

Seguimiento de Trayectorias en Robots Manipuladores: Revisión de Soluciones y Nuevas Propuestas

S. Torres, J.A. Méndez

*Departamento de Ingeniería de Sistemas y Automática y Arquitectura y Tecnología
de Computadores. Universidad de La Laguna.
Av. Astrof. Sánchez, s/n, Ed. de Física y Matemáticas. 38271, Tenerife, España.
E-mail: {storres, jamendez}@ull.es*

Resumen: El problema del seguimiento de trayectoria en robots manipuladores ha sido abordado aplicando una gran variedad de controladores, desde estructuras sencillas basadas en PD hasta otras más complejas basadas en controladores adaptativos y robustos. Estas últimas técnicas presentan inconvenientes como la presunción de ciertas cotas en los términos de la ecuación dinámica del robot o la no inclusión de las ligaduras del sistema en el algoritmo de control. En el presente trabajo se hace una revisión de las técnicas clásicas de control de manipuladores y se introduce un conjunto de técnicas novedosas de control robusto y de control predictivo, con las que se evitan los problemas comentados. En particular se describe un controlador con una acción robusta autoadaptativa, necesaria para evitar los errores en la cancelación de términos no lineales de la dinámica del robot. Este esquema es mejorado mediante técnicas predictivas que permiten la inclusión de las ligaduras de movimiento del robot en el algoritmo de control. Se incluyen resultados reales y en simulación en un robot PUMA-560 de Unimation que prueban la bondad de dichos controladores. Copyright © 2009 CEA.

Palabras Clave: Manipuladores robot, Control Robusto, Control Adaptativo, Control Autoadaptativo, Control Predictivo, Implementación, Tareas en tiempo real.

1. INTRODUCCIÓN

El problema de controlar el movimiento de robots manipuladores ha sido abordado tradicionalmente con controladores PID, controladores PD con compensación de gravedad o técnicas basadas en control dinámico inverso (Barrientos, 2007). Incluso han sido propuestas estrategias basadas en control difuso (Jin, 1998, Torres et al., 2000; Londoño y Arias, 2003).

Los controladores PID y difuso no tienen en cuenta la dinámica del sistema, pero sólo son excepciones, porque se utilizan únicamente para resolver el problema de regulación, por ejemplo, el movimiento de un punto a otro punto sin imponer el seguimiento de una trayectoria específica. El resto de controladores, empleados para solucionar el problema del seguimiento de trayectoria, utilizan el modelo dinámico para calcular la acción de control. Así, por ejemplo, con el modelo se evalúa el término de gravedad en el caso de compensación de gravedad o el término no lineal en el caso de control dinámico inverso (Freund, 1982), denominado en robótica como linealización por realimentación.

Esta última técnica está basada en un conocimiento exacto del modelo del robot y de sus parámetros dinámicos. Si se satisfacen estas dos premisas, se consigue una cancelación perfecta de la dinámica no lineal. Con ello se pueden aplicar controladores lineales que presentan en ese caso un rendimiento satisfactorio (Freund, 1982; Kreutz, 1989).

Otras técnicas que posibilitan la aplicación de controladores lineales son la linealización local en torno a la trayectoria deseada (Desa and Roth, 1985; Torres et al., 2002), la linealización haciendo uso de la propiedad de parametrización

lineal del robot rígido (Reed and Ioannou, 1988; Spong, 1992), y las técnicas basadas en la propiedad de pasividad (Spong, 1993; Alonge, 2003), las cuales reducen el costo computacional de las técnicas anteriores.

Sin embargo, los esquemas de control más utilizados son los controladores robustos (Tarn, 1984; Slotine, 1985; Spong and Widyasagar, 1987; Dawson et al., 1992; Spong, 1992), entre los cuales también se encuentran los controladores de estructura variable (Fu, 1990; Su, 1993; Mendes, 2002) o el control por saturación (Spong, 1987; Spong, 1992; Liu, 1993; Jaritz, 1996). También se han propuesto un extenso número de trabajos relacionados con esquemas de control adaptativo (Slotine and Li, 1987; Ortega and Spong, 1989) y de control híbrido (Su and Stepanenko, 1997).

En el presente trabajo el problema a solucionar es la cancelación imperfecta de la dinámica no lineal debida a incertidumbres para el caso de controladores robustos. Se considera que existen varias clases de incertidumbres en un sistema robótico (Ishii et al., 2001), 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) y, finalmente, c) incertidumbres tales como el ruido de medida.

Las perturbaciones externas no están consideradas en la clasificación anterior puesto que su aparición es puntual y se tratan de rechazar 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.

La mayoría de los controladores robustos están basados en el método directo de Lyapunov (Slotine and Li, 1987). 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 (Spong, 1992). Finalmente, se añade una acción robusta basada en las condiciones previas, usando 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.

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 tienen distintos parámetros dinámicos. La primera razón es la más común. Habitualmente el brazo manipulador debe transportar masas diferentes entre puntos diferentes 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. El presente trabajo introduce un esquema adaptativo para sintonizar automáticamente el parámetro de diseño robusto involucrado en esta acción robusta.

Por otra parte, los manipuladores están sujetos a limitaciones físicas impuestas al movimiento de cada una de las articulaciones, así como restricciones en las entradas máximas (o bien en la variación máxima de dichas entradas), debido a las limitaciones electrónicas asociadas a los actuadores. Evitar las situaciones de saturación de las entradas es fundamental en este caso, pues así se salvan problemas de estabilidad en el lazo cerrado de control. Con este objetivo, se presenta un controlador robusto que incorpora técnicas de control predictivo. Estas técnicas incorporan fácilmente las ligaduras del sistema a la ley de control.

En el caso de un sistema de dinámica tan rápida como es el manipulador, es de suma importancia el disponer de algoritmos computacionalmente eficientes. El control predictivo por interpolación (Kouvaritakis, 1998; Rossiter, 1998; Méndez, 2000) plantea leyes de control basadas en interpolaciones entre soluciones óptimas y soluciones factibles, las cuales son computacionalmente eficientes debido a la necesidad de resolver simplemente un problema de programación lineal o cuadrático. Por esta razón, en este trabajo se plantea el diseño de un controlador robusto predictivo implementado mediante estas técnicas de interpolación.

Este artículo se presenta dividido en las siguientes secciones. En la sección 2 se describen estrategias clásicas de control robusto y se hace una revisión de los distintos métodos para la sintonización de los mismos. En la sección 3 se describe una nueva estrategia de control robusto adaptativa, denominada ARC, y se muestran los resultados obtenidos con la misma en un robot PUMA-560. La sección 4 presenta un nuevo controlador robusto que incluye las ligaduras presentes en el sistema basado en técnicas de control predictivo. Este controlador se denomina RIAPC. A continuación, en la sección 5, se presentan resultados

obtenidos con esta estrategia comparados con estrategias clásicas de control robusto. Finalmente se presentan las conclusiones y líneas de trabajo actuales derivadas de este trabajo.

2. ESTRATEGIAS PARA EL SEGUIMIENTO ROBUSTO DE TRAYECTORIAS

Como se comentó en el apartado anterior, para conseguir un rendimiento satisfactorio en el seguimiento de trayectorias, se debe incorporar algún término de acción robusta para rechazar las perturbaciones que afectan al sistema.

La estructura básica de las técnicas de control robusto consta de tres partes: un bucle interno de linealización por realimentación, una ley de control estabilizante y una acción robusta. El esquema de control se puede observar en el diagrama de la figura 1, donde $x = [\theta \ \dot{\theta}]^T$ es el estado del robot, siendo θ el vector de posiciones de cada una de sus articulaciones, y el subíndice d expresa los valores de referencia.

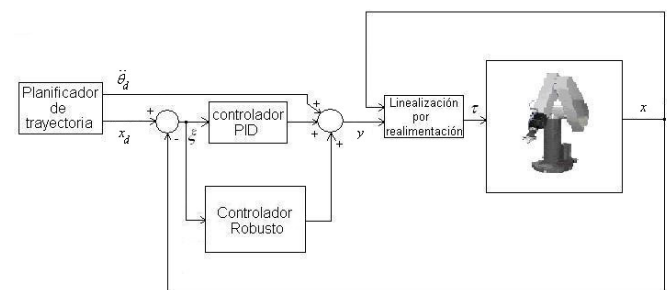


Figura 1: Esquema de control robusto.

Para el diseño del controlador, el modelo dinámico que se considerará para el manipulador vendrá representado por el siguiente conjunto de ecuaciones diferenciales no lineales y acopladas:

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

siendo $B(\theta(t))$ la matriz de inercia, $h(\theta(t), \dot{\theta}(t))$ el vector de fuerza centrífuga y de Coriolis, $c(\theta(t))$ el vector de fuerza gravitacional y $\tau(t)$ el vector de pares aplicados a las articulaciones.

2.1 Linealización del sistema.

El primer paso consiste en la linealización del modelo usando el conocido esquema de linealización por realimentación. 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:

$$\tau_k = B(\theta_k)y_k + h(\theta_k, \dot{\theta}_k) + c(\theta_k) \quad (2)$$

donde y_k es la nueva entrada al modelo lineal resultante en el instante k . θ_k y $\dot{\theta}_k$ se refieren a la posición y velocidad medidas. Se obtiene una cancelación satisfactoria de los términos no lineales 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:

$$y_k = \ddot{\theta}_k \quad (3)$$

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 control dinámico inverso (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 carga en la dinámica del sistema.

2.2 Ley de control.

Una vez se tiene el sistema linealizado, se define la ley de control, la cual consta de dos términos: un primer término estabilizante y una acción robusta que cancele los errores producto de la inexactitud en la cancelación de los términos no lineales del paso anterior (linealización por realimentación).

Una posibilidad para la ley de control estabilizante (Spong, 1992) es incluir en la misma 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:

$$y_k = \ddot{\theta}_{k,d} + K_D \dot{\tilde{\theta}}_k + K_p \tilde{\theta}_k \quad (4)$$

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

$$\ddot{\tilde{\theta}}_k + K_D \dot{\tilde{\theta}}_k + K_p \tilde{\theta}_k = 0 \quad (5)$$

Este sistema es asintóticamente estable si K_p y K_D son matrices definidas positivas. Además escogiendo una forma diagonal para estas matrices, el sistema resulta desacoplado. Cuando la posición y la velocidad inicial son nulas, K_p y K_D dan la velocidad de convergencia a cero del error de posición.

El segundo término de la ley de control, la acción robusta, es añadido para corregir la compensación imperfecta del término no lineal en (1), dado por el control de dinámica inversa (2). En la asunción de que sólo pueden ser obtenidas unas estimaciones de las matrices reales $B(\theta)$, $h(\theta, \dot{\theta})$ y $c(\theta)$, la ecuación (5) se convierte ahora en:

$$\ddot{\tilde{\theta}}_k + K_D \dot{\tilde{\theta}}_k + K_p \tilde{\theta}_k = \eta_k \quad (6)$$

donde η_k expresa las discrepancias entre los valores reales y los

estimados de las matrices en el instante de tiempo k (Sciavicco and Siciliano, 1996). 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.

Siguiendo el conocido método directo de Lyapunov (Slotine and Li, 1987), se puede diseñar un bucle de realimentación externo para el error que sea robusto a la incertidumbre η_k . Aplicando este método, el bucle externo viene dado por:

$$y_{k,r} = \frac{\rho}{\|D^T Q \xi_k\|} D^T Q \xi_k \quad (7)$$

donde $D_{2n \times n} = \begin{bmatrix} 0_{n \times n} & I_{n \times n} \end{bmatrix}^T$, $Q_{2n \times 2n}$ es una matriz definida positiva, $\xi_{2n \times 1} = \begin{bmatrix} \tilde{\theta}_{n \times 1} & \dot{\tilde{\theta}}_{n \times 1} \end{bmatrix}^T$ y ρ es un parámetro de diseño. n es el número de articulaciones del manipulador.

La ecuación (7) presenta problemas cuando el error se aproxima a cero. Para evitarlo se utiliza la siguiente expresión:

$$y_{k,r} = \begin{cases} \frac{\rho}{\|D^T Q \xi_k\|} D^T Q \xi_k & , \text{ si } \|D^T Q \xi_k\| \geq \varepsilon \\ \frac{\rho}{\varepsilon} D^T Q \xi_k & , \text{ si } \|D^T Q \xi_k\| < \varepsilon \end{cases} \quad (8)$$

donde ε viene determinada por las pruebas finales con el sistema real. La ley original (7) posee la propiedad de que todas las trayectorias del sistema bajo dicha ley de control alcanzan el subespacio $z_k = D^T Q \xi_k = 0$, denominado subespacio deslizante.

En este subespacio, la ley de control (7) cambia a una frecuencia idealmente infinita y el error tiende a cero con un transitorio que depende de Q , K_p y 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 (8) elimina el problema de las altas frecuencias de conmutación 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 $z_k = 0$, cuyas dimensiones dependen de la elección de ε .

Finalmente, la ley de control completa viene dada por:

$$y_k = \ddot{\theta}_{k,d} + K \xi_k + y_{k,r} \quad (9)$$

siendo $K = [K_p \quad K_D]$.

2.3. Sintonización del controlador robusto

El valor del parámetro de diseño ρ es importante para tener un buen comportamiento del sistema en lazo cerrado. Esto fue verificado mediante un estudio de simulación que confirmó los siguientes aspectos: 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.

En la literatura hasta la fecha este problema ha sido abordado de diferentes maneras. En primer lugar, del propio algoritmo propuesto por Spong (1992), se extrae una expresión para la elección de ρ al tratar de obtener una ley de control que establezca el sistema en lazo cerrado. En concreto, para que el sistema en lazo cerrado resulte asintóticamente estable se debe escoger:

$$\rho_k \geq \lambda_0 + \lambda_1 \|\xi_k\| \quad (10)$$

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 dependen de cotas superiores e inferiores de la matriz de inercia y los términos no lineales de (1), evaluados sobre la trayectoria de referencia, así como de las aceleraciones máximas de dicha trayectoria.

Tomando la igualdad de la ecuación (10), 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*) (Corless and Leitmann, 1981). 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 (1).

En el controlador de Liu and Goldenberg (1993) los autores proponen una parametrización de las matrices de inercia y de Coriolis y centrífuga, que conduce a una acción robusta bastante similar a (8), probando que el error de seguimiento es de nuevo u.u.b.. El parámetro de diseño robusto para cada ley de control (denominado como acotación de la incertidumbre), 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 (1996) 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. (1987) se propone una ley robusta similar a (8), donde ρ queda fijada por la medida del error de seguimiento después de la elección aleatoria de otras constantes. Y revisando el resto de bibliografía, la elección del parámetro de

diseño robusto no queda clara o no se establece fácilmente, o simplemente no se menciona.

Respecto a estas propuestas existentes, los objetivos particulares que se pretenden conseguir con las técnicas de control presentadas en este trabajo 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,
- Incorporar en la ley de control las restricciones presentes en el sistema.

3. CONTROL ROBUSTO ADAPTATIVO

Para dar una respuesta al problema de elección adecuada del parámetro de diseño robusto se propone, en este trabajo, una ley adaptativa para actualizar eficazmente el valor de este parámetro.

3.1 Algoritmo de adaptación del controlador robusto

Se propone la adaptación del parámetro de diseño del controlador robusto ρ mediante la siguiente ecuación:

$$\rho_k = \rho_{k-1} - \gamma \frac{\partial J_k}{\partial \rho_{k-1}} \quad (11)$$

donde $J(\rho)$ es una función de costo y 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 de diseño ρ . Se propone una función de costo 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:

$$J_k(\rho_{k-1}) = \frac{1}{2} \xi_k^T Q_{ad} \xi_k + \frac{1}{2} y_{k-1}^T R_{ad} y_{k-1} \quad (12)$$

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

$$\rho_k = \rho_{k-1} - \gamma \left[\xi_k^T Q_{ad} \frac{\partial \xi_k}{\partial \rho_{k-1}} + y_{k-1}^T R_{ad} \frac{\partial y_{k-1}}{\partial \rho_{k-1}} \right] \quad (13)$$

Para calcular las derivadas de (13), se aplica una aproximación de primer orden a los términos con derivadas. El error ξ_k puede ser aproximado por:

$$\xi_k = \begin{bmatrix} \tilde{\theta}_k \\ \tilde{\theta}_k \end{bmatrix} = \begin{bmatrix} \theta_{d,k} \\ \dot{\theta}_{d,k} \end{bmatrix} - \begin{bmatrix} \theta_k \\ \frac{\theta_k - \theta_{k-1}}{h} \end{bmatrix} \quad (14)$$

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

$$\frac{\partial \xi_k}{\partial \rho_{k-1}} = - \begin{bmatrix} \frac{\partial \theta_k}{\partial \rho_{k-1}} \\ \frac{1}{h} \frac{\partial \theta_k}{\partial \rho_{k-1}} - \frac{1}{h} \frac{\partial \theta_{k-1}}{\partial \rho_{k-1}} \end{bmatrix} = - \begin{bmatrix} \frac{\partial \theta_k}{\partial \rho_{k-1}} \\ \frac{1}{h} \frac{\partial \theta_k}{\partial \rho_{k-1}} \end{bmatrix} \quad (15)$$

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

$$\begin{aligned} x_k &= A_D x_{k-1} + B_D y_{k-1} \\ \theta_k &= C x_k \end{aligned} \quad (16)$$

donde x_k es el estado del robot en el instante k , A_D es la matriz principal del sistema, B_D la matriz de entrada (estas dos últimas en su versión discreta), y C la matriz de salida.

Entonces, la derivada de (15) es:

$$\frac{\partial \theta_k}{\partial \rho_{k-1}} = C B_D \frac{\partial y_{k-1}}{\partial \rho_{k-1}} \quad (17)$$

Si se asume que $\|D^T Q \xi_k\| < \varepsilon$, $\forall \varepsilon$, lo cual es cierto excepto quizás al principio del movimiento, la ley de control (9)-(8) se convierte en:

$$y_k = \ddot{\theta}_{k,d} + K \xi_k + \rho_k M \xi_k \quad (18)$$

donde $M = D^T Q / \varepsilon$. La expresión (18) conduce al cálculo de la derivada en la parte derecha de (17) como sigue:

$$\frac{\partial y_{k-1}}{\partial \rho_{k-1}} = M \xi_{k-1} \quad (19)$$

Haciendo uso de (19) y (17), la expresión (15) puede ser escrita como:

$$\frac{\partial \xi_k}{\partial \rho_{k-1}} = - \begin{bmatrix} C B_D M \xi_{k-1} \\ \frac{1}{h} C B_D M \xi_{k-1} \end{bmatrix} \quad (20)$$

Finalmente, las expresiones (19) y (20) pueden ser usadas para evaluar la ley de adaptación (13). El esquema de control propuesto en este caso se puede observar en la figura 2.

Una vez definido este último paso, se han presentado todas las partes del controlador propuesto en este trabajo. La estabilidad de este controlador queda probada bajo los supuestos del teorema enunciado en (Torres, *et al.*, 2007), donde se debe dar la siguiente relación entre las matrices de la función de costo:

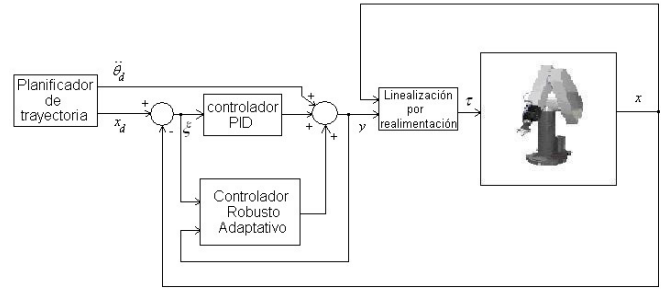


Figura 2: Esquema de control ARC

$$\frac{\|R_{ad}\|}{\|Q_{ad}\|} < \frac{\varepsilon}{y_M \|D^T Q\|} \|T\| \quad (21)$$

siendo y_M la entrada máxima al sistema linealizado y $T = \begin{bmatrix} C B_D & 1/h C B_D \end{bmatrix}^T$, representando $\|\cdot\|$ la norma-2 de una matriz.

Uno de los objetivos de este trabajo es la implementación del controlador robusto adaptativo ARC en un robot manipulador, comparando los resultados con los obtenidos con las estrategias clásicas de control robusto. De entre ellas, por ser la más representativa, escogeremos la estrategia de Spong (1992). En las siguientes secciones se describe el manipulador utilizado, así como los parámetros utilizados para el controlador y los resultados obtenidos.

3.2 Descripción del robot manipulador

El robot manipulador utilizado en este trabajo es un PUMA de la serie 560 de Unimation® (Figura 3). 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, 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.

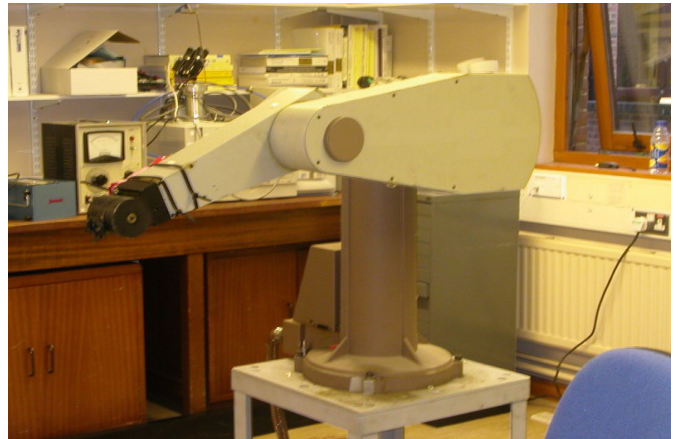


Figura 3: Robot manipulador PUMA-560 de Unimation empleado en los experimentos

Originalmente, este robot viene provisto de un controlador, el Unimation Mark II (Figura 4), 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.



Figura 4: Controlador Mark II del PUMA-560. En el módulo de control se disponen las nuevas tarjetas de interface con el PC a través de la tarjeta de adquisición Q8.

Para llevar a cabo el control del robot PUMA usando un PC con la configuración anterior, se modifica el controlador original (Becerra et al., 2004) 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 en 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 cuales se extraen las posiciones de las articulaciones.

Una de las principales dificultades encontradas en la implementación real del algoritmo de control ARC propuesto en este trabajo, y de otros algoritmos de control, es asegurar que los tiempos de adquisición de las medidas y los tiempos de cálculo del algoritmo y aplicación del resultado al sistema de actuación del manipulador no sobrepasan el tiempo de muestreo especificado para el algoritmo. Al tratarse de un algoritmo basado en modelo, se deben ejecutar cálculos de cierta complejidad que deben ser testeados previamente para asegurar su correcta implementación en tiempo real.

El modelo completo para el tipo particular de manipulador empleado en este trabajo, con las masas, longitudes, momentos de inercia, etc. se obtiene en Lewis et al. (2004).

3.3 Resultados del controlador ARC.

Se definen a continuación los parámetros del controlador utilizados en la obtención de los resultados que se muestran posteriormente. Para el término PD de la ley de control, se utilizan las siguientes matrices: $K_p = \text{diag}([24 \ 48 \ 48 \ 8 \ 8 \ 8])$ y

$K_D = \text{diag}([4 \ 6 \ 6 \ 6 \ 6 \ 6])$, siendo $\text{diag}(\cdot)$ la función que define una matriz diagonal. Para definir la acción robusta, se utilizan los siguientes valores: $\varepsilon=1$ y $Q = \text{diag}([0 \ 0 \ 0 \ 0 \ 0 \ 0, \ 1 \ 1 \ 1 \ 1 \ 1 \ 1])$. Para la función de coste empleada en la ley adaptativa, se utilizan las siguientes matrices: $R_{ad} = h * \text{diag}([25 \ 25 \ 25 \ 25 \ 25 \ 25])$ y $Q_{ad} = \text{diag}([100 \ 375 \ 250 \ 250 \ 250 \ 250, \ 250 \ 600 \ 500 \ 600 \ 600 \ 600])$. La elección de las matrices de costo viene condicionada por el cumplimiento de la relación (21) dada por el teorema que asegura la estabilidad de el algoritmo ARC. La velocidad de aprendizaje viene dada por $\gamma = 1$. En las diferentes pruebas se emplea un periodo de muestreo $h=0.001$.

En este trabajo se presentan resultados obtenidos para el caso de presencia de incertidumbre en el modelo o errores de modelado. El modelo estimado supuesto para este sistema, introducido en la sección 3.2, 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 mediante el esquema de linealización por realimentación.

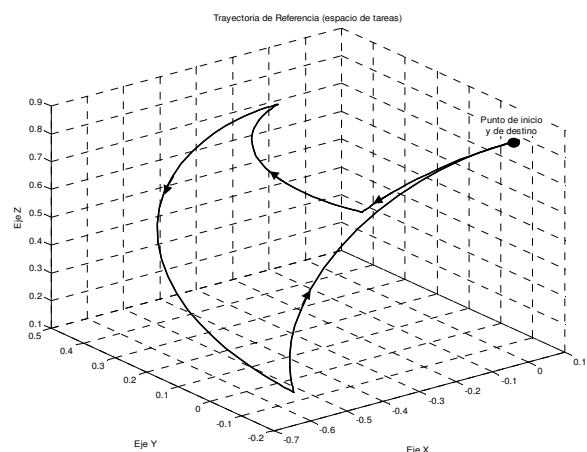


Figura 5: Trayectoria recorrida por el PUMA-560.

Los diferentes resultados han sido obtenidos escogiendo como trayectoria de referencia una 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 orden 7, los cuales aseguran un posicionamiento suave en cada uno de los puntos de paso y en el destino final. Dicha trayectoria (figura 5) 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).

En las figuras incluidas a continuación se representa la evolución del robot, para la trayectoria de referencia especificada, bajo la acción de un controlador PD, del controlador robusto bajo el

esquema original de Spong y del controlador robusto adaptativo propuesto en este trabajo. Para realizar los experimentos del controlador de Spong, se lleva a cabo un estudio en simulación previo para determinar el valor de las constantes de la ley adaptativa (10). Teniendo en cuenta los valores de los ángulos y velocidades de la trayectoria de referencia, se estiman, usando el modelo estimado, las cotas superior e inferior de la norma de la matriz de inercia, $B(\theta(t))$ de (1), así como la cota superior en el error en la norma del término no lineal $n(\theta(t), \dot{\theta}(t)) = h(\theta(t), \dot{\theta}(t)) + c(\theta(t))$. Para este segundo término, se calcula inicialmente la cota superior de la norma del término no lineal a lo largo de la trayectoria de referencia y se supone que el error 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 es $\lambda_1 = 2.6186$ y $\lambda_2 = 8.2092$.

Las figuras 6, 7, 8 y 9 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 $\tilde{\theta}$. Se puede comprobar que el peor comportamiento se obtiene para el controlador PD, 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 adaptativo robusto 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 de este método 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 10, 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. Esta evolución está referenciada como ARC_2 . 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, como puede observarse en la figura en la que los dos trazos confluyen a los mismos valores.

Este resultado prueba la bondad del algoritmo 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.

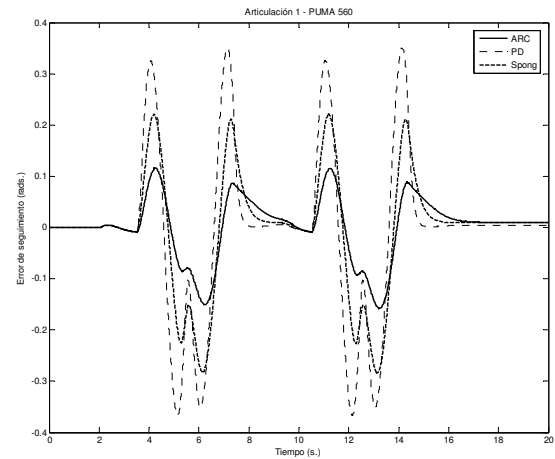


Figura 6: Error de seguimiento de la primera articulación bajo la acción de un controlador PD, el controlador de Spong y el controlador robusto adaptativo (ARC) propuesto.

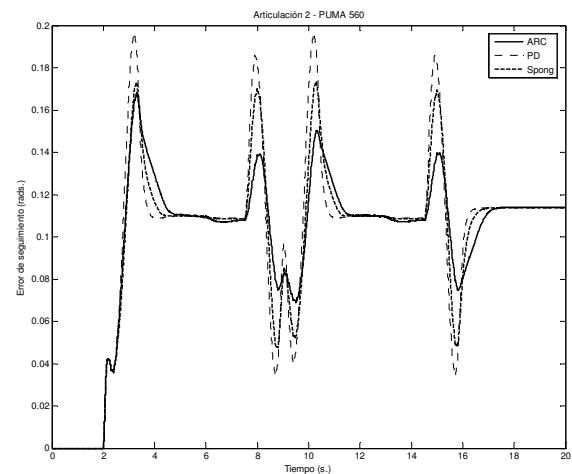


Figura 7: Error de seguimiento de la segunda articulación bajo la acción de un controlador PD, el controlador de Spong y el controlador robusto adaptativo (ARC) propuesto.

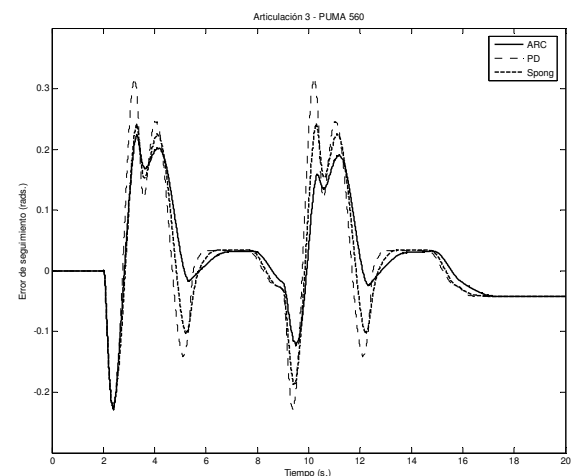


Figura 8: Error de seguimiento de la tercera articulación bajo la acción de un controlador PD, el controlador de Spong y el controlador robusto adaptativo (ARC) propuesto.

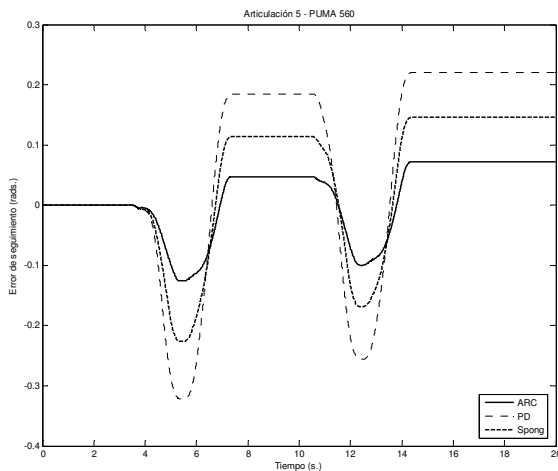


Figura 9: Error de seguimiento de la quinta articulación bajo la acción de un controlador PD, el controlador de Spong y el controlador robusto adaptativo (ARC) propuesto.

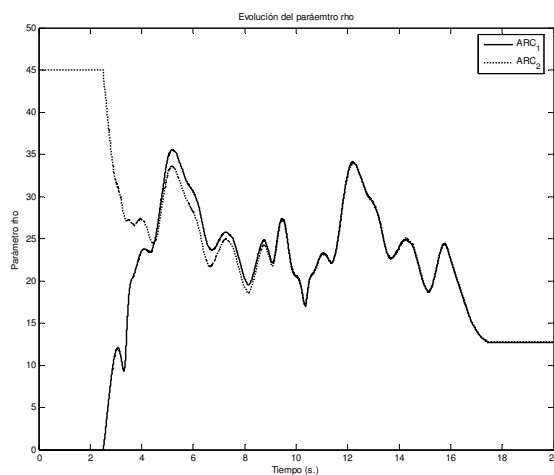


Figura 10: Evolución del parámetro de diseño robusto (ρ) para el controlador ARC en las pruebas mostradas en las figuras 3 a 6, indicada como ARC1, y partiendo de un valor inicial distinto, indicada como ARC2.

4. TÉCNICAS PREDICTIVAS PARA INCORPORAR LAS LIGADURAS DEL SISTEMA

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 (Poignet y Gautier, 2000; Woóblewski, 2004; Gu y Hu, 2004) 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 trabajo 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. Se trata del esquema de control RIAPC, que resulta de aplicar la ley autoadaptativa del parámetro de diseño robusto a una ley de control estabilizante con acción robusta.

Es este caso, la ley estabilizante está basada en un algoritmo predictivo con interpolación. Además, para una formulación sencilla de las ligaduras del sistema, se emplea un algoritmo de linealización local del modelo en lugar del esquema de linealización por realimentación empleado en el controlador ARC.

4.1 El control predictivo por 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 (Kouvaritakis, 1998; Rossiter, 1998) 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 este trabajo 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 las entradas que se deben aplicar a la planta vienen dadas por la resolución de la dinámica inversa. Las correcciones a dichas entradas vienen dadas en un primer caso por el algoritmo de interpolación LM (interpolación entre soluciones LQ y ML). Este esquema de control permitirá 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.

4.1.1 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:

$$J = \sum_{i=1}^N \left(\|r(k+i) - y(k+i|k)\|_2 \right)^2 + \eta \sum_{i=0}^{M-1} \left(\|\Delta u(k+i)\|_2 \right)^2 \quad (22)$$

una vez obtenidas las predicciones para un sistema cuya evolución temporal viene dada por la ecuación expresada en el espacio de los estados, y cuya ley de control es de la forma:

$$u(k) = -Kx(k) \quad (23) \quad \begin{aligned} w(0) &= (1 - \alpha)x(0) \\ z(0) &= \alpha x(0) \end{aligned} \quad (26)$$

la solución viene dada por el valor de la ganancia K para la cual la secuencia de entradas $u(k)$ minimiza la función de costo. Esta solución se designará como K_{LQ} .

4.1.2 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 con la ley de control (23), donde la función de costo (22) 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.

4.1.3 Interpolación LQ+ML.

La solución anterior, en sí misma, no es la solución deseable para el sistema. En (Kouvaritakis *et al*, 1998; Rossiter *et al*, 1998; Méndez, *et al*, 2000; Rossiter *et al*, 2004) 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 τ_{LQ} , y la solución factible ML, dada por τ_{ML} , el problema se reduce a una interpolación entre ambas de la forma:

$$\tau_{LM} = (1 - \alpha)\tau_{LQ} + \alpha\tau_{ML} \quad (24)$$

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:

$$x(0) = w(0) + z(0) \quad (25)$$

en donde:

Las predicciones se harán utilizando la ecuación:

$$\tau_{LM}(k) = -K_{LQ}\phi_{LQ}^k w(0) - K_{ML}\phi_{ML}^k z(0) \quad (27)$$

donde K_{LQ} es la ganancia obtenida del problema de control óptimo LQ sin ligaduras y K_{ML} es la ganancia correspondiente al problema de nivel medio ML. Además:

$$\begin{aligned} \phi_{LQ} &= A_D - B_D K_{LQ} \\ \phi_{ML} &= A_D - B_D K_{ML} \end{aligned} \quad (28)$$

siendo A_D y B_D las matrices de (16).

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 τ_{LM} sea factible (para lo cual se utilizarán las predicciones hasta el horizonte de predicción N) y, a su vez, minimice la función de costo (22). 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.

4.2 Controlador RIAPC

En la figura 5 se observa un esquema del controlador propuesto. Este controlador resulta de aplicar un esquema similar al controlador ARC. La ley de control estabilizante viene dada en este caso por la aplicación de un controlador predictivo por interpolación LM. Para una formulación sencilla de las ligaduras del sistema, el esquema de linealización elegido en este caso es una linealización local en torno a la trayectoria de consigna (Torres, *et al.*, 2000).

Figura 5: Esquema del controlador RIAPC.

Un controlador por dinámica inversa proveerá los pares $\tau_{k,o}$ necesarios para seguir idealmente la trayectoria de referencia. Las desviaciones de la misma son corregidas por la ley predictiva LM y la acción robusta autoadaptativa $\tau_{R,k}$ descrita anteriormente. Por lo cual, la ley de control viene dada por:

$$\tau_k = \tau_{k,o} + (1 - \alpha_{k,o}) K_{LQ} \tilde{\mathbf{x}}_k + \alpha_o K_{ML} \tilde{\mathbf{x}}_k + \tau_{R,k} \quad (29)$$

En el caso del controlador LQ, la función de costo a optimizar es:

$$J = \sum_{k=0}^{\infty} (\tilde{\mathbf{x}}^T V \tilde{\mathbf{x}} + \mathbf{u}^T W \mathbf{u}) \quad (30)$$

La solución a este problema es la ganancia K_{LQ} . Para obtener la solución ML se minimiza la expresión (30) pero considerando $W \gg V$. Para el cálculo de las predicciones en la entrada y en el estado se tienen las expresiones:

$$\tilde{\tau}_j = (1 - \alpha) K_{LQ} \phi_{LQ}^j \tilde{\mathbf{x}} + \alpha K_{ML} \phi_{ML}^j \tilde{\mathbf{x}} \quad (31)$$

$$\tilde{\mathbf{x}}_j = (1 - \alpha) \phi_{LQ}^j \tilde{\mathbf{x}} + \alpha \phi_{ML}^j \tilde{\mathbf{x}} \quad (32)$$

Con $j=0, \dots, n_c$ (siendo n_c el horizonte de control). Si se definen $\tilde{\mathbf{x}}_{\max}^+ = \max\{\mathbf{x} - \mathbf{x}_o\}$ como el máximo sobrepasamiento positivo permitido del estado respecto a la trayectoria de referencia \mathbf{x}_o y $\tilde{\mathbf{x}}_{\max}^- = \max\{\mathbf{x}_o - \mathbf{x}\}$, el máximo sobrepasamiento negativo, y τ_M el valor máximo de entrada, el problema de programación lineal a resolver en el algoritmo de interpolación es:

$$\begin{aligned} & \min_{\alpha} \alpha \\ & \text{sujeto a:} \\ & (-\phi_{LQ}^j \tilde{\mathbf{x}} + \phi_{ML}^j \tilde{\mathbf{x}}) \alpha \leq \tilde{\mathbf{x}}_{\max}^+ - \phi_{LQ}^j \tilde{\mathbf{x}} \\ & (-\phi_{LQ}^j \tilde{\mathbf{x}} + \phi_{ML}^j \tilde{\mathbf{x}}) \alpha \leq -\tilde{\mathbf{x}}_{\max}^- + \phi_{LQ}^j \tilde{\mathbf{x}} \\ & (-K_{LQ} \phi_{LQ}^j \tilde{\mathbf{x}} - K_{ML} \phi_{ML}^j \tilde{\mathbf{x}}) \alpha \leq \tau_M - \tau_o - K_{LQ} \phi_{LQ}^j \tilde{\mathbf{x}} \\ & (-K_{LQ} \phi_{LQ}^j \tilde{\mathbf{x}} - K_{ML} \phi_{ML}^j \tilde{\mathbf{x}}) \alpha \leq -\tau_M + \tau_o + K_{LQ} \phi_{LQ}^j \tilde{\mathbf{x}} \\ & 0 \leq \alpha \leq 1 \end{aligned} \quad (33)$$

5. RESULTADOS CON EL CONTROLADOR RIAPC

Se exponen a continuación los resultados obtenidos con los controladores LQ robusto, el controlador propuesto RIAPC y con el mismo controlador pero sin utilizar la acción robusta autoadaptativa, al que denominaremos IAPC.

En los tres casos se utiliza un modelo con incertidumbres paramétricas, consistentes en una desviación de los valores de las masas y longitudes respecto al robot real. Para comparar las estrategias de control propuestas, se realiza un experimento en el que se introduce un ruido de medida al sistema de manera que se produce la violación de las ligaduras.

Para la acción del controlador óptimo LQ se escogen las matrices $V = \text{diag}([4.8 \ 1.2 \ 4.8 \ 1.2 \ 3.2 \ 0.8 \ 3.2 \ 0.8 \ 3.2 \ 0.8])$ y $W = \text{diag}(0.001 * [25 \ 25 \ 25 \ 5 \ 5 \ 5])$. Para la acción robusta se escoge $\varepsilon = 1$ y $Q = \text{diag}([0 \ 0 \ 0 \ 0 \ 0 \ 0, \ 1 \ 1 \ 1 \ 1 \ 1])$. Las matrices de la función de costo de la ley adaptativa del son $R_{ad} = h * \text{diag}([25 \ 25 \ 25 \ 25 \ 25 \ 25])$ y $Q_{ad} = 100 * \text{diag}([1 \ 3.75 \ 3.75 \ 3.75 \ 2.5 \ 2.5, \ 2.5 \ 6 \ 6 \ 6 \ 6])$. Se escoge $\gamma = 1$.

Para la ley estabilizante predictiva con interpolación se tienen las matrices $V' = 0.1 \cdot V$ y $W' = 1e4 \cdot W$ para las matrices de la función de costo (30) el caso de la solución ML. Las restricciones impuestas al manipulador son:

$$\begin{aligned} \tau_M &= [75 \ 36.5 \ 11 \ 10 \ 10 \ 10] \\ \tilde{\mathbf{x}}_{\max}^+ &= [15 \ 15 \ 15 \ 5 \ 5 \ 5] * \pi / 180 \\ \tilde{\mathbf{x}}_{\max}^- &= -\tilde{\mathbf{x}}_{\max}^+ \end{aligned} \quad (34)$$

La trayectoria de referencia es similar a la empleada en los resultados mostrados en la sección 3.2, donde se han elegido los puntos de paso próximos a las ligaduras para probar la eficiencia del controlador propuesto. En las figuras 11, 12 y 13 se muestran, respectivamente, los resultados obtenidos para las tres primeras articulaciones del manipulador. La respuesta del manipulador antes del instante $t = 5$ s. es bastante similar para los tres controladores. A partir de ese momento, en el cual ocurre la perturbación, el comportamiento difiere bastante. Se puede observar como de nuevo en la articulación 2 se tiene una saturación en la entrada con los controladores LQ e IAPC.

En este caso, la estrategia IAPC no mejora los resultados obtenidos con el controlador LQ en cuanto que no evita la violación de las ligaduras y se tienen errores de seguimiento mayores. Este comportamiento también provoca saturación de las entradas en la articulación 3 y errores de seguimiento sensiblemente superiores.

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 12 y 13, 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.

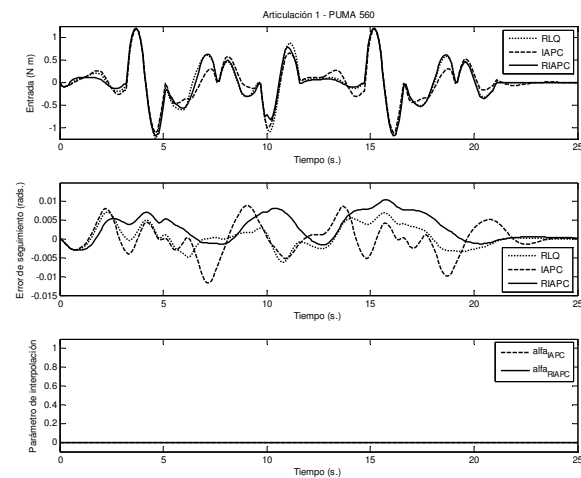


Figura 11: Evolución de la entrada, error de seguimiento y parámetro de interpolación para la articulación 1.

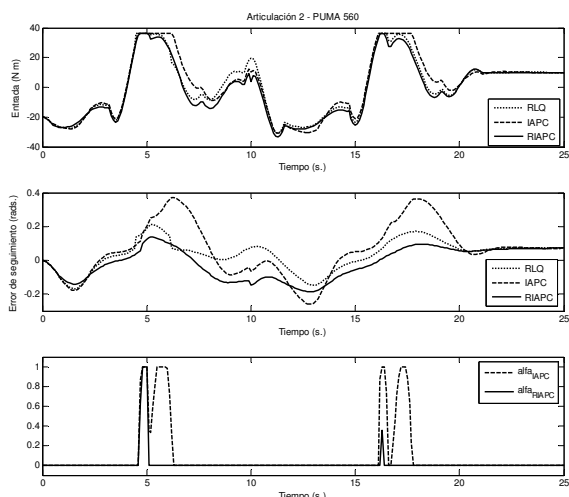


Figura 12: Evolución de la entrada, error de seguimiento y parámetro de interpolación para la articulación 2.

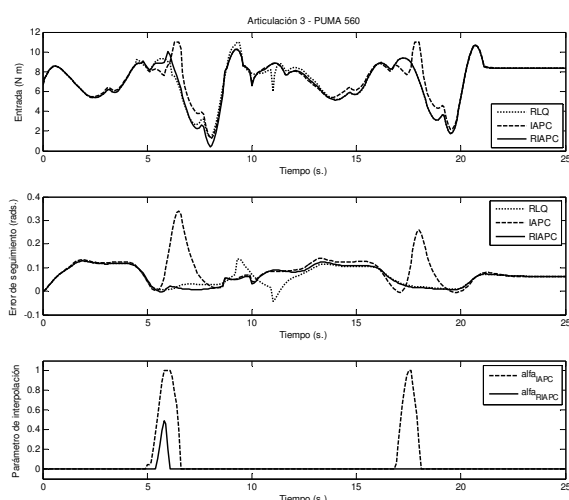


Figura 13: Evolución de la entrada, error de seguimiento y parámetro de interpolación para la articulación 3.

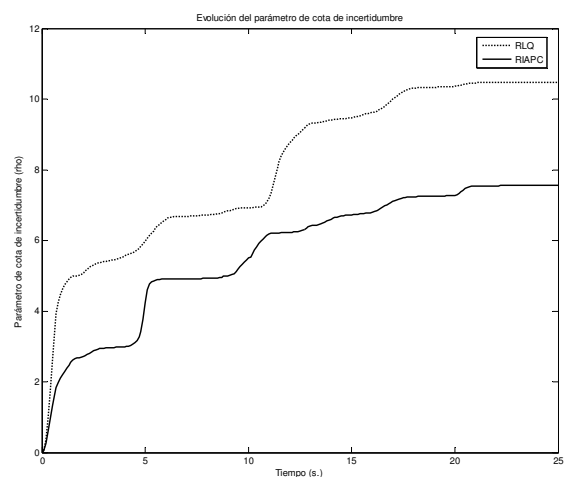


Figura 14: Evolución del parámetro de cota de incertidumbre.

Sin embargo, en la articulación 1 (figura 11) 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 14 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.

6. CONCLUSIONES

En este trabajo se proponen nuevas estrategias de control robusto para el caso de robots manipuladores que mejoran la eficacia de los algoritmos ideados hasta el momento. Los diseños de control robusto denominados ARC y RIAPC, rechazan convenientemente las perturbaciones a las que está sometido el sistema independientemente de la magnitud de las mismas, sin necesidad de realizar consideraciones previas sobre las incertidumbres y, específicamente para el segundo controlador, teniendo en cuenta las ligaduras del sistema.

Según se prueba en este trabajo, la elección del parámetro de diseño robusto de la ley de control robusto es fundamental a la hora de obtener resultados satisfactorios en el seguimiento de trayectorias. Para ello se ha ideado una estrategia adaptativa que actualice convenientemente el valor de este parámetro, resultando en el controlador ARC. La elección de la función de costo a minimizar es totalmente general y es válida para otras estructuras de manipuladores. Esta estrategia ha sido probada con eficiencia en un robot PUMA-560 de Unimation. Los resultados mostrados prueban como este nuevo controlador robusto adaptativo mejora la eficacia de un controlador PD y de un controlador robusto clásico, dado que el error de seguimiento en cada una de las articulaciones se reduce, en el mejor de los casos, hasta la mitad.

En cuanto al segundo controlador, RIAPC, se puede concluir que aporta 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, además, 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, aparte de que la adaptación del parámetro de cota de incertidumbre en el caso de la acción robusta no supone un incremento apreciable del costo computacional.

Dentro de las posibles ampliaciones a este trabajo, cabe mencionar el uso de redes neuronales para la obtención de un modelo de predicción para el manipulador, de forma que el tiempo involucrado en la ejecución del algoritmo RIAPC disminuya drásticamente. Este modelo neuronal se obtiene a partir de los datos de entrada-salida del manipulador en pruebas en lazo cerrado con trayectorias de referencia similares a las utilizadas con los controladores propuestos. Siguiendo esta misma idea, una de las líneas abiertas que surgen de este trabajo es la obtención de dicho modelo neuronal de predicción en línea, una vez realizada una inicialización del mismo de manera *off-line*. Otra ampliación del trabajo se centra en la aplicación de

estas estrategias a robots flexibles, en los cuales la presencia de incertidumbre y el rechazo a perturbaciones son aspectos de gran relevancia para el control efectivo de este tipo de manipuladores. Una última propuesta de ampliación de este trabajo es la utilización de estos algoritmos en técnicas de control híbrido, empleadas habitualmente en el control de fuerza, para que el rechazo efectivo de perturbaciones e incertidumbres en el brazo redunde en una tarea de manipulación más sencilla por parte del efector final.

REFERENCIAS

- F. Alonge, F. D'Ippolito and F.M. Raimondi (2003). An adaptive control law for robotic manipulator without velocity feedback. *Control Engineering Practice* **11** (9), 999-1005.
- A. Barrientos, L. Peñín, C. Balaguer and R. Aracil (2007). *Fundamentos de Robótica, 2ª ed.* McGraw Hill. España.
- V.M. Becerra, C.N.J. Cage, W.S. Harwin and P.M. Sharkey (2004). Hardware retrofit and computed torque control of a Puma 560 Robot. *IEEE Control Systems Mag.* **24** (5), 78-82.
- M. Corless and G. Leitmann (1981). Continuous-state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems. *IEEE Trans. on Automatic Control* **AC-26**, 1139-1141.
- D.M. Dawson, Z. Qu and J.J. Carroll (1992). Tracking control of rigid-link electrically-driven robot manipulators. *Int. Journal of Control* **56**, 991-1006.
- S. Desa and B. Roth (1985). Synthesis of control systems for manipulators using multivariable robust servo mechanism theory. *Int. J. Robotic Res.* **4**, 18-34.
- E. Freund (1982). Fast nonlinear control with arbitrary pole-placement for industrial robots and manipulators. *Int. Journal of Robotics Research* **1** (1), 65-78.
- L.C. Fu, T.L. Liao (1990). Globally Stable Robust Tracking of Nonlinear Systems Using Variable Structure Control and with an Application to a Robotic Manipulator. *IEEE Trans. on Automatic Control* **35**(12), 1345-1350.
- D. Gu, H. Hu (2004). Model Predictive Control for Simultaneous Robot Tracking and Regulation. In Proc.: *Int. Conf. on Intelligent Mechatronics and Automation*, Chengdu, China. 212-217.
- C. Ishii, T. Shen and Z. Qu (2001). Lyapunov recursive design of robust adaptive tracking control with L_2 -gain performance for electrically-driven robot manipulators. *Int. Journal of Control* **74** (8), 811-828.
- A. Jaritz and M.W. Spong (1996). An experimental comparison of robust control algorithms on a direct drive manipulator. *IEEE Trans. on Control Systems Technology* **4** (6), 627-640.
- Y. Jin (1998). Decentralized adaptive fuzzy control of robot manipulators. *IEEE Trans. on Systems, Man and Cybernetics – B: Cybernetics* **28** (1), 47-57.
- B. Kouvaritakis, J.A. Rossiter and M. Cannon (1998). Linear quadratic feasible predictive control. *Automatica* **34** (12), 1583-1592.
- K. Kreutz (1989). On manipulator control by exact linearization. *IEEE Trans. on Automatic Control* **34**, 763-767.
- F. Lewis, D. Dawson and C. Abdallah (2004). *Robot manipulator control*. Marcel Dekker. New York.
- G. Liu and A. Goldenberg (1993). On robust saturation control of robot manipulators. In Proc.: *32nd Conf. on Decision and Control*, Texas, USA. 2115-2120.
- N. Londoño and M. Arias (2003). Control de posición de un manipulador utilizando lógica difusa. In Proc.: *XI Congreso Latinoamericano de Control Automático*, Cuba.
- M.F. Mendes, W. Kraus Jr. and E. R. de Pieri (2002). Variable Structure Position Control of an industrial Robotic Manipulator. *Journal of the Brazilian Society of Mechanical Sciences*, **24** (3).
- J.A. Méndez, B. Kouvaritakis and J.A. Rossiter (2000). State-space approach to interpolation in MPC. *Int. Journal of Robust and Nonlinear Control* **10**, 27-38.
- R. Ortega and M.W. Spong (1989). Adaptive motion control of rigid robots: A tutorial. *Automatica* **25**, 877-888.
- P. Poignet and M. Gautier (2000). Nonlinear Model Predictive Control of a Robot Manipulator. In Proc.: *6th Int. Workshop on Advanced Motion Control*, Nagoya, Japan. 401-406.
- J. Reed and P. Ioannou (1988). Instability analysis and adaptive control of robot manipulators. In Proc.: *27th IEEE Conf. on Decision & Control*. Austin, Texas. 1607-1612.
- J.A. Rossiter, M.J. Rice, J. Schuurmans and B. Kouvaritakis (1998). A computationally efficient constrained predictive control law. In Proc.: *American Control Conference*, Philadelphia, USA.
- J.A. Rossiter, B. Kouvaritakis and M. Bacic (2004). Interpolation based computationally efficient predictive control. *Int. Journal of Control*, **77** (3), 290-301.
- L. Sciavicco and B. Siciliano (1996). *Modelling and Control of Robot Manipulator*. Springer-Verlag. London.
- J.J.E. Slotine (1985). Robust control of robot manipulators. *Int. J. of Robotics Research* **4**, pp. 49-64.
- J.J.E. Slotine and W. Li (1987). On the adaptive control of robot manipulators. *Int. Journal of Robotic Research* **6**, 49-59.
- M.W. Spong and M. Widyasagar (1987). Robust linear compensator for nonlinear robotic control. *IEEE Journal of Robotics and Automation* **RA-3**, 345-351.
- M.W. Spong, J.S. Thorp and J.M. Kleinwaks (1987). Robust microprocessor control of robot manipulators. *Automatica* **23**, 373-379.
- M.W. Spong (1992). On the robust control of robot manipulators. *IEEE Trans. on Automatic Control* **37**, 1782-1786.
- M.W. Spong (1993). Adaptive Control of Robot Manipulators: Design and Robustness. In Proc.: *American Control Conference*, San Francisco, California. 2826-2830.
- C.Y. Su and T.P. Leung (1993). A Sliding Mode Controller with Bound Stimulation for Robot Manipulators. *IEEE Trans. on Robotics and Automation* **9** (2), 208-214.
- C.Y. Su and Y. Stepanenko (1997). Redesign of hybrid adaptive/robust motion control of rigid-link electrically-driven robot manipulators. *IEEE Trans. on Robotics and Automation* **14**, 651-655.

- T.J. Tarn, A.K. Bejczy, A. Isidori and Y. Chen (1984). Nonlinear Feedback in Robot Arm Control. In Proc.: *23th IEEE Conf. on Decision and Control*, Las Vegas, USA.
- S. Torres, J.A. Méndez, L. Acosta, M. Sigut, G.N. Marichal and L. Moreno (2000). A decentralized fuzzy controller for a three link robot manipulator. In Proc.: *ICSC 2nd Symposia on Eng. of Intelligent Systems*. Paisley, Scotland, UK.
- S. Torres, J.A. Méndez, L. Acosta, M. Sigut and G.N. Marichal (2002). Disturbances rejection on a robot arm using an efficient predictive controller. In Proc.: *15th IFAC World Congress*, Barcelona, Spain.
- S. Torres, J.A. Méndez, L. Acosta and V.M. Becerra (2007). On improving the performance in robust controllers for robot manipulators with parametric disturbances. *Control Engineering Practice* **15** (5), 557-566.
- W. Wróblewski (2004). Implementation of a Model Predictive Control Algorithm for a 6dof Manipulator-Simulation Results. *4th IEEE Int. Workshop on Robot Motion and Control*, Puzczykowo, Poland. 209-212.