante en el que el robot recoge la masa de carga y se inicia el regreso al punto inicial.

El modelo dinámico del robot cambia y la acción robusta permite que los errores de seguimiento no tomen valores elevados. Este valor de ρ es escogido después de varios intentos en simulación.

Si el valor de ρ se incrementa, los errores de posición y velocidad decrecen. Este resultado se puede ver en la figura 2-4, donde el valor del parámetro de diseño robusto es, consecutivamente, $\rho=5$, $\rho=12$ y $\rho=17$.

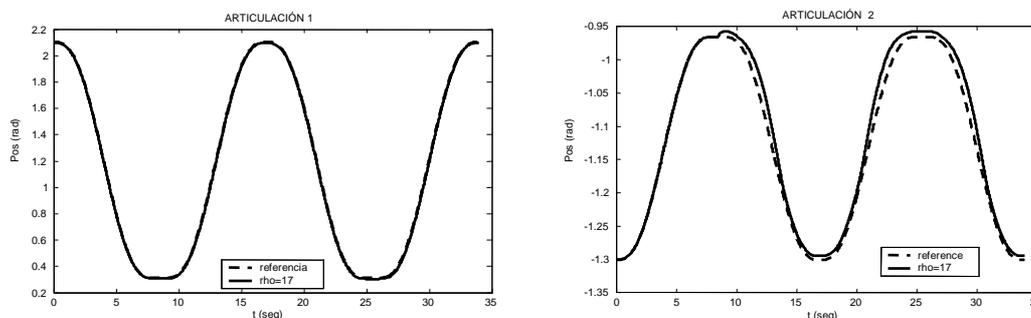


Figura 2-3: Trayectoria obtenida y deseada para las dos articulaciones con una acción robusta fija ($\rho=17$).

Experimento 2.1

Sistema: Manipulador de dos articulaciones, de masas 5.0 kg. y 4.5 kg., respectivamente, y longitudes 43 cm. cada una.

Controlador: Estrategia robusta de parámetros fijos.

* Acción PD: $\mathbf{K}_p = \text{diag}([155 \ 115])^1$, $\mathbf{K}_d = \text{diag}([12 \ 15])$.

* Acción robusta:

$$\varepsilon = 0.25,$$

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 10 & 0 & 1 & 0 \\ 0 & 30 & 0 & 5 \end{bmatrix}.$$

* Modelo del manipulador:

Manipulador de dos articulaciones obtenido de considerar la segunda y tercera articulación de un robot PUMA, que presenta incertidumbres paramétricas en masas y longitudes de las articulaciones.

Trayectoria de referencia:

- Trayectoria suave de 8 segundos de duración entre las posiciones 120.3° y 17.2° para la primera articulación y -74.55° y -55.6° para la segunda.

- En el punto final, el brazo toma una masa esférica de 1.5 kg. y 15 cm., retorna al punto inicial y, finalmente, de nuevo al punto final.

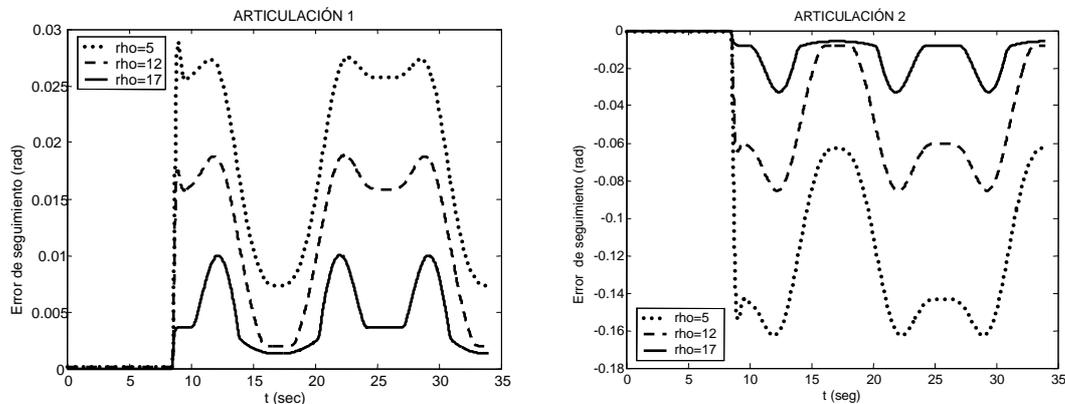


Figura 2-4: Comparación del error de seguimiento en el caso de utilizar un controlador robusto fijo con tres valores diferentes para ρ .

Si se sigue aumentando el valor del parámetro de cota de incertidumbre, los errores de seguimiento siguen decreciendo, pero las entradas a las articulaciones son cada vez mayores, con lo que empiezan a aparecer situaciones de saturación que conllevan un mal comportamiento del sistema en lazo cerrado.

Luego una elección adecuada de este parámetro es fundamental para obtener un buen comportamiento del sistema. En la literatura hasta la fecha este problema ha sido abordado de diferentes maneras. En primer lugar, del propio algoritmo propuesto por Spong [Spo92], se extrae una expresión para la elección de ρ al tratar de obtener una ley de control que estabilice el sistema en lazo cerrado.

En otros trabajos, la elección de la cota de incertidumbre no es mencionada o bien se obtiene de expresiones de máximo suponiendo los parámetros dinámicos y masa de transporte conocidos.

En los siguientes apartados se realiza un análisis de estas técnicas, sobre las que el presente trabajo introduce una innovación importante.

2.4.1 Algoritmo de Spong

En este trabajo se hace mención al algoritmo de Spong como el algoritmo de adaptación del parámetro de cota de incertidumbre, el cual se extrae de la propia formulación del controlador.

Del algoritmo propuesto por Spong [Spo92], se extrae una expresión para la elección de ρ al tratar de obtener una ley de control que estabilice el sistema en lazo cerrado. En concreto, se asumen ciertas las siguientes condiciones:

Para cualquier trayectoria planificada se puede encontrar una cota para las aceleraciones:

$$\sup_{t \geq 0} \|\ddot{\boldsymbol{\theta}}_d\| < Q_M, \quad \forall \ddot{\boldsymbol{\theta}}_d \quad (2-30)$$

Se puede encontrar una constante que acote las discrepancias entre la matriz de inercia del sistema real y la del modelo estimado:

$$\|\mathbf{I} - \mathbf{B}^{-1}(\boldsymbol{\theta})\hat{\mathbf{B}}(\boldsymbol{\theta})\| \leq \alpha \leq 1, \quad \forall \boldsymbol{\theta} \quad (2-31)$$

De hecho, teniendo las cotas superior e inferior para la inversa de la matriz de inercia:

$$0 < B_m \leq \|\mathbf{B}^{-1}(\boldsymbol{\theta})\| \leq B_M, \quad \forall \boldsymbol{\theta} \quad (2-32)$$

eligiendo para la estimación de la matriz de inercia:

$$\hat{\mathbf{B}} = \frac{2}{B_m + B_M} \mathbf{I} \quad (2-33)$$

se tiene que:

$$\|\mathbf{B}^{-1}(\boldsymbol{\theta})\hat{\mathbf{B}}(\boldsymbol{\theta}) - \mathbf{I}\| \leq \left\| B_M \frac{2}{B_M + B_m} \mathbf{I} - \mathbf{I} \right\| = \left\| \frac{B_M - B_m}{B_M + B_m} \right\| = \alpha < 1, \quad \forall \boldsymbol{\theta} \quad (2-34)$$

Se puede encontrar una cota para el término no lineal de la ecuación dinámica:

$$\|\hat{\mathbf{n}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) - \mathbf{n}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\| \leq \phi, \quad \forall \boldsymbol{\theta}, \dot{\boldsymbol{\theta}} \quad (2-35)$$

Esta cota es más difícil de evaluar en la práctica. Las posiciones y velocidades van a estar acotadas por sus valores máximos, por lo que $\|\hat{\mathbf{n}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\|$ está acotada. Podemos acotar la diferencia por un porcentaje a determinar de la norma $\|\hat{\mathbf{n}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\|$ evaluada sobre la trayectoria de referencia.

Bajo estas asunciones podemos acotar la incertidumbre η por:

$$\begin{aligned} \|\boldsymbol{\eta}\| &\leq \|\mathbf{I} - \mathbf{B}^{-1}\hat{\mathbf{B}}\| (\|\ddot{\boldsymbol{\theta}}_d\| + \|\mathbf{K}\|\|\xi\| + \|\mathbf{y}_r\|) + \|\mathbf{B}^{-1}\|\|\hat{\mathbf{n}} - \mathbf{n}\| \leq \\ &\leq \alpha Q_M + \alpha \|\mathbf{K}\|\|\xi\| + \alpha \rho + B_M \phi \end{aligned} \quad (2-36)$$

por lo que la condición (2-26) se puede reescribir tomando el peor caso para la incertidumbre:

$$\rho \geq \alpha Q_M + \alpha \|\mathbf{K}\|\|\xi\| + \alpha \rho + B_M \phi$$

que queda finalmente de la siguiente forma:

$$\rho \geq \frac{1}{1 - \alpha} (\alpha Q_M + B_M \phi + \alpha \|\mathbf{K}\|\|\xi\|) \quad (2-37)$$

que es la expresión para ρ que asegura la estabilidad del sistema en lazo cerrado. La expresión (2-37) puede ponerse de la siguiente forma:

$$\rho_k \geq \lambda_0 + \lambda_1 \|\xi_k\| \quad (2-38)$$

donde el índice k indica que ρ varía en cada instante de tiempo y $\|\xi_k\|$ es la norma-2 del vector de error.

Las constantes λ_0 y λ_1 se deducen rápidamente de la expresión (2-37):

$$\lambda_0 = \frac{1}{1-\alpha}(\alpha Q_M + B_M \phi) \quad (2-39)$$

$$\lambda_1 = \frac{\alpha}{1-\alpha} \|\mathbf{K}\|$$

Tomando la igualdad de la ecuación (2-38), se tiene una ley adaptativa para el parámetro de diseño robusto. Se demuestra que, siguiendo esta ley de adaptación de ρ , el sistema en lazo cerrado es u.u.b. (del inglés, *uniformly ultimately bounded*) [Cor81].

En general, el sistema presenta con este esquema un buen comportamiento, pero tiene dos problemas. El primero es que se necesita un estudio previo en simulación para determinar las cotas de las incertidumbres, para lo cual hay que disponer de un modelo lo más exacto posible. El segundo problema es que hay que realizar una serie de aproximaciones sobre términos no conocidos a priori, como es el error en los términos no lineales de (2-1).

2.4.2 Otros controladores robustos

En el controlador de Liu and Goldenberg [Liu93],[Liu94] los autores proponen una parametrización de las matrices de inercia y de Coriolis y centrífuga, que conduce a una acción robusta bastante similar a (2-29), probando que el error de seguimiento es de nuevo u.u.b. El parámetro de cota de incertidumbre para cada ley de control se elige midiendo la máxima diferencia entre los términos no lineales de las ecuaciones dinámicas del robot en el caso de no transportar ninguna masa y en el caso de transportar una carga máxima, pero es una expresión de acotación y se asume que las masas de carga y los parámetros dinámicos son bien conocidos.

En el trabajo de Jaritz and Spong [Jar96] se propone una comparación entre estos controladores robustos y otros, pero no se dice nada acerca de la elección del parámetro ρ en los diferentes casos.

En Spong et al. [Spo87c] se propone una ley robusta similar a (2-29), y ρ queda fijada por la medida del error de seguimiento después de la elección aleatoria de otras constantes.

En el resto de bibliografía revisada, la elección de este parámetro de cota de incertidumbre no queda clara o no se establece fácilmente, o simplemente no se menciona. Por esta razón, en esta tesis se propone un nuevo método para definir este parámetro, basado en la adaptación del mismo en función del error de seguimiento obtenido a lo largo de la trayectoria. De esta forma no hace falta un estudio previo de las cotas de incertidumbre del sistema o similar. Esta idea queda reflejada en el controlador ARC que se describe en el siguiente capítulo.

Controlador robusto adaptativo (ARC)

En este trabajo se introduce una nueva técnica que mejora algoritmos de control robustos ampliamente estudiados: el controlador robusto adaptativo, o ARC (Adaptive Robust Controller). Además se aportan resultados reales que muestran su eficacia para el rechazo a perturbaciones. Asimismo, se aporta un criterio de estabilidad que establece condiciones para la elección de las matrices de la función de costo, empleada en la ley de adaptación del parámetro robusto. Este controlador pretende dar una solución mejorada para el caso de rechazo a perturbaciones que no están o no pueden ser acotadas previamente. Además, el controlador muestra un comportamiento eficiente en el caso de incertidumbres en el modelo. La eficiencia de las estrategias se comprueba en el manipulador empleado en esta tesis: el robot PUMA 560 de Unimation. El objetivo del capítulo es la comparación de los resultados obtenidos con el controlador ARC frente a otras técnicas de control existentes, para comprobar los beneficios que aporta la nueva solución de control presentada. La primera parte del capítulo describe el diseño del controlador y la prueba de estabilidad del mismo. A continuación se describe el manipulador empleado, así como el modelo del mismo. A continuación se muestran los resultados para el controlador robusto de Spong, con los cuales se hace evidente la necesidad de introducir una nueva acción adaptativa para dicho controlador, lo que conlleva el planteamiento del controlador ARC.

3.1 Introducción

Como se comentó en el capítulo inicial, el propósito de esta tesis es la implementación de diversas técnicas de control para mejorar el rendimiento que ofrecen los controladores tradicionales aplicados sobre manipuladores robóticos. En particular, se pretende mejorar el rendimiento de las técnicas de control para el seguimiento de trayectorias, sin que el coste computacional asociado a los nuevos algoritmos aumente de forma significativa.

Otro objetivo es mejorar la robustez de las técnicas de control ante la presencia de incertidumbres paramétricas (incertidumbres en parámetros dinámicos, variaciones de masa de carga, etc.).

En el capítulo anterior se describió el controlador de Spong, un controlador robusto diseñado por la aplicación del método directo de Lyapunov en un manipulador bajo la acción de una ley de control PD estabilizante. La acción robusta que se añade a esta ley de control trata de compensar las diferencias entre la dinámica real del manipulador y el modelo estimado del mismo.

Esta acción robusta presenta un buen comportamiento en ciertas circunstancias, pero debe ser revisada al menos en dos casos. Primero, cuando el robot trabaja con diferentes masas de carga, y segundo, cuando el mismo controlador se usa con un manipulador que tiene distintos parámetros dinámicos. La primera razón es la más común. Habitualmente el brazo manipulador debe transportar masas diferentes entre distintos puntos en el espacio de trabajo del robot. La dinámica del sistema varía dependiendo de la región del espacio del trabajo y del valor de la masa. Debido a esto, la acción robusta debe variar adecuadamente. En esta tesis se introduce un esquema adaptativo (controlador ARC) para sintonizar automáticamente el parámetro de diseño robusto involucrado en esta acción robusta.

Los objetivos a conseguir con este controlador son:

- Mejorar el rendimiento de las técnicas clásicas de control de manipuladores en el problema del seguimiento de trayectorias, sin aumentar significativamente el costo computacional,
- Mejorar la robustez del controlador en presencia de perturbaciones paramétricas, tales como variaciones en la masa de carga o incertidumbres en los parámetros dinámicos,
- Evitar el cálculo previo de parámetros de diseño de la ley de control, mediante la autoadaptación en línea de la misma en función del error de seguimiento.

La técnica ARC resuelve estos problemas y, además, evita el ajuste inicial de los parámetros involucrados en el diseño de control mediante una adaptación *on-line* de los mismos. Este controlador se basa en el esquema de Spong [Spo92],[Sci00]. Consta de tres partes: un bucle interno de linealización por realimentación, una ley de control PD estabilizante y una acción robusta. Esta acción robusta es actualizada automáticamente durante el movimiento del robot, usando una ley adaptativa. El esquema de control se puede observar en el diagrama de la figura 3-1.

En el siguiente apartado se describe la autoadaptación del parámetro de diseño de la acción robusta. Esta ley adaptativa es añadida al esquema del controlador de Spong para conseguir los objetivos de control antes mencionados. A continuación, se describe el manipulador empleado así como

los resultados reales obtenidos con el mismo aplicando la técnica de control propuesta. Asimismo, se realiza una comparación de dichos resultados con los obtenidos con técnicas clásicas de control para probar el comportamiento satisfactorio de la nueva estrategia.

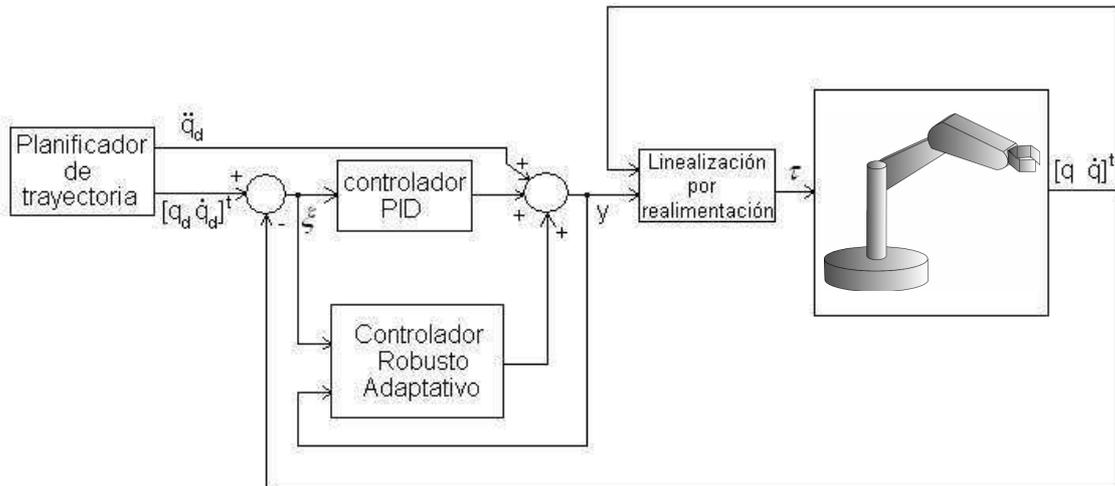


Figura 3-1: Esquema del controlador robusto adaptativo.

3.2 Ley autoadaptativa para el parámetro de cota de incertidumbre

Para dar una respuesta al problema de elección adecuada del parámetro de cota de incertidumbre, se propone en esta tesis una ley adaptativa que actualiza eficazmente el valor de este parámetro.

Para hacer esto se hace uso de la regla del MIT (o regla del gradiente descendente). Una vez dada una función del coste $J_k(\rho)$, la adaptación del parámetro se hace mediante la siguiente ecuación:

$$\rho_k = \rho_{k-1} - \gamma \frac{\partial J_k}{\partial \rho_{k-1}} \quad (3-1)$$

donde γ es la velocidad de aprendizaje de la ley de adaptación. Este término debe tener en cuenta el período de muestreo h para así tener una correcta actualización del parámetro ρ .

En este caso, la función de costo es cuadrática y está formada por dos términos. El primero de ellos pesa el error en el estado del robot y el segundo término pesa la entrada resultante al sistema linealizado:

$$J_k(\rho_{k-1}) = \frac{1}{2} \xi_k^T \mathbf{Q}_{ad} \xi_k + \frac{1}{2} \mathbf{y}_{k-1}^T \mathbf{R}_{ad} \mathbf{y}_{k-1} \quad (3-2)$$

donde la matriz $2n \times 2n$ \mathbf{Q}_{ad} pesa el error en el estado y la matriz $n \times n$ \mathbf{R}_{ad} pesa la influencia de las entradas en el sistema linealizado. Esta elección conduce a la siguiente ley de adaptación:

$$\rho_k = \rho_{k-1} - \gamma \left[\xi_k^T \mathbf{Q}_{ad} \frac{\partial \xi_k}{\partial \rho_{k-1}} + \mathbf{y}_{k-1}^T \mathbf{R}_{ad} \frac{\partial \mathbf{y}_{k-1}}{\partial \rho_{k-1}} \right] \quad (3-3)$$

Para calcular las derivadas de (3-3), se aplica una aproximación de primer orden a los términos con derivadas. El error ξ_k puede ser aproximado por:

$$\xi_k = \begin{bmatrix} \tilde{\theta}_k \\ \dot{\tilde{\theta}}_k \end{bmatrix} = \begin{bmatrix} \theta_{d,k} \\ \dot{\theta}_{d,k} \end{bmatrix} - \begin{bmatrix} \theta_k \\ \frac{\theta_k - \theta_{k-1}}{h} \end{bmatrix} \quad (3-4)$$

Su derivada con respecto a ρ_{k-1} es:

$$\frac{\partial \xi_k}{\partial \rho_{k-1}} = - \begin{bmatrix} \frac{\partial \theta_k}{\partial \rho_{k-1}} \\ \frac{1}{h} \frac{\partial \theta_k}{\partial \rho_{k-1}} \end{bmatrix} \quad (3-5)$$

Para calcular ésta derivada, se hace uso del modelo linealizado del sistema (2-3), expresado en el espacio de los estados:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}_D \mathbf{x}_{k-1} + \mathbf{B}_D \mathbf{y}_{k-1} \\ \theta_k &= \mathbf{C}_D \mathbf{x}_k \end{aligned} \quad (3-6)$$

donde \mathbf{x}_k es el estado del robot en el instante k , \mathbf{A}_D es la matriz principal del sistema, \mathbf{B}_D la matriz de entrada, y \mathbf{C}_D la matriz de salida. El subíndice D hace referencia a que se emplea la versión discreta de estas matrices.

Entonces, la derivada de (3-5) es:

$$\frac{\partial \theta_k}{\partial \rho_{k-1}} = \mathbf{C}_D \mathbf{B}_D \frac{\partial \mathbf{y}_{k-1}}{\partial \rho_{k-1}} \quad (3-7)$$

Si se asume que $\|\mathbf{D}' \mathbf{Q} \xi_k\| < \varepsilon$, $\forall \varepsilon$, lo cual es cierto excepto quizás al principio del movimiento, la ley de control (2-28) se convierte en:

$$\mathbf{y}_k = \ddot{\theta}_{k,d} + \mathbf{K} \xi_k + \rho_k \mathbf{M} \xi_k \quad (3-8)$$

donde $\mathbf{M} = \mathbf{D}^T \mathbf{Q} / \varepsilon$. La expresión (3-8) conduce al cálculo de la derivada en la parte derecha de (3-7) como sigue:

$$\frac{\partial \mathbf{y}_{k-1}}{\partial \rho_{k-1}} = \mathbf{M} \xi_{k-1} \quad (3-9)$$

Haciendo uso de (3-7) y (3-9), la expresión (3-5) puede ser escrita como:

$$\frac{\partial \xi_k}{\partial \rho_{k-1}} = - \left[\frac{\mathbf{C}_D \mathbf{B}_D \mathbf{M} \xi_{k-1}}{\frac{1}{h} \mathbf{C}_D \mathbf{B}_D \mathbf{M} \xi_{k-1}} \right] \quad (3-10)$$

Finalmente, las expresiones (3-9) y (3-10) pueden ser usadas para evaluar la ley de adaptación (3-3), con lo que la expresión final para el controlador robusto adaptativo viene dada por el siguiente conjunto de ecuaciones:

$$\begin{aligned} \mathbf{y}_k &= \ddot{\theta}_{k,d} + \mathbf{K} \xi_k + \mathbf{y}_{k,r} \\ \mathbf{K} &= [\mathbf{K}_p \quad \mathbf{K}_D] \end{aligned} \quad (3-11)$$

$$\mathbf{y}_{k,r} = \begin{cases} \frac{\rho_k}{\|\mathbf{D}' \mathbf{Q} \xi_k\|} \mathbf{D}' \mathbf{Q} \xi_k, & \text{si } \|\mathbf{D}' \mathbf{Q} \xi_k\| \geq \varepsilon \\ \frac{\rho_k}{\varepsilon} \mathbf{D}' \mathbf{Q} \xi_k, & \text{si } \|\mathbf{D}' \mathbf{Q} \xi_k\| < \varepsilon \end{cases} \quad (3-12)$$

$$\rho_k = \rho_{k-1} - \gamma \left[\xi_k^T \mathbf{Q}_{ad} \frac{\partial \xi_k}{\partial \rho_{k-1}} + \mathbf{y}_{k-1}^T \mathbf{R}_{ad} \frac{\partial \mathbf{y}_{k-1}}{\partial \rho_{k-1}} \right] \quad (3-13)$$

$$\frac{\partial \xi_k}{\partial \rho_{k-1}} = - \left[\frac{\mathbf{C}_D \mathbf{B}_D \mathbf{M} \xi_{k-1}}{\frac{1}{h} \mathbf{C}_D \mathbf{B}_D \mathbf{M} \xi_{k-1}} \right] \quad (3-14)$$

$$\frac{\partial \mathbf{y}_{k-1}}{\partial \rho_{k-1}} = \mathbf{M} \xi_{k-1} \quad (3-15)$$

$$\mathbf{M} = \mathbf{D}^T \mathbf{Q} / \varepsilon \quad (3-16)$$

A continuación se define el algoritmo completo a aplicar para la implementación de este controlador.

3.2.1 Algoritmo del controlador robusto auto-adaptativo propuesto

Se tienen inicialmente la trayectoria de consigna para cada una de las articulaciones, $\theta_{k,d}$, así como sus dos primeras derivadas, $\dot{\theta}_{k,d}$ y $\ddot{\theta}_{k,d}$. Dados los valores de las matrices:

- \mathbf{K}_p y \mathbf{K}_D , que conforman la matriz $\mathbf{K} = [\mathbf{K}_p \quad \mathbf{K}_D]$,
- \mathbf{R}_{ad} y \mathbf{Q}_{ad} , que definen la función de costo, sobre las que a continuación se establecerán condiciones de elección para que se asegure estabilidad asintótica del sistema en lazo cerrado,
- \mathbf{A}_D , \mathbf{B}_D y \mathbf{C}_D de la ecuación de estado (3-6), una vez elegido el período de muestro h ,
- \mathbf{D} , dado por la definición (2-19),

y escogiendo una matriz definida positiva \mathbf{Q} cualquiera, y un valor adecuado para ε , se aplica el siguiente algoritmo:

Paso 1) Escoger $\xi_o = \mathbf{0}$, $\mathbf{y}_o = \mathbf{0}$ y $\rho_0 = 0$. Hacer $k=1$.

Paso 2) Medir $\xi_k = [\tilde{\theta}_k \quad \dot{\tilde{\theta}}_k]^T$, con $\tilde{\theta}_k = \theta_{k,d} - \theta_k$ y $\dot{\tilde{\theta}}_k = \dot{\theta}_{k,d} - \dot{\theta}_k$, y rescatar su valor en $k-1$, ξ_{k-1} , así como la entrada al sistema linealizado en $k-1$, \mathbf{y}_{k-1} .

Paso 3) Calcular las derivadas (3-14) y (3-15), necesarias para la actualización del parámetro de cota de incertidumbre, teniendo en cuenta (3-16).

Paso 4) Calcular la actualización de ρ_k , mediante (3-13).

Paso 5) Calcular la acción robusta, $\mathbf{y}_{k,r}$, mediante (3-12).

Paso 6) Calcular la entrada de control al sistema linealizado, \mathbf{y}_k , mediante (3-11).

Paso 7) Hacer $k=k+1$, esperar hasta el siguiente período de muestreo e ir al Paso 2.

3.3 Consideraciones sobre estabilidad

Antes de aplicar este controlador, se formula un criterio de sintonización del controlador bajo el cual se asegure su estabilidad. Para ello se hace uso del teorema de estabilidad de Lyapunov, estableciendo bajo qué condiciones de elección de la función de costo de la ley de adaptación el sistema permanece estable. Para llevar a cabo este análisis, se parte de la formulación continua del

sistema, para luego hacer una aproximación al estado discreto y aplicar el teorema de estabilidad.

3.3.1 Dinámica del error

El modelo continuo del sistema viene dado por:

$$\mathbf{B}(\boldsymbol{\theta}(t))\ddot{\boldsymbol{\theta}}(t) + \mathbf{h}(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t)) + \mathbf{c}(\boldsymbol{\theta}(t)) = \boldsymbol{\tau}(t) \quad (3-17)$$

Este sistema tiene aplicado una entrada de linealización que viene dada por un conocimiento, más o menos aproximado, de las ecuaciones dinámicas del mismo. Dicha entrada de linealización viene dada por:

$$\boldsymbol{\tau}(t) = \hat{\mathbf{B}}(\boldsymbol{\theta}(t))\mathbf{y}(t) + \hat{\mathbf{h}}(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t)) + \hat{\mathbf{c}}(\boldsymbol{\theta}(t)) \quad (3-18)$$

donde las variables con el símbolo $\hat{}$ designan la estimación a la correspondiente variable real. Si se define:

$$\mathbf{n}(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t)) = \mathbf{h}(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t)) + \mathbf{c}(\boldsymbol{\theta}(t)) \quad (3-19)$$

y de igual forma para las variables estimadas, sustituyendo la entrada (3-18) en la ecuación (3-17), se tiene, obviando por comodidad en la formulación la dependencia temporal y de las coordenadas generalizadas así como de sus derivadas, que:

$$\ddot{\boldsymbol{\theta}} = \mathbf{B}^{-1}\hat{\mathbf{B}}\mathbf{y} + \mathbf{B}^{-1}(\hat{\mathbf{n}} - \mathbf{n}) \quad (3-20)$$

De esta relación se llega, sumando y restando la entrada al sistema lineal \mathbf{y} al segundo término de (3-20), a la siguiente ecuación de entrada-salida del sistema linealizado:

$$\ddot{\boldsymbol{\theta}} = \mathbf{y} - \boldsymbol{\eta} \quad (3-21)$$

donde:

$$\boldsymbol{\eta} = (\mathbf{I} - \mathbf{B}^{-1}\hat{\mathbf{B}})\mathbf{y} - \mathbf{B}^{-1}(\hat{\mathbf{n}} - \mathbf{n}) \quad (3-22)$$

Este término mide las discrepancias entre el sistema real y el modelo estimado del mismo. Nótese que, con estimaciones exactas del modelo real, se tiene una contribución nula a la ecuación de entrada-salida del sistema.

Según el esquema propuesto, la entrada de control al sistema (3-21) viene dada por:

$$\mathbf{y} = \ddot{\boldsymbol{\theta}}_d + \mathbf{K}_D \dot{\boldsymbol{\theta}} + \mathbf{K}_P \boldsymbol{\theta} + \mathbf{y}_r \quad (3-23)$$

con lo que se puede expresar el sistema de la siguiente manera:

$$\ddot{\tilde{\theta}} + \mathbf{K}_D \dot{\tilde{\theta}} + \mathbf{K}_p \tilde{\theta} + \mathbf{y}_r = \boldsymbol{\eta} \quad (3-24)$$

En términos del error ξ , la ecuación (3-24) puede ser escrita de la siguiente forma:

$$\dot{\xi} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K}_p & -\mathbf{K}_D \end{bmatrix} \xi + \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} (\boldsymbol{\eta} - \mathbf{y}_r) = \tilde{\mathbf{H}}\xi + \mathbf{D}(\boldsymbol{\eta} - \mathbf{y}_r) \quad (3-25)$$

Tanto la ecuación (3-24) como (3-25) expresan la dinámica del error del sistema en lazo cerrado bajo la ley de control robusta propuesta.

3.3.2 Estabilidad del sistema en lazo cerrado

Aplicamos el teorema de estabilidad de Lyapunov para el caso de sistemas discretos, ya que el tratamiento seguido a lo largo del presente capítulo, así como con el sistema real, es de este tipo.

Lo primero que se requiere es una aproximación discreta de la evolución del error del sistema. A continuación se aplicará el teorema de estabilidad mencionado y propondremos una condición para la elección de la ley de control adaptativa que asegure la estabilidad del sistema en lazo cerrado.

3.3.2.1 Discretización de la ecuación dinámica del error

El sistema (3-25) puede ser discretizado, suponiendo que el periodo de muestreo es lo suficientemente pequeño, de la siguiente forma:

$$\xi_{k+1} = \xi_k + h(\tilde{\mathbf{H}}\xi_k + \mathbf{D}(\boldsymbol{\eta}_k - \mathbf{y}_{k,r})) \quad (3-26)$$

el cual podemos resumir en:

$$\xi_{k+1} = \mathbf{J}\xi_k + h\mathbf{D}(\boldsymbol{\eta}_k - \mathbf{y}_{k,r}) \quad (3-27)$$

donde

$$\mathbf{J} = \mathbf{I} + h\tilde{\mathbf{H}} \quad (3-28)$$

siendo \mathbf{I} la matriz identidad de tamaño $2n \times 2n$.

3.3.2.2 Aplicación de la teoría de estabilidad de Lyapunov

Para analizar la estabilidad del sistema (3-27), proponemos la siguiente función de Lyapunov discreta:

$$V(\xi_k) = \xi_k^T \mathbf{Q} \xi_k > 0 \quad (3-29)$$

con $\mathbf{Q}^T = \mathbf{Q}$ y definida positiva. Esta función se anula para $\xi_k = \mathbf{0}$. Luego probando que $\Delta V(\xi_k) < 0$ se tiene que el sistema en lazo cerrado es estable. Definimos:

$$\Delta V(\xi_k) \stackrel{\Delta}{=} V(\xi_{k+1}) - V(\xi_k) \quad (3-30)$$

Establecemos las condiciones de estabilidad del sistema en lazo cerrado mediante el siguiente teorema:

Teorema 1

El sistema (3-27), con las definiciones de matrices dadas en (3-25) y (3-28), bajo la ley de control dada por las ecuaciones (3-11) a (3-16), es estable si se da la siguiente relación entre las matrices de la función de costo (3-2):

$$\frac{\|\mathbf{R}_{ad}\|}{\|\mathbf{Q}_{ad}\|} < \frac{\varepsilon}{y_M \|\mathbf{D}^T \mathbf{Q}\|} \|\mathbf{T}\| \quad (3-31)$$

y el cambio de la discrepancia del modelo estimado respecto al sistema real, expresado por (3-22), es prácticamente nulo entre un instante de tiempo y el siguiente; donde y_M es el máximo valor que puede tomar la entrada al sistema linealizado, \mathbf{Q} es una matriz simétrica definida positiva y \mathbf{T} es la matriz $[\mathbf{C}_D \mathbf{B}_D \quad 1/h \mathbf{C}_D \mathbf{B}_D]^T$, representando $\|\cdot\|$ la norma-2 de una matriz.

Demostración del Teorema 1

Teniendo en cuenta la función de Lyapunov escogida (3-29), se tiene que:

$$\begin{aligned} \Delta V(\xi_k) &= \xi_{k+1}^T \mathbf{Q} \xi_{k+1} - \xi_k^T \mathbf{Q} \xi_k = \\ &= \left[\xi_k^T \mathbf{J}^T + (\eta_k - \mathbf{y}_{k,r})^T h \mathbf{D}^T \right] \mathbf{Q} \left[\mathbf{J} \xi_k + h \mathbf{D} (\eta_k - \mathbf{y}_{k,r}) \right] - \xi_k^T \mathbf{Q} \xi_k = \\ &= \xi_k^T \mathbf{J}^T \mathbf{Q} \mathbf{J} \xi_k + \xi_k^T \mathbf{J}^T \mathbf{Q} h \mathbf{D} (\eta_k - \mathbf{y}_{k,r}) + (\eta_k - \mathbf{y}_{k,r})^T h \mathbf{D}^T \mathbf{Q} \mathbf{J} \xi_k + \\ &\quad + (\eta_k - \mathbf{y}_{k,r})^T h \mathbf{D}^T \mathbf{Q} h \mathbf{D} (\eta_k - \mathbf{y}_{k,r}) - \xi_k^T \mathbf{Q} \xi_k \end{aligned} \quad (3-32)$$

El segundo y tercer término de (3-32) son escalares, y cada uno es el traspuesto del otro, por lo que son iguales. Además se tiene que:

$$\mathbf{J}^T \mathbf{Q} \mathbf{J} = (\mathbf{I} + h\tilde{\mathbf{H}})^T \mathbf{Q} (\mathbf{I} + h\tilde{\mathbf{H}}) = \mathbf{Q} + h\mathbf{Q}\tilde{\mathbf{H}} + h\tilde{\mathbf{H}}^T \mathbf{Q} + h^2 \tilde{\mathbf{H}}^T \mathbf{Q} \tilde{\mathbf{H}} \quad (3-33)$$

Dado que \mathbf{Q} es simétrica y definida positiva, y $\tilde{\mathbf{H}}$ tiene todos sus autovalores negativos (ver ecuación (3-25)), se puede encontrar una matriz \mathbf{P} definida positiva tal que:

$$\mathbf{Q}\tilde{\mathbf{H}} + \tilde{\mathbf{H}}^T \mathbf{Q} = -\mathbf{P} \quad (3-34)$$

Por lo que:

$$\mathbf{J}^T \mathbf{Q} \mathbf{J} = \mathbf{Q} - h\mathbf{P} + h^2 \tilde{\mathbf{H}}^T \mathbf{Q} \tilde{\mathbf{H}} \quad (3-35)$$

Además:

$$\mathbf{J}^T \mathbf{Q} h \mathbf{D} = h(\mathbf{I} + h\tilde{\mathbf{H}})^T \mathbf{Q} \mathbf{D} = h\mathbf{Q} \mathbf{D} + h^2 \tilde{\mathbf{H}}^T \mathbf{Q} \mathbf{D} \quad (3-36)$$

$$\mathbf{D}^T \mathbf{Q} \mathbf{D} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{Q} \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} = \mathbf{Q} \quad (3-37)$$

Sustituyendo (3-35), (3-36) y (3-37) en (3-32) se tiene:

$$\begin{aligned} \Delta V(\xi_k) = & \xi_k^T \mathbf{Q} \xi_k - h \xi_k^T \mathbf{P} \xi_k + h^2 \xi_k^T \tilde{\mathbf{H}}^T \mathbf{Q} \tilde{\mathbf{H}} \xi_k + \\ & + 2h \xi_k^T \mathbf{Q} \mathbf{D} (\eta_k - \mathbf{y}_{k,r}) + 2h^2 \xi_k^T \tilde{\mathbf{H}}^T \mathbf{Q} \mathbf{D} (\eta_k - \mathbf{y}_{k,r}) + \\ & + h^2 (\eta_k - \mathbf{y}_{k,r})^T \mathbf{Q} (\eta_k - \mathbf{y}_{k,r}) - \xi_k^T \mathbf{Q} \xi_k \end{aligned} \quad (3-38)$$

que se puede poner, agrupado en un único sumando los términos que dependen de h^2 , como:

$$\Delta V(\xi_k) = -h \xi_k^T \mathbf{P} \xi_k + 2h \xi_k^T \mathbf{Q} \mathbf{D} (\eta_k - \mathbf{y}_{k,r}) + o(h^2) \quad (3-39)$$

Si el período de muestro es lo suficientemente pequeño, se puede despreciar dicho sumando frente a los sumandos anteriores, que son términos únicamente dependientes de h .

Entonces, para asegurar la estabilidad del sistema, debe ser que:

$$\xi_k^T \mathbf{Q} \mathbf{D} (\eta_k - \mathbf{y}_{k,r}) < 0 \quad (3-40)$$

Se define:

$$\mathbf{z}_k = \mathbf{D}^T \mathbf{Q} \xi_k \quad (3-41)$$

Entonces, dada la ley de control robusta (2-29), se tiene que:

$$\mathbf{z}_k^T (\boldsymbol{\eta}_k - \mathbf{y}_{k,r}) = \mathbf{z}_k^T \boldsymbol{\eta}_k - \rho_k \mathbf{z}_k^T \frac{\mathbf{z}_k}{\pi} \quad (3-42)$$

donde

$$\pi = \begin{cases} \|\mathbf{z}_k\|, & \text{si } \|\mathbf{z}_k\| \geq \varepsilon \\ \varepsilon, & \text{si } \|\mathbf{z}_k\| < \varepsilon \end{cases} \quad (3-43)$$

El segundo término de la diferencia en (3-42) es siempre positivo, por lo que se puede establecer la desigualdad siguiente:

$$\mathbf{z}_k^T (\boldsymbol{\eta}_k - \mathbf{y}_{k,r}) \leq \|\mathbf{z}_k^T\| \|\boldsymbol{\eta}_k\| - \rho_k \|\mathbf{z}_k^T\| \frac{\|\mathbf{z}_k\|}{\pi} \quad (3-44)$$

Caso a) $\|\mathbf{z}_k\| \geq \varepsilon$,

$$\mathbf{z}_k^T (\boldsymbol{\eta}_k - \mathbf{y}_{k,r}) \leq \|\mathbf{z}_k^T\| (\|\boldsymbol{\eta}_k\| - \rho_k) < 0 \quad (3-45)$$

que lleva a la condición:

$$\rho_k > \|\boldsymbol{\eta}_k\| \quad (3-46)$$

Caso b) $\|\mathbf{z}_k\| < \varepsilon$,

$$\mathbf{z}_k^T (\boldsymbol{\eta}_k - \mathbf{y}_{k,r}) \leq \|\mathbf{z}_k^T\| \|\boldsymbol{\eta}_k\| - \rho_k \|\mathbf{z}_k^T\| \frac{\|\mathbf{z}_k\|}{\pi} \quad (3-47)$$

que lleva a la condición:

$$\rho_k > \frac{\varepsilon}{\|\mathbf{z}_k\|} \|\boldsymbol{\eta}_k\| \quad (3-48)$$

Esta última condición es más restrictiva que la anterior puesto que $\|\mathbf{z}_k\| < \varepsilon$. Luego demostrando esta desigualdad, tendremos demostrada la estabilidad del sistema original. Esto se puede hacer por el método de inducción. Teniendo en cuenta que:

$$\boldsymbol{\eta}_k = (\mathbf{I} - \mathbf{B}_k^{-1} \hat{\mathbf{B}}_k) \mathbf{y}_k - \mathbf{B}_k^{-1} (\hat{\mathbf{n}}_k - \mathbf{n}_k) \quad (3-49)$$

$$\rho_k = \rho_{k-1} - \gamma \left(\mathbf{y}_{k-1}^T \mathbf{R}_{ad} \mathbf{M} \boldsymbol{\xi}_k - \boldsymbol{\xi}_k^T \mathbf{Q}_{ad} \begin{bmatrix} \mathbf{C}_D \mathbf{B}_D \mathbf{M} \boldsymbol{\xi}_{k-1} \\ \frac{1}{h} \mathbf{C}_D \mathbf{B}_D \mathbf{M} \boldsymbol{\xi}_{k-1} \end{bmatrix} \right) \quad (3-50)$$

En el instante inicial, $k=0$, la inecuación se cumple escogiendo ρ_0 tal que:

$$\rho_0 > \frac{\varepsilon}{\|\mathbf{z}_0\|} \|\boldsymbol{\eta}_0\| \quad (3-51)$$

Se supone cierto para el instante k :

$$\rho_k > \frac{\varepsilon}{\|\mathbf{z}_k\|} \|\boldsymbol{\eta}_k\| \quad (3-52)$$

y se demuestra para el instante $k+1$:

$$\rho_{k+1} > \frac{\varepsilon}{\|\mathbf{z}_{k+1}\|} \|\boldsymbol{\eta}_{k+1}\| - \gamma \left(\mathbf{y}_k^T \mathbf{R}_{ad} \mathbf{M} \boldsymbol{\xi}_k - \boldsymbol{\xi}_{k+1}^T \mathbf{Q}_{ad} \left[\frac{\mathbf{C}_D \mathbf{B}_D \mathbf{M} \boldsymbol{\xi}_k}{h} \right] \right) > \frac{\varepsilon}{\|\mathbf{z}_{k+1}\|} \|\boldsymbol{\eta}_{k+1}\| \quad (3-53)$$

Si h es lo suficientemente pequeño, se puede suponer que $\|\mathbf{z}_{k+1}\| \approx \|\mathbf{z}_k\|$. Entonces la desigualdad a demostrar es:

$$\frac{\varepsilon}{\|\mathbf{z}_k\|} (\|\boldsymbol{\eta}_{k+1}\| - \|\boldsymbol{\eta}_k\|) < -\gamma (\mathbf{y}_k^T \mathbf{R}_{ad} - \boldsymbol{\xi}_{k+1}^T \mathbf{Q}_{ad} \mathbf{T}) \mathbf{M} \boldsymbol{\xi}_k \quad (3-54)$$

donde $\mathbf{T} = \left[\mathbf{C}_D \mathbf{B}_D \quad \frac{1}{h} \mathbf{C}_D \mathbf{B}_D \right]^T$. Bajo la suposición del Teorema 1 de que la desviación del modelo estimado respecto al sistema real, expresada por la variable $\boldsymbol{\eta}_k$ en (3-49), es prácticamente constante entre un instante de tiempo y el siguiente, $\|\boldsymbol{\eta}_k\| \approx \|\boldsymbol{\eta}_{k+1}\|$, la desigualdad (3-54) queda de la siguiente manera:

$$(\mathbf{y}_k^T \mathbf{R}_{ad} - \boldsymbol{\xi}_{k+1}^T \mathbf{Q}_{ad} \mathbf{T}) \mathbf{M} \boldsymbol{\xi}_k < 0 \quad (3-55)$$

Lo que conduce a:

$$\mathbf{y}_k^T \mathbf{R}_{ad} \mathbf{M} \boldsymbol{\xi}_k < \boldsymbol{\xi}_{k+1}^T \mathbf{Q}_{ad} \mathbf{T} \mathbf{M} \boldsymbol{\xi}_k \quad (3-56)$$

La desigualdad (3-56) se cumple si se escogen convenientemente las matrices de la función de costo \mathbf{R}_{ad} y \mathbf{Q}_{ad} . Para establecer una condición, se toma norma de dicha inequación. Sin embargo, hay que tener en cuenta previamente ciertos aspectos concernientes a la negatividad de alguno de los términos. El término de la derecha de (3-56) solamente es negativo cuando, entre un instante de muestreo y el siguiente, los errores que más pesa el término $\mathbf{Q}_{ad} \mathbf{T} \mathbf{M}$ cambien a la vez de signo, lo cual es muy improbable. Luego se puede suponer que dicho término siempre es positivo.

Si el término de la izquierda es negativo, la desigualdad se cumplirá siempre, para cualquier elección de \mathbf{R}_{ad} y \mathbf{Q}_{ad} . Si dicho término es positivo, se puede poner:

$$\|\mathbf{y}_k^T\| \|\mathbf{R}_{ad}\| < \|\xi_{k+1}^T\| \|\mathbf{Q}_{ad}\| \|\mathbf{T}\| \quad (3-57)$$

Dado que se está analizando el caso más restrictivo b), se sabe que $\|\mathbf{D}^T \mathbf{Q} \xi_{k+1}\| < \varepsilon$. Además se puede establecer una cota para las entradas linealizadas \mathbf{y}_k , dada por el valor máximo que se puede aplicar, y_M . Entonces se tiene:

$$\|\mathbf{y}_k^T\| \|\mathbf{R}_{ad}\| < y_M \|\mathbf{R}_{ad}\| < \|\xi_{k+1}^T\| \|\mathbf{Q}_{ad}\| \|\mathbf{T}\| < \frac{\varepsilon}{\|\mathbf{D}^T \mathbf{Q}\|} \|\mathbf{Q}_{ad}\| \|\mathbf{T}\| \quad (3-58)$$

con lo que la elección de las matrices de costo que asegura la estabilidad del sistema debe cumplir:

$$\frac{\|\mathbf{R}_{ad}\|}{\|\mathbf{Q}_{ad}\|} < \frac{\varepsilon}{y_M \|\mathbf{D}^T \mathbf{Q}\|} \|\mathbf{T}\| \quad (3-59)$$

■

Debe tenerse en cuenta que esta condición es suficiente, esto es, el no cumplimiento de esta inecuación no asegura que el sistema en lazo cerrado no sea asintóticamente estable.

Con este teorema queda probada la estabilidad de la ley de control propuesta en lazo cerrado bajo ciertas condiciones, que básicamente se consiguen eligiendo un período de muestreo pequeño. En los siguientes apartados se exponen los resultados obtenidos con este controlador aplicado sobre un robot manipulador PUMA-560 y las buenas prestaciones que este controlador ofrece. Para ello se comparan los resultados obtenidos con técnicas existentes de control, como son el PID y el controlador robusto de Spong.

3.4 Descripción del robot manipulador

Las técnicas de control expuestas a lo largo de esta tesis se aplican sobre un robot manipulador PUMA-560 de Unimation[®], como el que se observa en la figura 3-2. Se trata de un manipulador bastante usado en la industria en tareas de ensamblaje y manipulación y, actualmente, en el campo de la investigación.

Cuenta con seis grados de libertad, tres en su estructura (cintura, hombro y codo) y tres en la muñeca (cabeceo, alabeo y guiñada), que le hacen alcanzar cualquier punto de su espacio de trabajo con cualquier orientación (siempre sujeto a las restricciones mecánicas del mismo). Dispone de seis servomotores de continua para el control de las articulaciones, cuyas posiciones se miden mediante encoders y potenciómetros.



Figura 3-2: Robot manipulador PUMA-560 empleado en las distintas pruebas¹.



Figura 3-3: Controlador *Mark II* del robot PUMA-560. En el modulo de control se incluye la nueva *interface* entre el manipulador y un PC, mediante una tarjeta de adquisición Q8.

¹ Este manipulador pertenece al Grupo de Investigación de Inteligencia Cibernética de la Escuela de Ingeniería de Sistemas de la Universidad de Reading, en el Reino Unido.

Originalmente, este robot viene provisto de un controlador, el Unimation Mark II, el cual se puede observar en la figura 3-3, que dispone de toda la electrónica necesaria para el control de los motores y la lectura de las posiciones de las articulaciones. Aunque este sistema es bastante robusto, la necesidad de incorporar algoritmos de control de alto nivel hace que resulte obsoleto. Dicha tarea puede ser llevada a cabo satisfactoriamente por un PC ejecutando Matlab[®] y/o Simulink[®] y algunas otras herramientas para el diseño, desarrollo y pruebas en tiempo real de esquemas avanzados de control.

Para llevar a cabo el control del robot PUMA usando un PC con la configuración anterior, se modifica el controlador original [Bec04] añadiendo una serie de tarjetas específicamente diseñadas para este propósito, del tipo TRC041. Estas tarjetas envían las señales de los potenciómetros y encoders a una tarjeta de adquisición Q8 de *Quanser*[®], conectada al bus PCI de un PC con procesador Intel Pentium 4, 2.4 GHz, bajo el sistema operativo Windows 2000.

El PC ejecuta aplicaciones de tiempo real creadas por Matlab/Simulink y envía las señales necesarias de referencia al controlador Mark II, mediante voltajes analógicos generados por la tarjeta Q8. Las lecturas de los potenciómetros sirven para calibrar los encoders, de los que se extraen las posiciones de las articulaciones.

Para llevar a cabo los algoritmos de control expuestos en esta tesis se necesita un modelo del PUMA-560. El modelo completo para este tipo particular de manipulador, con las masas, longitudes, momentos de inercia, etc. se obtiene en [Lew04]. Este trabajo se basa, a su vez, en el trabajo original de Armstrong *et al* [Arm86], en el cual se explicitan todos los parámetros dinámicos del manipulador PUMA-560: masas, longitudes, momentos de inercia, disposición de los centros de gravedad en los elementos del robot, etc. Además, se detallan todas las expresiones de las ecuaciones dinámicas, como son los términos de la matriz de inercia y de los vectores de Coriolis, centrífugo y gravitatorio.

3.5 Estrategias de control implementadas

En el capítulo anterior se exponen resultados que demuestran la conveniencia de una elección satisfactoria del parámetro de cota de incertidumbre del término robusto, para así tener un buen comportamiento del lazo cerrado.

Concretamente, tras realizar un estudio de prueba y error en simulación, se obtiene un valor adecuado de cota de incertidumbre para el que los errores de seguimiento son mínimos. Sin embargo, si se toman valores demasiado elevados, las entradas comienzan a saturarse, lo cual puede producir inestabilidad en lazo cerrado.

Con estos resultados previos, se comprueba que una definición satisfactoria de la acción robusta es fundamental para obtener un buen

comportamiento por parte del controlador. Para comparar las buenas prestaciones de la estrategia propuesta ARC, se compara con otro tipo de estrategias.

La primera es un controlador PD con compensación de gravedad (referido como PD+GC²) tal y como se define en el capítulo 1. Este controlador presenta un buen comportamiento en la mayoría de los casos en los que no hay incertidumbres en el modelo y la región de actuación del manipulador es limitada.

La segunda estrategia de control es el controlador robusto de Spong [Spo92]. Básicamente se trata del controlador anterior al que se añade un término robusto para compensar las diferencias entre el robot y el modelo del mismo, es decir, las incertidumbres del modelo. Tal y como se describe en el capítulo 2, este término surge de la aplicación del método directo de Lyapunov, con el que se establece asimismo un mecanismo de adaptación de dicho término robusto.

El controlador de Spong se tomará como una estrategia de control representativa de las estrategias robustas y servirá de comparación frente a la nueva estrategia ARC propuesta en esta tesis, la cual también se basa en la adaptación de la acción robusta mediante la variación del parámetro de cota de incertidumbre. Los resultados obtenidos con ambos controladores se compararán en diversos experimentos.

3.6 Resultados de control

En esta sección se exponen los resultados obtenidos con el controlador robusto adaptativo ARC. Con este algoritmo de control, el parámetro de cota de incertidumbre se adapta automáticamente para dar cuenta de las desviaciones de la trayectoria respecto a la de referencia por la presencia de incertidumbres paramétricas en el modelo.

Este resultado se compara con un esquema robusto clásico. En particular, por ser el más representativo, se escoge la estrategia de Spong [Spo92]. Tanto la estrategia ARC propuesta como la estrategia de Spong utilizan un controlador PD como controlador primario para realizar el seguimiento de trayectoria.

A continuación, se exponen los resultados generales obtenidos en dos experimentos distintos. En el primero, se comparan los resultados con distintas estrategias de control, mientras que en el segundo se aplica el controlador ARC a partir de la mitad de la trayectoria, para comprobar el efecto de la acción autoadaptativa en el error de seguimiento.

² PD+GC: Controlador PD con compensación de gravedad (*gravity compensation*)

Experimento 3.1

Sistema: Manipulador PUMA-560.

Controlador: ARC (comparativa con controlador PD+GC y de Spong).

* Acción PD: $\mathbf{K}_p = \text{diag}([24 \ 48 \ 48 \ 8 \ 8 \ 8])$,

$\mathbf{K}_D = \text{diag}([4 \ 6 \ 6 \ 6 \ 6 \ 6])$.

* Acción robusta: $\varepsilon=1$,

$\mathbf{Q} = \text{diag}([0 \ 0 \ 0 \ 0 \ 0 \ 0,1 \ 1 \ 1 \ 1 \ 1])$.

* Adaptación de parámetro de acción robusta:

$\mathbf{R}_{ad} = h \cdot \text{diag}([25 \ 25 \ 25 \ 25 \ 25 \ 25])$

$\mathbf{Q}_{ad} = 100 \cdot \text{diag}([1 \ 3,75 \ 2,5 \ 2,5 \ 2,5 \ 2,5,2,5 \ 6 \ 5 \ 6 \ 6 \ 6])$

$\gamma=1$, $h = 0.001$.

* Modelo del manipulador:

Modelo de manipulador de [Lew04], con incertidumbres paramétricas en masas y longitudes de las articulaciones.

Trayectoria de referencia:

- Trayectoria en el espacio de las articulaciones generada a través de la unión de cuatro puntos en el espacio de tareas del PUMA, unidos mediante polinomios de séptimo orden¹.

- Dicha trayectoria es recorrida dos veces a lo largo del movimiento. Las articulaciones a las que se dota de movimiento son las de la estructura del brazo y una de la muñeca (la quinta).

La trayectoria de referencia para las tres primeras articulaciones elegida para este experimento se representa en la figura 3-4. Se trata de una trayectoria típica de manipulación, donde el brazo recorre trayectorias suaves uniendo varios puntos definidos previamente en el espacio de tareas del robot. En la figura 3-5 se representa la trayectoria resultante en el espacio de las tareas. Se indica el punto inicial y de destino, así como el sentido en que se recorre la trayectoria.

Para el diseño de la parte adaptativa del controlador ARC es necesaria una elección adecuada de las matrices de la función de costo \mathbf{Q}_{ad} y \mathbf{R}_{ad} . Para ello se ha seguido el criterio que asegura la estabilidad de esta técnica, según el teorema de estabilidad que se propone en el capítulo 2.

Se presentan resultados obtenidos para el caso de presencia de incertidumbre en el modelo. El modelo estimado para este sistema, tiene varias diferencias con el modelo real del robot PUMA. Esto ha sido probado introduciendo las mismas entradas generadas por un controlador PID al robot real y al modelo estimado, y viendo las diferencias en la salida de ambos sistemas, especialmente en las tres primeras articulaciones. Esto nos lleva a

considerar la existencia de incertidumbres en los parámetros considerados en el modelo, de forma que no se consigue la cancelación de los términos no lineales únicamente mediante el esquema de linealización por realimentación.

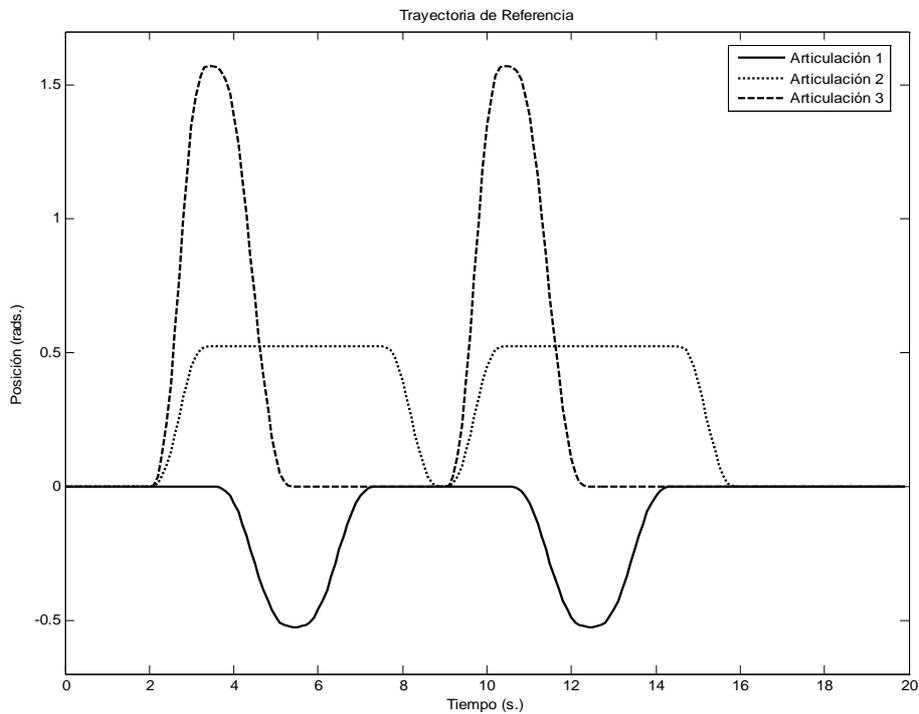


Figura 3-4: Trayectoria de referencia para el experimento 3.1 en el caso de las tres primeras articulaciones del manipulador.

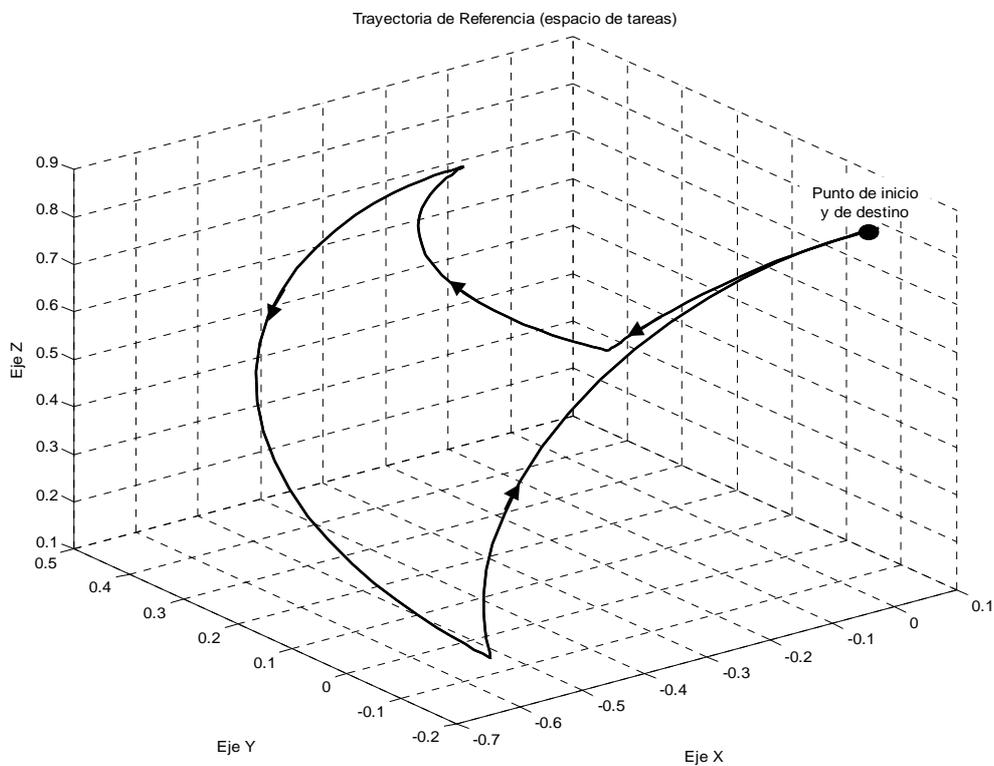


Figura 3-5: Trayectoria de referencia en el espacio de tareas correspondiente a las trayectorias articulares de la figura 3-4.

En las figuras 3-5, 3-6 y 3-7 se representa la evolución del robot, para la trayectoria de referencia especificada, bajo la acción del controlador robusto adaptativo ARC. Como se puede observar, presenta errores considerables respecto a la trayectoria de referencia. Esto se debe a que los parámetros de la acción PD han sido relajados respecto a sus valores óptimos para mostrar la eficiencia de la estrategia mostrada, respecto a los otros controladores implementados. Sin embargo, estos errores son pequeños comparados con los que se producen para otro tipo de estrategias funcionando en las mismas circunstancias, como se verá a continuación.

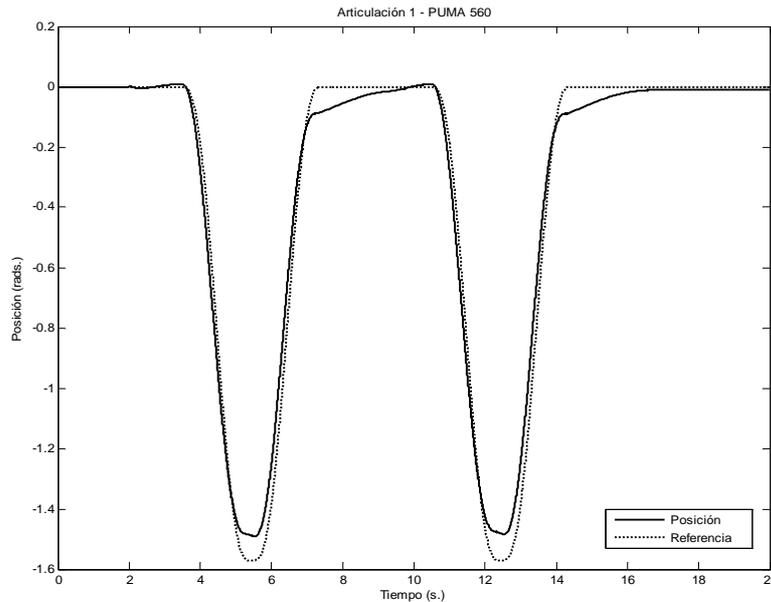


Figura 3-6: Trayectoria de la articulación 1 respecto a trayectoria de referencia bajo la acción de controlador ARC.

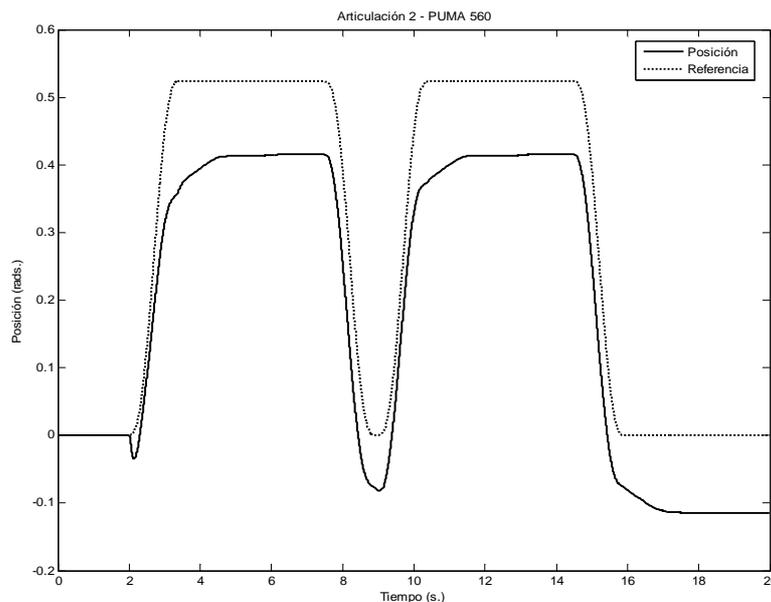


Figura 3-7: Trayectoria de la articulación 2 respecto a trayectoria de referencia bajo la acción del controlador ARC.

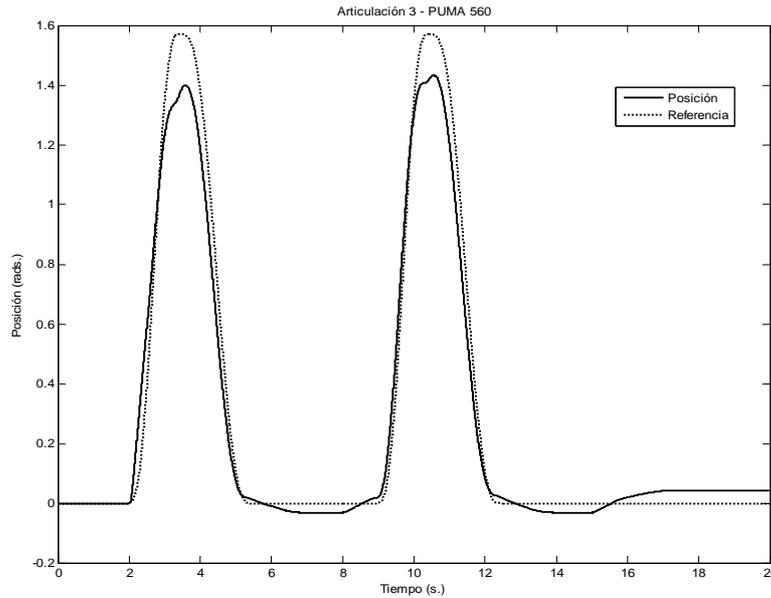


Figura 3-8: Trayectoria de la articulación 3 respecto a trayectoria de referencia bajo la acción del controlador ARC.

La estrategia robusta con la que se establece comparación es el controlador robusto de Spong. Para realizar los experimentos con este controlador, se ha tenido que realizar un estudio en simulación previo, determinando así el valor de las constantes de la ley adaptativa.

Teniendo en cuenta los valores de los ángulos y velocidades de la trayectoria de referencia, se estiman, usando el modelo del manipulador, las cotas superior e inferior de la norma de la matriz de inercia de la ecuación dinámica del manipulador, $\mathbf{D}(\boldsymbol{\theta}(t))$, así como la cota superior en la norma del error del término no lineal, $\|\hat{\mathbf{n}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\|$, donde $\mathbf{n}(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t)) = \mathbf{h}(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t)) + \mathbf{c}(\boldsymbol{\theta}(t))$.

Para este segundo término, se calcula inicialmente la cota superior de $\|\hat{\mathbf{n}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\|$ a lo largo de la trayectoria de referencia. Se supone entonces que el error que se produce en el mismo no sobrepasa el 10% del valor calculado, lo cual es comprobado a posteriori, una vez se han obtenido los resultados reales del robot. Para esta trayectoria, el valor de las constantes de la expresión de la ley de adaptación:

$$\rho_k = \lambda_0 + \lambda_1 \|\xi_k\| \quad (3-60)$$

son $\lambda_0 = 2.6186$ y $\lambda_1 = 8.2092$.

Las figuras 3-8, 3-9, 3-10 y 3-11 muestran comparativamente el comportamiento de cada uno de los controladores para las articulaciones 1, 2, 3 y 5, respectivamente, del PUMA-560. Se muestra en cada una de ellas el error de seguimiento $\tilde{\boldsymbol{\theta}}$ para las estrategias de control robusto de Spong y ARC, así como también para un controlador PD con compensación de gravedad (PD+GC).

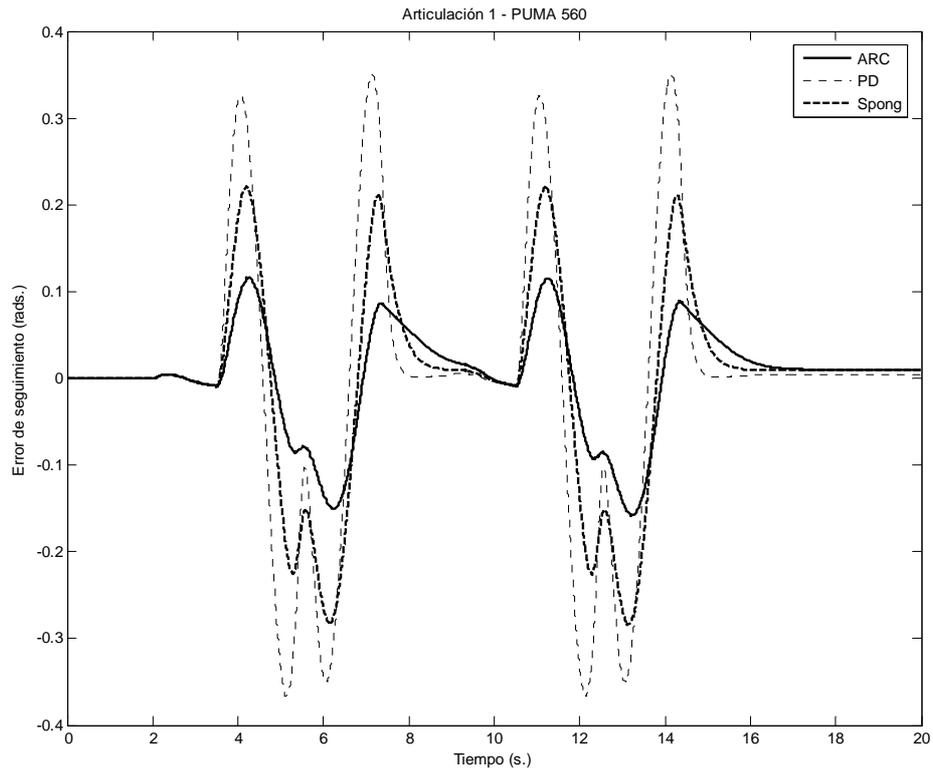


Figura 3-9: Error de seguimiento de la primera articulación bajo la acción de un controlador PD+GC, el controlador de Spong y el controlador robusto adaptativo (ARC) propuesto.

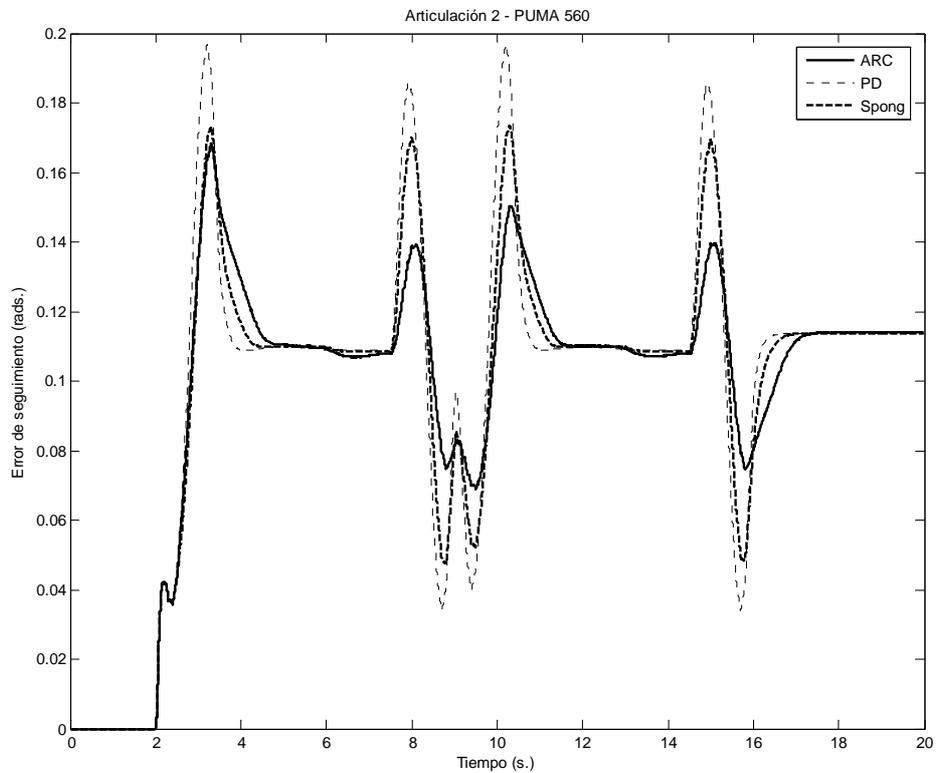


Figura 3-10: Error de seguimiento de la segunda articulación bajo la acción de un controlador PD+GC, el controlador de Spong y el controlador robusto adaptativo (ARC) propuesto.

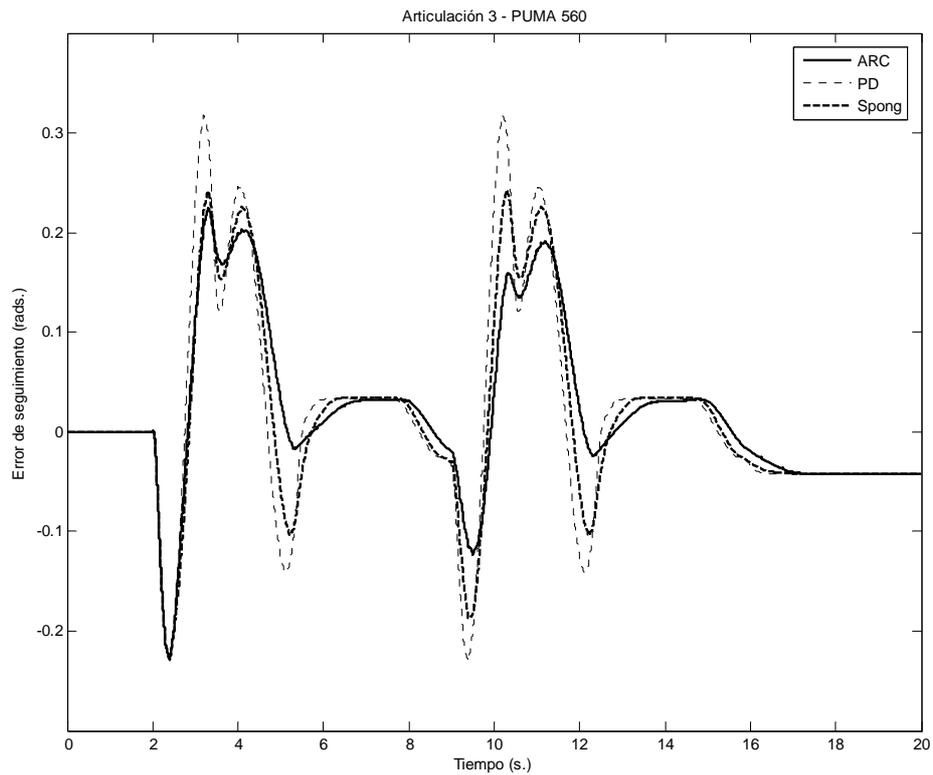


Figura 3-11: Error de seguimiento de la tercera articulación bajo la acción de un controlador PD+GC, el controlador de Spong y el controlador robusto adaptativo (ARC) propuesto.

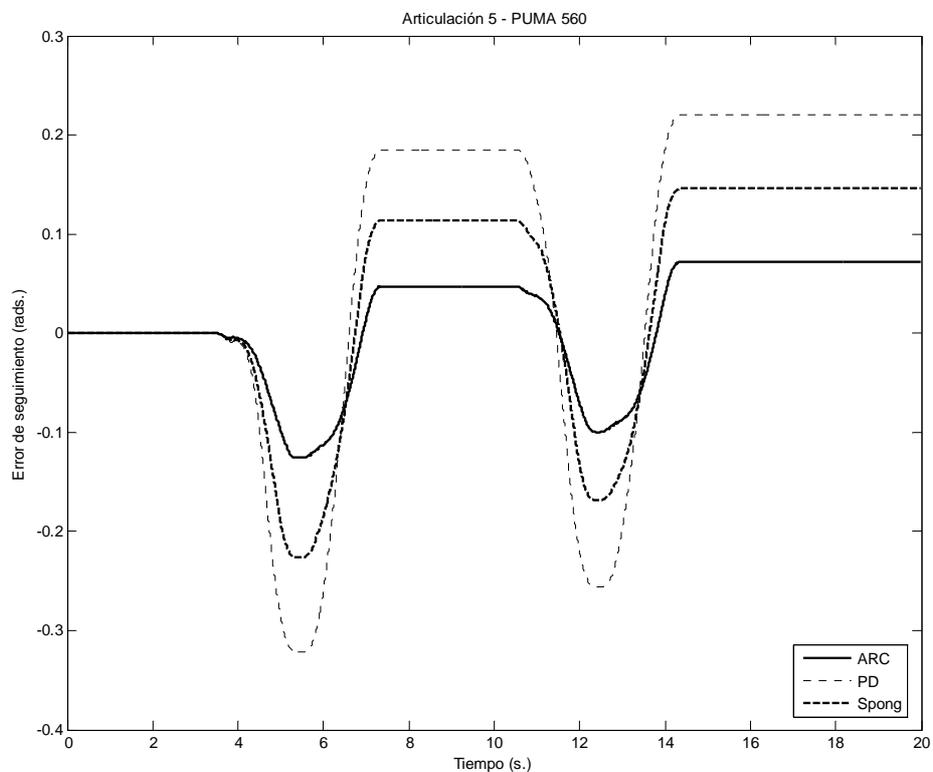


Figura 3-12: Error de seguimiento de la quinta articulación bajo la acción de un controlador PD+GC, el controlador de Spong y el controlador robusto adaptativo (ARC) propuesto.

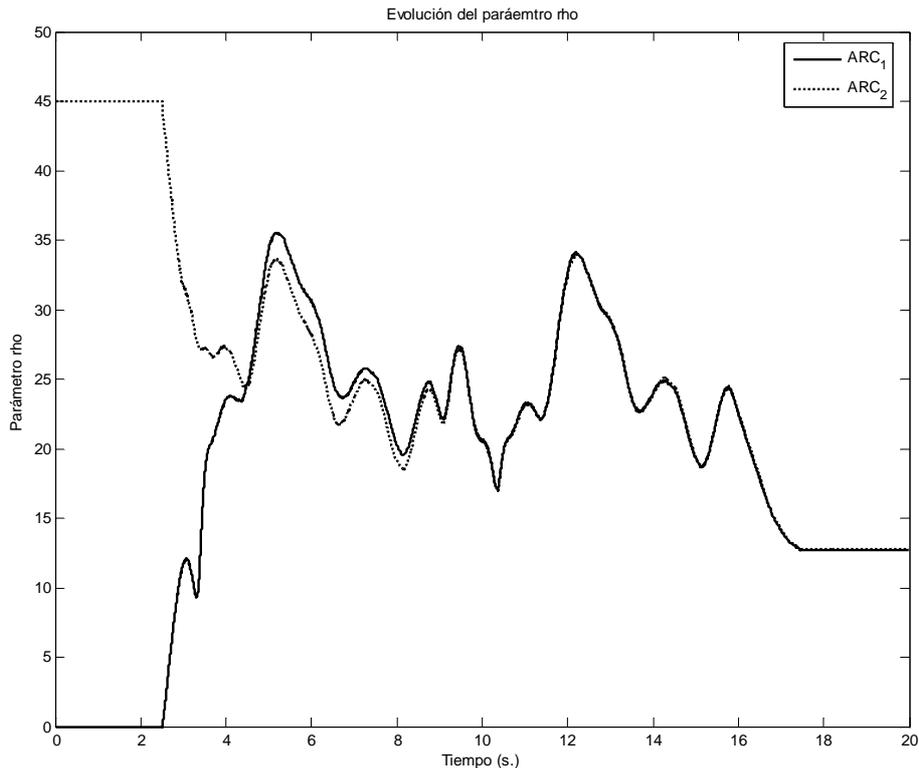


Figura 3-13: Evolución del parámetro de cota de incertidumbre (ρ) para el controlador ARC en las pruebas mostradas en las figuras Figura 3-9 a Figura 3-12, indicada como ARC_1 , y partiendo de un valor inicial distinto, indicada como ARC_2 .

Se puede comprobar que el peor comportamiento se obtiene para el controlador PD+GC, debido a las discrepancias existentes entre el modelo real y estimado del robot. La adición de un término robusto mejora la respuesta del controlador, como se puede ver en los resultados con el controlador robusto de Spong. Sin embargo, si se utiliza el controlador ARC propuesto, se obtiene un mejor resultado. Los errores son menores en todas las articulaciones, en el mejor de los casos, se reducen hasta la mitad, lo que conlleva un seguimiento de la trayectoria de consigna mucho más preciso.

La principal ventaja del controlador ARC, además del mejor comportamiento en cuanto al error de seguimiento de la trayectoria de referencia es que no es necesario el estudio previo en simulación sobre la trayectoria de referencia, sino que el valor del parámetro de diseño robusto se actualiza en cada instante de tiempo en función del error medido.

La evolución de dicho parámetro se puede observar en la figura 3-13, referenciada como ARC_1 . Inicialmente tiene un valor nulo. Transcurridos 2.5 segundos (2 segundos iniciales en los que el robot no se mueve y 0.5 segundos en los cuales no hay acción robusta), comienza la adaptación automática del parámetro, el cual se estabiliza en una banda de valores comprendidos entre $\rho = 15$ y $\rho = 35$. La evolución de las distintas articulaciones con esta adaptación resulta más suave que en los casos anteriores, además de reducirse el error en cada una de ellas.

En la misma figura se puede observar el resultado de otro experimento en el que se comienza de un valor más elevado $\rho = 48$. Esta evolución está referenciada como ARC₂. Las articulaciones tienen un comportamiento similar en este caso, salvo al principio de la trayectoria, donde los errores son algo menores debido al valor elevado del parámetro de diseño robusto. Transcurrido este periodo inicial, el comportamiento es el mismo debido a que la evolución del parámetro es idéntica.

Este resultado prueba el comportamiento satisfactorio del algoritmo de adaptación ya que, incluso partiendo de valores iniciales diferentes, para la misma trayectoria de referencia se obtiene finalmente un comportamiento bastante similar del sistema en lazo cerrado.

En la figura 3-14 se observa una comparación entre la evolución del parámetro de cota de incertidumbre ρ con el esquema ARC y con el esquema de Spong. Los valores que toma el parámetro son menores en ese caso debido a los valores inadecuados de las constantes de la ley adaptativa (3-60), lo que provoca errores de seguimiento mayores. La ley adaptativa de Spong se podría mejorar pero siempre dependiendo de las estimaciones que se hagan para las cotas en las incertidumbres, las cuales deben ser revisadas para cada trayectoria de referencia.

Un de las principales ventajas del algoritmo ARC es que no se necesita ningún tipo de estudio previo para establecer la ley de adaptación del parámetro robusto.

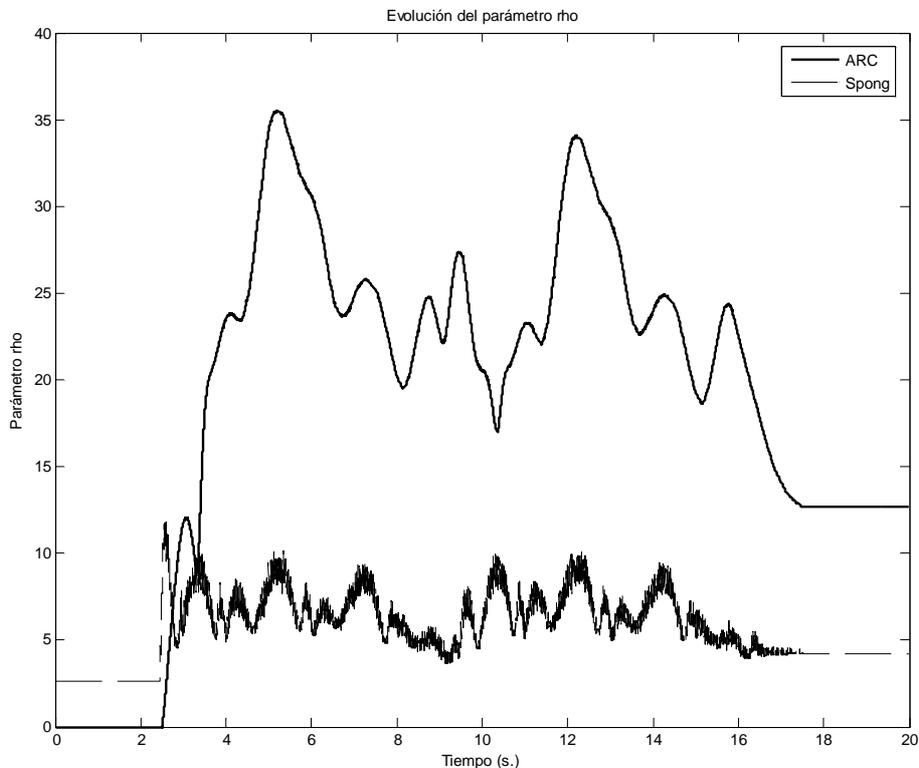


Figura 3-14: Evolución del parámetro de cota de incertidumbre para las estrategias ARC y Spong en los experimentos de las figura 3-9, Figura 3-10 y Figura 3-11.

Experimento 3.2

Sistema: Manipulador PUMA-560.

Controlador: PD+GC (primer tramo hasta 9 segundos) + ARC (segundo tramo a partir de 9 segundos)

* Acción PD: $\mathbf{K}_p = \text{diag}([24 \ 48 \ 48 \ 8 \ 8 \ 8])$,

$$\mathbf{K}_D = \text{diag}([4 \ 6 \ 6 \ 6 \ 6 \ 6]).$$

* Acción robusta: $\varepsilon=1$,

$$\mathbf{Q} = \text{diag}([0 \ 0 \ 0 \ 0 \ 0 \ 0,1 \ 1 \ 1 \ 1 \ 1]).$$

* Adaptación de parámetro de acción robusta:

$$\mathbf{R}_{ad} = h \cdot \text{diag}([25 \ 25 \ 25 \ 25 \ 25 \ 25])$$

$$\mathbf{Q}_{ad} = 100 \cdot \text{diag}([1 \ 3.75 \ 2.5 \ 2.5 \ 2.5 \ 2.5,2.5 \ 6 \ 5 \ 6 \ 6 \ 6])$$

$$\gamma=1, h = 0.001.$$

* Modelo del manipulador:

Modelo de manipulador de [Lew04], con incertidumbres paramétricas en masas y longitudes de las articulaciones.

Trayectoria de referencia:

- Trayectoria en el espacio de las articulaciones generada a través de la unión de cuatro puntos en el espacio de tareas del PUMA, unidos mediante polinomios de séptimo orden.
- Dicha trayectoria es recorrida dos veces a lo largo del movimiento. Las articulaciones a las que se dota de movimiento son las de la estructura del brazo y una de la muñeca (la quinta).

El objetivo de este segundo experimento es comparar en la misma trayectoria el comportamiento de la acción autoadaptativa sobre la acción robusta con el caso en el que no hay acción robusta. Para ello, se efectúa el recorrido de ida y vuelta en primer lugar bajo la acción de un controlador PD+GC para, a continuación, cambiar el tipo de controlador a uno ARC. En las figuras 3-14, 3-15 y 3-16 se observa la evolución del error de seguimiento bajo la acción de estos controladores para las tres primeras articulaciones. En el primer tramo, estos errores son del orden de 0.3 radianes para la primera y tercera articulación y de 0.2 radianes para la segunda articulación. En el segundo tramo, bajo la acción del controlador ARC, los errores oscilan entre los 0.15 radianes en la primera y segunda articulación y 0.2 radianes en la tercera. Tal y como se muestra, la disminución es significativa: del orden del 30% en el peor caso y del 50% en el mejor caso.

Hay que recordar que los errores de seguimiento pueden hacerse sensiblemente menores escogiendo valores mayores para las ganancias del algoritmo PD. Utilizar unos valores relajados de dichas ganancias respecto a los valores óptimos tiene como única finalidad el apreciar con claridad el buen comportamiento del controlador propuesto frente a estrategias de control robusto existentes.

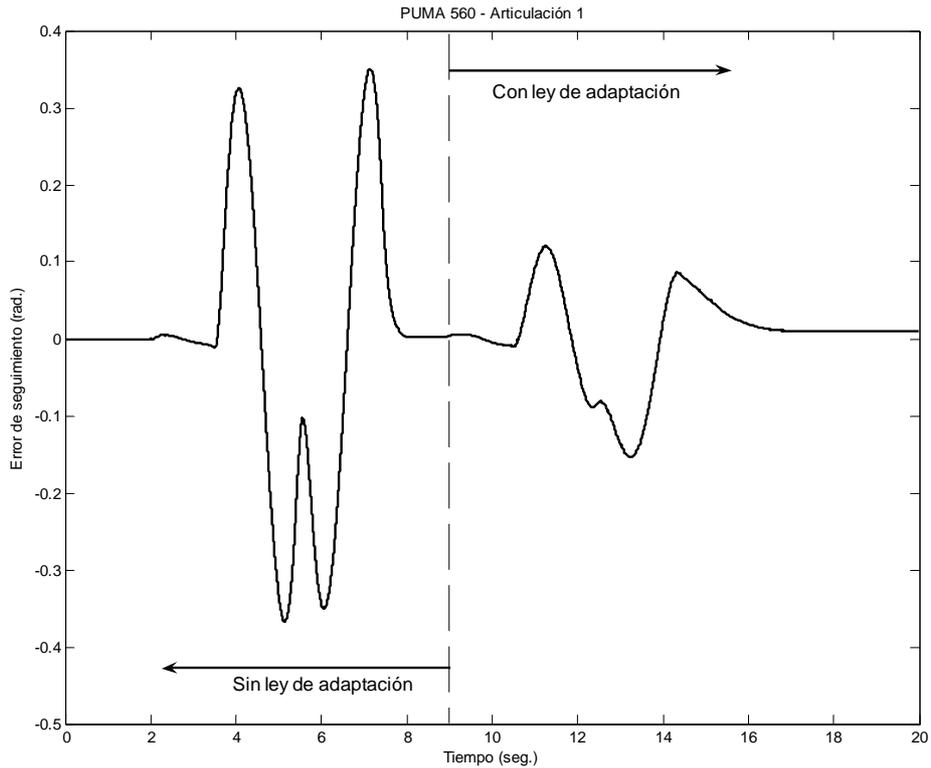


Figura 3-15: Error de seguimiento de la primera articulación bajo la acción de un controlador PD (en la primera parte de la trayectoria) y bajo la acción del controlador ARC (en la segunda parte).

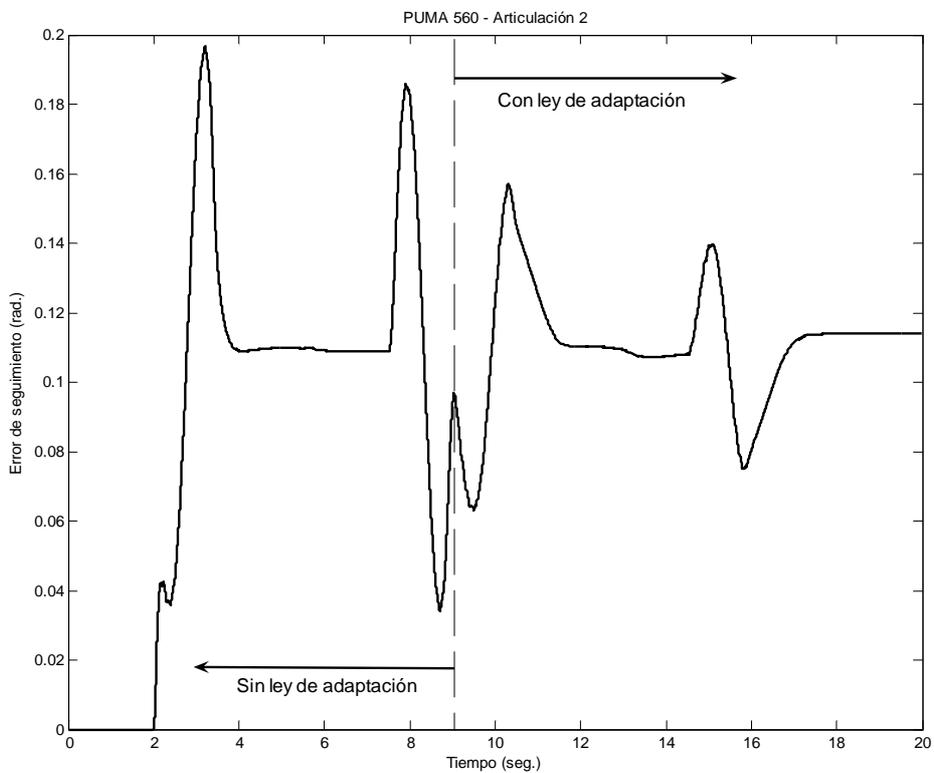


Figura 3-16: Error de seguimiento de la segunda articulación bajo la acción de un controlador PD (en la primera parte de la trayectoria) y bajo la acción del controlador ARC (en la segunda parte).

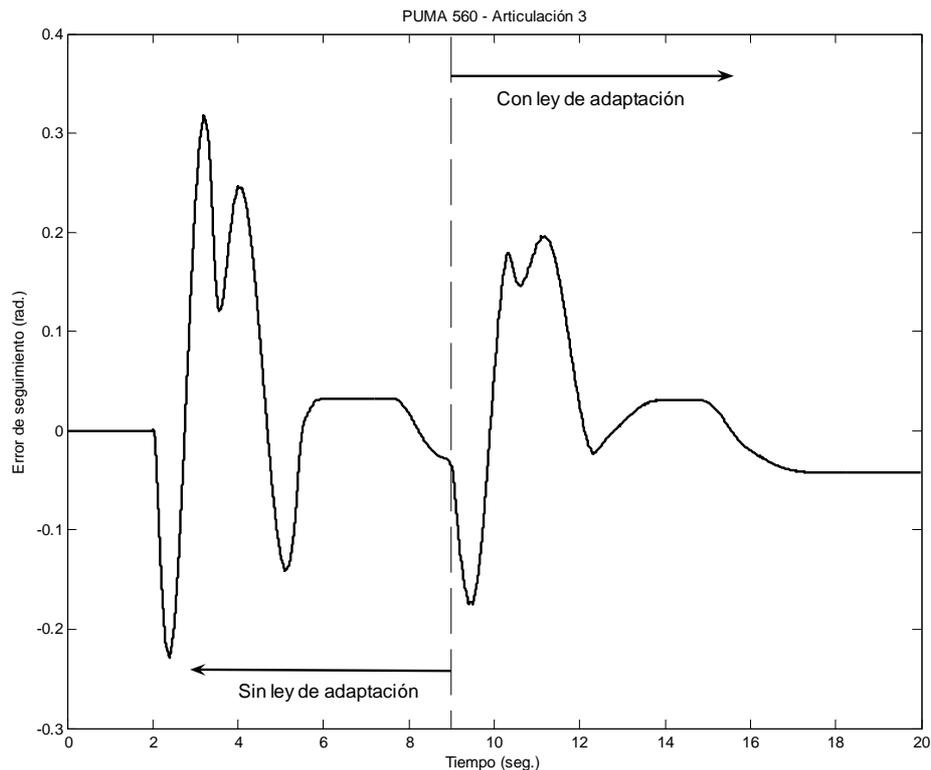


Figura 3-17: Error de seguimiento de la tercera articulación bajo la acción de un controlador PD (en la primera parte de la trayectoria) y bajo la acción del controlador ARC (en la segunda parte).

En la figura 3-18 se observa la evolución del parámetro de cota de incertidumbre para este experimento. En el primer tramo no hay acción robusta, por lo que este parámetro toma el valor $\rho = 0$, mientras que en el segundo se produce una autoadaptación del mismo, que conlleva la mencionada disminución del error de seguimiento, aún en presencia de las mencionadas incertidumbres paramétricas en el modelo.

Como conclusión, a la vista de los resultados obtenidos en los distintos experimentos realizados, el controlador propuesto ARC consigue un comportamiento satisfactorio para los objetivos de control planteados, en cuanto al seguimiento de trayectorias en presencia de incertidumbres en el modelo.

Dado un valor inicial para el parámetro (que incluso puede ser nulo), la ley de adaptación hace evolucionar el término robusto para conseguir una mejor eficiencia del mismo respecto a otros algoritmos de control robusto. Sin necesidad de realizar un estudio previo sobre acotación de incertidumbres o supuestos sobre las desviaciones máximas del modelo supuesto del manipulador, se consigue una adaptación del término robusto que consigue rebajar significativamente los errores de seguimiento de las trayectorias planificadas.

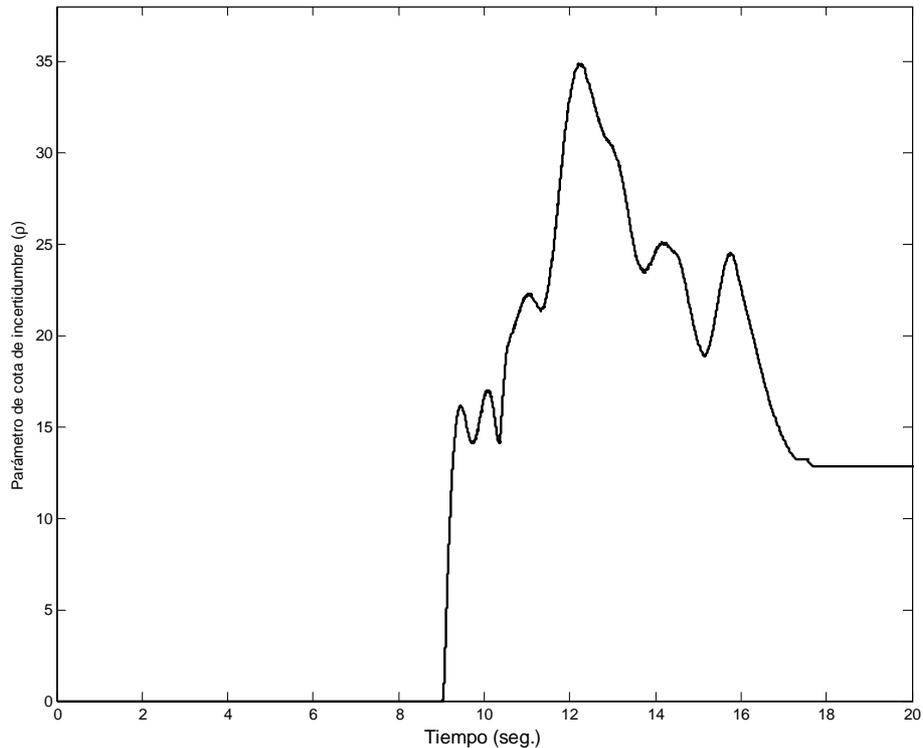


Figura 3-18: Evolución del parámetro de cota de incertidumbre para el caso del experimento 3.2.

Además, la carga computacional del algoritmo no es mucho mayor que para los algoritmos robustos existentes en la literatura. El mecanismo de adaptación de la acción robusta es lo suficientemente simple como para ser ejecutado en tiempo real por un procesador de gama media-alta.

Es de interés resaltar que en las pruebas realizadas sobre el manipulador PUMA-560 consideran un periodo de muestreo de 1 milisegundo, y que el algoritmo completo era ejecutado sin problemas. Asimismo, este periodo de muestreo es lo suficientemente pequeño como para considerar ciertas las premisas del teorema de estabilidad establecido en el capítulo anterior para el controlador ARC. Además, el modelo discreto asumido para el sistema es perfectamente válido para realizar el control del manipulador.

Con todo ello, se propone el controlador ARC como una estrategia de control robusta frente a incertidumbres en el modelo desconocidas o no acotadas y de bajo coste computacional.

Parte II

Inclusión de ligaduras
mediante técnicas
computacionalmente
eficientes

Algoritmos predictivos para la mejora de prestaciones

Uno de los problemas fundamentales que no abordan de manera eficiente los algoritmos clásicos de control de manipuladores es la inclusión de las restricciones a las que está sometido el sistema, tales como restricciones en el estado o en las entradas al sistema. La restricción en los valores de las entradas es especialmente importante, ya que la aparición de saturaciones en las entradas de los actuadores puede provocar la aparición de inestabilidad en el lazo cerrado, o bien que las prestaciones del controlador, en cuanto al seguimiento de la trayectoria de referencia, empeoren de manera apreciable. Las técnicas de control predictivo permiten una fácil introducción de las restricciones a la hora de realizar el cálculo de la ley de control, pues son parte de la formulación del problema de optimización que hay que resolver en cada instante de la trayectoria. Sin embargo, la formulación inicial del control predictivo MPC se basa en la resolución de dicho problema de optimización de manera on-line, con lo que para un sistema que requiere períodos de muestreo muy pequeños, como es el robot manipulador, se requiere una solución computacionalmente eficiente. En este capítulo se introduce una estrategia basada en interpolación que reduce la complejidad computacional del MPC, salvaguardando las propiedades del mismo en cuanto a optimalidad, rendimiento e inclusión de restricciones.

4.1 Introducción al control predictivo

El control predictivo está formado por una amplia variedad de metodologías de control que utilizan conceptos comunes, tales como optimización, predicción, horizonte móvil, etc. Tales metodologías incluyen el control óptimo, el control estocástico, el control multivariable o el control con restricciones ([Cam97] y más en detalle en [Cam95]).

Ya desde los primeros trabajos en este campo en los años sesenta, la utilización de un modelo del sistema que se va a controlar para realizar las

predicciones fue la nota común de los diseños de controladores predictivos. Surge entonces el Control Predictivo Basado en Modelos (MPC, *Model-Based Predictive Control*¹). Dicha estrategia, que en realidad aglutina a muchos algoritmos de control, se basa en resolver en cada instante de muestreo un problema de control óptimo de horizonte finito o infinito en lazo abierto y aplicar el comando resultante de esta optimización hasta el siguiente intervalo de muestreo. Resulta entonces una estrategia de lazo cerrado en la que la realimentación se utiliza para tomar nuevas medidas del sistema antes de resolver el siguiente problema de optimización.

Dicha optimización es llevada a cabo resolviendo un problema de minimización, tras haber definido una función de costo formada por la entrada que debe ser calculada² y el error futuro predicho para el sistema, utilizando para ello el modelo del sistema.

El control predictivo utiliza la idea de utilizar un horizonte móvil (*receding horizon*), consistente en desplazar el horizonte de predicción hacia el futuro en cada instante de muestreo, repitiéndose la optimización de la función de costo para el nuevo intervalo resultante. Esta idea queda reflejada en los primeros trabajos de Propoi [Pro63], aunque aplicado a otro tipo de estrategias.

Las distintas alternativas de control predictivo difieren en el tipo de modelo que se emplea para representar al sistema. Así, han surgido estrategias que utilizan modelos de respuesta impulso, respuesta escalón, función de transferencia o modelo de espacio de estados. Aunque al principio se utilizó con notable éxito el control predictivo en aplicaciones industriales, no fue hasta mucho después cuando estas técnicas se consolidaron como una alternativa eficaz a los controladores existentes, cuando aparecieron los primeros resultados que aseguraban la estabilidad de las mismas. Los campos de aplicación han sido muy variados, destacando las plantas eléctricas y petroquímicas, aunque también se ha aplicado en aplicaciones quirúrgicas, plantas de energía solar, servosistemas, etc.

En el siguiente apartado se describe en detalle el control predictivo basado en modelo. Se introducen en primer lugar las ideas básicas del controlador y, luego, se formulan los modelos de predicción y perturbación utilizados, la inclusión de ligaduras en el problema de control y la formulación del problema de optimización.

A continuación se describen los principales problemas que se deben solventar por parte de este tipo de estrategias, como son el coste computacional de las mismas y la factibilidad de las soluciones calculadas y las mejoras que se han ideado para acometerlas. De entre ellas, se describen los algoritmos predictivos con interpolación, pues son los empleados en esta tesis

¹ Inicialmente se denominaba con las siglas MBPC, pero esta notación ha sido paulatinamente sustituida por MPC.

² Frecuentemente se utiliza, en lugar de la entrada al sistema, los incrementos en la entrada.

para incluir las restricciones en el problema de control, manteniendo un bajo coste computacional del algoritmo de control.

Finalmente, se describen las especificidades que se han tenido en cuenta para la aplicación de estos algoritmos en el caso de manipuladores, el esquema de control empleado y la obtención de los modelos de predicción para este caso.

4.2 Control predictivo basado en modelo, MPC

Los controladores predictivos basados en modelo (MPC) aglutinan a un gran número de estrategias predictivas que comparten ideas comunes y que sólo se diferencian en detalles como son el tipo de modelo empleado, la elección de la función de costo del problema de optimización, el método de resolución de dicho problema de optimización, etc. Así, han surgido abreviaturas de controladores como IDCOM (*Identification/Command controller*, [Ric76],[Ric78]), DMC (*Dynamic Matrix Control*, [Cut80]), QDMC (*Quadratic programming Dynamic Matrix Control*, [Gar86]), IMC (*Internal Model Control*, [Gar82]), GPC (*Generalised Predictive Control*, [Cla87]), MPHIC (*Model Predictive Heuristic Control*, [Ric78]), MUSMAR (*Multistep Multivariable Adaptive Regulator*, [Men80]), etc.

Todas estas estrategias se pueden englobar en un conjunto de controladores cuyo objetivo es aplicar las entradas que llevarán al sistema a comportarse lo más eficazmente posible, dependiendo de las predicciones realizadas sobre el sistema en un horizonte temporal limitado. Para ello, se utiliza un modelo interno del proceso que se va a controlar. Además, esta decisión se realiza continuamente basándose en la respuesta que se obtenga del sistema.

4.2.1 Ideas básicas del controlador MPC

Las características principales del MPC se pueden resumir en los siguientes puntos:

- La ley de control depende del comportamiento predicho del sistema.
 - La predicción de las salidas y/o estados del sistema se realiza utilizando un modelo del sistema.
 - La entrada al sistema se determina resolviendo un problema de optimización que evalúa una cierta medida del comportamiento predicho (medidas sobre el error de seguimiento o sobre las entradas aplicadas, generalmente).
 - Se utiliza un horizonte móvil, en cuanto que se actualiza la entrada del controlador en cada instante de tiempo, situando el inicio del horizonte en el instante actual.
-

De esta manera surgen estrategias que contienen las siguientes ventajas:

- La idea básica del MPC es simple e intuitiva.
- El costo computacional de estas técnicas no es alto, pues requieren de unas matemáticas simples.
- Las ligaduras del sistema se incorporan de forma sencilla, ya que se tienen en cuenta de forma explícita en el algoritmo de optimización de la función de costo.
- El control de sistemas complejos, de procesos con retardo, sistemas inestables en lazo abierto, de fase no-mínima, sobreparametrizados e incluso multivariables es abordable de forma relativamente sencilla con estas técnicas.
- El diseño resultante de la aplicación de estas técnicas es un controlador lineal fácilmente implementable.

4.2.2 Esquema de control del MPC

Los controladores MPC se basan en el esquema que se detalla en la figura 4-1. La parte principal del algoritmo consiste en la definición del optimizador, es decir, en el proceso de optimización empleado para que, a partir de los errores predichos, se calcule la secuencia de entradas futuras que minimice la función y cumpla las ligaduras. La carga computacional del algoritmo de control depende de la complejidad computacional del algoritmo de optimización empleado para resolver el problema.

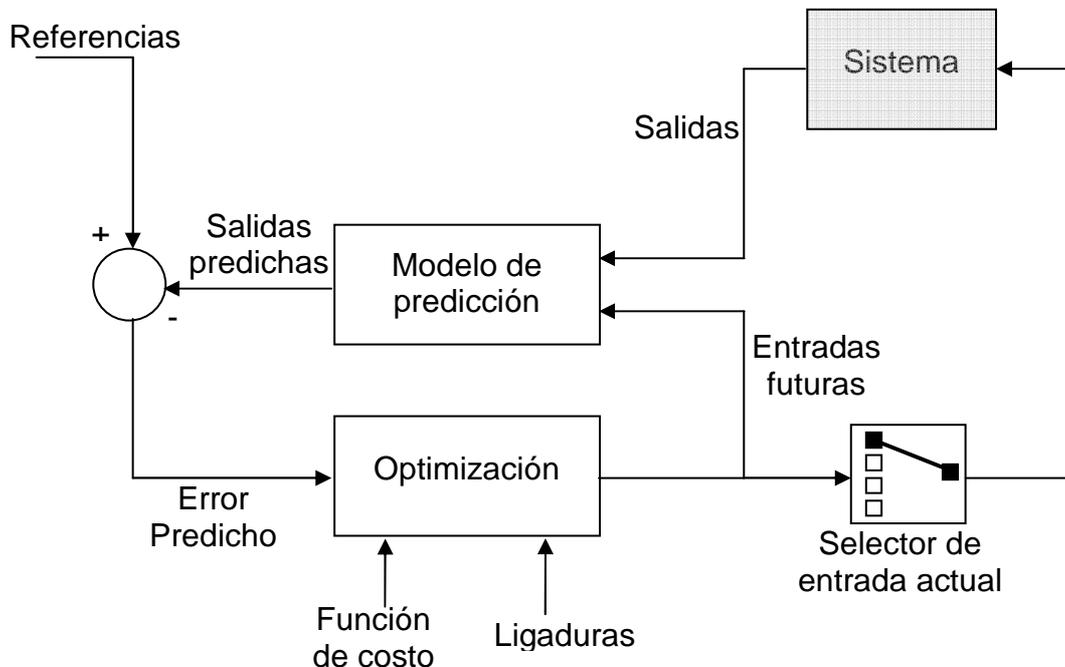


Figura 4-1 : Esquema de controlador MPC.

Como se observa, el modelo del sistema desempeña un papel fundamental, ya que se encarga de realizar las predicciones del sistema a partir de las entradas y salidas aplicadas con anterioridad, además de las entradas que se aplicarán en el futuro, pudiéndose así obtener una medida de los errores futuros del sistema. Estos errores son los que emplea el proceso de optimización para calcular las entradas futuras.

Es en este punto donde el controlador aplica el concepto de realimentación, ya que emplea medidas pasadas y actuales del sistema para realizar la predicción que utiliza el optimizador para generar las entradas futuras del sistema.

Así pues, los datos necesarios para llevar a cabo el algoritmo predictivo MPC son:

- la trayectoria de referencia o consignas del sistema,
- la función de costo del problema de optimización, la cual se elige generalmente de forma cuadrática para obtener su mínimo como una función lineal de las entradas y salidas anteriores, así como de la trayectoria de referencia,
- las ligaduras presentes en el sistema, las cuales dificultan el problema de optimización, haciendo que se pase de simplemente resolver la ecuación de Riccati (problema sin ligaduras), a resolver un problema de complejidad mucho mayor,
- los horizontes de predicción (un aumento del horizonte de predicción comporta una mejor solución, pero aumenta enormemente la dimensionalidad del problema).

4.2.3 Algoritmo del controlador MPC

Para la implementación de este controlador hay que seguir tres pasos en cada instante de la trayectoria, los cuales conforman el siguiente algoritmo:

Paso 0) Inicialización.

- Adoptar un modelo del sistema para la predicción de su comportamiento en instantes futuros.
- Fijar el horizonte de predicción N .
- Elegir una función de coste, generalmente cuadrática, que pese los errores de las salidas predichas respecto a la trayectoria de referencia. Generalmente, se elige una función cuadrática que incluya, además, la aplicación de entradas demasiado grandes al sistema (evitando que el controlador sea no factible).

Paso 1) Predicción.

- Predecir, en el instante de tiempo actual k , el valor de la salida del sistema para los instantes $k+1, \dots, k+N$, que se denota por
-

$\hat{y}(k+j|k)$ ³. Para ello se hace uso de las entradas y salidas hasta el instante k , así como de las entradas futuras $u(k+j|k)$, con $j=0, \dots, N-1$, resultantes del paso de optimización anterior.

Paso 2) Control.

- Calcular la secuencia de entradas $u(k+j|k)$ que minimicen la función de costo elegida y que cumplan las ligaduras impuestas al sistema. Generalmente, se trata de resolver un problema numérico de dimensión igual al horizonte de predicción N , el cual suele reducirse empleando un horizonte de control M , con $M < N$, de forma que a partir del instante $k+M$ y hasta el instante $k+N$ se aplique la entrada $u(k+M|k)$.

Paso 3) Horizonte móvil.

- Aplicar, en cada instante k , la primera entrada obtenida al resolver el problema de optimización del Paso 2. (El resto de entradas no se tienen en cuenta.)
- Mover, en el siguiente instante $k+1$, el horizonte de predicción (y el de control, si lo hubiere) colocando el principio del rango en dicho instante $k+1$, de forma que se formule de nuevo la optimización para calcular $u(k+1+j|k+1)$, con $j=0, \dots, N-1$.

Paso 4) Hacer $k=k+1$ e ir al Paso 1.

En la figura 4-2 se observa un ejemplo de aplicación del controlador MPC en el instante k , dada la trayectoria de referencia o consigna $y_r(k+j)$, con $j=1, \dots, N$, siendo N el horizonte de predicción, en donde se calculan las entradas futuras $u(k+M|k)$, donde M es el horizonte de control, en base a las salidas predichas $\hat{y}(k+j|k)$, utilizando el modelo de predicción del sistema.

4.2.4 Estabilidad en el MPC

Ya desde la aparición de los controladores predictivos expuestos en el apartado 4.2.1, y muchas otras estrategias afines, se trató de obtener resultados de estabilidad que aseguraran un comportamiento óptimo en lazo cerrado, cuando el sistema estuviera sujeto a ligaduras. Un resultado que asegura dicha estabilidad es imponer una restricción en el estado del sistema en el punto final de la trayectoria.

³ Esta es la notación utilizada habitualmente para denotar las predicciones. Significa, en este caso, las salidas predichas (de ahí el uso del símbolo $\hat{\cdot}$) para el instante $k+j$, haciendo uso de la información obtenida del sistema (entradas y salidas) hasta el instante k .

Así surgen los esquemas de control predictivo SIORHC (*Stabilizing Input-Output Receding-Horizon Control*, [Mos90]), que se plantean resolver un problema LQ con ligaduras en el error de seguimiento final y en las entradas al sistema. También el CRHPC (*Constrained Receding-horizon Predictive Control*, [Cla91]), basado en asegurar que la salida alcance la referencia en un rango de tiempo limitado. En la misma línea se sitúan los esquemas SGPC (*Stable Generalized Predictive Control*, [Kou92]) y CSGPC (*Constrained SGPC*, [Ros93]), en los que se incluyen en el lazo de realimentación un controlador de estabilización interna.

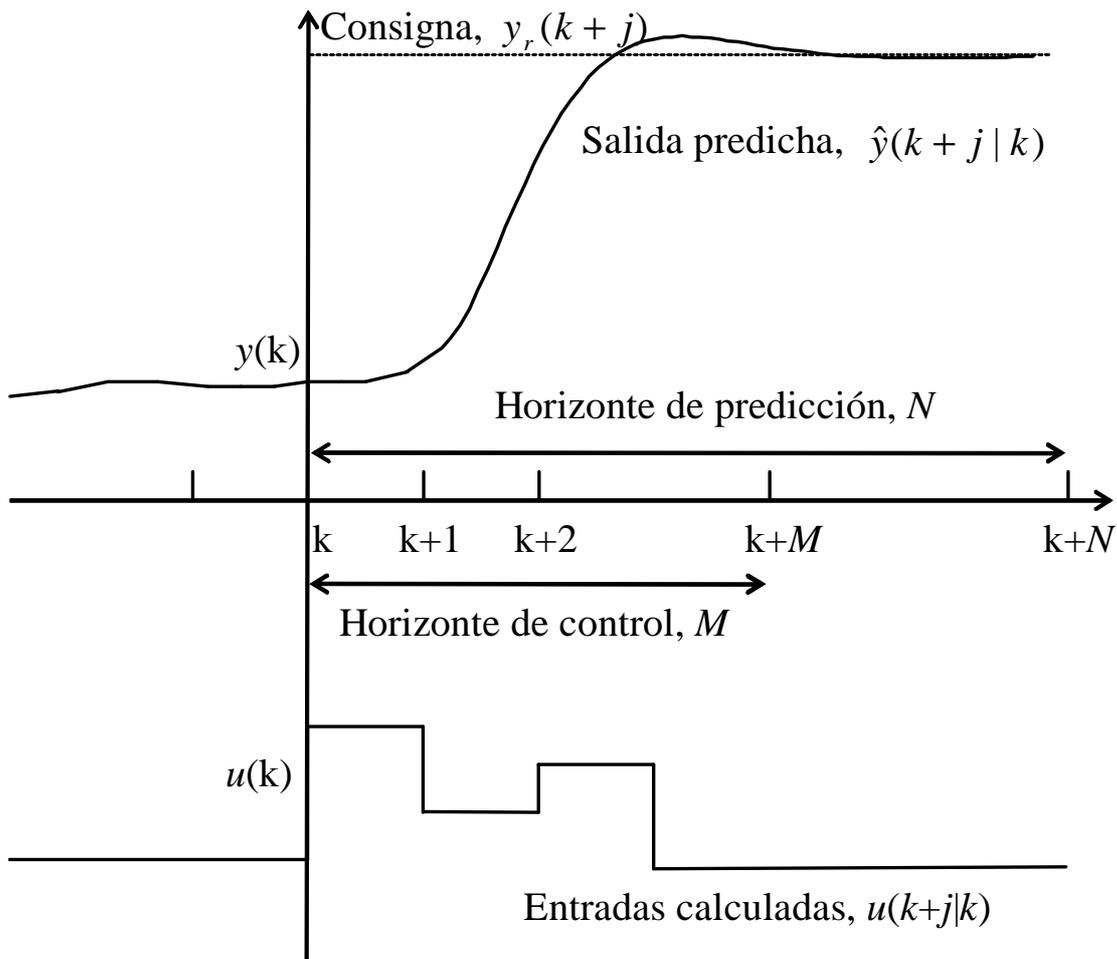


Figura 4-2 : Entradas de control calculadas por un MPC teniendo en cuenta la salida predicha por el modelo de predicción y la consigna para el sistema.

Para asegurar la factibilidad de las soluciones aportadas por los controladores predictivos SGPC y CSGPC, Kouvaritakis y Rossiter han propuesto diversas mejoras [Kou93], [Ros95], [Ros96] en la que se evita la pérdida de factibilidad de las estrategias por violaciones de las ligaduras debidas a las ligaduras impuestas para el seguimiento de trayectorias. Surge así el algoritmo MCSGPC (*Modified CSGPC*, [Ros96]) el cual, aún en presencia

de perturbaciones, garantiza estabilidad y seguimiento asintótico de la trayectoria de referencia [Gos97].

Sobre este tipo de estrategias se incidirá posteriormente, pues son las que se aplican al robot para tratar el problema de las restricciones impuestas al robot, tanto en las entradas como en el estado del sistema. Cabe nombrar aquí algunas otras estrategias predictivas, tales como el GPCW (*Generalised Predictive Control with End-point State Weighting*, [Dem93a]), el cual combina las soluciones GPC y CRHPC en función del valor que se le dé al peso sobre el estado final en la función de costo. Asimismo, una formulación del GPC para sistemas continuos es el CGPC (*Continuous GPC*, [Dem91]). Para sistemas multivariables continuos se propusieron los controladores MCGPC (*Multivariable CGPC*, [Dem92]) y MUPC (*Multivariable Unified Predictive Control*, [Vri94]).

4.2.5 Modelo predictivo del MPC

Seleccionar un modelo del proceso para el MPC es la parte más importante en el diseño del algoritmo de control predictivo, ya que una mala elección del mismo conlleva, en la mayoría de las ocasiones, un comportamiento insatisfactorio del controlador. El modelo debe representar fielmente la dinámica del sistema que se va a controlar, de forma que las predicciones que se hagan del mismo sean lo suficientemente veraces como para ser empleadas por el algoritmo de optimización. No debe elegirse un modelo que, por su complejidad, conlleve una ralentización en la aplicación del controlador.

La gran mayoría de estrategias utilizan un modelo lineal para representar la dinámica del sistema. Esto se debe a que la elección de un modelo no lineal conlleva métodos numéricos para el algoritmo de optimización más complejos y de difícil implementación, incluso carentes de convergencia a una solución de mínimo de la función de costo.

Además, teniendo en cuenta la propiedad de superposición de los sistemas lineales, se modela por un lado la dinámica del proceso y, por otro, las perturbaciones. Así, la salida global del modelo será la suma de las salidas correspondientes a ambos modelos.

Sin embargo, existen aproximaciones (se pueden ver excelentes resúmenes de diversas estrategias en [All99] y [Mor99], así como aplicaciones industriales de estas técnicas en [Qin00]) que tienen en cuenta la naturaleza no lineal del proceso que se va a controlar y utilizan un modelo no lineal del mismo, de forma que las predicciones en la salida se obtienen como:

$$\hat{y}(k+j|k) = f(k, \theta, \mathbf{U}(k), \mathbf{Y}(k)) \quad (4-1)$$

donde $f(\)$ es la función no lineal que describe el proceso, θ es un vector de parámetros del modelo y $\mathbf{U}(k)$ e $\mathbf{Y}(k)$ son vectores con la historia de las entradas y salidas aplicadas al sistema.

4.2.5.1 Modelos de predicción

Según el modelo predictivo utilizado para modelar la dinámica del sistema, surgen distintas formas de plantear las predicciones que serán utilizadas por el algoritmo de optimización para determinar la secuencia de control que minimice la función de costo elegida.

De entre los distintos algoritmos MPC, podemos extraer los siguientes modelos de predicción:

- Modelos de espacio de estados, utilizados en la estrategia PFC (*Predictive Functional Control*, en [Ric87], donde además aparece una primera aplicación del MPC a robots industriales, y en [Lee94]).
- Modelos de función de transferencia, utilizados en las estrategias UPC, MUSMAR y GPC, entre otras.
- Modelos FIR (*Finite Impulse Response*); dentro de esta categoría podemos encontrar los siguientes modelos:
 - Modelos de respuesta al impulso, utilizados en la estrategia IDCOM y algunos casos de GPC.
 - Modelos de respuesta al escalón, utilizados en las estrategias DMC y QDMC.

A) Modelos de espacio de estados

Dada la forma habitual de representación de un sistema en su formulación de espacio de estados:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \quad (4-2)$$

las predicciones se harán mediante la ecuación:

$$\hat{\mathbf{y}}(k+j|k) = \mathbf{C} \left[\mathbf{A}^j \mathbf{x}(k) + \sum_{i=1}^j \mathbf{A}^{j-i} \mathbf{B}u(k+i|k) \right] \quad (4-3)$$

B) Modelos de función de transferencia

Uno de los modelos más utilizados para representar la dinámica del sistema es el modelo CARIMA (*Controlled Auto-Regressive Integrated Moving Average model*, [Cla87]), dado por:

$$a(z^{-1})y(k) = b(z^{-1})u(k) + \frac{T(z^{-1})}{\Delta(z^{-1})}v(k) \quad (4-4)$$

donde z^{-1} es el operador retardo unitario y $\Delta(z^{-1}) = 1 - z^{-1}$, polinomio que añade al sistema de forma explícita un integrador para que presente un error estacionario nulo cuando se controla en lazo cerrado. $v(k)$ es una variable aleatoria de media cero que representa los efectos de perturbaciones y ruido de medida, mientras que $T(z)$ representa la dinámica de dichas perturbaciones en el sistema global. Los polinomios del modelo se escogen como:

$$a(z^{-1}) = 1 + a_1 z^{-1} + \dots + a_n z^{-n} \quad (4-5)$$

$$b(z^{-1}) = b_1 z^{-1} + \dots + b_n z^{-n} \quad (4-6)$$

siendo $\{a_i; i = 1, \dots, n\}$ y $\{b_i; i = 1, \dots, n\}$ los parámetros del modelo⁴. Habitualmente en vez del modelo (4-4), se toma el modelo en incrementos:

$$[\Delta(z^{-1})a(z^{-1})]y(k) = b(z^{-1})\Delta u(k) + T(z^{-1})v(k) \quad (4-7)$$

siendo $\Delta u(k) = u(k) - u(k-1)$ resultado de aplicar el operador $\Delta(z^{-1})$ sobre la entrada del sistema. De esta forma el último término, que es aleatorio de media cero, puede ser asumido que es cero en los instantes futuros, con lo que las predicciones vienen dadas como:

$$\hat{y}(k+j|k) = \frac{b(z^{-1})}{\Delta(z^{-1})a(z^{-1})} \Delta u(k+j|k) \quad (4-8)$$

C) Modelos de respuesta al impulso

En este caso, el sistema viene dado por:

$$y(k) = G(z^{-1})u(k) \quad (4-9)$$

donde

$$G(z^{-1}) = \sum_{i=0}^{\infty} G_i z^{-i} \quad (4-10)$$

siendo G_i las respuestas del sistema, en los instantes de muestreo, a una entrada impulso unitario de anchura el período de muestreo. La salida estimada del sistema tomará la forma:

$$\hat{y}(k+j|k) = \sum_{i=1}^n G_i u(k+j-i|k) \quad (4-11)$$

⁴ Los parámetros del modelo se obtienen utilizando un método de identificación estándar, como puede ser el de mínimos cuadrados o el de mínimos cuadrados recursivos, incluyendo factor de olvido, etc.

donde n es tal que, a partir del instante $k+n$, las constantes G_i se desprecian.

D) Modelos de respuesta al escalón

En este caso, la función que representa la respuesta del sistema al escalón es:

$$H(z^{-1}) = \sum_{i=0}^{\infty} H_i z^{-i} \quad (4-12)$$

donde H_i representa las respuestas del sistema ante una entrada escalón en los instantes de muestreo. Por definición, se tiene que:

$$H(z^{-1}) = G(z^{-1})/\Delta(z^{-1}) \quad (4-13)$$

Por tanto, la estimación de las salidas del sistema vendrá dada por:

$$\hat{y}(k+j|k) = \sum_{i=1}^n H_i \Delta u(k+j-i|k) \quad (4-14)$$

donde de nuevo se desprecian los términos H_i a partir del instante $k+n$.

E) Comparación de modelos de predicción

Los modelos de predicción basados en modelos FIR presentan como ventaja que no se necesita ninguna suposición acerca del orden del sistema o del retardo del mismo. Sin embargo, necesitan una gran cantidad de términos para modelar de forma precisa el sistema y garantizar convergencia. Además, la respuesta al impulso no es una formulación intuitiva en la práctica.

En el caso de los modelos de función de transferencia, se debe conocer el orden del sistema para así determinar el orden de los polinomios (4-5) y (4-6). Presentan, por el contrario, como ventaja que puede representar plantas inestables, incluso con un número de parámetros mínimo.

En cuanto a los modelos de espacio de estados, tienen como ventajas que se adaptan fácilmente al caso multivariable y ofrecen una buena representación interna (estados) del sistema. Sin embargo, la determinación del estado conllevará en ocasiones la necesidad de utilizar observadores. Además, pueden requerir un alto coste computacional para dar las estimaciones del sistema.

4.2.5.2 Modelo de perturbaciones

Los algoritmos MPC pueden incorporar fácilmente la presencia de perturbaciones en el modelo para, de esta manera, modelar efectos no deseados como el ruido de medida, las incertidumbres en los parámetros del modelo, las perturbaciones procedentes del entorno del sistema, etc.

El objetivo es disponer de un modelo del sistema más cercano a la realidad y, por tanto, cuyas predicciones sean más veraces. De esta forma se conseguirá un mejor rechazo a perturbaciones por parte del controlador.

La elección del modelo de perturbaciones para el modelo predictivo tiene también una gran importancia, ya que permite modelar perturbaciones ambientales, errores de modelado, ruido de medida, etc. Estos modelos de perturbaciones, tal y como se comentó en el apartado 4.2.5, dan una salida que, aplicando el principio de superposición para sistemas lineales, conformará la salida global junto a la salida del modelo de predicción elegido.

La mayoría de los controladores del tipo MPC utilizan para modelar las perturbaciones uno de los tres modelos siguientes.

A) Modelado de incertidumbres y acción integral

En el caso de sistemas que no estén afectados de incertidumbres o perturbaciones, se puede conseguir un control con error nulo en el estacionario sin necesidad de acción integral. Pero, en la práctica, las incertidumbres presentes en los parámetros del sistema y las perturbaciones requieren la presencia de un integrador para rechazar eficazmente su efecto [Ros03].

Lo más común, por tanto, es que el modelo de incertidumbres incluya en la ley predictiva un integrador, de forma que se consiga un seguimiento de trayectoria sin errores incluso en presencia de incertidumbres en los parámetros.

La inclusión de las perturbaciones es más complicada y depende muchísimo del sistema que se va a controlar. Por lo general, se asume un modelo de perturbaciones y, mediante prueba y error, se rediseña el mismo para obtener un control más satisfactorio.

B) Modelado del ruido de medida

En muchas ocasiones, el ruido de medida, así como otras incertidumbres presentes en el proceso, se asume que es ruido blanco y se ignora. Pero, en presencia de ruido coloreado, se suelen utilizar filtros específicos diseñados en el rango de frecuencias en cuestión.

Lo habitual es el uso de ciertos filtros paso-bajo para el rechazo a esta clase de perturbaciones, aunque su uso sistemático en controladores MPC no está todavía asentado entre la comunidad científica.

C) Modelo en el espacio de estados

Si se modela una perturbación en la salida, lo habitual es definirlo como ruido blanco integral, de modo que la salida global del modelo venga dada por:

$$y(k) = Cx(k) + d(k) \quad (4-15)$$

donde la perturbación $d(k)$ se modela como:

$$d(k) = d(k-1) + v(k) \quad (4-16)$$

siendo $v(k)$ una variable aleatoria de media cero. Para incluir esta perturbación en el modelo, se define un estado ampliado del modelo $z(k) = [x(k) \ d(k)]^T$ de forma que la ecuación de estado es:

$$\begin{aligned} z(k) &= \hat{\mathbf{A}}z(k-1) + \hat{\mathbf{B}}u(k-1) \\ y(k) &= \hat{\mathbf{C}}z(k) + v(k) \end{aligned} \quad (4-17)$$

donde:

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}; \quad \hat{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}; \quad \hat{\mathbf{C}} = [\mathbf{C} \ \mathbf{I}] \quad (4-18)$$

Si se modela una perturbación en el estado, entonces la ecuación de estado queda de la siguiente forma:

$$x(k+1) = Ax(k) + Bu(k) + Fd(k) \quad (4-19)$$

En este caso, la ecuación de estado queda de la misma forma que en (4-17) pero con la siguiente definición de matrices:

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{F} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}; \quad \hat{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}; \quad \hat{\mathbf{C}} = [\mathbf{C} \ \mathbf{0}] \quad (4-20)$$

D) Modelo de función de transferencia

La ecuación (4-4), o equivalentemente (4-7), que expresa el modelo del sistema siguiendo la formulación del modelo CARIMA, contiene el término de perturbaciones en el sistema:

$$d(k) = \frac{T(z^{-1})}{\Delta(z^{-1})} v(k) \quad (4-21)$$

En particular, si se escoge $T(z^{-1}) = 1$, se recupera la expresión (4-16) de perturbación en el espacio de estados.

E) Modelos FIR

En este caso, la perturbación se añade directamente a la salida, de forma que la expresión en el caso de modelos de respuesta al impulso (4-9) queda:

$$y(k) = G(z^{-1})u(k) + d(k) \quad (4-22)$$

Mientras que para los modelos de respuesta al escalón queda:

$$y(k) = H(z^{-1})\Delta u(k) + d(k) \quad (4-23)$$

4.2.6 Ligaduras en el MPC

Una de las principales ventajas que presenta la formulación del control predictivo MPC es la facilidad para definir las ligaduras a las que está sometido el proceso, ya que se incorporan de forma natural al algoritmo de optimización.

Las ligaduras más habituales a las que está sometido un sistema en la práctica son:

- Entradas mínima y máxima que se pueden aplicar al sistema, debidas a la potencia máxima que es capaz de suministrar el actuador asociado a dicha entrada, o bien a restricciones físicas del propio sistema:

$$u_{\min} \leq u \leq u_{\max} \quad (4-24)$$

- Variación máxima de la entrada al sistema, debida a la limitación de los actuadores para cambiar el valor de la entrada en cortos instantes de tiempo:

$$\Delta u_{\min} \leq \Delta u \leq \Delta u_{\max} \quad (4-25)$$

- Valores extremos de la salida del sistema, o también del estado del sistema, debidos a restricciones mecánicas, físicas, electrónicas, de seguridad, etc.

$$y_{\min} \leq y \leq y_{\max} \quad (4-26)$$

$$x_{\min} \leq \mathbf{x} \leq x_{\max}$$

Estas expresiones para las ligaduras se pueden expresar de manera única en notación matricial tal y como sigue [Men98]:

$$\mathbf{R}u \leq \mathbf{c} \quad (4-27)$$

donde

$$\mathbf{R} = \begin{bmatrix} \mathbf{I}_{NxN} \\ -\mathbf{I}_{NxN} \\ \mathbf{T} \\ -\mathbf{T} \\ \mathbf{G} \\ -\mathbf{G} \end{bmatrix}; \mathbf{c} = \begin{bmatrix} \mathbf{I}_{Nx1} \Delta u_{\max} \\ -\mathbf{I}_{Nx1} \Delta u_{\min} \\ \mathbf{I}_{Nx1} u_{\max} - \mathbf{I}_{Nx1} u(k-1) \\ -\mathbf{I}_{Nx1} u_{\min} + \mathbf{I}_{Nx1} u(k-1) \\ \mathbf{I}_{Nx1} y_{\max} - f \\ -\mathbf{I}_{Nx1} y_{\min} + f \end{bmatrix} \quad (4-28)$$

siendo \mathbf{G} un término resultante de la solución de la ecuación diofántica resuelta para obtener las predicciones necesarias para el algoritmo de optimización⁵, \mathbf{I}_{Nx1} un vector de dimensión N con todos sus elementos iguales a uno y \mathbf{T} una matriz triangular inferior con todos sus elementos iguales a uno.

Nótese que, expresando las ligaduras de esta forma, la resolución del algoritmo de optimización para obtener la ley de control se reduce a la resolución de un problema de programación cuadrática. Esto se puede ver claramente en el siguiente apartado, donde se da la expresión habitual de la función de costo elegida en el algoritmo de optimización.

Pueden aparecer otro tipo de ligaduras en la formulación del controlador predictivo. Una ligadura habitual consiste en forzar el estado final del sistema para que coincida con uno preestablecido. Este tipo de ligaduras añaden estabilidad a la ley de control. Otro tipo de ligaduras se definen para evitar las no linealidades presentes en el sistema, tales como zonas muertas o ciclos histeréticos. También se pueden imponer ligaduras para hacer que el sistema tenga un comportamiento no oscilatorio e, incluso, para restringir la amplitud de oscilación máxima (o sobrepasamiento) del sistema, con el fin de mejorar el comportamiento dinámico del mismo. Todas estas ligaduras se pueden expresar según la ecuación (4-27), con una elección adecuada de los vectores \mathbf{R} y \mathbf{c} .

4.2.7 Elección de la función de costo en MPC

El algoritmo de optimización del controlador MPC se basa en la minimización de una función de costo que incluye las predicciones calculadas por el modelo predictivo. Por tanto, la elección de la función de costo es fundamental para la obtención de un buen resultado por parte del controlador.

A continuación se exponen las distintas posibilidades de elección de la función de costo, de las que surgen los distintos problemas de optimización que hay que resolver para calcular la secuencia de control. Dichas posibilidades se obtienen de considerar horizontes de predicción finitos o infinitos y de tener en cuenta o no las ligaduras del sistema.

⁵ Habría que dirigirse a [Men98] para una explicación detallada de cada uno de los términos que aparecen en la expresión compacta de las ligaduras del sistema.

Estos problemas son exactamente los mismos que se plantean en el control óptimo, no en vano el control predictivo se puede definir como la resolución de un problema de control óptimo en cada instante de la trayectoria.

4.2.7.1 Posibilidades de formulación de la función de costo

En primer lugar, hay que destacar que la formulación del problema de optimización se puede plantear en el espacio de las entradas-salidas o bien en el espacio de los estados. Así, para el primero, una función de costo habitual es:

$$J = \sum_{i=1}^N \left(\|r(k+i) - y(k+i|k)\|_2 \right)^2 + \eta \sum_{i=0}^{M-1} \left(\|\Delta u(k+i)\|_2 \right)^2 \quad (4-29)$$

o bien:

$$J = \sum_{i=1}^N \left(\|e(k+i|k)\|_2 \right)^2 + \eta \sum_{i=0}^{M-1} \left(\|\Delta u(k+i)\|_2 \right)^2 \quad (4-30)$$

donde $\| \cdot \|_2$ expresa la norma-2 del argumento, $r(k)$ es la trayectoria de referencia, $e = r - y$ es el error de seguimiento y Δu es el incremento de la entrada respecto al instante anterior.

El factor η pondera la importancia de la variación de las entradas frente al seguimiento de la trayectoria de referencia. N es el horizonte de predicción⁶ y M es el horizonte de control, con lo que se supone que:

$$\Delta u(k+i) = 0 \quad , \quad i \geq M \quad (4-31)$$

Para el caso del planteamiento en el espacio de estados, la función de costo equivalente es:

$$J = \sum_{i=1}^N \left(\mathbf{x}(k+i|k)^T \mathbf{Q} \mathbf{x}(k+i|k) \right) + \sum_{i=0}^{M-1} \left(u(k+i)^T \mathbf{R} u(k+i) \right) \quad (4-32)$$

donde $\mathbf{Q} \geq 0$ y $\mathbf{R} > 0$ pesan la contribución de ambos términos. Ahora, el estado $\mathbf{x}(k)$ y las predicciones $\mathbf{x}(k+i|k)$ expresan la diferencia respecto al estado de consigna, por lo que la entrada $u(k)$ es la desviación respecto a la entrada que lleva al sistema a dicho estado de consigna.

⁶ En general, se define la sumatoria en los errores de seguimiento desde $i=N_1$ hasta $i=N_2$, pero lo habitual es tomar $N_1=1$ y $N_2=N$. Cuando el sistema presenta retardo se suele tomar un valor adecuado de N_1 para evitar que se hagan predicciones en el rango de tiempo afectado por el retardo.

En muchas ocasiones la función de costo se define de la siguiente forma:

$$J = \sum_{i=0}^N \left(x(k+i|k)^T \mathbf{Q} x(k+i|k) + u(k+i)^T \mathbf{R} u(k+i) \right) \quad (4-33)$$

Se supone que el estado actual de consigna es alcanzado, $\mathbf{x}(k) = 0$, por lo que el índice de la sumatoria se inicia en el valor $i = 0$.

El planteamiento de los distintos problemas de optimización se realiza según esta segunda representación, ya que será utilizada en los algoritmos de control empleados en el manipulador, como se verá en capítulos sucesivos.

Hay ciertos algoritmos MPC que definen funciones de costo diferentes. Por ejemplo, el IDCOM únicamente pesa los errores predichos $e(k+i|k)$ [Ric78].

4.2.7.2 Problema de optimización de tiempo infinito sin restricciones

Este caso corresponde al problema original de MPC en el cual la función de costo considera un horizonte de predicción N infinito y la optimización no tiene en cuenta las ligaduras:

$$\min_{u(k), u(k+1), \dots} \sum_{i=0}^{\infty} \left(\mathbf{x}(k+i|k)^T \mathbf{Q} \mathbf{x}(k+i|k) + u(k+i)^T \mathbf{R} u(k+i) \right) \quad (4-34)$$

La solución a este problema para el sistema (4-2), siendo A y B de forma que el sistema es controlable, es el controlador de realimentación de estados [Rod96]:

$$u(k) = -\mathbf{K}x(k) \quad (4-35)$$

donde la ganancia viene dada por:

$$\mathbf{K} = (\mathbf{B}^T \mathbf{S} \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^T \mathbf{S} \mathbf{A} \quad (4-36)$$

y $\mathbf{S} = \mathbf{S}^T$ es la solución única de la ecuación de Riccati:

$$\mathbf{S} = \mathbf{A}^T (\mathbf{S} - \mathbf{S} \mathbf{B} (\mathbf{B}^T \mathbf{S} \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^T \mathbf{S} \mathbf{A}) \mathbf{A} + \mathbf{Q} \quad (4-37)$$

La expresión (4-35) recibe el nombre de regulador cuadrático lineal o solución LQR (en adelante LQ), debido a que da solución de regulación del estado en torno al origen a un sistema lineal con una función de costo cuadrática.

4.2.7.3 Problema de optimización de tiempo finito sin restricciones

Cuando se considera la solución hasta el horizonte de predicción N , el problema que se va a resolver es:

$$\min_{u(k), u(k+1), \dots, u(k+N-1)} \left\{ \sum_{i=0}^{N-1} (\mathbf{x}(k+i|k))^T \mathbf{Q} \mathbf{x}(k+i|k) + u(k+i)^T \mathbf{R} u(k+i) + \mathbf{x}(k+N|k)^T \mathbf{Q}_N \mathbf{x}(k+N|k) \right\} \quad (4-38)$$

En este caso, la solución viene dada por el controlador por realimentación de estados:

$$u(k) = -\mathbf{K}(k)\mathbf{x}(k) \quad (4-39)$$

donde

$$\mathbf{K}(k) = (\mathbf{B}^T \mathbf{S}(k+1)\mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^T \mathbf{S}(k+1)\mathbf{A} \quad (4-40)$$

y las matrices $\mathbf{S}(k) = \mathbf{S}(k)^T$ vienen dadas de forma recursiva por las ecuaciones:

$$\begin{aligned} \mathbf{S}(N) &= \mathbf{Q}_N \\ \mathbf{S}(k) &= \mathbf{A}^T (\mathbf{S}(k+1) - \mathbf{S}(k+1)\mathbf{B}(\mathbf{B}^T \mathbf{S}(k+1)\mathbf{B} + \mathbf{R})\mathbf{B}\mathbf{S}(k+1))\mathbf{A} + \mathbf{Q}, \quad k = 1, \dots, N-1 \end{aligned} \quad (4-41)$$

4.2.7.4 Problema de optimización de tiempo finito con restricciones

Cuando se tienen en cuenta las restricciones comentadas en el apartado 4.2.6, el problema de minimización de una función cuadrática sujeta a ligaduras lineales del tipo (4-27) conlleva un problema de programación cuadrática, cuya resolución es sistemática.

En este caso, el problema que se debe resolver es:

$$\min_{u(k), u(k+1), \dots, u(k+N-1)} \left\{ \sum_{i=0}^{N-1} (\mathbf{x}(k+i|k))^T \mathbf{Q} \mathbf{x}(k+i|k) + u(k+i)^T \mathbf{R} u(k+i) + \mathbf{x}(k+N|k)^T \mathbf{Q}_N \mathbf{x}(k+N|k) \right\} \quad (4-42)$$

sujeeto a:

$$\begin{aligned} \mathbf{x}(k+i|k) &\in \mathfrak{X}, \quad i = 1, \dots, N-1 \\ u(k+i) &\in \mathfrak{U}, \quad i = 0, \dots, N-1 \\ \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \end{aligned}$$

donde $\mathfrak{X} \subseteq \mathfrak{R}^n$ y $\mathfrak{U} \subseteq \mathfrak{R}^m$ son politopos que contienen el origen de los espacios respectivos, siendo n el número de estados del sistema y m el número de

entradas. En ocasiones, se tiene en cuenta una restricción más para el estado $\mathbf{x}(k+N|k)$, para asegurar el cumplimiento de las restricciones y la estabilidad del lazo cerrado.

Para la solución del mismo se hace uso de la ecuación de evolución de estados contenida en las restricciones, de forma que:

$$\mathbf{x}(k+i|k) = \mathbf{A}^i \mathbf{x}(k) + \sum_{j=0}^{i-1} \mathbf{A}^j \mathbf{B} u(k-1-j|k) \quad (4-43)$$

Con lo que el problema (4-42) se puede reformular como un problema de programación cuadrática (QP, *Quadratic Program*):

$$\min_{\mathbf{u}(k+N)} \left(\mathbf{x}(k+N)^T \mathbf{F} \mathbf{u}(k+N) + \mathbf{u}(k+N)^T \mathbf{G} \mathbf{u}(k+N) \right) \quad (4-44)$$

sujeto a:

$$\mathbf{H} \mathbf{u}(k+N) \leq \mathbf{W} + \mathbf{L} \mathbf{x}(k)$$

donde

$$\mathbf{U}(k+N) = \left[u(k)^T \quad u(k+1)^T \quad \dots \quad u(k+N-1)^T \right]^T \in \mathfrak{R}^{N \cdot m} \quad (4-45)$$

es el vector a optimizar y \mathbf{F} , \mathbf{G} , \mathbf{H} , \mathbf{W} y \mathbf{L} se obtienen de la ecuación de estados, las restricciones en (4-42) y las matrices \mathbf{Q} , \mathbf{Q}_N y \mathbf{R} [Mac02].

Para este caso, pues, la solución viene dada por la resolución de un problema de programación cuadrática. Aunque se han estudiado algoritmos QP eficientes [Ger01], hay trabajos en los que se reduce el coste computacional de la estrategia. Así, en [Bem02] se idea una solución en la que la mayoría del cálculo se realiza *off-line*, aunque la complejidad del algoritmo es mayor a medida que el horizonte de predicción aumenta.

Dicha solución está basada en encontrar una solución afín a intervalos, o solución PWA (*PieceWise Affine*), definida previamente a la resolución del problema, a partir de una partición del espacio de entradas y salidas o del espacio de los estados. El problema de optimización queda entonces definido como un problema QP multiparamétrico. En [Ton03] se amplía esta idea evitando particiones innecesarias y el cálculo de la solución QP en cada uno de ellos.

En [Wan03] se idea un algoritmo para obtener un conjunto de leyes de control explícitas para el MPC que corresponden a una secuencia de elipsoides asintóticamente invariantes y estables. Dichos elipsoides se definen en el espacio de los estados de manera *off-line*, utilizando la teoría de LMIs (*Linear Matrix Inequalities*). Esta solución es de bajo coste computacional y garantiza robustez ante incertidumbres en el modelo para el lazo cerrado.

En [Roj03] se presenta un método basado en la descomposición en valores singulares del hessiano de la función de costo cuadrática que se pretende optimizar, para mostrar las componentes de las variables de decisión en el problema de optimización que tienen mayor influencia en la función de costo. Se obtiene así una solución subóptima para el CMPC que asegura la no violación de las ligaduras.

La estabilidad de las técnicas para solucionar este problema queda establecida en [Hea05] si se asegura la estabilidad de la planta en lazo abierto, la solución nula está incluida en las soluciones del problema de optimización QP asociado y es factible y las entradas tienen un peso alto en la función de costo definida.

4.2.7.5 Problema de optimización de tiempo infinito con restricciones

Este problema surge al considerar $N \rightarrow \infty$ en la expresión (4-42), con lo que la formulación del problema queda:

$$\min_{u(k), u(k+1), \dots} \left\{ \sum_{i=0}^{\infty} (\mathbf{x}(k+i|k))^T \mathbf{Q} \mathbf{x}(k+i|k) + u(k+i)^T \mathbf{R} u(k+i) \right\}$$

sujeto a:

$$\begin{aligned} \mathbf{x}(k+i|k) &\in \mathfrak{X}, \quad i > 0 \\ u(k+i) &\in \mathfrak{U}, \quad i \geq 0 \\ \mathbf{x}(k+1) &= \mathbf{A} \mathbf{x}(k) + \mathbf{B} u(k) \end{aligned}$$

(4-46)

La solución a este problema se denomina CLQR (*Constrained LQR*). Muchos autores han tratado de solucionar este problema de forma eficiente. Por ejemplo, en [Bem02] se establece un método basado en el cómputo *off-line* de la ley de control mediante la resolución de un problema multiparamétrico.

Es en el trabajo de Grieder *et al* [Gri03] donde se da una solución implementable y computacionalmente eficiente al mismo. Dicho algoritmo consiste en el cálculo del mínimo horizonte finito N_m para que el problema CLQR de tiempo finito sea igual al CLQR de tiempo infinito. El bajo coste computacional de la misma se basa en combinar el uso de una solución PWA con la solución al problema QP finito asociado al CLQR con horizonte igual a N_m .

La estabilidad de este algoritmo se analiza en [Gri04] mediante una función cuadrática de Lyapunov obtenida por LMI y garantizada por condiciones suficientes de existencia.

4.3 Control predictivo con interpolación

En la pretensión de conseguir algoritmos de control predictivo de fácil implementación, computacionalmente eficaces y que dieran solución a los problemas de factibilidad que presentan otros tipos de controladores MPC, comenzaron a aparecer ([Kou98],[Ros98]) soluciones que se basan en interpolaciones entre soluciones óptimas y soluciones factibles; por ejemplo, entre la solución LQ y una entrada estabilizante que presenta siempre garantía de factibilidad (en adelante, trayectoria de “nivel medio” o ML).

En la figura 4-3 se muestra un ejemplo de interpolación entre la solución LQ y una solución perteneciente al conjunto de soluciones factibles ML. Este algoritmo presentará, por tanto, un único grado de libertad, que es la variable de interpolación entre ambas soluciones. El resultado de la interpolación dará la solución óptima dentro del conjunto de las soluciones factibles. Si en el instante inicial existe una solución factible, se puede demostrar que la optimización sobre esta variable garantiza una solución factible, la cual convergerá hacia la solución LQ tan pronto como ésta se vuelva factible.

La desventaja de estas soluciones es que no garantizan la convergencia de la función de costo, por lo que, eventualmente, el sistema puede resultar inestable.

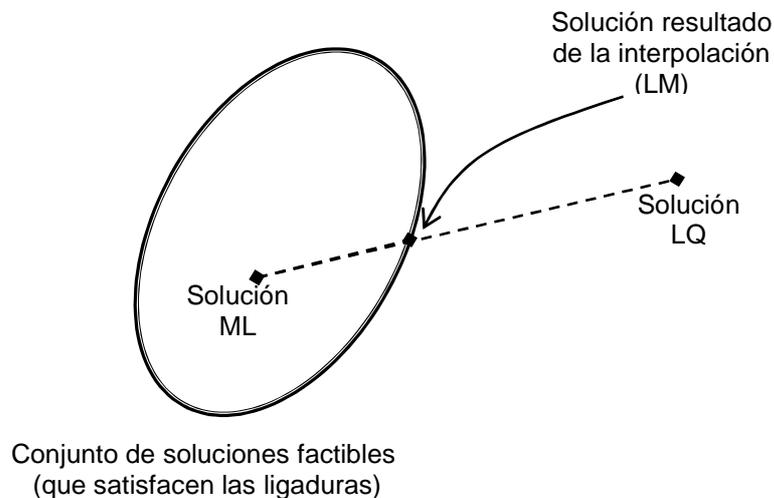


Figura 4-3: Esquema de interpolación entre la solución óptima del problema sin ligaduras y una solución factible de la clase de soluciones realizables denominadas de nivel medio (ML).

Para resolver este problema, otra estrategia de interpolación ([Ros98]) propone una interpolación entre la solución LQ y la “cola”. Esta segunda solución consiste en la trayectoria de entradas calculadas en el instante previo, salvando la correspondiente a ese instante (la cual fue aplicada al sistema). Con esta interpolación se garantiza la factibilidad, la convergencia de la función de costo y la convergencia hacia la solución óptima tan pronto como ésta se vuelva factible.

Sin embargo, presenta problemas de robustez cuando hay incertidumbres en el modelo o en presencia de perturbaciones, ya que las mismas pueden llevar a la cola fuera de la zona de factibilidad, por lo que conduce a una solución no factible que puede causar inestabilidad en el sistema en lazo cerrado.

En [Men00] este algoritmo queda desarrollado teniendo en cuenta la solución ML y realizando una interpolación entre las tres soluciones: LQ, ML y la cola (en adelante esta trayectoria se denominará LMT: LQ+ML+Tail). En este caso, si las perturbaciones llevasen a la cola fuera de la zona de factibilidad, es menos probable que la solución ML también caiga fuera de esa zona, por lo que la interpolación resultará siempre en una solución factible. El proceso de optimización da lugar a un problema de programación cuadrática de dos dimensiones. Este algoritmo presenta tres características: convergencia, optimalidad y factibilidad, además de presentar robustez frente a imprecisiones en el modelado y perturbaciones.

En esta tesis se implementa este algoritmo en un robot manipulador. El objetivo es que cada una de las articulaciones recorra unas trayectorias determinadas. Para ello, se inserta el sistema en un esquema de perturbación adaptativa, donde los comandos que se deben aplicar a la planta vienen dados por la resolución de la dinámica inversa. Las correcciones a dichos comandos vienen dadas en un primer caso por el algoritmo óptimo LQ, en un segundo caso por el algoritmo LM y, finalmente, por el algoritmo LMT. La eficacia de este último algoritmo será notable cuando las perturbaciones al sistema sean importantes. En ese caso, la interpolación entre las tres soluciones hará que el sistema tienda a alcanzar rápidamente las trayectorias predeterminadas para cada una de las articulaciones, sin que el sistema viole las ligaduras impuestas a las entradas.

4.3.1 Control LQ o lineal cuadrático.

Como ya se ha comentado, la solución LQ es la solución al problema de optimización sin ligaduras, obtenida mediante la resolución de la ecuación de Riccati.

Así, dado el problema de minimizar la función de costo definida en (4-29),

$$J = \sum_{i=1}^N \left(\|r(k+i) - y(k+i|k)\|_2 \right)^2 + \eta \sum_{i=0}^{M-1} \left(\|\Delta u(k+i)\|_2 \right)^2 \quad (4-29)$$

una vez obtenidas las predicciones mediante el modelo del sistema (y, tal vez, el modelo de perturbaciones), para un sistema cuya evolución temporal viene dada por la ecuación expresada en el espacio de los estados (4-2), y cuya ley de control es de la forma:

$$u(k) = -\mathbf{K}\mathbf{x}(k) \quad (4-47)$$

la solución viene dada por el valor de la ganancia \mathbf{K} para la cual la secuencia de entradas $u(k)$ minimiza la función de costo (4-29). Esta solución se designará como \mathbf{K}_{LQ} .

4.3.2 Control ML o de nivel medio

Dado que la solución óptima LQ es, por lo general, una estrategia que lleva al sistema a una solución no factible, se debe recurrir a una solución subóptima del problema de optimización.

Una primera opción es buscar una solución que siempre presente garantía de factibilidad. Esta solución se denomina de nivel medio (ML, "Mean Level"). Para conseguir esta solución se recurre a solucionar el problema cuadrático lineal para el sistema (4-2), con la ley de control (4-47), donde la función de costo (4-29) está definida de tal forma que el peso en las entradas al sistema es mucho mayor que en los estados. Así, basta con elegir $\eta \gg 1$ en dicha ecuación.

Por lo general, la solución dada por este controlador será una solución tal que la trayectoria recorrida por el sistema se alejará en mayor o menor medida de la trayectoria de referencia, pero asegurando siempre la no violación de las ligaduras impuestas a las entradas.

4.3.3 Interpolación LQ+ML

La solución anterior, en sí misma, no es la solución deseable para el sistema. En los trabajos de Kouvaritakis *et al* [Kou97],[Kou98], Rossiter *et al* [Ros96], [Ros98] y Méndez [Men00], esta solución se calcula para el caso en que las perturbaciones que afectan al sistema lleven al mismo a la región no factible. En ese caso, la solución óptima resulta no factible, por lo que, eventualmente, el algoritmo de control aplicará una interpolación entre la solución óptima LQ y la solución ML. La solución resultante es subóptima y se denomina LM (LQ+ML).

Una vez obtenidas ambas soluciones, es decir, la solución óptima LQ, dada por la secuencia de entradas u_{LQ} , y la solución factible ML, dada por u_{ML} , el problema se reduce a una interpolación entre ambas de la forma:

$$u_{LM} = (1 - \alpha)u_{LQ} + \alpha u_{ML} \quad (4-48)$$

Se trata, por tanto, de un problema con un único grado de libertad: el parámetro de interpolación $\alpha \in \mathfrak{R}$, $0 \leq \alpha \leq 1$, cuyos valores extremos corresponden a la solución LQ ($\alpha = 0$) y a la solución ML ($\alpha = 1$). Para la realización del mismo, se supone una división inicial del estado de la forma:

$$\mathbf{x}(0) = \mathbf{w}(0) + \mathbf{z}(0) \quad (4-49)$$

en donde:

$$\begin{aligned} \mathbf{w}(0) &= (1 - \alpha)\mathbf{x}(0) \\ \mathbf{z}(0) &= \alpha\mathbf{x}(0) \end{aligned} \quad (4-50)$$

Las predicciones se harán utilizando la ecuación:

$$u_{LM}(k) = -\mathbf{K}_{LQ}\boldsymbol{\Phi}_{LQ}^k \mathbf{w}(0) - \mathbf{K}_{ML}\boldsymbol{\Phi}_{ML}^k \mathbf{z}(0) \quad (4-51)$$

donde \mathbf{K}_{LQ} es la ganancia obtenida del problema de control óptimo LQ sin ligaduras y \mathbf{K}_{ML} es la ganancia correspondiente al problema de nivel medio ML. Además:

$$\begin{aligned} \boldsymbol{\Phi}_{LQ} &= \mathbf{A} - \mathbf{B}\mathbf{K}_{LQ} \\ \boldsymbol{\Phi}_{ML} &= \mathbf{A} - \mathbf{B}\mathbf{K}_{ML} \end{aligned} \quad (4-52)$$

siendo \mathbf{A} y \mathbf{B} las matrices de estado correspondientes al sistema en estudio (4-2).

El problema de control predictivo se reduce, pues, al cálculo del parámetro α en cada instante de la trayectoria, de forma que la solución resultante u_{LM} sea factible (para lo cual se utilizarán las predicciones hasta el horizonte de predicción M) y, a su vez, minimice la función de costo (4-29). Para ello, se resolverá en cada instante un problema de programación lineal, cuya implementación no conlleva un coste computacional elevado.

Esta solución será subóptima, dada por la solución más cercana a la solución LQ dentro del conjunto de las soluciones factibles, tal y como se observa en la figura 4-3.

4.3.4 Obtención de la cola en el control predictivo

El controlador anterior resuelve el problema de factibilidad, sin embargo, tiene como desventaja que no asegura la convergencia de la función de costo, por lo que eventualmente puede conducir a inestabilidad del lazo cerrado. Para solventar este caso se hace uso de otro tipo de interpolación, en la que se involucra a la solución óptima LQ del problema sin ligaduras y a la extensión al instante actual de la solución calculada en el instante anterior.

Esta última solución recibe el nombre de cola (*tail*) y está formada por las entradas calculadas en el instante previo que no fueron aplicadas al sistema (todas las entradas hasta el horizonte de control, salvo la primera). Para ver esto más claramente, considérese en el instante actual k la secuencia de entradas dadas por el algoritmo de optimización:

$$\hat{\mathbf{u}}(k) = [u(k|k) \ u(k+1|k) \ \dots \ u(k+N|k)] \quad (4-53)$$

La primera de las entradas se aplica en el instante actual, quedando el resto de entradas definidas como la cola en el instante actual:

$$\mathbf{u}_r(k) = [u(k+1|k) \ \dots \ u(k+N|k)] \quad (4-54)$$

Esta solución puede ser utilizada en el instante posterior $k+1$. Combinar esta solución con la solución óptima LQ da como resultado un problema de control predictivo cuya resolución se lleva a cabo nuevamente mediante un algoritmo de programación lineal, con un único parámetro de interpolación.

Con esta solución se garantiza la convergencia de la función de costo. Sin embargo, el mayor problema reside en la presencia de perturbaciones o incertidumbres en el sistema. Si la solución LQ es no factible, la cola estará muy cerca de la región no factible, por lo que la solución resultante resultará factible. Pero las perturbaciones pueden llevar la cola fuera de la zona de factibilidad, por lo que la interpolación dará como resultado una solución no factible.

4.3.5 Interpolación LQ+ML+Cola

Debido a lo anteriormente expuesto, y buscando un algoritmo que reúna las condiciones de factibilidad y de convergencia de la función de costo, aún en presencia de perturbaciones o incertidumbres en el modelo, se propone ([Ros98b] en su formulación en el espacio de entradas-salidas, [Men00] en el espacio de los estados) un algoritmo que inserte las dos soluciones antes expuestas: la solución de nivel medio ML que asegure la factibilidad de la solución resultante y la cola, que asegura la convergencia de la función de costo.

La entrada resultante del proceso de interpolación quedará formada por la siguiente expresión:

$$u_{LMT} = (1 - \alpha - \beta)u_{LQ} + \alpha u_t + \beta u_{ML} \quad (4-55)$$

donde los parámetros de interpolación α y β cumplen que:

$$\begin{aligned} 0 &\leq \alpha \leq 1 \\ 0 &\leq \beta \leq 1 \\ 0 &\leq \alpha + \beta \leq 1 \end{aligned} \quad (4-56)$$

Para resolver el problema de optimización se debe escoger la siguiente división del estado inicial:

$$\mathbf{x}(0) = \mathbf{w}(0) + \mathbf{z}(0) + \mathbf{v}(0) \quad (4-57)$$

De forma que las predicciones para la entrada vendrán dadas por:

$$u_{LMT}(k) = -\mathbf{K}_{LQ}\boldsymbol{\Phi}_{LQ}^k \mathbf{w}(0) - \mathbf{K}_{ML}\boldsymbol{\Phi}_{ML}^k (\mathbf{z}(0) + \mathbf{v}(0)) \quad (4-58)$$

donde \mathbf{K}_{LQ} es la ganancia obtenida de resolver el problema sin ligaduras LQ, \mathbf{K}_{ML} la ganancia obtenida de resolver el problema de nivel medio y $\boldsymbol{\Phi}_{LQ}$ y $\boldsymbol{\Phi}_{ML}$ definidas según (4-52). Resulta así que la división del estado inicial será:

$$\begin{aligned} \mathbf{w}(0) &= (1 - \alpha - \beta)\mathbf{x}(0) + \beta\boldsymbol{\Phi}_{LQ}\mathbf{w}(-1) \\ \mathbf{z}(0) &= \alpha\mathbf{x}(0) \\ \mathbf{v}(0) &= \beta(\mathbf{x}(0) - \boldsymbol{\Phi}_{LQ}\mathbf{w}(-1)) \end{aligned} \quad (4-59)$$

Esta solución incluye las soluciones anteriores si se escogen los parámetros de interpolación con los valores adecuados. Así, si $\alpha = 0$ y $\beta = 0$, se recupera la solución LQ,

$$u_{LMT}(k) = -\mathbf{K}_{LQ}\boldsymbol{\Phi}_{LQ}^k \mathbf{x}(0) \quad (4-60)$$

si $\alpha = 1$ y $\beta = 0$, se obtiene la solución ML,

$$u_{LMT}(k) = -\mathbf{K}_{ML}\boldsymbol{\Phi}_{ML}^k \mathbf{x}(0) \quad (4-61)$$

y, si $\alpha = 0$ y $\beta = 1$, se tiene la solución de la cola [Men00]:

$$u_{LMT}(k) = -\mathbf{K}_{LQ}\boldsymbol{\Phi}_{LQ}^{k+1} \mathbf{w}(-1) - \mathbf{K}_{ML}\boldsymbol{\Phi}_{ML}^{k+1} (\mathbf{z}(-1) + \mathbf{v}(-1)) \quad (4-62)$$

4.4 Principales dificultades del MPC

A la vista de lo comentado en apartados anteriores, la principal virtud de la formulación del MPC es su facilidad de implementación debido a que está basado en la resolución de un problema de optimización, que en la mayoría de los casos deriva en un problema QP. En la formulación de dicho problema, la inclusión de las ligaduras es bastante sencilla, lo cual es una ventaja bastante evidente respecto a otros esquemas de control.

La resolución del problema de optimización se realiza en cada instante de la trayectoria, por lo que se deben proveer algoritmos de bajo coste computacional.

En el caso de la robótica de manipulación que nos compete, estos dos puntos son de especial relevancia. En primer lugar, dadas las limitaciones que aparecen en la electrónica asociada a los actuadores de cada una de las articulaciones, las ligaduras en las entradas deben ser tenidas en cuenta en el algoritmo de control, con el fin de evitar las situaciones de saturación que lleven al lazo cerrado a una posible situación de inestabilidad. En segundo lugar, la

rápida dinámica de los manipuladores exige la utilización de tiempos de muestreo cortos, por lo que la eficiencia computacional se convierte en una exigencia para la aplicación de estas técnicas a los manipuladores.

4.4.1 Factibilidad de la solución en MPC

El primer problema que se debe abordar por parte de las técnicas de control predictivo es la factibilidad de las soluciones aportadas por el algoritmo de optimización del MPC. La propiedad de factibilidad es la que asegura que la solución al problema de control con ligaduras existe, esto es, que satisface dichas ligaduras.

En el caso de control predictivo, la ley de control viene dada por el primer comando de la serie de entradas resultantes de la resolución de un problema de minimización de una función de costo, generalmente sujeta a las restricciones impuestas por las ligaduras que afectan al sistema. La solución es factible si asegura la minimización de la función de costo sin que se violen las ligaduras.

En el problema original de control predictivo, se plantea la resolución del problema de control predictivo con horizonte de predicción infinito y sin ligaduras [Mor99]. Dicha solución es el controlador lineal cuadrático LQR (a partir de ahora, LQ), el cual fue estudiado en profundidad desde la aparición de los primeros controladores predictivos. Esta solución está basada en la aplicación de una ley de control de realimentación de variables de estado, utilizando unas ganancias calculadas a partir de la resolución de la ecuación de Riccati.

En el caso de considerar ligaduras, el problema de optimización resultante es de dimensión infinita, razón por la cual su aplicación práctica no es posible. Para obtener una solución realizable, se han ideado una serie de estrategias ([Lee97],[Sco98]) basadas en la reducción de la dimensionalidad del problema, lo cual conlleva a una solución subóptima del problema original.

Dicha solución consiste en resolver el algoritmo de optimización sobre un conjunto finito dentro de la clase de soluciones estables, lo cual conlleva a considerar las n_c primeras acciones de control como resultado del algoritmo de optimización, para luego aplicar la solución lineal y cuadrática LQ ([Kou98]). Esta solución también es conocida como estrategia de modo dual, con lo cual el problema de optimización se reduce a un problema de dimensionalidad n_c .

Para asegurar que la solución es factible, se necesitan valores de N elevados, con lo cual la carga computacional del algoritmo es muy grande. Este hecho limita la realización práctica de dicho controlador, sobre todo en plantas de dinámica muy rápida, como es el caso que nos compete de los robots manipuladores.

Otra situación que se debe tener en cuenta sobre la factibilidad de la solución es la presencia de perturbaciones en el sistema. Dado que

generalmente las restricciones de saturación del comando (4-24) no pueden ser excedidas, puede ocurrir que la perturbación al sistema en un instante determinado de tiempo lleve a la solución del problema de optimización hacia la región no factible. Se han propuesto diversas soluciones para este problema, como la suavización de las restricciones más fuertes impuestas a un sistema por introducción de un término de penalización en la función de costo ([Zhe95], [Qin96]).

4.4.2 Coste computacional

La segunda dificultad que se debe abordar es el coste computacional de las técnicas predictivas. Dicha complejidad viene dada por la resolución del algoritmo de optimización asociado a la obtención de la ley de control.

En este apartado veremos las técnicas habituales empleadas para la resolución del problema de optimización y, a continuación, diversas técnicas que mejoran la eficiencia computacional de las mismas.

4.4.2.1 Algoritmos de optimización en MPC

Los métodos que se utilizan para resolver el problema de optimización son ([Ros03]) el método de conjuntos activos (*Active Set Method*), el método de punto interior (*Interior Point Method*), el método de programación cuadrática multiparamétrica (*MPQP, MultiParametric Quadratic Programming*) y otros métodos basados en aproximaciones subóptimas del problema completo. Dichos métodos se describen en los subapartados siguientes.

A) Método de conjuntos activos

El método habitual para solucionar el problema QP es el de conjuntos activos. El objetivo de este método es separar las restricciones activas (las restricciones que se violan con la solución óptima sin restricciones) de las inactivas respecto a la solución óptima con restricciones. Entonces, se replantea el problema de optimización ignorando las restricciones inactivas.

Si la solución así calculada cumple las restricciones inactivas, entonces es factible y se testea si es la óptima del problema global mediante el uso de los multiplicadores de Lagrange asociados (que deben ser positivos). Si no es así, entonces no se tiene la solución óptima y se debe cambiar la separación realizada para las restricciones activas e inactivas.

B) Método de punto interior

El algoritmo se basa en replantear el problema de optimización de forma que las ligaduras pasen a formar parte de la función de costo, con un peso tal que se aproxime a infinito (típicamente eligiendo una función logarítmica) a medida que la restricción pase a ser activa. La solución óptima así calculada siempre cumplirá las restricciones.

La ventaja de este segundo método es que la convergencia del método está asegurada y, además, de forma más rápida que el anterior.

C) Método de programación cuadrática multiparamétrica

Este método, de reciente aplicación, plantea la solución del problema QP original por un problema multiparamétrico, en el cual se tienen en cuenta sólo las restricciones activas. Dado que corresponde a un problema de programación cuadrática con ligaduras de igualdad, la solución tiene una forma fija. Por tanto, la ley de control así calculada será la óptima dentro de un conjunto limitado de estados (cuando los multiplicadores de Lagrange son positivos).

Por tanto, el algoritmo se basa en el cálculo de de las regiones para las cuales la ley de control óptima es factible. Además, se deben reducir las regiones de forma que no se solapen. Dado que la forma de estas soluciones es fija, se puede hacer el cálculo *off-line*. El único cálculo *on-line* es identificar la región del espacio de estados en la que se encuentra el sistema e implementar la ley asociada a dicha región.

Por tanto, la principal ventaja del algoritmo es su reducido coste computacional, ya que la mayoría del cálculo se hace *off-line*. En las referencias dadas en el apartado 4.2.7.4 se pueden observar diversas técnicas que proveen soluciones computacionalmente eficientes basadas en este método. En particular, resultan de interés las técnicas que buscan una solución PWA (*PieceWise Affine*) ([Gri03], [Ton03], [Gri04], [Ker04]), con la cual se da una solución *off-line* al problema de optimización QP convertido en un problema multiparamétrico, una vez realizada la partición del espacio de los parámetros ([Gri03]). Otros autores proveen algoritmos distintos para la obtención de una solución de bajo coste computacional, como puede verse en [Roj03], [Wan03] y [Que04].

4.4.2.2 Algoritmos de optimización en MPC de bajo coste computacional

A continuación se presentan tres técnicas de bajo coste computacional para resolver el problema de optimización que se plantea en la estrategia de control MPC.

Los algoritmos de optimización son computacionalmente ineficientes cuando se deben realizar, en cada instante, las comparaciones entre las predicciones del modelo y las restricciones impuestas, para la verificación de que la solución obtenida del algoritmo es factible.

Esta comprobación debe realizarse para evitar que las predicciones no concuerden con el comportamiento del sistema y así favorecer el rendimiento del lazo cerrado, así como para evitar la aparición de inestabilidad o mal comportamiento del lazo cerrado por violación de las restricciones.

Sin embargo, la verificación de que la solución es factible se puede evitar en algunos casos. En el último algoritmo antes mencionado (ver 0) se

introduce el concepto de regiones en las cuales se calcula *off-line* una solución que resuelve el problema de optimización sin que el estado del sistema abandone dicha región. A esta región se le suele llamar conjunto invariante.

Las técnicas de bajo coste computacional que aquí se mencionan se basan en la propiedad de invariabilidad de estos conjuntos. La primera de ellas es denominada algoritmo de anidación (*nested*, [Tan92]), en el cual el único cálculo a realizar *on-line* es la pertenencia a una serie de conjuntos invariantes que son calculados de manera *off-line*, estando definida en cada uno de ellos la ley de control adecuada para los estados que abarcan dichos conjuntos.

Estos conjuntos invariantes quedan definidos como:

$$S_i = \{ \mathbf{x} \mid \mathbf{x}^T \mathbf{Y} \mathbf{x} \leq 1 \} \quad ; \quad \mathbf{Y} > 0 \quad (4-63)$$

donde se cumple que:

$$(\mathbf{A} - \mathbf{B}\mathbf{K}_i)^T \mathbf{Y} (\mathbf{A} - \mathbf{B}\mathbf{K}_i) \leq \mathbf{Y} \quad (4-64)$$

e \mathbf{Y} es lo suficientemente pequeño como para que la solución de control dada por la ganancia \mathbf{K}_i en la ley de control $u = -\mathbf{K}\mathbf{x}$ no viole las restricciones impuestas al estado y a la entrada. \mathbf{A} y \mathbf{B} son las matrices del modelo del sistema que se va a controlar, expresado en el espacio de los estados:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \\ y(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \quad (4-65)$$

Se dice, en ese caso, que \mathbf{K}_i es la ganancia asociada al conjunto invariante S_i . Los conjuntos invariantes así definidos están contenidos unos dentro de otros, a medida que las ganancias se van haciendo menores (por lo que las restricciones se cumplen cada vez con mayor probabilidad). Dichos conjuntos están limitados por el conjunto admisible máximo (MAS, *Maximal Admissible Set*), o conjunto en el cual se satisfacen todas las restricciones impuestas al estado y a la entrada con la ley más conservativa posible.

Otro algoritmo que rebaja el costo computacional es el algoritmo MPC eficiente (EMPC, *Efficient MPC*) ([Kou00]). En este algoritmo se emplea la idea de definición de conjuntos invariantes pero para la ley de control $u(k) = -\mathbf{K}\mathbf{x}(k) + c(k)$, donde $c(k)$ son parámetros que aumentan el volumen de los conjuntos invariantes definidos por (4-63) para aumentar la región factible y que se resuelven en el algoritmo de interpolación mediante su inclusión en el estado del sistema, para formar un estado aumentado.

Este algoritmo también disminuye el costo computacional del MPC clásico, ya que la mayoría del cómputo se realiza *off-line*, dejando como único cómputo *on-line* la verificación de pertenencia del estado a los conjuntos invariantes así como la minimización de una función de costo, más sencilla que

la original del MPC, dependiente de los parámetros $c(k)$. Aunque este método mejora la factibilidad de las soluciones, el costo computacional del mismo es mayor respecto a los algoritmos de interpolación de un grado de libertad que se mencionan a continuación. Sin embargo, es semejante al costo del algoritmo de interpolación de dos grados de libertad LMT.

Otros algoritmos de control predictivo que rebajan el coste computacional del MPC clásico son los algoritmos de control con interpolación ([Ros04a], [Ros04b]), previamente explicados en el apartado 4.3. El problema de optimización se reduce a un problema de programación lineal, cuando se realiza la interpolación entre dos soluciones o a un problema de programación cuadrática, cuando se realiza entre las tres soluciones (LQ, *Mean Level* y la cola).

Los algoritmos de interpolación mejoran el comportamiento del algoritmo de anidación en que la ley de control es menos conservativa cuando el estado no está cerca de los bordes del conjunto invariante, por lo que el comportamiento del lazo cerrado mejora considerablemente. Además, el cómputo de los problemas de programación lineal o cuadrática definidos en los algoritmos de interpolación es más fácil que el cómputo de la pertenencia a los conjuntos invariantes del algoritmo de anidación. Un ejemplo de esto se puede ver en [Ros05], donde se introducen y comparan distintas técnicas de interpolación aplicadas en sistemas LPV (*linear parameter varying*).

En esta tesis, dado que el objetivo es conseguir un algoritmo eficiente de bajo coste computacional para la aplicación en robots manipuladores, se propone la aplicación de técnicas predictivas basadas en interpolación. Con esta técnica se resuelve el problema de factibilidad. Tal y como quedó explicado en el apartado 4.3, el problema de optimización que se debe resolver en el control predictivo con interpolación incorpora las ligaduras en las entradas y en el estado del manipulador, de forma que la solución obtenida siempre es factible. Por ejemplo, el algoritmo LM aplica la solución LQ siempre que sea factible o, en su caso, una solución subóptima factible tan cercana como sea posible a la solución LQ. Por otro lado, las técnicas de interpolación aseguran la realización práctica de la solución debido a la baja carga computacional del algoritmo. Como se ha explicado en este apartado, el coste computacional de estas técnicas reduce el coste asociado tanto al MPC original como a los otros algoritmos de bajo coste computacional explicados anteriormente.

La principal dificultad a la hora de aplicar estos algoritmos es la necesidad de proveer un modelo lineal para realizar las predicciones. La aplicación del modelo completo no lineal del manipulador conlleva una evaluación compleja de las restricciones presentes en el sistema. Por tanto, se realizará una linealización de las ecuaciones para preservar el bajo coste computacional del algoritmo, a costa de aumentar dicho coste por la necesidad de obtener una linealización en cada instante de la trayectoria.

En el siguiente capítulo se explican los fundamentos del método de control empleado para su aplicación en brazos manipuladores y se exponen resultados de control del PUMA-560 con dicho controlador.

Control de perturbación adaptativa basado en interpolación (IAPC)

Las principales dificultades a la hora de aplicar técnicas de control predictivas, para la inclusión de las ligaduras del sistema, son el costo computacional del algoritmo y la existencia de una solución factible. En este capítulo se introduce un nuevo esquema de control, el controlador IAPC, que aporta una solución eficaz a estos problemas. Este controlador consiste en la corrección de las desviaciones respecto a la trayectoria nominal o de referencia, mediante un controlador predictivo lineal basado en interpolación, que asegura la factibilidad de la solución y cuyo coste computacional es bajo. Dadas las limitaciones existentes con el esquema "feedback linearisation" para obtener una definición lineal de las ligaduras, se escoge el método de linealización exacta para obtener el modelo de predicción. En el presente capítulo, además, se define una mejora sobre este controlador para el caso de incertidumbres en el modelo o perturbaciones al sistema. Para ello, se añade la acción robusta autoadaptativa vista en los capítulos 2 y 3. El esquema de control resultante, denominado RIAPC, presenta mejores prestaciones que la estrategia IAPC en el caso general del manipulador con restricciones y en presencia de incertidumbres y/o perturbaciones. Los resultados que se presentan muestran las excelentes prestaciones de estas técnicas para evitar que el sistema viole las restricciones impuestas al mismo.

5.1 Introducción

Uno de los principales problemas que se presenta en el campo de la robótica de manipulación, dada la alta exigencia en cuanto a frecuencia de muestreo, es la necesidad de algoritmos computacionalmente eficientes que resuelvan la tarea de control de una manera rápida.

Por otra parte, los manipuladores están sujetos a varias restricciones. Las más evidentes son las limitaciones físicas impuestas al movimiento de cada una de las articulaciones. Pero también hay restricciones en las entradas máximas que se le pueden aportar a los actuadores, así como en la máxima

variación de las mismas, debido a las limitaciones que impone la electrónica asociada a dichos actuadores. Evitar las situaciones de saturación de las entradas es fundamental en este caso, ya que así se evita la aparición de problemas de estabilidad en el lazo cerrado de control.

A modo de ejemplo, se expone el caso de un manipulador de dos articulaciones de revolución bajo la acción de un controlador predictivo tipo LQ sin ligaduras, como el explicado en el apartado 4.3.1. La acción de control queda definida por una entrada obtenida por un controlador de dinámica inversa y las correcciones a la misma aportadas por el controlador LQ. La trayectoria definida para el manipulador es tal que las restricciones en el estado nunca se violan.

El manipulador está afectado por una perturbación constante aditiva en la salida, de forma que la trayectoria final se separa de la trayectoria de referencia. Tal y como se observa en la figura 5-1, la perturbación lleva el sistema al máximo valor en la entrada en ambas articulaciones, lo que conlleva la inestabilidad del lazo cerrado, no pudiendo nunca recuperar la trayectoria de referencia.

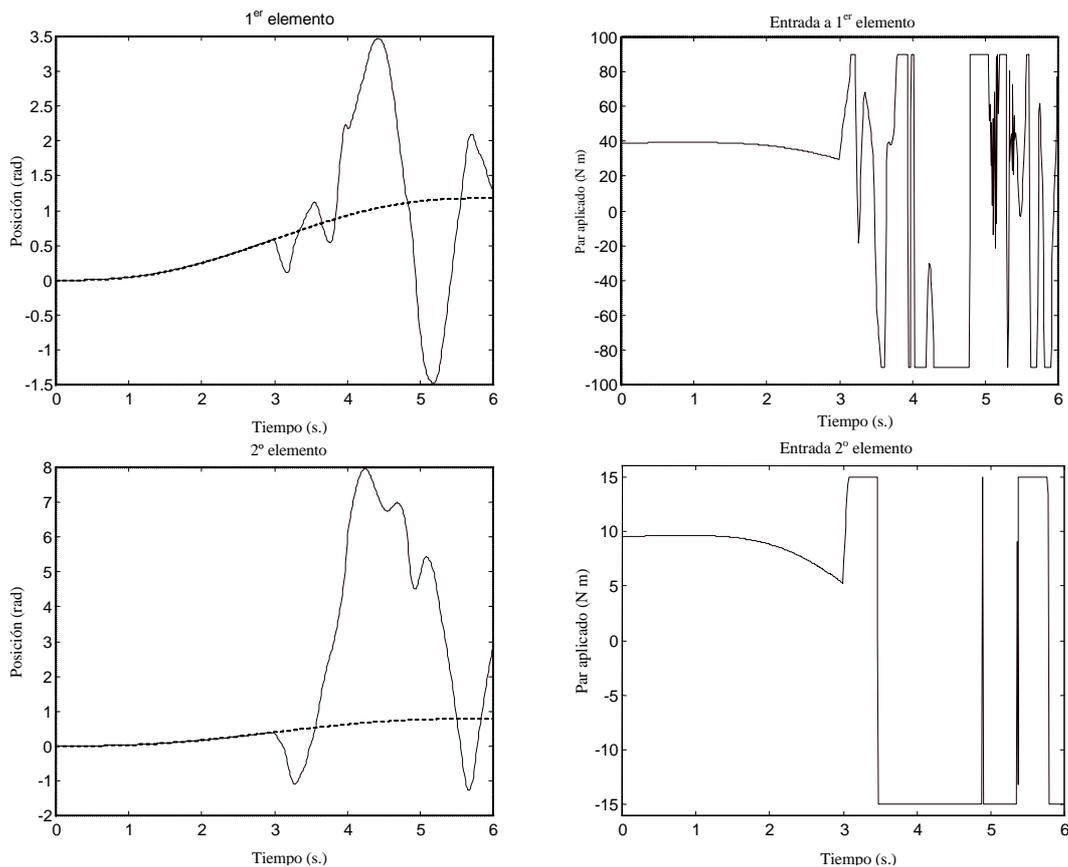


Figura 5-1: Evolución de un robot manipulador de dos articulaciones de revolución bajo la acción de una acción de control LQ. En la columna de la izquierda se observa la posición angular de ambas articulaciones (en línea sólida) frente a la de referencia (en línea discontinua). En la columna de la derecha se representan las entradas aplicadas en cada caso.

Estas dos razones (conseguir algoritmo computacionalmente eficiente y que tenga en cuenta las restricciones del manipulador), junto a las ventajas de los algoritmos predictivos MPC y, en particular, los algoritmos basados en interpolación, nos lleva a plantear un algoritmo predictivo con interpolación para el control de manipuladores.

El algoritmo presentado recibe el nombre de Controlador de Perturbación Adaptativa basado en Interpolación, o IAPC (*Interpolation based Adaptive Perturbation Control*). Con este controlador las ligaduras a las que está sujeto el manipulador pueden incorporarse de forma natural a la ley de control, a la vez que se asegura que el sistema siempre actúe en la zona factible.

5.2 Esquema de control IAPC

En la formulación del controlador predictivo MPC original, se supone un modelo lineal para el sistema de forma que en la resolución del problema de optimización se emplee este modelo para realizar las predicciones.

En el caso del robot manipulador, nos encontramos ante un sistema cuya dinámica se expresa en un modelo acoplado altamente no lineal. Por tanto, resulta totalmente inapropiado para la aplicación tanto de estas técnicas como de las técnicas basadas en interpolación.

A continuación se describen dos posibles soluciones para la obtención del modelo linealizado del sistema, la primera de las cuales no se aplica debido a la complejidad para la incorporación de las restricciones en la ley de control predictiva. La segunda solución incorpora una linealización instantánea.

5.2.1 Obtención del modelo mediante *feedback linearisation*

Una primera solución es tomar el esquema de control de linealización por realimentación (*feedback linearisation*) planteado en capítulos anteriores. Mediante este esquema se compensa la parte no lineal del modelo mediante una entrada no lineal calculada con las medidas actuales del estado del manipulador.

De esta forma se tiene un modelo lineal y desacoplado para el manipulador, consistente en un modelo de doble integrador para cada una de las articulaciones. Con este modelo se realizarán las predicciones del sistema para el cálculo de la acción de control.

Sin embargo, esta alternativa presenta un gran inconveniente. Dado que las entradas al robot están calculadas a través de una entrada linealizante que incluye los términos no lineales que se quieren compensar, existe una relación no lineal entre la entrada que se aplica al modelo linealizado y la entrada al robot real. Por tanto, las ligaduras en las entradas al manipulador, que tienen

una expresión sencilla para dicho modelo no lineal, presentan una expresión no lineal para el modelo linealizado y, lo que es más importante, dependientes del estado actual del manipulador.

Para ilustrar este hecho, recopilamos aquí las expresiones de la dinámica del manipulador y del esquema *feedback linearisation*. Las ecuaciones de la dinámica, evitando incluir la dependencia temporal de las variables, son:

$$\boldsymbol{\tau} = \mathbf{B}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{c}(\boldsymbol{\theta}) = \mathbf{B}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{n}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \quad (5-1)$$

La entrada de linealización al sistema es:

$$\boldsymbol{\tau} = \mathbf{B}(\boldsymbol{\theta})\mathbf{y} + \mathbf{n}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \quad (5-2)$$

Con lo que el sistema resultante respecto a la entrada \mathbf{y} es un sistema desacoplado de segundo orden que consiste en el doble integrador:

$$\ddot{\boldsymbol{\theta}} = \mathbf{y} \quad (5-3)$$

En términos del error $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}_d - \boldsymbol{\theta}$ y, expresando de la forma habitual el estado $\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{\boldsymbol{\theta}} & \dot{\tilde{\boldsymbol{\theta}}} \end{bmatrix}^T$, la ecuación de estado para el sistema linealizado queda:

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\mathbf{u} \quad (5-4)$$

donde

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} ; \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} ; \quad \mathbf{u} = \ddot{\boldsymbol{\theta}}_d - \mathbf{y} \quad (5-5)$$

siendo \mathbf{I} la matriz identidad y $\mathbf{0}$ una matriz nula de tamaños adecuados.

Para el caso del controlador LQ, se plantea la siguiente función de costo que debe ser optimizada:

$$J = \sum_{k=0}^{\infty} (\tilde{\mathbf{x}}^T \mathbf{V} \tilde{\mathbf{x}} + \mathbf{u}^T \mathbf{W} \mathbf{u}) \quad (5-6)$$

donde \mathbf{V} y \mathbf{W} son matrices que pesan el error de seguimiento y la entrada al sistema (5-4). Admitiendo una ley de control del tipo $\mathbf{u} = -\mathbf{K}\tilde{\mathbf{x}}$, el problema de optimización dará una solución \mathbf{K}_{LQ} que, aplicando (5-5), queda:

$$\mathbf{y} = \ddot{\boldsymbol{\theta}}_d + \mathbf{K}_{LQ} \tilde{\mathbf{x}} \quad (5-7)$$

Para el caso de considerar restricciones, no se puede aplicar este problema de dimensión infinita, por lo que se aplica una de las aproximaciones de bajo costo explicadas con anterioridad. En particular, si se aplica el esquema de interpolación de un grado de libertad LM, la ley de control es:

$$\mathbf{u} = (1 - \alpha)\mathbf{u}_{LQ} + \alpha\mathbf{u}_{ML} = -(1 - \alpha)\mathbf{K}_{LQ}\tilde{\mathbf{x}} - \alpha\mathbf{K}_{ML}\tilde{\mathbf{x}} \quad (5-8)$$

donde \mathbf{K}_{ML} es la solución del problema sin ligaduras obtenido de la minimización de (5-6), pero considerando $\mathbf{W} \gg \mathbf{V}$, y α es el parámetro de interpolación.

Para el cálculo de las predicciones se tiene:

$$\mathbf{u}_j = -(1 - \alpha)\mathbf{K}_{LQ}\boldsymbol{\varphi}_{LQ}^j\tilde{\mathbf{x}} - \alpha\mathbf{K}_{ML}\boldsymbol{\varphi}_{ML}^j\tilde{\mathbf{x}} \quad (5-9)$$

$$\tilde{\mathbf{x}}_j = (1 - \alpha)\boldsymbol{\varphi}_{LQ}^j\tilde{\mathbf{x}} + \alpha\boldsymbol{\varphi}_{ML}^j\tilde{\mathbf{x}} \quad (5-10)$$

con $j = 0, \dots, n_c$, siendo n_c el horizonte de control, $\boldsymbol{\varphi}_{LQ} = \mathbf{A} - \mathbf{B}\mathbf{K}_{LQ}$ y $\boldsymbol{\varphi}_{ML} = \mathbf{A} - \mathbf{B}\mathbf{K}_{ML}$.

Aplicando (5-5), las predicciones en la entrada (5-9) se convierten en:

$$\mathbf{y}_j = \ddot{\boldsymbol{\theta}}_d + (1 - \alpha)\mathbf{K}_{LQ}\boldsymbol{\varphi}_{LQ}^j\tilde{\mathbf{x}} + \alpha\mathbf{K}_{ML}\boldsymbol{\varphi}_{ML}^j\tilde{\mathbf{x}} \quad (5-11)$$

Luego, teniendo una restricción en el valor de la entrada de linealización y , se podrá formular el problema de optimización para el cálculo del parámetro de interpolación.

Sin embargo, para el cálculo de esa restricción hay que acudir a la expresión (5-2), para obtener:

$$\mathbf{y} = \mathbf{B}(\boldsymbol{\theta})^{-1}(\boldsymbol{\tau} - \mathbf{n}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})) \leq \|\mathbf{B}(\boldsymbol{\theta})^{-1}\|(\tau_M + \|\mathbf{n}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\|) = y_M \quad (5-12)$$

siendo τ_M el valor máximo de entrada al sistema no lineal.

Luego, el valor y_M necesario en la consideración de las ligaduras de la expresión (5-11) depende del valor que tomen las expresiones $\mathbf{B}(\boldsymbol{\theta})$ y $\mathbf{n}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ en los estados correspondientes a los instantes de tiempo en los que se realiza la predicción, los cuales se obtendrían de las predicciones (5-10).

Es por esta razón que el planteamiento con este esquema resulta complejo e inadecuado para su aplicación en la obtención del modelo lineal para el controlador predictivo.

5.2.2 Linealización exacta

Se opta pues por un esquema en el que las restricciones sean fácilmente expresables y que conserven su forma lineal. Para ello, se escoge el esquema de control que se observa en la figura 5-2. La acción de control viene dada por la suma de la entrada provista por un controlador de dinámica inversa (o de par calculado) y las correcciones hechas por un controlador predictivo lineal basado en interpolación.

Este esquema es conocido como control de perturbación adaptativa ([Lee85], [Ma85]), puesto que se basa en un esquema de control en el que las desviaciones de la respuesta nominal debida a las perturbaciones son corregidas por una acción de control basada en una identificación del sistema en cada instante (el resultado de identificación se adapta en cada instante al modelo del sistema perturbado).

Para calcular la acción de control predictiva, que incluirá las restricciones τ_M impuestas a la entrada del sistema no lineal así como las restricciones de posición y velocidad de las articulaciones, se debe contar con un modelo linealizado del sistema.

Este modelo se consigue mediante una linealización en torno al estado actual del manipulador, suponiendo que las desviaciones del sistema real respecto al modelo de referencia son pequeñas. Este esquema recibe el nombre de linealización exacta ([Ma85], [Sor94], [Lam07]), puesto que en cada instante se realiza una linealización del mismo. El modelo lineal resultante es utilizado para realizar las predicciones del controlador predictivo.

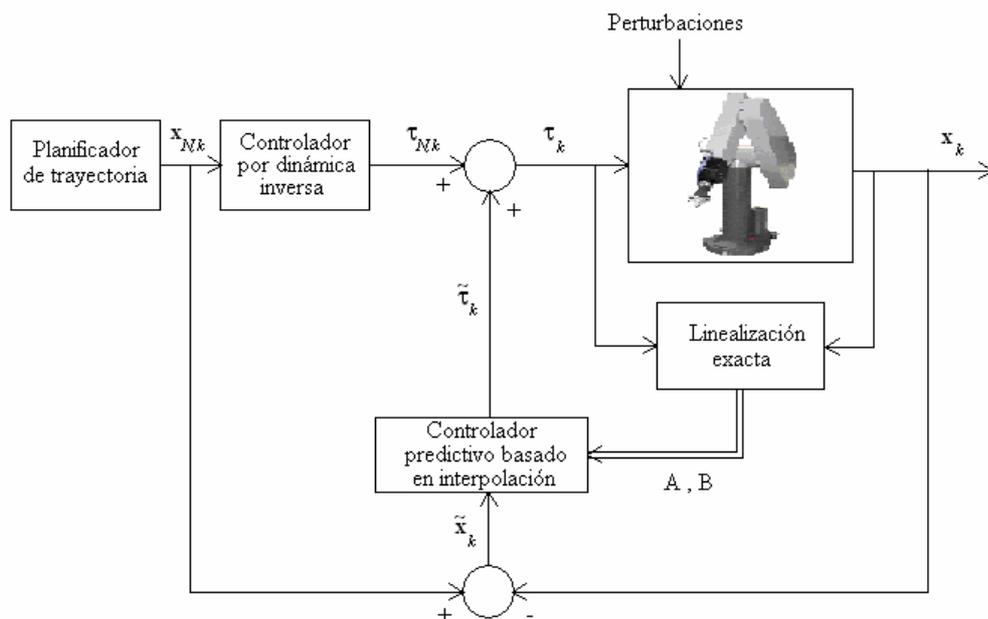


Figura 5-2: Esquema basado en el controlador de perturbación adaptativa utilizado para la aplicación de las técnicas predictivas en manipuladores.

Dado que la entrada al modelo linealizado y al sistema real sólo difieren en la entrada nominal, la calculada por el controlador de dinámica inversa, las ecuaciones para expresar las restricciones conservan su forma lineal y pueden ser incluidas fácilmente en el algoritmo de optimización.

5.3 Modelo de predicciones

El modelo de predicciones para el controlador predictivo, dadas las razones explicadas anteriormente, se obtiene tras aplicar un esquema de linealización instantánea sobre el modelo no lineal del manipulador.

En este caso, dado que la dinámica de las articulaciones pertenecientes a la muñeca no tienen un efecto muy acusado en la dinámica de las articulaciones del cuerpo del manipulador, se puede considerar la linealización del modelo consistente en las tres primeras articulaciones del manipulador, mientras que a las articulaciones de la muñeca se les puede aplicar un controlador más básico (en los capítulos siguientes, se aplicarán controladores PID, a los cuales se les añade una acción de control robusta para el rechazo de las perturbaciones e incertidumbres en el modelo).

El modelo para las tres primeras articulaciones del manipulador PUMA viene dado por la expresión (5-1), acotada a los tres primeros grados de libertad. Despejando las aceleraciones, se tiene:

$$\ddot{\theta}_1 = \frac{\tau_1 - n_1}{B_{11}} \quad (5-13)$$

$$\ddot{\theta}_2 = \frac{B_{33}(\tau_2 - n_2) - B_{23}(\tau_3 - n_3)}{B_{22}B_{33} - B_{23}B_{32}} \quad (5-14)$$

$$\ddot{\theta}_3 = \frac{-B_{23}(\tau_2 - n_2) + B_{22}(\tau_3 - n_3)}{B_{22}B_{33} - B_{23}B_{32}} \quad (5-15)$$

donde B_{ij} son los elementos de la matriz de inercia y n_i , los elementos del vector de términos no lineales de (5-1).

Entonces, se realiza la linealización en cada instante en torno al punto actual de la trayectoria de referencia, desarrollando las expresiones en serie de Taylor hasta primer orden. Así, para la primera articulación se tiene:

$$\ddot{\theta}_1 = \frac{\tau_1 - n_1}{B_{11}} = F_1(\theta_2, \theta_3; \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3; \tau_1) \quad (5-16)$$

Por lo que, desarrollando en serie de Taylor hasta primer orden en torno a la trayectoria de referencia, que se denota con el subíndice d , se tiene:

$$\begin{aligned} \ddot{\theta}_1 = F_1(\theta_2, \theta_3; \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3; \tau_1) \approx F_1]_d + \frac{\partial F_1}{\partial \theta_2}]_d (\theta_2 - \theta_{2d}) + \frac{\partial F_1}{\partial \theta_3}]_d (\theta_3 - \theta_{3d}) + \\ + \frac{\partial F_1}{\partial \dot{\theta}_1}]_d (\dot{\theta}_1 - \dot{\theta}_{1d}) + \frac{\partial F_1}{\partial \dot{\theta}_2}]_d (\dot{\theta}_2 - \dot{\theta}_{2d}) + \frac{\partial F_1}{\partial \dot{\theta}_3}]_d (\dot{\theta}_3 - \dot{\theta}_{3d}) + \frac{\partial F_1}{\partial \tau_1}]_d (\tau_1 - \tau_{1o}) \end{aligned} \quad (5-17)$$

donde τ_{1o} es el par aplicado a la primera articulación para seguir la trayectoria de referencia, es decir, el par calculado por el controlador de dinámica inversa. Teniendo en cuenta la definición habitual para el error de seguimiento $\tilde{\theta}_i = \theta_{id} - \theta_i$ y, definiendo $\tilde{\tau}_i = \tau_i - \tau_{io}$ como la diferencia de par que se va a aplicar respecto al par calculado τ_{io} , la expresión (5-17) queda:

$$\ddot{\theta}_1 = \frac{\partial F_1}{\partial \theta_2}]_d \tilde{\theta}_2 + \frac{\partial F_1}{\partial \theta_3}]_d \tilde{\theta}_3 + \frac{\partial F_1}{\partial \dot{\theta}_1}]_d \dot{\tilde{\theta}}_1 + \frac{\partial F_1}{\partial \dot{\theta}_2}]_d \dot{\tilde{\theta}}_2 + \frac{\partial F_1}{\partial \dot{\theta}_3}]_d \dot{\tilde{\theta}}_3 + \frac{\partial F_1}{\partial \tau_1}]_d \tilde{\tau}_1 \quad (5-18)$$

Se procede de semejante forma para las dos siguientes articulaciones. Agrupando los términos de las derivadas parciales para expresarlo de forma matricial, queda definido el modelo linealizado para el sistema:

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\tilde{\boldsymbol{\tau}} \quad (5-19)$$

el cual expresa las desviaciones, respecto a la trayectoria de referencia, de la trayectoria actual. Con este modelo linealizado de predicciones, se pueden incorporar al algoritmo de optimización las ligaduras para la entrada al sistema no lineal de forma directa.

Este proceso de linealización instantánea, además, no conlleva un costo computacional apreciable, pues las expresiones de las distintas derivadas parciales que componen las matrices de (5-19) se obtienen *off-line* y se actualizan sus valores con los valores actuales del estado. La linealización completa del modelo aparece en el Anexo G.

5.4 Problema de optimización

La ley de control que gobierna el lazo cerrado, adoptando el algoritmo de interpolación LM, queda entonces definida por:

$$\tilde{\boldsymbol{\tau}} = (1 - \alpha)\tilde{\boldsymbol{\tau}}_{LQ} + \alpha\tilde{\boldsymbol{\tau}}_{ML} = (1 - \alpha)\mathbf{K}_{LQ}\tilde{\mathbf{x}} + \alpha\mathbf{K}_{ML}\tilde{\mathbf{x}} \quad (5-20)$$

Las predicciones en la entrada y en el estado vienen dadas por:

$$\tilde{\boldsymbol{\tau}}_j = (1 - \alpha)\mathbf{K}_{LQ}\boldsymbol{\Phi}_{LQ}^j\tilde{\mathbf{x}} + \alpha\mathbf{K}_{ML}\boldsymbol{\Phi}_{ML}^j\tilde{\mathbf{x}} \quad (5-21)$$

$$\tilde{\mathbf{x}}_j = (1 - \alpha)\boldsymbol{\Phi}_{LQ}^j\tilde{\mathbf{x}} + \alpha\boldsymbol{\Phi}_{ML}^j\tilde{\mathbf{x}} \quad (5-22)$$

con $j = 0, \dots, n_c$. Las restricciones, en este caso, se imponen a la máxima entrada al manipulador y a la desviación máxima permitida respecto a la trayectoria de referencia.

Sea $\tilde{\mathbf{x}}_{\max}^+ = \max\{\tilde{\mathbf{x}} - \mathbf{x}_o\}$ el máximo sobrepasamiento positivo permitido del estado respecto a la trayectoria de referencia \mathbf{x}_o y $\tilde{\mathbf{x}}_{\max}^- = \max\{\mathbf{x}_o - \tilde{\mathbf{x}}\}$, el máximo sobrepasamiento negativo.

Entonces, las restricciones al estado pueden escribirse, utilizando (5-22), de la siguiente forma:

$$\left(-\boldsymbol{\varphi}_{LQ}^j \tilde{\mathbf{x}} + \boldsymbol{\varphi}_{ML}^j \tilde{\mathbf{x}}\right) \alpha \leq \tilde{\mathbf{x}}_{\max}^+ - \boldsymbol{\varphi}_{LQ}^j \tilde{\mathbf{x}} \quad (5-23)$$

$$-\left(-\boldsymbol{\varphi}_{LQ}^j \tilde{\mathbf{x}} + \boldsymbol{\varphi}_{ML}^j \tilde{\mathbf{x}}\right) \alpha \leq -\tilde{\mathbf{x}}_{\max}^- + \boldsymbol{\varphi}_{LQ}^j \tilde{\mathbf{x}} \quad (5-24)$$

De la misma forma, siendo $\boldsymbol{\tau}_M$ el valor máximo de entrada (y $-\boldsymbol{\tau}_M$ el mínimo), las restricciones a la entrada, utilizando (5-21),

$$\left(-\mathbf{K}_{LQ} \boldsymbol{\varphi}_{LQ}^j \tilde{\mathbf{x}} - \mathbf{K}_{ML} \boldsymbol{\varphi}_{ML}^j \tilde{\mathbf{x}}\right) \alpha \leq \boldsymbol{\tau}_M - \boldsymbol{\tau}_o - \mathbf{K}_{LQ} \boldsymbol{\varphi}_{LQ}^j \tilde{\mathbf{x}} \quad (5-25)$$

$$-\left(-\mathbf{K}_{LQ} \boldsymbol{\varphi}_{LQ}^j \tilde{\mathbf{x}} - \mathbf{K}_{ML} \boldsymbol{\varphi}_{ML}^j \tilde{\mathbf{x}}\right) \alpha \leq -\boldsymbol{\tau}_M + \boldsymbol{\tau}_o + \mathbf{K}_{LQ} \boldsymbol{\varphi}_{LQ}^j \tilde{\mathbf{x}} \quad (5-26)$$

donde $\boldsymbol{\tau}_o$ es el par calculado por el controlador de dinámica inversa para alcanzar el estado de referencia \mathbf{x}_o .

Entonces, el problema de optimización que se debe resolver deriva en el siguiente problema de programación lineal:

$$\begin{aligned} & \min_{\alpha} \alpha \\ & \text{sujeto a:} \\ & \left(-\boldsymbol{\varphi}_{LQ}^j \tilde{\mathbf{x}} + \boldsymbol{\varphi}_{ML}^j \tilde{\mathbf{x}}\right) \alpha \leq \tilde{\mathbf{x}}_{\max}^+ - \boldsymbol{\varphi}_{LQ}^j \tilde{\mathbf{x}} \\ & -\left(-\boldsymbol{\varphi}_{LQ}^j \tilde{\mathbf{x}} + \boldsymbol{\varphi}_{ML}^j \tilde{\mathbf{x}}\right) \alpha \leq -\tilde{\mathbf{x}}_{\max}^- + \boldsymbol{\varphi}_{LQ}^j \tilde{\mathbf{x}} \quad (5-27) \\ & \left(-\mathbf{K}_{LQ} \boldsymbol{\varphi}_{LQ}^j \tilde{\mathbf{x}} - \mathbf{K}_{ML} \boldsymbol{\varphi}_{ML}^j \tilde{\mathbf{x}}\right) \alpha \leq \boldsymbol{\tau}_M - \boldsymbol{\tau}_o - \mathbf{K}_{LQ} \boldsymbol{\varphi}_{LQ}^j \tilde{\mathbf{x}} \\ & -\left(-\mathbf{K}_{LQ} \boldsymbol{\varphi}_{LQ}^j \tilde{\mathbf{x}} - \mathbf{K}_{ML} \boldsymbol{\varphi}_{ML}^j \tilde{\mathbf{x}}\right) \alpha \leq -\boldsymbol{\tau}_M + \boldsymbol{\tau}_o + \mathbf{K}_{LQ} \boldsymbol{\varphi}_{LQ}^j \tilde{\mathbf{x}} \\ & 0 \leq \alpha \leq 1 \end{aligned}$$

El valor óptimo α_o obtenido de la resolución de este problema es utilizado en el instante actual para aplicar la ley de control:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_o + (1 - \alpha_o) \mathbf{K}_{LQ} \tilde{\mathbf{x}} + \alpha_o \mathbf{K}_{ML} \tilde{\mathbf{x}} \quad (5-28)$$

Con ello, se asegura un comportamiento óptimo que no viola las ligaduras impuestas. A continuación, se muestra el algoritmo completo del controlador IAPC:

- Paso 0) Fijar los valores de las restricciones $\tilde{\mathbf{x}}_{\max}^+$, $\tilde{\mathbf{x}}_{\max}^-$ y $\boldsymbol{\tau}_M$.
- Paso 1) Calcular las entradas nominales $\boldsymbol{\tau}_o$ mediante la trayectoria de referencia dada por $\mathbf{x}_o = [\boldsymbol{\theta}_d \quad \dot{\boldsymbol{\theta}}_d]^T$ y las aceleraciones $\ddot{\boldsymbol{\theta}}_d$, y haciendo uso de (5-1). Hacer $k=1$.
- Paso 2) Medir $\boldsymbol{\xi}_k = [\tilde{\boldsymbol{\theta}}_k \quad \dot{\tilde{\boldsymbol{\theta}}}_k]^T$, con $\tilde{\boldsymbol{\theta}}_k = \boldsymbol{\theta}_{k,d} - \boldsymbol{\theta}_k$ y $\dot{\tilde{\boldsymbol{\theta}}}_k = \dot{\boldsymbol{\theta}}_{k,d} - \dot{\boldsymbol{\theta}}_k$.
- Paso 3) Evaluar las derivadas parciales de (5-18) para formar las matrices del modelo linealizado (5-19).
- Paso 4) Obtener las ganancias \mathbf{K}_{LQ} y \mathbf{K}_{ML} resolviendo los correspondientes problemas de optimización de tiempo infinito sin restricciones (4-34).
- Paso 5) Obtener $\boldsymbol{\varphi}_{LQ}$ y $\boldsymbol{\varphi}_{ML}$ mediante (4-52).
- Paso 6) Resolver el problema de programación lineal (5-27) para obtener el valor óptimo α_o del parámetro de interpolación α .
- Paso 7) Aplicar la entrada al sistema (5-28).
- Paso 8) Hacer $k=k+1$, esperar hasta el siguiente período de muestreo e ir al Paso 2.

Como se puede observar, las entradas nominales resultado de la aplicación del controlador de dinámica inversa (5-1) se calculan *off-line* haciendo uso de la trayectoria de referencia. Con ello, se reduce parte de la complejidad computacional del algoritmo.

Una vez obtenido el modelo linealizado, se resuelven los problemas LQ para el cálculo de ganancias, los cuales no añaden al algoritmo un gran coste computacional, pues el algoritmo de solución de la ecuación de Riccati es sencillo y rápido. El elemento que aporta mayor coste computacional es el problema de programación lineal (5-27), el cual es tanto más complejo cuanto mayor sea el horizonte de predicción, por el número de restricciones que se deben evaluar.

Una forma de rebajar el coste computacional del algoritmo consiste en efectuar la linealización del modelo cada cierto número de períodos de muestreo, de forma que se mantenga el modelo calculado durante dichos instantes para la obtención de las ganancias \mathbf{K}_{LQ} y \mathbf{K}_{ML} , así como para las variables $\boldsymbol{\varphi}_{LQ}$ y $\boldsymbol{\varphi}_{ML}$, con lo que sólo habrá que replantear el problema de programación lineal para los nuevos valores de estado obtenidos.

En el siguiente apartado se expondrán los principales resultados obtenidos con esta técnica, verificando la eficacia de este método respecto a aquellos que no tienen en cuenta las restricciones a las que está sujeto el manipulador.

5.5 Resultados con el controlador IAPC

Se muestran aquí los resultados obtenidos con el controlador predictivo IAPC definido en el capítulo anterior. Con este controlador se da solución al problema que surge con las restricciones impuestas al manipulador. El caso habitual es que el sistema tenga restringido los valores máximo y mínimo de entrada al actuador. Además, existen ligaduras de tipo mecánico en cuanto a los valores máximos de posición alcanzadas por cada una de las articulaciones, así como las velocidades de las mismas.

Normalmente el planificador de la trayectoria que va a llevar a cabo el manipulador es el que tiene en cuenta estas ligaduras. Por tanto, aunque exista desviación del modelo asumido para el manipulador respecto al robot real, tanto las posiciones como las velocidades y las entradas cumplirán las ligaduras impuestas al sistema.

Los controladores presentados en capítulos anteriores no tienen en cuenta en su diseño las ligaduras del sistema, ya que asumen este planteamiento. Sin embargo, ante situaciones de presencia de ruido o incertidumbres paramétricas acusadas, dichas ligaduras se pueden violar. Es en estos casos donde la estrategia IAPC cobra importancia. Con la misma, se da una solución de control de bajo coste y que asegura la inviolabilidad de las ligaduras, aún en presencia de ruido o perturbaciones.

En los siguientes apartados se exponen los resultados en simulación del manipulador PUMA 560 para mostrar la eficiencia de la estrategia predictiva propuesta. Esta estrategia es comparada con controlador LQ con acción robusta. Esta estrategia se basa en el esquema IAPC, pero sin interpolación, aplicando sencillamente la solución LQ. Para dar cuenta de la posible desviación del modelo respecto del robot real, se añade la acción robusta autoadaptativa aplicada en el controlador ARC.

En primer lugar, se supone la no presencia de incertidumbres en el modelo. Este caso servirá para comprobar las prestaciones de la estrategia de control propuesta en cuanto a evitar la violación de las ligaduras.

A continuación, se incorporan incertidumbres paramétricas, en forma de desviaciones en las masas y longitudes de los elementos que componen el robot, de forma que se evalúa la respuesta de las distintas estrategias de control.

En el siguiente apartado se aplicará la solución ideada en el controlador ARC para dar respuesta a la presencia de incertidumbres en el modelo. Esta

nueva estrategia se denominará RIAPC y será definida e implementada para el manipulador en estudio.

5.5.1 Modelo sin incertidumbres

Se exponen a continuación los resultados obtenidos con los controladores LQ robusto e IAPC en el caso de utilizar un modelo no perturbado. Para ello se realiza un experimento (experimento 5.1) en el que se utiliza como modelo de predicciones el mismo modelo que para el cálculo de la dinámica del robot.

Las ligaduras impuestas al sistema son en la entrada y en el estado. Sin embargo, estas últimas son respetadas por los distintos controladores diseñados. La violación de las ligaduras se produce por la presencia de ruido de medida que afecta al sistema durante el experimento. A continuación se detallan las características de este experimento.

Experimento 5.1

Sistema:	Manipulador PUMA-560.
Controlador:	LQ Robusto (RLQ) e IAPC.
* Acción LQ:	$\mathbf{V} = \text{diag}([4.8 \ 1.2 \ 4.8 \ 1.2 \ 4.8 \ 1.2 \ 3.2 \ 0.8 \ 3.2 \ 0.8 \ 3.2 \ 0.8])$ $\mathbf{W} = \text{diag}(0.001 * [25 \ 25 \ 25 \ 5 \ 5 \ 5])$
* Acción robusta:	$\varepsilon = 1, \mathbf{Q} = \text{diag}([0 \ 0 \ 0 \ 0 \ 0 \ 0,1 \ 1 \ 1 \ 1 \ 1 \ 1])$
* Adaptación de parámetro de acción robusta:	$\mathbf{R}_{ad} = h \cdot \text{diag}([25 \ 25 \ 25 \ 25 \ 25 \ 25])$ $\mathbf{Q}_{ad} = 100 \cdot \text{diag}([1 \ 3.75 \ 2.5 \ 2.5 \ 2.5 \ 2.5, 2.5 \ 6 \ 5 \ 6 \ 6 \ 6])$ $\gamma = 1, h = 0.001$
* Acción ML:	$\mathbf{V}' = 0.1 \cdot \mathbf{V}$ $\mathbf{W}' = 1e4 \cdot \mathbf{W}$
* Modelo del manipulador:	Modelo de manipulador de [Lew04] sin perturbaciones.
* Restricciones:	$\boldsymbol{\tau}_M = [75 \ 36.5 \ 11 \ 10 \ 10 \ 10]$ $\tilde{\mathbf{x}}_{\max}^+ = [15 \ 15 \ 15 \ 5 \ 5 \ 5] * \frac{\pi}{180}, \tilde{\mathbf{x}}_{\max}^- = -\tilde{\mathbf{x}}_{\max}^+$
Trayectoria de referencia:	- Trayectoria en el espacio de las articulaciones generada a través de la unión de cuatro puntos en el espacio de tareas unidos mediante polinomios de séptimo orden. Dicha trayectoria es recorrida dos veces a lo largo del movimiento.

En las figuras 5-3, 5-4 y 5-5 se muestran, respectivamente, los resultados obtenidos para las tres primeras articulaciones del manipulador.

Se puede observar en la articulación 2, en los instantes de tiempo en torno a $t = 5s.$, que el controlador IAPC ofrece una mejor respuesta que el LQ, dado que evita la saturación de la entrada. Dicha saturación se produce por la presencia de ruido de medida en los valores de posición de las articulaciones.

El valor máximo de entrada para la segunda articulación es de 36.5 Nm. Con el controlador LQ se produce una saturación después de la cual el error de seguimiento llega a tomar un valor de 0.20 radianes. En el caso del controlador IAPC, esta situación de saturación no se produce gracias a la interpolación que se realiza de la solución LQ con la solución ML.

De esta forma, además, se consigue reducir el error de seguimiento hasta un valor de 0.15 radianes, un 25% menor que en el caso anterior.

En la tercera articulación (figura 5-5) también se observa que el error de seguimiento máximo se reduce de 0.12 radianes en el caso del controlador LQ a 0.05 radianes en el caso de la estrategia IAPC.

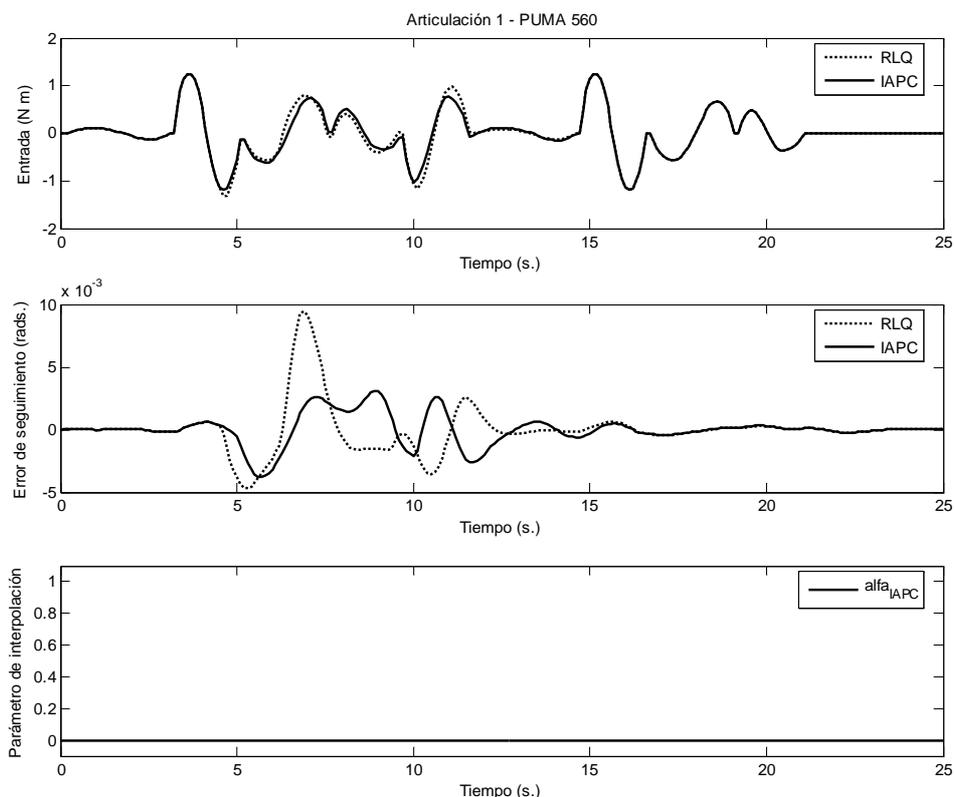


Figura 5-3: Evolución de la entrada, error de seguimiento y parámetro de interpolación para la articulación 1.

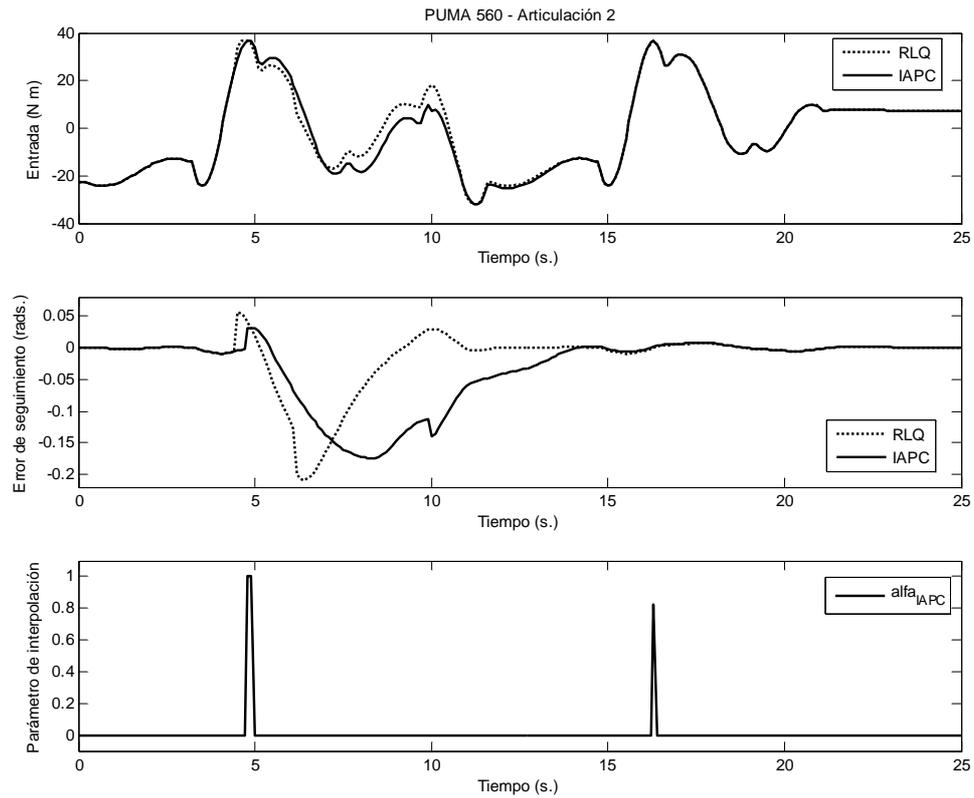


Figura 5-4: Evolución de la entrada, error de seguimiento y parámetro de interpolación para la articulación 2.

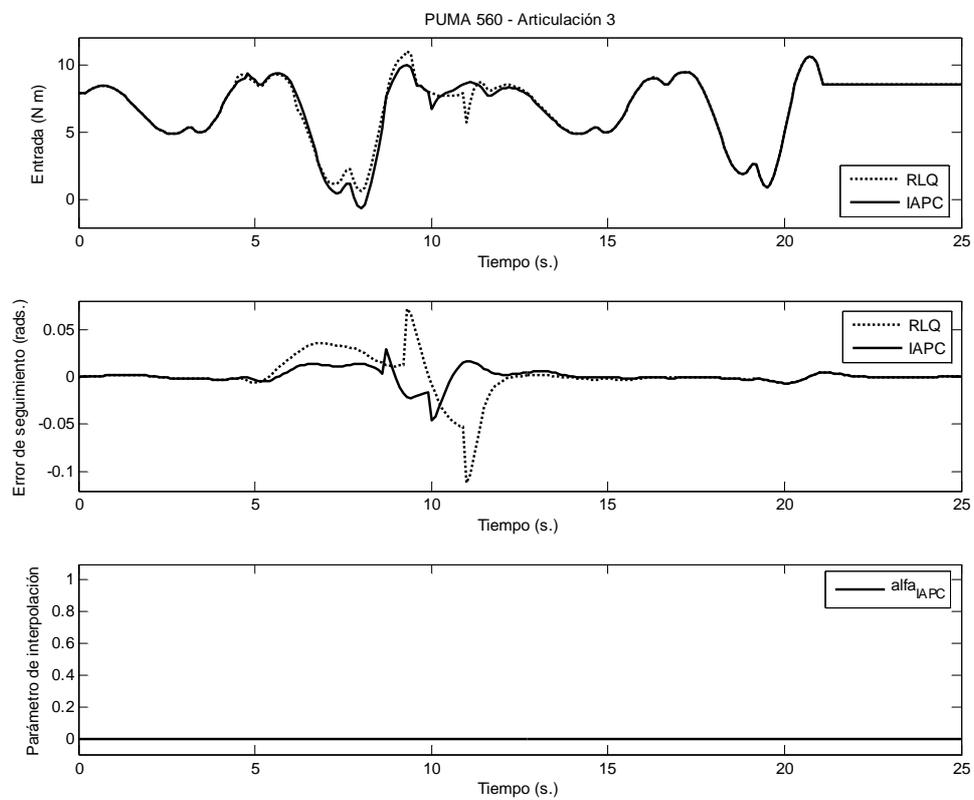


Figura 5-5: Evolución de la entrada, error de seguimiento y parámetro de interpolación para la articulación 3.

5.5.2 Modelo con incertidumbres

Se exponen a continuación los resultados obtenidos con los mismos controladores que en el apartado anterior, pero considerando la presencia de incertidumbres en el modelo del manipulador. Dicha incertidumbre consiste en considerar las masas y longitudes del manipulador con una desviación de un 15% respecto a sus valores reales.

Se consideran ligaduras en la entrada y el estado del sistema, y la violación de las mismas se provoca por la consideración de perturbaciones en forma de ruido de medida que afecta al sistema durante el experimento.

En el cuadro correspondiente al Experimento 2 se detallan los parámetros de controladores y demás aspectos del sistema utilizados para esta simulación.

Experimento 5.2

Sistema: Manipulador PUMA-560.

Controlador: LQ Robusto (RLQ) e IAPC.

* Acción LQ:

$$\mathbf{V} = \text{diag}([4.8 \ 1.2 \ 4.8 \ 1.2 \ 4.8 \ 1.2 \ 3.2 \ 0.8 \ 3.2 \ 0.8 \ 3.2 \ 0.8])$$

$$\mathbf{W} = \text{diag}(0.001 * [25 \ 25 \ 25 \ 5 \ 5 \ 5]).$$

* Acción robusta:

$$\varepsilon = 1, \mathbf{Q} = \text{diag}([0 \ 0 \ 0 \ 0 \ 0 \ 0,1 \ 1 \ 1 \ 1 \ 1 \ 1]).$$

* Adaptación de parámetro de acción robusta:

$$\mathbf{R}_{ad} = h \cdot \text{diag}([25 \ 25 \ 25 \ 25 \ 25 \ 25])$$

$$\mathbf{Q}_{ad} = 100 \cdot \text{diag}([1 \ 3.75 \ 2.5 \ 2.5 \ 2.5 \ 2.5, 2.5 \ 6 \ 5 \ 6 \ 6 \ 6])$$

$$\gamma = 1, h = 0.001.$$

* Acción ML:

$$\mathbf{V}' = 0.1 \cdot \mathbf{V}$$

$$\mathbf{W}' = 1e4 \cdot \mathbf{W}$$

* Modelo del manipulador:

Modelo de manipulador de [Lew04] en presencia de incertidumbres paramétricas (un 15% de desviación máxima en las masas y longitudes de las diferentes articulaciones).

* Restricciones:

$$\boldsymbol{\tau}_M = [75 \ 36.5 \ 11 \ 10 \ 10 \ 10]$$

$$\tilde{\mathbf{x}}_{\max}^+ = [15 \ 15 \ 15 \ 5 \ 5 \ 5] * \frac{\pi}{180}, \tilde{\mathbf{x}}_{\max}^- = -\tilde{\mathbf{x}}_{\max}^+$$

Trayectoria de referencia:

- Trayectoria en el espacio de las articulaciones generada a través de la unión de cuatro puntos en el espacio de tareas unidos mediante polinomios de séptimo orden. Dicha trayectoria es recorrida dos veces a lo largo del movimiento.

Los resultados de este experimento se muestran en las figuras Figura 5-6, Figura 5-7 y Figura 5-8, donde se representan las entradas, error de seguimiento y evolución del parámetro de interpolación para cada una de las articulaciones del cuerpo del manipulador.

Al contrario que el caso anterior, el controlador IAPC en este caso da una respuesta peor que el LQ Robusto. Como se puede ver en el caso de la articulación 2, incluso la saturación de la entrada es menor con el controlador LQ Robusto. Los errores de seguimiento son en general mayores con el controlador IAPC.

La razón de este comportamiento radica en que el controlador LQ con acción robusta presenta un buen rechazo a perturbaciones e incertidumbres debido a la acción robusta autoadaptativa añadida al controlador original LQ.

Luego se puede concluir a la vista de los resultados que, en presencia de incertidumbres, el controlador IAPC no resuelve completamente el problema de control ya que se pueden producir violación de las ligaduras.

La solución inmediata a este problema es la adición de la acción robusta autoadaptativa al controlador IAPC. Esta es la solución que se adopta en esta tesis para dar una solución global al problema del seguimiento de trayectorias por parte de manipuladores con ligaduras en presencia de incertidumbres y/o perturbaciones.

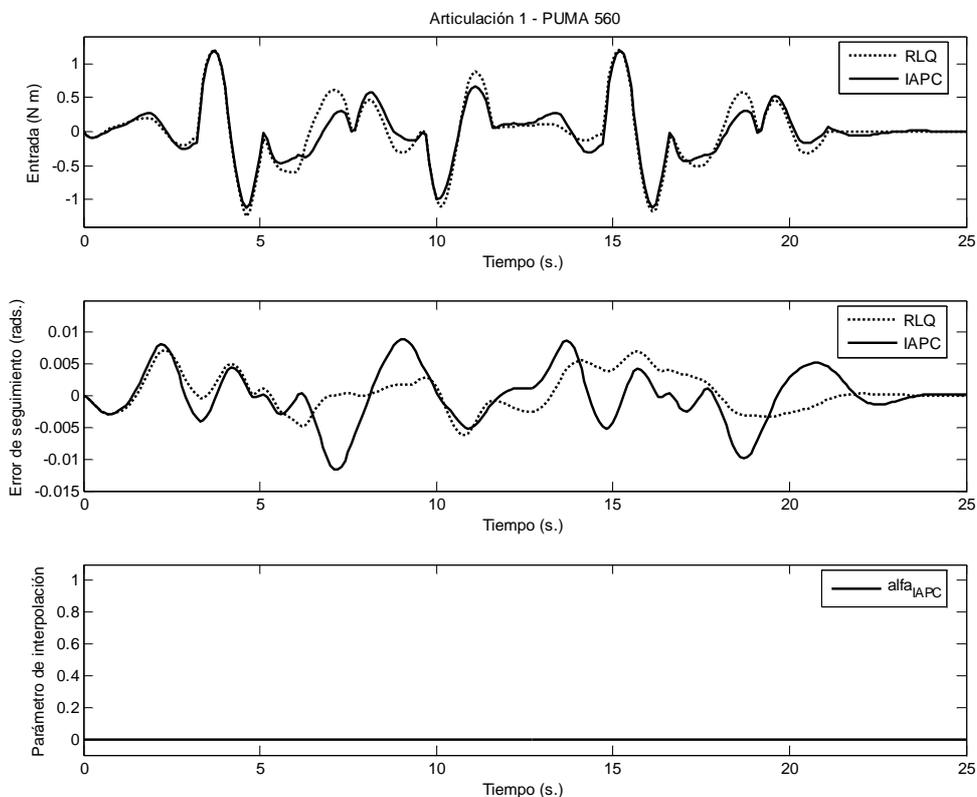


Figura 5-6: Evolución de la entrada, error de seguimiento y parámetro de interpolación para la articulación 1.

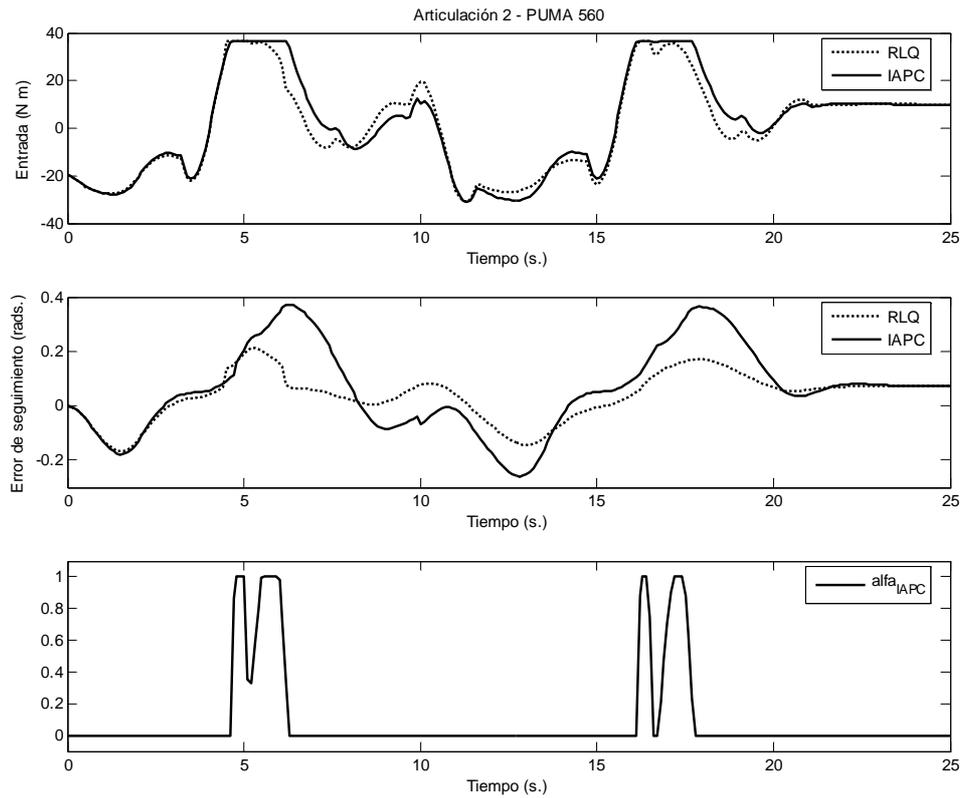


Figura 5-7: Evolución de la entrada, error de seguimiento y parámetro de interpolación para la articulación 2.

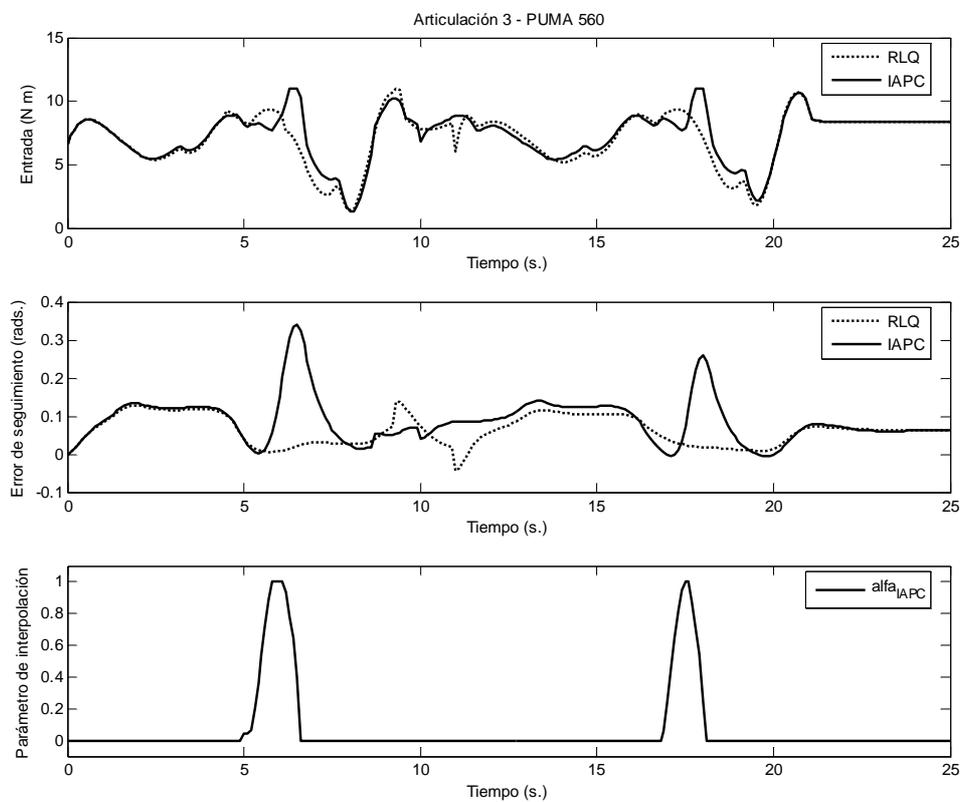


Figura 5-8: Evolución de la entrada, error de seguimiento y parámetro de interpolación para la articulación 3.

5.6 Añadiendo robustez para rechazo a perturbaciones: Controlador RIAPC

En el apartado anterior se demuestra que el controlador IAPC aporta una solución al caso de la presencia de ligaduras. El seguimiento de trayectorias resulta más eficiente y se respetan las ligaduras impuestas al sistema. Sin embargo, esta solución es eficiente si el sistema no presenta incertidumbres o perturbaciones. En caso contrario, otro tipo de soluciones presentan mejores prestaciones que el controlador IAPC.

Para mejorar el rendimiento de esta técnica, se añade a este controlador la acción robusta autoadaptativa que se aporta a otro tipo de controladores, como es el caso del controlador ARC o el LQ Robusto. De esta forma, esta acción de control se encarga de rechazar eficazmente las perturbaciones e incertidumbres en el modelo, mientras que el algoritmo IAPC se encarga de evitar que se violen las ligaduras. El controlador resultante se denominará RIAPC (*Robust Interpolated based Adaptive Perturbation Control*).

En los siguientes apartados se exponen los detalles de implementación del controlador RIAPC así como los resultados obtenidos con el mismo, en comparación con los obtenidos con los otros controladores.

5.6.1 Esquema de control RIAPC

El esquema de control RIAPC resulta de una generalización del controlador IAPC definido en el capítulo anterior, pero añadiendo la acción robusta autoadaptativa introducida en el capítulo 3. En la figura 5-9 se puede observar que en el cálculo de las entradas al sistema, se añade el nuevo término robusto definido para el rechazo de perturbaciones e incertidumbres en el modelo.

Así, la acción de control viene dada por:

$$\boldsymbol{\tau}_k = \boldsymbol{\tau}_{k,o} + (1 - \alpha_{k,o}) \mathbf{K}_{LQ} \tilde{\mathbf{x}}_k + \alpha_o \mathbf{K}_{ML} \tilde{\mathbf{x}}_k + \boldsymbol{\tau}_{R,k} \quad (5-29)$$

donde el nuevo término $\boldsymbol{\tau}_{R,k}$ viene dado por:

$$\boldsymbol{\tau}_{R,k} = \begin{cases} \frac{\rho_k}{\|\mathbf{D}^t \mathbf{Q} \tilde{\boldsymbol{\xi}}_k\|} \mathbf{D}^t \mathbf{Q} \tilde{\boldsymbol{\xi}}_k & , \text{ si } \|\mathbf{D}^t \mathbf{Q} \tilde{\boldsymbol{\xi}}_k\| \geq \varepsilon \\ \frac{\rho_k}{\varepsilon} \mathbf{D}^t \mathbf{Q} \tilde{\boldsymbol{\xi}}_k & , \text{ si } \|\mathbf{D}^t \mathbf{Q} \tilde{\boldsymbol{\xi}}_k\| < \varepsilon \end{cases} \quad (5-30)$$

Cada uno de los términos que aparecen en la ecuación ha sido definido con anterioridad (ver ecuaciones 2-17, 2-19, 2-20 y 2-52 a 2-55 para la parte robusta adaptativa y 4-34, 4-52, 4-83, 4-84 y 4-92 para la parte predictiva).

Experimento 5.3

Sistema: Manipulador PUMA-560.

Controlador: LQ Robusto, IAPC y RIAPC.

* Acción LQ:

$$\mathbf{V} = \text{diag}([4.8 \ 1.2 \ 4.8 \ 1.2 \ 4.8 \ 1.2 \ 3.2 \ 0.8 \ 3.2 \ 0.8 \ 3.2 \ 0.8])$$

$$\mathbf{W} = \text{diag}(0.001 * [25 \ 25 \ 25 \ 5 \ 5 \ 5])$$

* Acción robusta:

$$\varepsilon = 1, \mathbf{Q} = \text{diag}([0 \ 0 \ 0 \ 0 \ 0 \ 0,1 \ 1 \ 1 \ 1 \ 1 \ 1]).$$

* Adaptación de parámetro de acción robusta:

$$\mathbf{R}_{ad} = h \cdot \text{diag}([25 \ 25 \ 25 \ 25 \ 25 \ 25])$$

$$\mathbf{Q}_{ad} = 100 \cdot \text{diag}([1 \ 3.75 \ 2.5 \ 2.5 \ 2.5 \ 2.5, 2.5 \ 6 \ 5 \ 6 \ 6 \ 6])$$

$$\gamma = 1, h = 0.001.$$

* Acción ML:

$$\mathbf{V}' = 0.1 \cdot \mathbf{V}$$

$$\mathbf{W}' = 1e4 \cdot \mathbf{W}$$

* Modelo del manipulador:

Modelo de manipulador de [Lew04] en presencia de incertidumbres paramétricas (un 15% de desviación máxima en las masas y longitudes de las diferentes articulaciones).

* Restricciones:

$$\boldsymbol{\tau}_M = [75 \ 36.5 \ 11 \ 10 \ 10 \ 10]$$

$$\tilde{\mathbf{x}}_{\max}^+ = [15 \ 15 \ 15 \ 5 \ 5 \ 5] * \frac{\pi}{180}, \tilde{\mathbf{x}}_{\max}^- = -\tilde{\mathbf{x}}_{\max}^+$$

* Trayectoria de referencia:

- Trayectoria en el espacio de las articulaciones generada a través de la unión de cuatro puntos en el espacio de tareas unidos mediante polinomios de séptimo orden. Dicha trayectoria es recorrida dos veces a lo largo del movimiento.

La razón de este comportamiento es la presencia de incertidumbres en el modelo del manipulador. El controlador LQ las tiene en cuenta ya que se le ha añadido la acción robusta autoadaptativa del controlador ARC. Sin embargo, no es capaz de evitar la violación de las ligaduras impuestas al sistema.

Por esta razón se propone la estrategia RIAPC, ya que corrige las desviaciones producidas por las incertidumbres en el modelo utilizado para realizar las predicciones del controlador predictivo, a la vez que evita la violación de las ligaduras del sistema.

La eficiencia de este controlador puede ser corroborada en las figuras 5-11 y 5-12, correspondientes a las articulaciones 2 y 3, donde se aprecia que los errores de seguimiento son bajos y las entradas no alcanzan los valores de saturación.

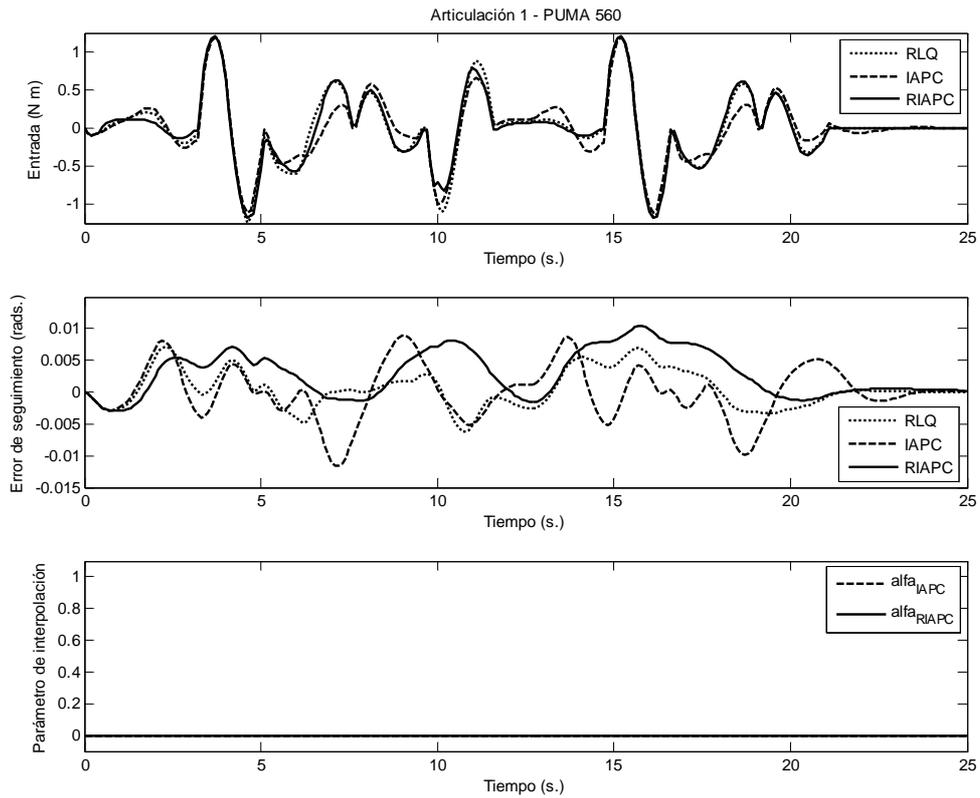


Figura 5-10: Evolución de la entrada, error de seguimiento y parámetro de interpolación para la articulación 1

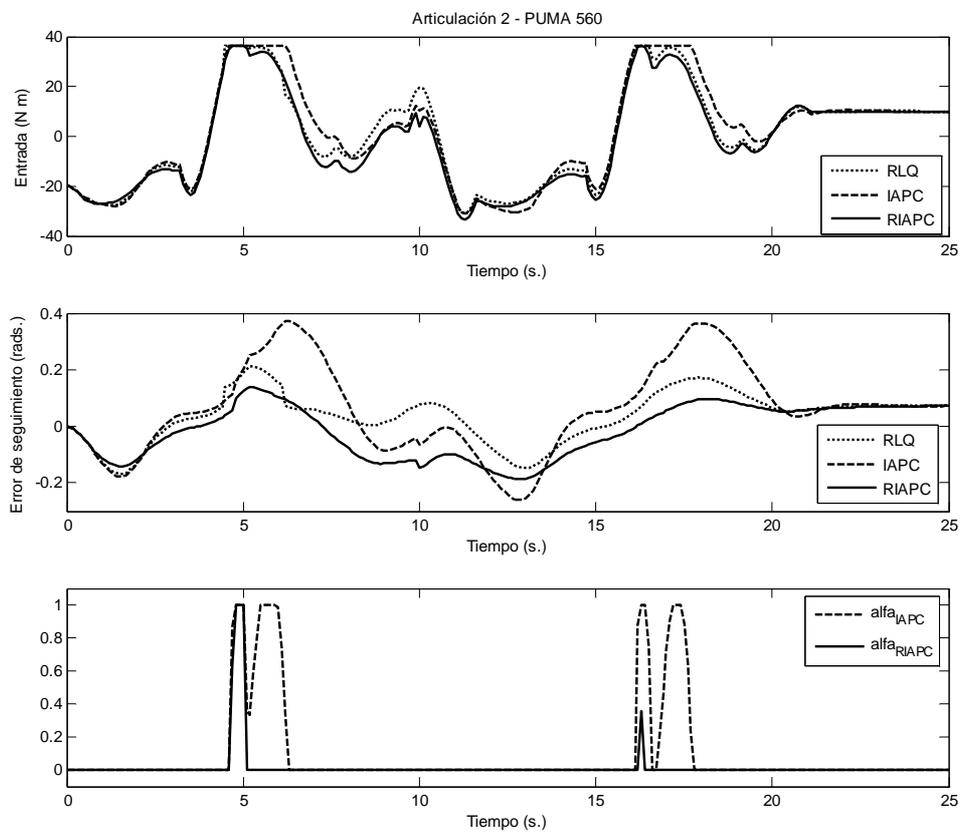


Figura 5-11: Evolución de la entrada, error de seguimiento y parámetro de interpolación para la articulación 2

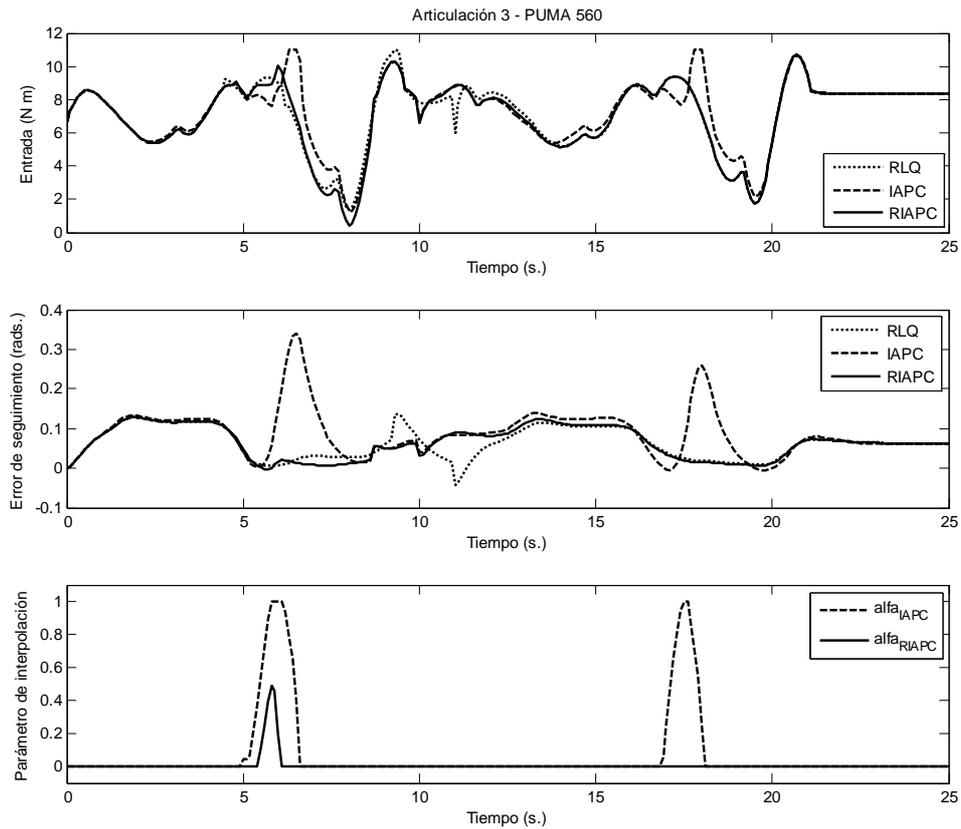


Figura 5-12: Evolución de la entrada, error de seguimiento y parámetro de interpolación para la articulación 3

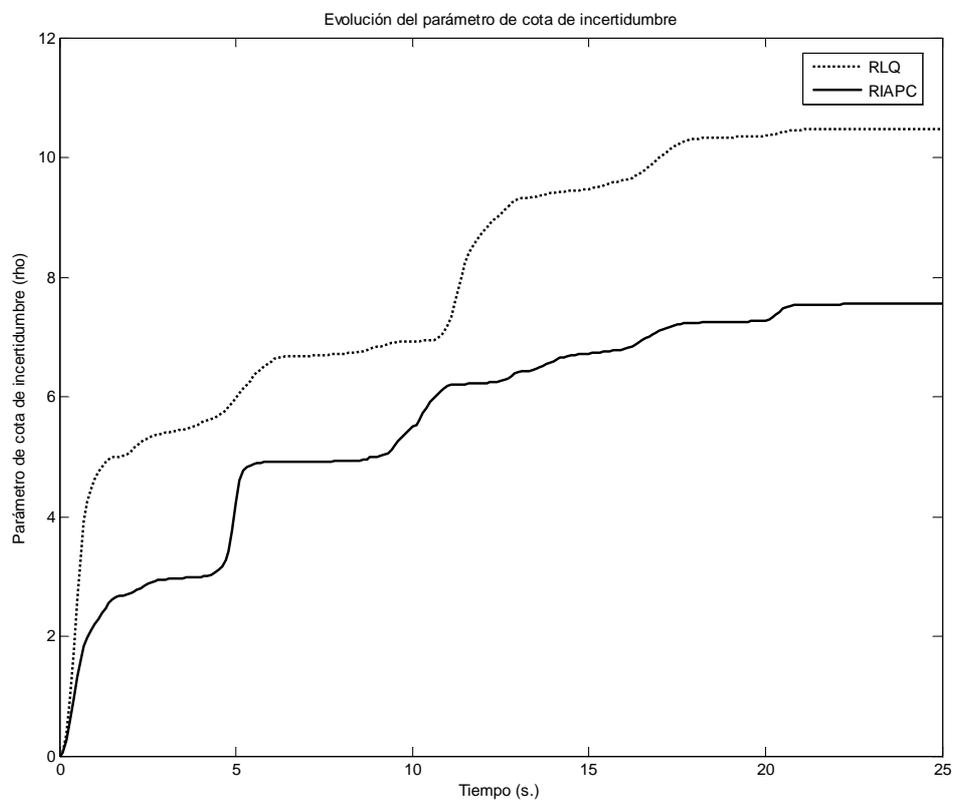


Figura 5-13: Evolución del parámetro de cota de incertidumbre

Sin embargo, en la articulación 1 (figura 5-10) no se consigue rebajar el error de seguimiento respecto a la respuesta ofrecida por el controlador LQ. Aún así, este resultado no reviste importancia debido a los bajos valores de error de seguimiento que se consiguen para esta articulación con los tres controladores.

En la figura 5-13 se representa la evolución del parámetro de cota de incertidumbre para los controladores LQ robusto y RIAPC. Como se observa, la evolución es similar. Sin embargo, en el segundo caso, dado que la interpolación provoca un mejor seguimiento de la consigna (ya que no se tienen situaciones de saturación de la entrada), la evolución de este parámetro es menor.

A la vista de los resultados obtenidos se puede concluir que el controlador RIAPC da una respuesta global a los distintos aspectos de control de manipuladores para el seguimiento de trayectorias, en cuanto a la eficacia del seguimiento, el rechazo a perturbaciones e incertidumbres en el modelo y la no violación de las ligaduras del sistema.

Esta solución no tiene un costo computacional excesivo, debido al uso de técnicas predictivas basadas en interpolación, las cuales deben resolver un problema de programación lineal sencillo en cada instante de muestreo. Además, la adaptación del parámetro de cota de incertidumbre en el caso de la acción robusta no supone, tal y como se explicó en el capítulo 3, un incremento apreciable del costo computacional.

Por ello se propone en esta tesis el controlador RIAPC como un controlador eficiente y de bajo costo computacional para resolver los principales problemas de control de manipuladores.

Parte III

**Mejora del costo
computacional de los
controladores usando
modelos neuronales**

Técnicas basadas en Redes Neuronales para la Identificación

La identificación del modelo dinámico de un sistema tiene gran importancia en sistemas complejos, multivariables, acoplados y/o en los que hay presentes no linealidades. Hay dos posibilidades de obtener un modelo para un sistema. Una es aplicar los principios físicos (mecánicas, eléctricas, termodinámicas, etc. Por los) que se rige el sistema y obtener el modelo matemático, con una mayor o menor exactitud en función de las aproximaciones que se hagan del sistema. Dado que este proceso resulta en ocasiones complejo, por el gran número de no linealidades presentes en el sistema o por la presencia de comportamientos dinámicos difícilmente modelables, se opta por la obtención de un modelo empírico para el sistema, resultado de la aplicación de técnicas de identificación basadas en redes neuronales usando datos recogidos del propio sistema que se quiere identificar. Para cumplir este objetivo se emplean habitualmente redes neuronales estáticas, en la que las neuronas se distribuyen en una red compuesta de capas y cuyas salidas son funciones no lineales de las entradas. Sin embargo, las redes neuronales dinámicas pueden favorecer la identificación de sistemas dinámicos, ya que el estado de las neuronas varía en el tiempo, de forma que pueden absorber la dinámica del sistema que tratan de identificar. Las redes neuronales dinámicas del tipo Hopfield están compuestas por una única capa de neuronas, parte de las cuales se dedican a generar la salida de la red. En este capítulo se introducen ciertas mejoras en los procesos de entrenamiento y validación de este tipo de redes para la identificación de sistemas dinámicos complejos, como puede ser el caso de los robots manipuladores estáticos.

6.1 Identificación de la dinámica del manipulador

En el capítulo 1 se estableció que las ecuaciones de la dinámica de un robot manipulador se obtienen habitualmente de la formulación matricial de Lagrange-Euler o de la recurrente de Newton-Euler, aunque otras formulaciones como la recursiva de Lagrange o la generalizada de d'Alambert-Lee pueden ser utilizadas para el mismo fin. La expresión resultante para la

dinámica da como resultado un conjunto de ecuaciones no lineales y fuertemente acopladas. Este conjunto de ecuaciones es el que se emplea en la mayoría de controladores para definir la entrada que se debe aplicar al robot para solucionar los problemas de control dinámico.

Así, el controlador por dinámica inversa o de par calculado obtiene la acción de control aplicando directamente las ecuaciones sobre la trayectoria planificada. La técnica de linealización por realimentación posibilita la aplicación de controladores lineales (por ejemplo, un controlador óptimo LQ o técnicas de control adaptativo) sobre un modelo linealizado del manipulador, mediante la cancelación de los términos no lineales de su dinámica, utilizando las mencionadas ecuaciones.

La propiedad de parametrización lineal, descrita en el capítulo 1 de esta tesis, utiliza también las ecuaciones de la dinámica para la linealización del modelo respecto a los parámetros dinámicos del mismo. Por último, en las estrategias de tipo predictivo se utilizan de nuevo las ecuaciones de la dinámica del manipulador como modelo de predicción para el cálculo de las acciones de control, haciendo que el sistema no viole las ligaduras presentes en su estado o en las entradas.

Describiendo con un poco más de detalle las ecuaciones de la dinámica, se tienen términos que dependen de las coordenadas generalizadas (ángulos o desplazamientos) como son los términos de inercia (pares de reacción inducidos por la aceleración que una articulación provoca sobre otra), las aceleraciones centrífuga y de Coriolis y los términos gravitatorios debido al peso de los elementos que componen la cadena articulada.

Además, las ecuaciones dinámicas también pueden contener términos que reflejan los rozamientos y fricciones presentes en el sistema. Estos términos dependen de parámetros de difícil evaluación, como pueden ser coeficientes de rozamiento viscoso o culombiano, a diferencia de los términos anteriores, que precisan el conocimiento de las masas, longitudes y momentos de inercia de cada uno de los elementos. Las fricciones, pues, añaden incertidumbre en el modelo calculado.

Además, existen otras no linealidades que solo aparecen en la construcción del prototipo mecánico, como pueden ser holguras en las articulaciones, zonas muertas de actuación de los motores, etc.

Estas características son difícilmente modelables de forma teórica, por lo que se recurre a modelos empíricos para llevar a cabo la identificación completa del sistema.

6.1.1 Identificación mediante redes neuronales

El uso de redes neuronales es una herramienta habitual en el campo de la identificación de sistemas dinámicos. Una red de estructura determinada es entrenada con datos de entrada-salida del sistema real que se va a identificar

para que aprenda el comportamiento dinámico del sistema. La red neuronal, por definición, realiza un mapeo no lineal entre los datos de entrada y salida, por lo que resultan muy convenientes para aprender la dinámica de sistemas no lineales.

En un principio, se emplearon redes estáticas tales como el perceptrón multicapa (MLP, de *Multi-Layer Perceptron*) [Rum86][Bro88]. Las entradas de estas redes corresponden a valores pasados de las entradas y salidas del sistema. Sin embargo, esta aproximación presenta algunos inconvenientes. El primero de ellos es la dificultad para elegir la estructura de entrada a la red. Otro inconveniente es que la estructura debe ser reentrenada cada vez que se cambie el período de muestreo del sistema. Además, el problema de control no lineal se entiende mejor cuando se formula en tiempo continuo que en la versión discreta.

Las redes neuronales dinámicas (DNN, de *Dynamic Neural Network*) de Hopfield [Hop82] y sus variantes [Hop84][Hop85][Hop86][Koi94], no presentan estos inconvenientes y son capaces de aproximar el comportamiento de sistemas dinámicos multivariables. Las DNNs se han mostrado útiles para la identificación de sistemas no lineales autónomos [Fum93][Kim98], sistemas afines a control multivariable [Gar99] y sistemas no lineales en general [Gar00]. Las condiciones de estabilidad para las DNN se han analizado en [Mat92] y [San99].

6.1.2 Mejoras en la identificación: inicialización de la DNN

Becerra *et al* [Bec02] han introducido algunas técnicas para la inicialización de los estados de las neuronas escondidas, para propósitos de entrenamiento y validación de redes neuronales con esta estructura. En particular, este trabajo propone, en el caso de entrenamiento, incluir dichos estados en el vector de parámetros de la DNN que se va a optimizar, así como inicializar estos estados de forma eficaz para el caso de validación. Así, se propone resolver un problema de optimización consistente en calcular los estados de dichas neuronas que conducen a igualar las derivadas de los estados de salida a las derivadas medidas empíricamente.

Con estas técnicas se resuelve uno de los problemas que plantea este tipo de estructuras neuronales para el caso de la identificación de sistemas, ya que la inicialización de los estados de las neuronas escondidas se ha hecho de forma aleatoria o, más habitualmente, igualándolos a cero.

En esta tesis se adopta esta idea para la inicialización de la DNN, aplicada a la identificación de un sistema robótico. Aunque las ecuaciones dinámicas están perfectamente estudiadas, el gran número de no linealidades no contempladas en el modelo, tal y como se comentó anteriormente, hace necesaria la obtención de un modelo empírico, basado en entradas y salidas reales del robot y, por tanto, más próximo al comportamiento dinámico real del mismo.

Además, para que el modelo empírico sea lo más fiable posible, se aportan técnicas de identificación que mejoran el resultado de identificación, como es el añadir un término a la función de costo que se va a optimizar de forma que impida que los estados de las neuronas escondidas adquieran valores elevados.

Este capítulo está organizado de la siguiente forma. En primer lugar, se introducen las DNN. A continuación, se describe el problema de entrenamiento de dicha red neuronal para, finalmente, aportar las técnicas de inicialización de los estados de las neuronas que redundan en una mejor identificación del sistema.

6.2 Redes neuronales dinámicas

Las redes neuronales dinámicas (DNN) en su forma recurrente fueron introducidas en el contexto de las memorias asociativas [Hop84][Koh99]. La realimentación añadida a la arquitectura de una red neuronal *feedforward* permite obtener un modelo dinámico en el espacio de los estados.

Dicho modelo está compuesto por un vector unidimensional de neuronas, cada una de las cuales viene descrita por la siguiente ecuación dinámica:

$$\dot{x}_i = -\beta_i x_i + \sum_{j=1}^n \omega_{ij} \sigma(x_j) + \sum_{j=1}^p \gamma_{ij} u_j \quad (6-1)$$

donde β_i , ω_{ij} y γ_{ij} son pesos ajustables, con $1/\beta_i$ constantes de tiempo positivas, x_i el estado de activación de la neurona i , $\{u_i\}_{i=1,\dots,p}$ las entradas a la red, con $p \leq n$, siendo n el número de neuronas de la red. La función $\sigma(\cdot)$ es típicamente una función no lineal tipo sigmoide, por ejemplo, la tangente hiperbólica, aplicada sobre el argumento (\cdot) . Este modelo puede observarse de forma gráfica en figura 6-1.

La red está formada por n neuronas, de las cuales se escogen las p primeras como salidas, dejando las $n-p$ restantes como neuronas escondidas. La red puede entonces expresarse de manera matricial como sigue:

$$\begin{aligned} \dot{x} &= -\beta \cdot x + \omega \cdot \sigma(x) + \gamma \cdot u \\ \hat{y} &= C_n \cdot x \end{aligned} \quad (6-2)$$

donde $x \in \mathfrak{R}^n$ es el vector de estado, $u \in \mathfrak{R}^p$ es el vector de entradas, $\omega \in \mathfrak{R}^{n \times n}$, $\sigma(x) = [\sigma(x_1) \dots \sigma(x_n)]$, $\gamma \in \mathfrak{R}^{n \times p}$, $\hat{y} \in \mathfrak{R}^p$ es el vector de salida, $C_n = [I_{p \times p} \quad 0_{p \times (n-p)}]$ y $\beta = \text{diag}([\beta_1 \dots \beta_n]) \in \mathfrak{R}^{n \times n}$ es una matriz diagonal.

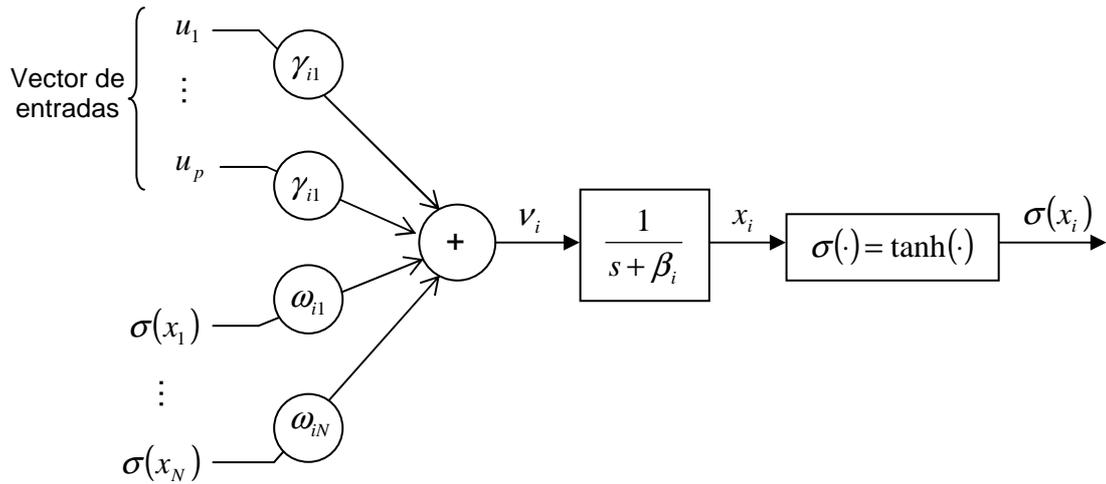


Figura 6-1: Diagrama de una neurona dinámica.

Esta estructura matricial puede observarse en la figura 6-2. Si $n > p$, se tienen $n - p$ neuronas escondidas, las cuales aumentan la capacidad de modelar la dinámica de un sistema [Hin86][Sej86].

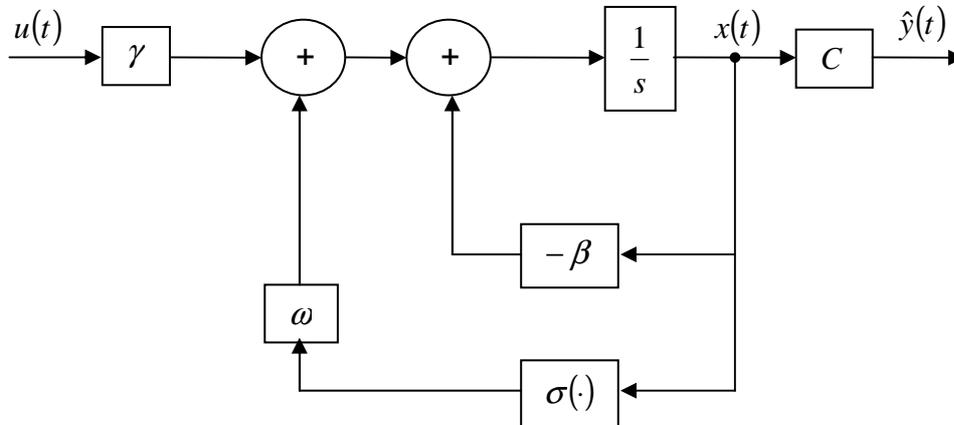


Figura 6-2: Diagrama de la red neuronal dinámica.

El estado de la DNN puede entonces considerarse dividido en los estados de salida $x_o \in \mathfrak{R}^p$ y los estados escondidos $x_h \in \mathfrak{R}^{n-p}$:

$$x = \begin{bmatrix} x_o \\ x_h \end{bmatrix} \quad (6-3)$$

6.3 Entrenamiento de la DNN

Para el ajuste de los parámetros de la DNN se dispone de una serie de datos obtenidos del sistema real que va a ser modelado. Supongamos un conjunto de entrenamiento compuesto por N pares de entrada-salida y tiempo de muestreo t_s :

$$\mathbf{Z}_N = [y(t_k), u(t_k)]_{k=1, \dots, N} \quad (6-4)$$

donde $y \in \mathfrak{R}^p$ es la salida medida, $u \in \mathfrak{R}^p$ es la entrada y k un índice temporal.

El problema de entrenamiento puede ser formulado como un problema de optimización. Sea $\theta \in \mathfrak{R}^{n_\theta}$ el vector de decisión, donde n_θ es el número de parámetros que se van a estimar. Si escogemos la función de costo dada por el error cuadrático medio,

$$V_N(\theta, \mathbf{Z}_N) = \frac{1}{2N} \sum_{k=1}^N [y(t_k) - \hat{y}(t_k | \theta)]^2 \quad (6-5)$$

donde $\hat{y}(t_k | \theta)$ es la salida de la DNN definida en (6-2), el problema de identificación no lineal puede ser descrito por el siguiente problema de optimización sin restricciones:

$$\min_{\theta} V_N(\theta, \mathbf{Z}_N) \quad (6-6)$$

En [Bec02] se propone la inclusión de los valores iniciales de los estados escondidos $x_h(t_0)$ en el vector de decisión θ formado como sigue:

$$\theta = \begin{bmatrix} \beta_d \\ \text{vec}(\omega) \\ \text{vec}(\gamma) \\ x_h(t_0) \end{bmatrix} \quad (6-7)$$

donde β_d es el vector columna de los elementos diagonales de la matriz β y $\text{vec}(\cdot)$ es un vector formado por las columnas de la matriz argumento (\cdot) .

De esta forma se evita una incorrecta identificación de los parámetros de la DNN. La inicialización de los valores de los estados escondidos a cero (tal y como se hace habitualmente) conduce a un reajuste de los parámetros de la red, para tratar de igualar las derivadas de los estados de salida de la red a las derivadas obtenidas experimentalmente. Esto conduce a la aparición de sesgo (bias) en dichos parámetros y, consecuentemente, a una mala identificación del sistema.

Una variante del problema de identificación, que ayuda al entrenamiento de la DNN, es escoger la siguiente función de costo modificada

$$V'_N(\theta, Z_N) = \frac{1}{2N} \sum_{k=1}^N \left\{ [y(t_k) - \hat{y}(t_k|\theta)]^2 + \mu |x_h(t_0)| \right\} \quad (6-8)$$

donde μ es una constante positiva que pesa los valores iniciales de los estados escondidos. De esta forma se evita que dichos valores sean elevados.

Una técnica eficaz que evita que el problema de optimización alcance mínimos locales es comenzar el entrenamiento desde distintas soluciones iniciales, escogidas de forma aleatoria. Otra solución alternativa que impide el estancamiento en un mínimo local es el uso de algoritmos genéticos, pero el costo computacional de los mismos es muy elevado.

6.4 Inicialización de la DNN

Para la validación de la DNN entrenada se emplea un conjunto de pares de entrada-salida diferente al de entrenamiento. Los valores iniciales de los estados escondidos obtenidos en entrenamiento no son válidos para este nuevo conjunto, por lo que se recurre a una estrategia para recalcular los mismos, consistente en inicializar los estados de las neuronas de salida con los valores medidos para la salida real del sistema,

$$x_0(t_0) = y(t_0) \quad (6-9)$$

y los estados de las neuronas escondidas resolviendo el siguiente problema de optimización:

$$x_h(t_0) = \arg \min_{x_h(t_0)} (\dot{y}(t_0) - \dot{x}_0(t_0))^2 \quad (6-10)$$

donde $\dot{y}(t_0)$ puede ser calculado de los datos obtenidos del sistema real y $\dot{x}_0(t_0)$ de la ecuación (6-1), una vez conocidos $x_0(t_0)$, $x_h(t_0)$, β , ω y γ . De esta forma, se asegura que las derivadas de las salidas de la DNN, $\dot{x}_0(t_0)$, son aproximadamente iguales al valor real $\dot{y}(t_0)$.

En el siguiente apartado se aplicarán estas técnicas de identificación al caso de un robot estático. Tal y como se ha comentado, la obtención de un modelo empírico resulta en ocasiones más adecuado que la utilización de un modelo matemático, ya que modela mejor ciertos comportamientos dinámicos que de la otra forma resultan muy complejos de modelar: la presencia de fricciones, la existencia de holguras o comportamientos histeréticos en los actuadores, las limitaciones mecánicas o eléctricas, etc. Además, evita el caso de existencia de incertidumbres en los parámetros dinámicos del sistema.

6.5 Identificación de un sistema robótico de piernas caminadoras

En este apartado se muestran los resultados de identificación de un sistema robótico en particular mediante el uso de DNN y la técnica de inicialización del estado de la red antes expuesta.

En primer lugar, se describe el sistema que se va a identificar, un robot compuesto por una pierna de dos grados de libertad. A continuación, se exponen los resultados de entrenamiento de la red neuronal definida para la identificación de la dinámica de dicho sistema, así como los resultados de validación utilizando la técnica de identificación de las neuronas de la red.

Estos resultados verifican el buen funcionamiento de la DNN como modelo neuronal, lo que permitirá su uso en controladores basados en modelo, con el consiguiente ahorro en coste computacional.

6.5.1 Descripción del sistema

El sistema real sobre el que se aplica este método de identificación mediante DNN, haciendo uso de los aspectos de entrenamiento y validación expuestos en el apartado anterior, es la pierna robótica mostrada en la figura 6-3 (el sistema completo consta de un par de piernas, totalmente iguales e independientes en cuanto a su actuación, por lo que se opta por hacer los experimentos de identificación en una de ellas).

Dicho sistema está compuesto por dos elementos, muslo y parte baja de la rodilla, accionados por sendos motores de corriente continua situados en las articulaciones de revolución, cadera y rodilla, respectivamente. El elemento final de la pierna es el pie, pero actualmente el sistema no dispone de motor para accionar dicha articulación, por lo que el sistema cuenta únicamente con dos grados de libertad.

Para la medición de los ángulos de las articulaciones se dispone de sendos potenciómetros situados en la parte interior de la pierna. La medida de las velocidades se ha realizado por diferenciación de las posiciones. Sin embargo, el sistema posee un diseño electrónico, basado en un convertidor frecuencia-voltaje, capaz de dar las velocidades de los elementos de la pierna.

Una comparación entre las velocidades medidas por uno y otro método muestra que son iguales salvo para velocidades pequeñas, donde el circuito electrónico presenta algunos problemas. Por esta razón se ha escogido la diferenciación como base para el cálculo de las velocidades.

Aún cuando pareciese no muy costoso obtener el modelo teórico de este sistema, ya que corresponde inicialmente al de una cadena articulada de dos elementos unida por articulaciones de revolución, la existencia de no

linealidades difíciles de modelar, además de las incertidumbres en los parámetros dinámicos del sistema, hace que la identificación mediante un modelo neuronal sea más apropiada para la obtención del modelo del sistema real en estudio. En particular, aparecen en el sistema distintos tipos de fricciones, zonas muertas y también algunas holguras en las articulaciones. Por esta razón se persigue obtener el modelo de forma empírica y no teórica.



Figura 6-3: Diferentes vistas de las piernas andadoras.

6.5.2 Resultados de identificación

El objetivo de este experimento es la obtención de un modelo neuronal mediante identificación del sistema robótico de dos articulaciones explicado anteriormente utilizando una DNN, de forma que contenga las no linealidades presentes en el sistema y que son difíciles de modelar teóricamente.

Se han obtenido de la planta real sendos conjuntos de 1595 pares de entrada-salida para entrenamiento y validación. Se han escogido las entradas en forma de sinusoides de distintas amplitudes y frecuencias, cubriendo, tanto para las entradas como para las salidas obtenidas, un amplio rango de valores en los que puede moverse el sistema real. Dichas señales han sido escaladas convenientemente. El problema de optimización fue resuelto utilizando la función *fminunc* de la toolbox de optimización de Matlab® versión 6.0 R.12, la cual utiliza el algoritmo de optimización sin restricciones Quasi-Newton.

La DNN provee como salidas los ángulos de las articulaciones, por lo que sus dos primeras neuronas se escogen como salidas, quedando el resto como neuronas escondidas.

Inicialmente cabe pensar que la estructura ideal para la DNN está constituida, como mínimo, por cuatro neuronas, ya que el sistema tiene cuatro estados: posiciones y velocidades de ambas articulaciones. Aun así, se ha hecho un estudio del error cuadrático medio para estructuras conteniendo entre tres y ocho neuronas, obteniéndose el mejor resultado para una estructura de cuatro neuronas, confirmando así la predicción anterior.

Los resultados obtenidos para ambas articulaciones, en el caso de entrenamiento, son los mostrados en la figura 6-4 y en la figura 6-5 respectivamente.

Como se puede observar, el comportamiento de la DNN es muy similar al del sistema real. Las situaciones en las que se aprecian no linealidades son bien resueltas por la red. El incluir los estados iniciales de las neuronas escondidas en el vector de entrenamiento evita la aparición de sesgo en los parámetros. Además, se observa que el entrenamiento se realiza en un tiempo menor.

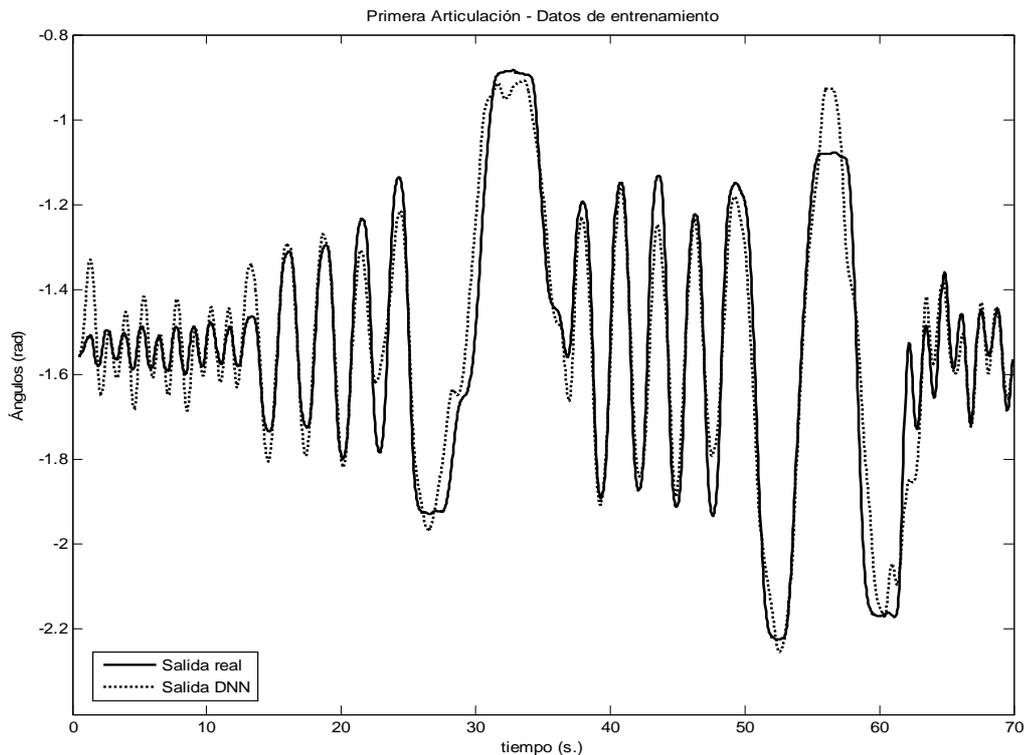


Figura 6-4: Ángulos medidos para la primera articulación y salida de la DNN para el caso de los datos de entrenamiento.

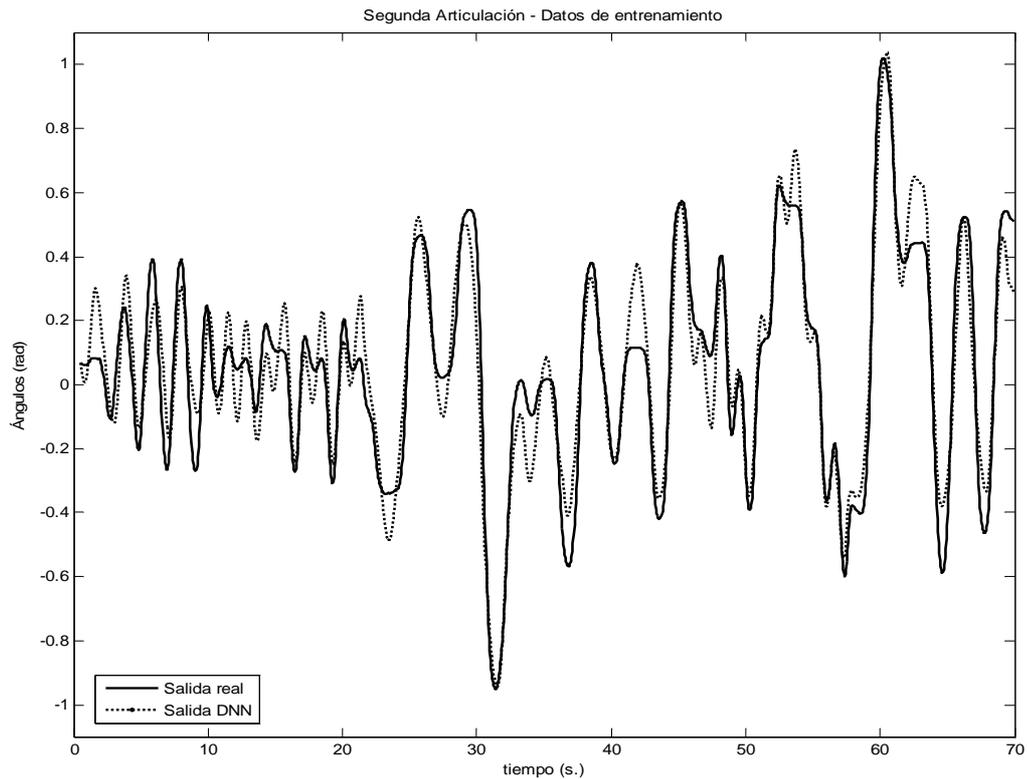


Figura 6-5: Ángulos medidos para la segunda articulación y salida de la DNN para el caso de los datos de entrenamiento.

Los valores de los parámetros obtenidos son los siguientes:

$$\beta_d = \begin{bmatrix} 5.0856 \\ 2.9197 \\ -3.3450 \\ 2.0491 \end{bmatrix} \quad (6-11)$$

$$\omega = \begin{bmatrix} 3.6484 & -4.1867 & 6.1201 & -4.0089 \\ -1.2547 & 1.7313 & -3.2064 & -4.3606 \\ -0.4706 & -0.9042 & -5.7358 & -2.5743 \\ 0.7214 & 0.3866 & 0.5750 & 1.7928 \end{bmatrix} \quad (6-12)$$

$$\gamma = \begin{bmatrix} -0.3534 & 0.0060 \\ -0.3835 & -0.7599 \\ 0.0249 & -1.0831 \\ -0.3994 & -0.8308 \end{bmatrix} \quad (6-13)$$

Para el caso de validación, se obtienen los resultados mostrados en la figura 6-6 y en la figura 6-7. Se puede observar el buen comportamiento de la DNN respecto a los datos reales.

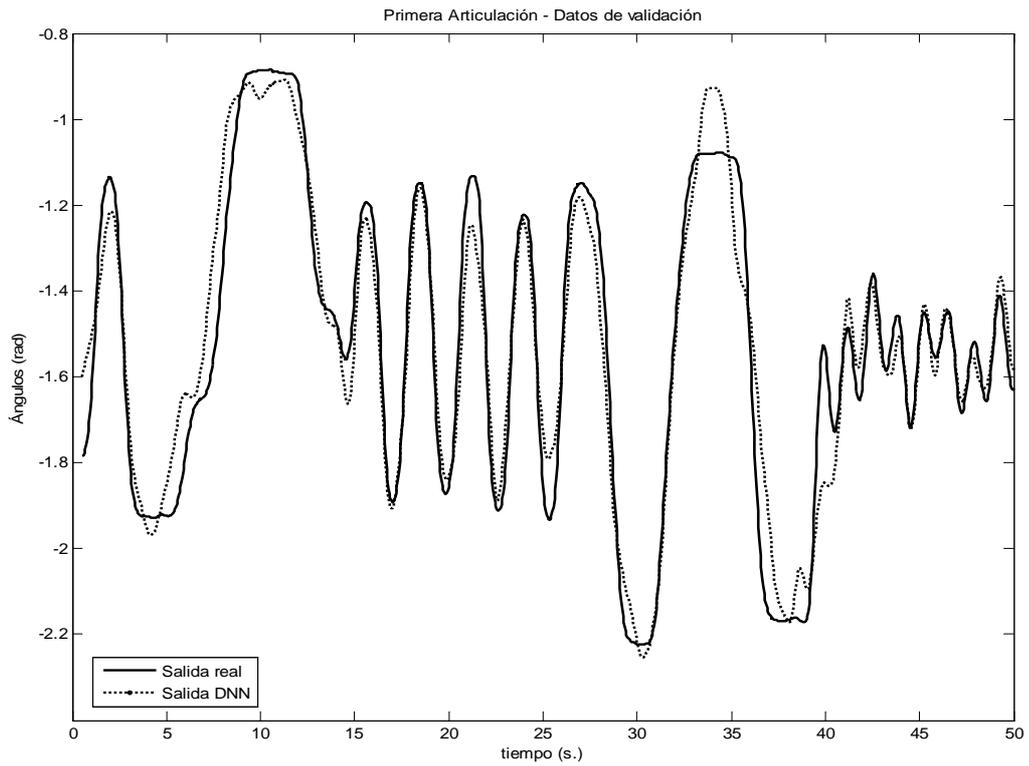


Figura 6-6: Ángulos de la primera articulación y salida de la DNN para el caso de los datos de validación.

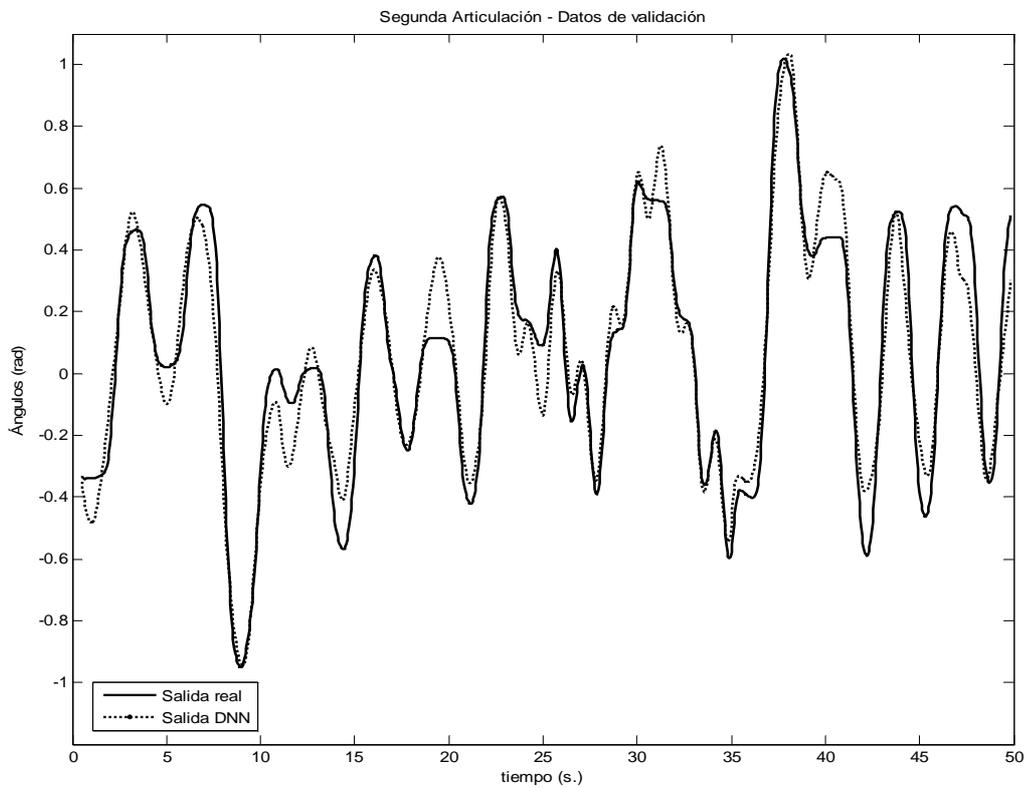


Figura 6-7: Ángulos de la segunda articulación y salida de la DNN para el caso de los datos de validación.

La elección de los valores de los estados iniciales de las neuronas escondidas es crucial para este caso, ya que la resolución previa del problema de optimización que minimiza el error en las velocidades hace que la DNN se sitúe inicialmente en un estado que le permite seguir eficazmente el comportamiento del sistema real.

Luego, como conclusión, se puede decir que una DNN de cuatro estados identifica bastante bien el comportamiento real del sistema, incluidos efectos de holguras y fricciones tan difíciles de modelar matemáticamente.

En el siguiente capítulo se aplicará esta técnica de identificación mediante DNN a la obtención de un modelo neuronal del manipulador que pueda ser empleado por los controladores para rebajar el coste computacional de los mismos. En concreto, el modelo empírico puede utilizarse para realizar las predicciones del comportamiento dinámico del robot necesarias para la evaluación de las restricciones por parte de los algoritmos IAPC y RIAPC. De esta forma se conseguirá evitar gran parte del cálculo que hay que hacer con el modelo matemático estimado del sistema. Además, el modelo neuronal será capaz de captar dinámicas no modeladas por el modelo estimado.

Reducción del costo computacional de los controladores usando DNN

La obtención de un modelo empírico fiable resulta de gran importancia a la hora del diseño de un controlador basado en modelo, principalmente por dos razones. La primera es que el modelo empírico resulta más sencillo que el modelo matemático equivalente para los sistemas complejos que presentan no linealidades o acoplamientos, como es el caso de los robots estáticos. El modelo resultante involucra un menor número de operaciones y puede ser utilizado con facilidad por los distintos algoritmos de control (para obtener la dinámica inversa, predicciones en la salida, etc.). La segunda razón es que el modelo empírico absorbe la dinámica del sistema que no es contemplada por los modelos matemáticos, como puede ser la presencia de fricciones, holguras en los accionamientos, limitaciones mecánicas o electrónicas, comportamientos histeréticos, etc. En este capítulo se incorpora esta técnica de identificación mediante el modelo empírico neuronal explicado en el capítulo 6 al controlador RIAPC diseñado para el robot manipulador. De esta forma se rebaja el costo computacional de dicho controlador. Se presentan resultados de entrenamiento y validación que prueban la correcta identificación del sistema con este tipo de redes neuronales. A continuación se incorpora el modelo neuronal al controlador RIAPC y se muestran los resultados obtenidos, que no varían sustancialmente respecto a los obtenidos en el capítulo 5, ofreciendo el controlador basado en redes neuronales un mejor rendimiento computacional.

7.1 Introducción

En el capítulo anterior se describe una técnica de entrenamiento y validación de una red neuronal dinámica (DNN) para la obtención de un modelo empírico de un robot estático. La importancia de esta técnica radica en que el modelo así obtenido contiene información del sistema que, por métodos

matemáticos, es bastante difícil de evaluar, como puede ser las fricciones mecánicas, las holguras de las articulaciones, etc. Además, las incertidumbres presentes en el modelo están incluidas en el modelo identificado, sin necesidad de tenerlas en cuenta en los algoritmos de control diseñados.

La desventaja de esta técnica es la necesidad de obtener un nuevo modelo empírico por cada cambio en la situación dinámica en la que se encuentre el robot. Así, habría que obtener un modelo para cada una de las cargas que puede manipular el robot.

Una solución a este inconveniente se aporta en [Car95], donde se entrenan varias redes para situaciones dinámicas distintas (por ejemplo, distintas masas de carga) y luego se combinan linealmente para obtener una solución para cualquier situación dinámica afectada de incertidumbre que se pudiera representar. En este trabajo se plantea esta solución con redes estáticas, para las cuales, además, se aporta un análisis de estabilidad en el que se incluyen los posibles errores de aprendizaje de las redes. Pero una idea similar se puede plantear para las DNN.

La novedad que se plantea en la identificación del sistema utilizando DNN, es la resolución de un problema de optimización previo para la inicialización de los estados de las neuronas escondidas. La idea consiste en medir las primeras derivadas en el instante inicial de los estados (o velocidades iniciales) y las aportadas por la red, de forma que se inicialicen los parámetros de la red para igualarlas tanto como sea posible. De esta forma se evita que en el entrenamiento aparezcan sesgos que intenten corregir esta desviación inicial, con lo que la identificación es más precisa.

En el presente capítulo se incorpora el modelo neuronal obtenido del robot manipulador PUMA al controlador RIAPC con el objetivo de rebajar el costo computacional del controlador. El modelo neuronal se utilizará en las predicciones del comportamiento dinámico del controlador que se emplean para la evaluación de las restricciones dentro de la resolución del problema de programación lineal que se debe resolver para obtener la ley de control del RIAPC (ver capítulos 4 y 5).

Se muestran resultados de entrenamiento y validación para verificar el modelo empírico obtenido del manipulador para el PUMA 560. Finalmente se incorpora esta técnica al controlador RIAPC y se muestran los resultados obtenidos para los mismos supuestos que los experimentos del capítulo 5.

7.2 Identificación del manipulador PUMA 560

En esta sección se presentan resultados de identificación obtenidos para el robot PUMA 560 empleado en esta tesis. Se hacen uso de las técnicas de entrenamiento y validación de DNN expuestas en el capítulo anterior.

Se han obtenido de la planta real sendos conjuntos de 2501 pares de entrada-salida para cada una de las articulaciones, de los cuales 714 se emplean para entrenamiento y 1044 para validación. No se utiliza el total de los datos pues se han eliminado zonas en las que no hay movimiento alguno del robot. Para la obtención de los datos de entrada y salida se han realizado pruebas en lazo cerrado con trayectorias variables en el tiempo, las cuales cubren el rango de ángulos y velocidades que después se requerirán en los experimentos de control.

Tal y como se hizo con las piernas caminadoras del capítulo anterior, estas señales han sido escaladas convenientemente y el problema de optimización fue resuelto utilizando la función *fminunc* de la toolbox de optimización de Matlab®, la cual utiliza el algoritmo de optimización sin restricciones Quasi-Newton.

Se realizan entrenamientos con redes DNN de 3 entradas (las entradas a cada una de las 3 primeras articulaciones del manipulador) y 3 salidas (las posiciones de dichas articulaciones). En este trabajo nos centramos en la identificación y el control de las tres primeras articulaciones, obviando las articulaciones presentes en la muñeca del manipulador. Para dar con la mejor estructura de red, se proponen diferentes experimentos de identificación con redes que van desde las 3 neuronas hasta las 10 neuronas. Se estudia en cada caso el error cuadrático medio obtenido con las diferentes redes, tanto para los pares entrada-salida de entrenamiento como de validación, para obtener una medida fiable de la bondad en la identificación de la dinámica del manipulador.

La mejor identificación se obtiene con una estructura de red de 5 neuronas, las 3 de salida y 2 escondidas. En ese caso, la evolución de la función de error cuadrático¹ se muestra en la figura 7-1. La elección de los pesos iniciales es aleatoria y, tras unas 6000 evaluaciones de la función error cuadrático (200 iteraciones del algoritmo de minimización), se tiene un modelo neuronal que aproxima la respuesta real del robot.

Se optó por no proseguir el entrenamiento de la red ya que, en ese caso, la red neuronal deja de generalizar para los casos de entrada y salida no contemplados en los pares de entrenamiento.

Los valores de los parámetros obtenidos son los siguientes:

$$\beta_d = \begin{bmatrix} 0.5848 \\ 0.5691 \\ 0.6337 \\ 0.3958 \\ 0.9100 \end{bmatrix} \quad (7-1)$$

¹ Función que se minimiza durante el entrenamiento de la red neuronal. Evalúa la suma de los cuadrados de los errores entre la salida de la red y las posiciones reales.

$$\omega = \begin{bmatrix} 0.5077 & 0.7855 & 0.3296 & -0.0097 & 0.3698 \\ -0.5585 & 0.3071 & 0.5667 & 0.2870 & -0.1374 \\ 0.7231 & 0.7879 & 0.3528 & -0.3103 & -0.6205 \\ 1.1762 & 0.6851 & 0.4669 & 0.6980 & 0.7364 \\ 0.6138 & -0.1999 & -0.0869 & 0.7375 & 0.1887 \end{bmatrix} \quad (7-2)$$

$$\gamma = \begin{bmatrix} 0.6821 & 0.1950 & 0.3651 \\ 0.2074 & 0.1970 & 0.3748 \\ -0.0659 & 0.4586 & 0.1392 \\ 0.4185 & 0.5066 & 0.0971 \\ 0.3263 & 0.4227 & 0.5515 \end{bmatrix} \quad (7-3)$$

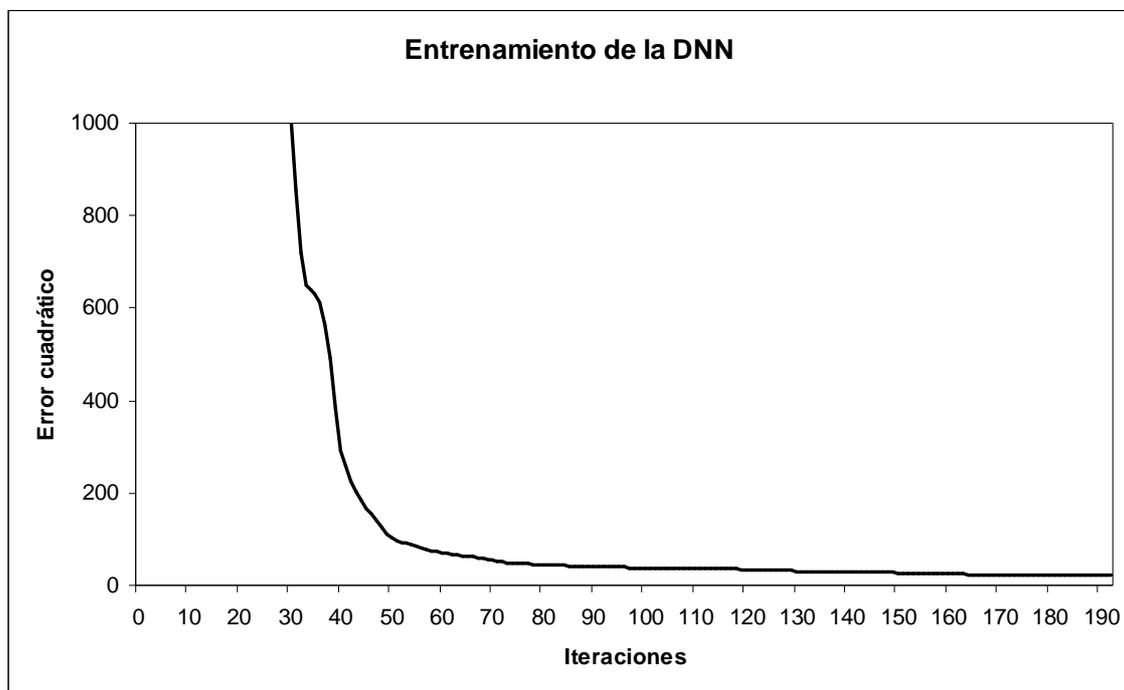


Figura 7-1: Evolución de la función error cuadrático durante el entrenamiento de la red neuronal dinámica de 5 neuronas utilizada como modelo neuronal del manipulador.

Los resultados obtenidos para todas las articulaciones, en el caso de entrenamiento, son los mostrados en la figura 7-2. respectivamente. Como se puede observar, el comportamiento de la DNN es muy similar al del sistema real. Las oscilaciones que se aprecian en las salidas de la red son producto de los errores que aún se tienen al finalizar la fase de entrenamiento. Sin embargo, esto no es relevante ya que las predicciones se harán como máximo hasta 15 ó 20 periodos de muestreo hacia delante, razón por la cual lo más importante será los primeros valores aportados por la red neuronal. Este mismo razonamiento se aplica a los resultados obtenidos para validación, los cuales se muestran en la figura 7-3.

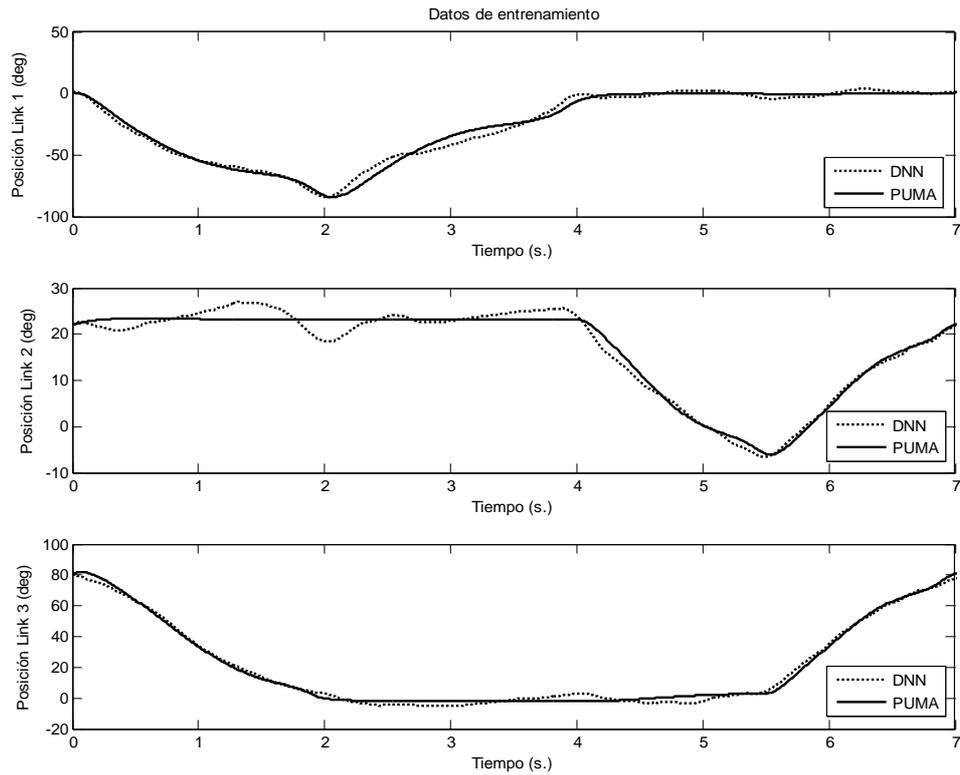


Figura 7-2: Entrenamiento de una DNN para las tres primeras articulaciones del manipulador PUMA-560.

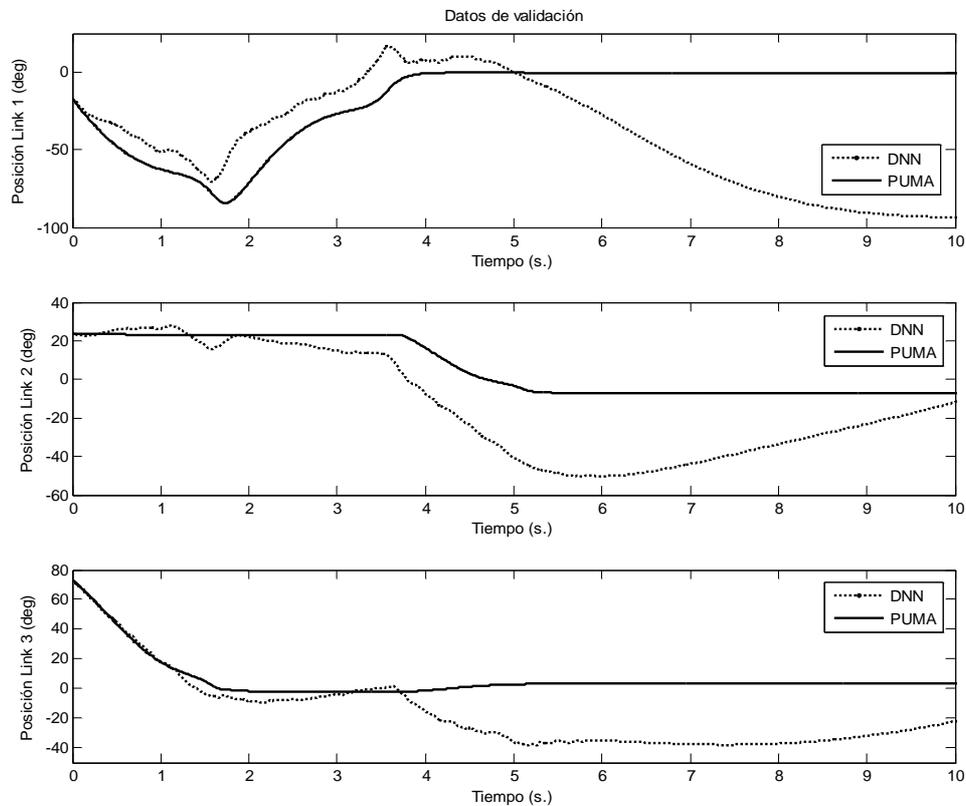


Figura 7-3: Validación de una DNN para las tres primeras articulaciones del manipulador PUMA-560.

Los buenos resultados obtenidos con este modelo dan pie a su incorporación al controlador predictivo RIAPC, diseñado en el capítulo 5 de esta tesis, con el objetivo de reducir el costo computacional de aquel algoritmo de control.

7.3 Incorporación del modelo neuronal al controlador RIAPC: Estrategia NRIAPC.

El principal requerimiento de cálculo por parte del algoritmo RIAPC, en cuanto al algoritmo de interpolación de las soluciones LQ y ML para obtener una solución factible, consiste en la evaluación de las predicciones del estado y las entradas del manipulador. Estas predicciones se realizan mediante las ecuaciones:

$$\mathbf{u}_j = -(1-\alpha)\mathbf{K}_{LQ}\boldsymbol{\varphi}_{LQ}^j\tilde{\mathbf{x}} - \alpha\mathbf{K}_{ML}\boldsymbol{\varphi}_{ML}^j\tilde{\mathbf{x}} \quad (7-4)$$

$$\tilde{\mathbf{x}}_j = (1-\alpha)\boldsymbol{\varphi}_{LQ}^j\tilde{\mathbf{x}} + \alpha\boldsymbol{\varphi}_{ML}^j\tilde{\mathbf{x}} \quad (7-5)$$

con $j=0,\dots,n_c$, siendo n_c el horizonte de control, α el parámetro de interpolación, $\boldsymbol{\varphi}_{LQ} = \mathbf{A} - \mathbf{B}\mathbf{K}_{LQ}$ y $\boldsymbol{\varphi}_{ML} = \mathbf{A} - \mathbf{B}\mathbf{K}_{ML}$, \mathbf{A} y \mathbf{B} las matrices del sistema linealizado y \mathbf{K}_{LQ} y \mathbf{K}_{ML} las ganancias de las soluciones óptima y “mean level” respectivamente, tal y como se vio en el capítulo 5 de esta tesis.

Como se puede apreciar, se debe acometer un cálculo en cada instante de tiempo que involucra la potencia j -ésima de una matriz $2n \times 2n$, donde n es el número de grados de libertad del manipulador. Es en este paso el que exige gran parte del costo computacional del algoritmo. De ahí la importancia de contar con un modelo empírico de más sencilla evaluación para realizar las predicciones.

En el caso de la red neuronal las predicciones del estado se obtienen de la resolución de la ecuación de la red dinámica (6-2):

$$\dot{\mathbf{x}} = -\boldsymbol{\beta} \cdot \mathbf{x} + \boldsymbol{\omega} \cdot \sigma(\mathbf{x}) + \boldsymbol{\gamma} \cdot \mathbf{u} \quad (7-6)$$

donde las matrices involucradas son el resultado de la identificación (7-1), (7-2) y (7-3). La entrada \mathbf{u} se sustituye por la ley de control interpolada:

$$\mathbf{u} = -(1-\alpha)\mathbf{K}_{LQ}\mathbf{x} - \alpha\mathbf{K}_{ML}\mathbf{x} \quad (7-7)$$

y se realiza la predicción del estado hasta n_c instantes de tiempo hacia adelante. Para ello se hace una aproximación a la expresión discreta equivalente de la ecuación (7-6). La resolución de esta ecuación es sensiblemente menos costosa que las expresiones que aparecen en (7-4) y (7-

5). Para el cálculo de las ganancias \mathbf{K}_{LQ} y \mathbf{K}_{ML} se debe hacer uso de las matrices \mathbf{A} y \mathbf{B} del sistema linealizado, por lo que el algoritmo de linealización exacta sigue siendo necesario.

Por todo ello, el esquema resultante del controlador, denominado NRIAPC (*Neural Robust Interpolated based Adaptive Perturbation Control*) es el que se muestra en la figura 7-4. El modelo neuronal (7-6) provee las predicciones al controlador RIAPC, el cual calcula la desviación de las entradas al sistema respecto a las entradas nominales calculadas mediante el controlador por dinámica inversa, de forma que se rechacen eficientemente las incertidumbres y perturbaciones presentes en el sistema.

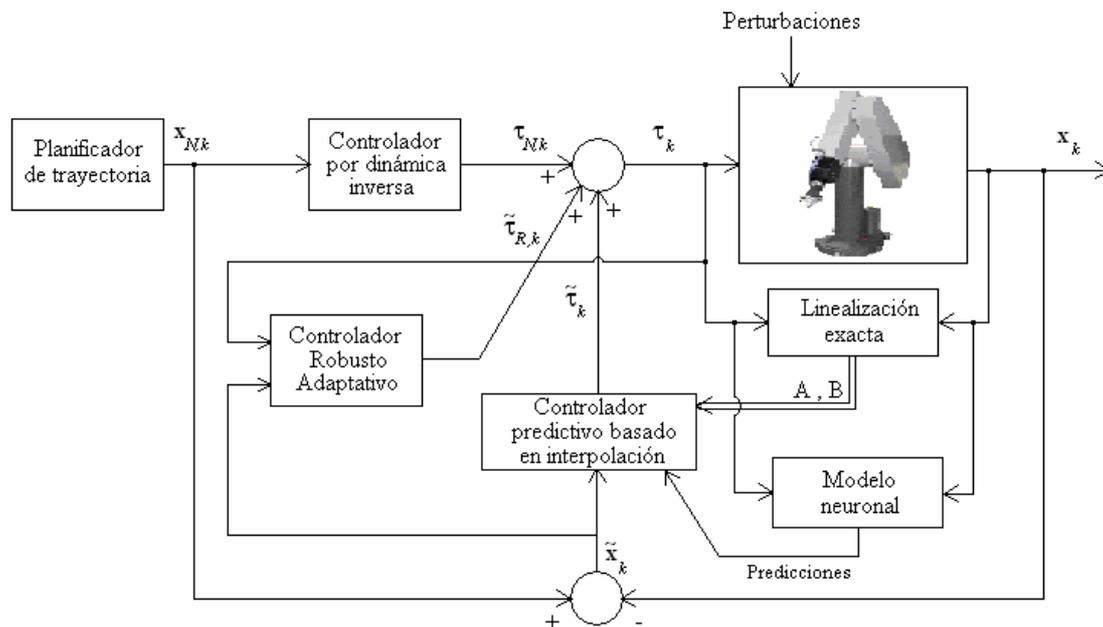


Figura 7-4: Esquema de control NRIAPC: Controlador RIAPC con predicción neuronal del estado futuro del sistema para el bloque de control predictivo basado en interpolación.

En el siguiente apartado se presentan resultados obtenidos con este esquema de control para el manipulador PUMA-560. Dichos resultados se comparan con los obtenidos con la estrategia RIAPC, donde se emplea el modelo matemático estimado del robot para realizar las predicciones de su comportamiento futuro.

7.4 Resultados con el controlador NRIAPC

Para probar la eficiencia del algoritmo, se presentan a continuación los resultados de un experimento de control sobre el manipulador PUMA-560, donde se comparan los resultados que se obtienen con la técnica RIAPC y con la técnica NRIAPC, basada en redes neuronales.

Experimento 7.1

Sistema:	Manipulador PUMA-560.
Controlador:	RIAPC y NRIAPC.
* Acción LQ:	$\mathbf{V} = \text{diag}([4.8 \ 1.2 \ 4.8 \ 1.2 \ 4.8 \ 1.2 \ 3.2 \ 0.8 \ 3.2 \ 0.8 \ 3.2 \ 0.8])$ $\mathbf{W} = \text{diag}(0.001 * [25 \ 25 \ 25 \ 5 \ 5 \ 5]).$
* Acción ML:	$\mathbf{V}' = 0.1 \cdot \mathbf{V}$ $\mathbf{W}' = 1e4 \cdot \mathbf{W}$
* Acción robusta:	$\varepsilon = 1, \mathbf{Q} = \text{diag}([0 \ 0 \ 0 \ 0 \ 0 \ 0,1 \ 1 \ 1 \ 1 \ 1 \ 1]).$
* Adaptación de parámetro de acción robusta:	$\mathbf{R}_{ad} = h \cdot \text{diag}([25 \ 25 \ 25 \ 25 \ 25 \ 25])$ $\mathbf{Q}_{ad} = 100 \cdot \text{diag}([1 \ 3.75 \ 2.5 \ 2.5 \ 2.5 \ 2.5, 2.5 \ 6 \ 5 \ 6 \ 6 \ 6])$ $\gamma = 1, h = 0.001.$
* Modelo del manipulador:	Modelo de manipulador de [Lew04] en presencia de incertidumbres paramétricas (un 15% de desviación máxima en las masas y longitudes de las diferentes articulaciones).
* Restricciones:	$\boldsymbol{\tau}_M = [75 \ 36.5 \ 11 \ 10 \ 10 \ 10]$ $\tilde{\mathbf{x}}_{\max}^+ = [15 \ 15 \ 15 \ 5 \ 5 \ 5] * \frac{\pi}{180}, \tilde{\mathbf{x}}_{\max}^- = -\tilde{\mathbf{x}}_{\max}^+$
* Trayectoria de referencia:	- Trayectoria en el espacio de las articulaciones generada a través de la unión de cuatro puntos en el espacio de tareas unidos mediante polinomios de séptimo orden. Dicha trayectoria es recorrida dos veces a lo largo del movimiento.

En la figura 7-5 se muestran los resultados obtenidos para la primera articulación con ambos controladores, mientras que en la figura 7-6 se muestran los resultados para la articulación 3.

Como se puede comprobar, salvo por algunas diferencias en los errores de seguimiento, el comportamiento es bastante similar, por lo que la ejecución de las trayectorias por parte del manipulador es casi idéntica. De nuevo, se consiguen salvar las situaciones de saturación de las entradas debido a la acción del controlador predictivo diseñado para tal fin.

La razón de este comportamiento es la bondad del modelo neuronal respecto al modelo real del manipulador, sobre todo en los instantes de tiempo

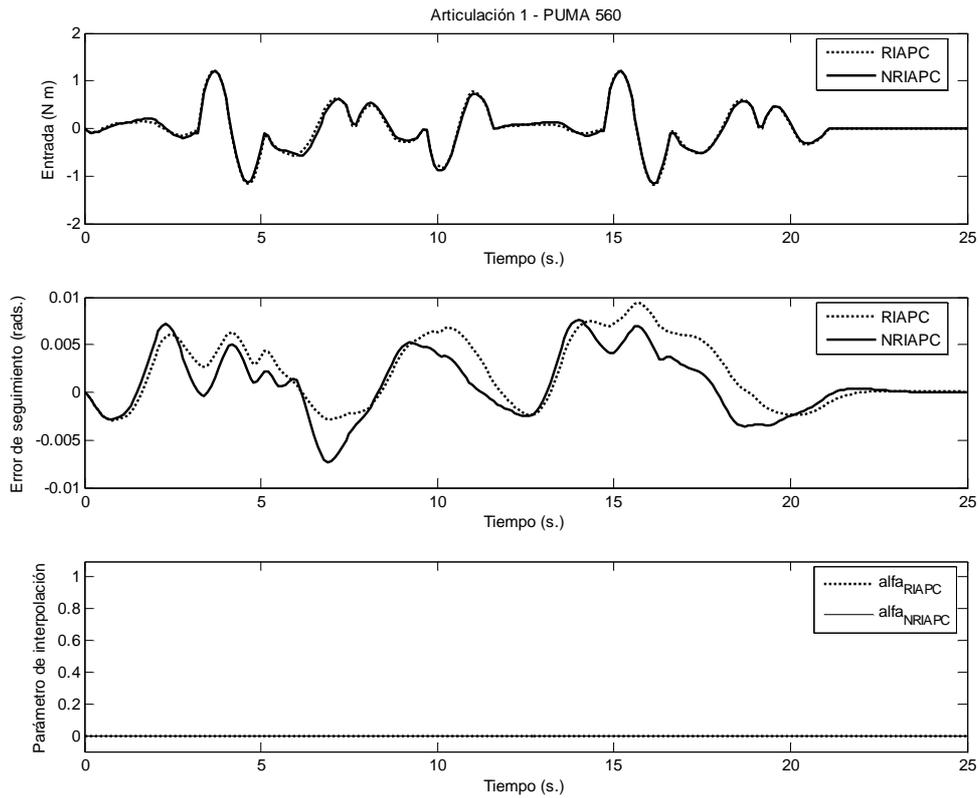


Figura 7-5: Evolución de la entrada, error de seguimiento y parámetro de interpolación para la articulación 1.

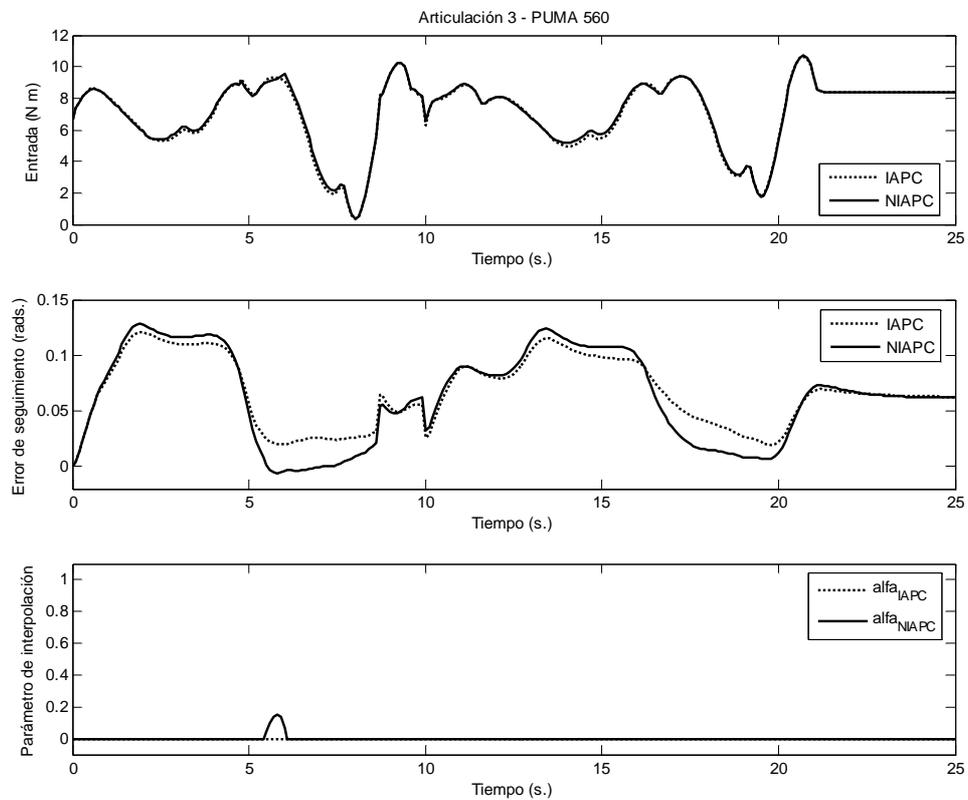


Figura 7-6: Evolución de la entrada, error de seguimiento y parámetro de interpolación para la articulación 3.

cercanos al instante actual², donde las predicciones son casi idénticas a las obtenidas con el modelo teórico. Este buen resultado es debido al algoritmo de inicialización del estado de la red, principalmente en lo referido al estado de las neuronas escondidas de la DNN.

Además, se ha conseguido rebajar bastante el costo computacional de la estrategia de control. Ahora, la mayor parte del cálculo vuelve a estar centrado en la resolución del problema de programación lineal del algoritmo de interpolación y no en la evaluación de las restricciones basadas en las predicciones del comportamiento del manipulador.

Estos buenos resultados permiten concluir en la propuesta de integrar un modelo empírico basado en redes neuronales dinámicas en los algoritmos de control propuestos, con lo que se consigue añadir la propiedad de eficiencia computacional a las propiedades ya conseguidas, como son el rechazo a incertidumbres del modelo, perturbaciones y cumplimiento de las restricciones del sistema.

² Los horizontes de predicción y control utilizados por el controlador son de $n_p = n_c = 10$ periodos de muestreo. Por tanto, la predicción temporal es de 0.01 segundos.

Conclusiones y líneas abiertas

En esta tesis se ha llevado a cabo el diseño y la implementación de diversas técnicas de control para manipuladores robóticos, que impulsan los diseños de control hasta ahora propuestos, añadiendo mejores prestaciones en términos de robustez y eficiencia computacional. Las técnicas de control para manipuladores existentes en la literatura, aunque resuelven problemas como el seguimiento eficaz de trayectorias en presencia de perturbaciones e incertidumbres, no tratan con eficacia problemas tales como la variabilidad de las condiciones dinámicas del robot, la ejecución satisfactoria de trayectorias para perturbaciones totalmente desconocidas, o las ligaduras a las que está sujeto el robot, tales como valores máximos en las entradas de los actuadores o valores límite de las posiciones y velocidades de las articulaciones. Además, es deseable el diseño de controladores que proporcionen una solución global a estos problemas, y que sean computacionalmente eficientes para su aplicación en tiempo real.

Es por esta razón que en esta tesis se proponen diversas técnicas de control, basadas en algoritmos adaptativos, robustos y predictivos, que conducen a una mejor consecución de los movimientos por parte del manipulador, aún en presencia de incertidumbres en el modelo del mismo, perturbaciones y teniendo en cuenta las ligaduras del sistema.

Se exponen a continuación las conclusiones a las que se ha llegado en esta tesis:

PRINCIPALES PROBLEMAS EN EL CONTROL DE MANIPULADORES

- A pesar de que las técnicas de control existentes para manipuladores robóticos están bien consolidadas y presentan buen rendimiento en la mayoría de situaciones, existen ciertas aplicaciones para las cuales su rendimiento puede ser notablemente mejorado, las cuales son de indudable interés en la solución del problema de seguimiento.
- El primer punto donde se pueden establecer mejoras sobre las estrategias existentes es en el caso de incertidumbres en el modelo que, o bien no están acotadas, o bien no pueden ser acotadas. Las

estrategias clásicas de control de manipuladores para el caso de incertidumbres aplican técnicas procedentes del control H_∞ para dar robustez a las soluciones aportadas por el controlador. Estas técnicas se llevan a la práctica normalmente mediante un controlador PID al que se añade una entrada de control que aporta robustez al lazo cerrado. Esto es así por el bajo orden del controlador resultante y el bajo número de parámetros a sintonizar. En ese caso, la estabilidad del lazo cerrado se asegura si se pueden establecer cotas para las incertidumbres presentes en el sistema. Pero, en muchas ocasiones, el sistema sufre variaciones en su dinámica respecto al modelo considerado que no pueden ser acotadas, como es el caso de variaciones en la masa de carga que transporta el manipulador, o el desconocimiento de la misma. Otro caso se presenta cuando ciertos parámetros dinámicos del sistema cambian por el uso del mismo, como son holguras y rozamientos, lo que dificulta la tarea de encontrar las cotas de incertidumbres que precisan las técnicas de control robustas.

- El segundo punto a mejorar es el relacionado con las restricciones al movimiento del manipulador. La solución inmediata y más utilizada en los controladores existentes es la de generar trayectorias que no violen estas ligaduras, y lo suficientemente alejadas de las mismas para que las perturbaciones al sistema no provoquen su violación. Sin embargo, la variabilidad de la dinámica a la que puede estar sujeto el manipulador, y la indefinición de las perturbaciones que le pueden afectar, hacen que esta medida pueda resultar insuficiente.
 - Otro punto a mejorar es el costo computacional de las técnicas de control. Las técnicas de control empleadas habitualmente para resolver problemas tales como el rechazo a perturbaciones o incertidumbres en el modelo son técnicas de control adaptativo y robusto. Es importante tener en cuenta que dichos algoritmos requieren una cantidad de cómputo que se incrementa con la complejidad de estas técnicas. Si, además, se tienen en cuenta la evaluación continua de las restricciones del sistema, el costo computacional crece aún más. Sin embargo, para disponer de modelos discretos que describan fehacientemente la dinámica del manipulador se requieren períodos de muestreo pequeños. Así, la utilización de las ecuaciones dinámicas de tiempo continuo discretizadas es factible. Por estas razones, el bajo costo computacional de las técnicas de control es un punto crucial a la hora del diseño de dichas estrategias.
 - Otro punto susceptible de mejoras en el control de manipuladores es la integración de técnicas de control para resolver los problemas antes expuestos. Los pequeños tiempos de muestreo requeridos para realizar eficazmente las tareas de control, normalmente por debajo de la centésima de segundo, obligan al diseño de controladores simples, los cuales no dan una solución global para el rechazo a perturbaciones, presencia de incertidumbres y existencia de ligaduras.
-

ESTRATEGIA DE CONTROL PARA EL RECHAZO A PERTURBACIONES E INCERTIDUMBRES EN EL MODELO

- En este trabajo se propone un controlador robusto adaptativo (ARC) para el rechazo a perturbaciones. A diferencia de los controladores existentes, este controlador no requiere ninguna condición previa sobre acotación de incertidumbres presentes en el sistema o sobre las perturbaciones que afectan al mismo.
 - Los controladores robustos diseñados hasta la fecha son el resultado de la aplicación de una ley de control lineal, por ejemplo un controlador PD, a la que se suma una acción robusta, que se encarga de complementar la acción de control cuando existen diferencias entre el modelo y el sistema real. Estos controladores deben poseer cierta información previa para ajustar su acción, como son las cotas de incertidumbres o máximas desviaciones del modelo respecto al sistema real. Es por ello que se requiere de un estudio en simulación previo para la determinación de las cotas de incertidumbre y ajustar los parámetros del controlador.
 - En el controlador propuesto se aporta una mejora a este método, utilizando una acción adaptativa que tenga en cuenta la dinámica no modelada, así como las perturbaciones que afecten al sistema. Dicha acción se basa en la adaptación del parámetro de cota de incertidumbre de la acción robusta y proponiendo una función de costo que pese la contribución del error de seguimiento (producto de las incertidumbres y perturbaciones no predichas en el modelo) y de la entrada en el sistema (para evitar saturaciones que lleven al sistema a la violación de ligaduras, que pueden causar inestabilidad en el lazo cerrado).
 - Los resultados muestran la eficacia de este controlador en comparación con las estrategias clásicas de control robusto para manipuladores. Dichos resultados se han obtenido en tiempo real utilizando un robot PUMA de la serie 560 de Unimation. En ellos se compara la acción de un controlador PID clásico, del controlador robusto de Spong y de la estrategia propuesta (ARC), donde los errores de seguimiento, en presencia de incertidumbres no modeladas, se reducen notablemente con la técnica ideada.
 - Se aporta además una prueba de estabilidad de este algoritmo, donde se prueba la convergencia del mismo para cualquier rango de incertidumbre presente en el modelo, con una elección adecuada de las matrices de la función de costo utilizada para la ley de adaptación del controlador propuesto.
-

ESTRATEGIA DE CONTROL TENIENDO EN CUENTA LAS LIGADURAS DEL SISTEMA

- Se propone asimismo un controlador predictivo para solucionar el problema de existencia de ligaduras. Este tipo de estrategia de control incorpora en su formulación de forma natural a las ligaduras. Las ligaduras habituales presentes en manipuladores son las restricciones al movimiento de las articulaciones y los valores máximos de entrada que pueden suministrar los actuadores. La violación de estas ligaduras pueden causar inestabilidad en el lazo cerrado o bien un comportamiento ineficaz del problema de seguimiento. Para rebajar la carga computacional del algoritmo se propone una estrategia predictiva basada en interpolación, donde en cada instante se resuelve únicamente un problema de optimización lineal, de forma que la solución interpolada resultante sea una solución óptima que no viole las ligaduras.
- El esquema de control propuesto, el controlador de perturbación adaptativa basado en interpolación (IAPC), está basado en el esquema de perturbación adaptativa clásico, donde la ley de control es la suma de la acción de control nominal (resultado de la aplicación de la dinámica inversa) y la acción de control lineal resultado del controlador predictivo basado en interpolación.
- Los resultados obtenidos con esta estrategia, obtenidos en simulación para el mismo robot que en el caso anterior, muestran la eficacia de esta técnica de control, en comparación con otros controladores que no incorporan las ligaduras en su formulación. La no violación de las ligaduras minimiza los errores de seguimiento que se producen con las otras estrategias de control.

EFICIENCIA COMPUTACIONAL DE LOS CONTROLADORES PROPUESTOS MEJORADA POR EL USO DE MODELOS EMPÍRICOS BASADOS EN REDES NEURONALES DINÁMICAS

- Aunque en las distintas estrategias propuestas en esta tesis se ha tratado de diseñar técnicas de control computacionalmente eficientes, se puede rebajar aún más el costo computacional asociado a las mismas si se emplean modelos empíricos para evitar el cómputo del modelo no lineal del manipulador. Para este fin se han utilizado las redes neuronales dinámicas tipo Höpfield. Estas redes aprenden (en la fase de entrenamiento) el comportamiento del sistema de forma eficiente, lo cual es verificado a posteriori en la fase de validación. En esta fase se ha utilizado una estrategia de inicialización del estado de las neuronas de la red, que se traduce en un mejor comportamiento de los resultados de validación.
-

- Se han presentado resultados relativos a la identificación del sistema con una red neuronal dinámica, que muestran la bondad del modelo empírico utilizado para el cálculo de la dinámica del manipulador. Los resultados han sido obtenidos, por una parte, en un sistema compuesto por unas piernas robóticas de dos articulaciones y, por otra parte, para el robot PUMA 560 de Unimation.
- El modelo empírico así obtenido se utiliza en los controladores anteriores, sustituyendo la evaluación del modelo teórico no lineal por la evaluación más simple de este modelo neuronal. Los resultados son satisfactorios en cuanto que el seguimiento de las trayectorias de referencia sigue ejecutándose con gran precisión, y además se reduce notablemente la carga computacional en la ejecución de los algoritmos de control.

INTEGRACIÓN DE LAS TÉCNICAS DE CONTROL PARA SOLUCIÓN CONJUNTA DE LOS DISTINTOS PROBLEMAS DE CONTROL

- Para dar solución al problema de las ligaduras y la presencia de incertidumbres o perturbaciones, se añade al controlador anterior una acción robusta adaptativa, igual a la planteada en el controlador ARC, para dar lugar al controlador RIAPC. Se da así una respuesta global para los principales problemas no resueltos en el control de manipuladores.
- Los resultados obtenidos muestran la mejora de incorporar la acción robusta adaptativa al controlador IAPC, para el caso de perturbaciones que afectan al sistema o existencia de incertidumbres no modeladas. El seguimiento de trayectoria es más preciso ante la presencia de perturbaciones y cualquier rango de incertidumbres en el modelo.

A la vista de los resultados obtenidos y como continuación al trabajo realizado, surgen nuevas líneas de investigación con el objetivo de aportar mayores avances a las propuestas que se han realizado en esta tesis. De entre estas líneas cabe destacar las que se citan a continuación, diferenciadas según sean líneas abiertas en cuanto al campo de aplicación de los algoritmos, o bien en cuanto a mejoras de los propios algoritmos.

ALGORITMO

- Una línea abierta en cuanto a los algoritmos definidos en esta tesis consiste en la gestión de las ligaduras por parte del controlador.
-

En este trabajo se ha propuesto una linealización exacta del modelo en cada instante de tiempo, con el fin de mantener una formulación lineal del problema de optimización y de las ligaduras que afectan al sistema.

El hecho de escoger el esquema de perturbación adaptativa en los controladores IAPC y RIAPC, diseñados para tener en cuenta de forma efectiva las ligaduras del sistema, se debe a que se desea mantener una estructura lineal para el controlador, ya que de la parte no lineal de la acción de control se encarga el controlador por dinámica inversa. Precisamente, el algoritmo de linealización por realimentación (que se utilizó en el algoritmo ARC), no se emplea para mantener una formulación lineal del problema de control, ya que en ese caso las ligaduras se impondrían sobre variables que son internas al modelo linealizado.

Pero, por otra parte, la eficiencia computacional del algoritmo de linealización por realimentación, donde se evalúan ciertas funciones no lineales correspondientes a la dinámica del manipulador, es mejor que la del algoritmo de linealización exacta, donde se deben evaluar un número considerable de derivadas parciales.

Por tanto, una posibilidad a estudiar es la aplicación del algoritmo de linealización por realimentación con ligaduras en variables internas al sistema linealizado resultante.

- Una alternativa consiste en utilizar herramientas de control predictivo no lineal para atacar el problema mencionado. Un primer caso se tiene cuando se quiere mantener la linealidad en el problema de optimización, pero con una formulación no lineal de las ligaduras. Un segundo caso se da cuando se plantea la optimización sobre el sistema sin linealizar, en cuyo caso el problema de optimización resulta no lineal. De cualquiera de las dos formas, el control predictivo no lineal aporta soluciones que serían de interés para su implantación en manipuladores con restricciones. Una de las ventajas es que se evita el proceso de linealización del sistema en cada instante de la trayectoria, con lo cual se gana en eficiencia computacional.
- En cuanto a los algoritmos, otra línea de estudio es la mejora en la obtención de los modelos empíricos de los manipuladores. En este trabajo se ha elegido la red neuronal dinámica como la estructura que mejor aprende la dinámica del sistema. Una mejora consistiría en la posibilidad de realizar un aprendizaje *on-line* de la red, con los datos de comportamiento dinámico reales del sistema.

Para ello se han de verificar dos premisas. La primera es que se debe realizar un entrenamiento *off-line* para la inicialización de la red, para que en los primeros instantes de la trayectoria el comportamiento predicho para el sistema por el modelo empírico no diverja completamente respecto al del manipulador. La segunda es que las trayectorias a recorrer deben ser lo suficientemente ricas en frecuencia

para que el aprendizaje *on-line* de la red sea efectivo. Se deberían, además, implementar métodos eficientes de aprendizaje para no aumentar el coste computacional del algoritmo de control.

- Si el entrenamiento *on-line* de la red fuera implementable con éxito, se podrían utilizar otro tipo de redes neuronales, incluidas las redes estáticas, para probar la eficiencia de las mismas a la hora de predecir el comportamiento dinámico del manipulador.

CAMPO DE APLICACIÓN

- Sin duda, uno de los campos de aplicación de las estrategias de control propuestas en esta tesis es el de los manipuladores flexibles. Este tipo de manipuladores tiene un gran interés en la actualidad, ya que los manipuladores se construyen con materiales ligeros, con características elásticas, cuyo objetivo es el de ejecutar trayectorias a velocidades elevadas. En los manipuladores flexibles, las perturbaciones son de gran importancia, ya que las mismas desvían al sistema respecto a las referencias de forma más acusada que en los manipuladores rígidos. Además, por la misma razón, las incertidumbres tienen un efecto considerable a la hora de conseguir una alta precisión en el seguimiento de trayectorias. Por tanto, un algoritmo eficiente en presencia de incertidumbres y en el rechazo de perturbaciones, como el que se propone en esta tesis, es de especial interés en este tipo de estructuras.
 - Otro de los campos de aplicación de este tipo de técnicas es en el control de fuerza. Tal y como se ha comentado, las técnicas de control híbrido han sido ampliamente utilizadas para resolver el problema conjunto del control de posición y fuerza. Estas técnicas podrían ser mejoradas por la inclusión de técnicas en las que se tienen en cuenta las perturbaciones e incertidumbres presentes en el sistema, como las que aquí se proponen, ya que la mayor precisión en el recorrido de trayectorias hará que la tarea de manipulación por parte del efector final sea más sencilla.
-

ANEXO A: Representación de Denavit-Hartenberg

La representación de Denavit-Hartenberg (D-H) consiste en utilizar *matrices de transformación homogéneas* 4x4 para describir la relación espacial entre dos elementos adyacentes de la cadena [Fu88].

Una rotación de un sistema de ejes respecto a otro fijo, se puede representar mediante las *matrices de rotación* 3x3 (R_{3x3}). Con ellas se pueden representar rotaciones en torno a ejes arbitrarios o una sucesión de rotaciones. La necesidad de ampliar estas matrices a las matrices homogéneas 4x4 surge de representar también las operaciones traslacionales del sistema de coordenadas ligado a los cuerpos.

A.1 Matriz de transformación

Una matriz de transformación tiene la siguiente estructura:

$$T = \left[\begin{array}{c|c} R_{3x3} & p_{3x1} \\ \hline f_{1x3} & \omega_{1x1} \end{array} \right] = \left[\begin{array}{c|c} \text{matriz de} & \text{vector de} \\ \text{rotación} & \text{posición} \\ \hline \text{transformación de} & \\ \text{perspectiva} & \text{escalado} \end{array} \right] \quad (\text{A-1})$$

La matriz T es una representación de la transformación de un vector expresado en coordenadas homogéneas de un sistema de referencia, en otro sistema de referencia distinto.

El vector de posición p_{3x1} representa el vector de posición del origen del sistema de referencia rotado y/o trasladado con respecto al sistema de referencia fijo.

La matriz f_{1x3} representa una transformación de perspectiva y ω_{1x1} es el factor de escala¹.

¹ Este factor normalmente será igual a 1 en aplicaciones de la robótica. En aplicaciones de informática gráfica podrá tomar valores positivos distintos de la unidad.

Según esta representación, un vector de posición $p = (p_x, p_y, p_z)^T$ viene dado en coordenadas homogéneas por el vector ampliado $(\omega p_x, \omega p_y, \omega p_z, \omega)^T$. Por ejemplo, si un sistema de ejes rota un ángulo α respecto al eje x de otro sistema fijo, o un ángulo β respecto al eje y , o un ángulo γ respecto al eje z , las matrices de transformación de coordenadas en esta representación vienen dadas respectivamente por:

$$T_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\text{sen} \alpha & 0 \\ 0 & \text{sen} \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{y,\beta} = \begin{bmatrix} \cos \beta & 0 & \text{sen} \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\text{sen} \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A-2})$$

$$T_{z,\gamma} = \begin{bmatrix} \cos \gamma & -\text{sen} \gamma & 0 & 0 \\ \text{sen} \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

De esta forma, un punto $p_{uvw} = (p_u, p_v, p_w, 1)^T$, expresado en coordenadas homogéneas del sistema móvil, se transformará en el punto $p_{xyz} = (p_x, p_y, p_z, 1)^T$, expresado en coordenadas homogéneas del sistema fijo, siguiendo la siguiente expresión:

$$p_{xyz} = T_{i,\theta} p_{uvw} \quad (\text{A-3})$$

donde i puede ser x, y o z , y θ puede ser α, β o γ .

Si, por ejemplo, se produce un movimiento de traslación pura de un sistema de referencia respecto al otro, la matriz de transformación adopta la forma:

$$T_{\text{tras}} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A-4})$$

donde d_x, d_y, d_z son las componentes traslacionales del movimiento.

Las matrices de transformación homogéneas se multiplican entre sí para expresar movimientos de rotaciones y traslaciones sucesivos de los sistemas de referencia, tal y como ocurre con las matrices de rotación $R_{3 \times 3}$.

A.2 Parámetros D-H

La representación D-H permite representar cada uno de los sistemas de coordenadas de las articulaciones, respecto del sistema de la articulación anterior. Ya habíamos dicho que el procedimiento habitual era colocar un sistema de ejes en cada articulación del brazo. Las matrices de transformación homogéneas permiten expresar la posición del efector final respecto de un sistema de coordenadas fijo, habitualmente colocado en la base, mediante transformaciones sucesivas hacia atrás.

Para ello, existe un método sistemático de escoger los sistemas de ejes de cada una de las articulaciones. De esa forma se obtienen una serie de parámetros geométricos, *parámetros D-H*, asociados a cada articulación y de gran importancia en el cómputo de las ecuaciones cinemáticas. Según este método, se elige un sistema de ejes (x_i, y_i, z_i) fijo en la articulación i , tal que se mueva solidariamente a dicha articulación, y un sistema inercial (fijo) en la base (x_0, y_0, z_0) . Por ejemplo, para un sistema de seis grados de libertad (seis articulaciones, por tanto) se tendrán siete sistemas de ejes. Las reglas para establecer esos sistemas de ejes son:

- El eje z_{i-1} se escoge a lo largo del eje de la articulación.
- El eje x_i es perpendicular al eje z_{i-1} .
- El eje y_i se elige de forma que el sistema de ejes sea dextrógiro².

Los parámetros antes mencionados describen completamente cualquier articulación, y se definen como sigue:

- θ_i : Ángulo de la articulación del eje x_{i-1} al eje x_i respecto del eje z_{i-1} , utilizando la regla de la mano derecha.
- d_i : Distancia desde el origen del sistema de ejes $i-1$ hasta la intersección del eje z_{i-1} con el eje x_i , a lo largo del eje z_{i-1} .
- a_i : Distancia desde la intersección del eje z_{i-1} con el eje x_i , hasta el origen del sistema i , a lo largo de x_i .
- α_i : Ángulo entre el eje z_{i-1} al eje z_i , respecto del eje x_i , utilizando la regla de la mano derecha.

En una articulación de revolución, θ_i es la variable de articulación, mientras que el resto de parámetros permanece constante. Para una articulación prismática, d_i es la variable de articulación.

² En animación gráfica se usa el convenio levógiro.

ANEXO B:

Cinemática directa

Las matrices de transformación y los parámetros Denavit-Hartenberg descritos en el Anexo A se utilizan para expresar un punto cualquiera en función de un sistema de coordenadas inercial. Con esta herramienta se puede resolver la cinemática directa del brazo articulado, consistente en el cálculo de la posición y orientación del efector final (en general, de la posición de todas las articulaciones) en función de las coordenadas generalizadas del robot.

Como se ha visto anteriormente, una vez establecidos los sistemas de coordenadas para cada una de las articulaciones, se pueden expresar las coordenadas generalizadas de una articulación respecto de las de la articulación anterior.

La matriz de transformación se demuestra, sin más que multiplicar las dos matrices de rotación asociadas a " α " y a " θ " y las dos matrices de traslación asociadas a " a " y a " d ", que tiene la forma:

$${}^{i-1}A_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B-1})$$

en el caso de una articulación de revolución y

$${}^{i-1}A_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & 0 \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B-2})$$

en el caso de una articulación prismática.

La matriz que especifica la localización del sistema de coordenadas i respecto del de la base, viene dada por el producto de estas matrices:

$${}^0A_i = {}^0A_1 A_2 \cdots {}^{i-1}A_i, \quad i = 1, 2, \dots, n \quad (\text{B-3})$$

Nótese que el problema cinemático directo se reduce entonces al cálculo de este producto y la evaluación de los términos de la matriz 0A_i , dadas las coordenadas generalizadas y los parámetros D-H del brazo. Resolver la cinemática directa equivale a tener un simulador geométrico virtual del robot, pudiendo presentarse los resultados de forma gráfica.

ANEXO C:

Cinemática inversa

El problema cinemático inverso es más habitual que el problema cinemático directo, pues normalmente lo que se trata de hacer es posicionar y orientar el efector final, hallando la configuración del brazo que lo hace posible.

Se han diseñado múltiples métodos para resolver este problema: la transformación inversa (Paul et al. [Pau81]), matrices duales (Denavit [Den56]), cuaterniones duales (Yang y Freudenstein [Yan64]), el álgebra de tornillo (Kohli y Soni [Koh75]), métodos iterativos (Uicker et al. [Uic64]) y métodos geométricos (Lee y Ziegler [Lee84]).

C.1 Solución general de la cinemática inversa

Si se parte de la ecuación (B-3) se tiene, en este caso, los términos de la matriz 0A_n , que nos indican la posición y orientación del efector final respecto al sistema de referencia de la base. Los términos del producto ${}^0A_i = {}^0A_1 {}^1A_2 \cdots {}^{i-1}A_i$ son funciones de senos y cosenos de las variables de articulación (ángulos y distancias). Se puede ver fácilmente que se trata de un sistema con más ecuaciones que incógnitas, por lo que la solución no será única.

La técnica más habitual consiste en dividir este problema en dos: por una parte se resuelve el sistema de ecuaciones correspondiente al brazo y por otro el de la muñeca. Si consideramos que el brazo tiene b articulaciones, podemos poner:

$${}^0A_n = {}^0A_1 {}^1A_2 \cdots {}^{b-1}A_b {}^bA_{b+1} \cdots {}^{n-1}A_n \quad (C-1)$$

La primera parte del producto, ${}^0A_b = {}^0A_1 {}^1A_2 \cdots {}^{b-1}A_b$, puede ser hallada teniendo en cuenta el punto a alcanzar por el brazo sin la muñeca. Dicho punto puede obtenerse, teniendo en cuenta la dirección de acercamiento deseada \bar{a} , de la siguiente expresión:

$$O_b = O_n - l_m \bar{a} \quad (C-2)$$

donde l_m es la longitud de la muñeca, O_n es el punto a alcanzar por la estructura y O_b es el punto que se desea calcular. Las variables de articulación del brazo, entonces, pueden ser calculadas mediante un método geométrico, como los que se presentan a continuación. Con dichas variables se evalúa el producto 0A_b .

El sistema de ecuaciones resultante

$${}^0A_n = {}^0A_b {}^bA_n \quad (\text{C-3})$$

es de más fácil resolución, pues sólo están involucradas las variables de articulación correspondientes a la muñeca. La matriz 0A_n puede ponerse de la siguiente forma:

$${}^0A_n = \begin{bmatrix} \bar{s} & \bar{n} & \bar{a} & \bar{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{C-4})$$

donde \bar{p} es el punto a alcanzar por la estructura, \bar{s} es el vector que indica la orientación de cierre o agarre del efector final, y $\bar{n} = \bar{a} \times \bar{s}$ es un vector normal al plano que forman la dirección de acercamiento \bar{a} y de cierre. Si extraemos en la ecuación (C-3) las matrices de rotación, se tiene que:

$$[\bar{s} \quad \bar{n} \quad \bar{a}] = {}^oR_b {}^bR_n \quad (\text{C-5})$$

Identificando término a término en esta ecuación matricial se obtienen las ecuaciones necesarias para calcular las variables correspondientes al efector final.

C.2 Casos particulares de resolución de la cinemática inversa

Para el caso de un robot con dos articulaciones de rotación, el método geométrico es el de más fácil aplicación. De la figura c-1 podemos obtener las ecuaciones que relacionan las variables de posicionamiento (x, y) con las coordenadas generalizadas (θ_1, θ_2) :

$$\begin{aligned} x &= l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ y &= l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) \end{aligned} \quad (\text{C-6})$$

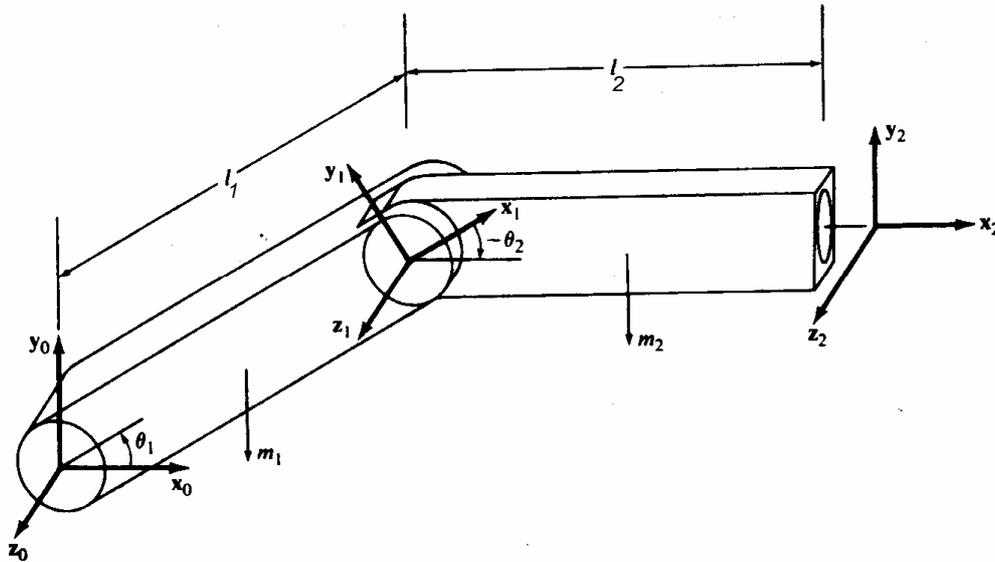


Figura C-1: Esquema simplificado del manipulador de dos elementos, donde se observa la disposición de los sistemas de referencia, las coordenadas generalizadas y los parámetros del brazo (masas y longitudes).

De las ecuaciones (C-6) se pueden obtener, sin más que aplicar directamente el teorema del coseno, las ecuaciones cinemáticas inversas:

$$\theta_2 = \arccos \frac{x^2 + y^2 - (l_1^2 + l_2^2)}{2l_1l_2} \quad (C-7)$$

$$\theta_1 = \arctan \frac{y}{x} - \arctan \frac{l_2 \sin \theta_2}{l_1 + l_2 \cos \theta_2}$$

Dada una posición cualquiera (x,y) , este sistema de ecuaciones o bien no tendrá solución (cuando el punto no es alcanzable por el brazo), o bien tendrá solución única (cuando el punto coincide con la máxima elongación del brazo) o bien tendrá dos soluciones, las correspondientes a las configuraciones *codo arriba* y *codo abajo*. Para obtener estas dos soluciones basta tomar las dos raíces del seno de θ_2 , de la siguiente forma:

$$\cos \theta_2 = \frac{x^2 + y^2 - (l_1^2 + l_2^2)}{2l_1l_2} = C_2$$

$$\sin \theta_2 = \pm \sqrt{1 - C_2^2} \quad (C-8)$$

$$\theta_2 = \arctan \left(\pm \frac{\sqrt{1 - C_2^2}}{C_2} \right)$$

En el caso de un robot de tres articulaciones, donde se añade al robot anterior una articulación giratoria en la base, se puede demostrar fácilmente que las ecuaciones de la cinemática inversa vienen dadas por:

$$\begin{aligned}\theta_3 &= \arccos \frac{x^2 + y^2 + z^2 - (l_2^2 + l_3^2)}{2l_2l_3} \\ \theta_2 &= \arctan \frac{z}{\sqrt{x^2 + y^2}} - \arctan \frac{l_3 \sin \theta_3}{l_2 + l_3 \cos \theta_3} \\ \theta_1 &= \arctan \frac{y}{x}\end{aligned}\quad (\text{C-9})$$

C.3 Planteamiento incremental de la cinemática inversa

El planteamiento presentado en este apartado es un planteamiento absoluto, en cuanto que obtenemos las posiciones cartesianas como una función de las variables generalizadas:

$$\bar{x} = f(\theta_1, \dots, \theta_n) \quad (\text{C-10})$$

donde \bar{x} es la posición cartesiana.

También se puede plantear un tratamiento relativo o incremental. Si derivamos en la ecuación (C-10), se tiene que

$$d\bar{x} = J(\theta_i)d\theta \quad (\text{C-11})$$

donde $J(\theta_i)$ es la matriz Jacobiana, dada por:

$$J = \begin{pmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \dots & \frac{\partial x}{\partial \theta_n} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \dots & \frac{\partial y}{\partial \theta_n} \\ \frac{\partial z}{\partial \theta_1} & \frac{\partial z}{\partial \theta_2} & \dots & \frac{\partial z}{\partial \theta_n} \end{pmatrix} \quad (\text{C-12})$$

y el subíndice i indica que dicha matriz se evalúa en las posiciones iniciales de las variables de articulación. De la ecuación (C-11) podemos obtener el incremento de variables angulares sin más que invertir la relación:

$$\Delta\theta = J^{-1}(\theta_i)\Delta\bar{x} \quad (\text{C-13})$$

siempre que no haya configuraciones singulares (aquellas para las que el jacobiano se anula).

Por tanto, si tenemos las posiciones cartesianas inicial y final, tendremos los correspondientes incrementos $\Delta \bar{x}$. Mediante la ecuación (C-13) podemos obtener los incrementos de las variables generalizadas $\Delta \theta$ con los cuales se tendrán determinadas las variables generalizadas finales.

ANEXO D: Formulación de Lagrange-Euler

La formulación de Lagrange-Euler que se presenta en este apartado está basada en la formulación empleada por Uicker en 1964 [Uic64]. Esta formulación está basada en la formulación D-H vista en el Anexo A. El modelo dinámico de robot obtenido mediante esta formulación resulta en ecuaciones bien estructuradas, donde aparecen de manera clara los pares y fuerzas (inercia, Coriolis, gravedad) que intervienen en el movimiento.

Para la obtención de dichas ecuaciones, se deben seguir los siguientes pasos:

- Colocar en cada elemento de la cadena poliarticulada de robot un sistema de referencia que cumpla con las normas de la representación D-H.
- Obtener las matrices de transformación 0A_i para cada elemento.
- Obtener las matrices

$$U_{ij} = \frac{\partial {}^0A_i}{\partial q_j} = \begin{cases} {}^0A_{j-1} Q_j {}^{j-1}A_i, & \text{si } j \leq i \\ 0 & \text{si } j > i \end{cases} \quad (\text{D-1})$$

donde:

$$Q_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{D-2})$$

si la articulación es de revolución y:

$$Q_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{D-3})$$

si la articulación es prismática.

d) Obtener las matrices:

$$U_{ijk} = \frac{\partial^0 U_{ij}}{\partial q_k} = \frac{\partial}{\partial q_k} \left(\frac{\partial^0 A_i}{\partial q_j} \right) = \begin{cases} {}^0 A_{j-1} Q_j^{j-1} A_{k-1} Q_k^{k-1} A_i & , \text{ si } i \geq k \geq j \\ {}^0 A_{k-1} Q_k^{k-1} A_{j-1} Q_j^{j-1} A_i & , \text{ si } i \geq j \geq k \\ 0 & , \text{ si } k > i \text{ o } j > i \end{cases} \quad (\text{D-4})$$

e) Obtener las matrices de inercias

$$J_i = \begin{bmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int y_i x_i dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int z_i x_i dm & \int z_i y_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i dm & \int y_i dm & \int z_i dm & \int dm \end{bmatrix} \quad (\text{D-5})$$

con las integrales definidas en el elemento i considerado, y (x_i, y_i, z_i) son las coordenadas del diferencial de masa dm respecto al propio sistema de coordenadas del elemento.

f) Obtener la matriz de inercia B de elementos b_{ij} definidos por:

$$b_{ij} = \sum_{k=\max(i,j)}^n \text{Tr}(U_{kj} J_k U_{ki}^T) \quad (\text{D-6})$$

donde $i, j = 1, \dots, n$; siendo n el número de grados de libertad. $\text{Tr}(\cdot)$ designa la función traza de una matriz.

g) Obtener los términos h_{ikm} de los elementos h_i del vector de fuerzas de Coriolis y centrípeta $H = [h_i]$, cuyos elementos están definidos por:

$$h_{ikm} = \sum_{j=\max(i,k,m)}^n \text{Tr}(U_{jkm} J_j U_{ji}^T) \quad (\text{D-7})$$

donde $i, k, m = 1, \dots, n$. El vector queda finalmente definido por:

$$h_i = \sum_{k=1}^n \sum_{m=1}^n h_{ikm} \dot{q}_k \dot{q}_m \quad (\text{D-8})$$

h) Obtener el vector de fuerzas de gravedad $C = [c_i]$ de elementos definidos por:

$$c_i = \sum_{j=1}^n (-m_j g U_{ji}^j r_j) \quad (\text{D-9})$$

con $i = 1, \dots, n$, donde $\mathbf{g} = (g_{x0}, g_{y0}, g_{z0}, 0)$ es el vector de gravedad expresado en el sistema de referencia de la base, y ${}^j \mathbf{r}_j$ es el vector de coordenadas homogéneas del centro de masa del elemento j expresado en el sistema de referencia del elemento i .

i) La ecuación dinámica del sistema viene dada finalmente por:

$$\boldsymbol{\tau} = \mathbf{B}\ddot{\mathbf{q}} + \mathbf{H} + \mathbf{C} \quad (\text{D-10})$$

Para facilitar la consecución de estos pasos es importante resaltar los siguientes aspectos:

- Las matrices de inercia J_i y \mathbf{B} son simétricas y definidas positivas, por lo que sólo habrá que calcular la mitad de los elementos fuera de la diagonal principal de dichas matrices.
- El término h_{ikm} representa la fuerza o el par que se genera en el elemento i como consecuencia del movimiento relativo de los elementos k y m . Luego se cumple que $h_{ikm} = h_{imk}$ y que $h_{iii} = 0$.

Con esto se tienen todas las ecuaciones necesarias para el cálculo de la dinámica de cualquier tipo de manipulador. Todas las operaciones que hay que realizar son puramente matriciales, siendo dichas matrices de tamaño 4×4 . Dado que en la resolución de dichas operaciones no se tiene en cuenta las propiedades de simetría de las matrices, este método resulta ineficiente, y la cantidad de operaciones a realizar depende de la cuarta potencia de los grados de libertad.

Sin embargo, la estructura de las ecuaciones obtenidas es clara, en cuanto que refleja de manera estructurada los pares y fuerzas que intervienen en el movimiento: inercias, fuerzas de Coriolis, centrífuga y gravitatoria.

Anexo E: Formulación de Newton-Euler

La formulación de Newton-Euler parte de las leyes de la mecánica de Newton. El desarrollo de la misma, en el caso de una cadena cinemática abierta, fue el objeto de estudio de muchos investigadores hacia los años 80 (Armstrong, Orín, Luh, Walker y Orin, ...).

Para su obtención, lo primero que se hace es plantear las ecuaciones cinemáticas de los elementos. Después se aplica el principio de d'Alembert para cada elemento en todos los instantes de tiempo: "la suma algebraica de las fuerzas aplicadas externamente y las fuerzas de reacción en una dirección dada es cero". Esto conduce finalmente a un conjunto de ecuaciones cuya principal característica es su recursividad.

Formulación recursiva de Newton-Euler

El sistema de ecuaciones se puede dividir en dos: un conjunto de ecuaciones que se aplican "hacia delante" y otro conjunto que se aplican "hacia atrás". En las primeras, la velocidad lineal, la velocidad angular y la aceleración lineal, se propagan desde el sistema de referencia (SR) de la base hasta el efector final. En las segundas, los pares y fuerzas ejercidos sobre cada elemento son calculados hacia atrás.

Si todos los parámetros que interviniesen en las ecuaciones (momentos de inercia, velocidades, posiciones de los centros de masa de cada elemento, fuerzas y momentos) los refiriéramos al SR de la base, tendrían que ser recalculados cada vez que el robot cambia de posición, por lo que cualquier algoritmo de control en tiempo real dejaría de ser eficiente.

En vez de eso, se opta por referir todas estas magnitudes a su propio SR, haciendo uso de las matrices de rotación ${}^{i-1}R_i$. Estas matrices transforman cualquier vector referido al sistema (x_i, y_i, z_i) al sistema $(x_{i-1}, y_{i-1}, z_{i-1})$, y tienen la forma:

$${}^{i-1}R_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i \end{bmatrix} \quad (\text{E-1})$$

Para referir las magnitudes de un elemento a su propio SR, basta con multiplicar por la matriz de rotación 0R_i [Luh80a]. Por ejemplo: si ω_i es la velocidad angular del elemento i referida al SR de la base, ${}^0R_i\omega_i$ está referida al sistema (x_i, y_i, z_i) .

Ecuaciones hacia delante.	$i = 1, 2, \dots, n.$
<p>Si la articulación i es de revolución:</p> ${}^iR_o\omega_i = {}^iR_{i-1}\left({}^{i-1}R_o\omega_{i-1} + z_o\dot{q}_i\right)$ ${}^iR_o\dot{\omega}_i = {}^iR_{i-1}\left[{}^{i-1}R_o\dot{\omega}_{i-1} + z_o\ddot{q}_i + \left({}^{i-1}R_o\omega_{i-1}\right) \times \left(z_o\dot{q}_i\right)\right]$ ${}^iR_o\dot{v}_i = \left({}^iR_o\dot{\omega}_i\right) \times \left({}^iR_o p_i^*\right) + \left({}^iR_o\omega_i\right) \times \left[\left({}^iR_o\omega_i\right) \times \left({}^iR_o p_i^*\right)\right] + {}^iR_{i-1}\left({}^{i-1}R_o\dot{v}_{i-1}\right)$ ${}^iR_o\bar{a}_i = \left({}^iR_o\dot{\omega}_i\right) \times \left({}^iR_o\bar{s}_i\right) + \left({}^iR_o\omega_i\right) \times \left[\left({}^iR_o\omega_i\right) \times \left({}^iR_o\bar{s}_i\right)\right] + {}^iR_o\dot{v}_i$	
<p>Si la articulación i es prismática:</p> ${}^iR_o\omega_i = {}^iR_{i-1}\left({}^{i-1}R_o\omega_{i-1}\right)$ ${}^iR_o\dot{\omega}_i = {}^iR_{i-1}\left({}^{i-1}R_o\dot{\omega}_{i-1}\right)$ ${}^iR_o\dot{v}_i = {}^iR_{i-1}\left(z_o\ddot{q}_i + {}^{i-1}R_o\dot{v}_{i-1}\right) + \left({}^iR_o\dot{\omega}_i\right) \times \left({}^iR_o p_i^*\right) + 2\left({}^iR_o\omega_i\right) \times \left({}^iR_{i-1}z_o\dot{q}_i\right) + \left({}^iR_o\omega_i\right) \times \left[\left({}^iR_o\dot{\omega}_i\right) \times \left({}^iR_o p_i^*\right)\right]$ ${}^iR_o\bar{a}_i = \left({}^iR_o\dot{\omega}_i\right) \times \left({}^iR_o\bar{s}_i\right) + \left({}^iR_o\omega_i\right) \times \left[\left({}^iR_o\omega_i\right) \times \left({}^iR_o\bar{s}_i\right)\right] + {}^iR_o\dot{v}_i$	
Ecuaciones hacia atrás.	$i = n, n-1, \dots, 1.$
${}^iR_oF_i = m_i {}^iR_o\bar{a}_i$ ${}^iR_oN_i = \left({}^iR_oI_i {}^oR_i\right) \left({}^iR_o\dot{\omega}_i\right) + \left({}^iR_o\omega_i\right) \times \left[\left({}^iR_oI_i {}^oR_i\right) \left({}^iR_o\omega_i\right)\right]$ ${}^iR_o f_i = {}^iR_{i+1}\left({}^{i+1}R_o f_{i+1}\right) + {}^iR_o F_i$ ${}^iR_o n_i = {}^iR_{i+1}\left[{}^{i+1}R_o n_{i+1} + \left({}^{i+1}R_o p_i^*\right) \times \left({}^{i+1}R_o f_{i+1}\right)\right] + \left({}^iR_o p_i^* + {}^iR_o\bar{s}_i\right) \times \left({}^iR_o F_i\right) + {}^iR_o N_i$ $\tau_i = \begin{cases} \left({}^iR_o n_i\right)^T \left({}^iR_{i-1}z_o\right) + b_i\dot{q}_i & , \text{ si artic. } i \text{ rotacional} \\ \left({}^iR_o f_i\right)^T \left({}^iR_{i-1}z_o\right) + b_i\dot{q}_i & , \text{ si artic. } i \text{ traslacional} \end{cases}$	

Tabla E-1 : Formulación recursiva de Newton-Euler.

Las ecuaciones recursivas hacia delante propagan la información cinemática desde la base al efector final, mientras que las ecuaciones recursivas hacia atrás calculan los pares y momentos en cada articulación desde el efector final hasta la base. Dichas ecuaciones quedan recogidas en la tabla e-1 [Fu88]. El significado de cada uno de los términos se expresa en la tabla e-2.

Como condiciones iniciales más usuales se suelen escoger $\omega_o = \dot{\omega}_o = v_o = 0$ y $\dot{v}_o = (g_x, g_y, g_z)^T$, para incluir la gravedad.

q_i	Coordenada generalizada del elemento i.
\bar{s}_i	Posición del CM del elemento i desde el origen del SR a la base.
\bar{v}_i	Velocidad lineal del CM del elemento i.
$\bar{a}_i = \frac{d\bar{v}_i}{dt}$	Aceleración lineal del CM del elemento i.
ω_i	Velocidad angular del elemento i.
τ_i	Momento aplicado a la articulación i.
m_i	Masa total del elemento i.
b_i	Coefficiente de rozamiento viscoso del elemento i.
p_i	Origen del SR i-ésimo respecto al SR (i-1)-ésimo.
F_i	Fuerza externa total ejercida sobre el CM del elemento i.
N_i	Momento externo total ejercido sobre el CM del elemento i.
I_i	Momento de inercia del elemento i respecto de su CM con respecto al SR de la base.
f_i	Fuerza ejercida por el elemento i-1 sobre el elemento i, respecto al SR $(x_{i-1}, y_{i-1}, z_{i-1})$.
n_i	Momento ejercido por el elemento i-1 sobre el elemento i, respecto al SR $(x_{i-1}, y_{i-1}, z_{i-1})$.
n	Número de grados de libertad del brazo.
z_o	Eje z del SR de la base: $(0,0,1)^T$.

Tabla E-2 : Notación empleada en la formulación recursiva de Newton-Euler.

Comparación con la formulación de Lagrange-Euler

La formulación de Newton-Euler permite la obtención de los momentos resultantes en cada instante de tiempo y para cada uno de los elementos, a través de un algoritmo recursivo de bajo coste computacional. Esto es así puesto que al utilizar solamente matrices de rotación y vectores de posición, el número de operaciones se reduce enormemente respecto de la formulación de Lagrange-Euler.

En la formulación de Lagrange-Euler se utilizan matrices homogéneas 4x4, por lo que el uso de las mismas para el cálculo de los momentos resultantes, a partir de las posiciones y sus derivadas, resulta en una gran cantidad de operaciones, tal y como se observa en la tabla e-3. Este número de operaciones se ve reducido en la formulación de Newton-Euler a una cantidad proporcional al número de grados de libertad del robot. Por esta razón, este

segundo conjunto de ecuaciones resulta más adecuado para un control en tiempo real.

	Lagrange-Euler	Newton-Euler
Multiplicaciones	$\frac{128}{3}n^4 + \frac{512}{3}n^3 + \frac{739}{3}n^2 + \frac{160}{3}n$	$132 n$
Sumas	$\frac{128}{3}n^4 + \frac{512}{3}n^3 + \frac{739}{3}n^2 + \frac{160}{3}n$	$111 n - 4$
Representación de la cinemática	Matrices homogéneas 4x4	Matrices de rotación y vectores de posición

Tabla E-3 : Comparación de la complejidad computacional de las formulaciones de Lagrange-Euler y Newton-Euler.

Tanto en la formulación de Lagrange-Euler como en la de Newton-Euler, uno de los aspectos importantes es el cálculo de los momentos de inercia de los distintos elementos que componen la estructura del manipulador. Este aspecto será examinado en el anexo F.

ANEXO F:

Momentos de Inercia

En las ecuaciones del movimiento del manipulador, uno de los puntos importantes es el cómputo de los términos de inercia. La matriz $\mathbf{B}(\boldsymbol{\theta}(t))$ se llama matriz de inercia y sus elementos están relacionados con las aceleraciones en cada uno de los elementos y con los pares de fuerza que éstas provocan. En estos términos, por tanto, intervienen los momentos de inercia de cada uno de los elementos.

Tanto en la formulación de Lagrange-Euler como en la de Newton-Euler, vistas en los dos anexos anteriores, estos términos deben ser evaluados para obtener los términos de inercia del manipulador. Así, en la primera formulación aparece la matriz de inercia (D-5) necesaria para calcular la matriz $\mathbf{B}(\boldsymbol{\theta}(t))$, mientras que en la segunda formulación aparecen los momentos de inercia en las ecuaciones hacia atrás (ver tabla E-1 y E-2).

Los elementos del tensor inercia I_{ij} de un sólido vienen definidos por la expresión:

$$I_{ij} = \int_V \left[\delta_{ij} \left(\sum_k x_k^2 \right) - x_i x_j \right] dm \quad (\text{F-1})$$

donde los índices i, j, k representan los ejes principales de inercia del sólido y δ_{ij} es la delta de Kronecker. La integral se extiende a todo el volumen V del sólido.

Para el caso de cuerpos rígidos bien definidos y de densidad uniforme, los momentos de inercia son fácilmente calculables. Así, por ejemplo, el momento de inercia I de una esfera homogénea de masa M y radio r , respecto a un eje que coincida con uno de sus diámetros, viene dado por:

$$I = \frac{2}{5} Mr^2 \quad (\text{F-2})$$

El momento de inercia de una varilla, considerada delgada, de longitud l , respecto a un eje perpendicular a la varilla y que pasa por su centro, viene dado por:

$$I = \frac{1}{12} Ml^2 \quad (\text{F-3})$$

Un teorema importante para el cálculo de momentos de inercia es el *teorema de los ejes paralelos* o *teorema de Steiner*. Este teorema dice que, siendo I_O el momento de inercia respecto a un eje O , el momento de inercia $I_{O'}$ respecto a un eje O' paralelo a O y que dista de aquél una distancia d , se puede calcular como:

$$I_{O'} = I_O + Md^2 \quad (\text{F-4})$$

siendo M la masa total del sólido.

A la vista de este teorema, sólo hará falta calcular los momentos de inercia respecto al centro de masas del sólido, pudiéndose calcular los momentos respecto a otros ejes mediante la aplicación directa del teorema. Con este teorema y la definición (F-1) se pueden calcular los términos de la matriz de inercia, que serán necesarios para los aspectos de control que veremos más adelante.

ANEXO G:

Linealización exacta

El modelo para las tres primeras articulaciones del manipulador PUMA viene dado por la expresión:

$$\boldsymbol{\tau} = \mathbf{B}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{c}(\boldsymbol{\theta}) = \mathbf{B}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{n}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \quad (\text{G-1})$$

acotada a los tres primeros grados de libertad. Despejando las aceleraciones, se tiene:

$$\ddot{\theta}_1 = \frac{\tau_1 - n_1}{B_{11}} \quad (\text{G-2})$$

$$\ddot{\theta}_2 = \frac{B_{33}(\tau_2 - n_2) - B_{23}(\tau_3 - n_3)}{B_{22}B_{33} - B_{23}B_{32}} \quad (\text{G-3})$$

$$\ddot{\theta}_3 = \frac{-B_{23}(\tau_2 - n_2) + B_{22}(\tau_3 - n_3)}{B_{22}B_{33} - B_{23}B_{32}} \quad (\text{G-4})$$

donde B_{ij} son los elementos de la matriz de inercia y n_i los elementos del vector de términos no lineales de (G-1). Para el manipulador PUMA-560 con el que se trabaja en esta tesis, los elementos de inercia vienen dados por¹:

$$B_{11} = \frac{1}{3}m_3l_3^2C_{23}^2 + m_3l_2l_3C_2C_{23} + \frac{1}{3}m_3l_2^2C_2 \quad (\text{G-5})$$

$$B_{22} = m_3l_2l_3C_3 + m_3l_2^2 + \frac{1}{3}m_2l_2^2 + \frac{1}{3}m_3l_3^2 \quad (\text{G-6})$$

$$B_{23} = B_{32} = \frac{1}{3}m_3l_2l_3C_3 + \frac{1}{3}m_3l_3^2 \quad (\text{G-7})$$

$$B_{33} = \frac{1}{3}m_3l_3^2 \quad (\text{G-8})$$

¹ Se tiene en cuenta la notación empleada en el capítulo 1 en cuanto a la manera de expresar las funciones trigonométricas que intervienen en las ecuaciones. Así, $S_i = \sin(\theta_i)$, $C_i = \cos(\theta_i)$, $S_{ij} = \sin(\theta_i + \theta_j)$ y $C_{ij} = \cos(\theta_i + \theta_j)$.

Mientras que los términos no lineales vienen dados por:

$$\begin{aligned} n_1 = & -m_3 l_2 l_3 \dot{\theta}_1 \dot{\theta}_3 C_2 S_{23} - \frac{2}{3} m_3 l_3^2 \dot{\theta}_1 \dot{\theta}_3 S_2 C_{23} - \frac{2}{3} m_2 l_2^2 \dot{\theta}_1 \dot{\theta}_2 S_2 C_2 - \\ & - 2m_3 l_2^2 \dot{\theta}_1 \dot{\theta}_2 S_2 C_2 - m_3 l_2 l_3 \dot{\theta}_1 \dot{\theta}_2 S_2 C_{23} - m_3 l_2 l_3 \dot{\theta}_1 \dot{\theta}_2 C_2 S_{23} - \\ & - 2m_3 l_3^2 \dot{\theta}_1 \dot{\theta}_2 S_{23} C_{23} \end{aligned} \quad (G-9)$$

$$\begin{aligned} n_2 = & -m_3 l_2 l_3 \dot{\theta}_2 \dot{\theta}_3 S_2 - \frac{1}{2} m_3 l_2 l_3 S_2 \dot{\theta}_3^2 + \frac{1}{3} m_2 l_2^2 S_2 C_2 \dot{\theta}_1^2 + \\ & + m_3 l_2^2 S_2 C_2 \dot{\theta}_1^2 + \frac{1}{3} m_3 l_2 l_3 S_2 C_{23} \dot{\theta}_1^2 + \frac{1}{2} m_3 l_2 l_3 C_2 S_{23} \dot{\theta}_1^2 + \\ & + \frac{1}{2} m_3 l_3^2 \dot{\theta}_1^2 S_{23} C_{23} + \frac{1}{2} m_2 l_2 g C_2 + m_3 l_2 g C_2 + \frac{1}{2} m_3 l_3 g C_{23} \end{aligned} \quad (G-10)$$

$$n_3 = \frac{1}{2} m_3 l_2 l_3 S_2 \dot{\theta}_2^2 + \frac{1}{2} m_3 l_2 l_3 C_2 S_{23} \dot{\theta}_1^2 + \frac{1}{3} m_3 l_3^2 S_{23} C_{23} \dot{\theta}_1^2 + \frac{1}{2} m_3 l_3 g C_{23} \quad (G-11)$$

Entonces, se realiza la linealización en cada instante en torno al punto actual de la trayectoria de referencia, desarrollando las expresiones en serie de Taylor hasta primer orden.

Primera articulación

La aceleración para esta articulación es una función no lineal de los ángulos, velocidades y entradas de la forma:

$$\ddot{\theta}_1 = \frac{\tau_1 - n_1}{B_{11}} = F_1(\theta_2, \theta_3; \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3; \tau_1) \quad (G-12)$$

Desarrollando en serie de Taylor hasta primer orden en torno a la trayectoria de referencia, que se denota con el subíndice d , se tiene:

$$\begin{aligned} \ddot{\theta}_1 = F_1(\theta_2, \theta_3; \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3; \tau_1) \approx & F_1 \Big|_d + \frac{\partial F_1}{\partial \theta_2} \Big|_d (\theta_2 - \theta_{2d}) + \frac{\partial F_1}{\partial \theta_3} \Big|_d (\theta_3 - \theta_{3d}) + \\ & + \frac{\partial F_1}{\partial \dot{\theta}_1} \Big|_d (\dot{\theta}_1 - \dot{\theta}_{1d}) + \frac{\partial F_1}{\partial \dot{\theta}_2} \Big|_d (\dot{\theta}_2 - \dot{\theta}_{2d}) + \frac{\partial F_1}{\partial \dot{\theta}_3} \Big|_d (\dot{\theta}_3 - \dot{\theta}_{3d}) + \frac{\partial F_1}{\partial \tau_1} \Big|_d (\tau_1 - \tau_{1o}) \end{aligned} \quad (G-13)$$

donde τ_{1o} es el par aplicado a la primera articulación para seguir la trayectoria de referencia, es decir, el par calculado por el controlador de dinámica inversa. Teniendo en cuenta la definición habitual para el error de seguimiento $\tilde{\theta}_i = \theta_{id} - \theta_i$ y, definiendo $\tilde{\tau}_i = \tau_i - \tau_{io}$ como la diferencia de par que se va a aplicar respecto al par calculado τ_{io} , la expresión (G-13) queda:

$$\ddot{\tilde{\theta}}_1 = \left[\frac{\partial F_1}{\partial \theta_2} \right]_d \tilde{\theta}_2 + \left[\frac{\partial F_1}{\partial \theta_3} \right]_d \tilde{\theta}_3 + \left[\frac{\partial F_1}{\partial \dot{\theta}_1} \right]_d \dot{\tilde{\theta}}_1 + \left[\frac{\partial F_1}{\partial \dot{\theta}_2} \right]_d \dot{\tilde{\theta}}_2 + \left[\frac{\partial F_1}{\partial \dot{\theta}_3} \right]_d \dot{\tilde{\theta}}_3 + \left[\frac{\partial F_1}{\partial \tau_1} \right]_d \tilde{\tau}_1 \quad (\text{G-14})$$

Los términos del desarrollo vienen dados por las siguientes expresiones:

$$\frac{\partial F_1}{\partial \theta_2} = -\frac{1}{B_{11}} \frac{\partial n_1}{\partial \theta_2} + \frac{n_1}{B_{11}^2} \frac{\partial B_{11}}{\partial \theta_2} \quad (\text{G-15})$$

$$\frac{\partial F_1}{\partial \theta_3} = -\frac{1}{B_{11}} \frac{\partial n_1}{\partial \theta_3} - \frac{F_1}{B_{11}} \frac{\partial B_{11}}{\partial \theta_3} \quad (\text{G-16})$$

$$\frac{\partial F_1}{\partial \dot{\theta}_1} = -\frac{1}{B_{11}} \frac{\partial n_1}{\partial \dot{\theta}_1} \quad (\text{G-17})$$

$$\frac{\partial F_1}{\partial \dot{\theta}_2} = -\frac{1}{B_{11}} \frac{\partial n_1}{\partial \dot{\theta}_2} \quad (\text{G-18})$$

$$\frac{\partial F_1}{\partial \dot{\theta}_3} = -\frac{1}{B_{11}} \frac{\partial n_1}{\partial \dot{\theta}_3} \quad (\text{G-19})$$

$$\frac{\partial F_1}{\partial \tau_1} = \frac{1}{B_{11}} \quad (\text{G-20})$$

siendo:

$$\begin{aligned} \frac{\partial n_1}{\partial \theta_2} = & -m_3 l_2 l_3 \dot{\theta}_1 \dot{\theta}_3 (-S_2 S_{23} + C_2 C_{23}) - \frac{2}{3} m_3 l_3^2 \dot{\theta}_1 \dot{\theta}_3 (C_{23}^2 - S_{23}^2) - \\ & - \left(\frac{2}{3} m_2 l_2^2 \dot{\theta}_1 \dot{\theta}_2 + 2m_3 l_2^2 \dot{\theta}_1 \dot{\theta}_2 \right) (C_2^2 - S_2^2) - \\ & - m_3 l_2 l_3 \dot{\theta}_1 \dot{\theta}_2 (C_2 C_{23} + S_2 S_{23}) - m_3 l_2 l_3 \dot{\theta}_1 \dot{\theta}_2 (-S_2 S_{23} + C_2 C_{23}) - \\ & - 2m_3 l_3^2 \dot{\theta}_1 \dot{\theta}_2 (C_{23}^2 - S_{23}^2) \end{aligned} \quad (\text{G-21})$$

$$\frac{\partial B_{11}}{\partial \theta_2} = -\frac{2}{3} m_3 l_3^2 C_{23} S_{23} + m_3 l_2 l_3 (-S_2 S_{23} + C_2 C_{23}) - \frac{2}{3} m_3 l_2^2 C_2 S_2 \quad (\text{G-22})$$

$$\begin{aligned} \frac{\partial n_1}{\partial \theta_3} = & -m_3 l_2 l_3 \dot{\theta}_1 \dot{\theta}_3 C_2 C_{23} - \frac{2}{3} m_3 l_3^2 \dot{\theta}_1 \dot{\theta}_3 (C_{23}^2 - S_{23}^2) + m_3 l_2 l_3 \dot{\theta}_1 \dot{\theta}_2 S_2 S_{23} - \\ & - m_3 l_2 l_3 \dot{\theta}_1 \dot{\theta}_2 C_2 C_{23} - 2m_3 l_3^2 \dot{\theta}_1 \dot{\theta}_2 (C_{23}^2 - S_{23}^2) \end{aligned} \quad (\text{G-23})$$

$$\frac{\partial B_{11}}{\partial \theta_3} = -\frac{2}{3} m_3 l_3^2 C_{23} S_{23} - m_3 l_2 l_3 C_2 S_{23} \quad (\text{G-24})$$

$$\begin{aligned} \frac{\partial n_1}{\partial \dot{\theta}_1} = & -m_3 l_2 l_3 \dot{\theta}_3 C_2 S_{23} - \frac{2}{3} m_3 l_3^2 \dot{\theta}_3 S_{23} C_{23} - \frac{2}{3} m_2 l_2^2 \dot{\theta}_2 S_2 C_2 - \\ & - 2m_3 l_2^2 \dot{\theta}_1 S_2 C_2 - m_3 l_2 l_3 \dot{\theta}_2 S_2 C_{23} - m_3 l_2 l_3 \dot{\theta}_2 C_2 S_{23} \\ & - 2m_3 l_3^2 \dot{\theta}_2 S_{23} C_{23} \end{aligned} \quad (G-25)$$

$$\begin{aligned} \frac{\partial n_1}{\partial \dot{\theta}_2} = & \frac{2}{3} m_2 l_2^2 \dot{\theta}_1 S_2 C_2 - 2m_3 l_2^2 \dot{\theta}_1 S_2 C_2 - m_3 l_2 l_3 \dot{\theta}_1 S_2 C_{23} - \\ & - m_3 l_2 l_3 \dot{\theta}_1 C_2 S_{23} - 2m_3 l_3^2 \dot{\theta}_1 S_{23} C_{23} \end{aligned} \quad (G-26)$$

$$\frac{\partial n_1}{\partial \dot{\theta}_3} = -m_3 l_2 l_3 \dot{\theta}_1 C_2 S_{23} - \frac{2}{3} m_3 l_3^2 \dot{\theta}_1 S_{23} C_{23} \quad (G-27)$$

Segunda articulación

Procediendo igual que para la primera articulación, la aceleración de la segunda articulación viene dada por:

$$\ddot{\theta}_2 = \frac{B_{33}(\tau_2 - n_2) - B_{23}(\tau_3 - n_3)}{B_{22}B_{33} - B_{23}B_{32}} = F_2(\theta_2, \theta_3; \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3; \tau_2, \tau_3) \quad (G-28)$$

con lo que desarrollando en serie de Taylor queda la expresión:

$$\begin{aligned} \ddot{\theta}_2 = & \frac{\partial F_2}{\partial \theta_2} \Big|_d \tilde{\theta}_2 + \frac{\partial F_2}{\partial \theta_3} \Big|_d \tilde{\theta}_3 + \frac{\partial F_2}{\partial \dot{\theta}_1} \Big|_d \dot{\tilde{\theta}}_1 + \frac{\partial F_2}{\partial \dot{\theta}_2} \Big|_d \dot{\tilde{\theta}}_2 + \frac{\partial F_2}{\partial \dot{\theta}_3} \Big|_d \dot{\tilde{\theta}}_3 + \\ & + \frac{\partial F_2}{\partial \tau_2} \Big|_d \tilde{\tau}_2 + \frac{\partial F_2}{\partial \tau_3} \Big|_d \tilde{\tau}_3 \end{aligned} \quad (G-29)$$

Los términos del desarrollo vienen dados por las siguientes expresiones:

$$\frac{\partial F_2}{\partial \theta_2} = \frac{-1}{B_{22}B_{33} - B_{23}B_{32}} \left[B_{33} \frac{\partial n_2}{\partial \theta_2} - B_{23} \frac{\partial n_3}{\partial \theta_2} \right] \quad (G-30)$$

$$\begin{aligned} \frac{\partial F_2}{\partial \theta_3} = & \frac{1}{B_{22}B_{33} - B_{23}B_{32}} \left[-B_{33} \frac{\partial n_2}{\partial \theta_3} + \frac{\partial B_{23}}{\partial \theta_3} n_3 + B_{23} \frac{\partial n_3}{\partial \theta_3} \right] - \\ & - \frac{F_2}{B_{22}B_{33} - B_{23}B_{32}} \left[\frac{\partial B_{22}}{\partial \theta_3} B_{33} - \frac{\partial B_{23}}{\partial \theta_3} B_{32} - B_{23} \frac{\partial B_{32}}{\partial \theta_3} \right] \end{aligned} \quad (G-31)$$

$$\frac{\partial F_2}{\partial \dot{\theta}_1} = \frac{-1}{B_{22}B_{33} - B_{23}B_{32}} \left[B_{33} \frac{\partial n_2}{\partial \dot{\theta}_1} - B_{23} \frac{\partial n_3}{\partial \dot{\theta}_1} \right] \quad (G-32)$$

$$\frac{\partial F_2}{\partial \dot{\theta}_2} = \frac{-1}{B_{22}B_{33} - B_{23}B_{32}} \left[B_{33} \frac{\partial n_2}{\partial \dot{\theta}_2} - B_{23} \frac{\partial n_3}{\partial \dot{\theta}_2} \right] \quad (\text{G-33})$$

$$\frac{\partial F_2}{\partial \dot{\theta}_3} = \frac{-1}{B_{22}B_{33} - B_{23}B_{32}} B_{33} \frac{\partial n_2}{\partial \dot{\theta}_3} \quad (\text{G-34})$$

$$\frac{\partial F_2}{\partial \tau_2} = \frac{B_{33}}{B_{22}B_{33} - B_{23}B_{32}} \quad (\text{G-35})$$

$$\frac{\partial F_2}{\partial \tau_3} = \frac{-B_{23}}{B_{22}B_{33} - B_{23}B_{32}} \quad (\text{G-36})$$

siendo:

$$\begin{aligned} \frac{\partial n_2}{\partial \theta_2} = & -m_3 l_2 l_3 \dot{\theta}_2 \dot{\theta}_3 C_2 - \frac{1}{2} m_3 l_2 l_3 \dot{\theta}_3^2 C_2 + \frac{1}{3} m_2 l_2^2 \dot{\theta}_1^2 (C_2^2 - S_2^2) + \\ & + m_3 l_2^2 \dot{\theta}_1^2 (C_2^2 - S_2^2) + \frac{1}{3} m_3 l_2 l_3 \dot{\theta}_1^2 (C_2 C_{23} + S_2 S_{23}) + \\ & + \frac{1}{3} m_3 l_2 l_3 \dot{\theta}_1^2 (-S_2 S_{23} + C_2 C_{23}) + \frac{1}{2} m_3 l_3^2 \dot{\theta}_1^2 (C_{23}^2 - S_{23}^2) - \\ & - \frac{1}{2} m_2 l_2 g S_2 - m_3 l_2 g S_2 - \frac{1}{2} m_3 l_3 g S_{23} \end{aligned} \quad (\text{G-37})$$

$$\begin{aligned} \frac{\partial n_3}{\partial \theta_2} = & \frac{1}{2} m_3 l_2 l_3 \dot{\theta}_2^2 C_2 + \frac{1}{2} m_3 l_2 l_3 \dot{\theta}_2^2 (-S_2 S_{23} + C_2 C_{23}) + \\ & + \frac{1}{3} m_3 l_3^2 \dot{\theta}_1^2 (C_{23}^2 - S_{23}^2) - \frac{1}{2} m_3 l_3 g S_{23} \end{aligned} \quad (\text{G-38})$$

$$\begin{aligned} \frac{\partial n_2}{\partial \theta_3} = & -\frac{1}{3} m_3 l_2 l_3 \dot{\theta}_1^2 S_2 S_{23} + \frac{1}{2} m_3 l_2 l_3 \dot{\theta}_1^2 C_2 C_{23} + \\ & + \frac{1}{2} m_3 l_3^2 \dot{\theta}_1^2 (C_{23}^2 - S_{23}^2) - \frac{1}{2} m_3 l_3 g S_{23} \end{aligned} \quad (\text{G-39})$$

$$\frac{\partial B_{23}}{\partial \theta_3} = \frac{\partial B_{32}}{\partial \theta_3} = -\frac{1}{3} m_3 l_2 l_3 S_3 \quad (\text{G-40})$$

$$\frac{\partial n_3}{\partial \theta_3} = \frac{1}{2} m_3 l_2 l_3 \dot{\theta}_2^2 C_2 C_{23} + \frac{1}{3} m_3 l_3^2 \dot{\theta}_1^2 (C_{23}^2 - S_{23}^2) - \frac{1}{3} m_3 l_3 g S_{23} \quad (\text{G-41})$$

$$\frac{\partial B_{22}}{\partial \theta_3} = -m_3 l_2 l_3 S_3 \quad (\text{G-42})$$

$$\begin{aligned} \frac{\partial n_2}{\partial \dot{\theta}_1} = & \frac{2}{3} m_2 l_2^2 S_2 C_2 \dot{\theta}_1 + 2 m_3 l_2^2 S_2 C_2 \dot{\theta}_1 + \frac{2}{3} m_3 l_2 l_3 \dot{\theta}_1 S_2 C_{23} + \\ & + m_3 l_2 l_3 \dot{\theta}_1 C_2 S_{23} + m_3 l_3^2 \dot{\theta}_1 S_{23} C_{23} \end{aligned} \quad (\text{G-43})$$

$$\frac{\partial n_3}{\partial \dot{\theta}_1} = m_3 l_2 l_3 \dot{\theta}_1 C_2 S_{23} + \frac{2}{3} m_3 l_3^2 \dot{\theta}_1 S_{23} C_{23} \quad (\text{G-44})$$

$$\frac{\partial n_2}{\partial \dot{\theta}_2} = -m_3 l_2 l_3 \dot{\theta}_3 S_2 \quad (\text{G-45})$$

$$\frac{\partial n_3}{\partial \dot{\theta}_2} = m_3 l_2 l_3 \dot{\theta}_2 S_2 \quad (\text{G-46})$$

$$\frac{\partial n_2}{\partial \dot{\theta}_3} = -m_3 l_2 l_3 \dot{\theta}_2 S_2 - m_3 l_2 l_3 \dot{\theta}_3 S_2 \quad (\text{G-47})$$

Tercera articulación

Procediendo igual que para la segunda articulación, la aceleración de la tercera articulación viene dada por:

$$\ddot{\theta}_3 = \frac{-B_{23}(\tau_2 - n_2) + B_{22}(\tau_3 - n_3)}{B_{22}B_{33} - B_{23}B_{32}} = F_3(\theta_2, \theta_3; \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3; \tau_2, \tau_3) \quad (\text{G-48})$$

con lo que desarrollando en serie de Taylor queda la expresión:

$$\begin{aligned} \ddot{\theta}_3 = & \left. \frac{\partial F_3}{\partial \theta_2} \right]_d \tilde{\theta}_2 + \left. \frac{\partial F_3}{\partial \theta_3} \right]_d \tilde{\theta}_3 + \left. \frac{\partial F_3}{\partial \dot{\theta}_1} \right]_d \dot{\tilde{\theta}}_1 + \left. \frac{\partial F_3}{\partial \dot{\theta}_2} \right]_d \dot{\tilde{\theta}}_2 + \left. \frac{\partial F_3}{\partial \dot{\theta}_3} \right]_d \dot{\tilde{\theta}}_3 + \\ & + \left. \frac{\partial F_3}{\partial \tau_2} \right]_d \tilde{\tau}_2 + \left. \frac{\partial F_3}{\partial \tau_3} \right]_d \tilde{\tau}_3 \end{aligned} \quad (\text{G-49})$$

Los términos del desarrollo vienen dados por las siguientes expresiones:

$$\frac{\partial F_3}{\partial \dot{\theta}_1} = \frac{1}{B_{22}B_{33} - B_{23}B_{32}} \left[B_{23} \frac{\partial n_2}{\partial \dot{\theta}_1} - B_{22} \frac{\partial n_3}{\partial \dot{\theta}_1} \right] \quad (\text{G-50})$$

$$\frac{\partial F_3}{\partial \dot{\theta}_2} = \frac{1}{B_{22}B_{33} - B_{23}B_{32}} \left[B_{23} \frac{\partial n_2}{\partial \dot{\theta}_2} - B_{22} \frac{\partial n_3}{\partial \dot{\theta}_2} \right] \quad (\text{G-51})$$

$$\frac{\partial F_3}{\partial \dot{\theta}_3} = \frac{B_{23}}{B_{22}B_{33} - B_{23}B_{32}} \frac{\partial n_2}{\partial \dot{\theta}_3} \quad (\text{G-52})$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & \dots \\ \left. \frac{\partial F_1}{\partial \tau_1} \right]_d & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ 0 & \left. \frac{\partial F_2}{\partial \tau_2} \right]_d & \left. \frac{\partial F_2}{\partial \tau_3} \right]_d & \dots \\ 0 & 0 & 0 & \dots \\ 0 & \left. \frac{\partial F_3}{\partial \tau_2} \right]_d & \left. \frac{\partial F_3}{\partial \tau_3} \right]_d & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (\text{G-59})$$

Bibliografía

- [Abd91] C. Abdallah, D.M. Dawson, P. Dorato, M. Jamshidi. "Survey of robust control for rigid robots" *Control Systems Magazine, IEEE*, 11(2), pp. 24-30, 1991.
- [Akh05] S. Akhtar, D. Bernstein. "Lyapunov-Stable Discrete-Time Model Reference Adaptive Control". *Proceedings of the American Control Conference, Portland, Oregon, USA*, pp. 3174-3179, 2005.
- [All99] F. Allgöwer, T.A. Badgwell, J.S. Qin, J.B. Rawlings, S.J. Wright. "Nonlinear predictive control and moving horizon estimation — an introductory overview". En: M.F. Paul (Ed.), *Advances in Control*, pp. 391–449. Berlin: Springer, 1999.
- [Alo03] F. Alonge, F. D'Ippolito, F.M. Raimondi. "An Adaptive Control Law for Robotic Manipulator without Velocity Feedback". *Control Engineering Practice*, 11(9), pp. 999-1005, 2003.
- [Alo04] F. Alonge, F. D'Ippolito, F.M. Raimondi. "Globally Convergent Adaptive and Robust Control of Robotic Manipulators for Trajectory Tracking". *Control Engineering Practice*, 12(9), pp. 1091-1100, 2004.
- [Alo07] F. Alonge, F. D'Ippolito, T. Raimondi. "A Control Law for Robotic Manipulators Based on a Filtered Signal to Generate PD Action and Velocity Estimates". *International Journal of Robotics and Automation*, 22 (2), pp. 2921-2936, 2007.
- [Ang89] J.M. Angulo. "Curso de Robótica". Madrid: Paraninfo, 1989.
- [Ara93] R. Araujo, G. Cook. "Comparison of Two Adaptive Control Methods for Robotic Applications". *Proc. of the International Conference on Industrial Electronics, Control, and Instrumentation (vol. 3)*, Maui, HI, USA, pp. 1819-1824, 1993.
- [Ari83] S. Arimoto, F. Miyazaki. "Stability and Robustness of PID Feedback Control for Robot Manipulators of Sensor Capability". *Proc. of the International Symposium on Robotics Research*, pp. 783-799, 1983.
- [Ark98] P.C. Arkin. "Behavior-Based Robotics". MIT Press, 1998.

- [Arm86] B. Armstrong, O. Khatib, J. Burdick. "The explicit dynamic model and inertial parameters of the puma 560 arm". Proc. IEEE International Conference on Robotics and Automation (ICRA'86), San Francisco, CA, pp. 510-518, 1986.
- [Ast95] K.J. Astrom, B. Wittenmark. "Adaptative Control". Addison-Wesley, 2nd edition, 1995.
- [Bar97] A. Barrientos, L. Peñín, C. Balaguer, R. Aracil. "Fundamentos de Robótica". McGraw Hill, 1997.
- [Bec02] V.M. Becerra, V.M. Calado, J.M.F. Calado, P.M. Silva, F. Garces. "System identification using dynamic neural networks: training and initialization aspects". Proceedings of the 15th World Congress of the IFAC, Barcelona, Spain, 2002.
- [Bec04] V.M. Becerra, C.N.J. Cage, W.S. Harwin, P.M. Sharkey. "Hardware retrofit and computed torque control of a Puma 560 Robot", IEEE Control Systems Mag., 24 (5), pp.78-82, 2004.
- [Bem02] A. Bemporad, M. Morari, V. Dua, E.N. Pistikopoulos. "The explicit linear quadratic regulator for constrained systems". Automatica, 38, pp. 3-20, 2002.
- [Boo84] W.J. Book. "Recursive Lagrangian Dynamics of Flexible Manipulators Arms". International Journal of Robotics Research, 3(3), pp. 87-101, 1984.
- [Bro88] D.S. Broomhead, D. Lowe. "Multivariable functional interpolation and adaptive networks". Complex Systems 2, pp. 321-355, 1988.
- [Cai89] L. Cai, A.A. Goldenberg. "Robust Control of Position and Force for a Robot Manipulator in Non-contact and Contact Tasks". Proc. of the American Control Conference, Pittsburgh, USA, pp. 1905-1911, 1989.
- [Cam95] E.F. Camacho, C. Bordons, "Model Predictive Control in the Process Industry", Springer-Verlag, 1995.
- [Cam97] E.F. Camacho, M. Berenguer, F.R. Rubio, "Advanced Control of Solar Plants", Springer-Verlag, 1997.
- [Car95] R. Carelli, E.F. Camacho, D. Patino. "Neural network based feedforward adaptive controller for robots". IEEE Transactions on Systems, Man, and Cybernetics. 25 (9), pp. 1281-1288, 1995.
- [Chi93] S. Chiaverini, L. Sciavicco. "The parallel approach to force / position

- control of robotic manipulators”, IEEE Transactions on Robotics and Automation, 9, pp. 361-373, 1993.
- [Chi96] P. Chiacchio, S. Chiaverini, B. Siciliano. “Direct and Inverse Kinematics for Coordinated Motion Tasks of a two-manipulator system”. Journal of Dynamic Systems, Measurement and Control, 118, pp. 691-697, 1996.
- [Cla87] D.W. Calrke, C. Mohtadi, P.S. Tuffs. “Generalised Predictive Control (Parts 1 and 2)”. Automatica, 23, pp. 137-160, 1987.
- [Cla91] D. Clarke, R. Scattolini. “Constrained Receding-horizon Predictive Control”. IEE Proceedings-D, 138 (4), pp. 347-354, 1991.
- [Cor81] M.J. Corless, G. Leitmann. “Continuous State Feedback Guaranteeing Uniform Ultimate Boundedness for Uncertain Dynamic Systems”. IEEE Transactions on Automatic Control, 26(5), pp.1139-1144, 1981.
- M.J. Corless, G. Leitmann. “Erratum to ‘Continuous State Feedback Guaranteeing Uniform Ultimate Boundedness for Uncertain Dynamic Systems’ ”. IEEE Transactions on Automatic Control, 28(2), p.249, 1983.
- [Cra86] J.J. Craig. “Introduction to robotics”. Addison-Wesley, 1986.
- [Cut80] C.R. Cutler, B.L. Ramaker. “Dynamic Matrix Control – a computer control algorithm”. Proceedings of American Control Conference, 1980.
- [Daw92] D.M. Dawson, Z. Qu, J.J. Carroll. “Tracking control of rigid-link electrically-driven robot manipulators”. Int. Journal of Control, 56. pp. 991-1006, 1992.
- [Dem91] H. Demircioglu, P.J. Gawthrop. “Continuos-time Generalized Predictive Control (CGPC)”. Automatica, 27 (1), pp. 55-74, 1991.
- [Dem92] H. Demircioglu, P.J. Gawthrop. “Multivariable Continuos-time Generalized Predictive Control (MCGPC)”. Automatica, 28 (4), pp. 697-713, 1992.
- [Dem93a] H. Demircioglu, D.W. Clarke. “Generalised Predictive Control with End-point State Weighting”. IEE Proceedings-D, 140 (4), pp. 275-281, 1993.
- [Des85] S. Desa, B. Roth. “Synthesis of Control Systems for Manipulators Using Multivariable Robust Servo Mechanism Theory”. Interantional

- Journal of Robotic Research, 4, pp. 18-34, 1985.
- [Fer07] A. Ferrara, L. Magnani. "Motion Control of Rigid Robot Manipulators via First and Second Order Sliding Modes". Journal of Intelligent and Robotic Systems, 48 (1), pp. 23-36, 2007.
- [Fre82] E. Freund. "Fast Nonlinear Control with Arbitrary Pole-placement for Industrial Robots and Manipulators". International Journal of Robotics Research, 1, pp. 65-78, 1982.
- [Fu88] K.S. Fu, R.C. González, C.S.G. Lee. "Robótica. Control, Detección, Visión e Inteligencia". McGraw Hill Interamericana, 1988.
- [Fu90] L.C. Fu, T.L. Liao. "Globally Stable Robust Tracking of Nonlinear Systems Using Variable Structure Control and with an Application to a Robotic Manipulator". IEEE Transactions on Automatic Control, 35(12), pp.1345-1350, 1990.
- [Fun93] K. Funahashi, Y. Nakamura. "Approximation of dynamical-systems by continuous-time recurrent neural networks". Neural Networks, 6 (6), pp. 801-806, 1993.
- [Gao93] W. Gao, J.C. Hung. "Variable Structure Control of Nonlinear Systems: A New Approach". IEEE Transactions on Industrial Electronics, 40(1), pp. 45-55, 1993.
- [Gar82] C.E. García, M. Morari. "Internal Model Control 1. A unifying review and some new results". I&EC Process Design and Development, 21, pp. 308-323, 1982.
- [Gar86] C.E. García, A.M. Morshedi. "Quadratic Programming Solution of Dynamic Matrix Control (QDMC)". Chemical Eng. Commun., 46, pp. 73-87, 1986.
- [Gar99] F. Garcés, C. Kambhampati, K. Warwick. "Dynamical recurrent neural networks for identification of a multivariable nonlinear evaporation system". International conference on Dynamic Control Systems, DYCONS99. World Scientific, Ottawa, 1999.
- [Gar00] F. Garcés. "Dynamic neural networks for approximate input-output linearization-decoupling of dynamic systems". PhD Thesis. University of Reading, 2000.
- [Ger01] E.M. Gertz, S.J. Wright. "Object-oriented software for quadratic programming". Mathematical Programming Technical Report, 01-02, Computer Sciences Department, University of Wisconsin-Madison, October 2001.

- [Gil87] E.G. Gilbert, I.J. Ha. "Robust tracking in nonlinear systems". IEEE Transactions on Automatic Control, 32, pp.763-771, 1987.
- [Gos97] J.R. Gossner, B. Kouvaritakis, J.A. Rossiter, "Stable Generalised Predictive Control with Constraints and Bounded Disturbances". Automatica, 33 (4), pp. 551-568, 1997.
- [Gro89] M.P. Groover, M. Weiss, R.N. Nagel, N.G. Odrey. "Robótica Industrial: Tecnología, programación y Aplicaciones". McGraw Hill Interamericana, 1989.
- [Gri03] P. Grieder, M. Lüthi, P.A. Parrillo, M. Morari, "A Stability & Feasibility of Constrained Receding Horizon Control". En: Proceedings of the European Control Conference, Cambridge, UK, 2003.
- [Gri04] P. Grieder, F. Borrelli, F.D. Torrisi, M. Morari. "Computation of the constrained infinite time linear quadratic regulator. Automatica, 40. pp. 701–708, 2004.
- [Gru89] R.A. Grupen, T.C. Henderson, I.D. McCammon. "A Survey of General-Purpose Manipulation". International Journal of Robotics Research, 8(1), pp. 38-62, 1989.
- [Has07] I. Hassanzadeh, A. Harifi, F. Arvani. "Design and Implementation of an Adaptive Control for a Robot". American Journal of Applied Sciences, 4 (2), pp. 56-59, 2007.
- [Hea05] W.P. Heath, A.G. Wills. "The inherent robustness of constrained linear model predictive control". Proceedings of the 16th IFAC World Congress, Prague, 2005.
- [Hin86] G.E. Hinton. "Learning distributed representations of concepts". 8th Annual Conference of the Cognitive Science Society. Amherst. Mass, 1986.
- [Hog85] N. Hogan. "Impedance control: An approach to manipulation". Journal of Dynamic Systems, Measurement and Control, 107, pp. 1-24, 1985.
- [Hop82] J.J. Hopfield. "Neural networks and physical systems with emergent collective computational abilities". Proceedings of the National Academy of Sciences of the USA – Biological Sciences, 79 (8), pp. 2554-2558, 1982.
- [Hop84] J.J. Hopfield. "Neurons with graded response have collective computational properties like those of 2-state neurons". Proceedings of the National Academy of Sciences of the USA –

- Biological Sciences, 81 (10), pp. 3088-3092, 1984.
- [Hop85] J.J. Hopfield, D.W. Tank. "Neural computation of decisions in optimization problems". *Biological Cybernetics*, 52 (3), pp. 141-152, 1985.
- [Hop86] J.J. Hopfield, D.W. Tank. "Computing with neural circuits – a model". *Science*, 233 (4764), pp. 625-633, 1986.
- [Hsi86] T.C. Hsia, "Adaptive Control of Robot Manipulators – A Review". *IEEE International Conference on Robotics and Automation*, 3, pp. 183-189, 1986.
- [Hun83] L.R. Hunt, R. Su, G. Mayer. "Global transformations of nonlinear systems". *IEEE Transactions on Automatic Control*, 28, pp. 24-31, 1983.
- [Hut96] S. Hutchinson, G. Hager, P. Corke. "A tutorial on visual servo control", *IEEE Transactions on Robotics and Automation*, San Diego, CA, pp. 3211-3216, 1996.
- [Ish01] C. Ishii, T. Shen, Z. Qu. "Lyapunov recursive design of robust adaptive tracking control with L2-gain performance for electrically-driven robot manipulators", *Int. Journal of Control*, 74 (8), pp. 811-828, 2001.
- [Jar96] A. Jaritz, M.W. Spong. "An Experimental Comparison of Robust Control Algorithms on a Direct Drive Manipulator". *IEEE Transactions on Control Systems Technology*, 4(6), pp. 627-640, 1996.
- [Jat03] F. Jatta, G. Legnani, A. Visioli. "Hybrid Force/Velocity Control of Industrial Manipulators with elastic transmissions". *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, Nevada, pp. 3276-3281, 2003.
- [Jat06] F. Jatta, G. Legnani, A. Visioli, G. Ziliani. "On the use of Velocity Feedback in Hybrid Force/Velocity Control of Industrial Manipulators". *Control Engineering Practice*, 14, pp. 1045-1055, 2006.
- [Jin98] Y. Jin. "Decentralized adaptive fuzzy control of robot manipulators", *IEEE Transactions on Systems, Man and Cybernetics – B: Cybernetics*, 28 (1), pp. 47-57, 1998.
- [Kel03] R. Kelly, V. Santibáñez. "Control de Movimiento de Robots Manipuladores". Madrid: Pearson Education, 2003.

- [Ker04] E.C. Kerrigan, J.M. Maciejowski. "Feedback min-max model predictive control using a simple linear program: robust stability and the explicit solution". *International Journal of Robust Nonlinear Control*, 14, pp. 395-413, 2004.
- [Kho88] P.K. Khosla, T. Kanade. "Experimental evaluation of nonlinear feedback and feedforward control schemes for manipulators". *International Journal of Robotics Research*, 7 (1), pp.18-28, 1988.
- [Kim98] M. Kimura, R. Nakano. "Learning dynamical systems by recurrent neural networks from orbits". *Neural Networks* 11 (9), pp. 1589-1599, 1998.
- [Koh89] T. Kohonen. "Self-organization and associative memory". 3rd ed. Springer-Verlag: Berlin; New York, 1989.
- [Koi94] P. Koiran. "Dynamics of discrete-time, continuous state Hopfield networks". *Neurla Computation*, 6 (3), pp. 459-468, 1994.
- [Koo94] K.M. Koo, J.H. Kim. "Robust Control of Robot Manipulators with Parametric Uncertainty". *IEEE Transactions on Automatic Control*, 39(6), pp. 1230-1233, 1994.
- [Kou92] B. Kouvaritakis, J.A. Rossiter, A.O.T. Chang. "Stable Generalised Predictive Control: an algorithm with guaranteed stability". *IEE Proceedings-D*, 139 (4), pp. 349-362, 1992.
- [Kou93] B. Kouvaritakis, J.A. Rossiter. "Multivariable Stable Generalised Predictive Control". *IEE Proceedings-D*, 140 (5), pp. 364-372, 1993.
- [Kou97] B. Kouvaritakis, J.A. Rossiter, M. Cannon, C.K.J. Wong. "Constrained stable predictive control: an l_2 -optimal approach". *Proceedings of the 5th IEEE MED Conference*, Chipre, 1997.
- [Kou98] B. Kouvaritakis, J.A. Rossiter, M. Cannon. "Linear quadratic feasible predictive control". *Automatica*, 34 (12), pp.1583-1592, 1998.
- [Kou00] B. Kouvaritakis, M. Cannon, J.A. Rossiter. "Removing the need for QP in constrained predictive control", *Proceedings of Advanced Control of Chemical Processes (ADCHEM) 2000*, Pisa, Italia, 2000.
- [Kre89] K. Kreutz. "On Manipulator Control by Exact Linearization", *IEEE Transactions on Automatic Control*, 34(7), pp. 763-767, 1989.
- [Krs95] M. Krstic, I. Kanellakopoulos, P.V. Kokotovic. "Nonlinear and Adaptive Control Design", Wiley, New York, 1995.

- [Lam07] R. Lamanna, R. Grimón. "Control neuronal predictivo por linealización instantánea". *Revista Iberoamericana de Automática e Informática Industrial (RIAI)*, 4 (2), pp. 90-97, 2007.
- [Lee85] C.S.G. Lee, M.J. Chung, "Adaptive perturbation control with feedforward compensation for robot manipulators". *Simulation*, 44 (3), pp. 127-136, 1985
- [Lee94] J.H. Lee, M. Morari, C. García. "State-space Interpretation of Model Predictive Control". *Automatica*, 30 (4), pp. 707-717, 1994.
- [Lee97] J.H. Lee, W.H. Kwon, J. Choi. "On Stability of constrained receding horizon control with finite terminal weighting matrix". *Proceedings of the European Control Conference 97*, Bruselas, Bélgica, 1997.
- [Leu91] T.P. Leung, Q.J. Zhou, C.Y. Su. "An Adaptive Variable Structure Model Following Control Design for Robot Manipulators". *IEEE Transactions on Automatic Control*, 36(3), pp. 347-352, 1991.
- [Lew04] F. Lewis, D. Dawson, C. Abdallah, "Robot manipulator control". New York: Marcel Dekker, 2004.
- [Liu93] G. Liu, A. Goldenberg. "On robust saturation control of robot manipulators", in *Proceedings of the 32nd Conference on Decision and Control*, Texas, USA, pp. 2115-2120, 1993.
- [Liu94] G. Liu, A.A. Goldenberg. "Asymptotically Stable Robust Control of Robot Manipulators", in *Proceedings of the International Conference on Robotics and Automation*, 4, San Diego, USA, pp. 2968-2973, 1994.
- [Lon03] N. Londoño, M Arias. "Control de posición de un manipulador utilizando lógica difusa". *XI Congreso Latinoamericano de Control Automático*, Cuba, 2003.
- [Luh83] J.Y.S. Luh. "Conventional controller design for industrial robots - a tutorial". *IEEE Transactions on Systems, Man and Cybernetics*, 13 (3), pp. 298-316, 1983.
- [Luh87] J.Y.S. Luh, Y.F. Zheng. "Constrained Relations Between Two Coordinated Industrial Robots for Motion Control". *International Journal of Robotics Research*, 6(3), pp. 60-70, 1987.
- [Ma85] Z. Ma, C.S.G. Lee, "Adaptive perturbation control with compensation for disturbances for robot manipulators". *Proceedings of the 24th IEEE Conference on Decision and Control*, 24, pp. 1510-1515, 1985.

- [Mac02] J.M. Maciejowski. Predictive Control with Constraints. Prentice Hall, 2002.
- [Mas85] M.T. Mason, J.K. Salisbury, "RobotHands and the Mechanic of Manipulation". MIT Press, Cambridge, MA, 1985.
- [Mat92] K. Matsuoka. "Stability conditions for nonlinear continuous neural networks with asymmetric connection weights". Neural Networks, 5 (3), pp- 495-500, 1992.
- [Men80] G. Menga, E. Mosca. "MUSMAR: Multivariable Adaptive Regulators Based on Multistep Cost Functionals". Advances in Control, D.G. Lainiotis y N.S. Tzannes (eds.), pp. 334-341. Reidel, Dordrecht, Holland, 1980.
- [Men98] J.A. Méndez. "Desarrollo de estrategias de control predictivas de alto rendimiento mediante la incorporación de técnicas de control robusto y de redes neuronales". Tesis Doctoral, Dep. de Física Fundamental y Experimental, Universidad de La Laguna, 1998.
- [Men00] J.A. Méndez, B. Kouvaritakis, J.A. Rossiter, "State-space approach to interpolation in MPC". International Journal of Robust and Nonlinear Control, 10, pp. 27-38, 2000.
- [Men02] M.F. Mendes, W. Kraus Jr., E. R. de Pieri. "Variable Structure Position Control of an industrial Robotic Manipulator". Journal of the Brazilian Society of Mechanical Sciences, 24(3), 2002.
- [Mid88] R.H. Middleton, G.C. Goodwin. "Adaptive computed torque control for rigid link manipulations". Systems and Control Letters, 10 (1), pp. 9-16, 1988.
- [Mil99] D.M. Miljanovic, E.A. Croft. "A Taxonomy for Robot Control". Proc. of the IEEE International Conference on Robotics and Automation, Detroit, Michigan, pp. 176-181, 1999.
- [Mor99] M. Morari, J.H. Lee. "Model predictive control: past, present and future". Computers and Chemical Engineering, 23, pp. 667-682, 1999.
- [Mos90] E. Mosca, J.M. Lemos, J. Zhang. "Stabilizing I/O Receding Horizon Control". IEEE Conference on Decision and Control, 1990.
- [Mus93] K.R. Muske, J.R. Rawlings. "Model predictive control with linear models". American Institute of Chemical Engineers (AIChE) Journal, 39 (2), pp. 262-287, 1993.

- [Oll01] A. Ollero. "Robótica. Manipuladores y robots móviles". Marcombo, 2001.
- [Ort88] R. Ortega, M.W. Spong. "Adaptive Motion Control of Rigid Robots: A Tutorial". Proc. of the IEEE Conference on Decision and Control, Austin, Texas, USA, pp. 1575-1584, 1988.
- [Ort89] R. Ortega, M.W. Spong "Adaptive motion control of rigid robots: A tutorial", Automatica, 25, pp. 877-888, 1989.
- [Pau81] R.P. Paul. "Robot Manipulators". MIT Press, 1981.
- [Qin96] S.J. Qin, T.A. Badgwell. "An overview of industrial predictive control technology". Proceedings of the 5th International conference on Chemical Process Control (CPC-V), pp.232-256, Tahoe City, CA, USA, 1996.
- [Qin00] S.J. Qin, T.A. Badgwell. "An overview of nonlinear model predictive control applications". En: Nonlinear Predictive Control, editores: F. Allgöwer y A. Zheng, pp. 369–392. Birkhäuser, 2000.
- [Que04] D.E. Quevedo, G.C. Goodwin, J.A. De Doná. "Finite Constraint Set Receding Horizon Quadratic Control". International Journal of Robust Nonlinear Control, 14 (4), pp. 355-377, 2004.
- [Rai81] M.H. Raibert, J.J. Craig. "Hybrid position/force control of manipulators", Journal of Dynamic Systems, Measurement and Control, 103(2), pp. 126-133, 1981.
- [Ree88] J. Reed, P. Ioannou. "Instability Analysis and Robust Adaptive Control of Robot Manipulators". Proc. of the IEEE Conference on Decision and Control, Austin, Texas, USA, pp. 1607-1612, 1988.
- [Ric76] J. Richalet, A. Rault, J.L. Testud y J. Papon. "Algorithmic Control of Industrial Processes". 4th IFAC Symposium on Identification and System Parameter Estimation. Tbilisi (URSS), 1976.
- [Ric78] J. Richalet, A. Rault, J.L. Testud y J. Papon. "Model Predictive Heuristic Control: Application to Industrial Processes". Automatica, 14 (5), pp. 413-428, 1978.
- [Ric87] J. Richalet, S. Abu el Ata-Doss, C. Arber, H.B. Kuntze, A. Jacobash, W. Schill. "Predictive Functional Control. Application to Fast and Accurate Robots". Proceedings of the 10th IFAC Congress, Munich, 1987.
- [Rod96] F. Rodríguez Rubio, M.J. López Sánchez. "Control Adaptativo y

- Robusto". Sevilla: Universidad, 1996.
- [Roj03] O.J. Rojas, G.C. Goodwin, A. Feuer, M.M. Serón. "A sub-optimal receding horizon control strategy for constrained linear systems". En: Proceedings of the American Control Conference 77, Denver, Colorado, 2003.
- [Ros93] J.A. Rossiter, B. Kouvaritakis, "Constrained Stable Generalised Predictive Control". IEE Proceedings-D, 140 (4), pp. 243-254, 1993.
- [Ros95] J.A. Rossiter, B. Kouvaritakis, J.R. Gossner. "Feasibility and Stability Results for Constrained Stable Generalised Predictive Control". Automatica, 31 (6), pp. 863-877, 1995.
- [Ros96] J.A. Rossiter, B. Kouvaritakis, "Guaranteeing Feasibility in Constrained Stable Generalised Predictive Control". IEE Proceedings-D, 143 (5), pp. 463-469, 1996.
- [Ros98] J.A. Rossiter, M.J. Rice, J. Schuurmans, B. Kouvaritakis. "A computationally efficient constrained predictive control law". American Control Conference 98, Philadelphia, USA, 1998.
- [Ros98b] J.A. Rossiter, B. Kouvaritakis, M.J. Rice. "A numerically robust state-space approach to stable predictive control strategies". Automatica, 34 (1), pp. 65-73, 1998.
- [Ros03] J.A. Rossiter. "Model-based Predictive Control – A Practical Approach". CRC Press LLC, Florida, 2003.
- [Ros04a] J.A. Rossiter, B. Kouvaritakis, M. Bacic, "Interpolation based computationally efficient predictive control". Int. Journal of Control, 77 (3), pp. 290-301, 2004.
- [Ros04b] J.A. Rossiter, B. Kouvaritakis, M. Bacic, "Interpolation based predictive control". Proceedings of the 2004 American Control Conference, Boston, Massachusetts, USA, pp. 228-232, 2004.
- [Ros05] J.A. Rossiter, B. Pluymers, B. De Moor, "The potential of interpolation for simplifying predictive control and application to LPV systems", International Workshop on Assessment and Future Directions of NMPC, Stuttgart, Alemania, 2005.
- [Rum86] D.E. Rumelhart, J.L. McClelland. "Parallel distributed processing: explorations in the microstructure of cognition". MIT Press. Cambridge, Mass, 1986.
- [San99] E.N. Sánchez, J.P. Pérez. "Input-to-state stability (ISS) analysis for

- dynamic neural networks". IEEE Transactions on Circuits and Systems I – Fundamental Theory and Applications, 46 (11), pp. 1395-1398, 1999.
- [Sch88] J. de Schutter, H. Van Brussel, "Compliant robot motion II. A control approach based on external control loops", International Journal of Robotics Research, 7 (4), pp. 18-33, 1988.
- [Sci00] L. Sciavicco, B. Siciliano. "Modelling and Control of Robot Manipulators". London: Springer-Verlag, 2nd edition, 2000.
- [Sco98] P.O.M. Sckaert, B. Rawlings. "Constrained linear quadratic regulation". IEEE Transactions on Automatic Control, 43 (8), pp. 1163-1169, 1998.
- [Sep97] R. Sepulchre, M. Jankovic, P.V. Kokotovic. "Constructive Nonlinear Control". Springer-Verlag, London, 1997.
- [Ser00] M.M. Serón, J.A. De Don, G.C. Goodwin. "Global analytical model predictive control with input constraints". En: Proceedings of the 39th IEEE Conference on Decision and Control, pp. 154-159, Sydney, Australia, 2000.
- [Sej86] T.J. Sejnowski, P.K. Kienker, G.E. Hinton. "Learning symmetry groups with hidden units – beyond the perceptron". Physica D 22 (1-3), pp. 260-275, 1986.
- [Sha96] P. Sharkey, D. Murray. "Dealy versus performance of visually guided systems". IEE Proc. on Control Theory and Applications, 143, pp. 436-447, 1996.
- [Sic88] B. Siciliano, W.J. Book. "A Singular Perturbation Approach to Control of Lifeweight Flexible Manipulators". International Journal of Robotics Reaserach, 7(4), pp. 79-90, 1980.
- [Sic96] B. Siciliano, L. Villani. "Adaptive Compliant Control of Robot Manipulators". Control Engineering Practice, 4(5), pp.705-712, 1996.
- [Sic97] B. Siciliano, L. Villani, "Six-degree-of-freedom impedance robot control", Proc. of the 8th International Conference on Advanced Robotics, Monterrey, CA, pp. 387-392, 1997.
- [Sic98] B. Siciliano. "Control in Robotics: Open Problems and Future Directions". Proc. of the IEEE International Conference on Control Applications, Trieste, Italy, pp. 81-85, 1998.

- [Sin85] S.N. Singh. "Adaptive Model Following Control of Nonlinear Robotic Systems". IEEE Transactions on Automatic Control, 30(11), pp. 1099-1100, 1985.
- [Slo85] J.J.E. Slotine, "Robust control of robot manipulators". Int Journal of Robotics Research, 4, pp. 49-64, 1985.
- [Slo87] J.J. Slotine, W. Li. "On the Adaptive Control of Robot Manipulators". International Journal of Robotic Research, 6(3), pp.49-59, 1987.
- [Slo88] J.J.E. Slotine. "Putting Physics in Control – The Example of Robotics". IEEE Control Systems Magazine, 8, pp. 12-17, 1988.
- [Sor94] O. Sorensen. "Neural Networks performing system identification for control applications". 3rd International Conference on Artificial Neural Networks, IEE Brighton, UK, pp. 172-176, 1994.
- [Spo87a] M.W. Spong, M. Vidyasagar. "Robust Linear Compensator Design for Nonlinear Robotic Control". IEEE Journal of Robotics and Automation, 3, pp. 345-351, 1987.
- [Spo87b] M.W. Spong. "Modelling and Control of Elastic Joint Robots". ASME Journal of Dynamic Systems, Measurement, and Control, 109, pp. 310-319, 1987.
- [Spo87c] M.W. Spong, J.S. Thorp, J.M. Kleinwaks. "Robust microprocessor control of robot manipulators". Automatica, 23, pp. 373-379, 1987.
- [Spo89] M.W. Spong, M. Vidyasagar. "Robot Dynamics and Control". John Wiley & Sons, 1989.
- [Spo92] M.W. Spong. "On the Robust Control of Robot Manipulators". IEEE Transactions on Automatic Control, 37(11), pp. 1782-1786, 1992.
- [Spo93] M.W. Spong. "Adaptive Control of Robot Manipulators: Design and Robustness". Proc. of the American Control Conference, San Francisco, California, pp. 2826-2830, 1993.
- [Spo03] M.W. Spong. "Robust and Adaptive Motion Control of Manipulators". Comunicación personal, 2003.
- [Su93] C.Y. Su, T.P. Leung. "A Sliding Mode Controller with Bound Stimulation for Robot Manipulators". IEEE Transactions on Robotics and Automation, 9(2), pp. 208-214, 1993.
- [Su97] C.Y. Su, Y. Stepanenko. "Redesign of hybrid adaptive/robust

- motion control of rigid-link electrically-driven robot manipulators". IEEE Transactions on Robotics and Automation, 14, pp. 651-655, 1997.
- [Tar84] T.J. Tarn, A.K. Bejczy, A. Isidori, Y. Chen. "Nonlinear Feedback in Robot Arm Control". Proc. of the IEEE Conference on Decision and Control, Las Vegas, USA, 1984.
- [Tan92] K.T. Tan, E.G. Gilbert, "Multimode controllers for linear discrete-time systems with general state and control constraints". Proceedings of the Optimisation Techniques and Applications World Scientific Congress, pp. 433-442, Singapur, 1992.
- [Ton03] P. Tondel, T.A. Johansen, A. Bemporad. "An algorithm for multi-parametric quadratic programming and explicit MPC solutions". Automatica, 39, pp. 489-497, 2003.
- [Tor00] S. Torres, J.A. Méndez, L. Acosta, M. Sigut, G.N. Marichal, L. Moreno. "A decentralized fuzzy controller for a three link robot manipulator". Proc. of the ICSC 2nd Symposia on Engineering of Intelligent Systems. Paisley, Scotland, UK. 2000.
- [Tor02] F. Torres, J. Pomares, P. Gil, S. Puente, R. Aracil. "Robots y Sistemas Sensoriales". Prentice Hall, 2002.
- [Tos88] S. Tosunoglu, D. Tesar. "State of the Art in Adaptive Control of Robotic Systems". IEEE transactions on Aerospace and Electronic Systems, 24(5), pp. 552-561, 1988.
- [Uch93] M. Uchiyama, P. Dauchez. "Symmetric kinematic formulation and non-master/slave coordinated control of two-arm robots". Advanced Robotics, 7, pp. 361-383, 1993.
- [Uic64] J.J. Uicker, J. Denavit, R.S. Hartenberg. "An iterative method for the displacement analysis of spatial mechanisms". ASME Journal of Applied Mechanics, 86 (2), pp. 309-314, 1964.
- [Vri94] R.A.J. de Vries, H.B. Verbruggen, "Multivariable Unified Predictive Control". Advances in Model-Based Predictive Control, pp. 84-102. D.W. Clarke, editor. Oxford University Press, 1994.
- [Wan88] S.P.T. Wang, C.Y. Kuo. "Nonlinear Robust Hybrid Control of Robot Manipulators". Proc. of the IEEE International Conference on Robotics and Automation (vol. 2), Albuquerque, New Mexico, pp. 1204-1209, 1988.
- [Wan03] Z. Wan, M.V. Kothare. "An efficient off-line formulation of robust

- modelo predictive control using linear matrix inequalities". *Automatica*, 39, pp. 837-846, 2003.
- [Whi85] D.E. Whitney, "Historical Perspective and State of the Art in Robot Force Control", *Proc. of the IEEE International Conference on Robotics and Automation* (vol. 2), St. Louis, US, pp. 262- 268, 1985.
- [Whi93] L.L. Whitcomb, A.A. Rizzi, D.E. Koditschek. "Comparative Experiments with a New Adaptive Controller for Robot Arms". *IEEE Transactions on Robotics and Automation*, 9(1), pp. 59-70, 1993.
- [Yos00] T. Yoshikawa, "Force Control of Robot Manipulators", *Proc. of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, pp. 220-226, 2000.
- [You78] K.K.D. Young. "Controller Design for a Manipulator Using Theory of Variable Structure Systems". *IEEE Transactions on Systems, Man, and Cybernetics*, 8, pp. 101-109, 1978.
- [You00] B.J. Young, J.R. Lawton, R.W. Beard. "Two Hybrid Control Schemes for Nonholonomic Robots", *Proc. of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, pp. 1824-1829, 2000.
- [Zha85] H. Zhang, R.P. Paul. "Hybrid Control of Robot Manipulators". *Proc. of the IEEE International Conference on Robotics and Automation* (vol. 2), St. Louis, US, 1985, pp. 602- 607.
- [Zha07] Y.S. Zhao, Y.Q. Zhang, J. Yang, L.P. Chen. "Enhanced Fuzzy Sliding Mode Controller for Robotic Manipulators". *International Journal of Robotics and Automation*, 22 (2), pp. 2921-2936, 2007.
- [Zhe95] Z.Q. Zheng, M. Morari. "Stability of model predictive control with mixed constraints". *IEEE Transactions on Automatic Control*, 40 (10), pp. 1818-1823, 1995.