

01126
481



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

TESIS CON FALLA DE ORIGEN

IMPLEMENTACION DE DIVERSOS ALGORITMOS DE CONTROL EN UN ROBOT INDUSTRIAL

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO MECANICO ELECTRICISTA
AREA: ELECTRICO ELECTRONICO

P R E S E N T A :
EDUARDO PEREZ CAMACHO

DIRECTOR DE TESIS:
DR. MARCO ANTONIO ARTEAGA PEREZ



MEXICO, D. F.

2003



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

TESIS CON
FALLA DE ORIGEN

A mi abuela, mi madre y mi hermano.
A todas las personas que lean esta obra.

Agradecimientos

TESIS CON
FALLA DE ORIGEN

Quisiera agradecer a muchas personas que me han ayudado a llegar hasta aquí, pero no las recuerdo a todas en este momento y para evitar problemas, prefiero no nombrarlas.

Quiero agradecer:

- A Jah por permitirme llegar hasta aquí.
- A mi familia.
- A todos mis amigos y profesores por apoyarme y confiar en mi.
- Aunque yo sé que nunca van a leer esta obra, agradezco grandemente a todas las personas que nunca creen en alguien por su apariencia, ya que este fue un gran motor para llegar hasta aquí.

También le doy las gracias a mi asesor por apoyarme.

TESIS CON
FALLA DE ORIGEN

Índice General

1	Introducción	3
2	Preliminares Matemáticos	7
2.1	Introducción	7
2.2	Ecuaciones de movimiento de Euler-Lagrange	9
2.3	Modelo dinámico de un robot	12
2.3.1	Energía cinética del manipulador	12
2.3.2	Energía potencial de un robot	16
2.3.3	Ecuaciones de movimiento de un robot	17
2.4	Propiedades del modelo dinámico	19
2.4.1	Propiedades de las matrices	19
2.4.2	Cotas para normas de matrices	20
2.4.3	Propiedades relacionadas con todo el modelo	21
2.5	Inclusión de la dinámica de los motores	22
3	Descripción de los algoritmos de control	25
3.1	Control Proporcional-Derivativo robusto	26
3.2	Control Proporcional-Derivativo con estimación de velocidad	27
3.3	Controlador Adaptable de Compensación Deseada (DCAL)	29
3.4	Control usando un observador lineal	31
3.5	Control y estimación de parámetros con sólo lectura de posición de articulación	32
4	Implementación de los algoritmos	35
4.1	Método de sintonización de ganancias	36
4.2	Equivalencia de ganancias	39
4.2.1	Ecuaciones de los controladores	39

4.2.2	Cálculo de ganancias	41
4.3	Experimentos	45
4.3.1	Experimentos sin utilizar modelo en la ley de control	46
4.3.2	Experimentos utilizando modelo en la ley de control	60
5	Conclusiones	71
A	Descripción de la interfaz PC-Robot	75
A.1	Partes que componen del equipo	76
A.2	Interfaz	78
B	Programas	81
	Bibliografía	111

TESIS CON
FALLA DE ORIGEN



Índice de Figuras

2.1	Ejes coordenados de acuerdo con las suposiciones de Denavit-Hartenberg.	10
2.2	Un punto r_i en el elemento i	12
2.3	Armadura de motor	22
2.4	Motor y Engranés	23
3.1	Sistema de control PD.	26
3.2	Sistema de control-observador.	28
3.3	Diagrama de bloques del controlador.	30
4.1	Lazo cerrado sólo con ganancia proporcional.	36
4.2	Respuesta de la planta a lazo cerrado con ganancia crítica.	37
4.3	Respuesta de las articulaciones en lazo cerrado con ganancia crítica.	38
4.4	Errores de seguimiento Trayectoria 1, Articulación 1 sin modelo.	48
4.5	Errores de seguimiento Trayectoria 1, Articulación 2 sin modelo.	49
4.6	Errores de seguimiento Trayectoria 1, Articulación 3 sin modelo.	50
4.7	Errores de seguimiento Trayectoria 1, Articulación 4 sin modelo.	51
4.8	Errores de seguimiento Trayectoria 1, Articulación 5 sin modelo.	52
4.9	Errores de seguimiento Trayectoria 1, Articulación 6 sin modelo.	53
4.10	Errores de seguimiento Trayectoria 2, Articulación 1 sin modelo.	54
4.11	Errores de seguimiento Trayectoria 2, Articulación 2 sin modelo.	55
4.12	Errores de seguimiento Trayectoria 2, Articulación 3 sin modelo.	56
4.13	Errores de seguimiento Trayectoria 2, Articulación 4 sin modelo.	57
4.14	Errores de seguimiento Trayectoria 2, Articulación 5 sin modelo.	58
4.15	Errores de seguimiento Trayectoria 2, Articulación 6 sin modelo.	59
4.16	Errores de seguimiento Trayectoria 1, Articulación 1 con modelo.	62
4.17	Errores de seguimiento Trayectoria 1, Articulación 2 con modelo.	62
4.18	Errores de seguimiento Trayectoria 1, Articulación 3 con modelo.	63

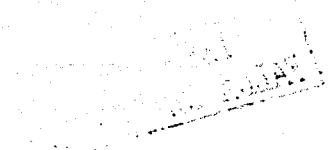
4.19	Errores de seguimiento Trayectoria 1, Articulación 4 con modelo.	63
4.20	Errores de seguimiento Trayectoria 1, Articulación 5 con modelo.	64
4.21	Errores de seguimiento Trayectoria 1, Articulación 6 con modelo.	64
4.22	Errores de seguimiento Trayectoria 2, Articulación 1 con modelo.	65
4.23	Errores de seguimiento Trayectoria 2, Articulación 2 con modelo.	66
4.24	Errores de seguimiento Trayectoria 2, Articulación 3 con modelo.	67
4.25	Errores de seguimiento Trayectoria 2, Articulación 4 con modelo.	68
4.26	Errores de seguimiento Trayectoria 2, Articulación 5 con modelo.	69
4.27	Errores de seguimiento Trayectoria 2, Articulación 6 con modelo.	70
A.1	Diseño de interfaz para control de Robot A465.	79

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN

Índice de Tablas

4.1	Ajuste de parámetros.	37
4.2	Ganancias K_{pi} y K_{di} ajustadas por método de oscilación	39
4.3	Tabla comparativa de índices de desempeño para la Trayectoria 1 sin modelo.	47
4.4	Tabla comparativa de índices de desempeño para la Trayectoria 2 sin modelo.	47
4.5	Tabla comparativa de índices de desempeño para la Trayectoria 1 sin modelo.	61
4.6	Tabla comparativa de índices de desempeño para la Trayectoria 2 sin modelo.	61



Capítulo 1

Introducción

TESIS CON
FALLA DE ORIGEN

Aunque la apariencia de los robots industriales no ha cambiado significativamente y muchos modelos actuales tienen una estructura y funcionamiento similares al primer manipulador PUMA que Unimation entregó en 1978 al centro de tecnología de General Motors en Warren, Michigan, lo cierto es que tanto en los aspectos mecánicos como en el control y la programación, los avances han sido importantes. Los robots actuales son más robustos, rápidos y fiables. Su capacidad de carga y repetitividad es comparativamente superior, y su programación se ha facilitado considerablemente.

A pesar de la gran cantidad de trabajo en algoritmos de control para robots manipuladores, algunos trabajos ilustran los beneficios y potencial desarrollo por simulaciones y solamente unos pocos han sido presentados con resultados experimentales en manipuladores actuales (Basañez B. 1999). La principal razón de esto son la falta de adecuados sistemas experimentales y la dificultad de obtener un modelo dinámico correcto del robot (Reyes y Kelly 2001).

Las simulaciones son flexibles y pueden guiar durante las etapas iniciales del diseño de los algoritmos de control, pero tales son incompletas por que abandonan aspectos prácticos, tales como dinámica de los actuadores, ruido en los sensores, etc. La validación experimental de algoritmos de control asegura su éxito potencial en aplicaciones en el mundo real. En este sentido el desarrollo de robot experimentales de arquitectura abierta es un paso esencial hacia la validación de nuevos y existentes algoritmos de control para

robot industriales (Mills *et al.* 1995).

La primer comparación de algoritmos de control fue presentada en 1983 por Asada *et al.* (Reyes y Kelly 2001), al implementar el controlador proporcional-derivativo (PD) y su combinación con un compensador en adelanto en el CMU Direct-Drive Arm I, el cual es un brazo mecánico en el cual los ejes de las articulaciones están acoplados a los rotores de los motores con alto torque. Al cual siguieron varios experimentos del mismo tipo desarrollados por An *et al.* en 1988 y 1989 presentaron la implementación de un simple control PD y un PD con compensador hacia delante en el MIT Serial Link Direct Drive Arm, Khosla y Kanade en 1988 implemento y comparo un control de torque computado con un PD mas feedforward en el CMU Direct-Drive Arm II, por mencionar algunos, en los cuales se implementaron controladores robustos y adaptables.

Posterior a esto se realizaron experimentos en brazos robot con engranes entre los que sobresalen Leahy en 1990, De Jager en 1994, Caccavale y Chiachchio en 1994, estrategias adaptables son implementadas por Berghuis *et al.* en 1995 en un robot rotatorio y finalmente en 1993 Tern *et al.* presenta resultados experimentales de la implementación en un brazo robot PUMA 560.

El presente trabajo presenta la implementación de cuatro algoritmos de control experimentalmente y el simple control PD, en un brazo robot moderno y se realiza una comparación de desempeño de estos, considerando los casos sin modelo y con modelo dinámico. Los resultados obtenidos muestran que el desempeño de los algoritmos considerando el modelo dinámico del brazo robot presenta un mejor desempeño con respecto al caso sin modelo y que en este trabajo las nuevas tecnicas de control mejoran el desempeño del control PD.

La tesis se encuentra ordenada de la siguiente forma. El Capitulo 2 contiene los preliminares matemáticos donde se expone el modelado dinámico de los robots manipuladores que se obtiene utilizando la formulación de Lagrange y algunas de sus propiedades. En el Capitulo 3 se describen los algoritmos de control que se implementaron en el robot industrial. En el Capitulo 4 se desarrolla la implementación de los algoritmos de control mencionados en el capitulo previo, junto con un método de sintonización de ganancias, los distintos experimentos y resultados obtenidos. Finalmente, en el Capitulo 5

se escriben las conclusiones a las que se llegó con los resultados experimentales.

TESIS CON
FALLA DE ORIGEN

Capítulo 2

Preliminares Matemáticos

2.1 Introducción

La dinámica se ocupa de la relación entre las fuerzas que actúan sobre un cuerpo y el movimiento que en él se originan. Por lo tanto, el modelo dinámico de un robot tiene por objetivo conocer mejor la relación entre el movimiento del robot y las fuerzas aplicadas en el mismo. Esta relación se obtiene mediante el denominado modelo dinámico, que relaciona matemáticamente:

- La localización del robot definida por sus variables articulares o por las coordenadas de localización de su extremo, y sus derivadas: velocidad y aceleración.
- Las fuerzas y pares aplicados en las articulaciones (o en los extremos del robot).
- Los parámetros dimensionales del robot, como longitud, masas y momentos de inercia de sus elementos.

La obtención de un modelo correcto para manipuladores es difícil, por lo tanto, este es uno de los aspectos más complejos de la robótica, lo que ha llevado a ser dar por conocido en numerosas ocasiones el modelo. Sin embargo, el modelo dinámico es imprescindible para conseguir los siguientes fines (Barrientos *et al.* 1997):

- Simulación de los movimientos del robot.
- Diseño y evaluación de la estructura mecánica del robot.
- Dimensionamiento (tamaño) de los actuadores.
- Diseño y evaluación del control dinámico del robot.

Este último punto es de gran importancia, puesto que de la calidad del control dinámico del robot depende la precisión y velocidad de sus movimientos. La gran complejidad existente en la obtención del modelo dinámico del robot ha motivado a que se realicen ciertas simplificaciones en el mismo, de manera que pueda ser utilizado en el diseño del controlador.

El modelo dinámico se puede obtener a partir de leyes físicas conocidas, tales como las leyes de la mecánica newtoniana y lagrangiana. Esto conduce al desarrollo de las ecuaciones de movimiento dinámico para las diversas articulaciones del manipulador en términos de los parámetros geométricos e inerciales de los elementos. Las formulaciones de Euler-Lagrange y Newton-Euler (Spong y Vidyasagar 1989), son las más empleadas. De estas formulaciones se obtienen diferentes formas de describir la dinámica del brazo robot.

La obtención del modelo dinámico de un manipulador basado en la formulación de Euler-Lagrange es simple y sistemática. Las relaciones de movimiento resultantes proporcionan ecuaciones de estado explícitas para la dinámica del robot, y se pueden utilizar para analizar y diseñar estrategias de control avanzadas en el espacio de las variables de la articulación, por lo que es el método que se estudiara en el presente capítulo .

TESIS CON
FALLA DE ORIGEN

2.2 Ecuaciones de movimiento de Euler-Lagrange

En la mayoría de los casos, la cinemática de un robot puede ser descrita de manera matricial y sistemática utilizando la convención de Denavit-Hartenberg (Spong y Vidyasagar 1989).

Este método consiste en un producto de cuatro transformaciones básicas que se listan a continuación:

1. Rotación alrededor del eje z_{i-1} por un ángulo θ_i
2. Traslación a lo largo del eje z_{i-1} por una distancia d_i
3. Traslación a lo largo del eje x_i por una distancia a_i
4. Rotación alrededor del eje x_i por un ángulo α_i

Estas transformaciones sirven para describir la relación traslacional y rotacional entre elementos adyacentes. Suponiendo que tenemos dos sistemas coordenados denotados por $i-1$ e i , respectivamente, entonces existe una sola matriz de transformación homogénea que representa las coordenadas del sistema i con respecto al sistema $i-1$. Los sistemas coordenados tienen dos características adicionales:

- El eje x_i es perpendicular al eje z_{i-1}
- El eje x_i interseca al eje z_{i-1} ,

como se muestra en la Figura 2.1. Bajo estas condiciones se puede afirmar que existen valores únicos para a , d , α y θ . Si la articulación i es de revolución, se tiene con base en Denavit-Hartenberg,

$${}^{i-1}A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

TESIS CON
FALLA DE ORIGEN

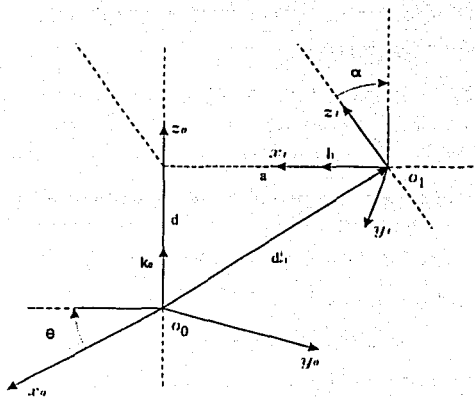


Figura 2.1: Ejes coordenados de acuerdo con las suposiciones de Denavit-Hartenberg.

o bien

$${}^{i-1}\mathbf{A}_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.1)$$

donde ${}^{i-1}\mathbf{A}_i$ representa la forma general de la matriz de transformación de coordenadas homogéneas. Es conveniente notar que el método de Denavit-Hartenberg sólo se puede aplicar a aquellos manipuladores cuyo ángulo de rotación θ_i se mide con respecto al eslabón inmediato anterior

En general, todos los elementos no nulos en la matriz ${}^{i-1}\mathbf{A}_i$ son función de θ_i ($\theta_1, \theta_2, \dots, \theta_i$), que es la variable de articulación del elemento i . α_i, a_i y d_i son parámetros conocidos de la estructura cinemática del brazo. Con el fin de deducir las ecuaciones de movimiento que son aplicadas a articulaciones de revolución, se utiliza la variable q_i para representar la coordenada generalizada de la articulación i , i.e, $q_i \equiv \theta_i$.

Por otro lado, las ecuaciones de movimiento general de un manipulador se pueden expresar convenientemente mediante la aplicación directa del método de Euler-Lagrange a sistemas no conservativos. Cuando es posible describir la cinemática de un robot utilizando Denavit-Hartenberg, la aplicación del método de Euler-Lagrange resulta en una descripción algorítmica y compacta de las ecuaciones de movimiento del manipulador. El algoritmo se representa mediante operaciones matriciales y facilita tanto el análisis como su realización en una computadora.

A continuación se presenta la descripción de la ecuación de Euler-Lagrange

$$\tau_i = \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} \quad i = 1, 2, \dots, n \quad (2.2)$$

donde

\mathcal{L} = función lagrangiana = energía cinética \mathcal{K} - energía potencial \mathcal{P} ;

\mathcal{K} = energía cinética total del brazo;

\mathcal{P} = energía potencial total del brazo;

q_i = coordenada generalizada del brazo;

\dot{q}_i = primera derivada respecto al tiempo de la coordenada generalizada q_i ;

τ_i = fuerza (o par) generalizado aplicado al sistema en la articulación i para mover el elemento i .

La ecuación de Euler-Lagrange requiere de la elección adecuada de un conjunto de coordenadas generalizadas para describir el sistema. Las coordenadas generalizadas son un conjunto de coordenadas de un sistema con respecto a un sistema de coordenadas de referencia que describen completamente la localización, posición y orientación. En general, las coordenadas generalizadas son las variables de las articulaciones definidas en cada una de las matrices de transformación de coordenadas homogéneas. En el caso de una articulación giratoria, $q_i = \theta_i$ (que es nuestro caso), mientras que para una articulación prismática $q_i = d_i$, que es la distancia recorrida por la articulación.

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN

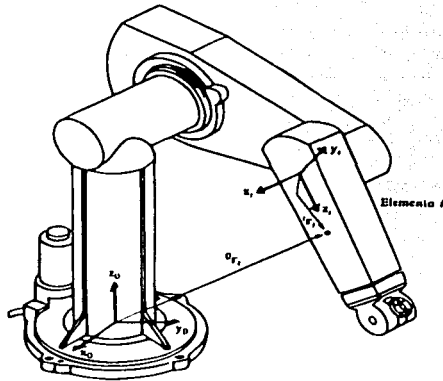


Figura 2.2: Un punto ${}^i r_i$ en el elemento i

2.3 Modelo dinámico de un robot

Como se vió anteriormente, para utilizar la metodología de Euler-Lagrange es necesario conocer las energías cinética y potencial del sistema a modelar. De esta forma, primero se obtendrán las ecuaciones que describen la energía cinética y energía potencial de un robot, para en una segunda parte obtener el modelo dinámico del mismo mediante la ecuación de Euler-Lagrange.

2.3.1 Energía cinética del manipulador

La obtención de la energía cinética de un robot requiere un conocimiento de la velocidad de cada partícula del mismo. A continuación se deducirá la velocidad de un punto fijo en el elemento i y se explorarán los efectos del movimiento de otra articulación sobre los puntos en este elemento.

Con referencia a la Figura 2.2, sea ${}^i r_i$ un punto fijo y en reposo en el elemento i , expresado en coordenadas homogéneas con respecto al sistema de coordenadas del elemento i :

$${}^i r_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}. \quad (2.3)$$

Sea ${}^0 r_i$ el mismo punto ${}^i r_i$ con respecto al sistema de coordenadas de la base, ${}^{i-1} A_i$ la matriz de transformación de coordenadas homogéneas que relaciona el desplazamiento espacial del sistema de coordenadas del elemento i -ésimo con respecto al sistema de coordenadas $(i-1)$ -ésimo, y ${}^0 T_i$ la matriz de transformación de coordenadas con respecto a la base; entonces, ${}^0 r_i$ está relacionado con el punto ${}^i r_i$ por

$${}^0 r_i = {}^0 T_i {}^i r_i, \quad (2.4)$$

donde

$${}^0 T_i = {}^0 A_1 {}^1 A_2 \dots {}^{i-1} A_i \quad i = 1, 2, \dots, n. \quad (2.5)$$

La velocidad de ${}^i r_i$ expresada en el sistema de coordenadas de la base se puede expresar como

$$\begin{aligned} {}^0 v_i \equiv v_i &= \frac{d}{dt} ({}^0 r_i) = \frac{d}{dt} ({}^0 T_i {}^i r_i) \\ {}^0 v_i &= {}^0 \dot{A}_1 {}^1 A_2 \dots {}^{i-1} A_i {}^i r_i + {}^0 A_1 {}^1 \dot{A}_2 \dots {}^{i-1} A_i {}^i r_i + \dots \\ &\quad + {}^0 A_1 {}^1 A_2 \dots {}^{i-1} \dot{A}_i {}^i r_i + {}^0 T_i \dot{{}^i r}_i \\ &= \left(\sum_{j=1}^i \frac{\partial {}^0 T_i}{\partial q_j} \dot{q}_j \right) {}^i r_i. \end{aligned} \quad (2.6)$$

La forma compacta anterior se obtiene porque $\dot{{}^i r}_i = 0$. La derivada parcial de ${}^0 A_i$ con respecto a q_j se puede calcular fácilmente con la ayuda de una matriz Q_i , que para una articulación de revolución, se define como:

$$Q_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.7)$$

TESIS CON
FALLA DE ORIGEN

Entonces se sigue que

$$\frac{\partial^{i-1} A_i}{\partial q_i} = \frac{\partial^{i-1} A_i}{\partial \theta_i} = Q_i^{i-1} A_i. \quad (2.8)$$

De lo anterior se obtiene,

$$\frac{\partial^0 A_i}{\partial q_j} = \begin{cases} {}^0 A_1 {}^1 A_2 \dots {}^{j-2} A_{j-1} Q_j^{j-1} A_j \dots {}^{i-1} A_i & \text{para } j \leq i \\ 0 & \text{para } j > i \end{cases} \quad (2.9)$$

para $i = 1, 2, \dots, n$. La ecuación (2.9) se puede interpretar como el efecto de movimiento de la articulación j sobre todos los puntos en el elemento i . Con el fin de simplificar las notaciones, se define

$$U_{ij} = \frac{\partial^0 A_i}{\partial q_j}.$$

Entonces la ecuación (2.9) se puede escribir como sigue para $i = 1, 2, \dots, n$:

$$U_{ij} = \begin{cases} {}^0 T_{j-1} Q_j^{j-1} T_i & \text{para } j \leq i \\ 0 & \text{para } j > i \end{cases} \quad (2.10)$$

Utilizando esta notación, v_i está dado por

$$v_i = \left(\sum_{j=1}^i U_{ij} \dot{q}_j \right) {}^i r_i. \quad (2.11)$$

Es conveniente apuntar que la derivada parcial de ${}^{i-1} A_i$ con respecto a q_i resulta en una matriz que no retiene la estructura de una matriz de transformación de coordenadas homogéneas. Para una articulación giratoria, el efecto de premultiplicar ${}^{i-1} A_i$ por Q_i es equivalente a intercambiar los elementos de las dos primeras filas de ${}^{i-1} A_i$, negando todos los elementos de la primera fila y anulando los elementos de las filas tres y cuatro. La ventaja de utilizar las matrices Q_i es que es posible utilizar todavía las matrices ${}^{i-1} A_i$ y aplicar las operaciones anteriores a ${}^{i-1} A_i$ cuando se premultiplica por Q_i .

Después de obtener la velocidad de la articulación de cada elemento, se necesita encontrar la energía cinética del elemento i . Sea K_i la energía

cinética del elemento i , $i = 1, 2, \dots, n$, expresada en el sistema de coordenadas de la base, y sea $d\mathcal{K}_i$ la energía cinética de una partícula con masa diferencial dm en el elemento i , entonces:

$$d\mathcal{K}_i = \frac{1}{2}(\dot{x}_i^2 + \dot{y}_i^2 + \dot{z}_i^2)dm \quad (2.12)$$

$$= \frac{1}{2}\text{traza}(\mathbf{v}_i \mathbf{v}_i^T)dm = \frac{1}{2}\text{Tr}(\mathbf{v}_i \mathbf{v}_i^T)dm \quad (2.13)$$

donde se utiliza un operador traza¹ en lugar de un producto escalar de vectores en la ecuación anterior para formar el tensor del cual se puede obtener la matriz de inercia del elemento \mathbf{J}_i (o matriz de pseudoinercia). Sustituyendo \mathbf{v}_i de la ecuación (2.11), la energía cinética de la masa diferencial es:

$$\begin{aligned} d\mathcal{K}_i &= \frac{1}{2}\text{Tr} \left(\sum_{p=1}^i \mathbf{U}_{ip} \dot{q}_p {}^i \mathbf{r}_i \left(\sum_{r=1}^i \mathbf{U}_{ir} \dot{q}_r {}^i \mathbf{r}_i \right)^T \right) dm \\ &= \frac{1}{2}\text{Tr} \left(\sum_{p=1}^i \sum_{r=1}^i \mathbf{U}_{ip} {}^i \mathbf{r}_i {}^i \mathbf{r}_i^T \mathbf{U}_{ir}^T \dot{q}_p \dot{q}_r \right) dm \\ &= \frac{1}{2}\text{Tr} \left(\sum_{p=1}^i \sum_{r=1}^i \mathbf{U}_{ip} ({}^i \mathbf{r}_i dm {}^i \mathbf{r}_i^T) \mathbf{U}_{ir}^T \dot{q}_p \dot{q}_r \right). \end{aligned} \quad (2.14)$$

La matriz \mathbf{U}_{ij} es la velocidad de cambio de los puntos ${}^i \mathbf{r}_i$ sobre el elemento i relativo al sistema de coordenadas de la base cuando q_j cambia. Es constante para todos los puntos en el elemento i e independiente de la distribución de masa del elemento i . También \dot{q}_i es independiente de la distribución de masa del elemento i , por lo que la energía cinética del elemento i está dado por:

$$\mathcal{K}_i = \int d\mathcal{K}_i = \frac{1}{2}\text{Tr} \left(\sum_{p=1}^i \sum_{r=1}^i \mathbf{U}_{ip} \left(\int {}^i \mathbf{r}_i {}^i \mathbf{r}_i^T dm \right) \mathbf{U}_{ir}^T \dot{q}_p \dot{q}_r \right) \quad (2.15)$$

¹ $\text{Tr}(\mathbf{A}) = \sum_{i=1}^n a_{ii}$

El término integral es la inercia de todos los puntos en el elemento i , (Fu *et al.* 1989):

$$J_i = \int {}^i r_i {}^i r_i^T dm = \begin{bmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int x_i y_i dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int x_i z_i dm & \int y_i z_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i dm & \int y_i dm & \int z_i dm & \int dm \end{bmatrix} \quad (2.16)$$

J_i se puede expresar en un tensor de inercia como:

$$J_i = \begin{bmatrix} \frac{-I_{xx} + I_{yy} + I_{zz}}{2} & I_{xy} & I_{xz} & m_i \bar{x}_i \\ I_{xy} & \frac{I_{xx} - I_{yy} + I_{zz}}{2} & I_{yz} & m_i \bar{y}_i \\ I_{xz} & I_{yz} & \frac{I_{xx} + I_{yy} - I_{zz}}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix} \quad (2.17)$$

donde ${}^i \bar{r}_i = (\bar{x}_i, \bar{y}_i, \bar{z}_i, 1)^T$ es el vector centro de masa del elemento i desde el sistema de coordenadas del elemento i -ésimo y expresado en el sistema de coordenadas del elemento i -ésimo.

De esta forma, la energía cinética total \mathcal{K} de un robot es:

$$\mathcal{K} = \sum_{i=1}^n \mathcal{K}_i = \frac{1}{2} \sum_{i=1}^n \text{Tr} \left(\sum_{p=1}^i \sum_{r=1}^i U_{ip} J_i U_{ir}^T \dot{q}_p \dot{q}_r \right) \quad (2.18)$$

$$= \frac{1}{2} \sum_{i=1}^n \sum_{p=1}^i \sum_{r=1}^i (\text{Tr} (U_{ip} J_i U_{ir}^T) \dot{q}_p \dot{q}_r) \quad (2.19)$$

que es una cantidad escalar. Obsérvese que las J_i son dependientes de la distribución de masa del elemento i y no de su posición o velocidad de movimiento, y se expresan con respecto al sistema de coordenadas i -ésimo. Por tanto, se necesita calcular la J_i solamente una vez para obtener la energía cinética de un robot.

2.3.2 Energía potencial de un robot

Sea \mathcal{P} la energía potencial de un robot y sea \mathcal{P}_i la energía potencial de cada uno de sus elementos:

$$\mathcal{P}_i = m_i g^T {}^0 \bar{r}_i = -m_i g^T ({}^0 A_i {}^i \bar{r}_i) \quad i = 1, 2, \dots, n. \quad (2.20)$$

Entonces, la energía potencial total del brazo se puede obtener sumando todas las energías potenciales de cada elemento:

$$\mathcal{P} = \sum_{i=1}^n \mathcal{P}_i = \sum_{i=1}^n -m_i \bar{g}^T ({}^0 A_i {}^i \bar{r}_i), \quad (2.21)$$

donde $\bar{g} = [g_x, g_y, g_z, 0]^T$ es un vector de gravedad expresado en el sistema de coordenadas de la base. Por ejemplo, si la gravedad va en la dirección del eje z_0 , se obtiene $\bar{g} = [0, 0, -g, 0]^T$, donde g es la constante gravitacional ($g = 9.8062 m/s^2$).

2.3.3 Ecuaciones de movimiento de un robot

De las ecuaciones (2.18) y (2.21), la función lagrangiana $\mathcal{L} = \mathcal{K} - \mathcal{P}$ está dada por:

$$\mathcal{L} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i (\text{Tr} (U_{ij} J_i U_{ik}^T) \dot{q}_p \dot{q}_r) + \sum_{i=1}^n m_i \bar{g}^T ({}^0 A_i {}^i \bar{r}_i). \quad (2.22)$$

Aplicando la formulación de Euler-Lagrange a la función lagrangiana del robot se obtiene la ecuación de movimiento de un robot de n grados de libertad

$$\tau(t) = H(q(t))\ddot{q}(t) + C(q(t), \dot{q}(t))\dot{q}(t) + g(q(t)) + D\dot{q}(t) + f_c(\dot{q}(t)) \quad (2.23)$$

donde:

$\tau \in \mathbb{R}^n$ = vector de par generalizado aplicado en las articulaciones $i = 1, 2, \dots, n$; esto es,

$$\tau = [\tau_1, \tau_2, \dots, \tau_n]^T \quad (2.24)$$

$q \in \mathbb{R}^n$ = vector de las variables de las articulaciones del brazo y se puede expresar como:

$$q = [q_1, q_2, \dots, q_n]^T \quad (2.25)$$

$H(q) \in \mathbb{R}^{n \times n}$ = matriz de inercia de cuyos elementos son:

$$h_{ij} = \sum_{j=\max(i,j)}^n \text{Tr}(U_{ki} J_k U_{kj}^T) \quad i, j = 1, 2, \dots, n \quad (2.26)$$

El coeficiente h_{ij} está relacionado con la aceleración de las variables de articulación. En particular, para $i = j$, h_{ii} está relacionado con la aceleración de la articulación i , donde el par τ_i actúa, mientras que para $i \neq j$, h_{ij} está relacionado al par (ó fuerza) de reacción inducido por la aceleración de la articulación j y actuando en la articulación i o viceversa. Como la matriz de inercia es simétrica y $\text{Tr}(A) = \text{Tr}(A^T)$, se puede demostrar que $h_{ij} = h_{ji}$.

$C(q, \dot{q})\dot{q} \in \mathbb{R}^n$ vector de fuerzas Centrífugas y de Coriolis, de $n \times 1$ donde los elementos de $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ estan dados por (Spong y Vidyasagar 1989):

$$c_{ij} = \sum_{k=1}^n c_{kji} \dot{q}_k \quad i, j = 1, 2, \dots, n \quad (2.27)$$

donde

$$c_{kji} = \frac{1}{2} \left\{ \frac{\partial h_{ij}}{\partial q_k} + \frac{\partial h_{ik}}{\partial q_j} - \frac{\partial h_{kj}}{\partial q_i} \right\}. \quad (2.28)$$

Los coeficientes c_{kji} son conocidos como los símbolos de Christoffel de primera clase. Los dos últimos índices, ij , están relacionados con las velocidades de las articulaciones i y j , cuya interrelación dinámica induce un par (ó fuerza) de reacción en la articulación i . Así el primer índice i se relaciona siempre con la articulación donde se "sienten" los pares (o fuerzas) de reacción inducidos por la velocidad. En particular, para $i = j$, c_{ii} está relacionado con la fuerza centrífuga generada por la velocidad de la articulación i y "sentida" en la articulación i , mientras que para $i \neq j$, c_{ij} está relacionada con la fuerza de Coriolis generada por las velocidades de las articulaciones i y j "sentida" en la articulación i . Además, se tiene que $c_{kji} = c_{jki}$.

$g(q) \in \mathbb{R}^n$ = vector de $n \times 1$ de la fuerza gravitacional cuyos elementos son:

$$g(q) = [g_1, g_2, \dots, g_n]^T$$

con

$$g_i = - \sum_{j=i}^n (m_j \bar{g}^T U_{ji}^j \bar{r}_j) \quad i = 1, 2, \dots, n. \quad (2.29)$$

$D \in \mathbb{R}^{n \times n}$ = matriz de coeficientes de fricción viscosa en las articulaciones.

$f_c \in \mathbb{R}^{n \times 1}$ = vector de términos de fricción de Coulomb cuyos elementos están dados por $\alpha_i \text{sign}(q_i)$, $i = 1, \dots, n$, $\alpha_i > 0$.

2.4 Propiedades del modelo dinámico

En esta sección varias propiedades relacionadas con el modelo (2.23) son presentadas. Algunas de ellas son propiedades físicas del sistema mientras que otras se obtienen del procedimiento que derivó en el modelo dinámico. De ahora en adelante, la norma Euclideana para vectores es usada, es decir $\text{Cap} \|\alpha\| \triangleq (\alpha^T \alpha)^{1/2}$, donde $\alpha \in \mathbb{R}^n$. La norma de una matriz A es la correspondiente norma inducida $\|A\| \triangleq (\lambda_{\max}(A^T A))^{1/2}$, donde $\lambda_{\max}(\cdot)$ y $\lambda_{\min}(\cdot)$ denotan el máximo y mínimo valor propio de una matriz, respectivamente. $\lambda_i(\cdot)$ denota el i -ésimo valor propio.

Las propiedades presentadas en esta sección, así como su demostración, se encuentran en Arteaga Pérez (1998).

2.4.1 Propiedades de las matrices

Propiedad 2.4.1. La matriz de inercia generalizada $H(q)$ es simétrica definida positiva. \triangle

Propiedad 2.4.2. La matriz:

$$N(q, \dot{q}) \triangleq \dot{H}(q) - 2C(q, \dot{q}) \in \mathbb{R}^{n \times n} \quad (2.30)$$

es antisimétrica. \triangle

Puesto que hay muchas posibles definiciones para $C(q, \dot{q})$, es conveniente señalar que $\dot{q}^T (\dot{H}(q) - 2C(q, \dot{q})) \dot{q} = 0$ siempre, no importando que definición

de $C(q, \dot{q})$ sea usada. Esta propiedad es muy importante, ya que está relacionada con la estructura pasiva del robot manipulador y es frecuentemente empleada en la demostración de muchos esquemas de control.

Propiedad 2.4.3. El vector $h_c(q, \dot{q})$ satisface las igualdades:

$$h_c(q, x, y) = C(q, x)y = C(q, y)x = h_c(q, y, x) \quad \forall x, y \in \mathbb{R}^n. \quad (2.31)$$

△

2.4.2 Cotas para normas de matrices

Las normas de las matrices del modelo (2.23) juegan un rol importante en el control de robots ya que son útiles para el diseño de muchos esquemas de control. Las cotas de las normas son especialmente ventajosas cuando la teoría de Lyapunov es usada (Khalil 1996).

Propiedad 2.4.4. La matriz de inercia generalizada $H(q)$ satisface:

$$\lambda_{\min}(H(q))\|y\|^2 \leq y^T H(q)y \leq \lambda_{\max}(H(q))\|y\|^2 \quad \forall y \in \mathbb{R}^n. \quad (2.32)$$

△

Propiedad 2.4.5. La matriz $H^{-1}(q)$ existe y satisface:

$$\lambda_{\max}^{-1}(H(q))\|y\|^2 \leq y^T H^{-1}(q)y \leq \lambda_{\min}^{-1}(H(q))\|y\|^2 \quad \forall y \in \mathbb{R}^n. \quad (2.33)$$

△

Propiedad 2.4.6. La matriz de inercia generalizada $H(q)$ satisface:

$$0 < \lambda_h \leq \|H(q)\| \leq \lambda_H < \infty \quad (2.34)$$

$$\lambda_h = \min_{\forall q \in \mathbb{R}^n} \lambda_{\min}(H(q)) \quad \lambda_H = \max_{\forall q \in \mathbb{R}^n} \lambda_{\max}(H(q)). \quad (2.35)$$

△

Propiedad 2.4.7. La matriz $H^{-1}(q)$ satisface:

$$0 < \sigma_h \leq \|H(q)\| \leq \sigma_H < \infty \quad (2.36)$$

$$\sigma_h = \min_{\forall q \in \mathbb{R}^n} \lambda_{\max}^{-1}(H(q)) \quad \sigma_H = \max_{\forall q \in \mathbb{R}^n} \lambda_{\min}^{-1}(H(q)). \quad (2.37)$$

△

Las propiedades anteriores son muy importantes debido a que muchas funciones de Lyapunov empleadas para mostrar la estabilidad de un esquema de control hacen uso de ellas.

Propiedad 2.4.8. El vector de torques gravitacionales $g(q)$ satisface:

$$\|g(q)\| \leq \sigma_g \quad \sigma_g > 0 \quad q \in \mathbb{R}^n. \quad (2.38)$$

△

2.4.3 Propiedades relacionadas con todo el modelo

Finalmente, se presentan dos propiedades que caracterizan a todo el modelo del sistema. Estas propiedades son utilizadas para el diseño de una gran variedad de esquemas de control, en particular en los empleados en el presente trabajo.

Propiedad 2.4.9. Con una apropiada definición de sus parámetros, el modelo (2.23) puede ser escrito como:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + D\dot{q} + g(q) + f_c(\dot{q}) = u = Y(q, \dot{q}, \ddot{q})\varphi, \quad (2.39)$$

donde $Y(q, \dot{q}, \ddot{q})$ se conoce como el regresor, $\varphi \in \mathbb{R}^p$ es el vector de parámetros y p es el número de parámetros obtenidos en el modelo del robot. △

Esta propiedad es muy importante porque indica que el modelo del robot es lineal en sus parámetros, lo que es útil en el diseño de controladores adaptables y en el desarrollo de procedimientos de identificación de los parámetros del robot.

Propiedad 2.4.10. La ecuación dinámica de un robot manipulador rígido dada por (2.23) define un mapeo pasivo de $\psi \rightarrow \dot{q}$, i. e.

$$\langle \psi, \dot{q} \rangle \triangleq \int_0^T \dot{q}^T \psi dt \geq -\beta, \quad (2.40)$$

it1 1 ulQ1 β > 0 PVT > 0r e12 u - Dq̇ ∈ ℝ^{n×1}. △

2.5 Inclusión de la dinámica de los motores

La ecuación (2.23) representa la dinámica de una cadena interconectada de cuerpos rígidos ideales, suponiendo que una fuerza generalizada τ actúa en las articulaciones. Se puede presumir que la i -ésima componente τ_i del vector de fuerza generalizada τ es un torque en el eje z_{i-1} . Esta fuerza generalizada es producida por un actuador, el cual puede ser eléctrico, hidráulico o neumático.

Los actuadores eléctricos son generalmente motores de corriente directa (CD) con imán permanente. El modelo dinámico de un motor CD con imán permanente controlado con la armadura es descrito por el siguiente conjunto de ecuaciones de acuerdo a la Figura 2.3

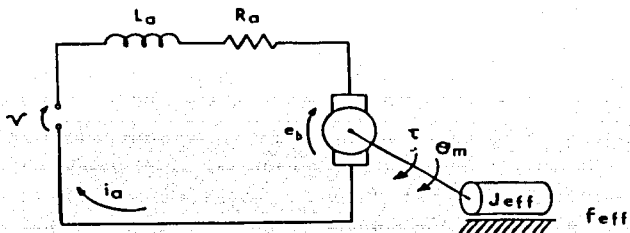


Figura 2.3: Armadura de motor

$$\tau_m = K_a i_a \quad (2.41)$$

$$v = R_a i_a + L_a \frac{di_a}{dt} + e_b \quad (2.42)$$

$$e_b = K_b \dot{\theta}_m \quad (2.43)$$

$$\theta_m = r\theta, \quad (2.44)$$

donde

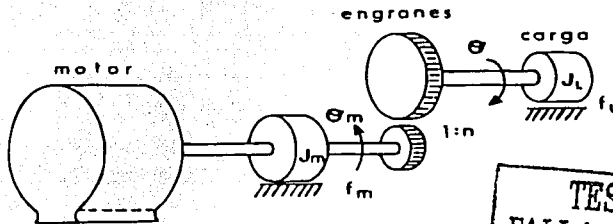


Figura 2.4: Motor y Engranes

TESIS CON
FALLA DE ORIGEN

K_a	constante motor-par [Nm/A]
R_a	resistencia de armadura [Ω]
L_a	inductancia de armadura [H]
K_b	constante de contra reacción electromotriz [V s]
τ_m	par en el eje del motor [rad]
i_a	corriente de armadura [A]
e_b	fuerza electromotriz [V]
θ_m	posición angular del eje del motor [rad]
θ	posición angular del eje de la carga [rad]
r	relación de reducción de los engranes (en general $r \gg 1$)
v	voltaje de armadura [V].

La ecuación de movimiento para el sistema mecánico de la Figura 2.4 es:

$$J_m \ddot{\theta}_m + f_m \dot{\theta}_m = \tau_m - \frac{\tau}{r}, \quad (2.45)$$

donde τ es el par neto aplicado después del juego de engranes sobre el eje de la carga, J_m es la inercia del rotor y $f_m \dot{\theta}_m$ es el efecto debido a la fricción del rotor con sus soportes. Debido a que la constante de tiempo eléctrica es mucho más rápida que la constante de tiempo mecánica, tradicionalmente la inductancia de armadura L_a es despreciada ($L_a \approx 0$) en la ecuación (2.42). Tomando en consideración lo anterior, substituyendo i_a de (2.42) en (2.41) y a su vez τ_m de (2.41) en (2.45) se obtiene la siguiente ecuación:

$$J_m \ddot{\theta}_m + f_m \dot{\theta}_m + \frac{K_a e_b}{R_a} + \frac{\tau}{r} = \frac{K_a}{R_a} v. \quad (2.46)$$

Finalmente, utilizando las relaciones de (2.43) y (2.44) y agrupando, la ecuación (2.46) se describe como:

$$J_m \ddot{\theta} + \left(f_m + \frac{K_a K_b}{R_a} \right) \dot{\theta} + \frac{\tau}{r^2} = \frac{K_a}{r R_a} v. \quad (2.47)$$

La ecuación anterior relaciona el voltaje v aplicado a la armadura del motor con el par τ aplicado a la carga en términos de la posición, velocidad y aceleración angular de la misma.

Retomando las ecuaciones (2.23) y (2.47), se obtendrá una ecuación dinámica para manipuladores cuyos actuadores sean motores de CD. La ecuación (2.47) es para un solo motor. Para el caso de n motores de CD (actuadores), el modelo dinámico que relaciona a los n voltajes v_i aplicados a las armaduras con los pares τ_i está dado por:

$$J_{m_i} \ddot{q} + \left(f_{m_i} + \frac{K_{a_i} K_{b_i}}{R_{a_i}} \right) \dot{q} + \frac{\tau_i}{r_i^2} = \frac{K_{a_i}}{r_i R_{a_i}} v_i, \quad (2.48)$$

o en forma matricial:

$$D_j \ddot{q} + D_f \dot{q} + D_n \tau = D_k v \quad (2.49)$$

para $i = 1, 2, \dots, n$, donde las matrices D_j , D_f , D_n y D_k son diagonales con elementos J_{m_i} , $f_{m_i} + \frac{K_{a_i} K_{b_i}}{R_{a_i}}$, $\frac{1}{r_i^2}$ y $\frac{K_{a_i}}{r_i R_{a_i}}$ respectivamente. De la última ecuación despejando τ , substituyendo en (2.23) y agrupando terminos se obtiene:

$$u = (H(q) + J) \ddot{q} + C(q, \dot{q}) \dot{q} + D \dot{q} + g(q) + f_c(\dot{q}). \quad (2.50)$$

En la ecuación anterior J es una matriz diagonal de $n \times n$ con elementos del producto $D_n^{-1} D_j$; D es una matriz diagonal de $n \times n$ con elementos del producto $D_n^{-1} D_f$, la cual representa la fricción por torque, u es un vector de $n \times 1$ que tiene componentes dadas por

$$u = D_n^{-1} D_k v \quad (2.51)$$

y tiene unidades de torque. Para la ecuación (2.50), las propiedades mencionadas en la sección anterior se siguen satisfaciendo.

TESIS CON
FALLA DE ORIGEN

Capítulo 3

Descripción de los algoritmos de control

Introducción

En este capítulo se describen los algoritmos de control que se implementarán en el brazo robot. Se pueden dividir en dos tipos: robustos y adaptables. Dentro de los primeros se contempla uno de los controladores básicos o convencionales, conocido como control Proporcional-Derivativo (PD). Algunos algoritmos de control necesitan de ganancias K_p y K_d , error de posición y error de velocidad de la articulación, mientras que en otros casos se necesita también conocimiento del modelo dinámico del robot, sus parámetros y ganancias propias del algoritmo de control (es decir, no necesariamente K_p y K_d).

Se trabajará con cuatro algoritmos de control robusto y uno adaptable. De los algoritmos que se describirán sólo tres se han implementado en simulación, con la excepción del control DCAL y el control PD que se utiliza cotidianamente en la industria.

3.1 Control Proporcional-Derivativo robusto

El controlador Proporcional-Derivativo (PD) es usado en la industria debido a su buen desempeño y a que sólo necesita de la lectura de la posición actual del sistema. La acción derivativa (velocidad del error) se determina por medio de métodos numéricos, lo que evita la instalación de tacómetros u otros dispositivos de lectura de velocidad, reduciendo con esto el costo y peso en el sistema a controlar.

Controlador Proporcional-Derivativo (PD)

La acción de control proporcional-derivativo se define por la siguiente ley de control:

$$\tau = -K_d \dot{\tilde{q}} - K_p \tilde{q} \quad (3.1)$$

donde $\tilde{q} = q - q_d$, representa el error de posición y $\dot{\tilde{q}}$ representa el error de velocidad, q y q_d son la trayectoria seguida y la trayectoria deseada por el robot la cual depende del tiempo, respectivamente. K_d y $K_p \in \mathbb{R}^{n \times n}$ son matrices diagonales definidas positivas, las cuales representan las ganancias derivativa y proporcional del controlador, respectivamente. La Figura 3.1 muestra un diagrama de bloques del controlador

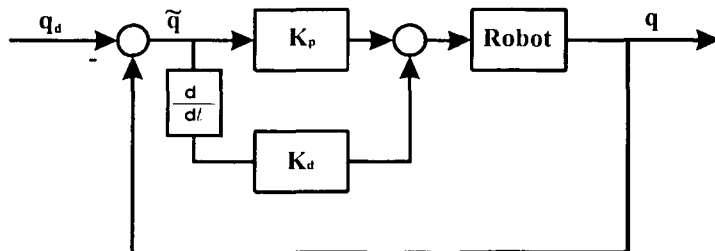


Figura 3.1: Sistema de control PD.

Esta ley de control proporciona un seguimiento asintótico de la trayectoria deseada, para cada una de las articulaciones del robot, la acción derivativa

lo hace un controlador de alta sensibilidad, es decir que responde a la velocidad del cambio del error y produce una corrección significativa antes de que la magnitud del error se vuelva demasiado grande. Aunque el control derivativo no afecta en forma directa al error en estado estacionario, añade amortiguamiento al sistema, con el inconveniente que la acción derivativa puede llegar a aumentar el ruido y saturar los actuadores.

3.2 Control Proporcional-Derivativo con estimación de velocidad

Durante la última década se ha incrementado el desarrollo en la investigación del diseño de estrategias de control mucho más sofisticadas para los robots. A pesar de estos esfuerzos, la mayoría de los robots industriales son controlados por alguna clase de retroalimentación de estado lineal (en la literatura comúnmente referido como un controlador PD). Una rasgo característico del controlador de retroalimentación de estado es que requiere de la lectura o medición de posición y velocidad. En la práctica, este requerimiento generalmente no se cumple ya que en aplicaciones de robótica sensores de alta precisión son usados para obtener la información de la posición pero no se suelen incluir tacómetros para medición de velocidad. Por esta razón una gran variedad de estrategias de control de robots han sido propuestos recientemente que no necesitan de la medición de velocidad puesto que incorporan un observador de velocidad en el lazo de control.

El controlador que se describirá en esta sección consta de dos partes (Berghuis y Nijmeijer 1994):

1. Un observador lineal que genera un error de estado estimado del error de posición.
2. Un controlador de retroalimentación lineal, que emplea el estimado del error de estado.

El hecho de que el controlador de retroalimentación de estado estimado es fácil de implementar y sólo necesita información de la posición, hace el controlador propuesto particularmente interesante desde un punto de vista práctico. Además produce un seguimiento práctico en el sentido que el error de seguimiento tiende a una vecindad de cero ya sea exponencialmente o

TESIS CON
FALLA DE ORIGEN

asintóticamente.

Las Propiedades 2.4.2, 2.4.3, 2.4.4 y 2.4.8 facilitan el diseño del controlador.

Para resolver el problema de control de seguimiento para (2.23) usando sólo retroalimentación de posición, se considera el sistema de control lineal de retroalimentación de salida de la Figura 3.2, donde q_d representa la trayectoria deseada a seguir por el robot.

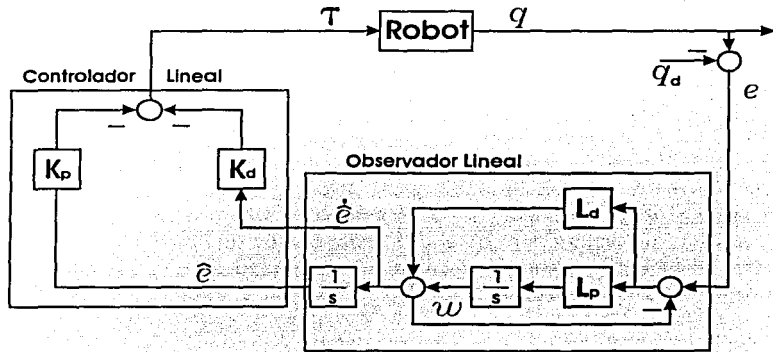


Figura 3.2: Sistema de control-observador.

La ley de control propuesta es dada por:

$$\tau = -K_d \dot{\hat{e}} - K_p \hat{e}. \quad (3.2)$$

El observador propuesto es:

$$\begin{aligned} \dot{\hat{e}} &= w + L_d(\tilde{q} - \hat{e}) \\ \dot{w} &= L_p(\tilde{q} - \hat{e}), \end{aligned} \quad (3.3)$$

donde $\tilde{q} = q - q_d$ es el error de posición, $\hat{e} \equiv \tilde{q} - q_d$ es el error estimado y \hat{q} es el estimado de q . $K_p > 0$ y $K_d > 0$ representan las ganancias proporcional y derivativa del controlador respectivamente, $L_p > 0$ y $L_d > 0$ representan

las ganancias proporcional y derivativa del observador respectivamente. Este sistema de control consiste de dos partes; el observador lineal (3.3) que genera un estimado del error de seguimiento \tilde{q} y el controlador lineal (3.2) que utiliza los estimados del error de estado en lazo de cerrado. Lo cual permite hacer la siguiente suposición en la estructura de K_p , K_d , L_p y L_d .

Para que el sistema sea estable en lazo cerrado se requiere que K_p , K_d , L_p , L_d satisfagan respectivamente (Berghuis y Nijmeijer 1994):

$$K_p = \lambda K_d \quad K_d = (k_d + \gamma)I \quad (3.4)$$

$$L_p = \lambda l_d \quad L_d = (l_d + \lambda)I \quad (3.5)$$

donde I es una matriz identidad y $\lambda > 0$, $k_d > 0$, $\gamma > 0$ y $l_d > 0$, son todos escalares. Las condiciones iniciales son cero para el esquema de observación.

3.3 Controlador Adaptable de Compensación Deseada (DCAL)

Una alternativa al uso de los observadores para solventar la falta de medición de velocidad consiste en filtrar la señal de posición de la articulación. El control propuesto en esta sección representa una modificación de la Ley de Adaptación de Compensación Deseada (DCAL) y técnicas de filtrado (Burg *et al.* 1997). Específicamente, un filtro que genera el error de velocidad; basado en la lectura de la posición de la articulación es propuesto para reemplazar la lectura de velocidad de la articulación y reducir la sobrecarga computacional. Este algoritmo de control asegura un seguimiento asintótico para la posición y la velocidad de las articulaciones. Este algoritmo de control ya se implemento en un brazo robot de dos articulaciones construido por Integrated Motion Inc.

El sistema dinámico del robot es lineal en los parámetros en el sentido:

$$Y(q_d, \dot{q}_d, \ddot{q}_d)\varphi = H(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + F_d\dot{q}_d + g(q_d) \quad (3.6)$$

TESIS CON
FALLA DE ORIGEN

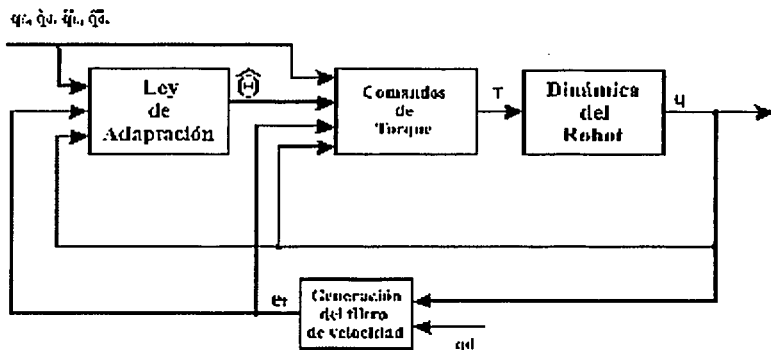


Figura 3.3: Diagrama de bloques del controlador.

donde q_d , \dot{q}_d y $\ddot{q}_d \in \mathbb{R}^n$, son la posición, velocidad y aceleración deseadas para la articulación, respectivamente. $Y_d = Y(q_d, \dot{q}_d, \ddot{q}_d)$ es el regresor deseado y contiene variables conocidas relacionadas a la estructura del robot.

El objetivo de control es proporcionar el torque de entrada para el seguimiento de posición de la articulación basado en el conocimiento de los parámetros inciertos del sistema y del conocimiento de las posiciones de las articulaciones. El propósito del controlador es forzar a las articulaciones a seguir una trayectoria deseada que satisface

$$\|q_d\| < \bar{q}_{d1}, \quad \|\dot{q}_d\| < \bar{q}_{d2} \quad \text{y} \quad \|\ddot{q}_d\| < \bar{q}_{d3}. \quad (3.7)$$

\bar{q}_{d1} , \bar{q}_{d2} y \bar{q}_{d3} son constantes conocidas positivas. Un diagrama de bloques del controlador propuesto se muestra en la Figura 3.3.

Para compensar la falta de una lectura directa de la señal de velocidad, el error de seguimiento de la articulación es filtrado usando el siguiente filtro dinámico:

$$\dot{p} = -(k-1)p - (k^2+1)\bar{q} \quad (3.8)$$

$$e_r = k\bar{q} + p \quad (3.9)$$

donde la variable $e_f \in \mathbb{R}^n$ es la salida del filtro y $p \in \mathbb{R}^n$ es usada como una variable interna del mismo. La constante positiva k es la ganancia del filtro. La entrada es el error de seguimiento de posición de la articulación, \tilde{q} , el cual será usado para determinar el comportamiento de \hat{q} . Por lo tanto como un sustituto de la lectura de \tilde{q} , la señal e_f será usada. La condición inicial en las variables internas se eligen como $p(0) = k\tilde{q}(0)$, con $e_f = 0$.

De esta forma, la ley de control se define como (Burg pmua1997):

$$\tau = Y_d \hat{\varphi} - k e_f - \tilde{q}. \tag{3.10}$$

Nótese que ha substituido φ por $\hat{\varphi}$. $\hat{\varphi}$ indica que el valor exacto del vector de parámetros es conocido y sólo se cuenta con un valor nominal aproximado.

3.4 Control usando un observador lineal

Como ya se ha mencionado, la mayoría de los esquemas de control requieren de medición de la velocidad de las articulaciones para su implementación. Mientras que este problema se solventó en la sección 3.3 por medio de un filtro. En esta sección se propone el uso de un observador lineal para estimar \tilde{q} , (Arteaga Pérez 2001). Además produce un seguimiento práctico en el sentido que el error de seguimiento y observación tienden a una vecindad de cero ya sea exponencial o asintóticamente.

Considérese las siguientes definiciones de los errores de posición y observación : $\tilde{q} \triangleq q - q_d$ y $z \triangleq q - \hat{q}$, respectivamente, donde q_d es una trayectoria deseada y acotada para q y \hat{q} denota su estimado.

Para la definición del controlador se requiere además de:

$$\dot{q}_r \triangleq \dot{q}_d - \Lambda(\hat{q} - q_d) = \dot{q}_d - \Lambda\tilde{q} + \Lambda \tag{3.11}$$

$$s \triangleq \dot{q} - \dot{q}_r = \tilde{\dot{q}} + \Lambda\tilde{q} - \Lambda z \tag{3.12}$$

$$\dot{q}_o \triangleq \dot{\hat{q}} - \Lambda z \tag{3.13}$$

$$r \triangleq \dot{q} - \dot{q}_o = \dot{z} + \Lambda z \tag{3.14}$$

donde Λ es está dada por $\Lambda \triangleq I + 2\Lambda_1$, y Λ_1 es una matriz diagonal positiva

definida. El controlador propuesto es el siguiente:

$$\begin{aligned} \tau &= \hat{H}(q)\dot{q}_r + \hat{C}(q, \dot{q}_r)\dot{q}_r + \hat{D}\dot{q}_r + \hat{g}(q) - K_R(\dot{q}_o - \dot{q}_r) - K_P(\tilde{q} + z) \\ &\triangleq Y(q, \dot{q}_r, \ddot{q}_r)\hat{\varphi} - K_R(\dot{q}_o - \dot{q}_r) - K_P(\tilde{q} + z) \end{aligned} \quad (3.15)$$

donde $K_R \triangleq \frac{1}{2}K_P + K_{R1}$, y $K_P, K_{R1} \in \mathbb{R}^{n \times n}$ son matrices diagonales positivas definidas. En este caso \cdot denota el valor nominal de los parámetros del modelo. El observador lineal se define como:

$$\dot{\hat{q}} = \hat{q}_o + \lambda z + k_d z \quad (3.16)$$

$$\ddot{\hat{q}}_o = \ddot{q}_r + k_d \Lambda z \quad (3.17)$$

donde k_d es una constante real positiva, llamada ganancia del observador. Las condiciones iniciales para el observador están dadas por $\hat{q}_o(0) = -(\Lambda z(0) + k_d z(0))$ y $\hat{q}(0) = q(0)$.

3.5 Control y estimación de parámetros con sólo lectura de posición de articulación

Los tres controladores vistos en las secciones anteriores tienen propiedades de robustes, es decir, estabilizan el sistema a pesar de que los parámetros nominales no concuerdan exactamente con los parámetros reales del manipulador. Una técnica comúnmente propuesta para elevar el desempeño de la ley de control es la adaptación paramétrica. En este caso, los parámetros del modelo, $\hat{\varphi}$, varían de acuerdo con la ley de adaptación dependiendo de los requerimientos del control.

Para poder diseñar un esquema de control adaptable, se requiere información sobre la estructura del modelo del sistema. Es el caso de los robots manipuladores, pues sus modelos dinámicos son descritos por ecuaciones diferenciales no lineales, con estructura conocida. Se puede suponer que sólo algunos parámetros son inciertos o de difícil medición, como inercias y masa de objetos manipulados por el robot.

El esquema de control adaptable que a continuación se presenta hace uso de un observador lineal para evitar la lectura de velocidad de las articulaciones (Arteaga Pérez 2002). Este esquema de control se basa en el esquema

de control con observador lineal de la sección anterior.

A las definiciones anteriores se agrega el error paramétrico dado por $\tilde{\varphi} \triangleq \hat{\varphi} - \Theta$, donde (\cdot) denota el estimado de (\cdot) .

El controlador y el observador siguen estando dado por (3.15)–(3.17). Sin embargo, esta vez $\hat{\varphi}$ en (3.15) no permanece constante, sino que varía con la siguiente ley de adaptación:

$$\dot{\hat{\varphi}}(t) = -\Gamma^{-1} Y^T(q, \dot{q}_r, \ddot{q}_r)(\hat{s} + \hat{\tau}), \quad (3.18)$$

donde $\Gamma \in \mathbb{R}^{n \times n}$ es la matriz diagonal definida positiva de ganancias de adaptación, con $\hat{s} + \hat{\tau} \triangleq (\hat{q} - \dot{q}_d) + \Lambda \bar{q}$. Las condiciones iniciales de los parámetros son cero. El acotamiento de los parámetros estimados es garantizado.

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN

Capítulo 4

Implementación de los algoritmos

Este capítulo trata lo relacionado a la implementación de los algoritmos de control descritos en el Capítulo 3. Estos algoritmos se implementaron en un robot industrial de seis grados de libertad ($n = 6, i = 1, 2, \dots, n$), de la marca *CRS Robotics* modelo A465 que se encuentra en el Laboratorio de Robótica de la División de Estudios de Posgrado de la Facultad de Ingeniería (DEPFI). La implementación se realizó desde una PC, mediante el software LabWindows 4.0.1 de la compañía National Instruments. Esto es posible gracias a la adaptación del brazo robot antes mencionado a un modo experimental (Castillo Sánchez 2002). Para conocer más detalles del equipo, consultar el Apéndice A.

Se programaron dos tipos de trayectorias de seguimiento para el robot; los resultados obtenidos en los experimentos se presentan en tablas y gráficas comparativas para errores de posición, velocidad y observación (en el caso que lo amerite) con el fin de evaluar el desempeño de cada algoritmo.

4.1 Método de sintonización de ganancias

Todos los algoritmos de control que se presentarán en el Capítulo 3 poseen, directa o indirectamente, un término proporcional y derivativo. Para realizar una comparación justa entre ellos, estas ganancias deben ser utilizadas en todos los casos. Si bien se pueden obtener mediante prueba y error, se decidió emplear un método de ajuste de ganancias propuesto por Ziegler y Nichols en 1942 (Åström y Wittenmark 1990), quienes se basaron en la práctica para desarrollarlo. Este método se basa en un lazo proporcional que sirve para calcular la ganancia a la cual el sistema comienza a oscilar. Con la ganancia y el periodo de oscilación, se pueden obtener las ganancias de un controlador PID.

El procedimiento se lleva a cabo con base en los siguientes pasos:

1. Utilizando un escalón como referencia y un control proporcional como se observa en la Figura 4.1, se comienza con un valor de ganancia K_p pequeño que se va incrementando hasta que el sistema comienza a oscilar de manera periódica.

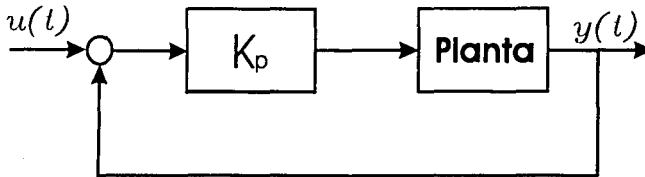


Figura 4.1: Lazo cerrado sólo con ganancia proporcional.

2. Registrar la ganancia crítica del controlador, $K_c = K_p$, y el periodo de oscilación T_c de la salida del controlador, como se muestra en la Figura 4.2.
3. Ajustar los parámetros del controlador de acuerdo con la Tabla 4.1:

De la Tabla 4.1 se obtiene directamente K_p ; las ganancias K_d y K_i se

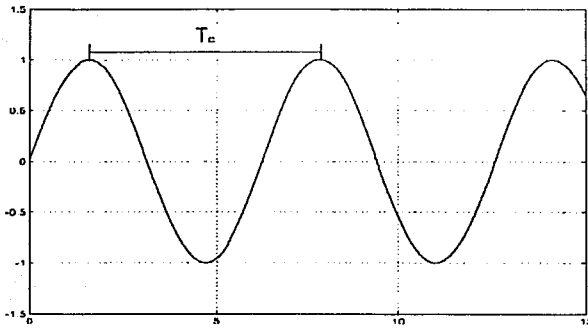


Figura 4.2: Respuesta de la planta a lazo cerrado con ganancia crítica.

	K_p	T_n	T_v
P	$0.5K_c$	—	—
PI	$0.45K_c$	$0.85T_c$	—
PID	$0.6K_c$	$0.5T_c$	$0.12T_c$

Tabla 4.1: Ajuste de parámetros.

obtienen de acuerdo con la siguiente ecuación:

$$u(t) = K_p y(t) + \frac{K_p}{T_n} \int_0^t y(\sigma) d\sigma + K_p T_v \dot{y}(t), \quad (4.1)$$

donde se observa que: $\frac{K_p}{T_n}$ es la ganancia integral y $K_p T_v$ es la ganancia derivativa. Este método se utilizó en cada articulación del robot. Como se puede observar, el método no contempla una sintonización para el controlador PD, por lo cual se utilizaron los parámetros de un PID haciendo cero la ganancia integrativa.

En las gráficas de la Figura 4.3 se presentan los resultados obtenidos al aplicar el método de Ziegler-Nichols, donde se puede apreciar el periodo crítico para cada una de las articulaciones. Con base en la Tabla 4.1, la

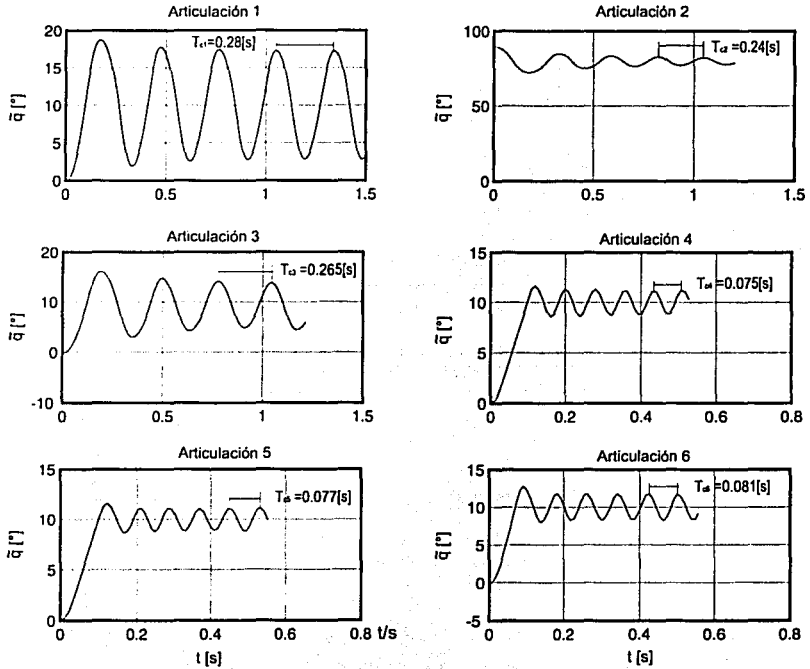


Figura 4.3: Respuesta de las articulaciones en lazo cerrado con ganancia crítica.

ecuación (4.1) y los resultados obtenidos en la Figura 4.3 se presentan en la Tabla 4.2 las ganancias obtenidas para cada una de las articulaciones junto con la ganancia crítica.

Articulación	K_{c_i}	T_{c_i}	K_{p_i}	K_{d_i}
1	70	0.280	42.0	1.4112
2	150	0.240	90.0	2.5920
3	120	0.265	72.0	2.2896
4	440	0.075	264.0	2.3760
5	460	0.077	276.0	2.5500
6	310	0.081	186.0	1.8079

Tabla 4.2: Ganancias K_{p_i} y K_{d_i} ajustadas por método de oscilación

4.2 Equivalencia de ganancias

En esta sección se presenta un análisis de la relación de ganancias existente entre el controlador PD y los controladores restantes, con el propósito de hacer el cálculo de ganancias sólo una vez.

4.2.1 Ecuaciones de los controladores

Se presentan aquí las ecuaciones de cada uno de los controladores considerando el modelo completo, es decir, se considera la dinámica de los motores. Las ecuaciones se presentan en relación con el voltaje.

Control PD

El ajuste de ganancias de la Tabla 4.2 está dado en terminos de voltaje, es decir

$$v = -K'_d \dot{q} - K'_p \tilde{q} \quad (4.2)$$

sin embargo la entrada de control utilizada en los algoritmos presentados en el Capítulo 3 es el momento/fuerza generalizado τ . De acuerdo con la ecuación (2.50), la relación entre τ y v está dada por

$$\tau = -D_n^{-1} D_k v. \quad (4.3)$$

Esto quiere decir que el PD neto que entra al sistema es

$$\begin{aligned}\tau &= -D_n^{-1}D_k K'_d \dot{\bar{q}} - D_n^{-1}D_k K'_p \bar{q} \\ &\triangleq -K_d \dot{\bar{q}} - K_p \bar{q}\end{aligned}\quad (4.4)$$

Estas ganancias K_d y K_p son las que tienen que ser siempre equivalentes. Por otro lado, para poder emplear los algoritmos del Capítulo 3 se requiere hacer

$$v = -D_k^{-1}D_n \tau_0, \quad (4.5)$$

donde τ_0 es la ley de control en términos de par. Por ello, alternativamente a (4.4), se puede utilizar (4.2) siempre y cuando el término PD se separe del resto de la ley de control.

Control PD con estimador de velocidad

Repetiendo el procedimiento aplicado a la ley de control PD, para (3.2) se tiene:

$$v = -K'_D \dot{\hat{e}} - K'_P \hat{e}. \quad (4.6)$$

Control DCAL

Aplicando el mismo procedimiento a (3.10),

$$\dot{v} = D_k^{-1}D_n(Y_d \dot{\hat{\varphi}} - k'e_r - \bar{q}). \quad (4.7)$$

Control con observador lineal

Aplicando el procedimiento a (3.15)

$$\begin{aligned}v &= -D_k^{-1}D_n(\hat{H}(q)\ddot{q}_r + \hat{C}(q, \dot{q}_r)\dot{q}_r + \hat{D}\dot{q}_r + \hat{g}(q) \\ &\quad - K'_R(\dot{q}_o - \dot{q}_r) - K'_P(\bar{q} + z)).\end{aligned}\quad (4.8)$$

Control adaptable con observador lineal

La ecuación de este apartado es la misma que la dada por (4.8). Sólo debe de agregarse la ley de adaptación dada en (3.18).

4.2.2 Cálculo de ganancias

En este apartado se deducirán una a una las relaciones de ganancias entre la ley de control PD y los restantes controladores. Esto se realizó a partir de las ecuaciones obtenidas en la Sección 4.2.1.

Controladores sin modelo

Control PD con estimación de velocidad

Con el fin de equiparar las ganancias para (3.2) y (3.3) se procede como sigue; primero se deriva \hat{e} :

$$\dot{\hat{e}} = \dot{\hat{q}} - \dot{q}_d, \quad (4.9)$$

luego se substituye \hat{e} y $\dot{\hat{e}}$ en (3.2)

$$\begin{aligned} \tau &= -K_d(\dot{\hat{q}} - \dot{q}_d) - K_p(\hat{q} - q_d) \\ &= -K_d(\dot{\hat{q}} - \dot{q}_d) - K_p(\hat{q} - q_d) + K_d(\dot{q} - \dot{q}) + K_p(q - q). \end{aligned} \quad (4.10)$$

Desarrollando los binomios y agrupando terminos se tiene:

$$\tau = -K_d(\dot{q} - \dot{q}_d) - K_p(q - q_d) - K_d(\dot{\hat{q}} - \dot{q}) - K_p(\hat{q} - q). \quad (4.11)$$

Recordando que $e = q - q_d$, se concluye que las ganancias K_d y K_p serán las mismas y por lo tanto la ecuación a implementar es (4.6).

Solo resta por determinar los valores de L_d y L_p que se utilizarán en el observador. Recordando las ecuaciones (3.4) y (3.5) se tiene

$$K_p = \lambda K_d \quad K_d = (k_d + \gamma)I$$

$$L_p = \lambda l_d \quad L_d = (l_d + \lambda)I$$

Como se observa, se necesita saber el valor de λ y l_d . De la ecuación (3.4) se calcula,

$$\lambda I = K_p K_d^{-1}. \quad (4.12)$$

El valor de l_d se obtuvo de manera experimental ($l_d = 250$).

Control DCAL

Para la ley de control dada por (3.10), sin considerar el modelo del robot se tiene:

$$\tau = -ke_f - e. \quad (4.13)$$

Con el fin de minimizar el error y mejorar el desempeño del controlador, se agrega la ganancia k_p , obteniendo la siguiente ecuación de control:

$$\tau = -ke_f - k_p e. \quad (4.14)$$

Si se compara esta ecuación con (3.1), se determina que la relación de ganancias es uno a uno, obteniéndose finalmente que la parte proporcional derivativa de la ley de control ésta dada por

$$\tau = -K_d e_f - K_p e. \quad (4.15)$$

Recordando (4.7) la ecuación a implementar es:

$$v = -K'_d e_f - K'_p e. \quad (4.16)$$

Control con observador lineal

Para la ley de control propuesta en (3.15), sin el modelo dinámico del robot se tiene¹:

$$\tau = -K_R(\dot{q}_o - \dot{q}_r) - \bar{K}_P(\tilde{q} + z) \quad (4.17)$$

Substituyendo (3.11) y (3.13) en (4.17) se tiene,

$$\begin{aligned} \tau &= K_R \dot{q}_d - K_R \hat{\dot{q}} - (K_R \Lambda + K_P) \tilde{q} + (2K_R \Lambda - K_P) z & (4.18) \\ &= K_R \dot{q}_d - K_R \hat{\dot{q}} - (K_R \Lambda + K_P) \tilde{q} + (2K_R \Lambda - K_P) z + K_R (\dot{q} - \hat{\dot{q}}) \\ &= -K_R \ddot{\tilde{q}} - (K_R \Lambda + K_P) \tilde{q} + K_R \dot{z} + (2K_R \Lambda - K_P) z \end{aligned}$$

Comparando término a término con (3.1), se obtienen las siguientes igualdades para la relación de ganancias,

$$K_R = K_d \quad (4.19)$$

$$\bar{K}_P = K_p - K_d \Lambda. \quad (4.20)$$

¹Con el fin de evitar confusiones en vez de utilizar K_P como en (4.17) se emplea \bar{K}_P para diferenciarla de la ganancia conocida K_P , que es distinta en significado a la que aparece en (4.17)

De las ecuaciones (4.17), (4.19) y (4.20) se tiene,

$$\tau = -K_d(\dot{q}_o - \dot{q}_r) - (K_p - K_d\Lambda)(\tilde{q} + z), \quad (4.21)$$

o bien

$$v = -K'_d(\dot{q}_o - \dot{q}_r) - (K'_p - K'_d\Lambda)(\tilde{q} + z). \quad (4.22)$$

El termino Λ se obtuvo de manera experimental, tomando el valor que proporcionó el mejor comportamiento del controlador.

Controladores con modelo

Control DCAL

Apartir de la ecuación (3.10) se tiene:

$$\tau = Y_d \hat{\varphi} - k e_f - e. \quad (4.23)$$

Substituyendo $Y_d \hat{\varphi}$ de (3.6) y u de (2.51) por τ , en la ecuación anterior se obtiene

$$\tau = D_n^{-1} D_k v = H(q_d) \ddot{q}_d + C(q_d, \dot{q}_d) \dot{q}_d + F_d \dot{q}_d + g(q_d) - k e_f - e. \quad (4.24)$$

Despejando v y agregando la ganancia k_p , como se realizó en (4.14)

$$v = D_k^{-1} D_n (H(q_d) \ddot{q}_d + C(q_d, \dot{q}_d) \dot{q}_d + F_d \dot{q}_d + g(q_d) - k e_f - k_p e), \quad (4.25)$$

de donde podemos obtener la siguiente relación de ganancias, de acuerdo con lo realizado para el caso sin modelo

$$K_d = D_k^{-1} D_n k \quad (4.26)$$

$$K_p = D_k^{-1} D_n k_p. \quad (4.27)$$

Despejando k y k_p se tiene:

$$k = D_n^{-1} D_k K_d \quad (4.28)$$

$$k_p = D_n^{-1} D_k K_p. \quad (4.29)$$

Finalmente, substituyendo (4.28) y (4.29) en (4.25) y retomando (4.7) se tiene la siguiente ecuación, que representa la ley de control a implementar

$$\mathbf{v} = D_k^{-1} D_n (H(q_d) \dot{q}_d + C(q_d, \dot{q}_d) \dot{q}_d + F_d \dot{q}_d + g(q_d)) - K'_d e_r + K'_p e, \quad (4.30)$$

donde \mathbf{v} representa el voltaje de entrada a los motores de CD.

Control con observador lineal

Considérese la ley de control propuesta en (3.15):

$$\tau = \hat{H}(q) \ddot{q}_r + \hat{C}(q, \dot{q}_r) \dot{q}_r + \hat{F} \dot{q}_r + \hat{G}(q) - K_R(\dot{q}_o - \dot{q}_r) - K_P(\bar{q} + z). \quad (4.31)$$

Substituyendo \mathbf{u} de (2.51) por τ , en la ecuación anterior se obtiene

$$D_n^{-1} D_k \mathbf{v} = \hat{H}(q) \ddot{q}_r + \hat{C}(q, \dot{q}_r) \dot{q}_r + \hat{F} \dot{q}_r + \hat{G}(q) - K_R(\dot{q}_o - \dot{q}_r) - K_P(\bar{q} + z). \quad (4.32)$$

Despejando \mathbf{v} se tiene

$$\mathbf{v} = D_n D_k^{-1} (\hat{H}(q) \ddot{q}_r + \hat{C}(q, \dot{q}_r) \dot{q}_r + \hat{F} \dot{q}_r + \hat{G}(q) - K_R(\dot{q}_o - \dot{q}_r) - K_P(\bar{q} + z)). \quad (4.33)$$

Para obtener la relación de ganancias, se realiza el mismo proceso que se utilizó en el caso sin modelo, obteniéndose lo siguiente:

$$K_R = D_n^{-1} D_k K'_d \quad (4.34)$$

$$K_P = D_n^{-1} D_k (K'_p - K'_d \Lambda). \quad (4.35)$$

Finalmente, substituyendo (4.34) y (4.35) en (4.33) y recordando (4.8) se tiene la siguiente ecuación donde \mathbf{v} representa el voltaje de salida que se aplicará a los motores de CD.

$$\mathbf{v} = D_n D_k^{-1} (\hat{H}(q) \ddot{q}_r + \hat{C}(q, \dot{q}_r) \dot{q}_r + \hat{F} \dot{q}_r + \hat{G}(q) - K_d(\dot{q}_o - \dot{q}_r) + (K_P - K_d \Lambda)(\bar{q} + z)). \quad (4.36)$$



Control adaptable

Retomando la ecuación(3.15),

$$\tau \triangleq Y(q, \dot{q}_r, \ddot{q}_r)\hat{\varphi} - K_R(\dot{q}_o - \dot{q}_r) - K_P(\bar{q} + z) \quad (4.37)$$

substituyendo u de (2.51) por τ , en la ecuación anterior se obtiene

$$D_n^{-1}D_k v \triangleq Y(q, \dot{q}_r, \ddot{q}_r)\hat{\varphi} - K_R(\dot{q}_o - \dot{q}_r) - K_P(\bar{q} + z). \quad (4.38)$$

Despejando v , substituyendo la relación de ganancias obtenida en (4.34) y (4.35) respectivamente y retomando (4.8) se tiene,

$$v \triangleq D_k^{-1}D_n(Y(q, \dot{q}_r, \ddot{q}_r)\hat{\varphi} - K'_d(\dot{q}_o - \dot{q}_r) + (K'_p - K'_d\Lambda)(\bar{q} + z)), \quad (4.39)$$

donde v representa el voltaje de salida que se aplicará a los motores de CD y $\hat{\varphi}$ se calcula de la integración del algoritmo de adaptación dado por (3.18).

4.3 Experimentos

Los experimentos que se realizaron consistieron en programar un par de trayectorias en cada una de las articulaciones del brazo robot para cada uno de los algoritmos de control presentados en el Capítulo 3. Los tipos de trayectorias son las siguientes:

1. Trayectoria suave: consiste en llevar a cada articulación del robot a una posición deseada final por medio de un polinomio de quinto orden. La ecuación que lo describe es la siguiente:

$$q_{di}(t) = a_i + b_it + c_it^2 + d_it^3 + e_it^4 + f_it^5 \quad (4.40)$$

2. Trayectoria senoidal: La ecuación que describe esta trayectoria se muestra a continuación(Reyes y Kelly 2001):

$$q_{di}(t) = (K_{1i} + K_{2i} \sin(\omega_i t))(1 - e^{K_{ei} t^3}) \quad (4.41)$$

donde K_{1i} representan un desplazamiento angular inicial, K_{2i} representa la amplitud de las oscilaciones, ω_i representa la frecuencias y K_{ei} es un valor de amortiguamiento para cada una de las articulaciones.

Se realizó un análisis comparativo con el propósito de observar el desempeño de cada uno de los algoritmos de control y poder determinar cual presenta el mejor desempeño con respecto a los demás algoritmos. El desempeño de cada uno de los algoritmos de control para cada una de las trayectorias se evaluó mediante el índice \mathcal{I} , que mide el promedio de la raíz cuadrada media del error de seguimiento, y está expresado por la siguiente ecuación:

$$\mathcal{I} = \sqrt{\frac{1}{T} \int_0^T \|\tilde{q}(t)\|^2 dt}. \quad (4.42)$$

Se escogió este índice de desempeño debido a que es el más adecuado para nuestro análisis comparativo, ya que este se hizo con base en el error que se genera en cada algoritmo en su seguimiento de trayectorias. El análisis se realizó mediante la presentación de los índices de desempeño para cada algoritmo de control en una tabla comparativa para el caso de los algoritmos que no incluyen la dinámica del robot y para el caso que incluye esta. Mediante graficas se presenta el comportamiento del error a lo largo de los experimentos para cada uno de los algoritmos de control y cada una las articulaciones del robot.

4.3.1 Experimentos sin utilizar modelo en la ley de control

Sólo se aplica a los algoritmos (4.4), (4.6), (4.16) y (4.22). A continuación se presentan las tablas de comparación y las graficas que muestran los errores de seguimiento para cada una de las articulaciones del robot.

La tabla 4.3 muestra los resultados de la primera trayectoria y la Tabla 4.4 para la segunda. Las gráficas correspondientes se muestran en las Figuras 4.4 a 4.9 y 4.10 a 4.15, respectivamente. A pesar que no se observa ninguna diferencia importante entre los distintos métodos, el Algoritmo (4.16) fue el que dio mejores resultados para ambas trayectorias.

Las magnitudes de los índices de desempeño para el caso de la trayectoria 1 son menores en un 40 % con respecto a los de la trayectoria 2. Esto se debió a que la velocidad del movimiento es lento en el caso de la trayectoria 1.

Articulación	Algoritmo			
	(4.4)	(4.6)	(4.16)	(4.22)
\mathcal{I}_1	0.312805	0.370498	0.271876	0.369740
\mathcal{I}_2	0.592262	0.582885	0.622130	0.580677
\mathcal{I}_3	0.860629	0.742073	0.712321	0.741227
\mathcal{I}_4	0.171340	0.110095	0.109876	0.180162
\mathcal{I}_5	0.185435	0.113038	0.119048	0.172680
\mathcal{I}_6	0.201095	0.147965	0.128728	0.186188
\mathcal{I}	1.137315	1.036575	1.005574	1.058421

Tabla 4.3: Tabla comparativa de índices de desempeño para la Trayectoria 1 sin modelo.

Articulación	Algoritmo			
	(4.4)	(4.6)	(4.16)	(4.22)
\mathcal{I}_1	0.705897	0.722828	0.667859	0.626976
\mathcal{I}_2	0.655217	0.658868	0.629472	0.659303
\mathcal{I}_3	0.648726	0.654253	0.625707	0.656815
\mathcal{I}_4	0.512057	0.483875	0.450236	0.480201
\mathcal{I}_5	0.509085	0.502318	0.488339	0.520142
\mathcal{I}_6	0.507268	0.551909	0.530796	0.560285
\mathcal{I}	1.458470	1.475025	1.398825	1.440224

Tabla 4.4: Tabla comparativa de índices de desempeño para la Trayectoria 2 sin modelo.

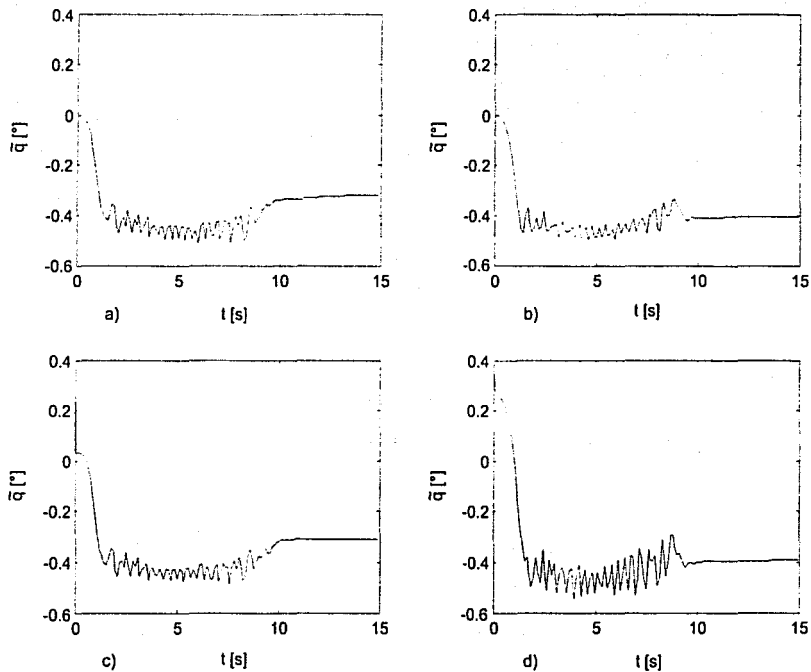


Figura 4.4: Errores de seguimiento Trayectoria 1, Articulación 1 sin modelo.

- a) Algoritmo (4.4). b) Algoritmo (4.6).
c) Algoritmo (4.16). d) Algoritmo (4.22).

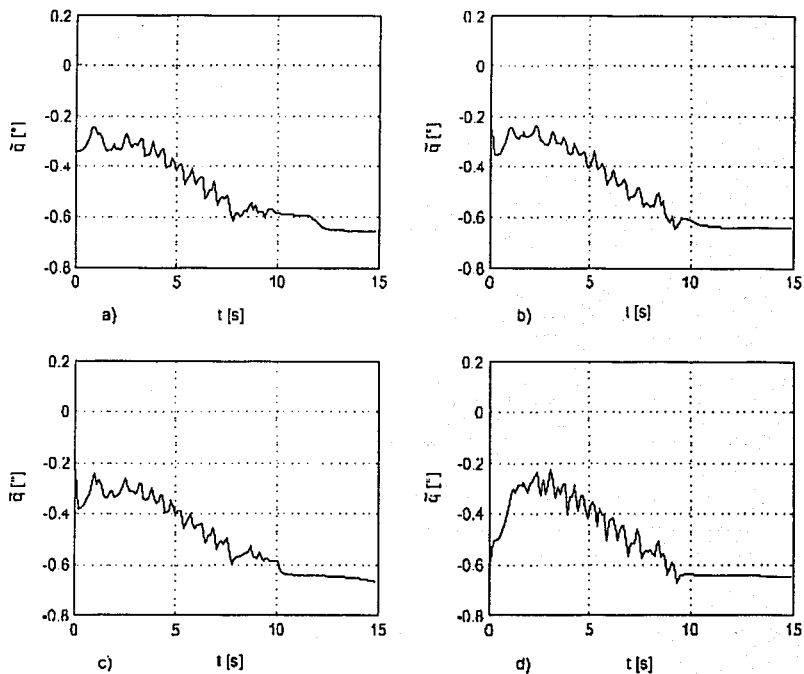


Figura 4.5: Errores de seguimiento Trayectoria 1, Articulación 2 sin modelo.

- a) Algoritmo (4.4). b) Algoritmo (4.6).
c) Algoritmo (4.16). d) Algoritmo (4.22).

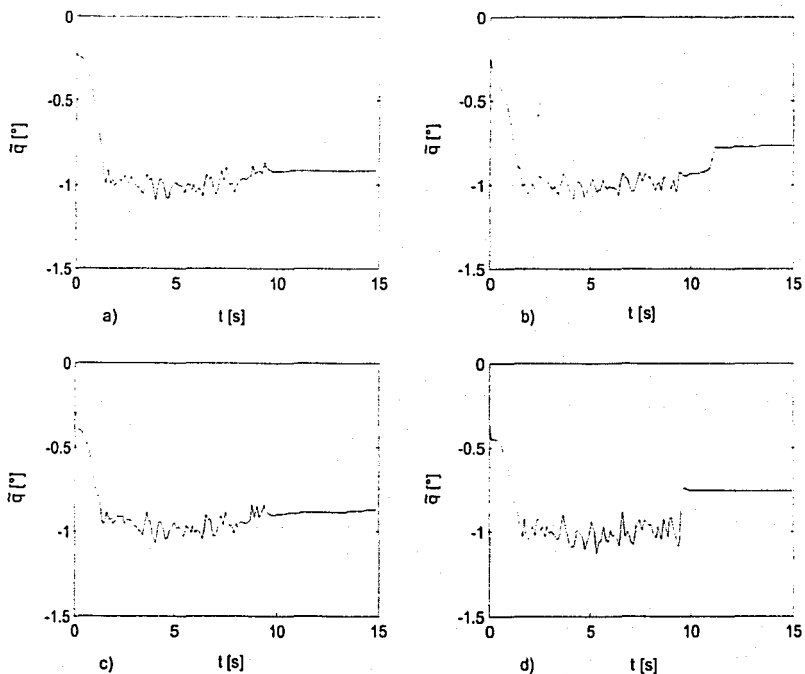


Figura 4.6: Errores de seguimiento Trayectoria 1, Articulación 3 sin modelo.

- a) Algoritmo (4.4). b) Algoritmo (4.6).
c) Algoritmo (4.16). d) Algoritmo (4.22).

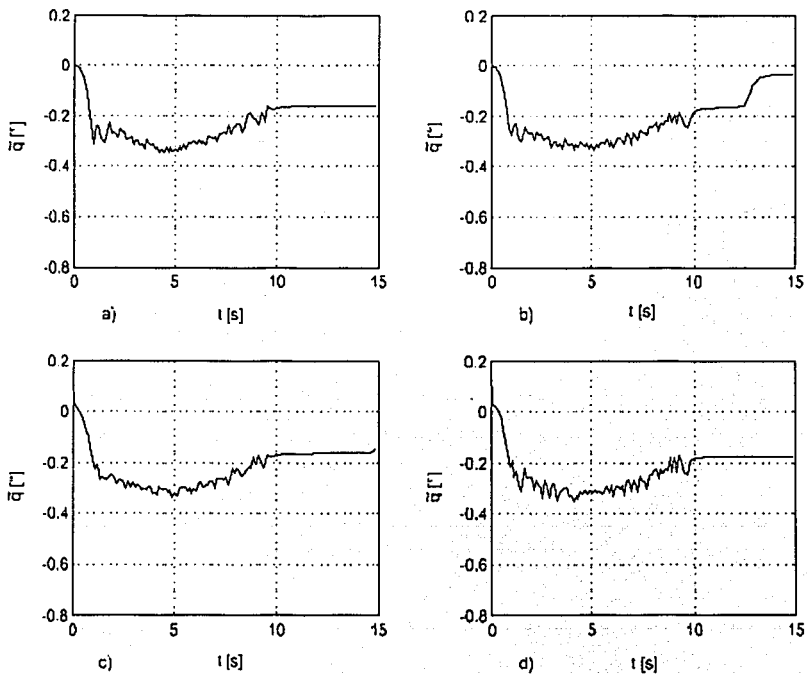


Figura 4.7: Errores de seguimiento Trayectoria 1, Articulación 4 sin modelo.

- a) Algoritmo (4.4). b) Algoritmo (4.6).
c) Algoritmo (4.16). d) Algoritmo (4.22).

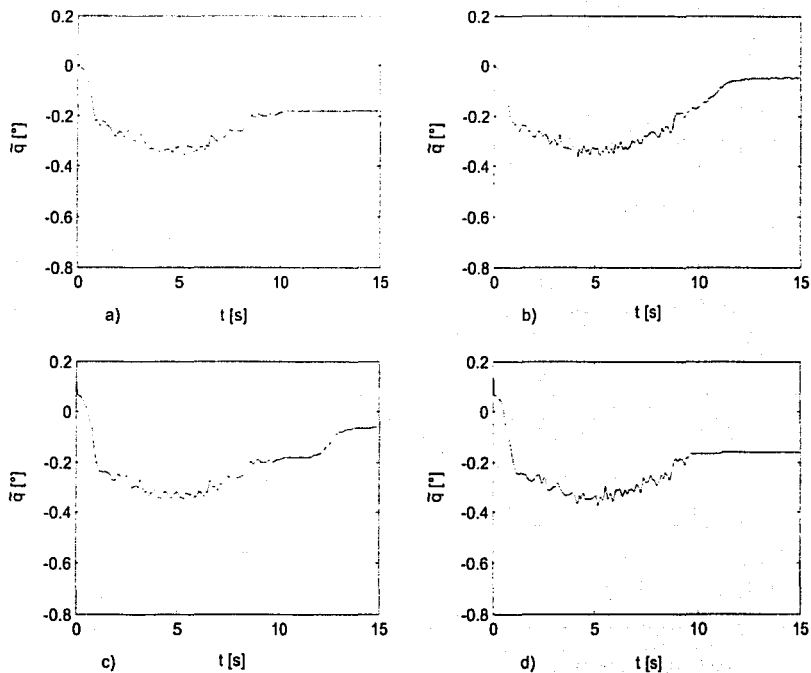


Figura 4.8: Errores de seguimiento Trayectoria 1, Articulación 5 sin modelo.

- a) Algoritmo (4.4). b) Algoritmo (4.6).
c) Algoritmo (4.16). d) Algoritmo (4.22).

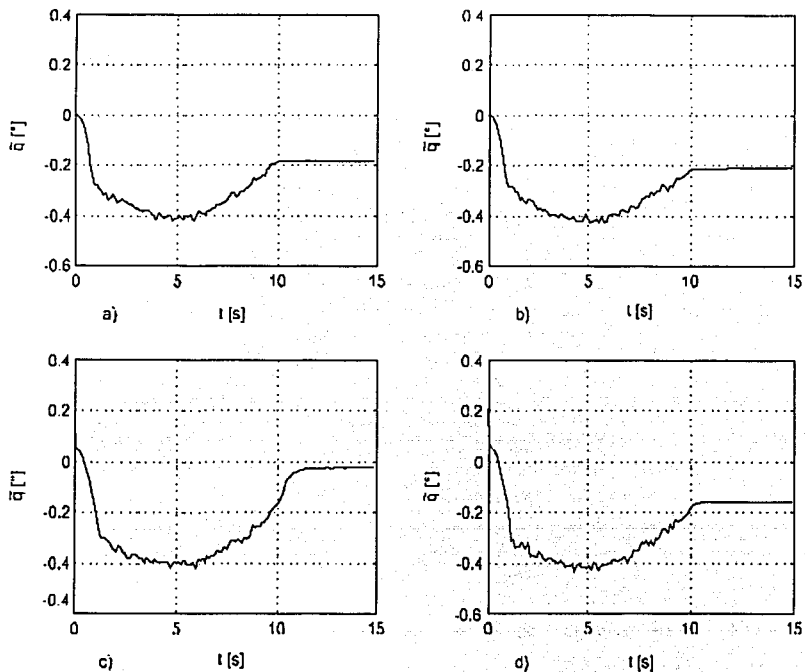


Figura 4.9: Errores de seguimiento Trayectoria 1, Articulación 6 sin modelo.

- a) Algoritmo (4.4). b) Algoritmo (4.6).
c) Algoritmo (4.16). d) Algoritmo (4.22).

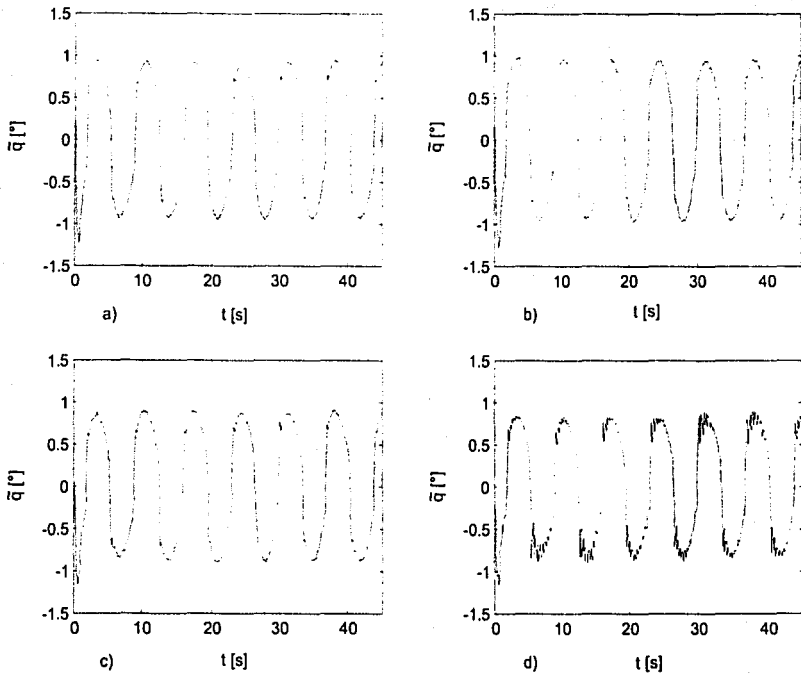


Figura 4.10: Errores de seguimiento Trayectoria 2, Articulación 1 sin modelo.

- a) Algoritmo (4.4). b) Algoritmo (4.6).
c) Algoritmo (4.16). d) Algoritmo (4.22).

TESIS CON
FALLA DE ORIGEN

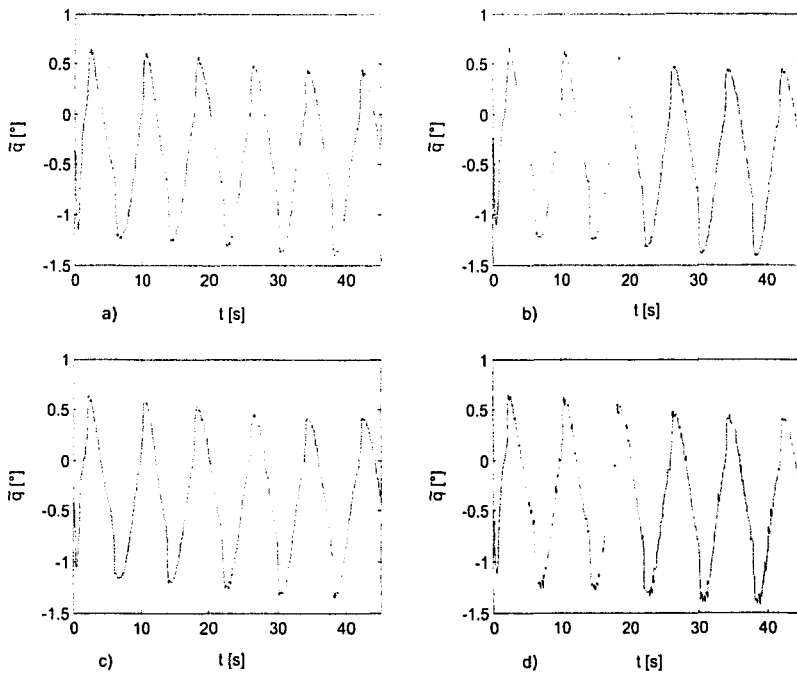


Figura 4.11: Errores de seguimiento Trayectoria 2, Articulación 2 sin modelo.
a) Algoritmo (4.4). b) Algoritmo (4.6).
c) Algoritmo (4.16). d) Algoritmo (4.22).

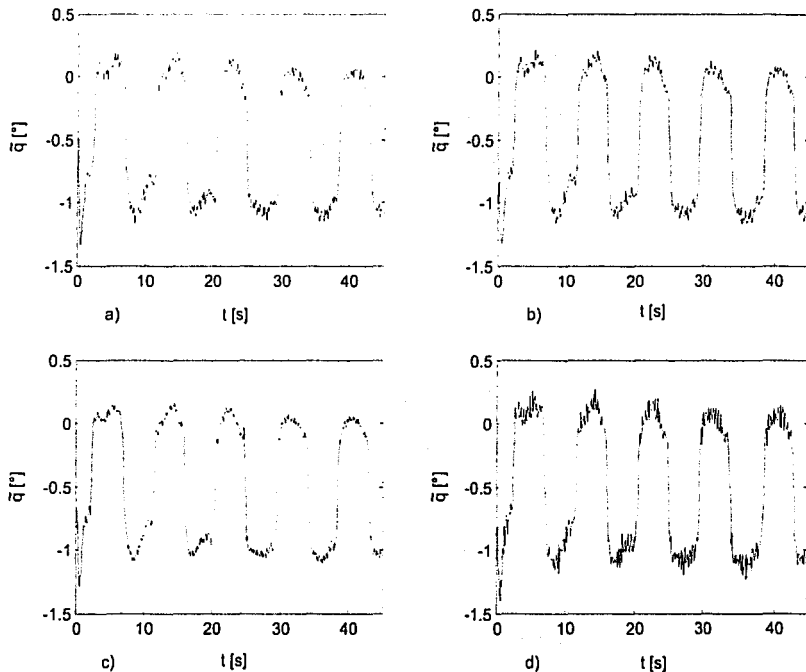


Figura 4.12: Errores de seguimiento Trayectoria 2, Articulación 3 sin modelo.

a) Algoritmo (4.4). b) Algoritmo (4.6).

c) Algoritmo (4.16). d) Algoritmo (4.22).

TESIS CON
FALLA DE ORIGEN

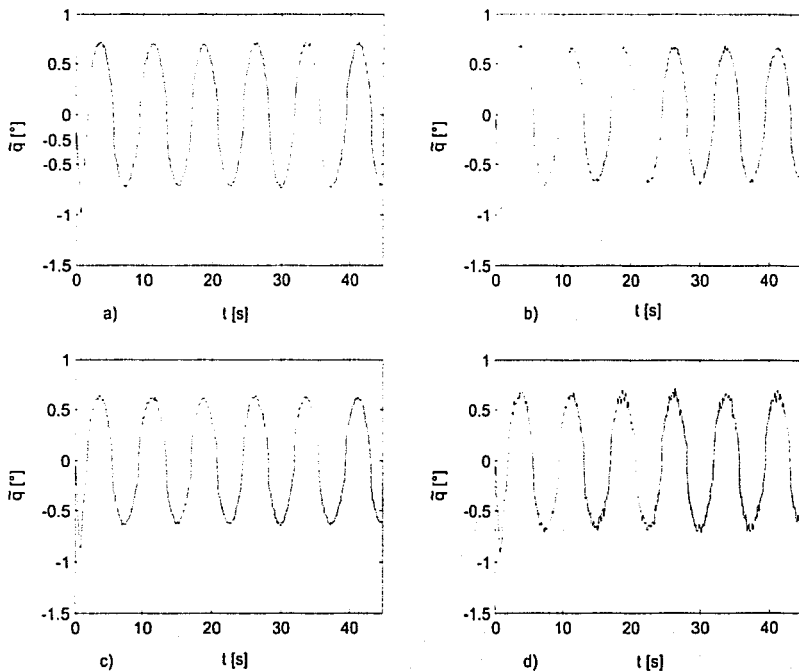


Figura 4.13: Errores de seguimiento Trayectoria 2, Articulación 4 sin modelo.

- a) Algoritmo (4.4). b) Algoritmo (4.6).
c) Algoritmo (4.16). d) Algoritmo (4.22).

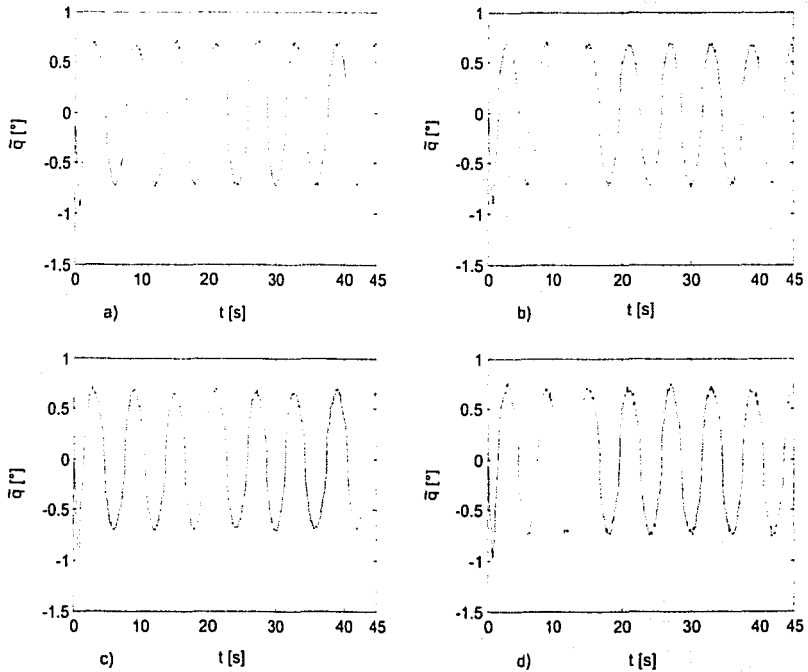


Figura 4.14: Errores de seguimiento Trayectoria 2, Articulación 5 sin modelo.

- a) Algoritmo (4.4). b) Algoritmo (4.6).
c) Algoritmo (4.16). d) Algoritmo (4.22).

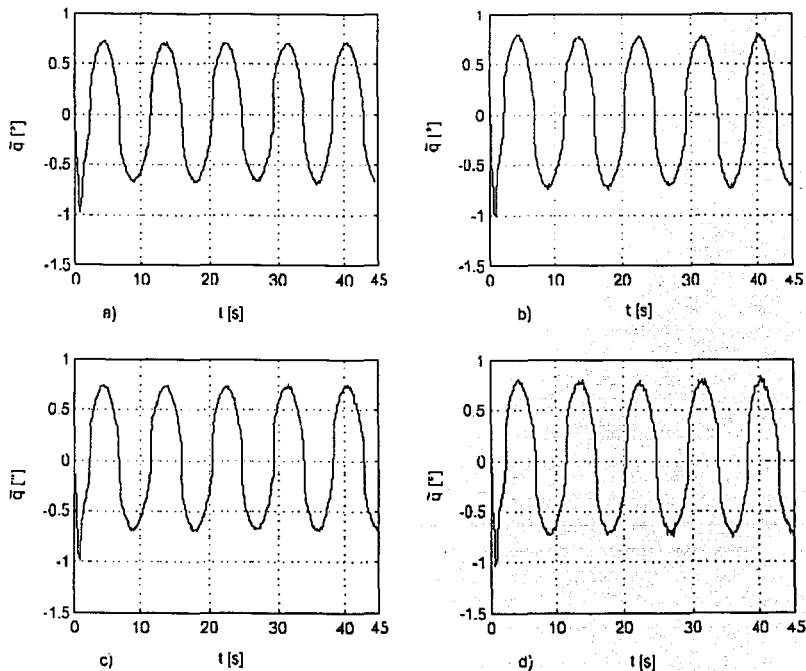
TESIS CON
FALLA DE ORIGEN

Figura 4.15: Errores de seguimiento Trayectoria 2, Articulación 6 sin modelo.

- a) Algoritmo (4.4). b) Algoritmo (4.6).
c) Algoritmo (4.16). d) Algoritmo (4.22).

4.3.2 Experimentos utilizando modelo en la ley de control

Sólo se aplica a los algoritmos dados por (4.30), (4.36) y (4.39). El modelo utilizado en los experimentos se estrajo de Martínez (2002). El algoritmo adaptable solo fue considerado para la segunda trayectoria con objeto de permitir una mejor estimación de los parámetros del sistema. La Tabla 4.5 muestra los resultados para los algoritmos (4.30) y (4.36) con la primera trayectoria. Observando el índice de desempeño se ven mejores resultados para (4.30). Los correspondientes resultados se observan en las Figuras (4.16) a (4.21).

Para la segunda trayectoria se emplea también el algoritmo adaptable. Puesto que se observo una convergencia paramétrica muy lenta, se hicieron varias corridas utilizando en cada caso el ultimo valor obtenido de la adaptación anterior como condición inicial. La primera vez se comienza con valores nulos. Como podría haberse esperado, mientras más tiempo se dio para la adaptación, los resultados mejoraron notoriamente. Por ello en las Figuras (4.22) a (4.27) se presentan para el caso la primera y la última corrida. De la misma forma se observa en la Tabla 4.6 que la última corrida para (4.39) proporcione los mejores resultados.

Los resultados que se observan de las tablas comparativas son, que al colocar la dinámica del robot y de los actuadores, redujo el índice de desempeño. Contrario al caso sin modelo, los índices de desempeño de la trayectoria 2 son menores en un mínimo de 10 % comparados con los de la trayectoria 1. El controlador adaptable conforme paso el tiempo de experimentación se comporto mejor, debido a que convergían los parametros.

Articulación	Algoritmo	
	(-1.30)	(-1.36)
\mathcal{I}_1	0.287515	0.243093
\mathcal{I}_2	0.256657	0.203344
\mathcal{I}_3	0.780833	0.482752
\mathcal{I}_4	0.219742	0.324683
\mathcal{I}_5	0.186979	0.268746
\mathcal{I}_6	0.216534	0.272617
\mathcal{I}	0.942536	1.081820

Tabla 4.5: Tabla comparativa de índices de desempeño para la Trayectoria 1 sin modelo.

Articulación	Algoritmo			
	Con modelo		Adaptable	
	(-1.30)	(-1.36)	(4.39)	
			Exp. inicial	Exp. final
\mathcal{I}_1	0.782318	0.133339	0.223153	0.219944
\mathcal{I}_2	0.350786	0.228602	0.134033	0.116180
\mathcal{I}_3	0.491425	0.356733	0.202617	0.185101
\mathcal{I}_4	0.443651	0.439475	0.255505	0.226902
\mathcal{I}_5	0.096417	0.109428	0.249017	0.106906
\mathcal{I}_6	0.125405	0.152154	0.161839	0.142561
\mathcal{I}	0.782318	0.922561	0.724293	0.598953

Tabla 4.6: Tabla comparativa de índices de desempeño para la Trayectoria 2 sin modelo.

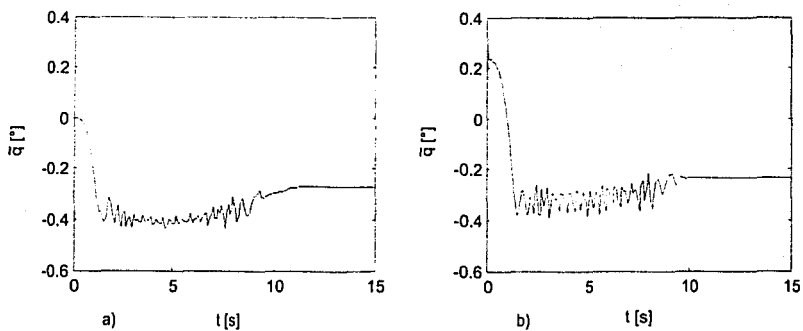


Figura 4.16: Errores de seguimiento Trayectoria 1, Articulación 1 con modelo.
 a) Algoritmo (4.30). b) Algoritmo (4.36).

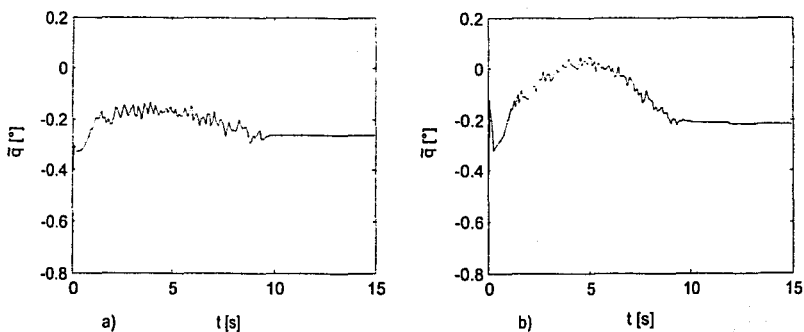


Figura 4.17: Errores de seguimiento Trayectoria 1, Articulación 2 con modelo.
 a) Algoritmo (4.30). b) Algoritmo (4.36).

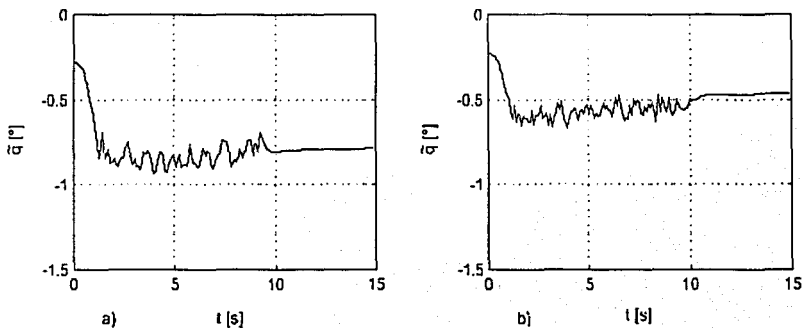
TESIS CON
FALLA DE ORIGEN

Figura 4.18: Errores de seguimiento Trayectoria 1, Articulación 3 con modelo.
a) Algoritmo (4.30). b) Algoritmo (4.36).

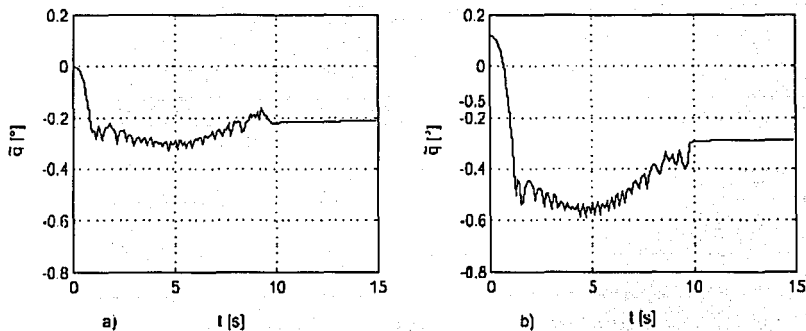


Figura 4.19: Errores de seguimiento Trayectoria 1, Articulación 4 con modelo.
a) Algoritmo (4.30). b) Algoritmo (4.36).

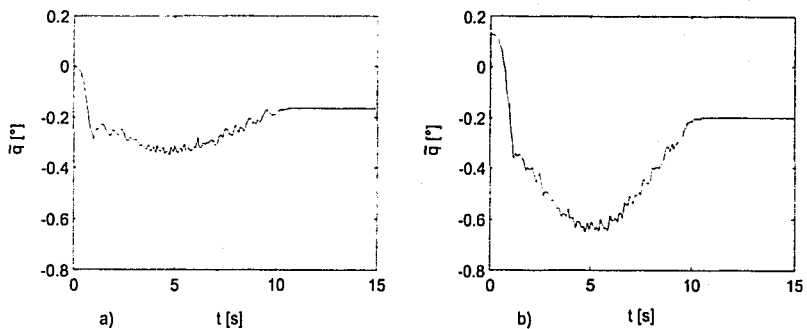


Figura 4.20: Errores de seguimiento Trayectoria 1, Articulación 5 con modelo.
 a) Algoritmo (4.30). b) Algoritmo (4.36).

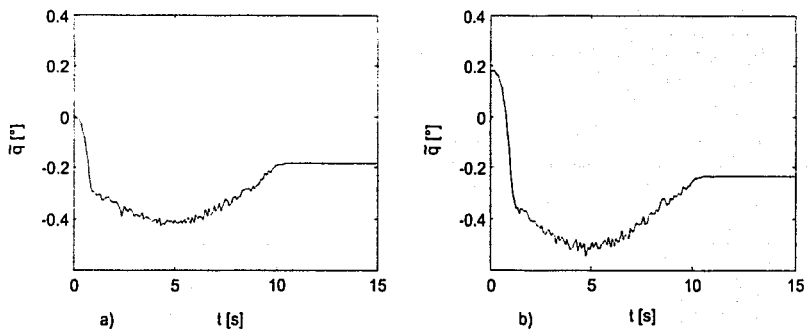


Figura 4.21: Errores de seguimiento Trayectoria 1, Articulación 6 con modelo.
 a) Algoritmo (4.30). b) Algoritmo (4.36).

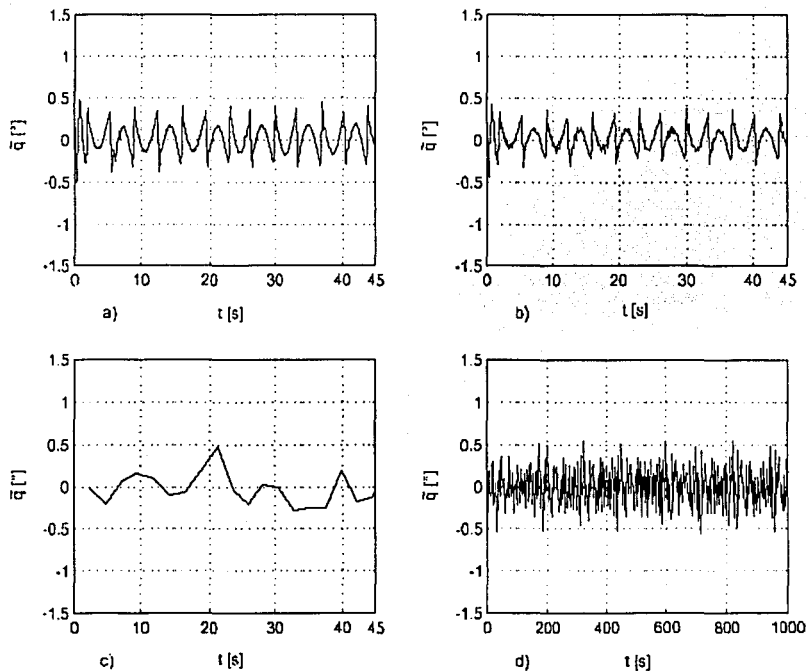


Figura 4.22: Errores de seguimiento Trayectoria 2, Articulación 1 con modelo.

a) Algoritmo (4.30).

b) Algoritmo (4.33).

c) Algoritmo (4.39) corrida inicial.

d) Algoritmo (4.39) corrida final.

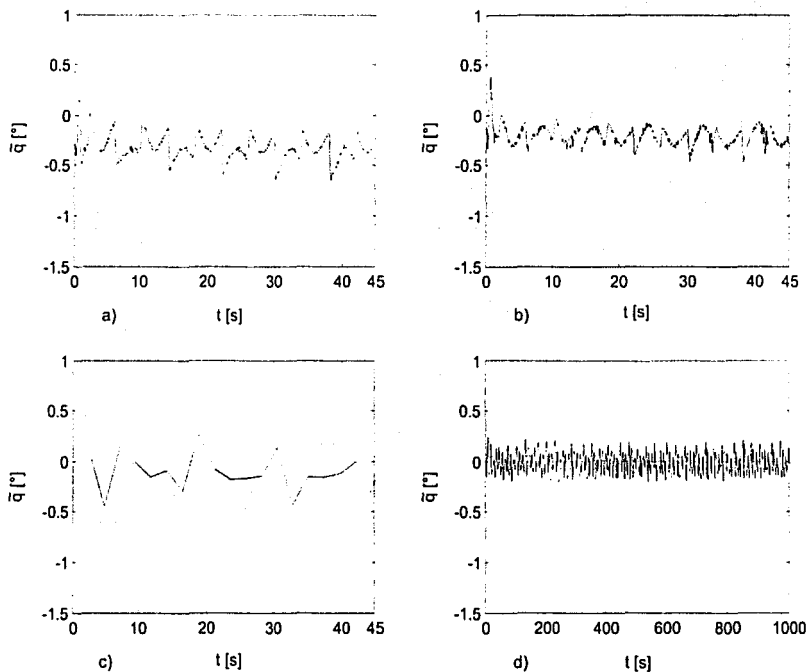


Figura 4.23: Errores de seguimiento Trayectoria 2, Articulación 2 con modelo.

a) Algoritmo (4.30).

b) Algoritmo (4.33).

c) Algoritmo (4.39) corrida inicial.

d) Algoritmo (4.39) corrida final.

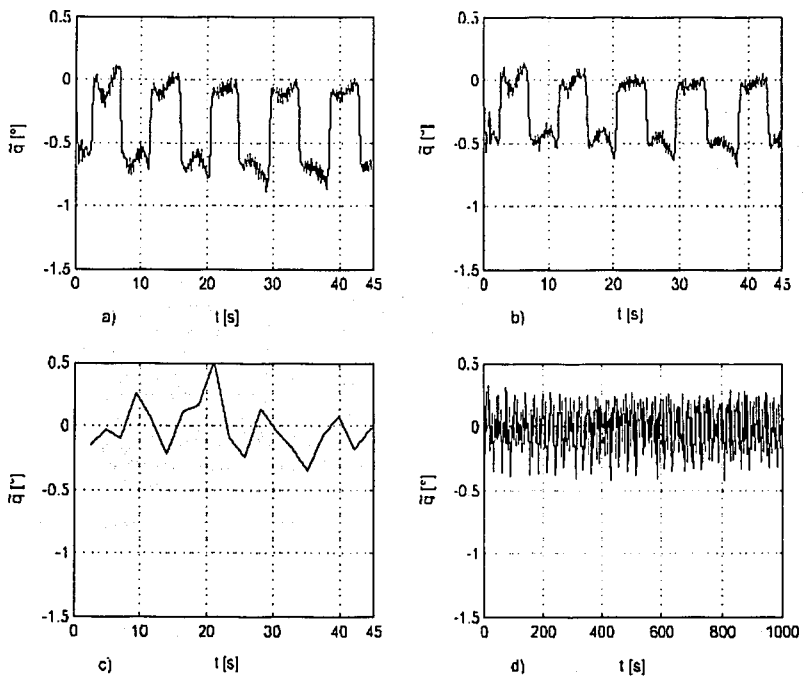
TESIS CON
FALLA DE ORIGEN

Figura 4.24: Errores de seguimiento Trayectoria 2, Articulación 3 con modelo.

a) Algoritmo (4.30).

b) Algoritmo (4.33).

c) Algoritmo (4.39) corrida inicial.

d) Algoritmo (4.39) corrida final.

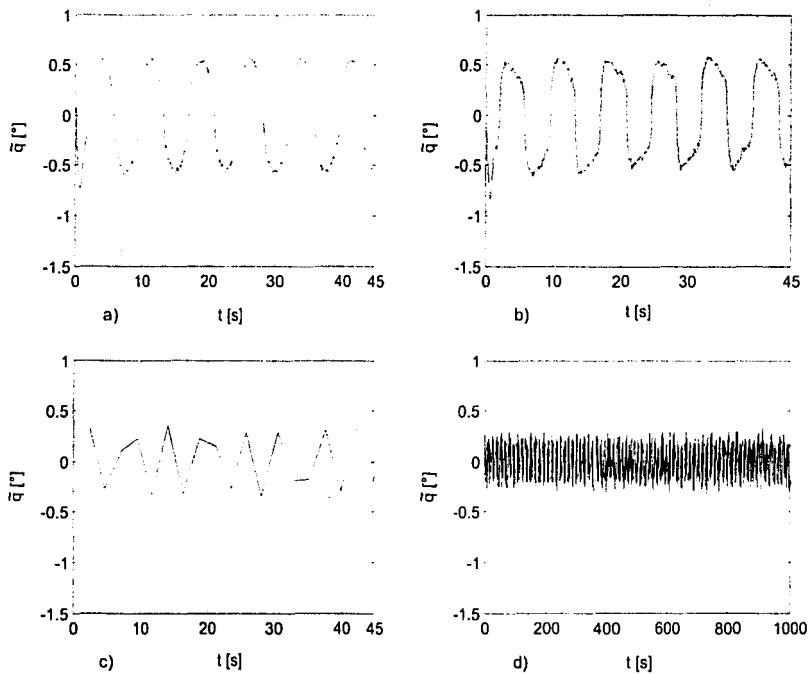


Figura 4.25: Errores de seguimiento Trayectoria 2, Articulación 4 con modelo.
a) Algoritmo (4.30). b) Algoritmo (4.33).
c) Algoritmo (4.39) corrida inicial. d) Algoritmo (4.39) corrida final.

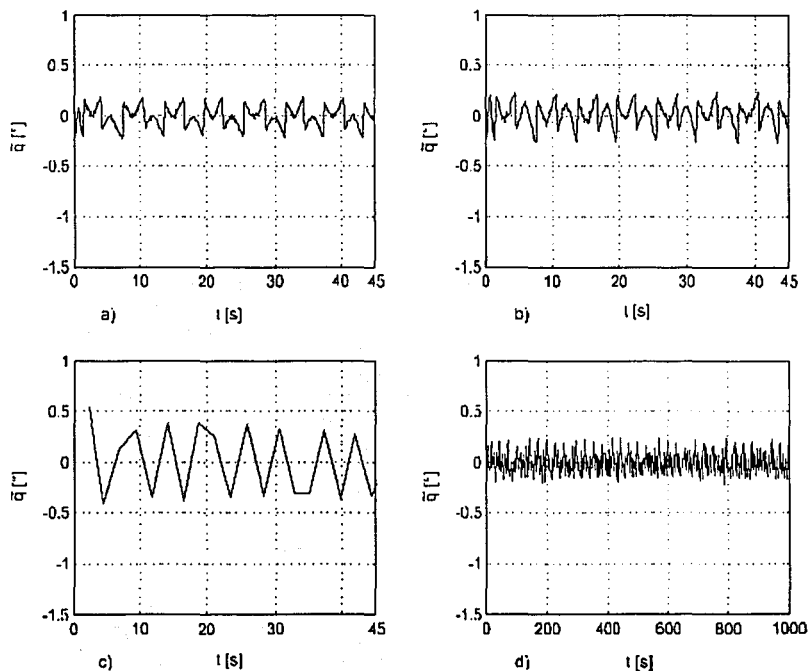


Figura 4.26: Errores de seguimiento Trayectoria 2, Articulación 5 con modelo.
a) Algoritmo (4.30).
b) Algoritmo (4.33).
c) Algoritmo (4.39) corrida inicial.
d) Algoritmo (4.39) corrida final.

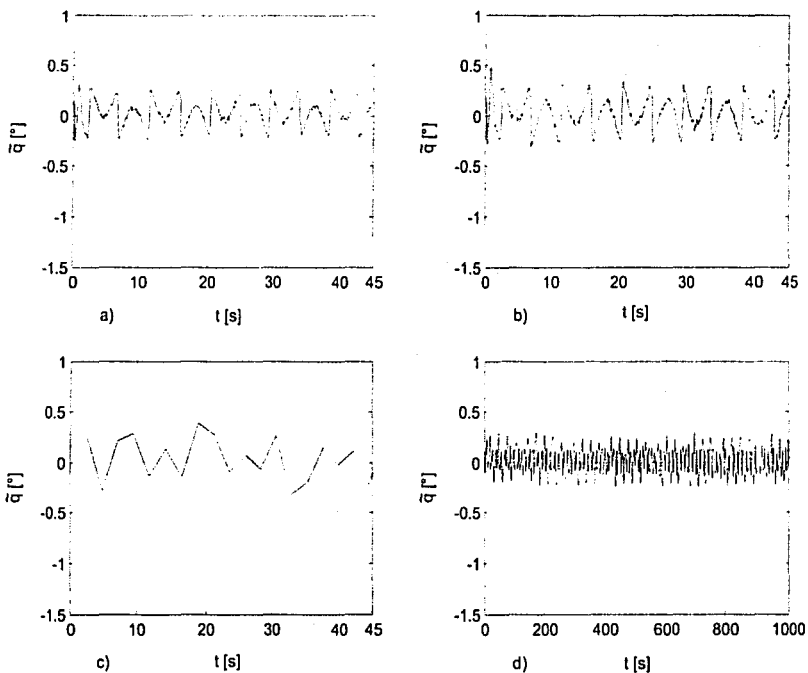


Figura 4.27: Errores de seguimiento Trayectoria 2, Articulación 6 con modelo.

a) Algoritmo (4.30).

b) Algoritmo (4.33).

c) Algoritmo (4.39) corrida inicial.

d) Algoritmo (4.39) corrida final.

Capítulo 5

Conclusiones

En esta tesis se han probado experimentalmente cinco algoritmos de control en un brazo robot industrial. El comportamiento de cuatro de estos controladores ha sido comparado con el controlador lineal PD, que es el más comúnmente utilizado. Para poder emitir una mejor conclusión de los algoritmos de control se definió y utilizó un índice de desempeño.

Los algoritmos de control son algoritmos conocidos en la literatura. Uno de ellos no requiere el modelo del robot, pues se basa en altas ganancias. Otros permiten trabajar con o sin el conocimiento del modelo, mientras que el algoritmo de control adaptable trabaja siempre con base en la estructura del modelo. Por ello se decidió dividir los experimentos en dos grupos: aquellos donde se incluye explícitamente el modelo del robot y aquellos donde no se incluye. Asimismo dos tipos de trayectorias, polinomios de quinto orden y acñales senoidales fueron programadas como referencias.

Para la primera trayectoria sin modelo dinámico del robot, si se considera sólo el índice de desempeño total, el algoritmo que presentó el mejor desempeño fue el dado por (4.16) cuyo índice de desempeño fue menor en 11%, seguido por (4.6), (4.22), comparados con (4.4). En lo que respecta al caso con modelo, el controlador que presentó el mejor desempeño fue el dado por (4.30).

Por otro lado, para la segunda trayectoria, en el caso de los algoritmos de control sin modelo, el mejor fue el dado por (4.16) cuyo índice de desempeño fue menor en 4%, seguido por (4.22), (4.6) comparados con (4.4).

Para el caso con modelo el de mejor desempeño fue el dado por (4.39). Cabe hacer notar que el algoritmo adaptable requiere de condiciones iniciales. De manera iterativa se comenzó con cero y luego sucesivamente se sustituyó el último valor después de 1200 segundos, con lo que se observó repetidamente una mejora en el resultado.

Si se comparan los índices de desempeño del caso sin modelo con el caso con modelo, se observan mejoras, las cuales oscilan entre un 17 y 14 % para la trayectoria 1 y para el caso de la trayectoria las mejoras oscilan entre el 36 y 58 %.

De todas estas observaciones se puede concluir que:

- Teóricamente el error de los algoritmos tenía que tender a una vecindad cerca de cero. Para el caso sin modelo este tuvo una buena tendencia a cero, ya que el índice de desempeño tuvo un máximo de 1.4 grados, algunos de los factores que influyeron a que este no fuera mejor son que en las leyes de control no se contemplaron gravedad y fricción. Para el caso con modelo el error se acercó más a cero ya que estos dos factores ya se contemplaron. Lo cual proporcione un índice de desempeño de 1 grado de error. También puede afectar el que la sintonización de ganancias no haya sido la más precisa. Pero en lo general se cumplieron los pronósticos de que el error tendería a una vecindad cercana de cero.
 - Para las trayectorias 1, los índices de desempeño fueron menores comparados con los de las trayectorias 2, considerando el caso sin modelo, esto se debió a que la velocidad de desplazamiento era suave. Y la diferencia entre cada uno pudo deberse a la respuesta de cada controlador.
 - En el caso de trayectorias donde los cambios de dirección son constantes y periódicos, el desempeño de los controladores presentó una mejora cuantificable utilizando el modelo del robot en la ley de control y a que los algoritmos de control están diseñados para el seguimiento de trayectorias.
 - Incluir el modelo dinámico del robot y la dinámica de los motores en la ley de control permite tener un mejor desempeño del robot en sus tareas.
-

- La implementación de algoritmos de control del tipo adaptable presenta una gran ventaja, ya que al calcular el valor de los parámetros del modelo del robot en línea permite tener un mejor desempeño del sistema.
- Con base en la experimentación, se puede concluir que factores que influyen negativamente son:
 - Las tarjetas de adquisición de datos.
 - El tiempo de muestreo.
 - La carga computacional.

Esto se observó, ya que al principio de los experimentos se contaba con una computadora con procesador Pentium III a 735 MHz, en este momento el tiempo de muestreo que se tenía era de entre los 10 y 12 ms, posteriormente se cambió de equipo, pasando a utilizar un procesador Pentium IV a 1.5GHz con lo cual el tiempo de muestreo se redujo a 8 ms. Pensando en que el tiempo de muestreo se reduciría más, se procedió a monitorear el tiempo de procesamiento de la tarjeta de adquisición de datos en la lectura de los encoders y la carga en los daqs. Lo que se observó, es que el tiempo de lectura de los 6 encoders era de 6 ms. La carga computacional que se dio fue debido a los controladores en los que se incluyó la dinámica del robot, dado que la cantidad de operaciones matemáticas fue mayor.

Apéndice A

TESIS CON
FALLA DE ORIGEN

Descripción de la interfaz PC-Robot

La simulación de algoritmos de control para robots por medio de una computadora ha tenido mucho impulso y ha desplazado la comprobación experimental física. Una de las desventajas de la simulación por medio de la computadora radica en no asegurar la confiabilidad de los resultados que la teoría pueda validar. Es decir, fenómenos tales como perturbaciones, incertidumbres paramétricas o dinámicas no modeladas son difíciles de incluir de manera apropiada en la simulación (Gudiño Lau y Arteaga Pérez 2000).

Crear laboratorios experimentales de robótica permite implementar algoritmos de control propios y probar el desempeño en condiciones reales de operación. Sin embargo esta opción tiene la desventaja de elevar costos considerablemente.

Por otro lado, la mayoría de las compañías que venden robots suelen entregar manipuladores en los que no es posible separar la etapa de potencia, protección y control. Esta sección se describe brevemente la conversión de un robot industrial a un robot experimental mediante la implementación de una interfaz entre las etapas de potencia, protección, motores y encoders, con la finalidad de que se permita la interacción PC-robot mediante programación estructurada, para obtener información más detallada consultar Castillo Sánchez (2002).

A.1 Partes que componen del equipo

A fin de conseguir el diseño de una interfaz PC-Robot, fué necesario conocer las características y componentes de cada uno de los sistemas involucrados. Sin profundidad de conocimiento se hace el desglose de cada sistema como sigue, el lector interesado puede consultar en Castillo Sánchez (2002):

Robot manipulador CRS Robotics A465. Fue diseñado con el mismo campo de acción de movimiento y levantamiento de carga promedio de un brazo humano. El Robot CRS A465 es una combinación de potencia y alta velocidad; los motores de c.d. que utiliza este robot poseen una constante mecánica y eléctrica que los hace tener buen desempeño y alto torque de respuesta (CRS 1997). Sus características son:

- Robot Articulado de seis grados de libertad.
- Servomotores de c.d.
- Encoders ópticos.
- Transmisión de engranes armónica.
- 31 Kg de peso.
- Salidas de señales de encoder.

Modulo de procesamiento, Protección y Amplificación C500 CRS.

Permite la salida de señales de encoder y la entrada de señales de control a la etapa amplificadora para su posterior salida al robot A465.

- Módulo multitarea C500C diseñado bajo el concepto de arquitectura descentralizada ó abierta.
 - Tarjeta de procesamiento PC-104/i486.
 - Lenguaje de Programación y aplicaciones RAPL-3.
 - Plataforma de control con esquema PID.
 - Circuitos de paro y emergencia.
-

- Detector continuo de fallas.
- 16 entradas digitales aisladas ópticamente.
- 12 salidas digitales aisladas ópticamente.
- Dos puertos seriales.
- Seis unidades de amplificación.

TESIS CON
FALLA DE ORIGEN

Accesorio de interfaz de movimiento universal (UMI-Flex6). Este accesorio de National Instruments permite la conexión de señales de encoders del robot a una tarjeta de adquisición de datos Flex Motion. También permite la conexión de otra tarjeta Flex Motion con señales de control de motores al módulo C500 para su amplificación correspondiente (Instruments 1999). Con su etapa de protección el módulo C500 evita daños en motores y en tarjetas de adquisición de datos.

- 6 unidades de conexión de encoders, con seis terminales de conexión cada unidad ($A, \bar{A}, B, \bar{B}, I, \bar{I}$).
- 6 unidades de conexión de amplificador/motores.
- Conexión de switch limite (límite hacia delante, límite de reversa, entrada de home).
- Conexiones de energía.
- Unidad con ocho entradas ó canales analógicos.
- Cable especial de entrada(100 pines) a tarjeta de adquisición de datos.

Tarjeta de adquisición de datos PCI Flex motion 6C. Tarjeta de National Instruments encargada de procesar señales digitales de encoders de robot hacia la PC y permite la salida de comandos de voltaje de motores en forma analógica (DAC's).

- Procesador DSP de tiempo real MC68331 a 32 bits.
 - Control de servomotores.
 - Seis Entradas de encoders de alta resolución ($A, \bar{A}, B, \bar{B}, I, \bar{I}$).
-

- Ocho canales analógicos de entrada.
- Seis salidas analógicas de ± 10 volts.
- Conexión de switch limite (límite hacia delante, límite de reversa, entrada de home).

Computadora Personal. Con programación estructurada procesa algoritmos de control, teniendo como entrada señales de posición de articulación y como salida señales de control de actuadores de c.d.

- Procesador Pentium IV a 1.5 Ghz.
- Lenguaje de programación LabWindows/CVI basado en ANSI "C" de NI con características graficas interactivas.

A.2 Interfaz

El diseño de esta interfaz tiene tres objetivos.

- Eliminar la etapa de procesamiento de control en el módulo original de fábrica mediante la interrupción en las señales de los sensores de posición. Utilizar la etapa de amplificación para las señales de comandos de voltaje de motores.
- Permitir que las señales de los sensores de posición del Robot CRS A465 se conecten desde una tarjeta UMI (Interfaz de movimiento univeral), a una computadora que tiene incorporada una tarjeta de adquisición de datos de 32 bits a través de un cable especial. Hay 36 canales (conexiones) de lectura de encoders con información digital del desplazamiento angular.
- Por medio de una unidad UMI conectar las seis señales analógicas de control de cada motor al módulo amplificador de potencia. Estas señales de control son previamente procesadas en la PC y posteriormente enviadas a la unidad DAC de la tarjeta de adquisición de datos a través de un cable especial.

El siguiente diagrama simplifica el diseño de la interfaz PC-Robot

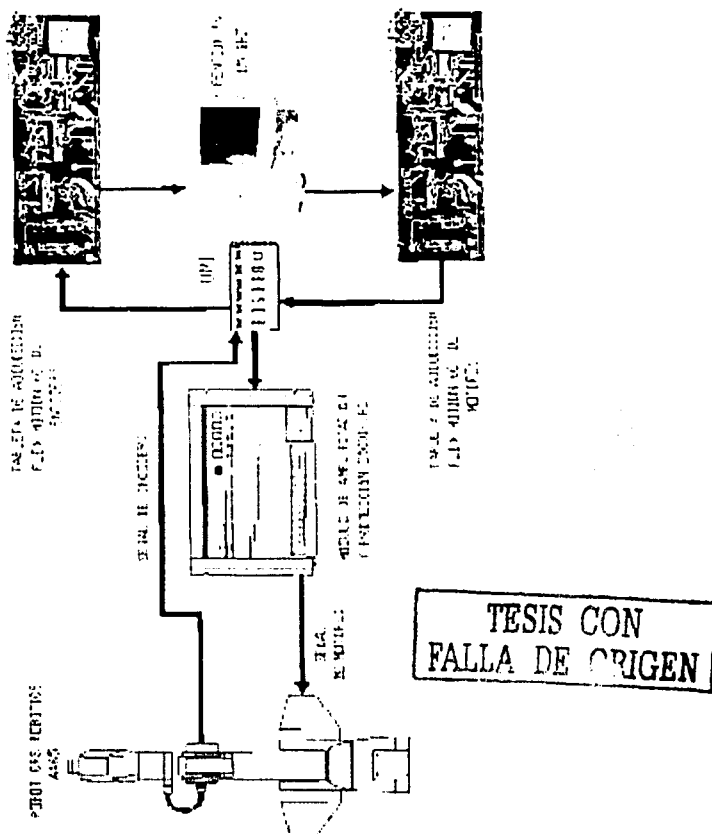


Figura A.1: Diseño de interfaz para control de Robot A465.

TESIS CON
FALLA DE ORIGEN

Apéndice B

Programa de control, adaptación y observador de velocidad del robot A465

```
// 19 de Diciembre del 2001    19 00 Hrs
// Integración por el Método Trapezoidal
// Programa en "C" de control adaptable para identificación de parámetros

#include <Analysis.h>
#include <Cvirts.h>
#include <Userint.h>
#include "Sincro A465.h"
#include "c \FlexMotion\include\Flexmtn.h"
#include <ansi_c.h>
#include <utility.h>

static int pan#Handl#;

#define HOST OFF // Return Variable to Host Computer (not FlexMotion program variable)
#define muestras 150000 // No. total de muestras de todo el sistema. Muestras = tiempo(s)/Tiempo muestreo
#define mues 600 // Sólo las muestras que se despligan en pantalla
#define almacena 300 // Var. q' indica el no. de instantes de muestreo Tiempo de almacen = 0.1(seg)/Tiempo Muestreo

u8 axisNum1 = 1; // Número de eje a mover (1)
u8 axisNum2 = 2; // Número de eje a mover (2)
u8 axisNum3 = 3; // Número de eje a mover (3)
u8 axisNum4 = 4; // Número de eje a mover (4)
u8 axisNum5 = 5; // Número de eje a mover (5)
u8 axisNum6 = 6; // Número de eje a mover (6)

i32 currentPosition; // Monitored current position of the axis

i32 qq1;
i32 qq2;
i32 qq3; // Variables de lectura de encoders
i32 qq4;
i32 qq5;
i32 qq6;

i32 status; // Verifica error
u8 boardID1 = 2; // Número de Tarjeta
u8 boardID2 = 1; // Número de Tarjeta
u8 programNum = 1;

int i+1.menos.asp;
double q1,q2,q3,q4,q5,q6;
double tiempo1,tiempof,tiempoa,Sincro;
```



```

double ka1 = 0.1876896, ka2=0.0563, ka3=0.0392; // Constante motor-par Nm/amp
double ke1 = 0.84, ka2 = 2.7, Ra3 = 0.9; // Resistencia de Armadura ohms
double kb1 = 0.142284519, kb2 = 0.0534760609, kb3 = 0.039152; // Constante contraelectromotriz Volts/rpm
double ja1=0.00119168, ja2=0.0000049, ja3=0.000001568; // Inercia del Motor
double iw=0.000128021689744; // Cnief. de amortiguamiento de fric. viscosa Nm/rpm
double ri=100, r2=100, r3=100, r4=101, r5=100, r6=101; // Radios de engranaje

double k1, k2, k3, k4, k5, k6, k7, k8, k9, k10, k11, k12, fw1, fw2, fw3, fw4, fw5, fw6, hel, ke2, ke3, ke4, ke5, ke6;
double qt(6)(1)-(0), qtp(6)(1)-(0), lam1, lam2, lam3, lam4, lam5, lam6;

double gankd(6)(0)-(0);
double signq1, signq2, signq3, signq4, signq5, signq6;

// Variable de la Modificacion
double kp1, kp2, kp3, kp4, kp5, kp6, modif(6)(0)-(0), msp(6)(0)-(0), ley3(6)(1)-(0), fint, tms;

int suma, fin, chance = 0, cont0, cont1, cont2, cont3, cont4, cont5;

double inter0, inter1, inter2, inter3, inter4, inter5;

double ag11, ag21, ag31, ag41, ag51, ag61, ag71, ag81, ag91, ag101, ag111;
double ag121, ag131, ag141, ag151, ag161, ag171, ag181, ag191, ag201, ag211, ag221;
double ag231, ag241, ag251, ag261, ag271, ag281, ag291, ag301, ag311, ag321, ag331;

double ag11.1, ag21.1, ag31.1, ag41.1, ag51.1, ag61.1, ag71.1, ag81.1, ag91.1, ag101.1, ag111.1;
double ag121.1, ag131.1, ag141.1, ag151.1, ag161.1, ag171.1, ag181.1, ag191.1, ag201.1, ag211.1, ag221.1;
double ag231.1, ag241.1, ag251.1, ag261.1, ag271.1, ag281.1, ag291.1, ag301.1, ag311.1, ag321.1, ag331.1;

double agp11, agp21, agp31, agp41, agp51, agp61, agp71, agp81, agp91, agp101, agp111;
double agp121, agp131, agp141, agp151, agp161, agp171, agp181, agp191, agp201, agp211, agp221;
double agp231, agp241, agp251, agp261, agp271, agp281, agp291, agp301, agp311, agp321, agp331;

double agm11.1, agp21.1, agp31.1, agp41.1, agp51.1, agp61.1, agp71.1, agp81.1, agp91.1, agp101.1, agp111.1;
double agp121.1, agp131.1, agp141.1, agp151.1, agp161.1, agp171.1, agp181.1, agp191.1, agp201.1, agp211.1, agp221.1;
double agp231.1, agp241.1, agp251.1, agp261.1, agp271.1, agp281.1, agp291.1, agp301.1, agp311.1, agp321.1, agp331.1;

double volt1[muestras]-(0), volt2[muestras]-(0), volt3[muestras]-(0);
double volt4[muestras]-(0), volt5[muestras]-(0), volt6[muestras]-(0);

double gam0, gam1, gam2, gam3, gam4, gam5, gam6, gam7, gam8, gam9, gam10, gam11, gam12, gam13, gam14, gam15, gam16, gam17;
double gam16, gam19, gam20, gam21, gam23, gam24, gam25, gam26, gam27, gam28, gam29, gam30, gam31, gam32, gam33, hd2;

double qp1g[muestras]-(0), qp2g[muestras]-(0), qp3g[muestras]-(0), qp4g[muestras]-(0);
qp5g[muestras]-(0), qp6g[muestras]-(0);

// Variable s de la estimador de velocidad
double ppi, pp2, pp3, pp4, pp5, pp6, kk=50.0, p1, p2, p3, p4, p5, p6, wf1, wf2, wf3, wf4, wf5, wf6, p1.1, p2.1, p3.1, p4.1, p5.1, p6.1;

// Variables para el Observador

double z1, z2, z3, z4, z5, z6, qh1, qh2, qh3, qh4, qh5, qh6, qhp1, qhp2, qhp3, qhp4, qhp5, qhp6, kr1, kr2, kr3, kr4, kr5, kr6, kd=2; //kd*43.0
double qhp1, qhp2, qhp3, qhp4, qhp5, qhp6, qhp7, qhp8, qhp9, qhp10, qhp11, qhp12, qhp13, qhp14, qhp15, qhp16, qhp17, qhp18, qhp19, qhp20, qhp21, qhp22, qhp23, qhp24, qhp25, qhp26, qhp27, qhp28, qhp29, qhp30, qhp31, qhp32, qhp33, qhp34, qhp35, qhp36, qhp37, qhp38, qhp39, qhp40, qhp41, qhp42, qhp43, qhp44, qhp45, qhp46, qhp47, qhp48, qhp49, qhp50, qhp51, qhp52, qhp53, qhp54, qhp55, qhp56, qhp57, qhp58, qhp59, qhp60, qhp61, qhp62, qhp63, qhp64, qhp65, qhp66, qhp67, qhp68, qhp69, qhp70, qhp71, qhp72, qhp73, qhp74, qhp75, qhp76, qhp77, qhp78, qhp79, qhp80, qhp81, qhp82, qhp83, qhp84, qhp85, qhp86, qhp87, qhp88, qhp89, qhp90, qhp91, qhp92, qhp93, qhp94, qhp95, qhp96, qhp97, qhp98, qhp99, qhp100, qhp101, qhp102, qhp103, qhp104, qhp105, qhp106, qhp107, qhp108, qhp109, qhp110, qhp111, qhp112, qhp113, qhp114, qhp115, qhp116, qhp117, qhp118, qhp119, qhp120, qhp121, qhp122, qhp123, qhp124, qhp125, qhp126, qhp127, qhp128, qhp129, qhp130, qhp131, qhp132, qhp133, qhp134, qhp135, qhp136, qhp137, qhp138, qhp139, qhp140, qhp141, qhp142, qhp143, qhp144, qhp145, qhp146, qhp147, qhp148, qhp149, qhp150, qhp151, qhp152, qhp153, qhp154, qhp155, qhp156, qhp157, qhp158, qhp159, qhp160, qhp161, qhp162, qhp163, qhp164, qhp165, qhp166, qhp167, qhp168, qhp169, qhp170, qhp171, qhp172, qhp173, qhp174, qhp175, qhp176, qhp177, qhp178, qhp179, qhp180, qhp181, qhp182, qhp183, qhp184, qhp185, qhp186, qhp187, qhp188, qhp189, qhp190, qhp191, qhp192, qhp193, qhp194, qhp195, qhp196, qhp197, qhp198, qhp199, qhp200, qhp201, qhp202, qhp203, qhp204, qhp205, qhp206, qhp207, qhp208, qhp209, qhp210, qhp211, qhp212, qhp213, qhp214, qhp215, qhp216, qhp217, qhp218, qhp219, qhp220, qhp221, qhp222, qhp223, qhp224, qhp225, qhp226, qhp227, qhp228, qhp229, qhp230, qhp231, qhp232, qhp233, qhp234, qhp235, qhp236, qhp237, qhp238, qhp239, qhp240, qhp241, qhp242, qhp243, qhp244, qhp245, qhp246, qhp247, qhp248, qhp249, qhp250, qhp251, qhp252, qhp253, qhp254, qhp255, qhp256, qhp257, qhp258, qhp259, qhp260, qhp261, qhp262, qhp263, qhp264, qhp265, qhp266, qhp267, qhp268, qhp269, qhp270, qhp271, qhp272, qhp273, qhp274, qhp275, qhp276, qhp277, qhp278, qhp279, qhp280, qhp281, qhp282, qhp283, qhp284, qhp285, qhp286, qhp287, qhp288, qhp289, qhp290, qhp291, qhp292, qhp293, qhp294, qhp295, qhp296, qhp297, qhp298, qhp299, qhp300, qhp301, qhp302, qhp303, qhp304, qhp305, qhp306, qhp307, qhp308, qhp309, qhp310, qhp311, qhp312, qhp313, qhp314, qhp315, qhp316, qhp317, qhp318, qhp319, qhp320, qhp321, qhp322, qhp323, qhp324, qhp325, qhp326, qhp327, qhp328, qhp329, qhp330, qhp331, qhp332, qhp333, qhp334, qhp335, qhp336, qhp337, qhp338, qhp339, qhp340, qhp341, qhp342, qhp343, qhp344, qhp345, qhp346, qhp347, qhp348, qhp349, qhp350, qhp351, qhp352, qhp353, qhp354, qhp355, qhp356, qhp357, qhp358, qhp359, qhp360, qhp361, qhp362, qhp363, qhp364, qhp365, qhp366, qhp367, qhp368, qhp369, qhp370, qhp371, qhp372, qhp373, qhp374, qhp375, qhp376, qhp377, qhp378, qhp379, qhp380, qhp381, qhp382, qhp383, qhp384, qhp385, qhp386, qhp387, qhp388, qhp389, qhp390, qhp391, qhp392, qhp393, qhp394, qhp395, qhp396, qhp397, qhp398, qhp399, qhp400, qhp401, qhp402, qhp403, qhp404, qhp405, qhp406, qhp407, qhp408, qhp409, qhp410, qhp411, qhp412, qhp413, qhp414, qhp415, qhp416, qhp417, qhp418, qhp419, qhp420, qhp421, qhp422, qhp423, qhp424, qhp425, qhp426, qhp427, qhp428, qhp429, qhp430, qhp431, qhp432, qhp433, qhp434, qhp435, qhp436, qhp437, qhp438, qhp439, qhp440, qhp441, qhp442, qhp443, qhp444, qhp445, qhp446, qhp447, qhp448, qhp449, qhp450, qhp451, qhp452, qhp453, qhp454, qhp455, qhp456, qhp457, qhp458, qhp459, qhp460, qhp461, qhp462, qhp463, qhp464, qhp465, qhp466, qhp467, qhp468, qhp469, qhp470, qhp471, qhp472, qhp473, qhp474, qhp475, qhp476, qhp477, qhp478, qhp479, qhp480, qhp481, qhp482, qhp483, qhp484, qhp485, qhp486, qhp487, qhp488, qhp489, qhp490, qhp491, qhp492, qhp493, qhp494, qhp495, qhp496, qhp497, qhp498, qhp499, qhp500, qhp501, qhp502, qhp503, qhp504, qhp505, qhp506, qhp507, qhp508, qhp509, qhp510, qhp511, qhp512, qhp513, qhp514, qhp515, qhp516, qhp517, qhp518, qhp519, qhp520, qhp521, qhp522, qhp523, qhp524, qhp525, qhp526, qhp527, qhp528, qhp529, qhp530, qhp531, qhp532, qhp533, qhp534, qhp535, qhp536, qhp537, qhp538, qhp539, qhp540, qhp541, qhp542, qhp543, qhp544, qhp545, qhp546, qhp547, qhp548, qhp549, qhp550, qhp551, qhp552, qhp553, qhp554, qhp555, qhp556, qhp557, qhp558, qhp559, qhp560, qhp561, qhp562, qhp563, qhp564, qhp565, qhp566, qhp567, qhp568, qhp569, qhp570, qhp571, qhp572, qhp573, qhp574, qhp575, qhp576, qhp577, qhp578, qhp579, qhp580, qhp581, qhp582, qhp583, qhp584, qhp585, qhp586, qhp587, qhp588, qhp589, qhp590, qhp591, qhp592, qhp593, qhp594, qhp595, qhp596, qhp597, qhp598, qhp599, qhp600, qhp601, qhp602, qhp603, qhp604, qhp605, qhp606, qhp607, qhp608, qhp609, qhp610, qhp611, qhp612, qhp613, qhp614, qhp615, qhp616, qhp617, qhp618, qhp619, qhp620, qhp621, qhp622, qhp623, qhp624, qhp625, qhp626, qhp627, qhp628, qhp629, qhp630, qhp631, qhp632, qhp633, qhp634, qhp635, qhp636, qhp637, qhp638, qhp639, qhp640, qhp641, qhp642, qhp643, qhp644, qhp645, qhp646, qhp647, qhp648, qhp649, qhp650, qhp651, qhp652, qhp653, qhp654, qhp655, qhp656, qhp657, qhp658, qhp659, qhp660, qhp661, qhp662, qhp663, qhp664, qhp665, qhp666, qhp667, qhp668, qhp669, qhp670, qhp671, qhp672, qhp673, qhp674, qhp675, qhp676, qhp677, qhp678, qhp679, qhp680, qhp681, qhp682, qhp683, qhp684, qhp685, qhp686, qhp687, qhp688, qhp689, qhp690, qhp691, qhp692, qhp693, qhp694, qhp695, qhp696, qhp697, qhp698, qhp699, qhp700, qhp701, qhp702, qhp703, qhp704, qhp705, qhp706, qhp707, qhp708, qhp709, qhp710, qhp711, qhp712, qhp713, qhp714, qhp715, qhp716, qhp717, qhp718, qhp719, qhp720, qhp721, qhp722, qhp723, qhp724, qhp725, qhp726, qhp727, qhp728, qhp729, qhp730, qhp731, qhp732, qhp733, qhp734, qhp735, qhp736, qhp737, qhp738, qhp739, qhp740, qhp741, qhp742, qhp743, qhp744, qhp745, qhp746, qhp747, qhp748, qhp749, qhp750, qhp751, qhp752, qhp753, qhp754, qhp755, qhp756, qhp757, qhp758, qhp759, qhp760, qhp761, qhp762, qhp763, qhp764, qhp765, qhp766, qhp767, qhp768, qhp769, qhp770, qhp771, qhp772, qhp773, qhp774, qhp775, qhp776, qhp777, qhp778, qhp779, qhp780, qhp781, qhp782, qhp783, qhp784, qhp785, qhp786, qhp787, qhp788, qhp789, qhp790, qhp791, qhp792, qhp793, qhp794, qhp795, qhp796, qhp797, qhp798, qhp799, qhp800, qhp801, qhp802, qhp803, qhp804, qhp805, qhp806, qhp807, qhp808, qhp809, qhp810, qhp811, qhp812, qhp813, qhp814, qhp815, qhp816, qhp817, qhp818, qhp819, qhp820, qhp821, qhp822, qhp823, qhp824, qhp825, qhp826, qhp827, qhp828, qhp829, qhp830, qhp831, qhp832, qhp833, qhp834, qhp835, qhp836, qhp837, qhp838, qhp839, qhp840, qhp841, qhp842, qhp843, qhp844, qhp845, qhp846, qhp847, qhp848, qhp849, qhp850, qhp851, qhp852, qhp853, qhp854, qhp855, qhp856, qhp857, qhp858, qhp859, qhp860, qhp861, qhp862, qhp863, qhp864, qhp865, qhp866, qhp867, qhp868, qhp869, qhp870, qhp871, qhp872, qhp873, qhp874, qhp875, qhp876, qhp877, qhp878, qhp879, qhp880, qhp881, qhp882, qhp883, qhp884, qhp885, qhp886, qhp887, qhp888, qhp889, qhp890, qhp891, qhp892, qhp893, qhp894, qhp895, qhp896, qhp897, qhp898, qhp899, qhp900, qhp901, qhp902, qhp903, qhp904, qhp905, qhp906, qhp907, qhp908, qhp909, qhp910, qhp911, qhp912, qhp913, qhp914, qhp915, qhp916, qhp917, qhp918, qhp919, qhp920, qhp921, qhp922, qhp923, qhp924, qhp925, qhp926, qhp927, qhp928, qhp929, qhp930, qhp931, qhp932, qhp933, qhp934, qhp935, qhp936, qhp937, qhp938, qhp939, qhp940, qhp941, qhp942, qhp943, qhp944, qhp945, qhp946, qhp947, qhp948, qhp949, qhp950, qhp951, qhp952, qhp953, qhp954, qhp955, qhp956, qhp957, qhp958, qhp959, qhp960, qhp961, qhp962, qhp963, qhp964, qhp965, qhp966, qhp967, qhp968, qhp969, qhp970, qhp971, qhp972, qhp973, qhp974, qhp975, qhp976, qhp977, qhp978, qhp979, qhp980, qhp981, qhp982, qhp983, qhp984, qhp985, qhp986, qhp987, qhp988, qhp989, qhp990, qhp991, qhp992, qhp993, qhp994, qhp995, qhp996, qhp997, qhp998, qhp999, qhp1000, qhp1001, qhp1002, qhp1003, qhp1004, qhp1005, qhp1006, qhp1007, qhp1008, qhp1009, qhp1010, qhp1011, qhp1012, qhp1013, qhp1014, qhp1015, qhp1016, qhp1017, qhp1018, qhp1019, qhp1020, qhp1021, qhp1022, qhp1023, qhp1024, qhp1025, qhp1026, qhp1027, qhp1028, qhp1029, qhp1030, qhp1031, qhp1032, qhp1033, qhp1034, qhp1035, qhp1036, qhp1037, qhp1038, qhp1039, qhp1040, qhp1041, qhp1042, qhp1043, qhp1044, qhp1045, qhp1046, qhp1047, qhp1048, qhp1049, qhp1050, qhp1051, qhp1052, qhp1053, qhp1054, qhp1055, qhp1056, qhp1057, qhp1058, qhp1059, qhp1060, qhp1061, qhp1062, qhp1063, qhp1064, qhp1065, qhp1066, qhp1067, qhp1068, qhp1069, qhp1070, qhp1071, qhp1072, qhp1073, qhp1074, qhp1075, qhp1076, qhp1077, qhp1078, qhp1079, qhp1080, qhp1081, qhp1082, qhp1083, qhp1084, qhp1085, qhp1086, qhp1087, qhp1088, qhp1089, qhp1090, qhp1091, qhp1092, qhp1093, qhp1094, qhp1095, qhp1096, qhp1097, qhp1098, qhp1099, qhp1100, qhp1101, qhp1102, qhp1103, qhp1104, qhp1105, qhp1106, qhp1107, qhp1108, qhp1109, qhp1110, qhp1111, qhp1112, qhp1113, qhp1114, qhp1115, qhp1116, qhp1117, qhp1118, qhp1119, qhp1120, qhp1121, qhp1122, qhp1123, qhp1124, qhp1125, qhp1126, qhp1127, qhp1128, qhp1129, qhp1130, qhp1131, qhp1132, qhp1133, qhp1134, qhp1135, qhp1136, qhp1137, qhp1138, qhp1139, qhp1140, qhp1141, qhp1142, qhp1143, qhp1144, qhp1145, qhp1146, qhp1147, qhp1148, qhp1149, qhp1150, qhp1151, qhp1152, qhp1153, qhp1154, qhp1155, qhp1156, qhp1157, qhp1158, qhp1159, qhp1160, qhp1161, qhp1162, qhp1163, qhp1164, qhp1165, qhp1166, qhp1167, qhp1168, qhp1169, qhp1170, qhp1171, qhp1172, qhp1173, qhp1174, qhp1175, qhp1176, qhp1177, qhp1178, qhp1179, qhp1180, qhp1181, qhp1182, qhp1183, qhp1184, qhp1185, qhp1186, qhp1187, qhp1188, qhp1189, qhp1190, qhp1191, qhp1192, qhp1193, qhp1194, qhp1195, qhp1196, qhp1197, qhp1198, qhp1199, qhp1200, qhp1201, qhp1202, qhp1203, qhp1204, qhp1205, qhp1206, qhp1207, qhp1208, qhp1209, qhp1210, qhp1211, qhp1212, qhp1213, qhp1214, qhp1215, qhp1216, qhp1217, qhp1218, qhp1219, qhp1220, qhp1221, qhp1222, qhp1223, qhp1224, qhp1225, qhp1226, qhp1227, qhp1228, qhp1229, qhp1230, qhp1231, qhp1232, qhp1233, qhp1234, qhp1235, qhp1236, qhp1237, qhp1238, qhp1239, qhp1240, qhp1241, qhp1242, qhp1243, qhp1244, qhp1245, qhp1246, qhp1247, qhp1248, qhp1249, qhp1250, qhp1251, qhp1252, qhp1253, qhp1254, qhp1255, qhp1256, qhp1257, qhp1258, qhp1259, qhp1260, qhp1261, qhp1262, qhp1263, qhp1264, qhp1265, qhp1266, qhp1267, qhp1268, qhp1269, qhp1270, qhp1271, qhp1272, qhp1273, qhp1274, qhp1275, qhp1276, qhp1277, qhp1278, qhp1279, qhp1280, qhp1281, qhp1282, qhp1283, qhp1284, qhp1285, qhp1286, qhp1287, qhp1288, qhp1289, qhp1290, qhp1291, qhp1292, qhp1293, qhp1294, qhp1295, qhp1296, qhp1297, qhp1298, qhp1299, qhp1300, qhp1301, qhp1302, qhp1303, qhp1304, qhp1305, qhp1306, qhp1307, qhp1308, qhp1309, qhp1310, qhp1311, qhp1312, qhp1313, qhp1314, qhp1315, qhp1316, qhp1317, qhp1318, qhp1319, qhp1320, qhp1321, qhp1322, qhp1323, qhp1324, qhp1325, qhp1326, qhp1327, qhp1328, qhp1329, qhp1330, qhp1331, qhp1332, qhp1333, qhp1334, qhp1335, qhp1336, qhp1337, qhp1338, qhp1339, qhp1340, qhp1341, qhp1342, qhp1343, qhp1344, qhp1345, qhp1346, qhp1347, qhp1348, qhp1349, qhp1350, qhp1351, qhp1352, qhp1353, qhp1354, qhp1355, qhp1356, qhp1357, qhp1358, qhp1359, qhp1360, qhp1361, qhp1362, qhp1363, qhp1364, qhp1365, qhp1366, qhp1367, qhp1368, qhp1369, qhp1370, qhp1371, qhp1372, qhp1373, qhp1374, qhp1375, qhp1376, qhp1377, qhp1378, qhp1379, qhp1380, qhp1381, qhp1382, qhp1383, qhp1384, qhp1385, qhp1386, qhp1387, qhp1388, qhp1389, qhp1390, qhp1391, qhp1392, qhp1393, qhp1394, qhp1395, qhp1396, qhp1397, qhp1398, qhp1399, qhp1400, qhp1401, qhp1402, qhp1403, qhp1404, qhp1405, qhp1406, qhp1407, qhp1408, qhp1409, qhp1410, qhp1411, qhp1412, qhp1413, qhp1414, qhp1415, qhp1416, qhp1417, qhp1418, qhp1419, qhp1420, qhp1421, qhp1422, qhp1423, qhp1424, qhp1425, qhp1426, qhp1427, qhp1428, qhp1429, qhp1430, qhp1431, qhp1432, qhp1433, qhp1434, qhp1435, qhp1436, qhp1437, qhp1438, qhp1439, qhp1440, qhp1441, qhp1442, qhp1443, qhp1444, qhp1445, qhp1446, qhp1447, qhp1448, qhp1449, qhp1450, qhp1451, qhp1452, qhp1453, qhp1454, qhp1455, qhp1456, qhp1457, qhp1458, qhp1459, qhp1460, qhp1461, qhp1462, qhp1463, qhp1464, qhp1465, qhp1466, qhp1467, qhp1468, qhp1469, qhp1470, qhp1471, qhp1472, qhp1473, qhp1474, qhp1475, qhp1476, qhp1477, qhp1478, qhp1479, qhp1480, qhp1481, qhp1482, qhp1483, qhp1484, qhp1485, qhp1486, qhp1487, qhp1488, qhp1489, qhp1490, qhp1491, qhp1492, qhp1493, qhp1494, qhp1495, qhp1496, qhp1497, qhp1498, qhp1499, qhp1500, qhp1501, qhp1502, qhp1503, qhp1504, qhp1505, qhp1506, qhp1507, qhp1508, qhp1509, qhp1510, qhp1511, qhp1512, qhp1513, qhp1514, qhp1515, qhp1516, qhp1517, qhp1518, qhp1519, qhp1520, qhp1521, qhp1522, qhp1523, qhp1524, qhp1525, qhp1526, qhp1527, qhp1528, qhp1529, qhp1530, qhp1531, qhp1532, qhp1533, qhp1534, qhp1535, qhp1536, qhp1537, qhp1538, qhp1539, qhp1540, qhp1541, qhp1542, qhp1543, qhp1544, qhp1545, qhp1546, qhp1547, qhp1548, qhp1549, qhp1550, qhp1551, qhp1552, qhp1553, qhp1554, qhp1555, qhp1556, qhp1557, qhp1558, qhp1559, qhp1560, qhp1561, qhp1562, qhp1563, qhp1564, qhp1565, qhp1566, qhp1567, qhp1568, qhp1569, qhp1570, qhp1571, qhp1572, qhp1573, qhp1574, qhp1575, qhp1576, qhp1577, qhp1578, qhp1579, qhp1580, qhp1581, qhp1582, qhp1583, qhp1584, qhp1585, qhp1586, qhp1587, qhp1588, qhp1589, qhp1590, qhp1591, qhp1592, qhp1593, qhp1594, qhp1595, qhp1596, qhp1597, qhp1598, qhp1599, qhp1600, qhp1601, qhp1602, qhp1603, qhp1604, qhp1605, qhp1606, qhp1607, qhp1608, qhp1609, qhp1610, qhp1611, qhp1612, qhp1613, qhp1614, qhp1615, qhp1616, qhp1617, qhp1618, qhp1619, qhp1620, qhp1621, qhp1622, qhp1623, qhp1624, qhp1625, qhp1626, qhp1627, qhp1628, qhp1629, qhp1630, qhp1631, qhp1632, qhp1633, qhp1634, qhp1635, qhp1636, qhp1637, qhp1638, qhp1639, qhp164
```



```
q3a[1-1] = 0.0;
q4a[1-1] = 0.0;
q5a[1-1] = 0.0;
q6a[1-1] = 0.0;

volt1[1] = 0.0;
volt2[1] = 0.0;
volt3[1] = 0.0;
volt4[1] = 0.0;
volt5[1] = 0.0;
volt6[1] = 0.0;
```

```
ag11 = 0.0;
ag21 = 0.0;
ag31 = 0.0;
ag41 = 0.0;
ag51 = 0.0;
ag61 = 0.0;
ag71 = 0.0;
ag81 = 0.0;
ag91 = 0.0;
ag101 = 0.0;
ag111 = 0.0;
ag121 = 0.0;
ag131 = 0.0;
ag141 = 0.0;
ag151 = 0.0;
ag161 = 0.0;
ag171 = 0.0;
ag181 = 0.0;
ag191 = 0.0;
ag201 = 0.0;
ag211 = 0.0;
ag221 = 0.0;
ag231 = 0.0;
ag241 = 0.0;
ag251 = 0.0;
ag261 = 0.0;
ag271 = 0.0;
ag281 = 0.0;
ag291 = 0.0;
ag301 = 0.0;
ag311 = 0.0;
ag321 = 0.0;
ag331 = 0.0;
```

// CONDICIONES INICIALES DE PARAMETROS

```
ag11_1 = 0.0;
ag21_1 = 0.0;
ag31_1 = 0.0;
ag41_1 = 0.0;
ag51_1 = 0.0;
ag61_1 = 0.0;
ag71_1 = 0.0;
ag81_1 = 0.0;
ag91_1 = 0.0;
ag101_1 = 0.0;
ag111_1 = 0.0;
ag121_1 = 0.0;
ag131_1 = 0.0;
ag141_1 = 0.0;
```

// CONDICIONES INICIALES
// U PARAMETROS NOMIALES

```
ag151_1 = 0.0;
ag161_1 = 0.0;
ag171_1 = 0.0;
ag181_1 = 0.0;
ag191_1 = 0.0;
ag201_1 = 0.0;
ag211_1 = 0.0;
ag221_1 = 0.0;
ag231_1 = 0.0;
ag241_1 = 0.0;
ag251_1 = 0.0;
ag261_1 = 0.0;
ag271_1 = 0.0;
```

```

ag201_1 = 0.0;
ag291_1 = 0.0;
ag301_1 = 0.0;
ag311_1 = 0.0;
ag321_1 = 0.0;
ag331_1 = 0.0;

```

```

agp11 = 0.0;
agp21 = 0.0;
agp31 = 0.0;
agp41 = 0.0;
agp51 = 0.0;
agp61 = 0.0;
agp71 = 0.0;
agp81 = 0.0;
agp91 = 0.0;
agp101 = 0.0;
agp111 = 0.0;
agp121 = 0.0;
agp131 = 0.0;
agp141 = 0.0;
agp151 = 0.0;
agp161 = 0.0;
agp171 = 0.0;
agp181 = 0.0;
agp191 = 0.0;
agp201 = 0.0;
agp211 = 0.0;
agp221 = 0.0;
agp231 = 0.0;
agp241 = 0.0;
agp251 = 0.0;
agp261 = 0.0;
agp271 = 0.0;
agp281 = 0.0;
agp291 = 0.0;
agp301 = 0.0;
agp311 = 0.0;
agp321 = 0.0;
agp331 = 0.0;

```

```

agp11_1 = 0.0;
agp21_1 = 0.0;
agp31_1 = 0.0;
agp41_1 = 0.0;
agp51_1 = 0.0;
agp61_1 = 0.0;
agp71_1 = 0.0;
agp81_1 = 0.0;
agp91_1 = 0.0;
agp101_1 = 0.0;
agp111_1 = 0.0;
agp121_1 = 0.0;
agp131_1 = 0.0;
agp141_1 = 0.0;
agp151_1 = 0.0;
agp161_1 = 0.0;
agp171_1 = 0.0;
agp181_1 = 0.0;
agp191_1 = 0.0;
agp201_1 = 0.0;
agp211_1 = 0.0;
agp221_1 = 0.0;
agp231_1 = 0.0;
agp241_1 = 0.0;
agp251_1 = 0.0;
agp261_1 = 0.0;
agp271_1 = 0.0;
agp281_1 = 0.0;
agp291_1 = 0.0;
agp301_1 = 0.0;
agp311_1 = 0.0;
agp321_1 = 0.0;
agp331_1 = 0.0;

```

```
// Condiciones iniciales del nuevo estimador de velocidad
```

```

p1 = 0.0;
p2 = 0.0;
p3 = 0.0;
p4 = 0.0;
p5 = 0.0;
p6 = 0.0;

p1.1 = 0.0;
p2.1 = 0.0;
p3.1 = 0.0;
p4.1 = 0.0;
p5.1 = 0.0;
p6.1 = 0.0;

pp1 = 0.0;
pp2 = 0.0;
pp3 = 0.0;
pp4 = 0.0;
pp5 = 0.0;
pp6 = 0.0;

ef1 = 0.0;
ef2 = 0.0;
ef3 = 0.0;
ef4 = 0.0;
ef5 = 0.0;
ef6 = 0.0;
// ASIGNACION DE VALORES PARA LAS TRAYECTORIAS
Cl(1);
crg = pow(180.2)/pow(pi,2);
C = pi/180.0;
k1 = 0.0+C;
k2 = 40.0+C +1.0;
k3 = 10.0+C +1.0;
k4 = 25.0+C +1.0;
k5 = 35.0+C +1.0;
k6 = 40.0+C +1.0;
k7 = 0.0+C;
k8 = 10.0+C +1.0;
k9 = 0.0+C;
k10 = 30.0+C +1.0;//65.0+C;
k11 = 0.0+C;
k12 = 30.0+C +1.0;//65.0+C;

ke1 = 0.5;
ke2 = 0.5;
ke3 = 7.5;
ke4 = 0.5;
ke5 = 0.5;
ke6 = 0.5;

fv1 = 2.0*pi/4.0; // Es la frecuencia correcta
fv2 = 2.0*pi/4.0;
fv3 = 2.0*pi/4.0;
fv4 = 2.0*pi/4.0;
fv5 = 2.0*pi/4.0;
fv6 = 2.0*pi/4.0;
//SUSTITUCION DE CONSTANTES EN TRAYECTORIAS
kee1 = pow(ke1,2);
kee2 = pow(ke2,2);
kee3 = pow(ke3,2);
kee4 = pow(ke4,2);
kee5 = pow(ke5,2);
kee6 = pow(ke6,2);

fvv1 = pow(fv1,2);
fvv2 = pow(fv2,2);
fvv3 = pow(fv3,2);
fvv4 = pow(fv4,2);
fvv5 = pow(fv5,2);
fvv6 = pow(fv6,2);
// MATRICES DE DINAMICA DE MOTORES
Dk[0][0] = (Ka1/Ra1)*(1/r1);
Dk[1][1] = (Ka1/Ra1)*(1/r2);
Dk[2][2] = (Ka1/Ra1)*(1/r3);
Dk[3][3] = (Ka2/Ra2)*(1/r4);
Dk[4][4] = (Ka2/Ra2)*(1/r5);

```

```

Dk[5][5] = (Ka3/Ra3)*(1/r6),

Dn[0][0] = 1/pow(r1,2),
Dn[1][1] = 1/pow(r2,2),
Dn[2][2] = 1/pow(r3,2),
Dn[3][3] = 1/pow(r4,2),
Dn[4][4] = 1/pow(r5,2),
Dn[5][5] = 1/pow(r6,2),

Df[0][0] = fm * ((Ka1+Kb1)/Ra1),
Df[1][1] = fm * ((Ka1+Kb1)/Ra1),
Df[2][2] = fm * ((Ka1+Kb1)/Ra1),
Df[3][3] = fm * ((Ka2+Kb2)/Ra2),
Df[4][4] = fm * ((Ka2+Kb2)/Ra2),
Df[5][5] = fm * ((Ka2+Kb3)/Ra3),

Dj[0][0] = Jm1,
Dj[1][1] = Jm1,
Dj[2][2] = Jm1,
Dj[3][3] = Jm2,
Dj[4][4] = Jm2,
Dj[5][5] = Jm3;

InvMatrix (Dn,6,Dn,1);
InvMatrix (Dk,6,Dk,1);

MatrixMul(Dn,1,Dj,6,6,G,Dn,1Dj);
MatrixMul(Dn,1,Df,6,6,G,Dn,1Df);
MatrixMul(Dn,1,Dk,6,6,G,Dn,1Dk);
MatrixMul(Dk,1,Dn,6,6,G,Dk,1Dn);

//***** ENTRADA DE GANANCIAS DESDE EL PANEL DE CONTROL *****
GetCtrlVal (panelHandle, PANEL_LAM_1, &lam1),
GetCtrlVal (panelHandle, PANEL_LAM_2, &lam2),
GetCtrlVal (panelHandle, PANEL_LAM_3, &lam3), //GANANCIAS DE OBSERVADOR
GetCtrlVal (panelHandle, PANEL_LAM_4, &lam4),
GetCtrlVal (panelHandle, PANEL_LAM_5, &lam5),
GetCtrlVal (panelHandle, PANEL_LAM_6, &lam6),

GetCtrlVal (panelHandle, PANEL_KD_1, &kd1),
GetCtrlVal (panelHandle, PANEL_KD_2, &kd2),
GetCtrlVal (panelHandle, PANEL_KD_3, &kd3),
GetCtrlVal (panelHandle, PANEL_KD_4, &kd4), // GANANCIAS DERIVATIVAS
GetCtrlVal (panelHandle, PANEL_KD_5, &kd5),
GetCtrlVal (panelHandle, PANEL_KD_6, &kd6),

GetCtrlVal (panelHandle, PANEL_KP_1, &kp1),
GetCtrlVal (panelHandle, PANEL_KP_2, &kp2),
GetCtrlVal (panelHandle, PANEL_KP_3, &kp3), // GANANCIAS PROPORCIONALES
GetCtrlVal (panelHandle, PANEL_KP_4, &kp4),
GetCtrlVal (panelHandle, PANEL_KP_5, &kp5),
GetCtrlVal (panelHandle, PANEL_KP_6, &kp6);

// INICIO DEL LOOP DE CONTROL
Sincro = Timer();
while (!cuustras && sal==1) {
  tiempo = Timer();
  if (i-->interm0+1) {
    ttt[cont0] = Timer() - Sincro;
    cont0 = cont0 + 1;
    interm0 = interm0 + almacena;
  }
  if (!cuustras & alerta) ttt[cont0] = Timer() - Sincro;
  t = Timer() - Sincro;
  fint = t;
  //***** LECTURA DE ENCODER *****
  status = flex_read_encoder_rtn (boardID1, axisNum1, &q1);
  status = flex_read_encoder_rtn (boardID1, axisNum2, &q2);
  status = flex_read_encoder_rtn (boardID1, axisNum3, &q3);
  status = flex_read_encoder_rtn (boardID1, axisNum4, &q4);
  status = flex_read_encoder_rtn (boardID1, axisNum5, &q5);
  status = flex_read_encoder_rtn (boardID1, axisNum6, &q6);
  // Empieza la Adaptación de Siotime-I1 (Mod. al Observador de Arteaga)
  // Empieza derivación de posición
  q1 = q1*0.0/100000.0; // Articulacion 1
  q1 = q1*PI/180.0;
  q2 = q2 * 100000.0; // Articulacion 2

```



```

q2g = qq2*90.0/100000.0; //grados
q2 = q2g*pi/180.0; //radianes
q3g = -qq3*90.0/100000.0; // Articulación 3
q3 = q3g*pi/180.0;
q4g = -qq4*90.0/50500.0; // Articulación 4
q4 = q4g*pi/180.0;
q5g = -qq5*90.0/50500.0; // Articulación 5
q5 = q5g*pi/180.0; //rad
q6g = -qq6*90.0/50500.0; // Articulación 6
q6 = q6g*pi/180.0; //rad

q1a[1] = q1; //q1a[1] = q1*180.0/pi;
q2a[1] = q2;
q3a[1] = q3;
q4a[1] = q4;
q5a[1] = q5;
q6a[1] = q6;

if(t==0)
(
q1a[1]-1 = q1;
q2a[1]-1 = q2;
q3a[1]-1 = q3;
q4a[1]-1 = q4;
q5a[1]-1 = q5;
q6a[1]-1 = q6;
)
//..... TRAYECTORIAS DESEADAS .....
q41 = (k1+k2 *sin(fw2t))*(1-exp(-k1*pow(t,3)));
q42 = (pi/2)*(k3+k4 *sin(fw2t))*(1-exp(-k2*pow(t,3)));
q43 = (k5+k6 *sin(fw3t))*(1-exp(-k3*pow(t,3)));
q44 = (k7+k8 *sin(fw4t))*(1-exp(-k4*pow(t,3)));
q45 = (k9+k10*sin(fw5t))*(1-exp(-k5*pow(t,3)));
q46 = (k11+k12*sin(fw6t))*(1-exp(-k6*pow(t,3)));

qp41 = (k1+k2 *sin(fw2t) )*(3.0*ke1*pow(t,2)+exp(-k1*pow(t,3)))+(k2 *fw2*cos(fw2t))*(1-exp(-k1*pow(t,3)));
qp42 = (k3+k4 *sin(fw2t) )*(3.0*ke2*pow(t,2)+exp(-k2*pow(t,3)))+(k4 *fw2*cos(fw2t))*(1-exp(-k2*pow(t,3)));
qp43 = (k5+k6 *sin(fw3t) )*(3.0*ke3*pow(t,2)+exp(-k3*pow(t,3)))+(k6 *fw3*cos(fw3t))*(1-exp(-k3*pow(t,3)));
qp44 = (k7+k8 *sin(fw4t) )*(3.0*ke4*pow(t,2)+exp(-k4*pow(t,3)))+(k8 *fw4*cos(fw4t))*(1-exp(-k4*pow(t,3)));
qp45 = (k9+k10 *sin(fw5t) )*(3.0*ke5*pow(t,2)+exp(-k5*pow(t,3)))+(k10*fw5*cos(fw5t))*(1-exp(-k5*pow(t,3)));
qp46 = (k11+k12 *sin(fw6t) )*(3.0*ke6*pow(t,2)+exp(-k6*pow(t,3)))+(k12*fw6*cos(fw6t))*(1-exp(-k6*pow(t,3)));

qp41 = (k1+k2 *sin(fw2t) )*(6.0*ke1*exp(-k1*pow(t,3))-9.9*ke1*pow(t,4)+exp(-k1*pow(t,3)));
qp42 = (k3+k4 *sin(fw2t) )*(6.0*ke2*exp(-k2*pow(t,3))-9.9*ke2*pow(t,4)+exp(-k2*pow(t,3)));
qp43 = (k5+k6 *sin(fw3t) )*(6.0*ke3*exp(-k3*pow(t,3))-9.9*ke3*pow(t,4)+exp(-k3*pow(t,3)));
qp44 = (k7+k8 *sin(fw4t) )*(6.0*ke4*exp(-k4*pow(t,3))-9.9*ke4*pow(t,4)+exp(-k4*pow(t,3)));
qp45 = (k9+k10 *sin(fw5t) )*(6.0*ke5*exp(-k5*pow(t,3))-9.9*ke5*pow(t,4)+exp(-k5*pow(t,3)));
qp46 = (k11+k12 *sin(fw6t) )*(6.0*ke6*exp(-k6*pow(t,3))-9.9*ke6*pow(t,4)+exp(-k6*pow(t,3)));
qp46 = (k11+k12 *sin(fw6t) )*(6.0*ke6*exp(-k6*pow(t,3))-9.9*ke6*pow(t,4)+exp(-k6*pow(t,3)));
qp46 = (k11+k12 *sin(fw6t) )*(6.0*ke6*exp(-k6*pow(t,3))-9.9*ke6*pow(t,4)+exp(-k6*pow(t,3)));
qp46 = (k11+k12 *sin(fw6t) )*(6.0*ke6*exp(-k6*pow(t,3))-9.9*ke6*pow(t,4)+exp(-k6*pow(t,3)));

q41[1] = q41 *180/pi; // grafica el segundo bloque, son las qd
q42[1] = q42 *180/pi;
q43[1] = q43 *180/pi;
q44[1] = q44 *180/pi;
q45[1] = q45 *180/pi;
q46[1] = q46 *180/pi;
// ERRORES DE POSICION
q1 = q1 - q41;
q2 = q2 - q42;
q3 = q3 - q43; // Esta entre -10 y 10
q4 = q4 - q44;
q5 = q5 - q45;
q6 = q6 - q46;
qt[0][0] = q1;
qt[1][0] = q2;
qt[2][0] = q3;
qt[3][0] = q4;
qt[4][0] = q5;
qt[5][0] = q6;
// Estimacion de la velocidad
pp1 = (kk *1) * p1 - (pow(kk,2) + 1) * qt1;
pp2 = (kk *1) * p2 - (pow(kk,2) + 1) * qt2;

```

TESIS CON
FALLA DE ORIGEN

```

pp3 = - (kk * i) * p3 - (pow(kk,2) * i) * qt3;
pp4 = - (kk * i) * p4 - (pow(kk,2) * i) * qt4;
pp5 = - (kk * i) * p5 - (pow(kk,2) * i) * qt5;
pp6 = - (kk * i) * p6 - (pow(kk,2) * i) * qt6;
ef1 = kk * qt1 + p1;
ef2 = kk * qt2 + p2;
ef3 = kk * qt3 + p3;
ef4 = kk * qt4 + p4;
ef5 = kk * qt5 + p5;
ef6 = kk * qt6 + p6;
qp1 = ef1 * qpd1;
qp2 = ef2 * qpd2;
qp3 = ef3 * qpd3; // Estimador de velocidad
qp4 = ef4 * qpd4;
qp5 = ef5 * qpd5;
qp6 = ef6 * qpd6;
p1 = p1 * h + pp1;
p2 = p2 * h + pp2;
p3 = p3 * h + pp3; // Integración
p4 = p4 * h + pp4;
p5 = p5 * h + pp5;
p6 = p6 * h + pp6;
qptg[1] = qp1;
qpg[1] = qp2;
qpg[1] = qp3; // Grafica de las VELOCIDADES
qpg[1] = qp4;
qpg[1] = qp5;
qpg[1] = qp6;
// ERRORES DE VELOCIDAD
qtp1 = qp1 - qpd1;
qtp2 = qp2 - qpd2;
qtp3 = qp3 - qpd3;
qtp4 = qp4 - qpd4;
qtp5 = qp5 - qpd5;
qtp6 = qp6 - qpd6;
// VECTOR DE ERRORES DE VELOCIDAD
qtp[0][0] = qtp1;
qtp[1][0] = qtp2;
qtp[2][0] = qtp3;
qtp[3][0] = qtp4;
qtp[4][0] = qtp5;
qtp[5][0] = qtp6;
// Asignacion para graficar la qt
if ((i==intern1+i)&&((muestras-i)>chance))
{
  q1gra[cont1] = qt1*(180.0/p1);
  q2gra[cont1] = qt2*(180.0/p1);
  q3gra[cont1] = qt3*(180.0/p1);
  q4gra[cont1] = qt4*(180.0/p1);
  q5gra[cont1] = qt5*(180.0/p1);
  q6gra[cont1] = qt6*(180.0/p1);
  cont1 = cont1 + 1;
  intern1 = intern1 + almazena;
}
// Calculo del índice de desempeño
ndqt = erg * ( pow(qt1,2) + pow(qt2,2) + pow(qt3,2) + pow(qt4,2) + pow(qt5,2) + pow(qt6,2) );
ndqt1 = erg * ( pow(qt1,2) );
ndqt2 = erg * ( pow(qt2,2) );
ndqt3 = erg * ( pow(qt3,2) );
ndqt4 = erg * ( pow(qt4,2) );
ndqt5 = erg * ( pow(qt5,2) );
ndqt6 = erg * ( pow(qt6,2) );
TL22 = ndqt * h * TL22;
TL221 = ndqt1 * h * TL221;
TL222 = ndqt2 * h * TL222;
TL223 = ndqt3 * h * TL223;
TL224 = ndqt4 * h * TL224;
TL225 = ndqt5 * h * TL225;
TL226 = ndqt6 * h * TL226;
// Fin del calculo del índice de desempeño
//CALCULO DE QRS
qrp1 = qpd1 - lam1 * qt1;
qrp2 = qpd2 - lam2 * qt2;
qrp3 = qpd3 - lam3 * qt3; // Esta entre -10 y 10 sólo se pasa una decimas
qrp4 = qpd4 - lam4 * qt4;
qrp5 = qpd5 - lam5 * qt5;
qrp6 = qpd6 - lam6 * qt6;

```

```

qrrp1 = qppd1 - lam1 * qtp1;
qrrp2 = qppd2 - lam2 * qtp2;
qrrp3 = qppd3 - lam3 * qtp3;
qrrp4 = qppd4 - lam4 * qtp4;
qrrp5 = qppd5 - lam5 * qtp5;
qrrp6 = qppd6 - lam6 * qtp6;
// ***** Seccion de Ley de Adaptación *****
sa1 = qtp1 + lam1 * qt1;
sa2 = qtp2 + lam2 * qt2;
sa3 = qtp3 + lam3 * qt3; // MODO DESLIZANTE
sa4 = qtp4 + lam4 * qt4;
sa5 = qtp5 + lam5 * qt5;
sa6 = qtp6 + lam6 * qt6;
sa[0][0] = sa1;
sa[1][0] = sa2;
sa[2][0] = sa3;
sa[3][0] = sa4;
sa[4][0] = sa5;
sa[5][0] = sa6;
// ***** Seccion de Ley de Adaptación ***** X
//***** FUNCION SIGNO *****
if(qrrp1 > 0)
  signq1 = 1;
else if(qrrp1 < 0)
  signq1 = -1;
else if(qrrp1 == 0)
  signq1 = 0;

if(qrrp2 > 0)
  signq2 = 1;
else if(qrrp2 < 0)
  signq2 = -1;
else if(qrrp2 == 0)
  signq2 = 0;

if(qrrp3 > 0)
  signq3 = 1;
else if(qrrp3 < 0)
  signq3 = -1;
else if(qrrp3 == 0)
  signq3 = 0;

if(qrrp4 > 0)
  signq4 = 1;
else if(qrrp4 < 0)
  signq4 = -1;
else if(qrrp4 == 0)
  signq4 = 0;

if(qrrp5 > 0)
  signq5 = 1;
else if(qrrp5 < 0)
  signq5 = -1;
else if(qrrp5 == 0)
  signq5 = 0;

if(qrrp6 > 0)
  signq6 = 1;
else if(qrrp6 < 0)
  signq6 = -1;
else if(qrrp6 == 0)
  signq6 = 0;
// ***** Seccion de Ley de Adaptación ***** X
// ***** Inicia Modelo dinamico del robot *****
c2q22q3a2q5 = cos(-2*qd5+2*qd3+2*qd2);
c2q22q3a2q5m2q6 = cos(-2*qd5+2*qd4+2*qd3+2*qd2);
c2q22q3a2q6 = cos(-2*qd6+2*qd3+2*qd2);
c2q22q3a4q5q2q6 = cos(2*qd5-2*qd5+2*qd3+2*qd2+qd4);
c2q22q3a4m2q5m2q6 = cos(-2*qd6-2*qd5+2*qd3+2*qd2-qd4);
cm2q42q5m2q6 = cos(-2*qd6+2*qd5+2*qd4);
c2q22q3a2q4m2q5m2q6 = cos(2*qd6-2*qd4+2*qd3+2*qd2-qd5);
c2q22q3a2q5q2q6 = cos(2*qd6-2*qd5+2*qd3+2*qd2);
c2q22q3a2q4m2q5m2q6 = cos(-2*qd6+2*qd4+2*qd3+2*qd2-qd5);
c5 = cos(qd5);
c2q22q3a4q5m2q6 = cos(2*qd6+qd5+2*qd3+2*qd2+qd4);
c2q22q3a4q5 = cos(-qd5+2*qd3+2*qd2);

```

c2q42q5	* cos (2*πq5+2*πq4) ;
c2q22q32q42q52q6	* cos (2*πq6+2*πq5+2*πq4+2*πq3+2*πq2) ;
c2q22q3m2q452q6	* cos (2*πq6+2*πq5+2*πq3+2*πq2+πq5) ;
cm2q42q52q6	* cos (2*πq6+2*πq5-2*πq4) ;
c2q22q3m2q52q6	* cos (2*πq6+2*πq5+2*πq3+2*πq2) ;
c2q22q3m4m5	* cos (-πq5-πq4+2*πq3+2*πq2) ;
c2q22q3m4m52q6	* cos (2*πq6-πq5+2*πq3+2*πq2-πq4) ;
c2q22q32q4	* cos (2*πq6+2*πq3+2*πq2) ;
c2q22q3m2q4m2q6	* cos (-2*πq6-2*πq4+2*πq3+2*πq2) ;
c2q22q3m2q5m2q6	* cos (πq5-πq4+2*πq3+2*πq2) ;
c2q22q3m2q6m2q6	* cos (2*πq6+2*πq3+2*πq2) ;
c2q22q3m4m2q6	* cos (-2*πq6-2*πq4+2*πq3+2*πq2) ;
c2q22q3m4m2q5m2q6	* cos (2*πq6-2*πq5+2*πq4+2*πq3+2*πq2) ;
c2q452q6	* cos (-πq5+πq4+2*πq3+2*πq2) ;
c2q22q3m2q5m2q6	* cos (2*πq6+2*πq5+2*πq4+2*πq3+2*πq2) ;
c2q22q3m2q6m2q6	* cos (-πq5+πq4+2*πq3+2*πq2) ;
c2q42q6	* cos (2*πq6+2*πq5) ;
c2q22q32q452q6	* cos (2*πq6+2*πq4+2*πq3+2*πq2+πq5) ;
c2q22q32q4m2q6	* cos (-2*πq6+2*πq4+2*πq3+2*πq2) ;
c2q6	* cos (2*πq6) ;
c2q22q32q6	* cos (2*πq6+2*πq3+2*πq2) ;
c2q6	* cos (2*πq6) ;
c2q22q3m2q42q5m2q6	* cos (-2*πq6+2*πq5-2*πq4+2*πq3+2*πq2) ;
c2q22q3m2q42q6	* cos (2*πq6-2*πq4+2*πq3+2*πq2) ;
c2q22q3q5	* cos (πq5+2*πq3+2*πq2) ;
c2q22q32q4m3q52q6	* cos (2*πq6+2*πq4+2*πq3+2*πq2-πq5) ;
c2q22q3m4m2q6	* cos (-2*πq6+2*πq4+2*πq3+2*πq2+πq4) ;
c23m2q4	* cos (2*πq4+πq3+πq2) ;
c232q4	* cos (2*πq4+πq3+πq2) ;
c23m2q4m2q5	* cos (-2*πq6+2*πq4+πq3+πq2) ;
c232q42q5	* cos (2*πq6+πq4+πq3+πq2) ;
c23m2q4m2q5	* cos (-2*πq6+2*πq4+πq3+πq2) ;
c2342q5	* cos (2*πq6+πq4+πq3+πq2) ;
c23m42q5	* cos (-2*πq6+2*πq4+πq3+πq2) ;
c23m2q452q6	* cos (2*πq6-2*πq4+πq3+πq2) ;
c23m2q6	* cos (-2*πq6+πq4+πq3+πq2) ;
c23m4m5	* cos (πq5+πq4+πq3+πq2) ;
c2345	* cos (πq5+πq4+πq3+πq2) ;
c2345	* cos (πq5+πq4+πq3+πq2) ;
c234m5	* cos (πq5+πq4+πq3+πq2) ;
cm2q45m2q6	* cos (-2*πq6-2*πq4+πq5) ;
c2q22q3m2q45m2q6	* cos (-2*πq6-2*πq4+2*πq3+2*πq2+πq5) ;
c2q22q3m2q452q6	* cos (2*πq6-2*πq4+2*πq3+2*πq2) ;
cm2q42q6	* cos (2*πq6-2*πq4) ;
c2q22q32q42q6	* cos (2*πq6+2*πq4+2*πq3+2*πq2) ;
c2q22q3m2q4m2q52q6	* cos (2*πq6-2*πq4+2*πq3+2*πq2) ;
c2q22q3m2q4m2q6	* cos (-2*πq6+πq5+2*πq4+2*πq3+2*πq2) ;
c2q22q3m4m2q6	* cos (2*πq6-πq5+2*πq3+2*πq2+πq4) ;
c2q22q3m452q6	* cos (πq5+πq4+2*πq3+2*πq2) ;
c2q22q3m2q42q52q6	* cos (2*πq6+2*πq5-2*πq4+2*πq3+2*πq2) ;
c2q4	* cos (2*πq2) ;
c2q45m2q6	* cos (2*πq6+2*πq4+πq5) ;
c2q22q3m42q52q6	* cos (2*πq6+2*πq5+2*πq4+2*πq3+2*πq2-πq4) ;
c2q22q32q4	* cos (2*πq4+2*πq3+2*πq2) ;
c2q22q3m4m5m2q6	* cos (-2*πq6-πq5+2*πq4+2*πq3+2*πq2) ;
c2q4	* cos (2*πq4) ;
c2q22q32q4m2q6	* cos (-2*πq6+2*πq4+2*πq3+2*πq2) ;
c2q22q3m4m2q6	* cos (-2*πq6-πq4+2*πq3+2*πq2) ;
c2q22q3m4m5m2q6	* cos (-2*πq6-πq5+2*πq4+2*πq3+2*πq2+πq4) ;
c2q42q5m2q6	* cos (-2*πq6+2*πq5+2*πq4) ;
c2q22q3m2q4	* cos (-2*πq4+2*πq3+2*πq2) ;
c2q22q3m4m2q6	* cos (-2*πq6-πq4+2*πq3+2*πq2) ;
c2q22q3m4m5m2q6	* cos (-2*πq6-πq5+2*πq4+2*πq3+2*πq2+πq4) ;
c2q42q5m2q6	* cos (-2*πq6+2*πq5+2*πq4) ;
c2q22q3m2q4m5m2q6	* cos (-2*πq6-2*πq4+2*πq3+2*πq2-πq5) ;


```

c42q8      * cos(2*qd6+qd4);
c562q6     * cos(-2*qd6+qd5);
c57q6      * cos(2*qd6+qd5);
c2q4m2q5  * cos(-2*qd5+2*qd4);
c462q5     * cos(-2*qd5+qd4);
c46s      * cos(qd5+qd5);

c2q234m5  * cos(-qd5+qd4+qd3+2*qd2);
c34m5     * cos(-qd5+qd4+qd3);
c34s      * cos(qd5+qd4+qd3);
c36        * cos(qd5+qd3);
c2q23m45  * cos(qd5+qd4+qd3+2*qd2);
c2q234s   * cos(qd5+qd4+qd3+2*qd2);
c2q235    * cos(qd5+qd3+2*qd2);
ca34m5    * cos(-qd5+qd4+qd3);
c2q234m5s * cos(-qd5+qd4+qd3+2*qd2);
cm35      * cos(qd5+qd3);
cm34s     * cos(qd5+qd4+qd3);
c2q23m4s  * cos(-qd5+qd3+2*qd2);
c2q23     * cos(2*qd2+qd3);
c3         * cos(qd3);
c2m4s     * cos(qd5+qd4+qd2);
c24s      * cos(qd5+qd4+qd2);
c2m4m5    * cos(-qd5+qd4+qd2);
c24m5     * cos(-qd5+qd4+qd2);
c24s      * cos(-qd5+qd4+qd2);
c2 = cos(qd2);

s35       * sin(qd5+qd3);
s34s      * sin(qd5+qd4+qd3);
s2q23     * sin(2*qd2+qd3);
s2q23q4m5 * sin(-qd5+qd4+qd3+2*qd2);
sm35      * sin(qd5+qd3);
s2q234s   * sin(qd5+qd4+qd3+2*qd2);
s2q23m5   * sin(-qd5+qd3+2*qd2);
s2q23m4s  * sin(-qd5+qd4+qd3+2*qd2);
sm3m4s    * sin(qd5+qd4+qd3);
s3         * sin(qd3);
s2q235    * sin(2*qd2+qd3+qd5);
s2q23m45  * sin(qd5+qd4+qd3+2*qd2);
s3m4s     * sin(qd5+qd4+qd3);
s2m4s     * sin(qd5+qd4+qd2);
s2m4m5    * sin(-qd5+qd4+qd2);
s24m5     * sin(-qd5+qd4+qd2);
s24s      * sin(qd5+qd4+qd2);
sm34s     * sin(qd5+qd4+qd3);
s2q235s   * sin(qd5+qd3+2*qd2);
s34m5     * sin(-qd5+qd4+qd3);
s2q23m45s * sin(qd5+qd4+qd3+2*qd2);
sm34m5    * sin(-qd5+qd4+qd3);

s2q22q32q4m2q5 * sin(-2*qd5+2*qd4+2*qd3+2*qd2);
s2q22q34m2q5  * sin(-2*qd5+qd4+2*qd3+2*qd2);
s2q22q3m2q4   * sin(-2*qd4+2*qd3+2*qd2);
s2m4m2q5      * sin(-2*qd5+2*qd4);
s2q42q5       * sin(2*qd5+2*qd4);
s2q22q3m42q5 * sin(2*qd5+qd4+2*qd3+2*qd2);
s2q22q32q4   * sin(2*qd4+2*qd3+2*qd2);
s2q22q342q5  * sin(2*qd5+qd4+2*qd3+2*qd2);
s2q22q3m2q5s * sin(-2*qd5+2*qd3+2*qd2);
s2q22q3m2q4m2q5 * sin(-2*qd5-2*qd4+2*qd3+2*qd2);
s2q22q32q5   * sin(2*qd5+2*qd3+2*qd2);
s2q22q32q42q5 * sin(2*qd5+2*qd4+2*qd3+2*qd2);
s2q22q3m4m2q5 * sin(-2*qd5+qd4+2*qd3+2*qd2);
s2q22q3m2q42q5 * sin(2*qd5-2*qd4+2*qd3+2*qd2);
s2q5         * sin(2*qd5);
s2q4         * sin(2*qd4);
s2q22q3     * sin(2*qd3+2*qd2);
s2q22q3m5  * sin(-qd5+2*qd3+2*qd2);
s2q22q3m45 * sin(-qd5+qd4+2*qd3+2*qd2);
s2q22q3s   * sin(qd5+2*qd3+2*qd2);
s2q22q3m4s * sin(qd5+qd4+2*qd3+2*qd2);
s2q22q345  * sin(qd5+qd4+2*qd3+2*qd2);
s2q22q34s  * sin(qd5);
s2q2       * sin(2*qd2);
s23m4m2q5  * sin(-2*qd5+qd4+qd3+qd2);
s23m2q4m2q5 * sin(-2*qd5-2*qd4+qd3+qd2);

```

```

a23a2q5 = * sin(-2*qd5+qd3+qd2);
a23a2q4 = * sin(2*qd4+qd3+qd2);
a23a2a2q5 = * sin(-2*qd5+qd4+qd3+qd2);
a23a2q4a2q5 = * sin(2*qd5-2*qd4+qd3+qd2);
a23a42q5 = * sin(2*qd5+qd4+qd3+qd2);
a23a2q42q5 = * sin(2*qd5+2*qd4+qd3+qd2);
a23a4a5 = * sin(-qd5+qd2+qd3+qd4);
a23a5 = * sin(qd3+qd3+qd5);
a23a2q5 = * sin(2*qd5+qd4+qd3+qd2);
a23a2q5 = * sin(2*qd5+qd4+qd3+qd2);
a23a2q5 = * sin(qd2+qd3+qd4);
a23a2q4a2q5 = * sin(-2*qd5+2*qd4+qd3+qd2);
a23a2q4 = * sin(-2*qd4+qd3+qd2);
a23a = * sin(qd2+qd3+qd4);
a23 = * sin(qd2+qd3);
a23a5 = * sin(qd5+qd2+qd3+qd4);
a23a45 = * sin(qd5+qd4+qd3+qd2);
a235 = * sin(qd2+qd3+qd5);
a23a4a5 = * sin(-qd5+qd2+qd3+qd4);
a2a2q5 = * sin(-2*qd5+qd4);
a45 = * sin(qd4+qd5);
a4a5 = * sin(qd4+qd5);
a42q5 = * sin(2+qd5+qd4);
a4 = * sin(qd4);
a23 = * sin(qd2+qd3);
a23a45 = * sin(qd5+qd2+qd3+qd4);
a23a4a5 = * sin(-qd5+qd2+qd3+qd4);
a23a5 = * sin(qd5+qd2+qd3+qd4);
a23a4a5 = * sin(-qd5+qd2+qd3+qd4);
a23a5 = * sin(qd2+qd3+qd5);
a23a5 = * sin(qd2+qd3+qd5);

// Datos de la Matriz de Inercia
w1 = (0.0625*c2q2q5+0.0938*c2q2q3a2q5+0.0625*c2q2q3a2q5+0.0625*c2q2q3a2q5+0.0156*c2q2q3a2q5+0.0313*c2q2q3a2q5
-0.0625*c2q2q3a2q5+0.0156*c2q2q3a2q5+0.0156*c2q2q3a2q5+0.0625*c2q2q3a2q5+0.0156*c2q2q3a2q5+0.0313*c2q2q3a2q5);
w2 = (0.5*a345+0.25*a345+0.5*a2q3a5+0.5*a2q3a5+0.25*a345+0.25*a345+0.25*a345+0.5*a345+0.25*a2q3a5+0.25*a2q3a5);
w3 = (0.25*c2q2q3a2q5+0.25*c2q2q3a2q5+0.5*c2q2q3a2q5+0.5*c2q2q3a2q5+0.25*c2q2q3a2q5+0.25*c2q2q3a2q5);
w4 = (0.1250*c2q2q3a2q5+0.1250*c2q2q3+0.1875*c2q2q3a2q5+0.1250*c2q2q3a2q5+0.0313*c2q2q3a2q5+0.1250*c2q5
+0.1250*c2q2q3a2q5+0.1250*c2q4+0.6250+0.0625*c2q2q3a2q5+0.1250*c2q2q3a2q5+0.0625*c2q2q3a2q5+0.0625*c2q2q3a2q5
+0.0625*c2q42q5+0.0313*c2q2q3a2q5+0.1875*c2q2q3a2q5+0.0313*c2q2q3a2q5+0.0625*c2q4a2q5);
w5 = (a3+2a2q3);
w6 = (0.1875*c2q2q3a2q5+0.0625*c2q2q3a2q5+0.1875*c2q2q3a2q5+0.0313*c2q2q3a2q5+0.1250*c2q4+0.0313*c2q2q3a2q5
+0.0313*c2q2q3a2q5+0.1250*c2q2q3a2q5+0.0625*c2q2q3a2q5+0.1250*c2q2q3a2q5+0.1250*c2q5+0.1250*c2q2q3a2q5
-0.1250*c2q2q3a2q5+0.0625*c2q2q3+0.1250*c2q2q3a2q5+0.1250*c2q2q3a2q5+0.0313*c2q2q3a2q5+0.0625*c2q4a2q5);
w7 = (0.1250*c2q2q3a2q5+0.1250*c2q2q3a2q5+0.25*c2q2q3a2q5+0.25*c2q2q3a2q5+0.75+0.25*c2q4);
w8 = (0.1250*c2q2q3a2q5+0.25*c2q2q3a2q5+0.25*c2q2q3a2q5+0.1250*c2q2q3a2q5+0.75);
w9 = (0.5*c2q2q3+0.5);
w10 = (0.5*c2q2q3+0.5);
w11 = (0.5*c2q2q3);
w12 = (0.5*c2q2q3);
w13 = (0.0625*c23a2q5+0.0625*c23a2q5+0.0625*c23a2q5+0.0313*c23a2q5+0.0625*c23a2q5+0.0625*c23a2q5+0.0313*c23a2q5
+0.0313*c23a2q5+0.0625*c23a2q5+0.0313*c23a2q5);
w14 = (0.25*a2a5+0.25*a2a5+0.25*a2a5+0.25*a2a5);
w15 = (0.25*c23a4a5+0.25*c23a4a5+0.25*c23a4a5+0.25*c23a4a5);
w16 = (0.1250*c23a2q4+0.0625*c23a2q4+0.0625*c23a2q4+0.0625*c23a2q4+0.0625*c23a2q4+0.1250*c23a2q5+0.1250*c23a2q5
+0.1250*c23a2q5+0.0625*c23a2q4+0.1250*c23a2q4+0.1250*c23a2q4+0.1250*c23a2q4);
w17 = (0.0625*c23a2q4+0.0625*c23a2q4+0.1250*c23a2q4+0.1250*c23a2q4+0.1250*c23a2q4+0.1250*c23a2q4+0.0625*c23a2q4);
w18 = (0.25*c23a2q4+0.25*c23a2q4);
w19 = (0.25*c23a2q4+0.25*c23a2q4);
w20 = (0.0625*c23a2q5+0.0625*c23a2q5+0.0625*c23a2q5+0.0625*c23a2q5+0.1250*c23a2q5+0.1250*c23a2q5+0.75*c23);
w21 = (0.25*c23a5+0.25*c23a5+0.25*c23a5+0.25*c23a5);
w22 = (0.25*c23a5+0.25*c23a5+0.25*c23a5+0.25*c23a5);
w23 = (0.1250*c23a2q5+0.1250*c23a2q5+0.1250*c23a2q5+0.25*c23a2q5+0.25*c23a2q5+0.1250*c23a2q5+0.5*c23);
w24 = (0.5*c23+0.1250*c23a2q5+0.1250*c23a2q5+0.25*c23a2q5+0.1250*c23a2q5+0.1250*c23a2q5+0.25*c23a2q5);
w25 = (0.25*a2a5+0.25*a2a5+0.25*a2a5+0.25*a2a5);
w26 = (0.25*c23a+0.25*c23a);
w27 = (0.25*c23a5+0.25*c23a5+0.25*c23a5+0.25*c23a5);
w28 = (0.5*c23a+0.5*c23a);
w29 = (0.5*c23a5+0.5*c23a5+0.25*c23a5+0.25*c23a5+0.25*c23a5);
w30 = (0.1250*c2q5+0.1250*c2q4+0.6250+0.0625*c2q4a2q5+0.0625*c2q4a2q5);
w31 = (0.5*a345+0.5*a345+0.5*a345+0.5*a345+0.5*a345);
w32 = (0.25*c2q4+0.1250*c2q4a2q5+0.75+0.1250*c2q4a2q5+0.25*c2q5);
w33 = (0.1250*c2q4a2q5+0.75+0.25*c2q5+0.1250*c2q4a2q5+0.25*c2q4);
w34 = (0.5*c2q4+0.5);

```

TESIS CON
FALLA DE ORIGEN


```

Ya[1][13] = c2;
Ya[1][14] = s22;
Ya[1][15] = 0;
Ya[1][16] = qrrp2; // Terminos de los motores
Ya[1][17] = 0;
Ya[1][18] = 0;
Ya[1][19] = 0;
Ya[1][20] = 0;
Ya[1][21] = 0.0;
Ya[1][22] = a1gnq2*(qrp2);
Ya[1][23] = 0.0;
Ya[1][24] = 0.0;
Ya[1][25] = 0.0;
Ya[1][26] = 0.0;
Ya[1][27] = 0.0; // Friccion viscosa
Ya[1][28] = qrp2;
Ya[1][29] = 0.0;
Ya[1][30] = 0.0;
Ya[1][31] = 0.0;
Ya[1][32] = 0.0;

Ya[2][0] = w13*qrp1+w30*qrrp2+w30*qrrp3+w37*qrrp4+0.5*c4*qrrp5+w44*qrrp6+w86*qrpl+w94*qrp2+w94*qrp3
          +w104*qrp4+w111*qrp5+w116*qrp6;
Ya[2][1] = (w36*qrrp2+w117*qrpl+w119*qrp2)-1;
Ya[2][2] = (w15*qrpl+2*c5*qrrp2+2*c5*qrpp3+w39*qrrp4+w45*qrpp5+w88*qrpl-w98*qrp2-w98*qrp3+w106*qrp4+w113*qrp5)-1;
Ya[2][3] = w16*qrpp1+w32*qrpp2+w32*qrpp3+w40*qrpp4+c4*qrpp5+w89*qrpl+w96*qrp2+w98*qrp3+w107*qrp4+w114*qrp5;
Ya[2][4] = c3*qrpp4+w115*qrpl+w120*qrp2;
Ya[2][5] = w17*qrpp1+w33*qrpp2+w33*qrpp3+w41*qrpp4+c6*qrpp5+w90*qrpl+w97*qrp2+w97*qrp3+w108*qrp4+w115*qrp5;
Ya[2][6] = w18*qrpp1+w34*qrpp2+w34*qrpp3+w91*qrpl-w100*qrp2-w100*qrp3+w109*qrp4;
Ya[2][7] = w19*qrpp1+w35*qrpp2+w35*qrpp3+w92*qrpl+w100*qrp2+w100*qrp3+w110*qrp4;
Ya[2][8] = qrrp2*qrpp3-w70*qrpl;
Ya[2][9] = rrrp2*qrpp3-w70*qrpl;
Ya[2][10] = 0;
Ya[2][11] = 0;
Ya[2][12] = w41;
Ya[2][13] = 0;
Ya[2][14] = s23;
Ya[2][15] = 0; // Terminos de los motores;
Ya[2][16] = 0;
Ya[2][17] = qrrp3;
Ya[2][18] = 0;
Ya[2][19] = 0;
Ya[2][20] = 0;
Ya[2][21] = 0.0; // Friccion de Coulomb
Ya[2][22] = 0.0;
Ya[2][23] = a1gnq3*(qrp3);
Ya[2][24] = 0.0;
Ya[2][25] = 0.0;
Ya[2][26] = 0.0;
Ya[2][27] = 0.0; // Friccion viscosa
Ya[2][28] = 0.0;
Ya[2][29] = qrp3;
Ya[2][30] = 0.0;
Ya[2][31] = 0.0;
Ya[2][32] = 0.0;

Ya[3][0] = w20*qrpp1+w37*qrpp2+w37*qrpp3+w46*qrpp4+c6*qrpp6+w121*qrpl+w126*qrp2+w126*qrp3+w133*qrp4+w135*qrp5
          +w138*qrp6;
Ya[3][1] = (w21*qrpp1+w38*qrpp2+w122*qrpl+w127*qrp2)-1;
Ya[3][2] = (w22*qrpp1+w39*qrpp2+w39*qrpp3+w123*qrpl+w128*qrp2+w128*qrp3)-1;
Ya[3][3] = w23*qrpp1+w40*qrpp2+w40*qrpp3+w47*qrpp4+w124*qrpl+w129*qrp2+w129*qrp3+w134*qrp4+w136*qrp5;
Ya[3][4] = 0;
Ya[3][5] = w24*qrpp1+w41*qrpp2+w41*qrpp3+w48*qrpp4+w125*qrpl+w130*qrp2+w130*qrp3+w134*qrp4+w137*qrp5;
Ya[3][6] = -c23*qrpp1*qrpp4+w79*qrpl+w131*qrp2+w131*qrp3;
Ya[3][7] = -c23*qrpp1*qrpp4+w78*qrpl+w132*qrp2+w132*qrp3;
Ya[3][8] = 0;
Ya[3][9] = 0;
Ya[3][10] = 0;
Ya[3][11] = 0;
Ya[3][12] = -wg2;
Ya[3][13] = 0.0;
Ya[3][14] = 0;
Ya[3][15] = 0; // Terminos de los motores
Ya[3][16] = 0;
Ya[3][17] = 0;
Ya[3][18] = qrrp4;
Ya[3][19] = 0;

```

TESIS CON
FALLA DE ORIGEN

```

Ya(3)(20) = 0;
Ya(3)(21) = 0.0; // Friccion de Coulomb
Ya(3)(22) = 0.0;
Ya(3)(23) = 0.0;
Ya(3)(24) = signq4+(qrp4);
Ya(3)(25) = 0.0;
Ya(3)(26) = 0.0;
Ya(3)(27) = 0.0; // Friccion viscosa
Ya(3)(28) = 0.0;
Ya(3)(29) = 0.0;
Ya(3)(30) = qrp4;
Ya(3)(31) = 0.0;
Ya(3)(32) = 0.0;

Ya(4)(0) = w26+qrppl+0.5*c4+qrp2+0.5*c4+qrp3+0.5*qrp5+w139+qrp1+w142+qrp2+w142+qrp3+w147+qrp4+w150+qrp6;
Ya(4)(1) = (w27+qrppl+w42+qrp2+w140+qrp1+w143+qrp2)*-1;
Ya(4)(2) = (w27+qrppl+w43+qrp2+w45+qrp3+w141+qrp1+w144+qrp2+w144+qrp3)*-1;
Ya(4)(3) = w28+qrppl+c4+qrp2+c4+qrp3+qrp5+w84+qrp1+w145+qrp2+w145+qrp3+w148+qrp4;
Ya(4)(4) = 0;
Ya(4)(5) = w28+qrppl+c4+qrp2+c4+qrp3+qrp5+w83+qrp1+w146+qrp2+w146+qrp3+w149+qrp4;
Ya(4)(6) = 0;
Ya(4)(7) = 0;
Ya(4)(8) = 0;
Ya(4)(9) = 0;
Ya(4)(10) = 0;
Ya(4)(11) = 0; //Terminos de gravedad
Ya(4)(12) = wg;
Ya(4)(13) = 0;
Ya(4)(14) = 0;
Ya(4)(15) = 0; // Terminos de los motores
Ya(4)(16) = 0;
Ya(4)(17) = 0;
Ya(4)(18) = 0;
Ya(4)(19) = qrp5;
Ya(4)(20) = 0;
Ya(4)(21) = 0.0; // Friccion de Coulomb
Ya(4)(22) = 0.0;
Ya(4)(23) = 0.0;
Ya(4)(24) = 0.0;
Ya(4)(25) = signq5+(qrp5);
Ya(4)(26) = 0.0;
Ya(4)(27) = 0.0; // Friccion viscosa
Ya(4)(28) = 0.0;
Ya(4)(29) = 0.0;
Ya(4)(30) = 0.0;
Ya(4)(31) = qrp5;
Ya(4)(32) = 0.0;

Ya(5)(0) = w29+qrppl+w44+qrp2+w44+qrp3+c5+qrp4+qrp5+w85+qrp1+w161+qrp2+w161+qrp3+w152+qrp4+w153+qrp6;
Ya(5)(1) = 0;
Ya(5)(2) = 0;
Ya(5)(3) = 0;
Ya(5)(4) = 0;
Ya(5)(5) = 0;
Ya(5)(6) = 0;
Ya(5)(7) = 0;
Ya(5)(8) = 0;
Ya(5)(9) = 0;
Ya(5)(10) = 0;
Ya(5)(11) = 0;
Ya(5)(12) = 0; //Terminos de gravedad
Ya(5)(13) = 0;
Ya(5)(14) = 0;
Ya(5)(15) = 0; // Terminos de los motores
Ya(5)(16) = 0;
Ya(5)(17) = 0;
Ya(5)(18) = 0;
Ya(5)(19) = 0;
Ya(5)(20) = qrp6;
Ya(5)(21) = 0.0; // Friccion de Coulomb
Ya(5)(22) = 0.0;
Ya(5)(23) = 0.0;
Ya(5)(24) = 0.0;
Ya(5)(25) = 0.0;
Ya(5)(26) = signq6+(qrp6);
Ya(5)(27) = 0.0; // Friccion viscosa
Ya(5)(28) = 0.0;

```

TESIS CON FALLA DE ORIGEN

```

Yaf5[29] = 0.0;
Yaf5[30] = 0.0;
Yaf5[31] = 0.0;
Yaf5[32] = qrp6;

```

```

// ***** Ley Adaptable *****

```

```

Transpose (Ya,6,33,Yatp);

```

```

gam1 = 0.00001 +1.0. // Matriz de Inercia 1
gam2 = 0.00001 +1.0. // Matriz de Inercia 2
gam3 = 0.00001 +1.0. // Matriz de Inercia 3
gam4 = 0.00001 +1.0. // Matriz de Inercia 4
gam5 = 0.00001 +1.0. // Matriz de Inercia 5
gam6 = 0.00001 +1.0. // Matriz de Inercia 6
gam7 = 0.00001 +1.0. // Matriz de Inercia 7
gam8 = 0.00001 +1.0. // Matriz de Inercia 8
gam9 = 0.00001 +1.0. // Matriz de Inercia 9
gam10 = 0.00001 +1.0. // Matriz de Inercia 10
gam11 = 0.00001 +1.0. // Matriz de Inercia 11
gam12 = 0.00001 +1.0. // Matriz de Inercia 12

```

```

gam13 = 2.0 +1.0. // Gravitacional
gam14 = 2.0 +1.0. // Gravitacional
gam15 = 2.0 +1.0. // Gravitacional

```

```

gam16 = 0.15 +1.0. // Friccion de Coulomb 1
gam17 = 0.15 +1.0. // Friccion de Coulomb 2
gam18 = 0.19 +1.0. // Friccion de Coulomb 3
gam19 = 0.0015 +1.0. // Friccion de Coulomb 4
gam20 = 0.0015 +1.0. // Friccion de Coulomb 5
gam21 = 0.0015 +1.0. // Friccion de Coulomb 6

```

```

gam22 = 0.5 +1.0. // Motores 1
gam23 = 0.5 +1.0. // Motores 2
gam24 = 0.5 +1.0. // Motores 3
gam25 = 0.01 +1.0. // Motores 4
gam26 = 0.02 +1.0. // Motores 5
gam27 = 0.05 +1.0. // Motores 6

```

```

gam28 = 0.5 +1.0. // Friccion Viscosa 1
gam29 = 0.5 +1.0. // Friccion Viscosa 2
gam30 = 0.5 +1.0. // Friccion Viscosa 3
gam31 = 0.01 +1.0. // Friccion Viscosa 4
gam32 = 0.02 +1.0. // Friccion Viscosa 5
gam33 = 0.01 +1.0. // Friccion Viscosa 6

```

```

Casma[0][0] = gam1; // Parametro de la matriz de inercia y de coriolis
Casma[1][1] = gam2;
Casma[2][2] = gam3;
Casma[3][3] = gam4;
Casma[4][4] = gam5;
Casma[5][5] = gam6;
Casma[6][6] = gam7;
Casma[7][7] = gam8;
Casma[8][8] = gam9;
Casma[9][9] = gam10;
Casma[10][10] = gam11;
Casma[11][11] = gam12;
Casma[12][12] = gam13; // Gravedad
Casma[13][13] = gam14;
Casma[14][14] = gam15; // Friccion Coulomb
Casma[15][15] = gam16;
Casma[16][16] = gam17;
Casma[17][17] = gam18;
Casma[18][18] = gam19;
Casma[19][19] = gam20; // Motores
Casma[20][20] = gam21;
Casma[21][21] = gam22;
Casma[22][22] = gam23;
Casma[23][23] = gam24;
Casma[24][24] = gam25; // Friccion Viscosa
Casma[25][25] = gam26;
Casma[26][26] = gam27;
Casma[27][27] = gam28;
Casma[28][28] = gam29;

```



```
Gamma[29][29] = -gam30;
Gamma[30][30] = -gam31;
Gamma[31][31] = -gam32;
Gamma[32][32] = -gam33;
```

```
MatrixMul(Gamma, Yexp, 33, 33, 6, mult1);
```

```
MatrixMul(mult1, aa, 33, 6, 1, agp);
```

```
agp11 = agp[01][01];
agp21 = agp[11][01];
agp31 = agp[21][01];
agp41 = agp[31][01];
agp51 = agp[41][01];
agp61 = agp[51][01];
agp71 = agp[61][01];
agp81 = agp[71][01];
agp91 = agp[81][01];
agp101 = agp[91][01]; //Cantidades muy pequeñas
agp111 = agp[101][01];
agp121 = agp[111][01];
agp131 = agp[121][01];
agp141 = agp[131][01];
agp151 = agp[141][01];
agp161 = agp[151][01];
agp171 = agp[161][01];
agp181 = agp[171][01];
agp191 = agp[181][01];
agp201 = agp[191][01];
agp211 = agp[201][01];
agp221 = agp[211][01];
agp231 = agp[221][01];
agp241 = agp[231][01];
agp251 = agp[241][01];
agp261 = agp[251][01];
agp271 = agp[261][01];
agp281 = agp[271][01];
agp291 = agp[281][01];
agp301 = agp[291][01];
agp311 = agp[301][01];
agp321 = agp[311][01];
agp331 = agp[321][01];
```

TESIS CON
FALLA DE ORIGEN

```
// printf("%d %f %f %f %f %f %f\n", i, agp[16][0], agp[17][0], agp[18][0], agp[19][0], agp[20][0], agp[21][0]);
```

```
// Integracion Metodo de Trapezoidal (Garantiza la estabilidad del sistema sin importar el Tiempo de Muestreo)
```

```
ag11 = ag11_1 + (hd2 * agp11) + (hd2 * agp11_1);
ag21 = ag21_1 + (hd2 * agp21) + (hd2 * agp21_1);
ag31 = ag31_1 + (hd2 * agp31) + (hd2 * agp31_1);
ag41 = ag41_1 + (hd2 * agp41) + (hd2 * agp41_1);
ag51 = ag51_1 + (hd2 * agp51) + (hd2 * agp51_1);
ag61 = ag61_1 + (hd2 * agp61) + (hd2 * agp61_1);
ag71 = ag71_1 + (hd2 * agp71) + (hd2 * agp71_1);
ag81 = ag81_1 + (hd2 * agp81) + (hd2 * agp81_1);
ag91 = ag91_1 + (hd2 * agp91) + (hd2 * agp91_1);
ag101 = ag101_1 + (hd2 * agp101) + (hd2 * agp101_1);
ag111 = ag111_1 + (hd2 * agp111) + (hd2 * agp111_1);
ag121 = ag121_1 + (hd2 * agp121) + (hd2 * agp121_1);
ag131 = ag131_1 + (hd2 * agp131) + (hd2 * agp131_1);
ag141 = ag141_1 + (hd2 * agp141) + (hd2 * agp141_1);
ag151 = ag151_1 + (hd2 * agp151) + (hd2 * agp151_1);
ag161 = ag161_1 + (hd2 * agp161) + (hd2 * agp161_1);
ag171 = ag171_1 + (hd2 * agp171) + (hd2 * agp171_1);
ag181 = ag181_1 + (hd2 * agp181) + (hd2 * agp181_1);
ag191 = ag191_1 + (hd2 * agp191) + (hd2 * agp191_1);
ag201 = ag201_1 + (hd2 * agp201) + (hd2 * agp201_1);
ag211 = ag211_1 + (hd2 * agp211) + (hd2 * agp211_1);
ag221 = ag221_1 + (hd2 * agp221) + (hd2 * agp221_1);
ag231 = ag231_1 + (hd2 * agp231) + (hd2 * agp231_1);
ag241 = ag241_1 + (hd2 * agp241) + (hd2 * agp241_1);
ag251 = ag251_1 + (hd2 * agp251) + (hd2 * agp251_1);
ag261 = ag261_1 + (hd2 * agp261) + (hd2 * agp261_1);
ag271 = ag271_1 + (hd2 * agp271) + (hd2 * agp271_1);
ag281 = ag281_1 + (hd2 * agp281) + (hd2 * agp281_1);
```

```

ag291 = ag291.1 + ((hd2 = agp291) * (hd2 = agp291.1));
ag301 = ag301.1 + ((hd2 = agp301) * (hd2 = agp301.1));
ag311 = ag311.1 + ((hd2 = agp311) * (hd2 = agp311.1));
ag321 = ag321.1 + ((hd2 = agp321) * (hd2 = agp321.1));
ag331 = ag331.1 + ((hd2 = agp331) * (hd2 = agp331.1));

```

```
if ((i==intern2+1) && (suustran-1>chance)){
```

```

    ag1[cont2] = ag1;
    ag2[cont2] = ag2;
    ag3[cont2] = ag3;
    ag4[cont2] = ag4;
    ag5[cont2] = ag5;
    ag6[cont2] = ag6;
    ag7[cont2] = ag7;
    ag8[cont2] = ag8;
    ag9[cont2] = ag9;
    ag10[cont2] = ag10;
    ag11[cont2] = ag11;
    ag12[cont2] = ag12;
    ag13[cont2] = ag13;
    ag14[cont2] = ag14;
    ag15[cont2] = ag15;
    ag16[cont2] = ag16;
    ag17[cont2] = ag17;
    ag18[cont2] = ag18;
    ag19[cont2] = ag19;
    ag20[cont2] = ag20;
    ag21[cont2] = ag21;
    ag22[cont2] = ag22;
    ag23[cont2] = ag23;
    ag24[cont2] = ag24;
    ag25[cont2] = ag25;
    ag26[cont2] = ag26;
    ag27[cont2] = ag27;
    ag28[cont2] = ag28;
    ag29[cont2] = ag29;
    ag30[cont2] = ag30;
    ag31[cont2] = ag31;
    ag32[cont2] = ag32;
    ag33[cont2] = ag33;

```

```

    cont2 = cont2 + 1;
    intern2 = intern2 + almecane;
}

```

```

ag11.1 = ag1;
ag21.1 = ag2;
ag31.1 = ag3;
ag41.1 = ag4;
ag51.1 = ag5;
ag61.1 = ag6;
ag71.1 = ag7;
ag81.1 = ag8;
ag91.1 = ag9;
ag101.1 = ag10;
ag111.1 = ag11;
ag121.1 = ag12;
ag131.1 = ag13;
ag141.1 = ag14;
ag151.1 = ag15;
ag161.1 = ag16;
ag171.1 = ag17;
ag181.1 = ag18;
ag191.1 = ag19;
ag201.1 = ag20;
ag211.1 = ag21;
ag221.1 = ag22;
ag231.1 = ag23;
ag241.1 = ag24;
ag251.1 = ag25;
ag261.1 = ag26;
ag271.1 = ag27;
ag281.1 = ag28;
ag291.1 = ag29;
ag301.1 = ag30;
ag311.1 = ag31;

```

TESIS CON
 FALLA DE ORIGEN

```
ag321.1 = ag321;
ag331.1 = ag331;
```

```
// asignacion del valor anterior de parametros punto
```

```
agp11.1 = agp11;
agp21.1 = agp21;
agp31.1 = agp31;
agp41.1 = agp41;
agp51.1 = agp51;
agp61.1 = agp61;
agp71.1 = agp71;
agp81.1 = agp81;
agp91.1 = agp91;
agp101.1 = agp101;
agp111.1 = agp111;
agp121.1 = agp121;
agp131.1 = agp131;
agp141.1 = agp141;
agp151.1 = agp151;
agp161.1 = agp161;
agp171.1 = agp171;
agp181.1 = agp181;
agp191.1 = agp191;
agp201.1 = agp201;
agp211.1 = agp211;
agp221.1 = agp221;
agp231.1 = agp231;
agp241.1 = agp241;
agp251.1 = agp251;
agp261.1 = agp261;
agp271.1 = agp271;
agp281.1 = agp281;
agp291.1 = agp291;
agp301.1 = agp301;
agp311.1 = agp311;
agp321.1 = agp321;
agp331.1 = agp331;
```

**TESIS CON
FALLA DE ORIGEN**

```
ag[0][0] = ag11;
ag[1][0] = ag21;
ag[2][0] = ag31;
ag[3][0] = ag41;
ag[4][0] = ag51;
ag[5][0] = ag61;
ag[6][0] = ag71;
ag[7][0] = ag81;
ag[8][0] = ag91;
ag[9][0] = ag101; // Asignacion de variables sin filtrar
ag[10][0] = ag111;
ag[11][0] = ag121;
ag[12][0] = ag131;
ag[13][0] = ag141;
ag[14][0] = ag151;
ag[15][0] = ag161;
ag[16][0] = ag171;
ag[17][0] = ag181;
ag[18][0] = ag191;
ag[19][0] = ag201;
ag[20][0] = ag211;
ag[21][0] = ag221;
ag[22][0] = ag231;
ag[23][0] = ag241;
ag[24][0] = ag251;
ag[25][0] = ag261;
ag[26][0] = ag271;
ag[27][0] = ag281;
ag[28][0] = ag291;
ag[29][0] = ag301;
ag[30][0] = ag311;
ag[31][0] = ag321;
ag[32][0] = ag331;
```

```

// ..... Ley de Control .....
MatrizMul(Ya,ag,6,33,1,leyc), // Maximo valor que toma es de 1:5 mas-menos
MatrizMul(DK_1Dn,leyc,6,6,1,leyc1); // Multiplica por la dinamica de los motores
// ..... Cananacia .....
mkd[0][0] = -kr1,
mkd[1][1] = -kr2,
mkd[2][2] = -kr3,
mkd[3][3] = -kr4,
mkd[4][4] = -kr5,
mkd[5][5] = -kr6,

MatrizMul(mkd,ev,6,6,1,gankd);

MatrizMul(DK_1Dn,gankd,6,6,1,leyc2); // Multiplica por la dinamica de los motores
// ..... Modificacion .....
mkp[0][0] = -kp1,
mkp[1][1] = -kp2,
mkp[2][2] = -kp3,
mkp[3][3] = -kp4,
mkp[4][4] = -kp5,
mkp[5][5] = -kp6,

MatrizMul(mkp,ep,6,6,1,modif),

MatrizMul(DK_1Dn,modif,6,6,1,leyc3); // Multiplica por la dinamica de los motores
// ..... Fin de la Modificacion .....
Add2D (leyc1,leyc2,6,1,volt); // Maximo valor que toma es de 100 mas-menos
Add2D (volt,leyc3,6,1,volt); // Modificacion de la ley de control -kpqt

volt[1] = volta[0][0];
volt[2] = volta[1][0];
volt[3] = volta[2][0]; // Vector sólo para graficar señal sin filtrar
volt[4] = volta[3][0];
volt[5] = volta[4][0];
volt[6] = volta[5][0];

if (volta[0][0] >= sat)
    volta[0][0] = sat,
else
    if (volta[0][0] <= -sat)
        volta[0][0] = -sat,
    else
        volta[0][0] = volta[0][0],

if (volta[1][0] >= sat)
    volta[1][0] = sat,
else
    if (volta[1][0] <= -sat)
        volta[1][0] = -sat,
    else
        volta[1][0] = volta[1][0];

if (volta[2][0] >= sat)
    volta[2][0] = sat,
else
    if (volta[2][0] <= -sat)
        volta[2][0] = -sat,
    else
        volta[2][0] = volta[2][0];

if (volta[3][0] >= sat)
    volta[3][0] = sat,
else
    if (volta[3][0] <= -sat)
        volta[3][0] = -sat;

```

TESIS CON
FALLA DE ORIGEN

**TESIS CON
FALLA DE ORIGEN**

```

else
    volts[3][0] = volts[3][0];

if (volts[4][0] >= sat)
    volts[4][0] = sat;
else
    if (volts[4][0] <= -sat)
        volts[4][0] = -sat;
    else
        volts[4][0] = volts[4][0];

if (volts[5][0] >= sat)
    volts[5][0] = sat;
else
    if (volts[5][0] <= -sat)
        volts[5][0] = -sat;
    else
        volts[5][0] = volts[5][0];

volts[0][0] = volts[0][0] * 3276.8;
volts[1][0] = volts[1][0] * 3276.8;
volts[2][0] = volts[2][0] * 3276.8;
volts[3][0] = volts[3][0] * 3276.8;
volts[4][0] = volts[4][0] * 3276.8;
volts[5][0] = volts[5][0] * 3276.8;
// printf("%d %f %f %f %f\n", i, tau[0][0], tau[1][0], tau[2][0], tau[3][0], tau[4][0]);

status = file_load_dac (boardID2, 0x31, volts[0][0], 0xFF); //Salida de datos por el DAC
status = file_load_dac (boardID2, 0x32, volts[1][0], 0xFF); //Salida de datos por el DAC
status = file_load_dac (boardID2, 0x33, -volts[2][0], 0xFF); //Salida de datos por el DAC
status = file_load_dac (boardID2, 0x34, -volts[3][0], 0xFF); //Salida de datos por el DAC
status = file_load_dac (boardID2, 0x35, -volts[4][0], 0xFF); //Salida de datos por el DAC
status = file_load_dac (boardID2, 0x36, -volts[5][0], 0xFF); //Salida de datos por el DAC

// Tarjeta controlador y adaptación
tiempof = Timer();
tiempomil = tiempof-tiempoi;
diff = tiempof - tiempoi;

if (diff < h){
    dly = h - diff;
    Delay(dly - 0.001101);
}
diff = (tiempof+dly) - tiempoi;
dly = 0;

Tdelay[i] = diff;

if ((muestras-1)<chance) { alerta=chance; }

SetCtrlVal (panelHandle, PANEL_tiempon, alerta);

i=i+1;
fin = 1;
ProcessSystemEvents();
} // fin del While

//finaliza esquema de adaptación, control y observador
//***** ROBMA L2 *****
t[fin-1] = fint; // Quitar el -1 para el bloqueo
tt[cont] = fint - 0.001101;

L22 = TL22/T;
L2 = sqrt(L22); // Calculo de la norma del error de todo el robot para T segundos
L21 = TL21/T;
L1 = sqrt(L21); // Calculo de la norma del error para la primera articulación
L222 = TL222/T;
L2 = sqrt(L222); // Calculo de la norma del error para T segundos
L223 = TL223/T;
L3 = sqrt(L223); // Calculo de la norma del error para T segundos
L224 = TL224/T;
L4 = sqrt(L224); // Calculo de la norma del error para T segundos
L225 = TL225/T;
L5 = sqrt(L225); // Calculo de la norma del error para T segundos

```

```
L226 = TL226/T;
La9 = sqrt(L226); // Calculo de la norma del error para T segundos
L221 = TL221/T1;
L21 = sqrt(L221); // Calculo de la norma del error para T segundos
L221a = TL221a/T1;
La1 = sqrt(L221a); // Calculo de la norma del error para T segundos
L222 = TL222/T1;
La2 = sqrt(L222); // Calculo de la norma del error para T segundos
L223 = TL223/T1;
La3 = sqrt(L223); // Calculo de la norma del error para T segundos
L224 = TL224/T1;
La4 = sqrt(L224); // Calculo de la norma del error para T segundos
L225 = TL225/T1;
La5 = sqrt(L225); // Calculo de la norma del error para T segundos
L226 = TL226/T1;
La6 = sqrt(L226); // Calculo de la norma del error para T segundos
L221a = TL221a/T1a;
L21a = sqrt(L221a); // Calculo de la norma del error para T segundos
L221aa = TL221aa/T1aa;
La1aa = sqrt(L221aa); // Calculo de la norma del error para T segundos
L222a = TL222a/T1a;
La2a = sqrt(L222a); // Calculo de la norma del error para T segundos
L223a = TL223a/T1a;
La3a = sqrt(L223a); // Calculo de la norma del error para T segundos
L224a = TL224a/T1a;
La4a = sqrt(L224a); // Calculo de la norma del error para T segundos
L225a = TL225a/T1a;
La5a = sqrt(L225a); // Calculo de la norma del error para T segundos
L226a = TL226a/T1a;
La6a = sqrt(L226a); // Calculo de la norma del error para T segundos
}
break;
}
return 0;
}
```

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN

Bibliografía

Arteaga Pérez, M. A. 1998. On the properties of a dynamic model of flexible robot manipulators. *ASME Journal of Dynamic Systems, Measurement, and Control* 120. 8-14.

Arteaga Pérez, M. A. 2001. Robust control of robots by using a linear observer. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics AIM '01*. 571-576.

Arteaga Pérez, M. A. 2002. Robot control and parameter estimation with only joint position measurements. Submitted in *Automatica*.

Åström, K. J. y B. Wittenmark. 1990. *Computer-Controlled Systems*. New Jersey: Prentice-Hall.

Barrientos, A., F. Peñin, C. Balaguer y R. Aracil. 1997. *Conceptos Básicos de Robótica*. México, D. F.: McGraw-Hill.

Basañez B., L. 1999. Actualidad y perspectivas de la robótica [en línea]. *Producción Mecánica*. Mayo 1999, numero = 2. <<http://www.metalunivers.com/1pm/Pm02/Robotica.html>> [Consulta: 25 enero 2002].

Berghuis, H. y H. Nijmeijer. 1994. Robust control of robots via linear estimated state feedback. *IEEE Transactions on Automatic Control* 39(10). 2159-2162.

Burg, T., D. Dawson y P. Vedagarbha. 1997. A redesigned dcal controller without velocity measurements: theory and demonstration. *Robotica* 15. 337-346.

- Castillo Sánchez, A. M. 2002. *Adaptación de Dos Robots Industriales para su Utilización en el Desarrollo de Nuevas Técnicas y Algoritmos de Control*. Tesis de Licenciatura. ENEP Aragon, Universidad Nacional Autónoma de México.
- CRS, R. C. 1997. *Service Manual*. CRS, Robotics Corporation. Ontario. Canada.
- Fu, K. S., R. C. Gonzalez y C. S. G. Lee. 1989. *Robotics: Control, Sensing, Vision and Intelligence*. México, D. F.: McGraw-Hill.
- Gudiño Lau, J. y M. A. Arteaga Pérez. 2000. daptation of an industrial robot for use in research (Adaptación de un robot industrial para su utilización en investigación).. *AMRob Memorias del 2do Congreso de Robótica 2000*. Toluca, México. 94-99.
- Instruments, N. 1999. *Universal motion Interface (UMI) Accesory*. National Instruments. USA.
- Khalil, H. K. 1996. *Nonlinear Systems, 2nd ed.*. Englewood Cliffs, New Jersey: Prentice-Hall.
- Martínez, J. C. 2002. Parameters identification of an industrial robot with different adaptive schemes. Universidad Veracruzana, México.
- Mills, J., P. Baines, T. Chang, S. Chew, T. Jones, S. Lam y A. Raba-di. 1995. Development of a robot control test platform. *IEEE Robotics and Automation Magazine*.
- Reyes, F. y R. Kelly. 2001. Experimental evaluation of model-based controllers on a direct-drive robot arm. *Mechatronics* 11. 267-282.
- Spong, M. W. y M. Vidyasagar. 1989. *Robot Dynamics and Control*. New York: John Wiley & Sons.
-