

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Ingeniería de Sistemas y Automática



TESIS DOCTORAL

**ESTUDIO E IMPLEMENTACIÓN DE
CONTROLADORES DINÁMICOS DE ROBOTS
INDUSTRIALES EN TIEMPO REAL**

Nuria Rosillo Guerrero

València, 2009

TESIS DOCTORAL

**ESTUDIO E IMPLEMENTACIÓN DE
CONTROLADORES DINÁMICOS DE ROBOTS
INDUSTRIALES EN TIEMPO REAL**

Por
Nuria Rosillo Guerrero

Ingeniera Informática de la
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Presentada en la
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Para la obtención del grado de
DOCTOR INGENIERO INFORMÁTICA

Directores de Tesis
Dr. D. Ángel Valera Fernández
Dr. D. Vicente Mata Amela

València, 2009

A mis padres José y Carmen por todo

Agradecimientos

Con este trabajo se culmina un período importante en mi carrera profesional y académica, período que no hubiera acabado de no ser por la colaboración de ciertas personas a las que pretendo enviar mi más sincero agradecimiento en estas líneas.

En primer lugar, por su gran implicación en esta tesis, a mis directores Ángel Valera y Vicente Mata, sin cuya intervención, ayuda, consejo, apoyo y comprensión ante ausencias raras no lo hubiera conseguido. También a la gente del DIMM, que me acogieron como a uno más. En especial a Francesc Benimeli por su ayuda en mil y una cosas, y al resto de becarios y doctorandos que han pasado por el laboratorio, también a Toni Besa por su ayuda con los parámetros geométricos y a Josep Lluís Suñer por echarme un cable con la generación de trayectorias y todo el apoyo prestado.

En un término profesional, también quiero dar las gracias a D. Fernando Sánchez, director de la Escuela Superior de Enseñanzas Técnicas y a D. Javier Muñoz director del Departamento de Ciencias Físicas, Matemáticas y de la Computación de la Universidad Cardenal Herrera-CEU por su apoyo y ánimos, así como a todos mis compañeros que me han dado aliento y han tenido siempre una palabra de ánimo. Muchas gracias.

Finalmente, en el lugar más especial quiero darles las gracias a mis padres José y Carmen, por todo lo que me han dado y me siguen dando. También a mis hermanos, Víctor y Xavier así como a Macu y Manuel por ser tan especiales y estar a mi lado incondicionalmente, a mi sobrina Carlota por su sonrisa y a Blanca quien es un miembro más de mi familia. Y no me quiero olvidar de mis amigos, Lola, David L, Tere y Victor (y bolita), Toni y Carol (por supuesto a Héctor), David V. y Paula, y a Carlos y al resto de personas que forman parte de mi vida por sus ánimos (Boomerang, Boomerang va por vosotros).

Espero no haber olvidado a nadie. Mi más sincera disculpa a quien se sienta omitid@. Gracias a tod@s.

Resumen

El control de robots manipuladores tiene por objeto mejorar la calidad, productividad y funcionalidad de dichos sistemas industriales. Para ello, se hace necesario que el diseño de los métodos de control tengan en cuenta el carácter no lineal de la dinámica de estos sistemas. Por este motivo la presente tesis se centra en el control no lineal de sistemas robotizados, por lo tanto se aborda el modelado dinámico de robots y el desarrollo de controladores dinámicos así como su implementación en entornos de tiempo real.

Por una parte, se desarrolla el modelo dinámico de un robot manipulador a partir de las ecuaciones de la dinámica de acuerdo al formalismo de Gibbs-Appell. Para ello, se asume el robot constituido por barras rígidas y pares ideales. Este algoritmo está formulado de una forma recursiva logrando por ello un coste computacional lineal, que permite desarrollar aplicaciones de tiempo real. Además se tiene en cuenta que los algoritmos de resolución del modelo dinámico necesitan conocer de forma fiable los parámetros físicos del robot, por lo que se tienen que desarrollar procedimientos para poder utilizar un conjunto de parámetros base identificados tanto en el Problema Dinámico como a la hora de construir la ecuación del movimiento del robot.

Por otra parte, a partir del modelo del robot se aborda el control dinámico de éste. Los controladores de movimiento desarrollados son controladores basados en el modelo del robot, es decir, aquellos que tienen en cuenta la descripción no lineal del sistema robotizado. De todos los posibles controladores de esta clase que existen, nos centramos en dos tipos, los controladores por dinámica inversa y los controladores basados en pasividad. Los primeros alcanzan el objetivo de seguimiento de trayectoria mediante la linealización por realimentación de la dinámica no lineal del robot. El segundo grupo modifica la energía mecánica del sistema con el fin de alcanzar el objetivo de control.

Finalmente, para poder validar y comprobar todos los desarrollos realizados en la tesis se trabaja con un robot industrial real, en este caso se ha utilizado un robot PUMA 560. Uno de los principales problemas que se encuentra cuando se intenta establecer el control de un robot industrial, ya sea el robot utilizado en esta tesis como cualquier otro, es su propia unidad de control, ya que generalmente es un sistema totalmente cerrado. Para resolver este tipo de problemas en la tesis se ha desarrollado una arquitectura abierta de control basada en un PC industrial.

Abstract

The control of robot manipulators is intended to improve quality, productivity and functionality of these industrial systems. For this, it is necessary that the design of control methods take into account the nonlinear dynamics of these systems. Therefore this thesis focuses on nonlinear control of robotic systems, therefore the dynamic models of robots and the development of dynamic control is addressed, and its implementation in real time environments.

On the one hand, the dynamic model of a robotic manipulator is developed starting from the dynamics equations according to the Gibbs-Appell formalism. To do this, the robot is assumed to be constituted by rigid links and ideal pairs. This algorithm is formulated in a recursive way so the computational cost is linear, which enables real-time applications. It also takes into account that the dynamic resolution algorithms of the model need to know a reliable way the physical parameters of the robot, which must develop procedures to use a base set of identified parameters in the dynamic problem as constructing the equation of motion of the robot.

On the other hand, from the robot model, the dynamic control is addressed. The motion controllers developed are based on the robot model, ie those which take into account the description of nonlinear robotic system. Of all the possible controllers of this type that exist; we focus on two types of controllers, the inverse dynamic controllers and the controllers based on passivity. The first reach the aim trajectory tracking by using the feedback linearization of the nonlinear dynamics of the robot. The second changes the mechanical energy of the system in order to achieve the aim of control.

Finally, to validate and verify all the developments made in the thesis are working with a real robot, in this case with a robot PUMA 560. A major problem is that when trying to establish the control of an industrial robot, the robot is already used in this thesis as any other, is its own control unit, which is usually a totally closed system. To resolve

such problems in the thesis has developed an open architecture for control based on an industrial PC.

Resum

El control de robots manipuladors té per objecte millorar la qualitat, productivitat i funcionalitat d'aquests sistemes industrials. Per a això, es fa necessari que el disseny dels mètodes de control tinguin en compte el caràcter no lineal de la dinàmica d'aquests sistemes. Per aquest motiu la present tesi es centra en el control no lineal de sistemes robotitzats, per tant s'aborda el modelatge dinàmic de robots i el desenvolupament de controladors dinàmics així com la seva implementació en entorns de temps real.

D'una banda, es desenvolupa el model dinàmic d'un robot manipulador a partir de les equacions de la dinàmica d'acord al formalisme de Gibbs-Appell. Per això, s'assumeix el robot constituït per barres rígides i parells ideals. Aquest algorisme està formulat d'una manera recursiva assolint per això un cost computacional lineal, que permet desenvolupar aplicacions de temps real. A més es té en compte que els algoritmes de resolució del model dinàmic necessiten conèixer de manera fiable els paràmetres físics del robot, per la qual cosa s'han de desenvolupar procediments per poder utilitzar un conjunt de paràmetres base identificats tant en el Problema Dinàmic com a l'hora de construir l'equació del moviment del robot.

D'altra banda, a partir del model del robot s'aborda el control dinàmic d'aquest. Els controladors de moviment desenvolupats són controladors basats en el model del robot, és a dir, aquells que tenen en compte la descripció no lineal del sistema robotitzat. De tots els possibles controladors d'aquesta classe que existeixen, ens centrem en dos tipus, els controladors per dinàmica inversa i els controladors basats en passivitat. Els primers assoleixen l'objectiu de seguiment de trajectòria mitjançant la linealització per realimentació de la dinàmica no lineal del robot. El segon grup modifica l'energia mecànica del sistema amb la finalitat d'assolir l'objectiu de control.

Finalment, per poder validar i comprovar tots els desenvolupaments realitzats en la tesi es treballa amb un robot industrial real, en aquest cas s'ha utilitzat un robot PUMA

560. Un dels principals problemes que es troba quan s'intenta establir el control d'un robot industrial, ja sigui el robot utilitzat en aquesta tesi com qualsevol altre, és la seva pròpia unitat de control, ja que generalment és un sistema totalment tancat. Per resoldre aquest tipus de problemes en la tesi s'ha desenvolupat una arquitectura oberta de control basada en un PC industrial.

Tabla de Contenidos

CAPÍTULO 1	INTRODUCCIÓN	1
1.1.	MOTIVACIÓN	1
1.2.	ANTECEDENTES	3
1.3.	OBJETIVOS	7
1.4.	ESTRUCTURA DE LA TESIS	9
CAPÍTULO 2	MODELADO DE SISTEMAS ROBOTIZADOS	11
2.1.	MODELADO CINEMÁTICO	11
2.1.1.	Problema cinemático Directo	14
2.1.2.	Problema cinemático inverso	15
2.2.	MODELADO DINÁMICO	16
2.2.1.	Obtención de un algoritmo eficiente para la resolución del Problema Dinámico Inverso mediante la formulación de Gibbs-Appell	17
2.3.	EMPLEO DE LOS PARÁMETROS IDENTIFICADOS EXPERIMENTALMENTE	39
2.3.1.	Aplicación de la resolución del PDI y PDD en estrategias de control	40

CAPÍTULO 3	CONTROL CINEMÁTICO DE SISTEMAS	
	ROBOTIZADOS	47
3.1.	INTRODUCCIÓN	47
3.2.	TIPOS DE TRAYECTORIAS	51
3.2.1.	Trayectorias Punto a Punto	51
3.2.2.	Trayectorias coordinadas o isócronas	52
3.3.	INTERPOLACIÓN DE TRAYECTORIAS	53
3.3.1.	Generación de curvas mediante la Interpolación de Hermite	54
3.3.2.	Generación de trayectorias en el espacio de nudos del robot mediante un interpolador cúbico	58
3.4.	IMPLEMENTACIÓN DE LAS SERIES DE FOURIER	63
3.4.1.	Parametrización de la trayectoria	66
CAPÍTULO 4	CONTROL DINÁMICO DE ROBOTS	
	MANIPULADORES	69
4.1.	INTRODUCCIÓN	69
4.2.	CONTROL POR DINÁMICA INVERSA	73
4.2.1.	Introducción	73
4.2.2.	Control Punto a Punto	77
4.2.3.	Seguimiento de la Trayectoria	85

4.2.4.	Seguimiento de la Trayectoria con Acción Integral	92
4.3.	CONTROL POR PASIVIDAD	96
4.3.1.	Introducción	96
4.3.2.	Controladores punto a punto por pasividad	96
 CAPÍTULO 5 PLATAFORMA EXPERIMENTAL DE CONTROL DE ROBOTS INDUSTRIALES. APLICACIÓN AL PUMA 560		 129
5.1.	ARQUITECTURA HARDWARE DEL PUMA 560	129
5.1.1.	Brazo Robot	130
5.1.2.	Unidad de Control	130
5.1.3.	Arquitectura Original	132
5.1.4.	Arquitectura de Control Abierta	135
5.1.5.	Arquitectura de Control Propuesta	135
5.1.6.	Circuito de freno	139
5.2.	ARQUITECTURA SOFTWARE DE CONTROL	141
5.2.1.	Introducción a la Real-Time Workshop	141
5.2.2.	La Arquitectura Abierta de la Real-Time Workshop	143
5.2.3.	S-Function	144

- xviii -*Estudio e implementación de controladores dinámicos de robots industriales en tiempo real*

5.2.4.	Modo Externo	146
5.2.5.	Embedded MATLAB Function	148
5.2.6.	Diseño del Algoritmo de Control en tiempo real	149
CAPÍTULO 6 CONCLUSIONES		153
6.1.	APORTACIONES	156
6.2.	PUBLICACIONES	157
6.2.1.	Futuras líneas de Investigación	158
CAPÍTULO 7 REFERENCIAS BIBLIOGRÁFICAS		161
ANEXO A ESTRATEGIAS DE CONTROL NO LINEAL DE ROBOTS		167
A.1.	CONTROL POR DINÁMICA INVERSA	168
A.1.1.	Control Punto a Punto	168
A.1.2.	Seguimiento de la Trayectoria	172
A.1.3.	Seguimiento de la Trayectoria con Acción Integral	176
A.2.	CONTROL POR PASIVIDAD	184
A.2.1.	Controlador PD con compensación de la gravedad	184
A.2.2.	PD con compensación de la gravedad en la posición deseada	189

A.2.3.	Regulador con sólo medida de la posición	192
A.2.4.	Controlador sin medida de la velocidad con compensación de las fuerzas de gravedad en la posición deseada	197
A.2.5.	Controlador con diferenciación aproximativa	201
A.2.6.	Controlador con diferenciación aproximativa con compensación de las fuerzas de gravedad en la posición deseada	205
A.2.7.	Controlador PID	210
A.2.8.	Controlador PI ² D	223
A.3.	HERRAMIENTAS PARA EL ANÁLISIS DE LA ESTABILIDAD	238
ANEXO B GENERACIÓN AUTOMÁTICA DE ALGORITMOS DE CONTROL DE ROBOTS		245
B.1.	GENERACIÓN AUTOMÁTICA DE CÓDIGO DE ESQUEMAS DE SIMULACIÓN	245
B.2.	GENERACIÓN AUTOMÁTICA DE CÓDIGO EN LOS ESQUEMAS DE CONTROL	258

Lista de Símbolos Más Empleados

\bar{g}	Aceleración de la gravedad.
G	Función de Gibbs o energía de las aceleraciones.
q_i	Posición angular generalizada correspondiente a la barra i .
\dot{q}_i	Primera derivada de la posición generalizada para la barra i .
\ddot{q}_i	Segunda derivada de la posición generalizada para la barra i .
${}^i\ddot{\vec{r}}_{O_i}$	Aceleración del origen del sistema de referencia local de la barra i .
${}^i\ddot{\vec{r}}_{G_i}$	Aceleración del centro de gravedad de la barra i .
${}^{i-1}\vec{r}_{O_{i-1},O_i}$	Vector de traslación que relaciona los sistemas de referencia locales de dos barras consecutivas.
${}^i\vec{r}_{O_i,G_i}$	Vector de posición del centro de gravedad de la barra i en el sistema de referencia local de la barra.
${}^{i-1}\mathbf{R}_i$	Matriz de rotación que relaciona dos sistemas de referencia locales consecutivos.
$M(\vec{q})$	Matriz de Inercias de la ecuación general del movimiento
$C(q, \dot{q})$	Vector de fuerzas de coriolis
$G(q)$	Vector de fuerzas gravitatorias
\mathbf{W}	Matriz de observación del sistema.
$\vec{\tau}$	Vector de fuerzas generalizadas en los nudos.

$\bar{\Phi}$	Vector de parámetros dinámicos del sistema.
${}^i\bar{\omega}_i$	Velocidad angular de la barra i .
${}^i\dot{\bar{\omega}}_i$	Aceleración angular de la barra i .
e	Error de posición
e_v	Error de velocidad
K_p	Ganancias Proporcionales
K_d	Ganancias Derivativas
K_i	Ganancias Integrales

CAPÍTULO 1

INTRODUCCIÓN

1.1. MOTIVACIÓN

En los últimos años la investigación en robótica se está centrando en el perfeccionamiento e incremento de la autonomía de los sistemas robotizados. Tradicionalmente los robots se han utilizado en entornos industriales realizando tareas preprogramables tales como manipulación de materiales, montaje de productos, soldadura en arco y punto a punto, pintura con spray, pulido, etc., de forma que la introducción de los robots en las áreas y procesos industriales está suponiendo un incremento en la calidad, productividad y flexibilidad, justificando plenamente la inversión de dichos equipos. De esta forma los robots no sólo se están empleando en tareas relativamente simples (que permiten liberar a las personas de tareas duras, repetitivas y, en ocasiones, peligrosas) sino que actualmente también los podemos encontrar en aplicaciones más complejas como por ejemplo la automatización avanzada en industrias, investigaciones espaciales, operaciones submarinas, etc.

De esta forma, la mayoría de los robots actuales están compuestos por el manipulador mecánico, el equipo de control, los equipos de potencia y fuentes de

alimentación, y los distintos dispositivos sensores. En el pasado el diseño de un sistema robot consistía en la integración de estos cuatro componentes que habían sido diseñados por separado y sin una interacción completa entre si. Sin embargo, los fuertes requerimientos impuestos (niveles más altos en calidad, productividad y funcionalidad) por las nuevas aplicaciones de los sistemas robotizados no pueden satisfacerse con esta filosofía de trabajo, haciendo necesaria una nueva filosofía de diseño que combine varias disciplinas ingenieriles tales como la ingeniería mecánica, la ingeniería eléctrica y la ingeniería informática.

Si bien la tendencia está cambiando en los últimos años, en general los sistemas de control de robots industriales actuales son controladores de posición cerrados que, si bien son válidos para la mayoría de aplicaciones industriales, tienen serias limitaciones de uso en el campo de la investigación.

Gracias a la disponibilidad de potentes microprocesadores a bajo coste se han logrado grandes avances en la teoría y aplicación del control no lineal de robots manipuladores, lo que proporciona sistemas más fiables, precisos y con mayores prestaciones. En cuanto a la teoría podemos destacar los desarrollos realizados en las áreas de la linealización por realimentación, control pasivo y técnicas de adaptación no lineal. En cuanto a las aplicaciones, se han desarrollado algunos sistemas prácticos de control no lineal como sistemas de control de vuelo fly by wire para aviones, robots avanzados o sistemas espaciales. Como resultado, el tema de control de robots está ocupando un lugar cada vez más importante en la ingeniería de control y es una parte fundamental de la formación de los ingenieros de control.

Se puede decir que una de las principales motivaciones de la presente tesis fue el hecho de investigar en el campo del control de sistemas robotizados para intentar incrementar la capacidad de éstos cara al desarrollo de estrategias de control más complejas que permitan mejorar la precisión, aumentar los requerimientos sobre prestaciones del robot (como por ejemplo la velocidad de ejecución de una trayectoria) o el grado de precisión y repetibilidad de un determinado movimiento.

1.2. ANTECEDENTES

Durante las pasadas décadas la robótica ha sido objeto de intensas investigaciones en el campo de los sistemas de control debido a las no linealidades inherentes a estos sistemas. Por la misma razón han sido ampliamente utilizados en la teoría de control no lineal para ejemplarizar conceptos generales de dicha teoría, generándose de esta forma una gran variedad de métodos de control para sistemas robotizados. Así, los requerimientos actuales en cuanto a funcionamiento y estabilidad de los sistemas robotizados hacen necesario que el diseño de los métodos de control tenga en cuenta el carácter no lineal de la dinámica de estos sistemas. Por este motivo la presente tesis se centra en el control no lineal de sistemas robotizados.

Se puede decir que los métodos de control se pueden clasificar según el objetivo de control del robot. Según esta clasificación se pueden distinguir dos tipos: control punto a punto (o regulación) y control de seguimiento de trayectoria (o control de movimiento). El control punto a punto es más restrictivo por lo que está tomando una especial importancia el control de seguimiento de trayectoria al aumentar el área de aplicaciones de los sistemas robot (aplicaciones tradicionales como pintado y soldadura así como nuevas aplicaciones como prospecciones minerales submarinas, estaciones espaciales etc.).

Al principio, los robots eran pesados y con las articulaciones rígidas lo que, junto a los mecanismos de transmisión rápidos, permitía que la dinámica de estos sistemas robot se pudieran describir por un conjunto de ecuaciones simples, por lo que un control por realimentación lineal del estado para cada grado de libertad, por ejemplo, proporcionaba un control aceptable de acuerdo con los estándares de la época. Sin embargo estos sistemas robot no satisfacen los requerimientos actuales por dos razones fundamentalmente: para alcanzar velocidades razonables estos sistemas pesados necesitan acciones de control altas, poco interesantes desde un punto de vista económico y ambiental. Además, la necesidad de mecanismos de transmisión de alta velocidad

puede introducir efectos no deseables, tales como flexibilidad, fricción y retroceso, introduciendo limitaciones en cuanto a funcionamiento y precisión. Para superar estas deficiencias existe una tendencia de desarrollo de sistemas robotizados ligeros accionados por motores directos. Sin embargo, la caracterización de la dinámica de estos sistemas es altamente no lineal y acoplada. Por ello es necesario tener en cuenta todos estos efectos no lineales de la dinámica del sistema robot para poder obtener un control de movimiento adecuado. En definitiva el poder incorporar al esquema de control del robot su comportamiento dinámico permite incrementar significativamente sus prestaciones sin tener que modificar significativamente su estructura mecánica.

A la hora de generar las ecuaciones del movimiento que gobiernan el comportamiento dinámico del robot se pueden considerar una notable variedad de alternativas. Se puede partir de diversos Principios de la Dinámica como son la formulación de Lagrange-Euler, Newton-Euler, Kane, etc. Además, las ecuaciones de movimiento pueden ser expresadas de forma cerrada o recursiva, y dichas, Dichas ecuaciones se pueden expresar mediante diferentes tipos de variables para expresar medidas físicas. Finalmente los algoritmos correspondientes a dichas ecuaciones pueden ser implementados mediante programación simbólica o estrictamente numérica.

Varios autores han hecho evidente que la eficiencia computacional de los algoritmos dinámicos depende fundamentalmente de la forma en que los cálculos están organizados más que del principio en que están basados. Esta idea ya fue propuesta por (Hollerbach, 1980), en el cual el problema dinámico fue reformulado para robots usando el principio de Lagrange de forma recursiva y usando matrices de rotación de 3x3 en lugar de las matrices de transformación homogéneas de 4x4. Mediante este método, la complejidad computacional se puede reducir de $O(n^4)$ a $O(n)$.

En esta tesis se ha resuelto el Problema Dinámico Inverso de sistemas robotizados que tienen barras rígidas y pares ideales basándose en las ecuaciones de Gibbs-Appell. El algoritmo propuesto está formulado de una forma recursiva usando vectores para expresar la mayoría de las magnitudes físicas incluidas en ellas (velocidades angulares,

aceleraciones angulares, etc). Para alcanzar la mayor eficiencia computacional, las magnitudes incluidas en la función de Gibbs están expresadas respecto al sistema de referencia local en las barras. De esta forma, se ha logrado un algoritmo con un coste computacional lineal $O(n)$, lo que va a permitir desarrollar aplicaciones de tiempo real.

Hay que tener en cuenta que los algoritmos de resolución del modelo dinámico necesitan conocer de forma fiable los parámetros físicos del robot (parámetros geométricos y parámetros dinámicos). Mientras que hay mucha información disponible de los parámetros geométricos no ocurre lo mismo en el caso de los dinámicos ya que ésta es sumamente escasa e incluso contradictoria. Por ello, en esta tesis se han utilizado un conjunto de parámetros identificados obtenidos en (Benimeli, 2006) mediante técnicas de identificación. Es importante resaltar que los parámetros identificados experimentalmente no son exactamente los parámetros físicos que aparecen en el modelo dinámico subyacente en la identificación; sino los denominados parámetros base. Este conjunto de parámetros base corresponden a algunos parámetros iniciales y a combinaciones lineales de los mismos, lo que refleja el hecho que en el caso de las cadenas cinemáticas los parámetros no suelen influir aisladamente en el comportamiento dinámico global sino en combinación con otros parámetros, donde esta combinación viene dada por la topología del sistema mecánico. El problema radica pues en la utilización de este conjunto de parámetros base identificados tanto en el Problema Dinámico como a la hora de construir la ecuación del movimiento del robot. En ambos casos se han desarrollado en esta tesis los procedimientos correspondientes.

Una vez obtenido el modelo del robot se puede abordar el control dinámico de éste. Los controladores de movimiento desarrollados en la tesis son controladores basados en el modelo del robot, que son aquellos que tienen en cuenta la descripción no lineal del sistema robotizado. Aunque en esta clase de controladores se pueden encontrar diferentes categorías, la tesis se ha centrado en dos clases de controladores, los controladores por dinámica inversa y los controladores basados en pasividad. Los primeros alcanzan el objetivo de seguimiento de trayectoria mediante la linealización por

realimentación de la dinámica no lineal del robot. El segundo grupo modifica la energía mecánica del sistema con el fin de alcanzar el objetivo de control.

En esta tesis los algoritmos de control desarrollados se basan en el PDI usando la formulación de Gibbs-Appell comentada anteriormente. La resolución del PDI con un coste computacional lineal permite la implementación del control dinámico de movimiento basándose en este algoritmo y así poder implementar distintas estrategias de control de tiempo real. Estas estrategias de control se pueden simular e incluso se pueden sintonizar las matrices de ganancias de los controladores implementados gracias al hecho de resolver el PDD, así como se pueden cancelar los términos gravitaciones de los controladores por pasividad no lineal a partir también de la resolución del PDD.

Para poder validar y comprobar todos los desarrollos realizados en la tesis se necesitaría trabajar con un robot industrial real, en este caso se ha utilizado un robot PUMA 560. Se trata de un robot industrial de 6 grados de libertad con pares de revolución y barras rígidas. Uno de los principales problemas que se encuentra cuando se intenta establecer el control de un robot industrial, ya sea el robot utilizado en esta tesis como cualquier otro, es su propia unidad de control, ya que generalmente es un sistema totalmente cerrado. Una de las razones por las cuales este sistema es cerrado se debe al hecho de que utiliza su propio sistema operativo y además es imposible modificar el sistema de control, incluso los valores de las ganancias. Estas características provocan que los usuarios tengan serios problemas para implementar estrategias de control tanto avanzadas como convencionales. Para resolver este tipo de problemas en la tesis se ha desarrollado una arquitectura abierta de control basada en un PC industrial. Este PC está equipado con un conjunto de tarjetas de adquisición de datos, conectándose directamente a la etapa de potencia del armario de control del robot industrial utilizado. De esta forma se puede implementar de una forma sencilla las diferentes estrategias de control propuestas.

1.3. OBJETIVOS

Los objetivos principales son el establecimiento del control dinámico y cinemático de un sistema robotizado industrial, el PUMA 560. Para ello se ha implementado una arquitectura sencilla, económica e independiente de la unidad original de control, de la cual únicamente se utiliza su unidad de potencia.

Los objetivos indicados anteriormente se pueden detallar de la siguiente manera:

- A la hora de establecer el control dinámico del robot, se necesitará obtener el modelo dinámico del robot por las razones ya comentadas en la sección anterior. Para ello, se seleccionará uno de los distintos métodos de obtención de dicho modelo, en este caso se basará en la notación de Gibbs-Appell. Esta formulación aportará varias ventajas a la hora de resolver el Problema Dinámico Inverso (PDI): El resultado del PDI es una formulación bien estructurada puesto que los términos dinámicos (inerciales, rozamientos viscosos y gravitacionales) de la ecuación del robot aparecen de forma separada, lo que facilitará el desarrollo y la implementación del control de los robots. Además, es una formulación recursiva, lo que proporcionará un coste computacional $O(n)$, adecuado para el desarrollo de aplicaciones de tiempo real.
- Otro de los objetivos de la investigación además del PDI, también será desarrollar un simulador del robot resolviendo el Problema Dinámico Directo (PDD). Dicho simulador es muy adecuado para poder probar fuera de línea tanto las estrategias de control desarrolladas como la sintonización de parámetros de dichas estrategias.
- Una vez resuelto del modelado dinámico de robots se procederá a establecer el control dinámico y cinemático de robots. Sabiendo que objetivo del control dinámico es hacer que las distintas articulaciones del robot sigan (con el menor error posible) las referencias generadas por el control cinemático. Los

controladores no lineales que se desarrollarán serán los siguientes: Controladores pasivos que son una clase de controladores que, modificando la energía del sistema controlador-robot, establecen el control del robot. Estos controladores cancelan los términos gravitacionales del sistema, por lo que se podrá calcular el término gravitatorio a partir de la resolución del PDI. Los otros tipos de controladores son los controladores por dinámica inversa o por calculado, en este caso estos controladores establecen el control realizando una linealización por realimentación, compensando de este modo los términos no lineales del robot. Gracias a la formulación de Gibbs-Appell se podrá implementar distintos algoritmos de control que tengan en cuenta la estimación realizada en línea.

- Estos algoritmos comentados son útiles en el caso de conocer de forma fiable los parámetros físicos del robot, en el caso de que no sea así se podrá utilizar parámetros identificados, este conjunto de parámetros identificados se adaptarán tanto a la resolución del PDI como a la obtención de los términos de la ecuación del movimiento con vistas a la resolución del PDD o a la incorporación de alguno de sus términos a esquemas de control ya comentados.
- Respecto al control cinemático de robots, se sabe que dicho control tiene como principal objetivo generar las referencias de movimiento al control dinámico. Para ello se generarán distintos tipos de referencias de movimiento, por ejemplo trayectorias de tipo rectilíneo, basadas en Splines cúbicas naturales o mediante puntos intermedios con interpoladores cúbicos o incluso series finitas de Fourier que son necesarias para poder generar referencias excitables que permiten identificar el modelo dinámico del PUMA 560, y más concretamente sus parámetros inerciales. A la hora de generar las referencias en el espacio cartesiano y resolver el problema que estudia el movimiento de robot con respecto a un sistema de referencia, se hace necesario resolver tanto el Problema Cinemático Directo como el Inverso. En el caso del problema cinemático directo se puede resolver de una manera directa a partir de la utilización de matrices de

transformación homogéneas, e independientemente de la configuración del robot. En el caso del problema cinemático inverso se puede resolver mediante relaciones trigonométricas y geométricas.

- Una vez implementadas las distintas estrategias de control satisfactoriamente, realizará una comparativa entre todas ellas. Para poder realizar dicha comparativa lo ideal será implementar dichas estrategias de control sobre un sistema real, en este caso se implementarán sobre un robot industrial PUMA 560, sobre el cual se desarrollará una arquitectura de control abierta que permitirá establecer el control de una forma independiente a la unidad de control del sistema, es decir, se realizará sobre un PC industrial conectado a la etapa de potencia del armario de control.
- Una vez finalizados dichos objetivos, la programación del robot también se podría realizar fuera de línea a partir de una simulación o bien a partir de la información proporcionada directamente por el usuario.

1.4. ESTRUCTURA DE LA TESIS

Una vez introducidos la motivación y los objetivos perseguidos, se procede a presentar la estructura de esta tesis, junto con una breve descripción del contenido de cada capítulo. En el Capítulo 2, se describe el modelo cinemático y dinámico del robot manipulador, así como los fundamentos de la formulación dinámica de Gibbs-Appell, a partir de la cual ha sido obtenido el modelo dinámico. Igualmente, se trata el empleo de los parámetros inerciales identificados experimentalmente que permite aplicar la resolución tanto del PDI como del PDD en las estrategias de control. El Capítulo 3 se ha dedicado la descripción del control cinemático y los diferentes tipos de trayectorias implementadas para la realización de diferentes estrategias de control de robots manipuladores. En el Capítulo 4, se describen los distintos algoritmos de control

implementados, entre los que se encuentran los controladores por dinámica inversa y los controladores pasivos, así como resultados obtenidos de la aplicación sobre el robot PUMA 560. El Capítulo 5 describe la arquitectura de control utilizada en la presente tesis. Se trata de una arquitectura abierta donde la unidad de control está constituida por un PC, lo cual implica una gran libertad a la hora de implementar estrategias de control complejas. Además también se describe la arquitectura software utilizada, ya sea para realizar el control sobre el robot real mediante lenguajes de alto nivel accediendo directamente al hardware o bien utilizando herramientas de generación automática de código como sería la utilización de la Real-Time Workshop que permite tanto controlar el robot real como simularlo. Finalmente, en el Capítulo 6 se exponen las conclusiones extraídas del presente trabajo y se indican algunos aspectos que se han considerado susceptibles de ser mejorados o desarrollados en futuros trabajos. En el Anexo A se incluyen resultados reales de los controladores implementados así como su demostración y en el Anexo B se incluyen estos mismos controladores implementados en Simulink utilizando la Real-Time Workshop y resultados de la simulación del sistema.

CAPÍTULO 2

MODELADO DE SISTEMAS ROBOTIZADOS

2.1. MODELADO CINEMÁTICO

En esta sección, se ha llevado a cabo la descripción del modelo cinemático de un robot con configuración de cadena cinemática abierta.

La cinemática del robot estudia el movimiento del mismo sin atender a las causas que lo provocan. De la misma forma se puede decir que la cinemática estudia como varía la geometría del robot a lo largo del tiempo. Usualmente en robots ese estudio comprende posición, velocidad y aceleración.

Como es sabido existen dos problemas fundamentales a resolver en la cinemática del robot (*Figura 2-1*); el primero se conoce como el Problema Cinemático Directo, y consiste en determinar cuál es la posición y orientación del extremo final del robot (elemento terminal), con respecto a un sistema de coordenadas que se toma como referencia, conocidos los valores de las denominadas variables de nudo y los parámetros geométricos constantes de los elementos del robot; el segundo, denominado Problema

Cinemático Inverso, resuelve las variables de nudo que corresponden a una posición y orientación dadas del elemento terminal del robot.

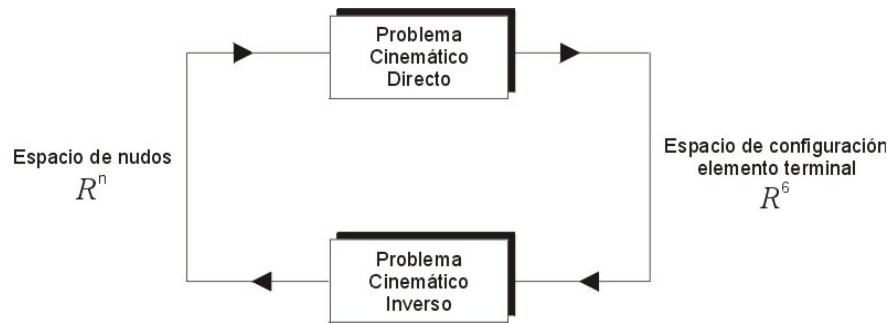


Figura 2-1 Problemas cinemáticos

Como es habitual en el campo de la Robótica, se empleará la notación de Dénavit-Hartenberg (D-H) (Denavit, J. y Hartenberg, R.S., 1955) para obtener la modelización cinemática del robot industrial PUMA 560. Hay que destacar que la aplicación de la notación D-H a mecanismos de cadena cinemática abierta conduce a un conjunto de coordenadas independientes, lo cual es muy conveniente, tanto desde el punto de vista del control como de la simulación. La aplicación de D-H permite describir la relación entre sistemas de referencia adyacentes mediante cuatro parámetros a_i , α_i , d_i y q_i . De estos cuatro parámetros, los tres primeros son constantes, y definen la geometría del robot, mientras que el cuarto es variable y corresponde a la coordenada de nudo, en adelante, coordenada generalizada.

En el esquema mostrado en la *Figura 2-2* se muestra el proceso de asignación de sistemas de referencia locales a la barra fija, barras móviles y elemento terminal del robot según la convención de D-H, bajo el criterio de J.J. Craig (Craig, 1984).

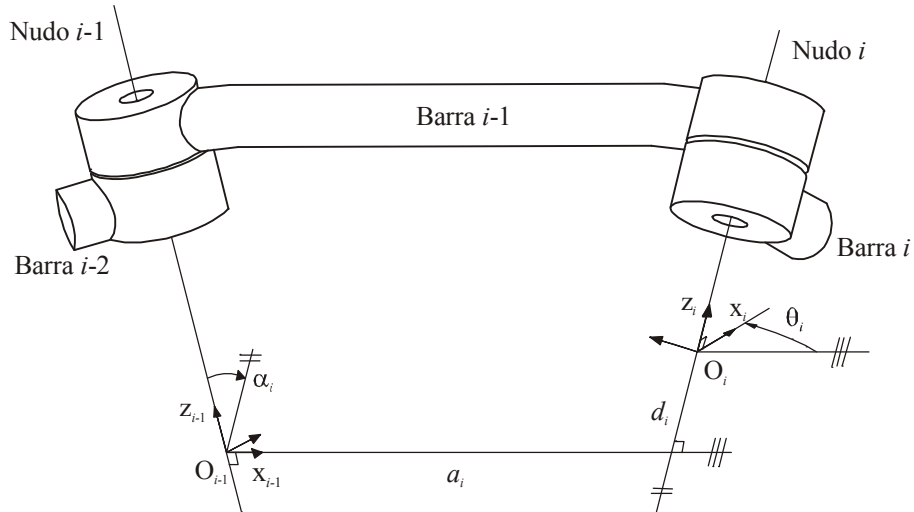


Figura 2-2 Asignación de sistemas de referencia según D-H

A partir de la notación de D-H es inmediata la obtención de la matriz de transformación homogénea que relaciona dos sistemas de referencia adyacentes,

$${}^{i-1}A_i = \begin{bmatrix} \cos q_i & -\sin q_i & 0 & a_i \\ \cos \alpha_i \cdot \sin q_i & \cos \alpha_i \cdot \cos q_i & -\sin q_i & d_i \cdot \sin \alpha_i \\ \sin \alpha_i \cdot \sin q_i & \sin \alpha_i \cdot \cos q_i & \cos q_i & -d_i \cdot \cos \alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Cabe resaltar la composición de la matriz anterior. Se puede definir en cada caso un vector de traslación ${}^{i-1}\vec{r}_{O_{i-1},O_i}$ y una matriz de rotación ${}^{i-1}\mathbf{R}_i$ que relacionan dos sistemas de referencia locales consecutivos.

$${}^{i-1}\mathbf{R}_i = \begin{bmatrix} \cos q_i & -\sin q_i & 0 \\ \cos \alpha_i \cdot \sin q_i & \cos \alpha_i \cdot \cos q_i & -\sin q_i \\ \sin \alpha_i \cdot \sin q_i & \sin \alpha_i \cdot \cos q_i & \cos q_i \end{bmatrix} \quad (2.2)$$

$${}^{i-1}\vec{r}_{O_{i-1},O_i} = \begin{bmatrix} a_i \\ d_i \sin \alpha_i \\ -d_i \cos \alpha_i \end{bmatrix} \quad (2.3)$$

Una vez establecida la herramienta que permite relacionar sistemas de referencia consecutivos, se discutirán los problemas cinemáticos directo e inverso de posición.

2.1.1. Problema cinemático Directo

Como ya se ha indicado anteriormente el problema cinemático directo pretende la determinación de la posición y orientación del sistema de referencia asignado al elemento terminal del robot en función de las variables de nudo. Para los casos de los robots con pares de revolución se tendrá que $q_i \equiv \theta_i$.

Para un robot serie con N grados de libertad, el Problema Cinemático Directo de posición queda reducido a la siguiente expresión:

$$T = {}^0A_1(q_1) \cdot {}^1A_2(q_2) \cdot \dots \cdot {}^{n-1}A_n(q_n) \quad (2.4)$$

donde T es la matriz de transformación homogénea que describe la posición y orientación del elemento terminal con respecto a un sistema de referencia ligado a la base del manipulador. Dado un conjunto de variables de nudo $[q_1, q_2, \dots, q_n]^t$, la correspondiente matriz T es única y su obtención constituye el problema cinemático directo de posición.

2.1.2. Problema cinemático inverso

El objetivo del problema cinemático inverso consiste en determinar los valores de las coordenadas de nudo para que el extremo se posicione y oriente según una determinada localización espacial.

A diferencia de lo que ocurre con el Problema Cinemático Directo de posición, el Problema Cinemático Inverso precisa de la resolución de un sistema de ecuaciones no lineales de tipo trascendental. Esto es, dados los valores de la matriz T , se trata de obtener los valores de $\vec{q} = [q_1, q_2, \dots, q_n]^T$ que satisfacen las ecuaciones (2.4).

Este sistema de ecuaciones no lineales puede ser resuelto mediante métodos numéricos (Newton-Raphson), pero el tiempo de cómputo impide su aplicación en tareas de control en tiempo real. Por ello, se busca una solución explícita del sistema de ecuaciones no lineales. Como es bien sabido, un sistema de ecuaciones no lineales no necesariamente admiten solución explícita. (Pieper, 1968). Pieper estableció que para un robot con seis grados de libertad es condición suficiente para su resolubilidad, la existencia de tres nudos de revolución consecutivos cuyos ejes se intersecten en un solo punto para todas las posibles configuraciones del brazo. La práctica totalidad de robots industriales verifican esta condición, agrupando sus tres últimos grados de libertad en lo que se denomina “muñeca esférica”.

En la literatura se han propuestos diversos procedimientos para la obtención de las soluciones explícitas del Problema Cinemático Inverso para un robot del tipo PUMA 560. En esta tesis se ha utilizado el propuesto por (Fu, González y Lee, 1987).

Hay que indicar que una de las ventajas evidentes de las soluciones explícitas, es que permite obtener todas las configuraciones posibles que verifican la posición y orientación del elemento terminal deseada, en el caso de un PUMA 560 son 8 las posibles soluciones. A efectos de control, hay que elegir solo una de ellas, en primer lugar se descartarían todas aquellas configuraciones físicamente no factibles, es decir, que no

cumplen con las restricciones al movimiento presentes en la estructura mecánica de los nudos. Como criterio para seleccionar una de entre las válidas, en este trabajo, se ha empleado el criterio de máxima proximidad en el espacio de nudos a la configuración precedente.

2.2. MODELADO DINÁMICO

La dinámica se ocupa de la relación entre las fuerzas que actúan sobre un cuerpo y el movimiento que en él se origina. Por lo tanto, el modelo dinámico de un robot tiene por objetivo conocer la relación entre el movimiento del robot y las fuerzas implicadas en el mismo.

Esta relación se plasma en las denominadas ecuaciones del movimiento, que relacionan el movimiento del sistema mecánico, en este caso el robot, con las causas que provocan dicho movimiento. En definitiva relacionan posición, velocidad y aceleración del robot (expresadas en términos de coordenadas generalizadas y sus derivadas temporales), con los pares de fuerzas actuantes, en el caso de que el robot esté constituido por pares de rotación. A partir de este modelo dinámico, se pueden considerar dos tipos básicos de problemas, el Problema Dinámico Inverso (PDI) o de Análisis de Fuerzas y el Problema Dinámico Directo (PDD) o problema de simulación del movimiento.

En la presente tesis se han propuestos algoritmos computacionalmente eficientes basados en las ecuaciones de Gibbs-Apell, para la resolución del PDI y la obtención de los términos que componen la ecuación del movimiento. Posteriormente se aplicarán resultados de indentificación de parámetros inerciales tanto a dicho problema dinámico inverso como a la obtención de los términos de la ecuación del movimiento, necesarios para resolución del problema dinámico directo.

2.2.1. Obtención de un algoritmo eficiente para la resolución del Problema Dinámico Inverso mediante la formulación de Gibbs-Appell

2.2.1.1. Introducción

En los últimos cuarenta años se han propuesto numerosos algoritmos para la resolución del PDI siempre buscando incrementar la mayor eficiencia computacional. Los primeros trabajos se remontan a 1965, en ese año Uicker (Uicker, 1965) formuló un método basado en las ecuaciones de Lagrange y matrices de transformación homogéneas de 4×4 . Ese mismo año, Hooker y Margulies (Hooker y Margulies, 1965) formularon un método, basado en las ecuaciones de Newton-Euler. Posteriormente, Kahn (Kahn, 1969) aplicó el método de Uicker a cadenas cinemática abiertas. En estos algoritmos basados en diferentes Principios de Dinámica (Lagrange, Newton-Euler, Kane), las ecuaciones de movimientos están expresadas de forma cerrada, por lo que su complejidad computacional era elevada, generalmente de orden $O(n^4)$.

El primer algoritmo recursivo que resuelve el PDI, del que se tiene noticias, se debe a Stepanenko y Vukobratovic, (Stepanenko y Vukobratovic, 1976). Estos autores se basan en el principio de la dinámica de Newton-Euler. Ese mismo año, Orin (Orin et al, 1980), introdujeron el sistema de coordenadas local y empleando las ecuaciones de Newton-Euler obtuvieron un algoritmo de complejidad $O(n^2)$. Un año después Luh, Walker y Paul (Luh, Walker y Paul, 1980) desarrollaron una formulación recursiva eficiente para resolver el PDI, el método se basa en Newton-Euler y posee una complejidad de orden $O(n)$. Paralelamente Hollerbach (Hollerbach, 1980) presentó un algoritmo recursivo basado en las ecuaciones de Lagrange-Euler con el mismo coste computacional aunque el número de operaciones aritméticas era elevado. Posteriormente, Renaud (Renaud, 1983) propuso un método, basado en las ecuaciones de Lagrange, con el cual se lograban algoritmos eficientes, respecto al número de operaciones, pero de orden $O(n^3)$.

La aplicación de las ecuaciones de Kane en el campo de la robótica, fue iniciada en 1983 por Kane y Levinson, (Kaner y Levinson, 1983), quienes resolvieron los problemas dinámicos directo e inverso con una complejidad similar a los algoritmos más eficientes de la época. Posteriormente (Angeles, Ma y Rojas, 1989) utilizaron notación tensorial para desarrollar, a partir de las mismas ecuaciones de Kane, un algoritmo eficiente para la resolución del PDI.

En la siguiente tabla se muestran algunos de los algoritmos propuestos para resolver el PDI de robots con barras rígidas y articulaciones ideales.

Tabla 2-1 Algunos algoritmos para resolver el PDI

Autores	Principios Dinámicos	Formulación	Tipos de variables	Tipos de resolución	Tipos de procesamiento
(Hollerbach, 1980)	Lagrange-Euler	Recursiva	Matricial	Numérica	Secuencial
(Luh et al, 1980)	Newton-Euler	Recursiva	Vectorial	Numérica	Secuencial
(Angeles, et al, 1989)	Kane	Recursiva	Tensorial	Numérica	Secuencial
(Balafoutis y Patel, 1991)	Newton-Euler	Recursiva	Tensorial	Numérica	Secuencial
(Khalil y Kleifinger, 1987)	Newton-Euler	Recursiva	Vectorial	Simbólica	Secuencial
(Lee and Chang, 1986)	Newton-Euler	Recursiva	Vectorial	Numérica	Paralelo

Mención especial merecen los algoritmos hechos a medida, los cuales sacan ventaja de las características especiales de un robot industrial en particular. Ejemplos de éstos se proponen en (Khosla y Newman, 1985; Murray y Newman, 1988), ambos basados en la formulación de Newton-Euler, expresados en forma recursiva y con vectores. Sin embargo, el primero es numérico y el segundo simbólico.

Varios autores (Balafoutis y Patel, 1991) han hecho evidente que la eficiencia computacional de los algoritmos dinámicos depende fundamentalmente de la forma en que los cálculos están organizados más que del principio en que están basados. Esta idea ya fue propuesta por (Hollerbach, 1980), en el cual el problema dinámico fue reformulado para robots usando el principio de Lagrange de forma recursiva y usando matrices de rotación de 3x3 en lugar de las matrices de transformación homogéneas de 4x4. Mediante este método, la complejidad computacional se puede reducir de $O(n^4)$ a $O(n)$. No obstante, la complejidad computacional del algoritmo de Hollerbach era tres veces superior al de Luh, Walker y Paul, como ya se ha mencionado anteriormente.

Por otra parte, cabe destacar que es posible observar diferencias importantes en la eficiencia computacional asignada a algoritmos con las mismas características aplicadas a robots del mismo tipo. En (Luh, et al, 1980) puede encontrarse un algoritmo basado en la formulación de Newton-Euler, implementado de una forma recursiva, usando vectores y resuelto de una forma numérica y secuencial, al cual Hollerbach asignó una complejidad computacional de $150n-48$ multiplicaciones y $131n-48$ sumas, siendo n el número de grados de libertad del robot. (Fu, et al, 1987) proporciona una versión del mismo algoritmo con una complejidad de $117n-24$ multiplicaciones y $103n-21$ sumas. (Zomaya, 1992) da una complejidad de $150n$ multiplicaciones y $116n$ sumas. Finalmente, (Craig 1986) da una complejidad de $126n-99$ multiplicaciones y $106n-92$ sumas.

Las diferencias observadas vienen fundamentalmente del criterio usado para contar operaciones (por ejemplo, si las operaciones que contienen multiplicaciones por variables con valor 0 ó 1 son detectadas). Por consiguiente, parece necesario indicar claramente el criterio seguido para contar operaciones cuando se compara la eficiencia de algoritmos.

Las ecuaciones de Gibbs-Appell fueron introducidas por Gibbs en 1879 y formalizadas por Appell veinte años después. Sin embargo, en el campo de la dinámica de robots son pocas las publicaciones de trabajos basados en ellas. (Renaud, 1975) sobresale entre las primeras referencias de la publicación de las ecuaciones de Gibbs-

Appell para la modelación dinámica de robots, en la cual hace comentarios sobre el anterior trabajo de E.P. Popov. (Vukobratovic y Kircanski, 1985) que desarrolló un algoritmo cerrado con un coste computacional de $O(n^3)$ computacional complejidad. (Desoyer y Lugner, 1989) desarrollaron un algoritmo recursivo para resolver el PDI en robots usando la matriz Jacobiana para evitar derivadas algebraicas o numéricas. El coste computacional del algoritmo propuesto es de $O(n^3)$.

En la presente tesis se han aplicado las ecuaciones de Gibbs-Appell para resolver el PDI y el PDD de robots que tienen barras rígidas y pares ideales. Este primer algoritmo propuesto tiene una complejidad computacional de $O(n)$. Este algoritmo está formulado de una forma recursiva, usando vectores para expresar la mayoría de las magnitudes físicas incluidas en ellas (velocidades angulares, aceleraciones angulares, etc). Para alcanzar la mayor eficiencia computacional, las magnitudes incluidas en la función de Gibbs están expresadas respecto al sistema de referencia local en las barras. La complejidad computacional de este algoritmo se comparará con el de Luh, Walker y Paul. Se puede hacer porque usan el mismo tipo de formulación (recursiva), y las magnitudes físicas comunes están expresadas de la misma forma. Se debe de afirmar que el criterio de evaluación del número de operaciones será el mismo en ambos algoritmos.

2.2.1.2. Aplicación al PUMA 560 del Algoritmo de Resolución del PDI

En esta sección, se describen las ecuaciones de Gibbs-Appell, y se presenta una formulación para resolver el Problema Dinámico Inverso sobre robots manipuladores. Como se ha comentado anteriormente en el modelado cinemático, los robots se modelan siguiendo la notación de Denavit-Hartenberg modificada, la cual considera cuatro parámetros α_i, θ_i, a_i y d_i . También, como ya se ha comentado, el sistema de referencia i está relacionado con el sistema de referencia $i-1$ mediante la matriz de rotación ${}^{i-1}R_i$ y el vector de posición ${}^{i-1}\vec{r}_{O_{i-1},O_i}$.

Las ecuaciones de la dinámica de Gibbs-Appell provienen de la definición de la función de Gibbs (también conocida como de la energía de las aceleraciones). Cuando fueron escritas en su forma original para un sólido arbitrario compuesto por n partículas elementales con masas m_i una aceleración a_i es (considerando un sistema de referencia inercial):

$$G = \frac{1}{2} \sum_{i=1}^n m_i a_i^2 \quad (2.5)$$

La función de Gibbs para el sólido rígido i -ésimo está dada por

$$G_i = \frac{1}{2} m_i (\ddot{\vec{r}}_{G_i})^T \cdot \ddot{\vec{r}}_{G_i} + \frac{1}{2} (\dot{\vec{\omega}}_i)^T \cdot \mathbf{I}_{G_i} \cdot \dot{\vec{\omega}}_i + (\dot{\vec{\omega}}_i)^T \cdot [\vec{\omega}_i \times (\mathbf{I}_{G_i} \cdot \vec{\omega}_i)] \quad (2.6)$$

donde m_i es la masa de la barra i -ésima, \mathbf{I}_{G_i} es la matriz 3x3 que representa la matriz centroide de inercia de la barra i -ésima, $\vec{\omega}_i$ y $\dot{\vec{\omega}}_i$ son vectores tridimensionales que representan la velocidad y aceleración angular de la barra i -ésima y, $\ddot{\vec{r}}_{G_i}$ es el vector tridimensional que representa la aceleración de del centro de masa de la barra i -ésima. Se considera un sistema de referencia inercial para expresar estas magnitudes.

Es posible expresar estas magnitudes tensoriales y vectoriales considerando un sistema de referencia localizado en la barra i -ésima, así pues la expresión anterior se puede expresar como sigue:

$$G_i = \frac{1}{2} m_i ({}^0\mathbf{R}_i \cdot {}^i \ddot{\vec{r}}_{G_i})^T \cdot {}^0\mathbf{R}_i \cdot {}^i \ddot{\vec{r}}_{G_i} + \frac{1}{2} ({}^0\mathbf{R}_i \cdot {}^i \dot{\vec{\omega}}_i)^T \cdot {}^0\mathbf{R}_i \cdot {}^i \mathbf{I}_{G_i} \cdot ({}^0\mathbf{R}_i)^T \cdot ({}^0\mathbf{R}_i \cdot {}^i \dot{\vec{\omega}}_i) + ({}^0\mathbf{R}_i \cdot {}^i \dot{\vec{\omega}}_i)^T \cdot \left\{ {}^0\mathbf{R}_i \cdot {}^i \vec{\omega}_i \times \left[{}^0\mathbf{R}_i \cdot {}^i \mathbf{I}_{G_i} \cdot ({}^0\mathbf{R}_i)^T \cdot {}^0\mathbf{R}_i \cdot {}^i \vec{\omega}_i \right] \right\} \quad (2.7)$$

la expresión (2.7) puede ser rescrita como sigue:

$$G_i = \frac{1}{2} m_i (\ddot{\vec{r}}_{G_i})^T \cdot ({}^0 \mathbf{R}_i)^T \cdot {}^0 \mathbf{R}_i \cdot {}^i \ddot{\vec{r}}_{G_i} + \frac{1}{2} ({}^i \dot{\vec{\omega}}_i)^T \cdot ({}^0 \mathbf{R}_i)^T \cdot {}^0 \mathbf{R}_i \cdot {}^i \mathbf{I}_{G_i} \cdot ({}^0 \mathbf{R}_i)^T \cdot {}^0 \mathbf{R}_i \cdot {}^i \dot{\vec{\omega}}_i + \\ + ({}^i \dot{\vec{\omega}}_i)^T \cdot ({}^0 \mathbf{R}_i)^T \cdot \{ {}^0 \mathbf{R}_i \cdot {}^i \vec{\omega}_i \times [{}^0 \mathbf{R}_i \cdot {}^i \mathbf{I}_{G_i} \cdot ({}^0 \mathbf{R}_i)^T \cdot {}^0 \mathbf{R}_i \cdot {}^i \vec{\omega}_i] \} \quad (2.8)$$

y teniendo en cuenta la naturaleza ortogonal de la matriz de rotación, el escalar G_i podría ser dado por:

$$G_i = \frac{1}{2} m_i ({}^i \ddot{\vec{r}}_{G_i})^T \cdot {}^i \ddot{\vec{r}}_{G_i} + \frac{1}{2} ({}^i \dot{\vec{\omega}}_i)^T \cdot {}^i \mathbf{I}_{G_i} \cdot {}^i \dot{\vec{\omega}}_i + ({}^i \dot{\vec{\omega}}_i)^T \cdot [{}^i \vec{\omega}_i \times {}^i \mathbf{I}_{G_i} \cdot {}^i \vec{\omega}_i] \quad (2.9)$$

donde ${}^i \ddot{\vec{r}}_{G_i}$, ${}^i \vec{\omega}_i$, ${}^i \dot{\vec{\omega}}_i$ y ${}^i \mathbf{I}_{G_i}$ están expresadas en el sistema de referencia i -ésimo.

Para un sistema consistente en n cuerpos, la función de Gibbs de ese sistema podría ser dada por

$$G = \sum_{i=1}^n G_i \quad (i = 1, 2, \dots, n) \quad (2.10)$$

Las ecuaciones del movimiento de Gibbs-Appell se obtienen derivando respecto a las aceleraciones generalizadas \ddot{q}_j , obteniendo de esta forma las fuerzas inerciales que se corresponden a las fuerzas externas generalizadas, τ_j

$$\tau_j = \sum_{i=j}^n \frac{\partial G_i}{\partial \ddot{q}_j} \quad (j = 1, 2, \dots, n) \quad (2.11)$$

esto es,

$$\tau_j = \sum_{i=j}^n \left\{ m_i ({}^i \ddot{\mathbf{r}}_{G_i})^T \cdot \frac{\partial {}^i \ddot{\mathbf{r}}_{G_i}}{\partial \ddot{q}_j} + ({}^i \dot{\boldsymbol{\omega}}_i)^T \cdot {}^i \mathbf{I}_{G_i} \cdot \frac{\partial {}^i \dot{\boldsymbol{\omega}}_i}{\partial \ddot{q}_j} + \left(\frac{\partial {}^i \dot{\boldsymbol{\omega}}_i}{\partial \ddot{q}_j} \right)^T \cdot [{}^i \boldsymbol{\omega}_i \times ({}^i \mathbf{I}_{G_i} \cdot {}^i \boldsymbol{\omega}_i)] \right\} \quad (2.12)$$

La formulación para la resolución del Problema Dinámico Inverso se podría obtener reorganizando e identificando dos términos diferentes de (2.12) como sigue:

$$\tau_j = \underbrace{\sum_{i=j}^n \left\{ \left(\frac{\partial {}^i \dot{\boldsymbol{\omega}}_i}{\partial \ddot{q}_j} \right)^T \cdot [{}^i \mathbf{I}_{G_i} \cdot {}^i \dot{\boldsymbol{\omega}}_i + {}^i \boldsymbol{\omega}_i \times ({}^i \mathbf{I}_{G_i} \cdot {}^i \boldsymbol{\omega}_i)] \right\}}_{A_j} + \underbrace{\sum_{i=j}^n \left\{ \left(\frac{\partial {}^i \ddot{\mathbf{r}}_{G_i}}{\partial \ddot{q}_j} \right)^T \cdot m_i \cdot {}^i \ddot{\mathbf{r}}_{G_i} \right\}}_{B_j} \quad (2.13)$$

Hay que resaltar que la expresión de abajo es similar a la propuesta por (Angeles, et al, 1989) para resolver el PDI basado en la formulación dinámica de Kane.

Para obtener las fuerzas generalizadas, las velocidades angulares, las aceleraciones del origen del sistema de coordenadas de las barras y las aceleraciones de los centros de masa de las barras se pueden utilizar las siguientes expresiones recursivas ya conocidas:

$${}^i \boldsymbol{\omega}_i = {}^i R_{i-1} \cdot {}^{i-1} \boldsymbol{\omega}_{i-1} + {}^i \bar{\mathbf{z}}_i \cdot \dot{q}_i \quad (2.14)$$

$${}^i \dot{\boldsymbol{\omega}}_i = {}^i R_{i-1} \cdot {}^{i-1} \dot{\boldsymbol{\omega}}_{i-1} + {}^i \bar{\mathbf{z}}_i \cdot \ddot{q}_i + {}^i R_{i-1} \cdot {}^{i-1} \boldsymbol{\omega}_{i-1} \times ({}^i \bar{\mathbf{z}}_i \cdot \dot{q}_i) \quad (2.15)$$

$${}^i \ddot{\mathbf{r}}_{O_i} = {}^i R_{i-1} \left[{}^{i-1} \ddot{\mathbf{r}}_{O_{i-1}} + {}^{i-1} \boldsymbol{\omega}_{i-1} \times ({}^{i-1} \boldsymbol{\omega}_{i-1} \times {}^{i-1} \bar{\mathbf{r}}_{O_{i-1}, O_i}) + {}^{i-1} \dot{\boldsymbol{\omega}}_{i-1} \times {}^{i-1} \bar{\mathbf{r}}_{O_{i-1}, O_i} \right] + (1 - \rho_i) \left[{}^i \bar{\mathbf{z}}_i \cdot \ddot{q}_i + 2({}^i \boldsymbol{\omega}_i \times {}^i \bar{\mathbf{z}}_i \cdot \dot{q}_i) \right] \quad (2.16)$$

$${}^i \ddot{\mathbf{r}}_{G_i} = {}^i \ddot{\mathbf{r}}_{O_i} + {}^i \boldsymbol{\omega}_i \times ({}^i \boldsymbol{\omega}_i \times {}^i \bar{\mathbf{r}}_{O_i, G_i}) + {}^i \dot{\boldsymbol{\omega}}_i \times {}^i \bar{\mathbf{r}}_{O_i, G_i} \quad (2.17)$$

donde ${}^i \vec{z}_i = [0 \ 0 \ 1]^T$, y la variable ρ_i nos permite distinguir entre articulaciones de revolución ($\alpha_i = 1$) y prismáticas ($\theta_i = 0$).

Desarrollando el término A_j :

$$A_j = \sum_{i=j}^n \left\{ \left({}^i \mathbf{R}_j \cdot \frac{\partial {}^j \dot{\bar{\omega}}_j}{\partial \ddot{q}_j} \right)^T \cdot \left[{}^i \mathbf{I}_{G_i} \cdot {}^i \dot{\bar{\omega}}_i + {}^i \bar{\omega}_i \times ({}^i \mathbf{I}_{G_i} \cdot {}^i \bar{\omega}_i) \right] \right\} \quad (2.18)$$

Para obtener el término $\frac{\partial {}^i \dot{\bar{\omega}}_i}{\partial \ddot{q}_j}$, empezamos desde la expresión (2.15). Esta

derivada se puede obtener mediante un proceso recursivo como sigue:

$$\text{Si } i < j \quad \frac{\partial {}^i \dot{\bar{\omega}}_i}{\partial \ddot{q}_j} = [0 \ 0 \ 0]^T$$

$$\text{Si } i > j \quad \frac{\partial {}^i \dot{\bar{\omega}}_i}{\partial \ddot{q}_j} = {}^i \mathbf{R}_{i-1} \cdot {}^{i-1} \mathbf{R}_{i-2} \cdots {}^j \mathbf{R}_j \cdot \frac{\partial {}^j \dot{\bar{\omega}}_j}{\partial \ddot{q}_j} \quad (2.19)$$

$$\text{Si } i = j \quad \frac{\partial {}^i \dot{\bar{\omega}}_i}{\partial \ddot{q}_j} = {}^i \vec{z}_i$$

Teniendo en cuenta la expresión (2.19), el término A_j podría ser rescrito como sigue:

$$A_j = \left(\frac{\partial {}^j \dot{\bar{\omega}}_j}{\partial \ddot{q}_j} \right)^T \cdot \sum_{i=j}^n \left\{ {}^j \mathbf{R}_i \cdot \left[{}^i \mathbf{I}_{G_i} \cdot {}^i \dot{\bar{\omega}}_i + {}^i \bar{\omega}_i \times ({}^i \mathbf{I}_{G_i} \cdot {}^i \bar{\omega}_i) \right] \right\} \quad (2.20)$$

En esta última expresión, se puede observar que hay que hay términos concurrentes con lo cual se podría reducir la complejidad de cálculo. A continuación se presenta una expresión que permite obtener los términos de una forma recursiva hacia atrás.

$$A_j = \left(\frac{\partial {}^j \dot{\bar{\omega}}_j}{\partial \ddot{q}_j} \right)^T \cdot {}^j \bar{\alpha}_j \quad (2.21)$$

donde

$${}^j \bar{\alpha}_j = {}^j \mathbf{I}_{G_j} \cdot {}^j \dot{\bar{\omega}}_j + {}^j \bar{\omega}_j \times ({}^j \mathbf{I}_{G_j} \cdot {}^j \bar{\omega}_j) + {}^j \mathbf{R}_{j+1} \cdot {}^{j+1} \bar{\alpha}_{j+1} \quad (2.22)$$

el término B_j se puede obtener usando el desarrollo de $\frac{\partial {}^i \ddot{\bar{r}}_{G_i}}{\partial \ddot{q}_j}$. Este término proviene

derivando la expresión (2.17), obteniendo

$$\frac{\partial {}^i \ddot{\bar{r}}_{G_i}}{\partial \ddot{q}_j} = \frac{\partial {}^i \ddot{\bar{r}}_{O_i}}{\partial \ddot{q}_j} + \frac{\partial {}^i \dot{\bar{\omega}}_i}{\partial \ddot{q}_j} \times {}^i \bar{r}_{O_i, G_i} \quad (2.23)$$

El término $\frac{\partial {}^i \ddot{\bar{r}}_{O_i}}{\partial \ddot{q}_j}$ para articulaciones de revolución se obtiene utilizando la expresión (2.16):

$$\frac{\partial {}^i \ddot{\bar{r}}_{O_i}}{\partial \ddot{q}_j} = {}^i \mathbf{R}_{i-1} \cdot \left(\frac{\partial {}^{i-1} \ddot{\bar{r}}_{O_{i-1}}}{\partial \ddot{q}_j} + \frac{\partial {}^{i-1} \dot{\bar{\omega}}_{i-1}}{\partial \ddot{q}_j} \times {}^{i-1} \bar{r}_{O_{i-1}, O_i} \right) \quad (2.24)$$

De una forma similar, para las articulaciones prismáticas se podría obtener:

$$\frac{\partial {}^i \ddot{\vec{r}}_{O_i}}{\partial \ddot{q}_j} = {}^i \mathbf{R}_{i-1} \cdot \frac{\partial {}^{i-1} \ddot{\vec{r}}_{O_{i-1}}}{\partial \ddot{q}_j} + {}^i \mathbf{R}_{i-1} \cdot \frac{\partial {}^{i-1} \dot{\vec{\omega}}_{i-1}}{\partial \ddot{q}_j} \times {}^{i-1} \vec{r}_{O_{i-1}, O_i} + {}^i \vec{z}_i \cdot \frac{\partial \ddot{q}_i}{\partial \ddot{q}_j} \quad (2.25)$$

Resaltar que hay un término adicional en la expresión (2.25) relacionada con la (2.24), la cual debe ser incluida si la articulación i -ésima es prismática:

$${}^i \vec{z}_i \cdot \frac{\partial \ddot{q}_i}{\partial \ddot{q}_j} \quad (2.26)$$

Para desarrollar el término B_j , considerando las expresiones (2.23) y (2.24), se usan las siguientes expresiones

$$\frac{\partial {}^i \ddot{\vec{r}}_{G_i}}{\partial \ddot{q}_j} = \frac{\partial {}^i \dot{\vec{\omega}}_i}{\partial \ddot{q}_j} \times {}^i \vec{r}_{O_{j-1}, G_i} \quad (2.27)$$

Esta expresión, cuando es substituida en la descripción ya citada de cada término B_j , nos daría:

$$B_j = \sum_{i=j}^n \left\{ \left(\frac{\partial {}^i \dot{\vec{\omega}}_i}{\partial \ddot{q}_j} \right)^T \cdot \left({}^i \vec{r}_{O_{j-1}, G_i} \times m_i {}^i \ddot{\vec{r}}_{G_i} \right) \right\} \quad (2.28)$$

Aplicando las propiedades de los productos vectoriales y usando otra vez la expresión (2.19), tendríamos:

$$B_j = - \left(\frac{\partial {}^j \dot{\vec{\omega}}_j}{\partial \ddot{q}_j} \right)^T \cdot {}^j \vec{\beta}_j \quad (2.29)$$

donde

$${}^j \vec{\beta}_j = \sum_{i=j}^n \left[{}^j R_i \cdot \left(m_i \cdot {}^i \ddot{r}_{G_i} \times {}^i \vec{r}_{O_{j-1}, G_i} \right) \right] \quad (2.30)$$

Esta expresión se podría calcular de una forma recursiva como sigue:

$${}^j \vec{\beta}_j = m_j \cdot {}^j \ddot{r}_{G_j} \times {}^j \vec{r}_{O_{j-1}, G_j} + {}^j \vec{\varphi}_j \times {}^j \vec{r}_{O_j, O_{j+1}} + {}^j R_{j+1} \cdot {}^{j+1} \vec{\beta}_{j+1} \quad (2.31)$$

donde

$${}^j \vec{\varphi}_j = {}^j R_{j+1} \cdot \left(m_{j+1} \cdot {}^{j+1} \ddot{r}_{G_{j+1}} + {}^{j+1} \vec{\varphi}_{j+1} \right) \quad (2.32)$$

2.2.1.3. Algoritmo de complejidad n

Esta formulación nos lleva a un algoritmo para resolver el Problema Dinámico Inverso para robots con una complejidad computacional de $O(n)$. La descripción del algoritmo y el análisis de su complejidad se muestran ahora (notar que a la hora de comparar con otros algoritmos, solo se consideran articulaciones de revolución y la base del robot es fija).

Paso 1.- Cálculo de las matrices de rotación y los vectores de traslación (para robots con únicamente articulaciones de revolución, el vector ${}^{i-1} \vec{r}_{O_{i-1}, O_i}$ puede ser calculado fuera de línea)

For $i = 1, 2, \dots, n$ do:

$${}^{i-1}\mathbf{R}_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 \\ \cos \alpha_i \cdot \sin \theta_i & \cos \alpha_i \cdot \cos \theta_i & -\sin \alpha_i \\ \sin \alpha_i \cdot \sin \theta_i & \sin \alpha_i \cdot \cos \theta_i & \cos \alpha_i \end{bmatrix}$$

$${}^{i-1}\vec{r}_{O_{i-1}, O_i} \leftarrow {}^{i-1}\vec{r}_{O_{i-1}, O_i}$$

Paso 2.- Cálculo de las velocidades y aceleraciones

Inicialización:

$${}^0\ddot{\vec{r}}_{O_0} = [0 \quad 0 \quad g]^T$$

$${}^0\ddot{\vec{r}}_{G_0} = {}^0\ddot{\vec{r}}_{O_0}$$

Hacer:

$${}^1\vec{\omega}_1 = {}^1\vec{z}_1 \cdot \dot{q}_1$$

$${}^1\dot{\vec{\omega}}_1 = {}^1\vec{z}_1 \cdot \ddot{q}_1$$

$${}^1\ddot{\vec{r}}_{O_1} = {}^1\mathbf{R}_0 \cdot {}^0\ddot{\vec{r}}_{O_0}$$

For $i = 2, 3, \dots, n$ do:

$${}^i \vec{u}_i = {}^i R_{i-1} \cdot {}^{i-1} \vec{r}_{i-1}$$

$${}^i \vec{\omega}_i = {}^i \vec{u}_i + {}^i \vec{z}_i \cdot \dot{q}_i$$

$${}^i \dot{\vec{\omega}}_i = {}^i R_{i-1} \cdot {}^{i-1} \dot{\vec{\omega}}_{i-1} + {}^i \vec{z}_i \cdot \ddot{q}_i + {}^i \vec{u}_i \times ({}^i \vec{z}_i \cdot \dot{q}_i)$$

$${}^i \ddot{\vec{r}}_{O_i} = {}^i R_{i-1} \left[{}^{i-1} \ddot{\vec{r}}_{O_{i-1}} + {}^{i-1} \vec{\omega}_{i-1} \times ({}^{i-1} \vec{\omega}_{i-1} \times {}^{i-1} \vec{r}_{O_{i-1}, O_i}) + {}^{i-1} \dot{\vec{\omega}}_{i-1} \times {}^{i-1} \vec{r}_{O_{i-1}, O_i} \right]$$

For $i = 1, 2, \dots, n$ do:

$${}^i \ddot{\vec{r}}_{G_i} = {}^i \ddot{\vec{r}}_{O_i} + {}^i \vec{\omega}_i \times ({}^i \vec{\omega}_i \times {}^i \vec{r}_{O_i, G_i}) + {}^i \dot{\vec{\omega}}_i \times {}^i \vec{r}_{O_i, G_i}$$

Paso 3.- Cálculo de los términos $\frac{\partial {}^i \dot{\vec{\omega}}_i}{\partial \dot{q}_i}$

For $i = 1, 2, \dots, n$ do:

$$\frac{\partial {}^i \dot{\vec{\omega}}_i}{\partial \dot{q}_i} = {}^i \vec{z}_i$$

Paso 4.- Cálculo de los términos A_i .

(En este paso se ha omitido el cálculo de ${}^i R_{i+1} \cdot {}^{i+1} \vec{\alpha}_{i+1}$ y se ha añadido en los cálculos involucrados en el término B , para conseguir ahorrar en el número de

operaciones. Puesto que los términos A_i son escalares y como consecuencia de la notación de D-H, éstas son iguales al tercer componente de los vectores $\vec{\alpha}$. Los cálculos para $i=1$ se pueden simplificar porque solo se necesita ${}^1\alpha_1$. Además, tenemos en cuenta que ${}^1\vec{\omega}_1$ y ${}^1\dot{\vec{\omega}}_1$ tienen las componentes uno y dos iguales a cero)

For $i = n, n-1, \dots, 1$ do:

$${}^i\vec{\alpha}_i = {}^i I_{G_i} \cdot {}^i \dot{\vec{\omega}}_i + {}^i \vec{\omega}_i \times ({}^i I_{G_i} \cdot {}^i \vec{\omega}_i)$$

For $i = 1, 2, \dots, n$ do:

$$A_i = \left(\frac{\partial {}^i \dot{\vec{\omega}}_i}{\partial \dot{q}_i} \right)^T \cdot {}^i \vec{\alpha}_i$$

Paso 5.- Cálculo de los términos B_i .

For $i = 1, 2, \dots, n$ do:

$${}^i \vec{k}_i = m_i \cdot {}^i \ddot{r}_{G_i}$$

Hacer:

$${}^{n-1} \vec{\varphi}_{n-1} = {}^{n-1} R_n \cdot {}^n \vec{k}_n$$

For $i = n-2, n-3, \dots, 1$ do:

$${}^i \vec{\varphi}_i = {}^i R_{i+1} \cdot \left({}^{i+1} \vec{k}_{i+1} + {}^{i+1} \vec{\varphi}_{i+1} \right)$$

Hacer:

$${}^n \vec{\beta}_n = {}^n \vec{k}_n \times {}^n \vec{r}_{O_n, G_n}$$

For $i = n - 1, n - 2, \dots, 1$ do:

$${}^i \vec{\beta}_i = {}^i \vec{k}_i \times {}^i \vec{r}_{O_i, G_i} + {}^i \vec{\phi}_i \times {}^i \vec{r}_{O_i, O_{i+1}} + {}^i \mathbf{R}_{i+1} \cdot ({}^{i+1} \vec{\beta}_{i+1} - {}^{i+1} \vec{\alpha}_{i+1})$$

For $i = 1, 2, \dots, n$ do:

$$B_i = - \left(\begin{array}{c} \partial^i \dot{\vec{\omega}}_i \\ \partial \ddot{q}_i \end{array} \right)^T \cdot {}^i \vec{\beta}_i$$

Paso 6.- Cálculo de las fuerzas generalizadas

For $i = 1, 2, \dots, n$ do:

$$\tau_i = A_i + B_i$$

La complejidad computacional del algoritmo propuesto se resume en la *Tabla 2-2*. Además, la complejidad computacional del algoritmo de Luh, Walker y Paul se presenta en la *Tabla 2-3*.

Tabla 2-2 Complejidad computacional del algoritmo propuesto

Paso	Complejidad	
	(×)	(+)
Paso 1	4n	0
Paso 2	62n-75	46n-61
Paso 3	0	0
Paso 4	24n-23	18n-18
Paso 5	31n-38	26n-37
Paso 6	0	n

Tabla 2-3 Complejidad computacional para robots con $n \geq 3$

Algoritmo	Complejidad		$n = 6$
	(×)	(+)	
Luh, Walker y Paul	(×)	$121n - 112$	614
	(+)	$90n - 82$	458
Algoritmo propuesto	(×)	$121n - 136$	590
	(+)	$91n - 116$	430

Como se puede apreciar de la *Tabla 2-3*, la complejidad computacional del algoritmo propuesto está muy cerca del algoritmo basado en Newton-Euler.

También hay que destacar que la notación de D-H siguiendo la convención de Paul, no tiene grandes diferencias, siendo en este caso la complejidad computacional de $133n - 74$ (\times) y $97n - 71$ (+) para un robot que solo tiene articulaciones de revolución, así como, para robots con seis grados de libertad, la complejidad computacional sería de 724 (\times) y 511 (+).

2.2.2. Obtención de un algoritmo eficiente para la resolución del Problema Dinámico Directo mediante la formulación de Gibbs-Appell

2.2.2.1. Introducción

Como se ha comentado anteriormente, la dinámica de los sistemas robóticos ha sido objetos de investigación durante las últimas décadas, por lo que existe una extensa literatura tanto de la resolución del PDI, visto en el anterior apartado, como para la resolución del PDD.

El que parece ser el primer algoritmo recursivo que resuelve el PDD de robots manipuladores fue desarrollado por (Vereshchangin, 1974) y se basa en el principio de Gauss. En ese mismo trabajo, Vereshchangin utiliza la función de Gibbs, como parte del principio de Gauss para desarrollar un método que resuelve el PDD con una complejidad de orden $O(n^2)$. En 1979 (Armstrong, 1979) desarrolló un método de orden $O(n)$, basado en las ecuaciones de Newton-Euler para modelar el comportamiento dinámico de mecanismos con pares esféricos. Tres años después, (Walker y Orin, 1982) propusieron cuatro métodos para resolver el PDD. El tercer método propuesto era de complejidad $O(n^2)$ y se basaba en el cómputo de la matriz de inercia del sistema mediante el método del cuerpo rígido compuesto. Este método requiere de la resolución de un sistema lineal, por lo que su complejidad total es de orden $O(n^3)$. Este es el método más conocido para la resolución del PDD, en el que previamente se debe obtenerse la matriz de inercias y los

vectores de fuerzas centrífugas, coriolis y gravitatorias, ya que, el proceso de obtención de las aceleraciones de nudos se realiza mediante la obtención, planteamiento y resolución de un sistema de ecuaciones simultáneas.

2.2.2.2. Aplicación al PUMA 560 del Algoritmo de Resolución del PDD

En esta sección se ha desarrollado el problema dinámico directo mediante las ecuaciones de Gibbs-Appell, utilizando el método del Sólido Rígido Compuesto propuesto por Walker y Orin (Walker y Orin, 1982). Se ha obtenido un algoritmo que simula el comportamiento dinámico de robots industriales, mediante expresiones que deben integrarse en función al tiempo. Dicha ecuación de movimiento tiene la siguiente forma general:

$$M(\bar{q}) \cdot \ddot{\bar{q}} + h(\bar{q}, \dot{\bar{q}}) + \bar{n}(\bar{q}) = \bar{\tau} \quad (2.33)$$

Siendo:

$$M(\bar{q}) \quad \text{Matriz de Inercias del sistema} \equiv M(q, \theta)$$

$$\bar{h}(\bar{q}, \dot{\bar{q}}) \quad \text{términos centrífugos y de coriolis} \equiv C(q, \dot{q}, \theta)$$

$$\bar{n}(\bar{q}) \quad \text{términos gravitatorios} \equiv G(q, \theta)$$

$$\bar{\tau} \quad \text{fuerzas generalizadas}$$

Esta ecuación suele presentarse de la siguiente forma

$$M(\bar{q})\ddot{\bar{q}} + \bar{b}(\bar{q}, \dot{\bar{q}}) = \bar{\tau} \quad (2.34)$$

donde el vector $\vec{b}(\vec{q}, \dot{\vec{q}})$, conocido como el vector Bias, agrupa los términos centrífugos, coriolis y gravitatorios.

Obtención del Vector Bias

El vector *bias* se obtiene al resolver el problema dinámico inverso anulando las aceleraciones de nudo ($\ddot{\vec{q}} = 0$). Por lo tanto substituyendo en la ecuación (2.33) nos quedaría: $\vec{b}(\vec{q}, \dot{\vec{q}}) = \vec{h}(\vec{q}, \dot{\vec{q}}) + \vec{n}(\vec{q})$

De esta manera la complejidad computacional del vector *bias* es:

$90n-77$ sumas

$121n-77$ productos

Así para un manipulador de seis grados de libertad se tienen por lo tanto 463 sumas y 649 productos.

Obtención del Vector de Gravedad

El vector *de gravedad* se obtiene al resolver el problema dinámico inverso anulando las aceleraciones y velocidades de nudo ($\ddot{\vec{q}} = 0, \dot{\vec{q}} = 0$). Por lo tanto substituyendo en la ecuación (2.33) nos quedaría: $\vec{n}(\vec{q})$

Por lo tanto la complejidad computacional sería como en el apartado anterior de $O(n)$.

Obtención de la Matriz de Inercias del sistema

Una vez calculado el vector *Bias*, de una forma eficiente, gracias a las ecuaciones de la dinámica inversa de Gibbs-Appell, faltaría obtener el tensor de inercia para poder resolver el problema dinámico directo, tal y como se presentó anteriormente.

Para la obtención del tensor de inercia generalizado se han seguido los siguientes pasos:

Paso 1: Cálculo del vector de masa compuesta:

$$M_j = m_j + M_{j+1} \quad j = n, n-1, \dots, 1$$

siendo $M_{n+1} = 0$ y m_j la masa de la articulación correspondiente.

Paso 2: Cálculo del centro de gravedad del cuerpo compuesto:

Previamente hay que hallar el vector ${}^j\vec{r}_{O_{j-1},G_j}$. Al ser el par de revolución, el vector es fijo por lo que se calcula fuera de línea.

$${}^n\vec{r}_{O_{n-1},C_n} = {}^n\vec{r}_{O_{n-1},G_n}$$

$${}^i\vec{r}_{O_{j-1},C_j} = \frac{1}{M_j} \left[m_j \cdot {}^j\vec{r}_{O_{j-1},G_j} + M_{j+1} \cdot \left({}^j\vec{r}_{O_{j-1},O_j} + {}^{j+1}\vec{r}_{O_j,C_{j+1}} \right) \right] \quad j = n-1, \dots, 1$$

Paso 3: Tensor de inercia compuesto:

$${}^nE_n = {}^nI_{G_n}$$

$$\begin{aligned}
 {}^j E_j &= {}^j R_{j+1} \cdot {}^{j+1} E_{j+1} \cdot ({}^j R_{j+1})^T + \\
 &+ M_{j+1} \cdot \left[({}^j \vec{r}_{O_{j-1}, O_j})^T \cdot {}^j \vec{r}_{O_{j-1}, O_j} \cdot I_{3 \times 3} - {}^j \vec{r}_{C_j, C_{j+1}} \cdot ({}^j \vec{r}_{C_j, C_{j+1}})^T \right] + \\
 &+ {}^j I_{G_j} + m_j \cdot \left[({}^j \vec{r}_{C_j, G_j})^T \cdot {}^j \vec{r}_{C_j, G_j} \cdot I_{3 \times 3} - {}^j \vec{r}_{C_j, G_j} \cdot ({}^j \vec{r}_{C_j, G_j})^T \right] \quad j = n-1, \dots, 1
 \end{aligned}$$

donde:

$${}^j \vec{r}_{C_j, C_{j+1}} = {}^j \vec{r}_{O_{j-1}, O_j} + {}^j R_{j+1} \cdot {}^{j+1} \vec{r}_{O_j, C_{j+1}} - {}^j \vec{r}_{O_{j-1}, C_j} \quad j = n-1, \dots, 1$$

$${}^n \vec{r}_{C_n, C_{n+1}} = [0 \quad 0 \quad 0]^T$$

igualmente:

$${}^j \vec{r}_{C_j, G_j} = {}^j \vec{r}_{O_{j-1}, O_j} + {}^j \vec{r}_{O_j, G_j} - {}^j \vec{r}_{O_{j-1}, C_j} \quad j = n-1, \dots, 1$$

$${}^n \vec{r}_{C_n, G_n} = [0 \quad 0 \quad 0]^T$$

Paso 4: Diagonal principal del tensor de inercia generalizado:

$$D_{ii} = M_i \cdot ({}^i \ddot{\vec{r}}_{C_i})^T \cdot \left(\frac{\partial {}^i \ddot{\vec{r}}_{C_i}}{\partial \ddot{q}_i} \right) + ({}^i \dot{\vec{w}}_i)^T \cdot {}^i E_i \cdot \left(\frac{\partial {}^i \dot{\vec{w}}_i}{\partial \dot{q}_i} \right)$$

siendo:

$${}^i \dot{\vec{w}}_i = Q_i^T \left[{}^{i-1} \dot{\vec{\omega}}_{i-1} + \vec{z}_0 \cdot \ddot{q}_i + {}^{i-1} \vec{\omega}_{i-1} \times (\vec{z}_0 \cdot \dot{q}_i) \right]$$

$$\frac{\partial^i \dot{\tilde{\omega}}_i}{\partial \dot{q}_i} = Q_i \cdot \bar{z}_0$$

$${}^i \ddot{\tilde{r}}_{C_i} = {}^i \dot{\tilde{\omega}}_i \times {}^i \tilde{r}_{O_{i-1}, C_i}$$

$$\frac{{}^i \ddot{\tilde{r}}_{C_i}}{\partial \ddot{q}_i} = {}^i \dot{\tilde{\omega}}_i \times {}^i \tilde{r}_{O_{i-1}, C_i}$$

Paso 5: Términos restantes del tensor de inercia generalizado:

$$D_{ij} = M_{j,i} \cdot ({}^j \ddot{\tilde{r}}_{C_j})^T \cdot \left(\frac{\partial^j \ddot{\tilde{r}}_{C_j}}{\partial \ddot{q}_j} \right) + ({}^i \dot{\tilde{\omega}}_i)^T \cdot {}^j E_j \cdot \left(\frac{\partial^j \dot{\tilde{\omega}}_j}{\partial \ddot{q}_i} \right)$$

siendo:

$$\frac{\partial^j \dot{\tilde{\omega}}_j}{\partial \dot{q}_i} = Q_{i-1} \cdot \bar{z}_0$$

$$\frac{{}^j \ddot{\tilde{r}}_{C_j}}{\partial \ddot{q}_i} = \frac{{}^i \dot{\tilde{\omega}}_j}{\partial \ddot{q}_i} \times {}^i \tilde{r}_{O_{i-1}, C_j}$$

a su vez:

$${}^j \tilde{r}_{O_{i-1}, C_j} = {}^j \tilde{r}_{O_{i-1}, O_{j-1}} + {}^j \tilde{r}_{O_{j-1}, C_j}$$

Para un manipulador de seis grados de libertad se tienen por lo tanto 773 sumas y 941 productos.

Complejidad total:

La complejidad del problema dinámico directo completo es:

$$\frac{10n^2 + 173n - 162(+)}{2} + \frac{25n^2 + 437n - 342}{2}(x)$$

Para un manipulador de seis grados de libertad se tienen por lo tanto 1236 sumas y 1590 productos.

2.3. EMPLEO DE LOS PARÁMETROS IDENTIFICADOS EXPERIMENTALMENTE

En el apartado anterior se han desarrollado tanto un algoritmo aplicable al control en tiempo real de un robot industrial del tipo PUMA, como otro a su simulación. Estos algoritmos son útiles en el caso de conocer de forma fiable los parámetros físicos del robot. Dichos parámetros se pueden agrupar en:

- Parámetros geométricos.
- Parámetros dinámicos.

Se entienden por parámetros geométricos los constantes de la tabla D-H asociada al robot. Indicar que existe un nivel de información razonable (fabricantes de robots, literatura científica) acerca de estos parámetros. En cualquier caso en la presente tesis se han empleado los datos obtenidos en (Besa, 2002) mediante técnicas de calibración cinemática.

En el caso de los parámetros dinámicos se incluyen los inerciales (masas, localización de centros de gravedad y tensores de inercia) de los sólidos que forman el robot y coeficientes de fricción. Mencionar que a diferencia de lo que ocurre con el caso de los parámetros geométricos la información disponible en el caso de los dinámicos es sumamente escasa e incluso contradictoria, véase (Corke, Armstrong-Hélouvy, 1995 y Corke y Armstrong-Hélouvy 1994) por ello en la presente tesis se ha tenido en cuenta la posibilidad de utilizar parámetros identificados, para ello se han empleado resultados obtenidos en (Benimeli, 2006) mediante técnicas de identificación de parámetros dinámicos. Hay que mencionar que en el presente trabajo y partiendo de desarrollos previos del citado autor, se adaptarán tanto a la resolución del PDI como a la obtención de los términos de la ecuación del movimiento con vistas a la resolución del PDD o a la incorporación de alguno de sus términos a esquemas de control, ya que, el conjunto de parámetros identificados no se pueden utilizar en la resolución tanto del PDI como del PDD puesto que son combinaciones lineales de los físicos.

2.3.1. Aplicación de la resolución del PDI y PDD en estrategias de control

De un modo general y como se ha comentado en el apartado anterior, la ecuación del movimiento que gobierna el movimiento de un sistema mecánico multicuerpo puede expresarse en la forma:

$$M(\bar{q}, \dot{\bar{q}}, \ddot{\bar{q}}) \cdot \ddot{\bar{q}} + \bar{C}(\bar{q}, \dot{\bar{q}}) \cdot \dot{\bar{q}} + G(\bar{q}) = \bar{\tau} \quad (2.35)$$

donde $M(\bar{q}, \dot{\bar{q}}, \ddot{\bar{q}})$ es la matriz de masas del sistema que depende (además de la geometría) de las coordenadas generalizadas \bar{q} empleadas para modelizar el mecanismo, $\bar{C}(\bar{q}, \dot{\bar{q}})$ es el vector que agrupa las fuerzas centrífugas y de Coriolis, y depende además

de las velocidades generalizadas \vec{q} , $G(\vec{q})$ son los términos gravitaciones, mientras que $\vec{\tau}$ es el vector de fuerzas generalizadas.

Como se indica en (Benimeli, 2006) la ecuación (2.35) se puede expresar para cualquier sistema mecánico multicuerpo como el siguiente sistema de ecuaciones lineal respecto a los parámetros dinámicos y reescrito en forma matricial.

$$\vec{\tau} = W(\vec{q}, \dot{\vec{q}}, \ddot{\vec{q}}) \cdot \Phi \quad (2.36)$$

siendo \mathbf{W} la denominada matriz de observación del sistema. La representación del modelo dinámico (2.36) es redundante en dichos parámetros dinámicos, esto es, existen infinitos conjuntos de parámetros dinámicos que satisfacen las ecuaciones de la dinámica. En definitiva, el sistema lineal (2.36) presenta un problema de deficiencia de rango en la matriz de coeficientes \mathbf{W} .

Dicha deficiencia de rango está motivada por el hecho que las restricciones cinemáticas que las uniones entre las partes mecánicas imponen sobre la estructura del robot manipulador hacen que algunos de los parámetros inerciales no afecten en absoluto al comportamiento dinámico del manipulador, mientras que otros lo hagan tan sólo en combinaciones lineales, no siendo por tanto identificables de forma independiente. Por ello, será necesario proceder a la reducción del modelo, previamente a la resolución del sistema de ecuaciones en (2.36), a fin de que tan sólo aparezca en él un conjunto mínimo de parámetros que identifiquen totalmente y de forma independiente el comportamiento del sistema mecánico, obteniéndose así una matriz de observación de rango completo.

En (Benimeli, 2006) se ha procedido a la determinación de dicho conjunto mínimo de parámetros inerciales, denominados parámetros base, desafortunadamente, dichos parámetros no son directamente aplicables directamente en la implementación de los controladores por dinámica inversa, dichos controladores están basados en la resolución del PDI utilizando el algoritmo propuesto en el apartado anterior. En el caso de tener que

realizar el control por dinámica inversa cuando no son conocidos los parámetros dinámicos, se tendrá que utilizar los parámetros identificados. Por tanto, en lugar de utilizar la expresión de la ecuación general del movimiento (2.35), se utilizará la siguiente expresión:

$$\vec{\tau} = W^* (\vec{q}, \dot{\vec{q}}, \ddot{\vec{q}}) \cdot \Phi^* \quad (2.37)$$

donde $\vec{\tau}$ es el vector de fuerzas generalizadas, W^* es la matriz de observación base y Φ^* es el vector base de los parámetros inerciales, vector que para este caso es:

$$\Phi^* = [{}^n\vec{I}_n \quad \dots \quad {}^1\vec{I}_1 \quad m_n \cdot {}^n\vec{r}_{O_n G_n} \quad m_n \quad \dots \quad m_1 \cdot {}^1\vec{r}_{O_1 G_1} \quad m_1]^T \quad (2.38)$$

Cuando se realiza la implementación de controladores pasivos en los cuales se necesita la obtención de algún elemento de la ecuación general del movimiento (2.35), se necesita resolver el problema dinámico directo, al ser necesario aplicar el control en tiempo real, se resuelve el problema dinámico directo en tiempo real, si no se conoce el conjunto de parámetros dinámicos y se tienen que utilizar los parámetros identificados por lo que sería necesario utilizar la expresión (2.37).

2.3.1.1. Cálculo del problema dinámico directo del robot en tiempo real para las estrategias de Control

Como se ha visto anteriormente, a partir de la matriz de observación del sistema y de la identificación del vector base de los parámetros inerciales se puede obtener el vector de pares generalizados del sistema (2.37).

A partir de esta expresión se puede plantear el cálculo de la ecuación dinámica del sistema robotizado (2.35)

Sabiendo que a partir de la obtención de la matriz de observación W y del vector base de parámetros inerciales $\vec{\Phi}$, se hallan los elementos de la ecuación general del movimiento.

Para ello se ha propuesto un algoritmo secuencial en el que se han ido obteniendo estos elementos, este algoritmo se basa en los algoritmos de Luh-Walker y Paul (Luh, Walker and Paul, 1980) para resolver el PDI y de Walter y Orin (Walker and Orin, 1982) para resolver el PDD.

Analizando los términos gravitacionales $G(\vec{q})$ se puede comprobar que éstos dependen únicamente de las coordenadas generalizadas \vec{q} . Por ello se puede plantear la obtención de dichos términos inicializando a cero las velocidades y aceleraciones generalizadas ($\dot{\vec{q}} = 0$ y $\ddot{\vec{q}} = 0$) de la matriz de observación W de (2.36).

De esta manera si sustituimos en (2.36) obtendremos $\vec{\tau}^*$ y si sustituimos en (2.35) obtendremos $G(\vec{q}) = \vec{\tau}$, al ser ambas la misma ecuación pero escritas de forma diferente se podría hallar el término Gravitatorio que posteriormente nos servirá para calcular el resto, ya que, como se ha comentado anteriormente este término depende únicamente de las coordenadas generalizadas \vec{q} .

$$W(\vec{q}, 0, 0) \cdot \vec{\Phi} = \vec{G}(\vec{q}) \quad (2.39)$$

Para hallar los términos centrífugos y de Coriolis $C(\vec{q}, \dot{\vec{q}})$ podemos observar que éstos dependen tanto de las coordenadas como de las velocidades generalizadas. Por ello, si se inicializan a cero las aceleraciones generalizadas en la ecuación (2.36):

$$W(\vec{q}, \dot{\vec{q}}, 0) \cdot \vec{\Phi} = \vec{C}(\vec{q}, \dot{\vec{q}}) + \vec{G}(\vec{q}) \quad (2.40)$$

De esta forma se puede obtener $C(\vec{q}, \dot{\vec{q}})$ restando directamente las ecuaciones (2.40) y (2.39):

$$(W(\vec{q}, \dot{\vec{q}}, 0) \cdot \vec{\Phi} - \vec{G}(\vec{q})) = \vec{C}(\vec{q}, \dot{\vec{q}}) \quad (2.41)$$

El último término que falta obtener es la matriz de inercias. Éste depende, además del vector que agrupa los parámetros inerciales $\vec{\Phi}$, de la geometría de las coordenadas generalizadas \vec{q} empleadas para modelizar el mecanismo.

Para obtener la matriz de inercias se pueden utilizar los términos gravitacionales y de Coriolis que se acaban de calcular:

$$M(\vec{q}) \cdot \ddot{\vec{q}} = \vec{\tau} - \vec{G}(\vec{q}) - \vec{C}(\vec{q}, \dot{\vec{q}}) \quad (2.42)$$

Sin embargo, hay que tener en cuenta que $M(\vec{q})$ es una matriz 6x6 y no un vector, por lo que no se puede obtener directamente como se ha hecho en los términos anteriores. Para poder solucionar el problema se pueden obtener las columnas independientemente mediante el siguiente cálculo:

$$\begin{bmatrix} m_{11} & m_{12} & \cdots & m_{16} \\ \vdots & \ddots & & \vdots \\ m_{61} & m_{62} & \cdots & m_{66} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} m_{11} \\ m_{21} \\ \vdots \\ m_{61} \end{bmatrix}$$

Así, para poder obtener la matriz completa lo que se hace es calcular la columna i -ésima de dicha matriz, uniéndolas posteriormente:

$$(W(\vec{q}, 0, \vec{e}_i) \cdot \vec{\Phi} - \vec{G}(\vec{q})) = \vec{M}_i(\vec{q}) \quad (2.43)$$

siendo $\vec{M}_i(\vec{q})$ la columna i de la matriz de inercias y $\vec{e}_i = [0 \dots 1 \dots 0]^T$ un vector columna donde la posición i será 1 e indicará la columna a hallar.

Una vez se han hallado todas las columnas de la matriz de Inercias, se obtendría la matriz completa $M(\vec{q})$.

De esta forma se ha obtenido un algoritmo capaz de obtener cada uno de los elementos de la ecuación general del movimiento, para su posterior utilización en diferentes estrategias de control realizadas en tiempo real.

CAPÍTULO 3

CONTROL CINEMÁTICO DE SISTEMAS ROBOTIZADOS

3.1. INTRODUCCIÓN

En esta sección se ha llevado a cabo la descripción del control cinemático y los diferentes tipos de trayectorias implementadas para la realización de diferentes estrategias de control de robots manipuladores.

El control cinemático permite establecer las trayectorias que debe seguir cada articulación del robot a lo largo del tiempo para lograr los objetivos que se fijen (puntos de destino en el espacio de nudos, trayectorias cartesianas, identificación de los parámetros inerciales etc). Estas trayectorias se seleccionan atendiendo a las restricciones físicas propias del robot industrial y a ciertos criterios de calidad como precisión o continuidad.

Se puede esquematar el funcionamiento del control cinemático del siguiente modo:

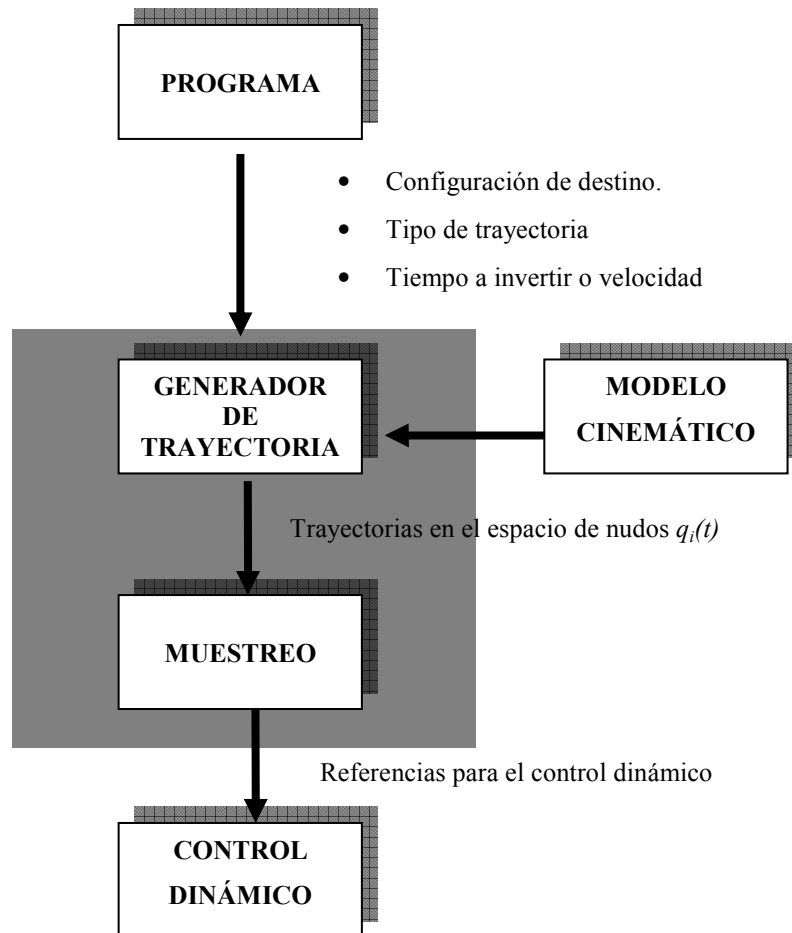


Figura 3-1 Funcionamiento del control cinemático

Como se observa en la figura anterior, el control cinemático recibe como entrada los datos procedentes del programa del robot escrito (puntos de destino, tipo de trayectoria, velocidad) y apoyándose en el modelo cinemático del robot, ya visto en el capítulo anterior, se establecen las trayectorias para cada articulación como funciones de

tiempo. Estas trayectorias se muestrean con un período T (en esta tesis el período de muestreo es de 10 ms), generándose en cada instante kT un vector de referencias articulares para los algoritmos de control dinámico implementados que se verán en el siguiente capítulo.

De una manera general el control cinemático realiza las siguientes funciones:

- Muestrea la trayectoria cartesiana obteniendo un número finito de puntos de dicha trayectoria. Cada uno de estos puntos vendrá dado por la 6-tupla $(x, y, z, \alpha, \beta, \gamma)$, es decir, tres coordenadas cartesianas y tres ángulos de Euler.
- Resolviendo el Problema Cinemático Inverso, visto también en el capítulo anterior, se convierte cada uno de esos puntos en sus correspondientes variables de nudo $[q_1, q_2, \dots, q_6]^t$. Como se ha comentado en el anterior capítulo se ha tenido en cuenta la posible solución múltiple de la resolución del Problema Cinemático Inverso, y la selección de solo una de las posibles configuraciones.
- Se interpola los puntos en el espacio de nudos obtenidos, generando para cada variable articular una expresión $q_i(t)$ que pase por ellos.
- Se muestrea de la trayectoria articular para generar referencias al control dinámico.

A la hora de implementar este control en la presente tesis se ha tenido únicamente en cuenta el hecho de tener que pasar de un punto (llámese A) a otro (llámese B) en un tiempo determinado. Sin más restricciones que el tiempo para pasar entre ambos puntos, y las posiciones y orientaciones inicial y final. En este trabajo se han generado las trayectorias directamente en el espacio de nudos. En el caso de que sea necesario

establecer puntos de paso entre los puntos de inicio y fin, al igual que en el caso anterior las trayectorias se generan en el espacio de nudos directamente.

También existe la posibilidad de describir una determinada trayectoria, en este caso el movimiento deberá ser establecido en el espacio cartesiano, lo que ocurre es que la generación de trayectorias no son tan sencillas dado que presentan problemas con la existencia de múltiples soluciones o la ausencia de las mismas.

Como se ha mencionado anteriormente, en la presente tesis sólo se especifican las posiciones y orientaciones de los puntos de inicio y fin, e incluso se da la opción de especificar puntos intermedios, sin importar el camino descrito por el extremo del robot, se puede decir que las trayectorias a implementar serán sencillas, una de las razones es porque no existe ninguna limitación para la trayectoria descrita por el extremo del robot, así que cada articulación se puede mover sin tener en cuenta el movimiento del resto, es decir, de forma totalmente independiente.

En este caso el método empleado consta de los siguientes pasos:

- Conversión con ayuda de la cinemática inversa de los puntos cartesianos inicial, final e intermedios a valores en el espacio de nudos para cada una de las articulaciones.
- Interpolación de los valores articulares para cada articulación de forma sencilla garantizando la máxima suavidad. Puede ser con o sin puntos intermedios.
- Movimiento de cada articulación de forma independiente según la trayectoria establecida para ella.

3.2. TIPOS DE TRAYECTORIAS

3.2.1. Trayectorias Punto a Punto

En este tipo de trayectorias cada articulación evoluciona desde su posición inicial a la final sin realizar consideración alguna sobre el estado o evolución de las demás articulaciones. Normalmente, cada actuador trata de llevar a su articulación al punto de destino en el menor tiempo posible, pudiéndose distinguir dos casos: movimiento eje a eje y movimiento simultáneo de ejes.

En este tipo de trayectorias una vez la articulación ha alcanzado su posición final se debe mantener en esa posición hasta que llegue la última, al no poder frenar cada articulación independientemente se debe mantener activo el control para que se mantenga la referencia.

3.2.1.1. Movimiento eje a eje

En este caso sólo se mueve un eje cada vez. Comenzará a moverse la primera articulación, y una vez que ésta haya alcanzado su posición final lo hará la segunda, y así sucesivamente. Para realizar este tipo de movimiento las posiciones se deben mantener tanto en la posición inicial, hasta que le toque su turno, como en la posición final, hasta que finalice el movimiento la sexta articulación. Este tipo de movimiento da obviamente un mayor tiempo de ciclo.

3.2.1.2. Movimiento simultáneo de los ejes

Cuando se realizar este tipo de movimiento los actuadores comienzan simultáneamente a mover las articulaciones a una velocidad específica para cada una de

ellas. Dado que la distancia a recorrer y las velocidades serán diferentes, cada una acabará su movimiento en un instante diferente.

El movimiento completo del robot no acabará hasta que se alcance definitivamente el punto final, lo que se producirá cuando el eje más lento alcance su posición. De esta manera, el tiempo total invertido en el movimiento coincidirá con el del eje que más tiempo emplee en realizar su movimiento. Se puede dar el caso de que el resto de actuadores hayan forzado su movimiento a una velocidad y aceleración elevada, viéndose obligados finalmente a esperar a la articulación más lenta manteniendo su posición final, puesto como ya se ha comentado anteriormente, no se pueden frenar las articulaciones independientemente y el control continuaría, siendo la posición referencia la posición final alcanzada.

En esta tesis aunque se implementaron las trayectorias punto a punto, no se han utilizado para implementar el control en tiempo real del robot.

3.2.2. Trayectorias coordinadas o isócronas

Finalmente se han implementado y utilizado este tipo de trayectorias, una de las razones ha sido para evitar que algunos actuadores trabajen forzando sus velocidades y aceleraciones, teniendo que esperar después la finalización del movimiento de la articulación más lenta.

Lo que se ha realizado en la presente tesis ha sido un cálculo previo fuera del bucle de control para averiguar cual será la articulación más lenta y qué tiempo invertirá. En este caso se ha considerado la misma velocidad para todas las articulaciones, por lo que, la articulación más lenta será aquella que recorra una mayor distancia.

Una vez conocido el tiempo de la articulación más lenta, lo que se ha hecho ha sido relantizar el movimiento del resto de los ejes para que inviertan el mismo tiempo en

su movimiento, acabando todos ellos simultáneamente. Se tiene así que todas las articulaciones se coordinan comenzando y acabando su movimiento a la vez, adaptándose todas a la más lenta.

3.3. INTERPOLACIÓN DE TRAYECTORIAS

Como se ha indicado en la sección anterior, una de las funciones del control cinemático es la de unir una sucesión de puntos en el espacio de nudos por lo que se quiere que pasen las articulaciones del robot en un instante dado. Además, junto con las condiciones de posición-tiempo, es conveniente añadir restricciones en la velocidad y aceleración de paso por los puntos, de manera que se asegure la suavidad de la trayectoria, generalmente la suavidad se expresa matemáticamente a través de la continuidad en la primera derivada, incluso en ocasiones en la segunda derivada; y se limiten las velocidades y aceleraciones máximas. Estas restricciones garantizarán que los actuadores sean capaces de implementar la trayectoria final.

Para ello se ha seleccionado funciones polinómicas cuyos coeficientes se ajustarán al imponer las condiciones de contorno: posiciones, velocidades y aceleraciones. Además estos cálculos de los coeficientes, como su posterior utilización para generar puntos de referencia al control dinámico, se realizan en tiempo real, por lo que serán funciones simples en el cálculo.

A continuación se presentan las funciones interpoladoras que se han implementado en esta tesis.

3.3.1. Generación de curvas mediante la Interpolación de Hermite

Este tipo de interpolación se ha implementado para realizar la implementación de una trayectoria articular con los puntos de inicio y fin.

Como se ha comentado sólo se ha considerado que existen los puntos inicial y final, es decir, no hay puntos intermedios. Al no existir restricción en cuanto la forma de la trayectoria para pasar de un punto inicial a otro final las posibles soluciones son diversas. Sin embargo, para conseguir movimientos suaves, al menos se imponen cuatro condiciones de contorno, las dos primeras aseguran que se comienza y termina en los puntos adecuados:

$$q(t_{inicio}) = q(0) = q_{inicio} \quad (3.1)$$

$$q(t_{fin}) = q_{fin} \quad (3.2)$$

y las dos siguientes garantizan que las velocidades de inicio y final son nulas:

$$\dot{q}(t_{inicio}) = \dot{q}(0) = 0 \quad (3.3)$$

$$\dot{q}(t_{fin}) = 0 \quad (3.4)$$

A partir de estas condiciones se pueden emplear diversos métodos de interpolación, en este trabajo se ha empleado la Interpolación de Hermite.

Para realizar esta interpolación, se ha partido del hecho de que se supone que se tienen $m + 1$ puntos: P_0, P_1, \dots, P_m por los que se desea dibujar una curva que los interpole, de forma que cada par de puntos sucesivos se conecta por un segmento de curva distinto.

El segmento i -ésimo va de P_i a P_{i+1} , y el parámetro t , tiempo, varía entre el valor t_i a t_{i+1} . Como cada segmento $r_i(t)$ se representa paramétricamente como $((X_i(t), Y_i(t)))$, lo que se hace en este tipo de interpolación es averiguar como se determinan las funciones a partir de los puntos $P_i = (x_i, y_i)$.

Para obtener la interpolación de Hermite se considera el siguiente polinomio cúbico:

$$Y_i(t) = a_i + b_i t + c_i t^2 + d_i t^3 \quad (3.5)$$

Para cada segmento se conoce la coordenada y del punto inicial del segmento, y_i , que viene dada por:

$$Y_i(0) = y_i = a_i \quad (3.6)$$

La coordenada y del punto inicial del segmento, y_{i+1} , viene dada por:

$$Y_i(1) = y_{i+1} = a_i + b_i + c_i + d_i \quad (3.7)$$

Como se tienen cuatro coeficientes para determinar completamente $Y_i(t)$: a_i, b_i, c_i, d_i , se requieren otras dos ecuaciones para poder resolverla. La forma en que se realiza con la interpolación de Hermite consiste en seleccionar arbitrariamente las primeras derivadas D_i de $Y(t)$ en cada nodo t . Se sabe que estas derivadas tienen un significado concreto en el modelado de trayectorias para sistemas robotizados y sus características son las que permiten definir los valores de los coeficientes que se tienen que utilizar.

La derivada de $Y(t)$ en el punto inicial del segmento, D_i , viene dada por:

$$\dot{Y}_i(0) = D_i = b_i \quad (3.8)$$

La derivada de $Y(t)$ en el punto final del segmento, D_{i+1} , viene dada por:

$$\dot{Y}_i(1) = D_{i+1} = b_i + 2c_i + 3d_i \quad (3.9)$$

Dándole valores a D_i y a D_{i+1} , ya se está en condiciones de resolver el sistema.

Como se ha utilizado D_i como la derivada en el extremo izquierdo del segmento i -ésimo (como $\dot{Y}_i(0)$) y como la derivada en el extremo derecho del segmento $(i-1)$ -ésimo (como $\dot{Y}_{i-1}(1)$), la función global $Y(t)$ tiene la primera derivada continua.

A la hora de implementar, en tiempo real, esta curva es importante definir un tiempo t durante el cual se genera la curva. Este tiempo se reparametriza en el rango $[0,1]$ haciendo el siguiente cambio:

$$t^* = \frac{t - t_i}{t_{i+1} - t_i} \quad (3.10)$$

O lo que es lo mismo, para implementar la interpolación de Hermite considerando el polinomio (3.5), se ha resuelto la ecuación planteada definiendo el valor de la derivada de $Y(t)$ en los puntos inicial y final.

Como es sabido la derivada de $Y(t)$ representa a la velocidad, por lógica, se considera que la velocidad al inicio del movimiento es cero, igual que al final del movimiento. Sabiendo esto, los coeficientes quedan determinados de la siguiente forma:

$$a_i = y_i \text{ posición inicial} \quad (3.11)$$

$$b_i = 0 \text{ ya que la velocidad inicial es } 0 \quad (3.12)$$

$$c_i = 3(y_i + y_{i+1}) \quad (3.13)$$

$$d_i = 2(y_i - y_{i+1}) \quad (3.14)$$

Así pues, la trayectoria generada tendrá el siguiente aspecto:

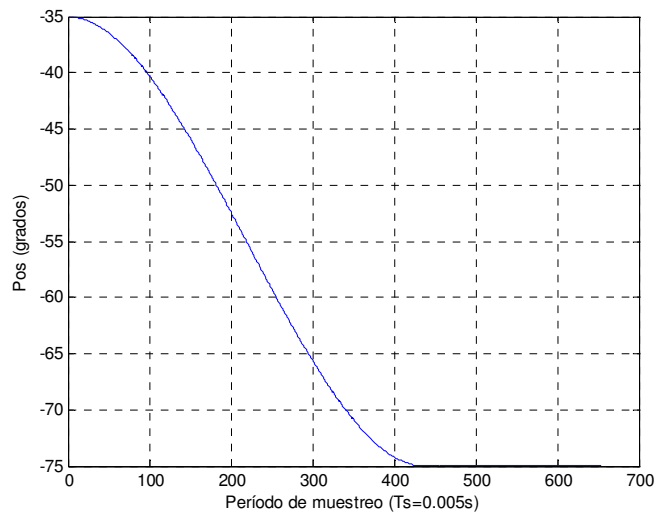


Figura 3-2 Curva mediante Interpolación de Hermite

Viendo la forma de la trayectoria se puede pensar que el sistema realiza un seguimiento de la referencia preciso, ya que se comienza con incrementos de posición pequeños y se van aumentando éstos para finalmente, terminar también con incrementos de posición pequeños, con lo que la referencia es lo más parecido al comportamiento del sistema real. Esta forma de entrada hace que la velocidad también posea una forma de

campana, es decir, se comienza con poca velocidad para ir incrementándola hasta que al aproximarse a la posición final, se vuelve a decrementar, hay una aceleración inicial progresiva y un frenado al ir llegando a la posición final. Lo anterior hará que efectos tan negativos como son la zona muerta y la inercia se puedan llegar a obviar.

3.3.2. Generación de trayectorias en el espacio de nudos del robot mediante un interpolador cúbico

En ocasiones, además de los puntos inicial y final de la trayectoria, también se dan referencias de puntos de paso. En principio se consideró el método anterior tomando cada punto de paso como fin de un tramo y comienzo del siguiente; sin embargo esto supone que cada articulación se pare justo en los puntos de paso (condiciones de velocidad nula, ecuaciones 3.3 y 3.4), sin que esta circunstancia sea necesaria.

Este tipo de interpolación se puede usar sustituyendo las condiciones de contorno (3.3) y (3.4) por otras que expresen cierto valor de velocidad articular conocido para los puntos inicial y final de la trayectoria:

$$\dot{q}(t_{inicial}) = \dot{q}_{inicial} \quad (3.15)$$

$$\dot{q}(t_{final}) = \dot{q}_{final} \quad (3.16)$$

Las cuatro condiciones de contorno de las expresiones (3.1), (3.2), (3.3) y (3.4), que permitiría una interpolación cúbica de una trayectoria en el espacio de nudos con puntos inicio y fin, se pueden satisfacer con un polinomio de grado 3:

$$q(t) = a + b \cdot t + c \cdot t^2 + d \cdot t^3 \quad (3.17)$$

Para que cumpla la condición (3.1), teniendo en cuenta la ecuación (3.17):

$$q(t_{\text{inicio}}) = q(0) \Rightarrow a = q_{\text{inicio}} \quad (3.18)$$

y para la condición (3.2):

$$q(t_{\text{fin}}) = a + b \cdot t_{\text{fin}} + c \cdot t_{\text{fin}}^2 + d \cdot t_{\text{fin}}^3 = q_{\text{fin}} \quad (3.19)$$

Si se deriva con respecto al tiempo la expresión (3.17) se tiene:

$$\dot{q}(t) = b + 2 \cdot c \cdot t + 3 \cdot d \cdot t^2 \quad (3.20)$$

y teniendo en cuenta las ecuaciones (3.3) y (3.4):

$$\dot{q}(t_{\text{inicio}}) = \dot{q}(0) = b \Rightarrow b = 0 \quad (3.21)$$

$$\dot{q}(t_{\text{fin}}) = 2 \cdot c \cdot t_{\text{fin}} + 3 \cdot d \cdot t_{\text{fin}}^2 = 0 \quad (3.22)$$

Por tanto, una vez conocidos los valores a y b, quedan dos ecuaciones con dos incógnitas para determinar c y d, las correspondientes a las expresiones (3.19) y (3.22).

Una vez resueltas se tiene:

$$\begin{aligned} a &= q_{\text{inicio}} \\ b &= 0 \\ c &= \frac{3(q_{\text{fin}} - q_{\text{inicio}})}{t_{\text{fin}}^2} \\ d &= \frac{-2(q_{\text{fin}} - q_{\text{inicio}})}{t_{\text{fin}}^3} \end{aligned} \quad (3.23)$$

En este caso, donde existen puntos intermedios y ya se ha sustituido las condiciones de contorno de las expresiones (3.3) y (3.4) por las expresiones (3.15) y (3.16) con lo que ahora resulta teniendo en cuenta la ecuación (3.17):

$$\dot{q}(t_{inicio}) = \dot{q}(0) = b \Rightarrow b = \dot{q}_{inicio} \quad (3.24)$$

$$\dot{q}(t_{fin}) = b + 2 \cdot c \cdot t_{fin} + 3 \cdot d \cdot t_{fin}^2 = \dot{q}_{fin} \quad (3.25)$$

puesto que los valores de a y b quedan determinados por las expresiones (3.18) y (3.24) respectivamente, las expresiones (3.19) y (3.25) constituyen un sistema de dos ecuaciones con dos incógnitas que una vez resuelto se obtiene:

$$\begin{aligned} a &= q_{inicio} \\ b &= \dot{q}_{inicio} \\ c &= \frac{3(q_{fin} - q_{inicio})}{t_{fin}^2} - \frac{2 \cdot \dot{q}_{inicio} + \dot{q}_{fin}}{t_{fin}} \\ d &= \frac{-2(q_{fin} - q_{inicio})}{t_{fin}^3} + \frac{\dot{q}_{fin} + \dot{q}_{inicio}}{t_{fin}^2} \end{aligned} \quad (3.26)$$

De esta forma se puede obtener la trayectoria en el espacio de nudos a seguir entre dos puntos intermedios mediante interpolación cúbica. Sin embargo, el problema realmente está en determinar los valores de contorno relativos a la velocidad (ecuaciones (3.21) y (3.22)). Hay varias formas de determinar estos valores, en este trabajo las velocidades de paso son dadas por el usuario.

Como se ha comentado anteriormente a la hora de implementar este tipo de trayectorias mediante interpolaciones cúbicas con puntos intermedios se asegura que haya continuidad tanto en posición, como en velocidad y aceleración.

Al introducir las velocidades de paso entre los distintos tramos se asegura dicha continuidad, pero a la hora de hallar los coeficientes a, b, c y d de la ecuación (3.17) se necesitará el tiempo de paso, por lo que, se calcula a partir de la posición y la velocidad.

Dado que el PUMA 560 tiene 6 grados de libertad, se puede plantear su movimiento entre dos configuraciones determinando, en el espacio de nudos, el paso por

$m-1$ configuraciones intermedias, en $m-1$ instantes de tiempo (este tiempo se ha calculado a partir de la velocidad de paso). Esto se puede expresar de la siguiente manera.

$$t = t_j \Rightarrow q_1(t_j), q_2(t_j), \dots, q_n(t_j) \quad n = 6 \quad j = 1, 2, \dots, m \quad (3.27)$$

Siendo q_i las coordenadas generalizadas del robot, quedando determinada la configuración por el conjunto de n coordenadas generalizadas. Por lo tanto, existen m tramos de movimiento y $m + 1$ configuraciones de paso.

Se hace necesario determinar las funciones temporales, una para cada grado de libertad, que permiten el paso del robot por las diferentes configuraciones. Las funciones estarán definidas a tramos, comprendidos entre dos configuraciones consecutivas, y en cada tramo la función será un polinomio de grado 3. Con esto se consigue que las funciones desplazamiento de cada grado de libertad tengan derivadas primera y segunda continuas, correspondientes a las velocidades y aceleraciones de nudo.

Para un grado de libertad i , la función desplazamiento del nudo tendrá la siguiente forma.

$$\forall t \in [t_{j-1}, t_j] \Rightarrow q_{i,j}(t) = a_{i,j} + b_{i,j} \cdot t + c_{i,j} \cdot t^2 + d_{i,j} \cdot t^3 \quad (3.28)$$

Dado que cada polinomio tiene 4 coeficientes, hay n (6) grados de libertad y m tramos entre dos configuraciones consecutivas, existe un total de $4 \cdot m \cdot n$ coeficientes de polinomios a obtener.

Las condiciones que deben cumplir los polinomios y sus derivadas primera y segunda son las siguientes.

$$\begin{aligned}
 q_{i,j}(t_{j-1}) &= a_{i,j} + b_{i,j} \cdot t_{j-1} + c_{i,j} \cdot t_{j-1}^2 + d_{i,j} \cdot t_{j-1}^3 \\
 q_{i,j}(t_j) &= a_{i,j} + b_{i,j} \cdot t_j + c_{i,j} \cdot t_j^2 + d_{i,j} \cdot t_j^3 \\
 \dot{q}_{i,1}(t_0) &= b_{i,1} + 2 \cdot c_{i,1} \cdot t_0 + 3 \cdot d_{i,1} \cdot t_0^2 = 0 \\
 \dot{q}_{i,m}(t_m) &= b_{i,m} + 2 \cdot c_{i,m} \cdot t_m + 3 \cdot d_{i,m} \cdot t_m^2 = 0
 \end{aligned} \tag{3.29}$$

El total de ecuaciones será $4 \cdot m \cdot n$, con lo que será un sistema resoluble.

En este caso se han obtenido los coeficientes a, b, c y d utilizando la aplicación Macsyma dando como resultado las siguientes soluciones:

$$\begin{aligned}
 a &= -\frac{(t_0(-\dot{q}_0 t_f^3 - 3q_0 t_f^2) + q_0 t_f^3 + t_0^2 * ((\dot{q}_0 - \dot{q}_f) t_f^2 + 3q_f t_f) + t_0^3 * (\dot{q}_f t_f - q_f))}{-t_f^3 + 3t_0 t_f^2 - 3t_0^2 t_f + t_0^3} \\
 b &= \frac{(-\dot{q}_0 t_f^3 + t_0 * ((-2\dot{q}_f - \dot{q}_0) t_f^2 + 6q_f t_f - 6q_0 t_f) + t_0^2 (\dot{q}_f + 2\dot{q}_0) t_f + t_0^3 \dot{q}_f)}{-t_f^3 + 3t_0 t_f^2 - 3t_0^2 t_f + t_0^3} \\
 c &= \frac{((\dot{q}_f - 2\dot{q}_0) t_f^2 + t_0 * ((\dot{q}_0 - \dot{q}_f) t_f + 3q_f - 3q_0) + 3q_f t_f - 3q_0 * t_f + t_0^2 (2\dot{q}_f + \dot{q}_0))}{-t_f^3 + 3t_0 t_f^2 - 3t_0^2 t_f + t_0^3} \\
 d &= \frac{(\dot{q}_f - \dot{q}_0) t_f + 2q_f + t_0 (\dot{q}_f + \dot{q}_0) - 2q_0}{-t_f^3 + 3t_0 t_f^2 - 3t_0^2 t_f + t_0^3}
 \end{aligned}$$

El cálculo de estos coeficientes se realiza fuera del bucle de control, una vez conocidas las configuraciones intermedias, el tiempo (obtenido a partir de la velocidad de paso), el número de tramos y la velocidad. Una vez obtenida la matriz de coeficientes se puede general la curva en tiempo real dentro del bucle de control.

En la siguiente tabla se muestran tanto la posición inicial como la final así como los puntos intermedios interpolados (medidos en grados).

Tabla 3-1 Ejemplo de interpolación cúbica con puntos intermedios

Posición Inicial	Posición 1	Posición 2	Posición 3	Posición Final
21.0580	-3.6770	1.7384	-10.3058	21.0580

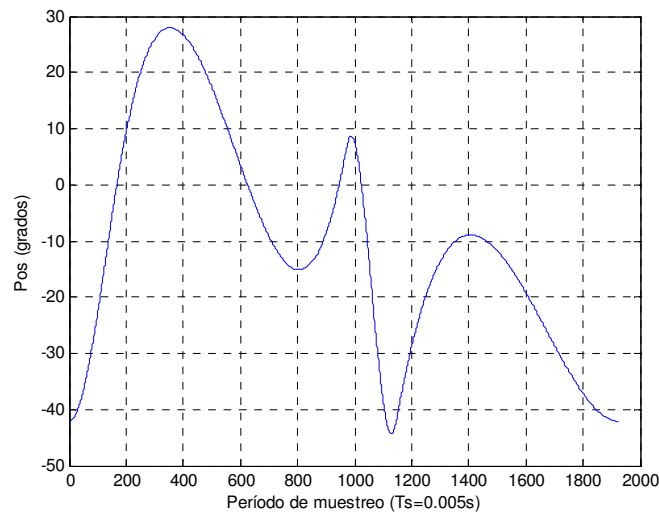


Figura 3-3 Curva con Interpolación cúbica con puntos intermedios

3.4. IMPLEMENTACIÓN DE LAS SERIES DE FOURIER

En esta tesis se ha realizado tanto un control preciso del sistema robotizado como una simulación realista del mismo, por tanto, se requiere un modelo dinámico del robot preciso. Si se quiere diseñar controladores avanzados, por ejemplo el control por par calculado, nos tendremos que basar en el modelo del robot, y su rendimiento depende directamente de la precisión del modelo (Spong y Vidyasagar, 1989; Freund, 1982; Torfs

y De Schutter, 1992; Markiewicz, 1973). Lo mismo ocurre con las simulaciones, sin un buen modelo dinámico, las ejecuciones no son realistas.

Como se ha comentado en el capítulo anterior la identificación experimental es la única forma eficiente para obtener modelos de robots precisos tanto en sus especificaciones como en confianza y validación. Ya que los datos que proporcionan los fabricantes son insuficientes, imprecisos o a menudo inexistentes. Y además, las medidas directas de los parámetros físicos no son realistas debido a la complejidad de muchos robots.

Hay que reconocer que una identificación segura, precisa y eficiente necesita experimentos especialmente diseñados. Cuando se está diseñando un experimento para la identificación de un robot manipulador, es esencial considerar si la excitación es suficiente para proporcionar precisión y rapidez a la estimación de parámetros con la presencia de perturbaciones. La influencia de las perturbaciones, tales como el ruido de las medidas y las perturbaciones de los actuadores, sobre los parámetros estimados depende directamente del conjunto de ecuaciones que genera los parámetros (Golub y Van Loan, 1989; Lawson y Hanson, 1974). La condición de este conjunto de ecuaciones depende de la excitación durante el experimento de identificación.

La generación de trayectorias de robot que optimiza la condición de la estimación de parámetros incluye optimizaciones no lineales con restricciones de movimiento. La principal diferencia entre las aproximaciones que se han realizado radica en la parametrización de la trayectoria de excitación. Los parámetros que describen la trayectoria de excitación son los grados de libertad del problema de optimización. (Armstrong, 1989) describe una aproximación en la cual los grados de libertad son los elementos de una secuencia de aceleraciones de nudo. Este es el más general, pero requiere un gran número de grados de libertad, por tanto, la optimización es voluminosa. Además, las restricciones de movimiento son difíciles de satisfacer. (Gautier, 1993) optimiza una combinación lineal del número de condiciones y del supuesto equilibrio del conjunto de ecuaciones que generan los parámetros. Los grados de libertad son un

conjunto finito de los ángulos de las articulaciones y las velocidades separadas en el tiempo. La trayectoria actual es continua y suave, y se calcula interpolando una línea entre los puntos optimizados, asumiendo que las aceleraciones iniciales y finales son cero y usando un polinomio de quinto orden. Se pueden satisfacer las restricciones de movimiento. (Otani y Kakizaki, 1993) usan trayectorias que son una combinación de un coseno y una rampa, tal que las velocidades de nudo cambian senoidalmente entre cero y su valor máximo. La excitación es optimizada a través de la selección cuidadosa de la frecuencia y la amplitud de los movimientos senoidales para cada articulación.

En nuestro caso para identificar el modelo dinámico del PUMA 560, y más concretamente sus parámetros inerciales, se utiliza otra parametrización de las trayectorias de excitación. Combina las ventajas del método descrito en (Gautier, Janin y Presse, 1993) con nuevas e interesantes características. La trayectoria de excitación para cada articulación es una suma finita de funciones seno y coseno armónicas, esto es, una serie de Fourier finita. La amplitud de las funciones de seno y coseno son los grados de libertad que se han de optimizar. Esta aproximación garantiza trayectorias periódicas que es ventajoso porque permite dominar datos promediados en el tiempo, que mejora la señal respecto a la relación con el ruido de los datos experimentales y permite la estimación de las características del ruido de medida. Esta información es valiosa en el caso de máxima probabilidad de la estimación de los parámetros. El uso de la serie finita de Fourier permite:

- Especificar el ancho de banda de las trayectorias de excitación, tal que la excitación de la flexibilidad del robot puede ser o bien evitada completamente o bien resaltada de forma intencionada.
- Calcular las velocidades y aceleraciones de nudo de un modo analítico a partir de la respuesta medida. Para este objetivo, las lecturas de los encoders son al principio aproximadas, en el sentido de mínimos cuadrados, como una suma finita de funciones seno y coseno.

Ninguno de los métodos anteriormente comentados poseen estas características.

Si se extiende la longitud de la series de Fourier al infinito, nos permite hacer una aproximación de cualquier función periódica arbitraria, mientras que las funciones no periódicas pueden ser aproximadas si las funciones seno y coseno no armónicas están permitidas en las series de Fourier. En este sentido, el método utilizado aproxima la propuesta de Armstrong (Armstrong, 1989), pero con la ventaja de que se pueden encontrar con las restricciones de movimiento. Además, las simulaciones muestran que las series de Fourier con solo unos pocas funciones seno y coseno armónicas para cada articulación son suficientes para reducir el número de condiciones del conjunto de ecuaciones que generan los parámetros para valores por debajo de 10, por tanto, no es necesario más grados de libertad en el espacio de parámetros y permite trayectorias arbitrarias.

3.4.1. Parametrización de la trayectoria

En el presente trabajo, se implementó un tipo de función analítica a fin de llevar a cabo la parametrización de la trayectoria en el espacio de nudos: una serie finita de Fourier con m armónicos (Swevers et al. 1997). La función cubre de forma continua la totalidad de la trayectoria considerada.

De acuerdo a la parametrización mediante series de Fourier, las expresiones correspondientes de las variables generalizadas a saber posición, velocidad y aceleración angular en los nudos, quedarían del siguiente modo:

$$q_i(t) = q_{i0} + \sum_{j=1}^m \left[\frac{a_{ij}}{2\pi f_f \cdot j} \sin(2\pi \cdot j \cdot t) - \frac{b_{ij}}{2\pi f_f \cdot j} \cos(2\pi \cdot j \cdot t) \right] \quad (3.30)$$

$$\dot{q}_i(t) = \sum_{j=1}^m \left[\frac{a_{ij}}{f_f} \cos(2\pi \cdot j \cdot t) + \frac{b_{ij}}{f_f} \sin(2\pi \cdot j \cdot t) \right] \quad (3.31)$$

$$\ddot{q}_i(t) = \sum_{j=1}^m \left[-\frac{2\pi \cdot j}{f_f} a_{ij} \cdot \sin(2\pi \cdot j \cdot t) + \frac{2\pi \cdot j}{f_f} b_{ij} \cdot \cos(2\pi \cdot j \cdot t) \right] \quad (3.32)$$

con f_f la frecuencia fundamental de las series de Fourier. Esta serie de Fourier especifica una función periódica con período $T_f = 1 / f_f$. La frecuencia fundamental es común a todas las articulaciones, para preservar la periodicidad de la totalidad de la excitación del robot. Cada serie de Fourier contiene $2xN_i + 1$ parámetros, que constituyen los grados de libertad para el problema de optimización: a_l^i y b_l^i , para $l = 1$ a N_i , que son las amplitudes de las funciones seno y coseno, y q_{i0} que es el offset sobre la trayectoria de posición. Este offset determina la configuración sobre la cual ocurrirá la excitación del robot.

En nuestro caso los coeficientes se pasaron mediante fichero, pudiendo elegir entre distintas series, y lo que se hizo fue generar la trayectoria para la posición, como mínimo esta trayectoria se repite una única vez, pero como ya se ha comentado al ser periódicas lo que se hace es elegir el número de repeticiones.

Tabla 3-2 Implementación para cada una de las articulaciones

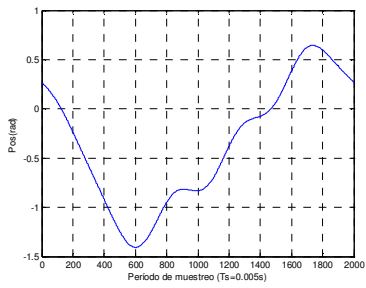


Figura 3-4 Articulación 1

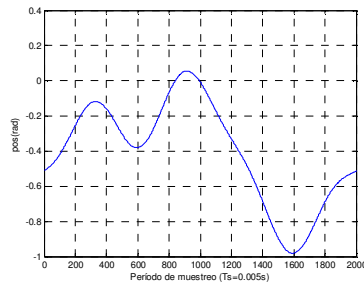


Figura 3-5 Articulación 2

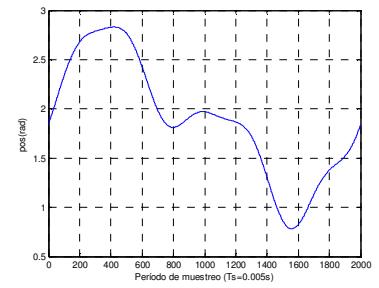


Figura 3-6 Articulación 3

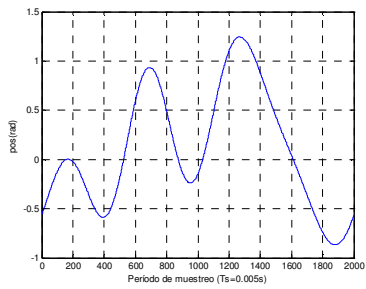


Figura 3-7 Articulación 4

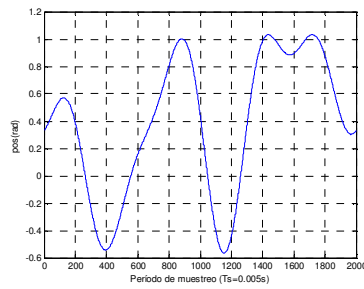


Figura 3-8 Articulación 5

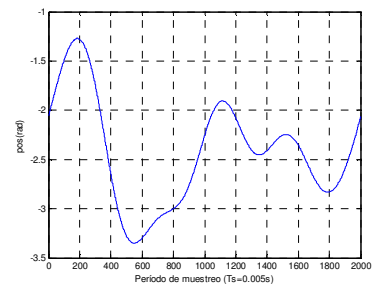


Figura 3-9 Articulación 6

CAPÍTULO 4

CONTROL DINÁMICO DE ROBOTS

MANIPULADORES

4.1. INTRODUCCIÓN

En estos últimos años, la disponibilidad de potentes microprocesadores a bajo coste ha supuesto grandes avances en la teoría y aplicación del control no lineal de robots manipuladores, lo que proporciona sistemas más fiables, precisos y con mayores prestaciones. En cuanto a la teoría podemos destacar los desarrollos realizados en las áreas de la linealización por realimentación, control pasivo y técnicas de adaptación no lineal. En cuanto a las aplicaciones, se han desarrollado algunos sistemas prácticos de control no lineal como sistemas de control de vuelo fly by wire para aviones, robots avanzados o sistemas espaciales. Como resultado, el tema de control de robots está ocupando un lugar cada vez más importante en la ingeniería de control y es una parte fundamental de la formación de los ingenieros de control.

El control no lineal debe abordar dos aspectos: el análisis y diseño de sistemas de control que contengan al menos una componente no lineal. Así, en el análisis se determinarán las características del comportamiento del sistema, mientras que en el diseño se tendrá que construir un controlador que permita que la planta cumpla las características o especificaciones deseadas de comportamiento en bucle cerrado. De esta forma recientemente se ha mostrado un gran interés por el desarrollo y aplicaciones de la metodología del control no lineal, ya que:

- **Mejora los sistemas de control existentes:** los métodos de control lineales son supuestamente la clave de los modelos lineales de pequeño rango de operación. Cuando se requiere un rango de operación elevado, un control lineal es pobre y puede ser inestable ya que las no linealidades que aparezcan en el sistema no pueden ser compensadas. Los controladores no lineales, por otra parte, pueden manipular directamente las no linealidades dentro de un rango de operación grande. Esto se demuestra en el problema del movimiento de un robot. Cuando se usa un control lineal para controlar el movimiento de un robot se descuidan las fuerzas no lineales asociadas al movimiento de las articulaciones del robot. Así, la precisión del controlador se degrada más rápido que se incrementa la velocidad del movimiento ya que algunas de las fuerzas involucradas (como pueden ser la de Coriolis y la centrípeta) varían con el cuadrado de la velocidad.
- **Análisis de no linealidades duras:** a veces tenemos sistemas cuya naturaleza discontinua no nos permite una aproximación lineal. Estas no linealidades se denominan no linealidades duras (como es el caso de la saturación, zonas muertas, juego e histéresis etc., que por desgracia se encuentran a menudo en la ingeniería de control...), y generalmente causan efectos no deseados en el control del sistema tales como inestabilidades, ciclos límite, etc.
- **Tratamiento de modelos desconocidos:** en el diseño de controladores lineales, a veces, es necesario asumir que los parámetros del modelo del sistema son razonablemente bien conocidos. Sin embargo, algunos problemas de control

involucran incertidumbres en los parámetros del modelo (cambios bruscos de los parámetros etc.). De esta forma un control lineal inexacto (o valores obsoletos de los parámetros del modelo) puede presentar degradación en la precisión e incluso inestabilidad.

- **Simplicidad de diseño:** Un buen diseño de control no lineal puede ser en ocasiones más simple y más intuitivo que uno lineal. Esto que a priori parece paradójico se deduce del hecho de que los controladores no lineales están a menudo profundamente relacionados con la física de la planta a controlar.

En el pasado, la aplicación de métodos de control no lineal estaba limitada por la dificultad computacional asociada al análisis y diseño de este tipo de control. En los últimos años, sin embargo, los avances en tecnología de computadores han resuelto este problema existiendo últimamente un interés considerable por la investigación y aplicación de métodos de control no lineal. Si bien es posible encontrar diversas formas de abordar el desarrollo de control dinámico no lineal de robots, esta tesis se ha centrado en las áreas siguientes:

- **Control no lineal mediante la teoría de Lyapunov**

Generalmente, para un sistema de control la componente más importante a determinar es su estabilidad. Si el sistema es lineal e invariante en el tiempo, se dispone de muchos criterios de estabilidad (criterios de estabilidad de Nyquist, Routh, etc.). Sin embargo, si el sistema es no lineal o lineal pero variable en el tiempo estos criterios no se pueden aplicar. Un método más general para la determinación de estabilidad de sistemas no lineales y/o variables en el tiempo es el segundo método de Lyapunov (también denominado método directo de Lyapunov).

Se sabe de la teoría clásica de la mecánica que un sistema vibratorio es estable si su energía total (una función positiva definida) es continuamente decreciente (lo que significa que la derivada temporal de la energía total debe ser negativa

definida) hasta alcanzar un estado de equilibrio. El segundo método de Lyapunov está basado en este hecho: si el sistema tiene un estado de equilibrio asintóticamente estable, la energía acumulada del sistema desplazado dentro del dominio de atracción cae al crecer el tiempo hasta tomar finalmente su valor mínimo en el estado de equilibrio.

Sin embargo, para sistemas puramente matemáticos, no existe una manera simple de definir una función, energía. Para eliminar esta dificultad Lyapunov introdujo la denominada función, de Lyapunov, que se puede considerar como una función de energía ficticia. Esta idea es más general que la de energía y es de aplicación más amplia. De hecho, cualquier función escalar que satisfaga las hipótesis de los teoremas de estabilidad de Lyapunov puede servir como función de Lyapunov.

• **Linealización por realimentación no lineal**

En su forma más simple, la linealización por realimentación pretende cancelar las no linealidades de un sistema para que su dinámica en bucle cerrado sea lineal. La idea de la linealización por realimentación puede aplicarse a una clase de sistemas no lineales descritos mediante lo que se denomina forma canónica controlable. De esta forma se dice que un sistema tiene forma canónica controlable si su dinámica se puede expresar como:

$$\dot{x}^{(n)} = f(x) + b(x)u \quad (4.1)$$

donde:

u : entrada de control

$$x = \begin{bmatrix} x_1 & \dot{x} & \dots & x^{(n-1)} \end{bmatrix}^T : \text{vector de estado} \quad (4.2)$$

$f(x)$: función de estados no lineal

En sistemas que se pueden expresar de la forma anterior podemos cancelar las no linealidades usando la entrada de control compuesta por una acción de control lineal v (que se calcula mediante un controlador lineal), obteniéndose a partir de ella una acción de control no lineal u de entrada al proceso (que contendrá las no linealidades del sistema).

La estructura general del control por linealización se podría esquematizar de la forma siguiente:

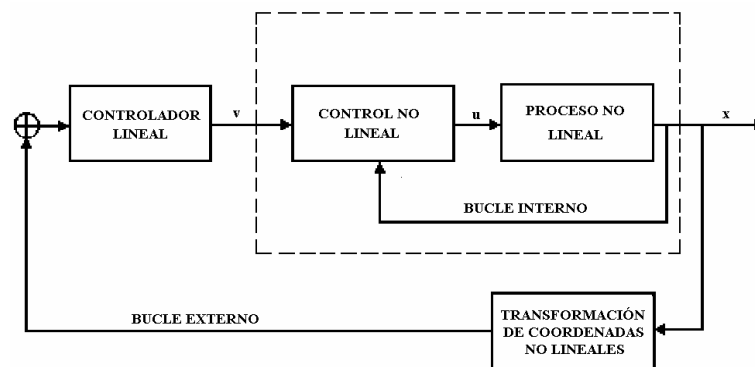


Figura 4-1 Estructura General Control por Linealización

4.2. CONTROL POR DINÁMICA INVERSA

4.2.1. Introducción

El esquema de control por dinámica inversa fue desarrollado en los primeros años 70 bajo el nombre de control de par calculado. Uno de los primeros experimentos que se

llevaron a cabo con el control por dinámica inversa fue en (Markiewicz, 73). La búsqueda de métodos para simplificar el modelo dinámico y, por tanto, el coste computacional que interviene en el método de control por dinámica inversa fue estudiado en (Aubin, Canudas de Wit y Sidaoui, 1991) entre otros. En este caso se ha realizado mediante un algoritmo recursivo basado en la formulación de Gibbs-Apell visto en el capítulo 2.

Un estudio de las propiedades del modelo dinámico utilizado para el control puede encontrarse en (Grimm, 90).

Los controladores por dinámica inversa (o linealización por cancelación) cancela las no linealidades de un sistema para que su dinámica en bucle cerrado sea lineal.

Se puede aplicar a los sistemas que se pueden expresar en forma canónica controlable:

$$x^{(n)} = f(x) + b(x)u \quad (4.3)$$

Como es sabido, estos sistemas se pueden controlar utilizando la entrada de control siguiente:

$$u = \frac{1}{b(x)}[a - f(x)] \quad (4.4)$$

por tanto, se obtiene una relación simple de entrada/salida:

$$x^{(n)} = a \quad (4.5)$$

En este caso, en el controlador vamos a generar la acción de control necesaria para llevar al sistema a la posición deseada. Para ello lo realizamos en dos partes.

La primera de ellas genera la “*aceleración de control*”, a , o control de acción lineal que se calcula en función de la posición, posición deseada, velocidad, velocidad

deseada, aceleración deseada y el tipo de seguimiento sobre la trayectoria que hayamos decidido aplicar, esto es, punto a punto, seguimiento de la trayectoria o este último pero con acción integral. Se puede decir que se calcula la *aceleración* que hay que suministrar al sistema para que éste responda correctamente.

La segunda de ellas calcula la entrada al proceso u que contendrá las no linealidades del sistema.

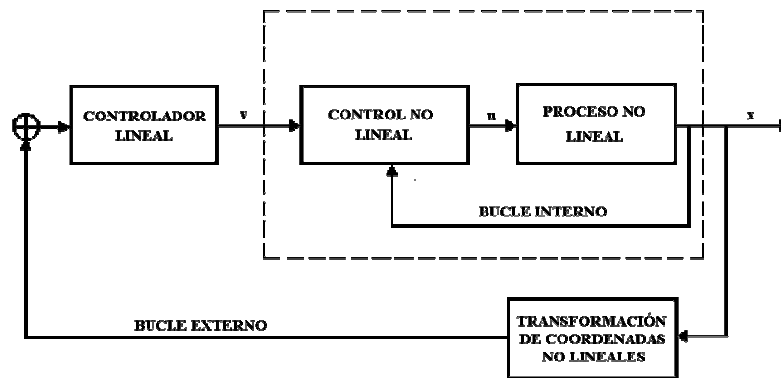


Figura 4-2 Estructura general control por dinámica inversa

La estructura general del control por dinámica inversa se podría esquematizar de la forma siguiente:

Para calcular el control por dinámica inversa partimos de la siguiente ecuación ya vista anteriormente.

$$\tau = M(q, \theta) a + C(q, \dot{q}, \theta) \dot{q} + G(q, \theta) \quad (4.6)$$

Despejando la variable de estado:

$$\ddot{q} = M^{-1}(q, \theta) (\tau - C(q, \dot{q}, \theta) \dot{q} - G(q, \theta)) \quad (4.7)$$

$$x^{(n)} = f(x) + b(x)u \begin{cases} f(x) \equiv M^{-1}(q, \theta)(-C(q, \dot{q}, \theta)\dot{q} - G(q, \theta)) \\ b(x) \equiv M^{-1}(q, \theta) \end{cases} \quad (4.8)$$

La acción de control será, por lo tanto,

$$\begin{aligned} \tau &= \frac{1}{M^{-1}(q, \theta)} [a - M^{-1}(q, \theta)(-C(q, \dot{q}, \theta) - G(q, \theta))] \\ u &= \frac{1}{b} [a - f] \end{aligned} \quad (4.9)$$

Trabajando con esta expresión:

$$\tau = M(q, \theta)a + C(q, \dot{q}, \theta)\dot{q} + G(q, \theta) \quad (4.10)$$

donde a es un nuevo vector $n \times 1$ que se describirá posteriormente.

A partir de estas ecuaciones:

$$M(q, \theta)\ddot{q} = M(q, \theta)a, \text{ siendo } M \text{ una matriz simétrica obtenemos } \ddot{q} = a$$

Pues bien, se puede establecer un control sobre la trayectoria, calculando a de la forma que se expone en la tabla siguiente:

Tabla 4-1 Control sobre la trayectoria

Tipo de control sobre la trayectoria	a
Punto a punto	$-Kp * e - Kd * v$
Seguimiento de la trayectoria	$\ddot{q}d - Kp * e - Kd * v$
Seguimiento de la trayectoria con acción integral	$\ddot{q}d - Kp * e - Kd * v - Ki * \int_0^1 e dt$

donde a_d es la aceleración deseada y el error de posición y de velocidad se definen de la siguiente forma:

$$e = q - q_d$$

$$e_v = v - v_d$$

Como ya se ha mencionado en el capítulo 2, el control propuesto está basado en las ecuaciones de la dinámica inversa de Gibbs-Appell. Dichas ecuaciones serán la encargada de calcular el par de fuerzas que recibirá el robot para generar un movimiento correcto. Estas ecuaciones toman como parámetros la posición, velocidad y la aceleración de control.

4.2.2. Control Punto a Punto

Para realizar este tipo de control sobre la trayectoria calculamos el término a de la ecuación de la dinámica de la siguiente forma:

$$-Kp * e - Kd * v \tag{4.11}$$

donde

$$e = q - q_d$$

Una vez calculada la aceleración de control se resuelve el problema dinámico inverso en base a las ecuaciones de la dinámica inversa de Gibbs-Appell. Como resultado se obtiene el par de fuerzas que recibirá el robot para generar un movimiento correcto.

Se puede resumir con la siguiente ley de Control que también viene especificada en la siguiente figura:

$$\tau = M(q, \theta)(-K_d \dot{q} - K_p e) + C(q, \dot{q}, \theta)\dot{q} + G(q, \theta)$$

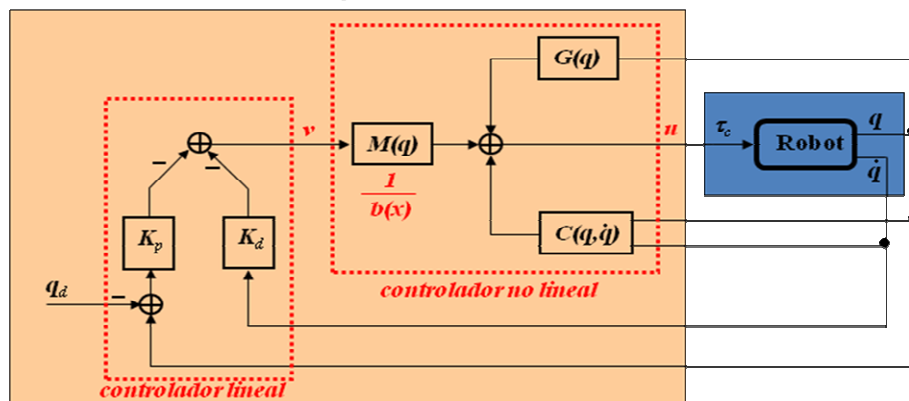


Figura 4-3 Esquema de control del controlador por dinámica inversa con control punto a punto

4.2.2.1. Resultados reales de las ejecuciones para una curva de Hermite sin puntos intermedios

Tabla 4-2 Comparación de referencia y posición para las 6 articulaciones del controlador por dinámica inversa punto a punto
(abcisas: posición (grados); ordenadas: período de muestreo ($T_s=0.005s$))

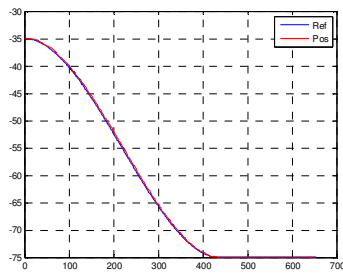


Figura 4-4 Articulación 1

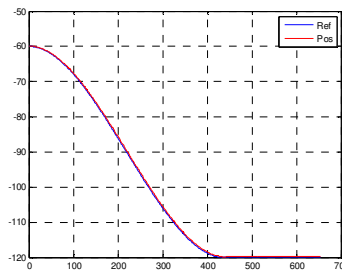


Figura 4-5 Articulación 2

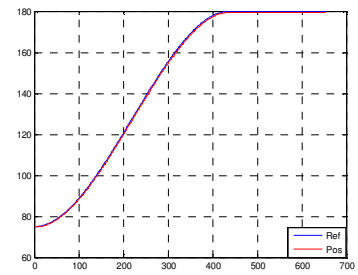


Figura 4-6 Articulación 3

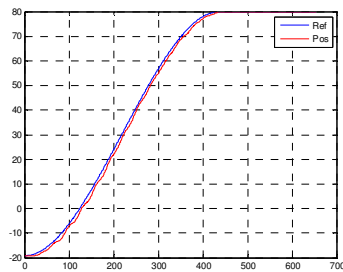


Figura 4-7 Articulación 4

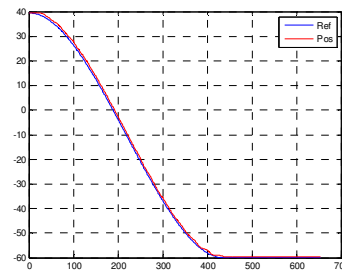


Figura 4-8 Articulación 5

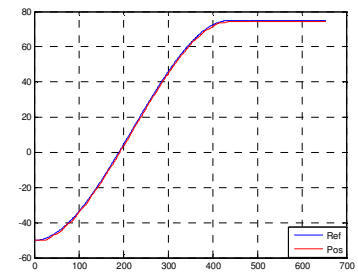


Figura 4-9 Articulación 6

Tabla 4-3 Acciones de Control del controlador por dinámica inversa punto a punto
(abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))

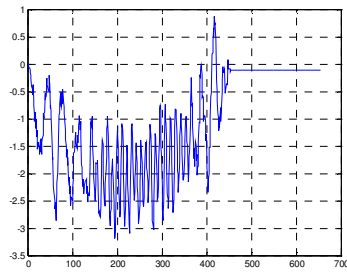


Figura 4-10 Articulación 1

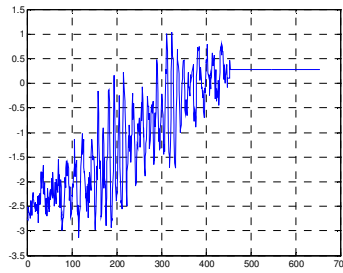


Figura 4-11 Articulación 2

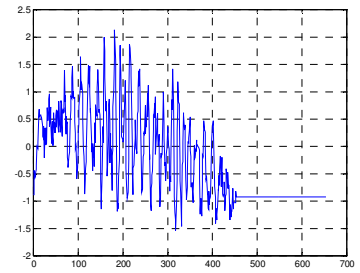


Figura 4-12 Articulación 3

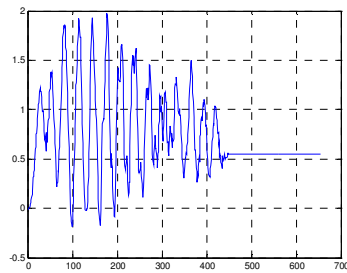


Figura 4-13 Articulación 4

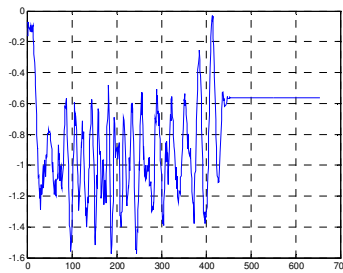


Figura 4-14 Articulación 5

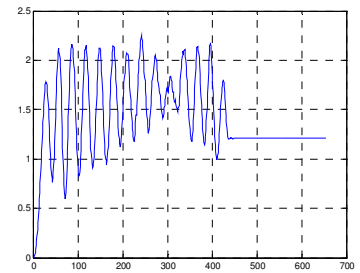


Figura 4-15 Articulación 6

4.2.2.2. Resultados reales de las ejecuciones para una trayectoria en el espacio cartesiano del Robot con curvas polinomiales de tercer orden con interpolación en puntos intermedios

Tabla 4-4 Comparación de referencia y posición para las 6 articulaciones del controlador por dinámica inversa punto a punto (abscisas: posición (grados); ordenadas: período de muestreo (Ts=0.005s))

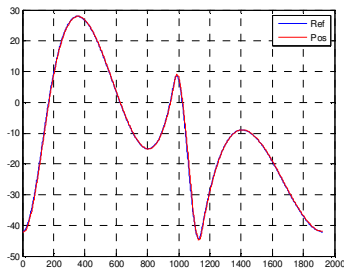


Figura 4-16 Articulación 1

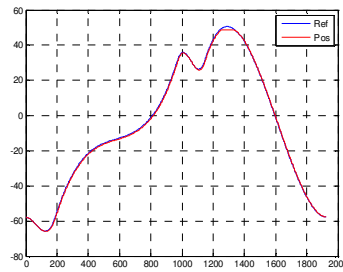


Figura 4-17 Articulación 2

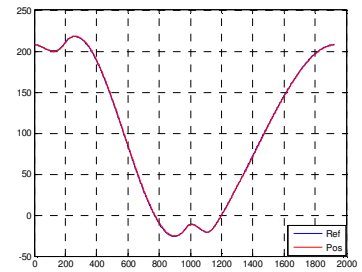


Figura 4-18 Articulación 3

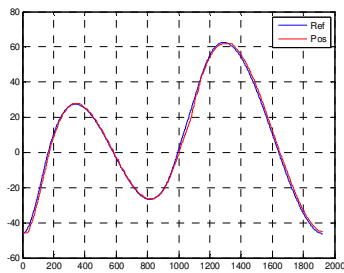


Figura 4-19 Articulación 4

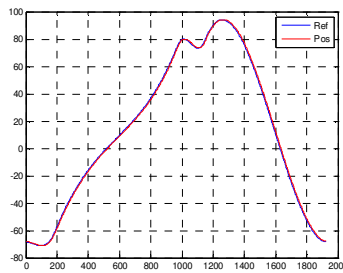


Figura 4-20 Articulación 5

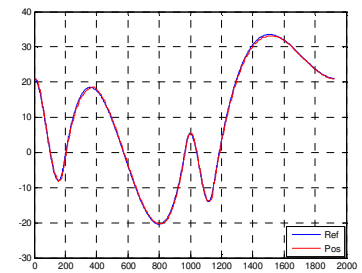


Figura 4-21 Articulación 6

Tabla 4-5 Acciones de Control del controlador por dinámica inversa punto a punto
(abcisas: acción de control (Nm); ordenadas: periodo de muestreo ($T_s=0.005s$))

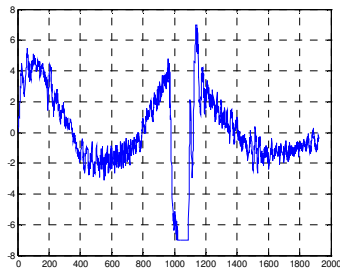


Figura 4-22 Articulación 1

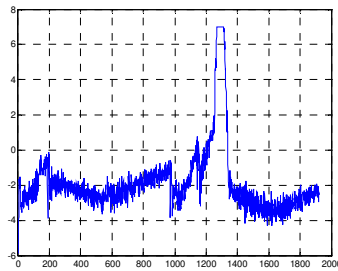


Figura 4-23 Articulación 2

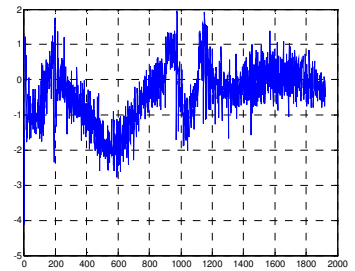


Figura 4-24 Articulación 3

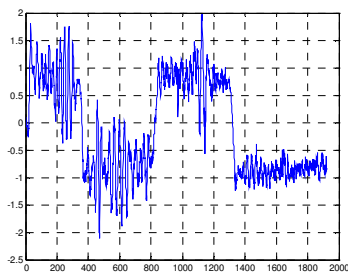


Figura 4-25 Articulación 4

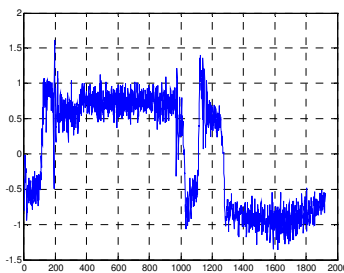


Figura 4-26 Articulación 5

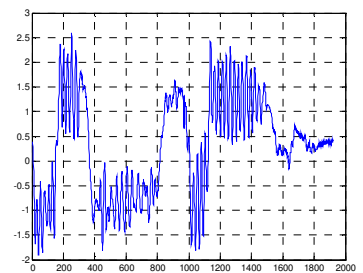


Figura 4-27 Articulación 6

4.2.2.3. Resultados reales de las ejecuciones para una trayectoria de Fourier

Tabla 4-6 Comparación de referencia y posición para las 6 articulaciones del controlador por dinámica inversa punto a punto

(abcisas: posición (rad); ordenadas: período de muestreo ($T_s=0.005s$))

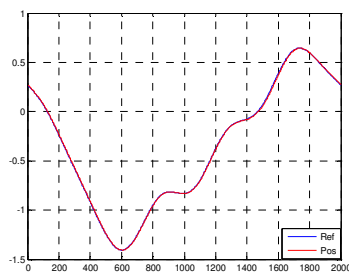


Figura 4-28 Articulación 1

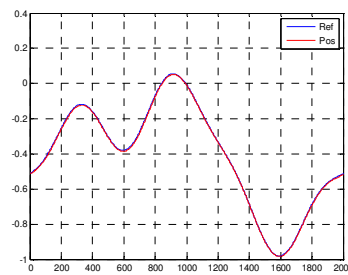


Figura 4-29 Articulación 2

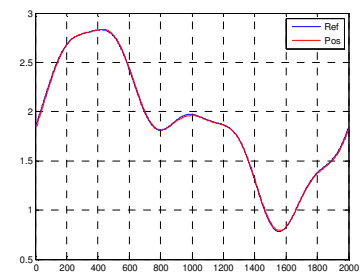


Figura 4-30 Articulación 3

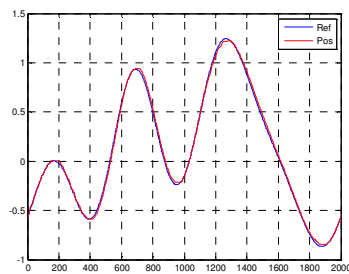


Figura 4-31 Articulación 4

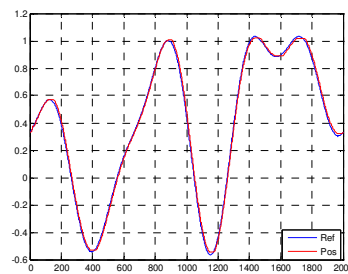


Figura 4-32 Articulación 5

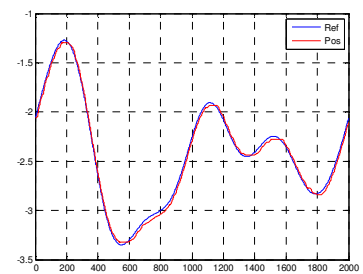


Figura 4-33 Articulación 6

Tabla 4-7 Acciones de Control del controlador por dinámica inversa punto a punto
(abcisas: acción de control (Nm); ordenadas: periodo de muestreo ($T_s=0.005s$))

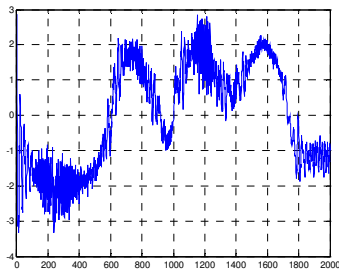


Figura 4-34 Articulación 1

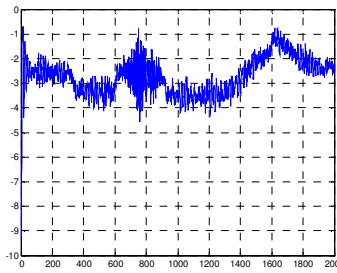


Figura 4-35 Articulación 2

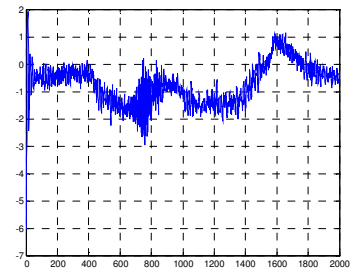


Figura 4-36 Articulación 3

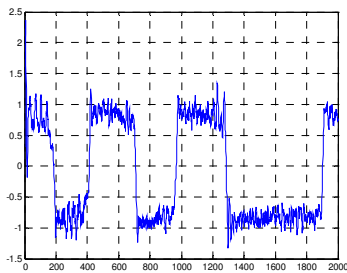


Figura 4-37 Articulación 4

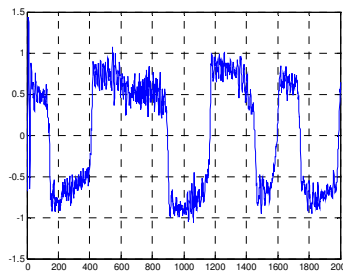


Figura 4-38 Articulación 5

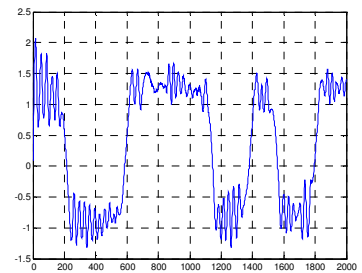


Figura 4-39 Articulación 6

4.2.3. Seguimiento de la Trayectoria

Para realizar este tipo de control sobre la trayectoria calculamos el término a de la ecuación de la dinámica de la siguiente forma:

$$\ddot{q}_d - K_p * e - K_d * \dot{e}, \tag{4.12}$$

donde

$$e = q - q_d$$

\ddot{q}_d es la aceleración deseada.

Una vez calculada la aceleración de control se resuelve el problema dinámico inverso en base a las ecuaciones de la dinámica inversa de Gibbs-Appell. Como resultado se obtiene el par de fuerzas que recibirá el robot para generar un movimiento correcto. La siguiente figura y ley de control nos muestra un esquema de este controlador:

$$\tau = M(q, \theta)(\ddot{q}_d - K_d \dot{e} - K_p e) + C(q, \dot{q}, \theta)\dot{q} + G(q, \theta)$$

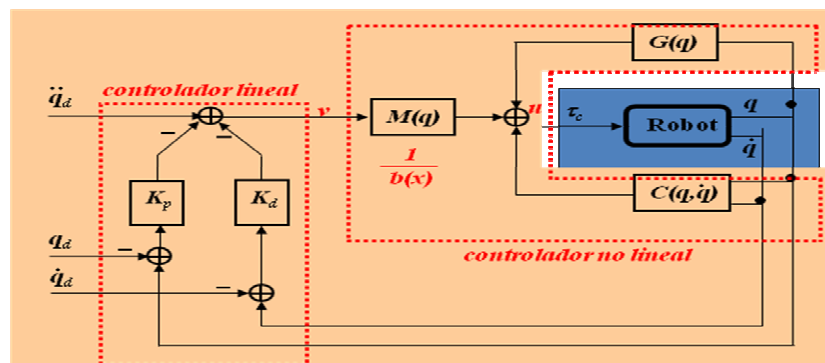


Figura 4-40 Esquema del controlador por dinámica inversa con seguimiento de la trayectoria

4.2.3.1. Resultados reales de las ejecuciones para una curva de Hermite sin puntos intermedios

Tabla 4-8 Comparación de referencia y posición para las 6 articulaciones del controlador por dinámica inversa con seguimiento de la trayectoria
(abcisas: posición (grados); ordenadas: período de muestreo ($T_s=0.005s$))

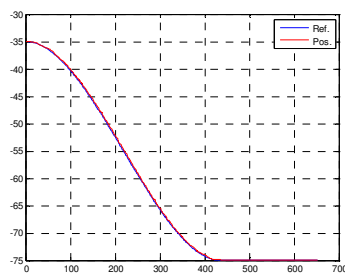


Figura 4-41 Articulación 1

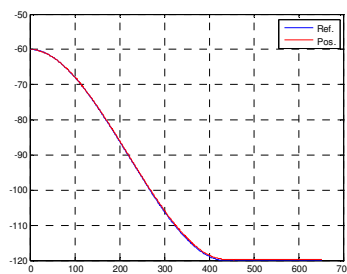


Figura 4-42 Articulación 2

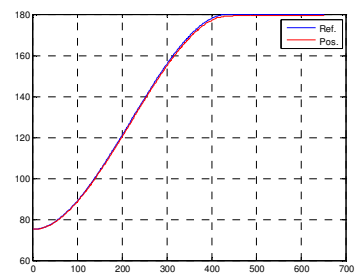


Figura 4-43 Articulación 3

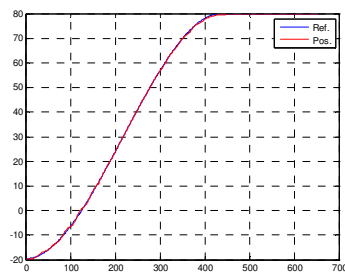


Figura 4-44 Articulación 4

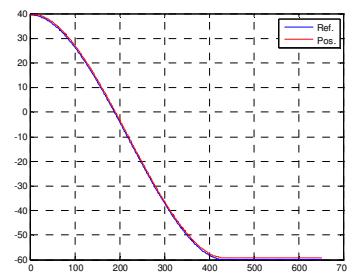


Figura 4-45 Articulación 5

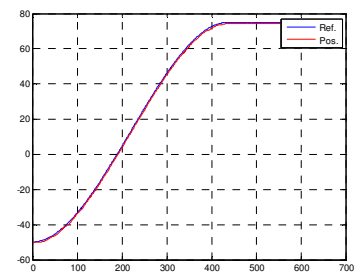


Figura 4-46 Articulación 6

Tabla 4-9 Acciones de control del controlador por dinámica inversa con seguimiento de la trayectoria
(abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))

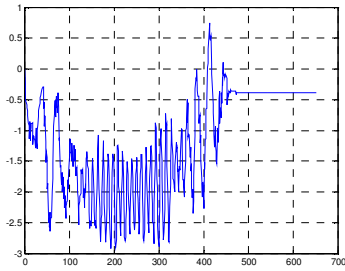


Figura 4-47 Articulación 1

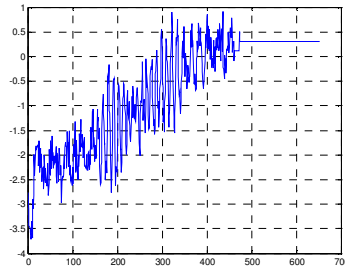


Figura 4-48 Articulación 2

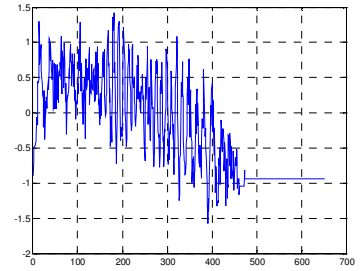


Figura 4-49 Articulación 3

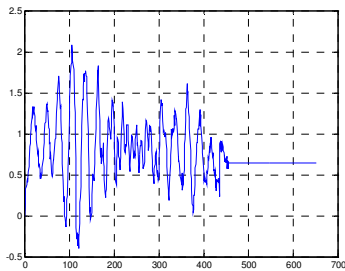


Figura 4-50 Articulación 4

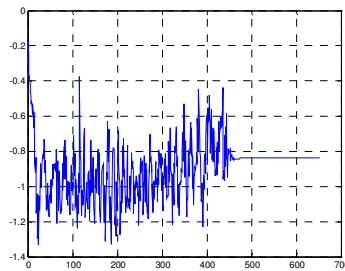


Figura 4-51 Articulación 5

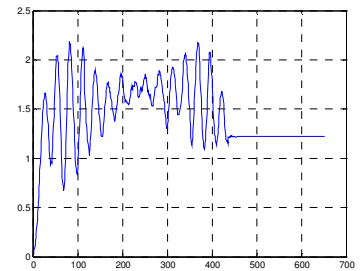


Figura 4-52 Articulación 6

4.2.3.2. Resultados reales de las ejecuciones para una trayectoria en el espacio cartesiano del Robot con curvas polinomiales de tercer orden con interpolación en puntos intermedios

Tabla 4-10 Comparación de referencia y posición para las 6 articulaciones del controlador por dinámica inversa con seguimiento de la trayectoria
(abcisas: posición (grados); ordenadas: período de muestreo ($T_s=0.005s$))

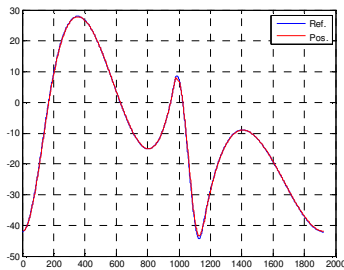


Figura 4-53 Articulación 1

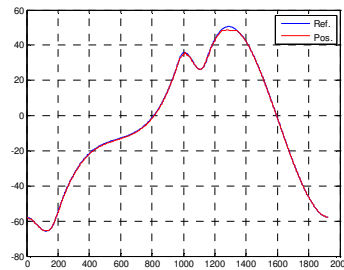


Figura 4-54 Articulación 2

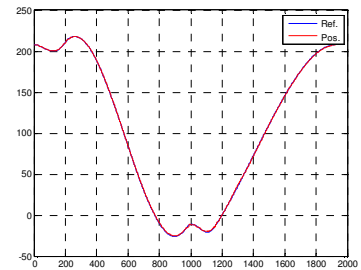


Figura 4-55 Articulación 3

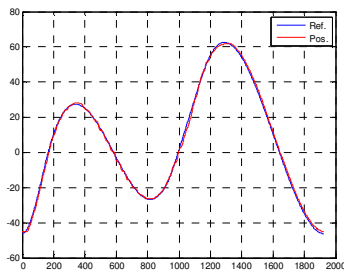


Figura 4-56 Articulación 4

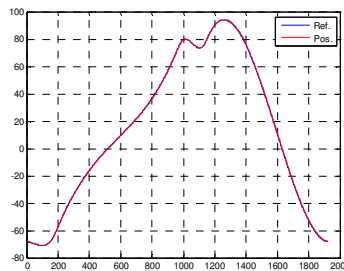


Figura 4-57 Articulación 5

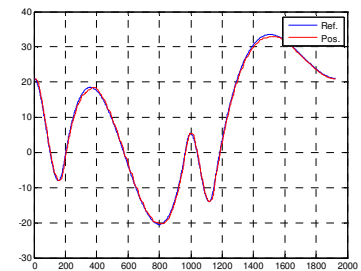


Figura 4-58 Articulación 6

Tabla 4-11 Acciones de control del controlador por dinámica inversa con seguimiento de la trayectoria
(abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))

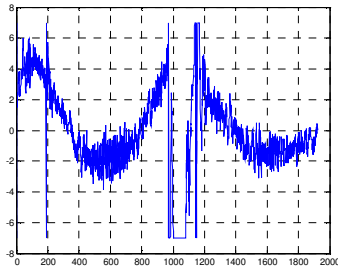


Figura 4-59 Articulación 1

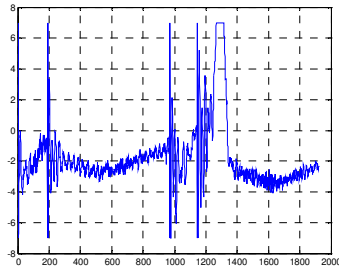


Figura 4-60 Articulación 2

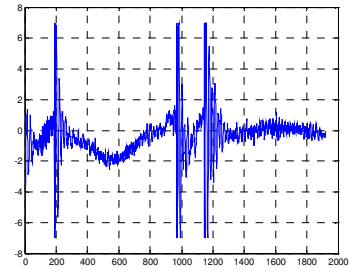


Figura 4-61 Articulación 3

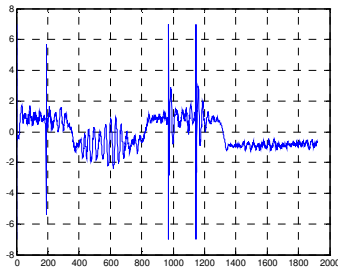


Figura 4-62 Articulación 4

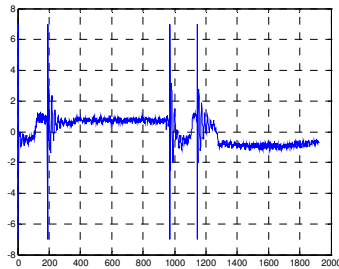


Figura 4-63 Articulación 5

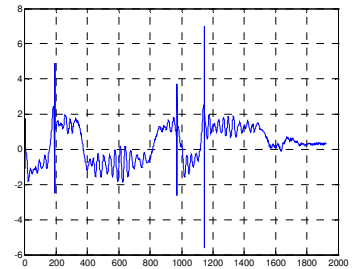


Figura 4-64 Articulación 6

4.2.3.3. Resultados reales de las ejecuciones para una trayectoria de Fourier

Tabla 4-12 Comparación de referencia y posición para las 6 articulaciones del controlador por dinámica inversa con seguimiento de la trayectoria (abcisas: posición (rad); ordenadas: período de muestreo (Ts=0.005s))

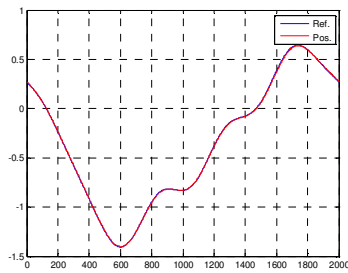


Figura 4-65 Articulación 1

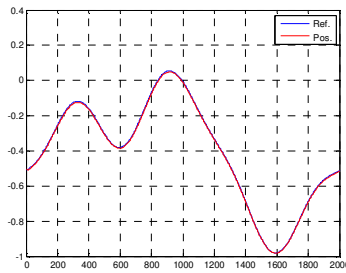


Figura 4-66 Articulación 2

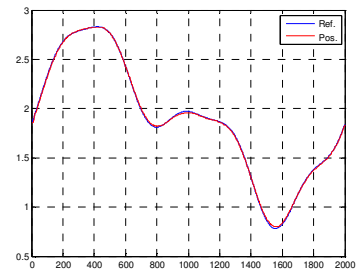


Figura 4-67 Articulación 3

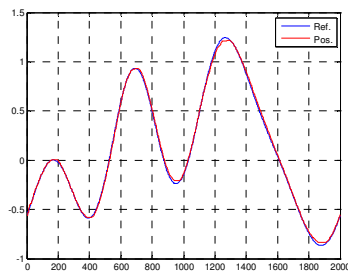


Figura 4-68 Articulación 4

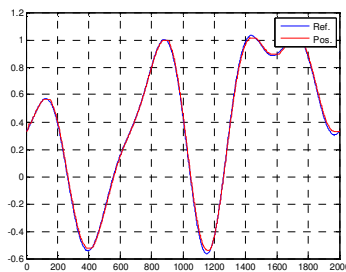


Figura 4-69 Articulación 5

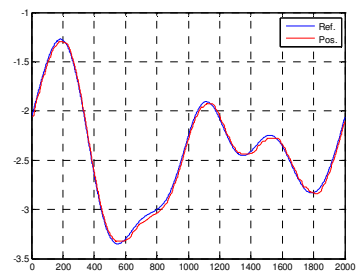


Figura 4-70 Articulación 6

Tabla 4-13 Acciones de control del controlador por dinámica inversa con seguimiento de la trayectoria
(abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))

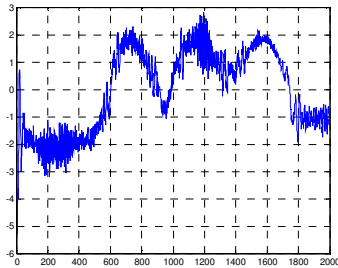


Figura 4-71 Articulación 1

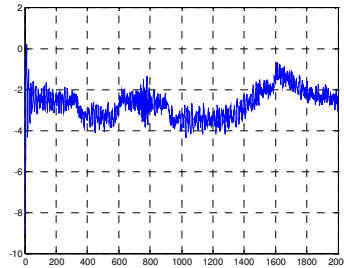


Figura 4-72 Articulación 2

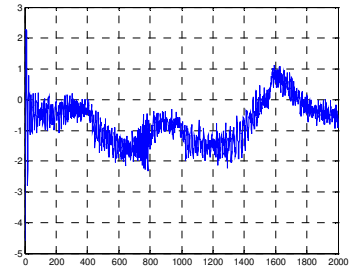


Figura 4-73 Articulación 3

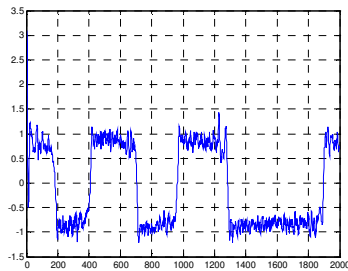


Figura 4-74 Articulación 4

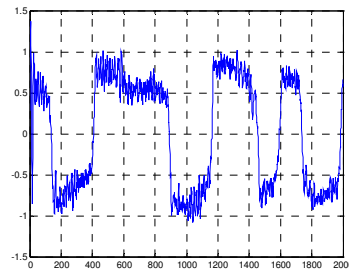


Figura 4-75 Articulación 5

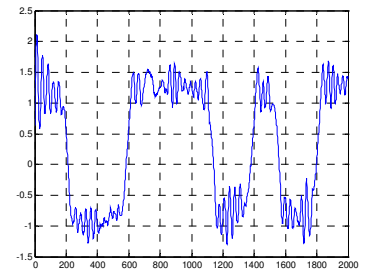


Figura 4-76 Articulación 6

4.2.4. Seguimiento de la Trayectoria con Acción Integral

Para realizar este tipo de control sobre la trayectoria calculamos el término a de la ecuación de la dinámica de la siguiente forma:

$$\ddot{q}_d - K_p * e - K_d * v - K_i * \int_0^1 e \cdot dt \quad (4.13)$$

donde

$$e = q - q_d$$

\ddot{q}_d es la aceleración deseada.

Una vez calculada la aceleración de control se resuelve el problema dinámico inverso en base a las ecuaciones de la dinámica inversa de Gibbs-Appell. Como resultado se obtiene el par de fuerzas que recibirá el robot para generar un movimiento correcto.

A continuación se muestra la ley de Control asociada así como un esquema del controlador:

$$\tau = M(q, \theta) \left(\ddot{q}_d - K_d \dot{e} - K_p e - K_i \int_0^t e(u) du \right) + C(q, \dot{q}, \theta) \dot{q} + G(q, \theta)$$

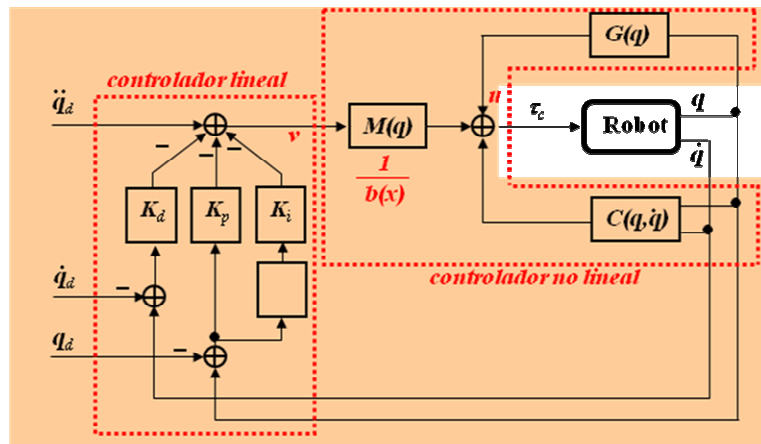


Figura 4-77 Esquema del controlador por dinámica inversa con seguimiento de la trayectoria con acción integral

4.2.4.1. Resultados reales de las ejecuciones para una trayectoria en el espacio cartesiano del Robot con curvas polinomiales de tercer orden con interpolación en puntos intermedios

Tabla 4-14 Comparación de referencia y posición para las 6 articulaciones del controlador por dinámica inversa con seguimiento de la trayectoria con acción intergral (abcisas: posición (grados); ordenadas: período de muestreo ($T_s=0.005s$))

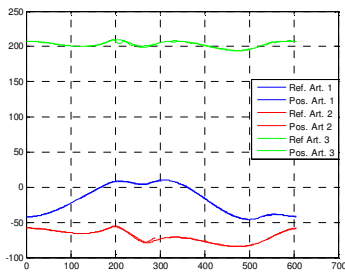


Figura 4-78 Articulaciones 1, 2 y 3

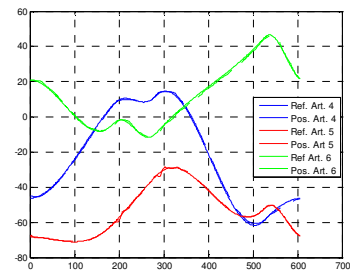


Figura 4-79 Articulaciones 4, 5 y 6

Tabla 4-15 Acciones de Control del controlador por dinámica inversa con seguimiento de la trayectoria con acción intergral (abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))

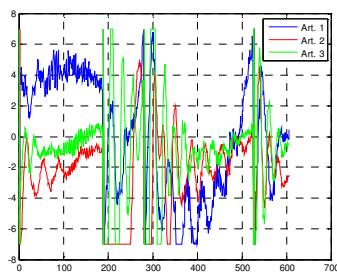


Figura 4-80 Articulaciones 1, 2 y 3

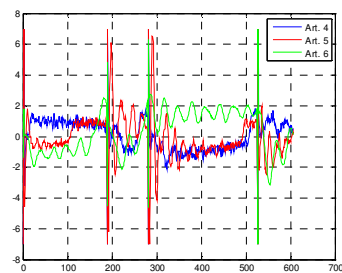


Figura 4-81 Articulaciones 4, 5 y 6

Cuando la acción proporcional (junto con la derivativa) ya no da más de sí es cuando tiene que hacer efecto la acción integral. Si ésta actúa antes de hora (mayor K_i) el sistema sobrepasará la posición deseada al suministrar más acción de control, aunque también será la acción integral la que haga que se vuelva hasta la posición deseada, convirtiendo al sistema en un sistema subamortiguado. En cambio, si actúa más tarde (menor K_i) la acción integral comenzará más tarde a ayudar a la acción proporcional, haciendo que el sistema tarde más en llegar a la posición final.

A la hora de realizar las pruebas en todos los controladores se ha observado el efecto de la zona muerta y, en el caso de la necesidad de utilizar del modelo dinámico del robot la falta de precisión en este modelo puesto que no se ha desarrollado con los parámetros inerciales identificados, en Capítulo 6 se estudia la obtención de la ecuación del movimiento y el control por dinámica inversa con los parámetros inerciales ya identificados. Estas últimas nos ha ayudado a ver la función de la acción integral. Cuando la acción proporcional ya no surte efecto, es cuando actúa la acción integral, ésta es la que suministra la acción de control necesaria para el movimiento, pero si esta junto con la proporcional no supera la zona muerta, ni el rozamiento de los motores, la fricción ,etc del modelo dinámico, no surtirá efecto, pero cuando la integral del controlador se carga suficientemente se consigue superar estos contratiempos y , por consiguiente, el esperado movimiento, pero cuando se mueve se descarga la integral con lo que se repite el ciclo. La acción integral es la que hace que asintóticamente el sistema llegue a la posición final.

4.3. CONTROL POR PASIVIDAD

4.3.1. Introducción

En los últimos años ha cobrado una gran importancia la metodología de controladores de sistemas robot basados en la pasividad. Estos controladores afrontan el problema del control de robots explotando la estructura física del sistema, especialmente la propiedad de la pasividad (de ahí su nombre). La idea de esta filosofía de diseño de controladores es modificar o modelar la energía natural del sistema robot de forma que se alcance el objetivo de control.

Así pues en este apartado se van a analizar distintas estrategias pasivas que van a resolver un tipo de problema de control: el control punto a punto. Estos controladores del primer tipo (o reguladores) se encargan únicamente de fijar la posición final de llegada para los distintos elementos del sistema robotizado. Además del problema de la regulación existe otro más general y complejo en el que se debe especificar dónde se quiere llegar (posición final) y la forma con que se quiere llegar, resolviendo en este caso el problema del servosistema. De esta forma, en este problema se deberán fijar las referencias necesarias (de posición, velocidad y aceleración) para obtener el seguimiento de las trayectorias.

4.3.2. Controladores punto a punto por pasividad

Los reguladores basados en la pasividad con estrategias punto a punto se pueden ver como casos particulares de la siguiente ley de control general:

$$\tau_e = -K_p e - Kd_v - g \quad (4.14)$$

Se puede observar cómo la ley de control está compuesta por tres términos: el error de posición (con $e = q - q_d$), el término de velocidad (o una estimación de ella, según la estrategia de control) que aportará amortiguación al sistema, y por último, el término de compensación de la gravedad.

Así, los elementos v y g varían según la estrategia de control de acuerdo a la tabla siguiente:

Tabla 4-16 Controladores Pasivos Punto a Punto para Robots Rígidos

Regulador	g	v
PD + G(q)	$-G(q, \theta)$	\dot{q}
PD + G(q _d)	$-G(q_d, \theta)$	\dot{q}
Compensador lineal + G(q)	$-G(q, \theta)$	$\dot{x} = -Lx + K_d \dot{q}$
Compensador lineal + G(q _d)	$-G(q_d, \theta)$	$\dot{x} = -Lx + K_d \dot{q}$
Diferenciación aproximativa + G(q)	$-G(q, \theta)$	$diag \left\{ \frac{b_i p}{p + a_i} \right\}$
Diferenciación aproximativa + G(q _d)	$-G(q_d, \theta)$	$diag \left\{ \frac{b_i p}{p + a_i} \right\}$
PID	$K_i \int_0^t e dt$	\dot{q}
PI ² D	$K_i \int_0^t (e + diag \left\{ \frac{b_i p}{p + a_i} \right\} q) dt$	$diag \left\{ \frac{b_i p}{p + a_i} \right\}$

El primero de los controladores se trata del PD con compensación del término de la gravedad. El controlador fue propuesto por (Takegaki, Arimoto, 1981) y ha servido de

base para el desarrollo de otros muchos controladores. Las propiedades de la pasividad unidas al modelo del manipulador también fueron estudiadas por (Takegaki, Arimoto, 1981). Estas propiedades se usaron en (Arimoto, Miyazaki, 1983), para demostrar la estabilidad del control PD. La estrategia de control tiene una estructura muy sencilla, puesto que tiene por un lado una realimentación lineal del estado así como la compensación de las fuerzas gravitatorias. La parte de compensación de la gravedad junto con la parte de realimentación proporcional modifican la energía potencial del sistema robot en la forma deseada, mientras que el término derivativo estabiliza asintóticamente el sistema.

La segunda estrategia también fue propuesta por los mismos autores, tratándose de una variación del anterior. En este caso se sustituye el término de compensación de las fuerzas de gravedad $G(q, \theta)$ por la compensación de dichas fuerzas en la posición deseada. Tal como se podrá comprobar posteriormente, este controlador tiene un comportamiento equivalente al primero pero tiene la ventaja de que el término de la gravedad se puede calcular fuera de línea, lo que proporciona una estrategia más eficiente y más simple de implementar.

Las cuatro estrategias de control siguientes también se basan en el de Takegaki y Arimoto y dan una solución global y asintóticamente estable al control de posición de sistemas robot usando únicamente la realimentación de la posición. Emplean también la técnica de modificación de la energía pero en lugar de reducir la energía mediante realimentación de la velocidad lo hacen utilizando un compensador lineal de primer orden o bien utilizando un filtro (Kelly, 1993).

Así, las leyes de control con compensador lineal son similares a la del PD con compensación de la gravedad puesto que el controlador está compuesto por una compensación de la gravedad y por un control proporcional derivativo, utilizando en lugar de la medida de la velocidad un compensador lineal (Berghuis, Nijmeijer, 1992).

En este tipo de controladores también se tiene la posibilidad de compensar el término de la gravedad en cada instante o en la posición final deseada. En este último caso, tal como ocurría con el controlador de Takegaki y Arimoto, se mantiene la estabilidad global y asintótica del sistema siempre que la ganancia K_p sea suficientemente grande.

Además del compensador lineal se puede obtener la estimación de la velocidad utilizando otro método adicional, la diferenciación aproximativa. En estas estrategias se aprecia como además de la compensación del término de la gravedad tenemos la parte proporcional derivativa que utiliza un filtro para obtener la estimación de la medida de la velocidad, pudiéndose realizar la compensación de la gravedad (al igual que en los casos anteriores) en cada instante o en la posición deseada. Como antes en este último caso se garantiza la estabilidad global y asintótica del sistema reduciéndose además la complejidad computacional del algoritmo de control puesto que el cálculo de la matriz de gravedad se puede calcular fuera de línea.

Un problema común a todos los controladores citados anteriormente reside en la compensación de las fuerzas gravitatorias ya que se debe realizar el cálculo del término de la compensación de la gravedad, y obtener una buena estimación de estas fuerzas de gravedad no es sencillo puesto que dependen de la masa de la carga del sistema robotizado, y esta masa muchas veces es desconocida. Además, en estos reguladores cualquier error en la estimación de las fuerzas de gravedad puede provocar una variación en el punto de equilibrio y por tanto un error de posición en régimen permanente. Una forma de reducir este error consiste en utilizar ganancias altas pero esto provoca que el sistema sea más sensible a los ruidos además de producir excitación de alta frecuencia.

Un remedio práctico para compensar los efectos de la gravedad es introducir una acción integral en la ley de control. De esta forma el controlador tipo PID incorpora en la ley de control un término integral (la integral del error) que sustituye el término de la gravedad por la acción integral (Wen, Murphy, 1990).

La última estrategia de control basada en la pasividad que se va a estudiar tiene al igual que antes una estructura tipo PID. Fue presentada por (Ortega, Loria, Kelly, 1991) y evita tanto el problema del conocimiento de la carga del robot gracias a un primer término integral como el problema de realimentación de la velocidad que tenían los controladores tipo PD mediante la adición de un segundo término integral en el filtro de la velocidad. Estos dos integradores son los que motivan el nombre de la estrategia de control: PI²D. En este caso se puede observar como el controlador está constituido de cuatro partes. La primera $(-K_p e)$ corresponde al término proporcional. La segunda $(-K_i \int e dt)$ es un remedio práctico para compensar los efectos de la gravedad (término integral en un PID); la tercera $(-K_d \dot{\vartheta})$ es el término derivativo que además incorpora un filtro para obtener la velocidad evitando los problemas que surgían de la medida directa de ésta, y por último tenemos un cuarto término con una acción integral extra en el filtrado de la velocidad.

Los resultados y pruebas se han realizado sobre el robot real, lo que se hizo fue aplicar diferentes trayectorias tanto en el espacio de nudos como en el espacio cartesiano, a partir de la posición en la que estaba dar a cada una de las articulaciones una posición final y una velocidad común. Esta posición y velocidad fueron elegidas arbitrariamente con el fin de probar distintas configuraciones. Todas las articulaciones se mueven a la vez, y se toma como tiempo total aquella más lenta (por velocidad y distancia a recorrer), así se consiguió que todo el robot se moviese de una forma uniforme.

A partir de las cuentas leídas en cada uno de los encoders se calculó en radianes las posiciones absolutas por las que iban pasando cada una de las articulaciones, mostrándose en grados con el fin de facilitar su lectura.

4.3.2.1. PD + G(q)

Como se ha comentado el controlador PD + compensación de la gravedad fue propuesto en (Takegaki y Arimoto, 1981). Este control es un PD con compensación de la

gravedad en la posición actual, el cual necesita un conocimiento de $G(q, \theta)$ en toda la trayectoria, cuya ley de control es:

$$\tau_e = -K_p e - K_d \dot{q} + G(q, \theta) \quad (4.15)$$

donde:

\dot{q} es la señal de velocidad obtenida mediante derivación de la posición.

$e \equiv q - q_d$: error de posición.

q_d : posición deseada.

$K_p = K_p^T > 0, K_d = K_d^T > 0$: ganancias proporcionales y derivativas

El controlador consta de dos partes. La primera es una realimentación lineal del estado mientras que la otra es una compensación de las fuerzas gravitatorias. La parte de compensación de la gravedad junto con la parte de realimentación proporcional modifican la energía potencial del sistema robot de la forma deseada. Mientras que el término derivativo estabiliza asintóticamente el sistema.

Para obtener la compensación de las fuerzas gravitatorias partimos de la ecuación de la dinámica:

$$\tau = M(q, \theta)a + C(q, \dot{q}, \theta)\dot{q} + G(q, \theta) \quad (4.16)$$

haciendo que $a = 0$ y $\dot{q} = 0$ y resolviendo el PDI tal y como aparece en capítulo 2 de la presente tesis.

4.3.2.2. Resultados reales de las ejecuciones para una trayectoria en el espacio de nudos del Robot con curvas de Hermite sin puntos intermedios

Tabla 4-17 Comparación de referencia y posición para las 6 articulaciones del controlador PD+G(q)
(abcisas: posición (grados); ordenadas: período de muestreo (Ts=0.005s))

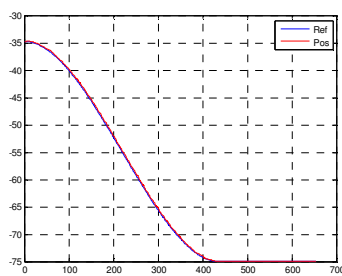


Figura 4-82 Articulación 1

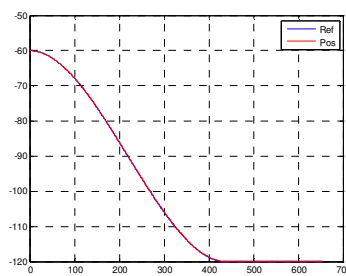


Figura 4-83 Articulación 2

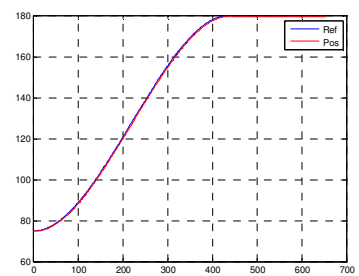


Figura 4-84 Articulación 3

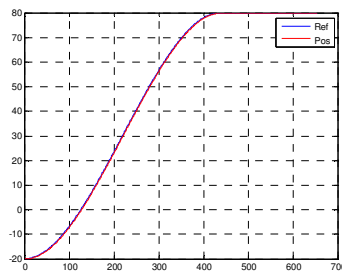


Figura 4-85 Articulación 4

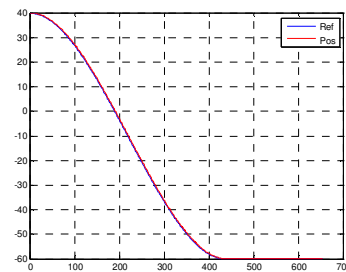


Figura 4-86 Articulación 5

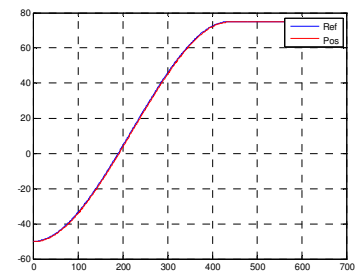


Figura 4-87 Articulación 6

Tabla 4-18 Acciones de control del controlador PD+G(q)
(abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))

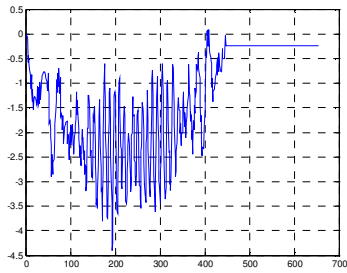


Figura 4-88 Articulación 1

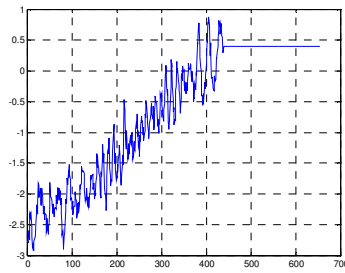


Figura 4-89 Articulación 2

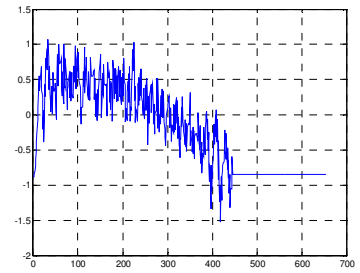


Figura 4-90 Articulación 3

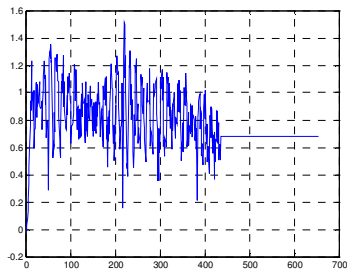


Figura 4-91 Articulación 4

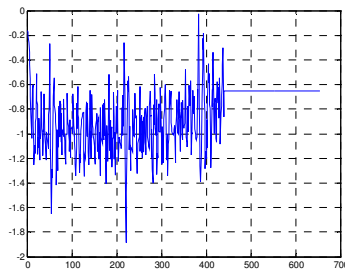


Figura 4-92 Articulación 5

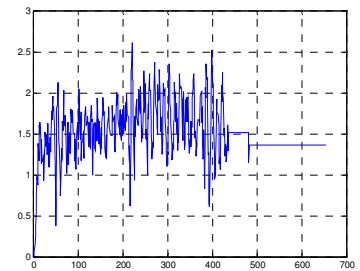


Figura 4-93 Articulación 6

4.3.2.3. Resultados reales de las ejecuciones para una trayectoria en el espacio cartesiano del Robot con curvas polinomiales de tercer orden con interpolación en puntos intermedios

**Tabla 4-19 Comparación de referencia y posición para las 6 articulaciones del controlador PD+G(q)
(abscisas: posición (grados); ordenadas: período de muestreo (Ts=0.005s))**

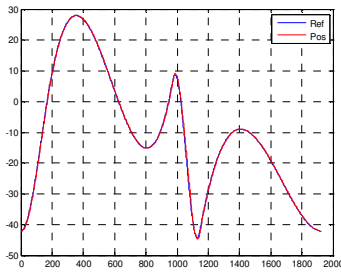


Figura 4-94 Articulación 1

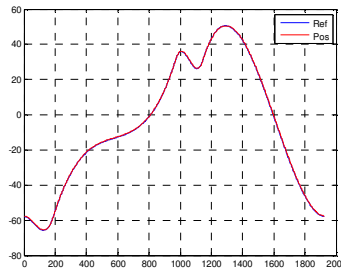


Figura 4-95 Articulación 2

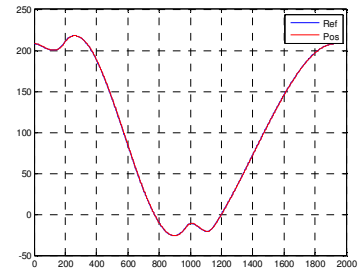


Figura 4-96 Articulación 3

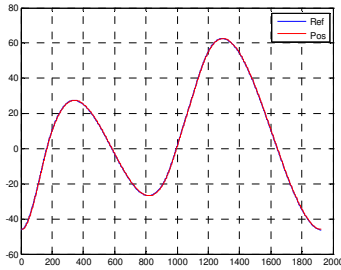


Figura 4-97 Articulación 4

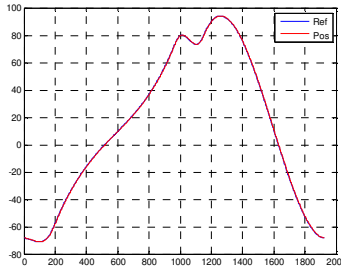


Figura 4-98 Articulación 5

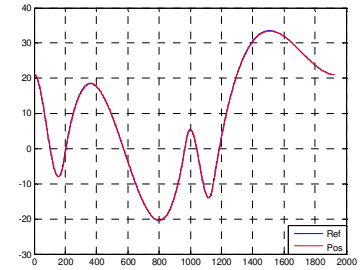


Figura 4-99 Articulación 6

Tabla 4-20 Acciones de control del controlador PD+G(q)
(abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))

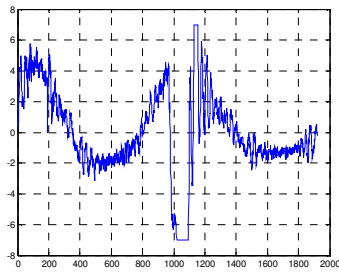


Figura 4-100 Articulación 1

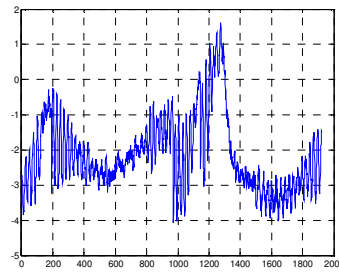


Figura 4-101 Articulación 2

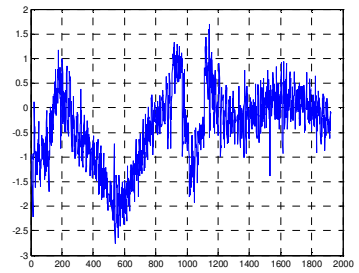


Figura 4-102 Articulación 3

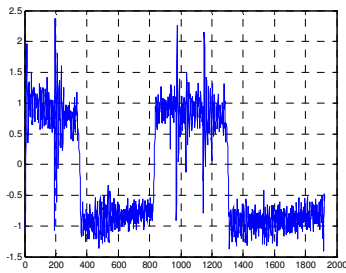


Figura 4-103 Articulación 4

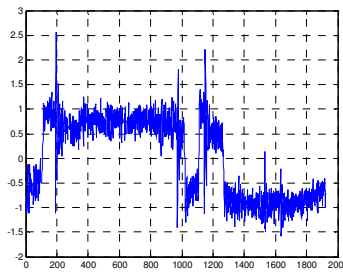


Figura 4-104 Articulación 5

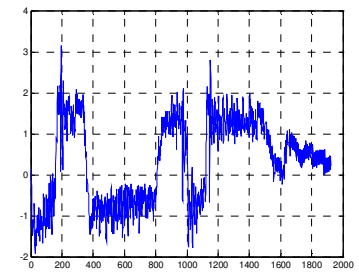


Figura 4-105 Articulación 6

4.3.2.4. Resultados reales de las ejecuciones para una trayectoria de Fourier

Tabla 4-21 Comparación de referencia y posición para las 6 articulaciones del controlador PD+G(q)
(abcisas: posición (rad); ordenadas: periodo de muestreo (Ts=0.005s))

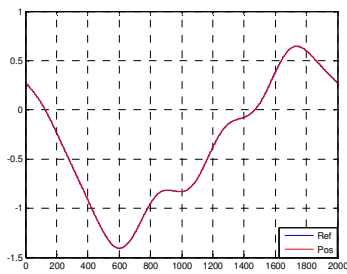


Figura 4-106 Articulación 1

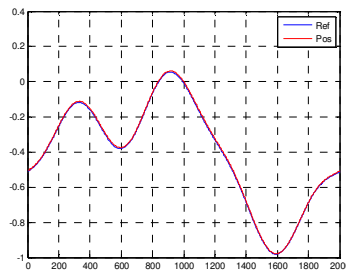


Figura 4-107 Articulación 2

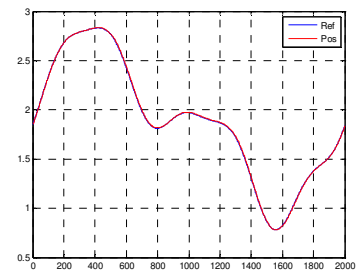


Figura 4-108 Articulación 3

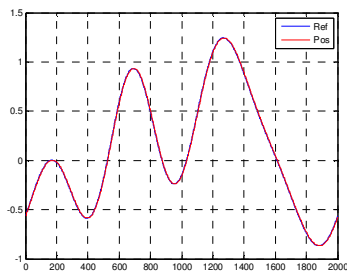


Figura 4-109 Articulación 4

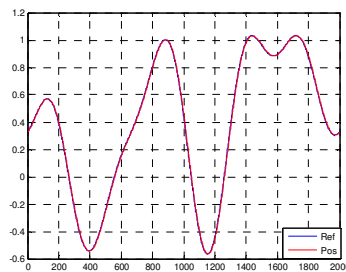


Figura 4-110 Articulación 5

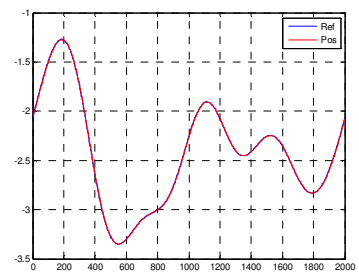


Figura 4-111 Articulación 6

Tabla 4-22 Acciones de control del controlador PD+G(q)
(abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))

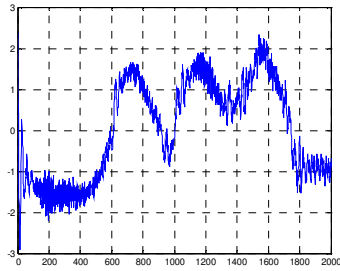


Figura 4-112 Articulación 1

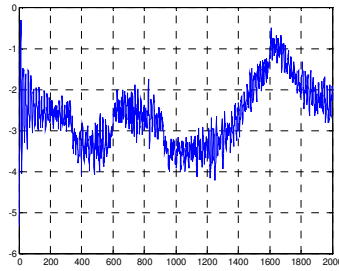


Figura 4-113 Articulación 2

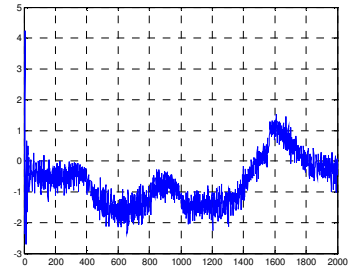


Figura 4-114 Articulación 3

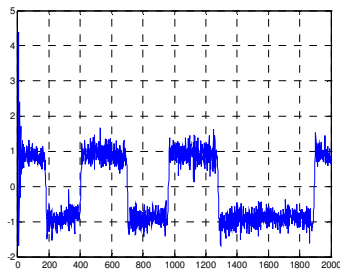


Figura 4-115 Articulación 4

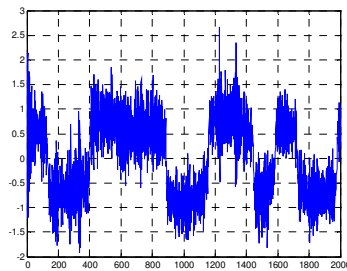


Figura 4-116 Articulación 5

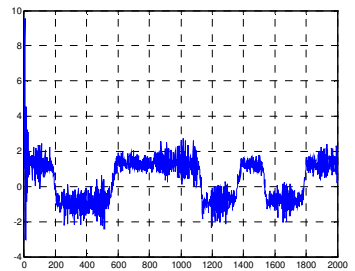


Figura 4-117 Articulación 6

Este controlador realiza una compensación de las fuerzas de gravedad a lo largo de toda la trayectoria. El problema de realizar este tipo de compensación es que cualquier error en la estimación de las fuerzas de gravedad puede provocar una variación en el punto de equilibrio y, por tanto, un error de posición en régimen permanente. Tener una buena estimación de la fuerza de la gravedad no es sencillo pues depende, en este caso, del modelo dinámico del robot.

4.3.2.5. PD+G(qd)

Este controlador también fue propuesto por (Takegaki y Arimoto, 1998). El PD con compensación de la gravedad en la posición deseada, donde la compensación de dichas fuerzas se realiza en la posición deseada $G(q_d, \theta)$ esta es la variación y su ley de control es:

$$\tau_e = -K_p e - K_d \dot{q} + G(q_d, \theta) \quad (4.17)$$

donde:

\dot{q} es la señal de velocidad obtenida mediante derivación de la posición.

$e \equiv q - q_d$: error de posición.

q_d : posición deseada.

$K_p = K_p^T > 0, K_d = K_d^T > 0$: ganancias proporcionales y derivativas

El controlador consta de dos partes. La primera es una realimentación lineal del estado mientras que la otra es una compensación de las fuerzas gravitatorias. La parte de compensación de la gravedad junto con la parte de realimentación proporcional

modifican la energía potencial del sistema robot de la forma deseada. Mientras que el término derivativo estabiliza asintóticamente el sistema.

Para obtener la compensación de las fuerzas gravitatorias partimos de la ecuación de la dinámica (4,16), haciendo que $a = 0$ y $\dot{q} = 0$.

4.3.2.6. Resultados reales de las ejecuciones para una trayectoria en el espacio de nudos del Robot con curvas de Hermite sin puntos intermedios

Tabla 4-23 Comparación de referencia y posición para las 6 articulaciones del controlador PD+G(qd)
(abcisas: posición (grados); ordenadas: período de muestreo (Ts=0.005s))

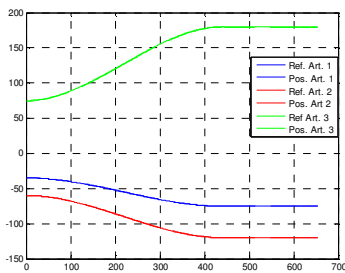


Figura 4-118 Articulaciones 1, 2 y 3

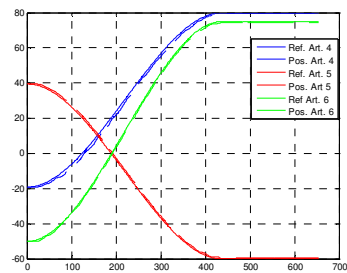


Figura 4-119 Articulaciones 4, 5 y 6

Tabla 4-24 Acciones de control del controlador PD+G(qd)
 (abcisas: acción de control (Nm); ordenadas: período de muestreo (Ts=0.005s))

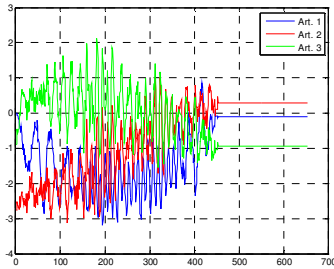


Figura 4-120 Articulaciones 1, 2 y 3

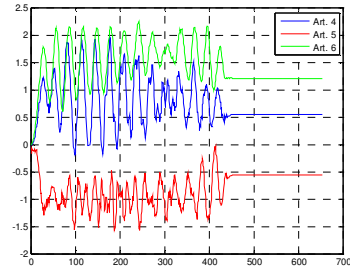


Figura 4-121 Articulaciones 4, 5 y 6

4.3.2.7. PD+CL(q)

Para este controlador, la ley de control que se propone en (Berghuis y Nijmeijer, 1992) es la siguiente:

$$\begin{aligned} \tau &= -K_p e - K_d \dot{x} + G(q, \theta) \\ \dot{x} &= -Lx + K_d e \end{aligned} \tag{4.18}$$

donde L : matriz diagonal positiva

En esta estrategia no se utiliza la velocidad del manipulador, sino que se obtiene una estimación a partir de un compensador lineal de primer orden.

Es importante destacar el cálculo de x y \dot{x} , ya que depende de éste el que la aproximación sea más o menos acertada. Como se aprecia el cálculo de x implica la realización de una integral, para ello se ha utilizado el método trapezoidal que viene dado por la siguiente fórmula:

$$x_k = x_{k-1} + \text{frec} \frac{x_k + x_{k-1}}{2} \quad (4.19)$$

donde *frec* es la frecuencia de muestreo

Cuando se aplica este algoritmo, las variables las inicializamos a un valor nulo, o sea, a cero. Si inicializamos a cero:

$$\begin{aligned} x &= 0 + 0.5 * \text{frec}(0 + 0) = 0 \\ \dot{x} &= L * e - K_d * 0 = L * e \end{aligned} \quad (4.20)$$

Vemos que tenemos \dot{x} con un valor $L * e$, pero no sabemos realmente si es un valor deseable. Analicemos que sentido tiene \dot{x} : para nosotros se trata de una aproximación de la velocidad, ahora bien, partiendo de un sistema en reposo, en el primer instante de muestreo la velocidad del sistema es cero, valor que no coincide con el calculado. Si queremos que inicialmente este valor sea cero, deberemos inicializar \dot{x} a un valor que proporcione el valor deseado.

Este valor se calcularía, en el caso de que el sistema parta del reposo, de la siguiente forma:

$$\dot{x} = 0 \rightarrow 0 = L + e_0 - K_d * x_0 \rightarrow x_0 = \frac{L * e_0}{K_d} \quad (4.21)$$

Por consiguiente debemos inicializar el valor de \dot{x} a $\frac{L * e_0}{K_d}$, donde e_0 corresponde con el error inicial. De la misma forma deberemos iniciar x a cero, ya que se parte del reposo.

Se observó que si inicializamos a 0 se toman valores no deseados en x . En cambio usando la inicialización calculada, los valores son más acertados.

4.3.2.8. Resultados reales de las ejecuciones para una trayectoria en el espacio cartesiano del Robot con curvas polinomiales de tercer orden con interpolación en puntos intermedios

Tabla 4-25 Comparación de referencia y posición para las 6 articulaciones del controlador PD+CL(q)
(abcisas: posición (grados); ordenadas: período de muestreo (Ts=0.005s))

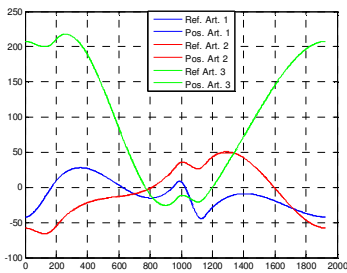


Figura 4-122 Articulaciones 1, 2 y 3

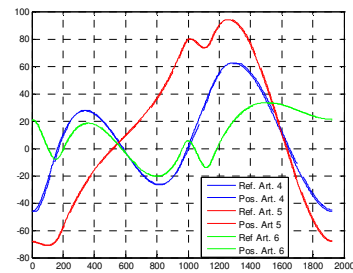


Figura 4-123 Articulaciones 4, 5 y 6

Tabla 4-26 Acciones de control del controlador PD+CL(q)
(abcisas: acción de control (Nm); ordenadas: período de muestreo (Ts=0.005s))

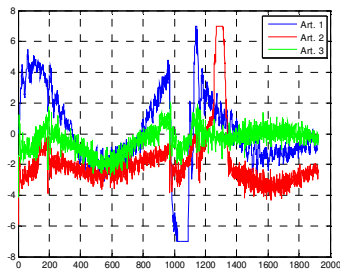


Figura 4-124 Articulaciones 1, 2 y 3

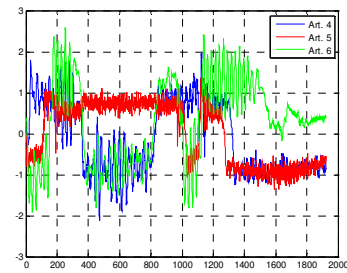


Figura 4-125 Articulaciones 4, 5 y 6

Cuando fijamos K_p y variamos L , obtenemos que cuanto menos es L mayor ganancia tiene la aproximación pero es más inexacta. En cambio si fijamos L y variamos K_p el efecto es el contrario, cuanto mayor es K_p mayor es la ganancia de aproximación y además no pierde exactitud.

Según se muestra en la ecuación del compensador lineal, la matriz L entra directamente en el cálculo de la aproximación, el ir acompañada de un signo negativo implica que el aumento de L influya en una disminución de x y la disminución de L en un aumento de x (fijando la K_p). Mientras que un aumento de K_p constata un aumento de en x y la disminución de la K_p implicará una disminución de x .

Como se puede apreciar, la aproximación que se hace de la velocidad es buena pero amplificada, no se ha podido mejorar debido a que en este controlador la K_d entra directamente tanto en la aproximación de la velocidad como en la dinámica del controlador, por ello la sintonización se hace complicada, porque variar K_d significa actuar en las dos ecuaciones.

El efecto sobre la posición es que disminuyendo L o aumentando K_p obtenemos una aproximación con mayor ganancia pero no se llega a la posición deseada mientras que aumentando L o disminuyendo K_p se sobrepasa.

4.3.2.9. PD+CL(qd)

Para este controlador, la ley de control que se propone en (Berghuis y Nijmeijer, 1992) es la siguiente:

$$\begin{aligned} \tau &= -K_p e - K_d \dot{x} + G(qd, \theta) \\ \dot{x} &= -Lx + K_d e \end{aligned} \quad (4.22)$$

donde L : matriz diagonal positiva

La única diferencia estriba en el hecho de que el término gravitatorio se calcula respecto a la posición deseada no la real, por tanto, se podría hacer, y de hecho se hace, fuera de línea, con el consiguiente ahorro de cálculo, ya que el resultado es exactamente igual.

4.3.2.10. Resultados reales de las ejecuciones para una trayectoria en el espacio de nudos del Robot con curvas de Hermite sin puntos intermedios

Tabla 4-27 Comparación de referencia y posición para las 6 articulaciones del controlador PD+CL(qd)
(abscisas: posición (grados); ordenadas: período de muestreo (Ts=0.005s))

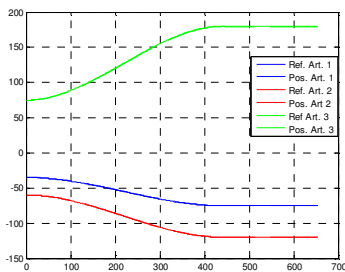


Figura 4-126 Articulaciones 1, 2 y 3

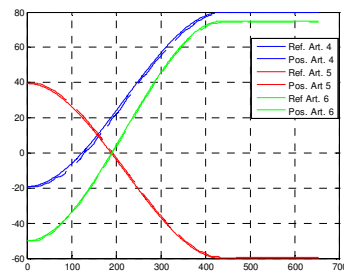


Figura 4-127 Articulaciones 4, 5 y 6

Tabla 4-28 Acciones de control del controlador PD+CL(qd)
 (abcisas: acción de control (Nm); ordenadas: período de muestreo (Ts=0.005s))

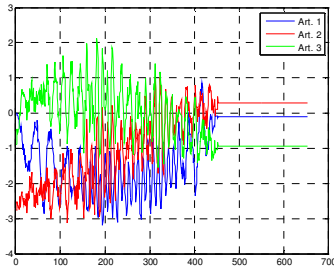


Figura 4-128 Articulaciones 1, 2 y 3

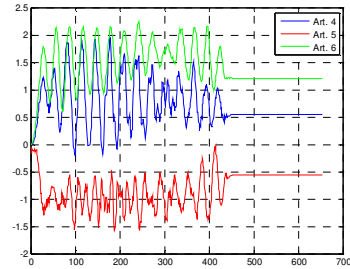


Figura 4-129 Articulaciones 4, 5 y 6

4.3.2.11. PD+DA(q)

La ley de control es la siguiente:

$$\begin{aligned} \tau &= -K_p e - K_d \dot{\vartheta} + G(q, \theta) \\ \dot{\vartheta} &= -A\vartheta + Bq \end{aligned} \tag{4.23}$$

donde A , B : matrices diagonales definidas positivas.

Un método alternativo para obtener la velocidad a la aproximación lineal es la utilización de una diferenciación aproximativa. Para evitar la medida de la velocidad en este caso se realiza un filtro de la posición de las articulaciones q , obteniendo así una buena aproximación de la velocidad de las articulaciones.

En este caso las constantes que se tienen que sintonizar son cuatro las constantes K_p, K_d, A y B . Este controlador tiene la ventaja de que si se realiza una buena

estimación de la velocidad, la sintonización de los valores de K_d y K_p es tan simple como copiarlos del controlador PD y retocarlos un poco si fuera necesario, ya que al contrario que en el compensador lineal, no hay ninguna constante en la ecuación de la ley de control que afecte o se use en la estimación de la velocidad.

Por lo que se ha señalado, lo primero que intentaremos hacer es una buena estimación de la velocidad. Para ello usaremos para el cálculo de ϑ mediante un algoritmo con las siguientes ecuaciones:

$$\begin{aligned}\dot{x} &= -A \cdot x - A \cdot B \cdot q \\ \vartheta &= x + B \cdot q\end{aligned}\tag{4.24}$$

Observamos que con estos valores sintonizados, el sistema llega a la posición final deseada con un comportamiento sobreamortiguado.

Al igual que en el compensador lineal en este controlador tenemos una integral, y de igual forma según la inicialización de ésta se llevará a unos valores u otros.

La realización de la integral se calculará, como en el compensador lineal, con el método trapezoidal.

El cálculo de la aproximación de la velocidad ϑ se efectuará mediante las ecuaciones ya vistas:

$$\begin{aligned}\dot{x} &= -A \cdot x - A \cdot B \cdot q \\ \vartheta &= x + B \cdot q\end{aligned}\tag{4.25}$$

Generalmente cuando se aplica el algoritmo para calcular la aproximación, las variables las inicializaremos a un valor nulo, o sea, a cero.

Si inicializamos a cero veamos que ocurriría, si partimos del reposo la velocidad inicial deberá ser nula, pero en este caso tenemos que A y B no son matrices nulas, por tanto $A \cdot B$ tampoco será nula, la frecuencia de muestreo $frec$ tampoco es nula y si tomamos q arbitraria podemos considerarla como no nula, esto implica una $\vartheta(v)$ no nula, valor no correcto al partir de un sistema en reposo. Si consideramos el filtro, \dot{x} podemos tomarla como una aproximación a ϑ , por tanto inicialmente también nula. Si queremos que estos valores inicialmente sean 0, deberemos inicializar x a un valor que proporcione el valor deseado. Este valor se calcularía, en el caso de que el sistema parta del reposo, de la siguiente forma:

$$\vartheta = 0 \rightarrow 0 = -B \cdot q_0 - x_0 \rightarrow x_0 = -B \cdot q_0 \quad (4.26)$$

veamos que este valor es aceptable en la otra ecuación:

$$x = -B \cdot q_0 \rightarrow x = -A \cdot (-B \cdot q_0) - A \cdot B \cdot q_0 = A \cdot B \cdot q_0 - A \cdot B \cdot q_0 = 0 \quad (4.27)$$

por consiguiente debemos iniciar el valor de x a $-B^* q_0$, donde q_0 corresponde con la posición inicial. De la misma forma deberemos iniciar \dot{x} a cero para conseguir una inicialización correcta.

4.3.2.12. Resultados reales de las ejecuciones para una trayectoria en el espacio de nudos del Robot con curvas de Hermite

Tabla 4-29 Comparación de referencia y posición para las 6 articulaciones del controlador PD+DA(q)
(abcisas: posición (grados); ordenadas: período de muestreo (Ts=0.005s))

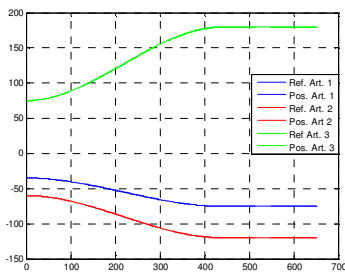


Figura 4-130 Articulaciones 1, 2 y 3

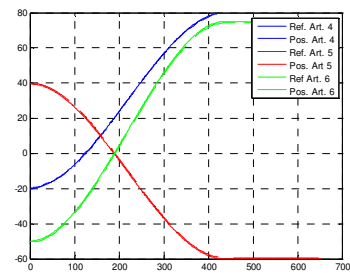


Figura 4-131 Articulaciones 4, 5 y 6

Tabla 4-30 Acciones de control del controlador PD+DA(q)
(abcisas: acción de control (Nm); ordenadas: período de muestreo (Ts=0.005s))

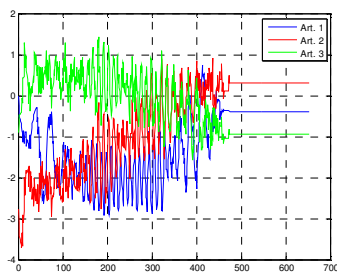


Figura 4-132 Articulaciones 1, 2 y 3

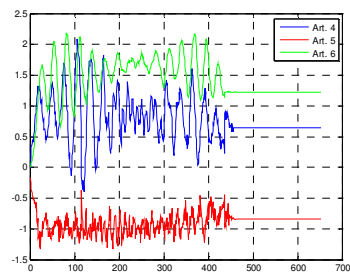


Figura 4-133 Articulaciones 4, 5 y 6

Un aumento en A y en B corresponde a una disminución de la ganancia de la aproximación y una disminución de A y B con un aumento de la ganancia, más concretamente el aumento de A significa una disminución en la ganancia de la aproximación, mientras que su decremento implica un aumento en la ganancia. Caso contrario es lo que ocurre con B , si aumentamos B aumentamos la ganancia de aproximación y si disminuimos B disminuimos también la ganancia de la aproximación.

Una mayor o menor ganancia en la aproximación de la velocidad hace que la acción derivativa sea mayor o menor (si no variamos K_p). Si la aproximación tiene una ganancia mayor la acción derivativa será más grande lo que hará disminuir la acción de control haciendo que no llegue a la posición final, mientras que si la ganancia de la aproximación es menor la acción derivativa será más pequeña lo que hará aumentar la acción de control haciendo que sobrepase la posición final.

4.3.2.13. PD+DA(qd)

La ley de control es la siguiente:

$$\begin{aligned}\tau &= -K_p e - K_d \dot{\vartheta} + G(qd, \theta) \\ \dot{\vartheta} &= -A\vartheta + Bq\end{aligned}\tag{4.28}$$

donde A , B : matrices diagonales definidas positivas.

La única diferencia estriba en el hecho de que el término gravitatorio se calcula respecto a la posición deseada no la real, por tanto, se podría hacer, y de hecho se hace, fuera de línea, con el consiguiente ahorro de cálculo, ya que el resultado es exactamente igual.

4.3.2.14. Resultados reales de las ejecuciones para una trayectoria de Fourier

Tabla 4-31 Comparación de referencia y posición para las 6 articulaciones del controlador PD+DA(qd)
(abcisas: posición (rad); ordenadas: período de muestreo ($T_s=0.005s$))

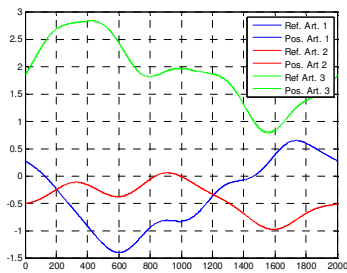


Figura 4-134 Articulaciones 1, 2 y 3

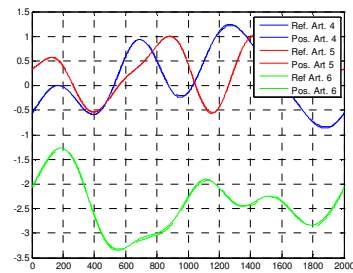


Figura 4-135 Articulaciones 4, 5 y 6

Tabla 4-32 Acciones de control del controlador PD+DA(qd)
(abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))

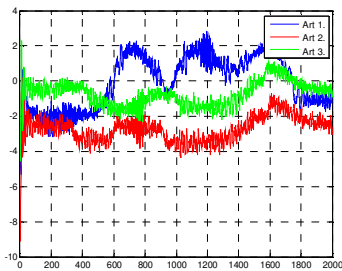


Figura 4-136 Articulaciones 1, 2 y 3

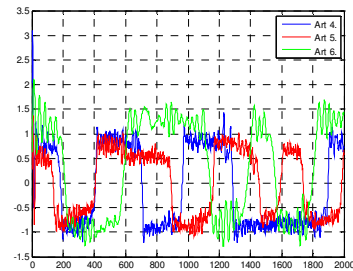


Figura 4-137 Articulaciones 4, 5 y 6

4.3.2.15. PID

Para este controlador, la ley de control que se propone es la siguiente:

$$\tau_e = -K_p e - K_d \dot{q} - K_i \int edt \quad (4.29)$$

donde

\dot{q} es la señal de velocidad obtenida mediante derivación de la posición.

$e \equiv q - q_d$: error de posición.

q_d : posición deseada.

$\int edt$: integral del error

$K_p = K_p^T > 0, K_d = K_d^T > 0, K_i = K_i^T > 0$: ganancias proporcionales, derivativas e integrales.

Como se ha comentado anteriormente, este controlador combina una acción de control proporcional junto con un término derivativo y un término integral. El término proporcional tiene un significado de control “actual” puesto que su valor depende del error actual entre la referencia y la posición de los elementos del robot. Puesto que el término derivativo utiliza la derivada, éste representa una tendencia y por eso se suele decir que es un control “futuro” en el sentido que se anticipa al error. Por último, el integral, que se puede decir que es un error “pasado”; cuanto mayor es el error acumulado, mayor será la acción de control.

4.3.2.16. Resultados reales de las ejecuciones para una trayectoria en el espacio cartesiano del Robot con curvas polinomiales de tercer orden con interpolación en puntos intermedios

Tabla 4-33 Comparación de referencia y posición para las 6 articulaciones del controlador PID (abcisas: posición (grados); ordenadas: período de muestreo ($T_s=0.005s$))

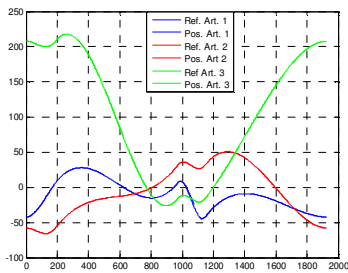


Figura 4-138 Articulaciones 1, 2 y 3

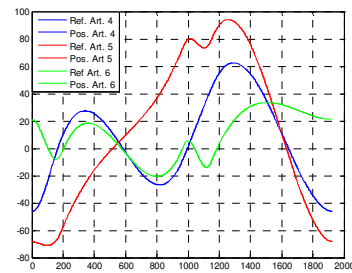


Figura 4-139 Articulaciones 4, 5 y 6

Tabla 4-34 Acciones de Control del controlador PID (abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))

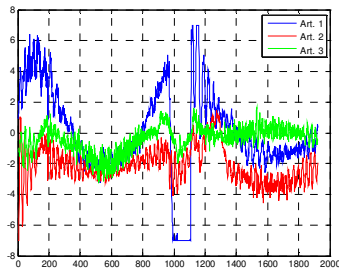


Figura 4-140 Articulaciones 1, 2 y 3

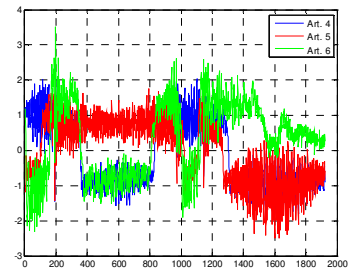


Figura 4-141 Articulaciones 4, 5 y 6

Este controlador es el más utilizado en el control de sistemas físicos, esto es debido a su sencilla implementación y también al hecho, que nos proporciona muy buenos resultados.

Como desventaja se puede también indicar que es difícil de ajustar las constantes. Analicemos que ocurre si se varía la constante proporcional: si la disminuimos lo que ocurre es que se cargue la integral, aumentando la acción integral, así lo que se consigue es que el sistema alcance la referencia sobrepasándola, y la vuelve a perder al cambiar el sentido la acción de control para volverla a alcanzar. Resultado, oscilación del sistema. En caso de aumentar la acción proporcional, se aumenta la velocidad del sistema que hace que el sistema tenga una respuesta subamortiguada que no es deseable, pero como actúa la acción integral el sistema sigue la referencia pero aumentamos considerablemente las acciones de control, con lo que saturamos. Esta constante saturación hace que se puedan dañar los motores.

Si variamos la acción derivativa disminuyéndola, se produce una oscilación en el sistema a la hora de seguir la referencia. En cambio, si es demasiado alta el sistema sobrepasa la posición final más que antes, y que tiende a volver a la posición correcta más lentamente.

Si variamos la acción integral no se observa que haya mucha variación en la respuesta del sistema ante cambios en la acción integral, únicamente un aumento de la acción proporcional cuando la disminuimos.

4.3.2.17. PI²D

Para este controlador, la ley de control propuesta es la siguiente:

$$\tau = -K_p e - K_i \int_0^t e dt - K_d \dot{\vartheta} - K_i \int_0^t \dot{\vartheta} dt \quad (4.30)$$

$$\dot{\vartheta} = -A\vartheta + B\dot{q}$$

Que también podemos ver de forma más cómoda como:

$$\begin{aligned} \tau &= -K_p e - K_v \vartheta - K_i \left(\int_0^t e dt + \int_0^t \vartheta dt \right) \\ \dot{\vartheta} &= -A\vartheta + B\dot{q} \end{aligned} \tag{4.31}$$

Como podemos apreciar se trata de un PID con una aproximación de la velocidad por diferenciación aproximativa al que se le añade otra acción integral, también de la aproximación de la velocidad

Los valores de las constantes K_p , K_d y K_i , serán muy parecidas a las del PID y las A y B se deberán recalcular para sintonizar con la estructura del controlador. La K_i se tendrá que modificar un poco debido a la inclusión del segundo integrador.

Hay que comentar que puede ser que no se llegue a apreciar la acción integral pero que en realidad si hay.

Como se puede observar pasa exactamente lo mismo que con el PID, la acción integral (ahora doble), hace que el sistema confluya asintóticamente a la posición final.

La ley de control podemos verla de otra forma para mejorar su conversión a algoritmo de la siguiente forma:

$$\begin{aligned}
 \tau &= -K_p e - K_v \vartheta - K_i \rho \\
 \rho &= \xi + \psi \\
 \dot{\xi} &= e \\
 \dot{\psi} &= \vartheta \\
 \dot{\vartheta} &= -A\vartheta + B\dot{q}
 \end{aligned}
 \tag{4.32}$$

El cálculo de la aproximación de la velocidad se realizará como se indicó en el apartado correspondiente al regulador con diferenciación aproximativa, mientras que la integral del error como se indicó en el apartado del PID. El cálculo de la integral que nos quedase realiza de la misma forma por el método trapezoidal usado en la integral del error.

Otra forma de cálculo sería unir las dos integrales en una sólo, consiguiendo así menos operaciones:

$$\rho = \xi + \psi \rightarrow \dot{\rho} = \dot{\xi} + \dot{\psi} = e + \vartheta
 \tag{4.33}$$

usando el método trapezoidal:

$$\rho_k = \rho_{k-1} + \text{frec} \cdot \frac{\rho_k + \rho_{k-1}}{2} \rightarrow \rho_{k-1} + \text{frec} \cdot \frac{(e_k + \vartheta_k) + (e_{k-1} + \vartheta_{k-1})}{2}
 \tag{4.34}$$

Con este método ahorramos unas cuantas operaciones además de una variable.

4.3.2.18. Resultados reales de las ejecuciones para una trayectoria de Fourier

Tabla 4-35 Comparación de referencia y posición para las 6 articulaciones del controlador PI²D
(abcisas: posición (rad); ordenadas: período de muestreo (Ts=0.005s))

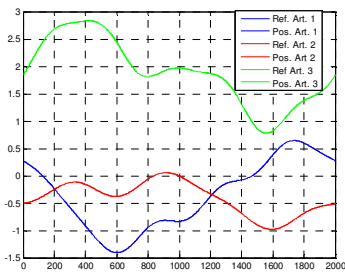


Figura 4-142 Articulaciones 1, 2 y 3

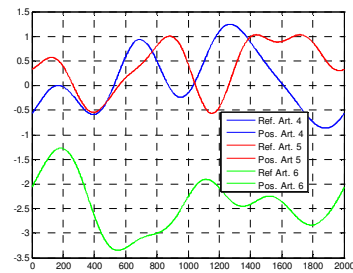


Figura 4-143 Articulaciones 4, 5 y 6

Tabla 4-36 Acciones de control del controlador PI²D
(abcisas: acción de control (Nm); ordenadas: período de muestreo (Ts=0.005s))

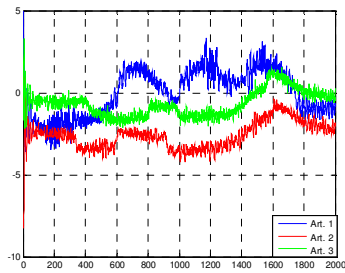


Figura 4-144 Articulaciones 1, 2 y 3

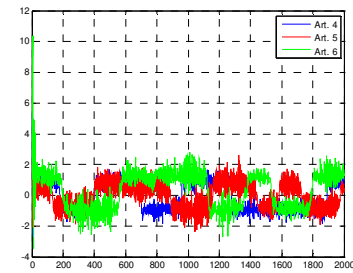


Figura 4-145 Articulaciones 4, 5 y 6

CAPÍTULO 5

PLATAFORMA EXPERIMENTAL DE

CONTROL DE ROBOTS INDUSTRIALES.

APLICACIÓN AL PUMA 560

5.1. ARQUITECTURA HARDWARE DEL PUMA 560

El robot PUMA 560 es un robot manipulador industrial provisto de seis grados de libertad y constituido exclusivamente por pares de revolución. En esencia, el robot está integrado por dos partes diferenciadas, las cuales se describen a continuación: la unidad de control y el brazo robot, ambos unidos eléctricamente mediante dos mangueras.

5.1.1. Brazo Robot

El brazo robot se compone, como se mencionaba anteriormente, de seis barras unidas por pares de revolución actuados mediante motores de corriente continua de 40 voltios. Cada uno de los pares está provisto de un potenciómetro y de un encoder incremental, los cuales permiten determinar las posiciones angulares correspondientes.

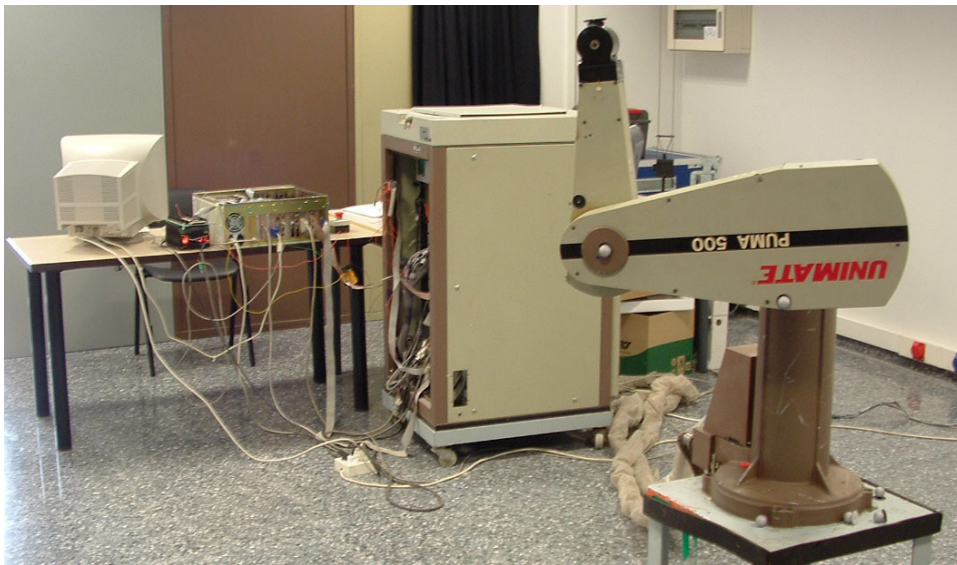


Figura 5-1 PUMA 560

Además se le puede acoplar la herramienta en la última articulación. En complemento a ello se ha colocado un control por aire de apertura / cierre, como por ejemplo para una mano

5.1.2. Unidad de Control

La unidad de control de un PUMA 560 está formada por los siguientes módulos:

- **Fuente de Potencia:** Este módulo se encarga de suministrar la corriente a todo el robot, tanto los altos voltajes para el brazo como las bajas tensiones para los demás módulos de la unidad de control. Posee varios transformadores para convertir el suministro de la red eléctrica a los voltajes necesarios.
- **Panel de Control:** Permite el inicio y la parada del robot. Inicialmente, el transformador de baja potencia es conectado a fin de alimentar la unidad de control para, posteriormente, proceder a la conexión de los transformadores de alta potencia, los cuales proporcionan el suministro de potencia para el brazo robot. Ambos transformadores están situados en la fuente de potencia.
- **Módulo de Control:** Posibilita el control del robot. A fin de programar el robot, pueden conectarse una unidad de disco y un terminal a este módulo. El módulo está provisto de tarjetas de memoria, un procesador LSI-11/02, procesadores 6503, etc.
- **Módulo I/O:** Este módulo, controlado por el anterior, actúa como puente de comunicación con otros sistemas. Dispone de varios puertos de entrada y salida.
- **Módulo Amplificador:** Amplifica la potencia suministrada por la fuente de alimentación a fin de suministrar al brazo robot de acuerdo a la referencia recibida del módulo de control. Incluye siete tarjetas amplificadoras: tres a alta potencia para los tres primeros nudos, tres a baja potencia para los nudos de la muñeca y uno para el control de calentamiento de los amplificadores, el sistema de frenos y la operación de la herramienta.

En la siguiente figura se muestra una imagen real de la unidad de control del PUMA 560.



Figura 5-2 Imagen de la Unidad de Control del PUMA 560

5.1.3. Arquitectura Original

El robot manipulador PUMA 560 está provisto de un controlador UNIMATE MARK III, el cual incluye un procesador DEC LSI-11/02, seis paneles con servos digitales y analógicos y seis paneles amplificadores, tres a alta y tres a baja potencia. Cada uno de estos paneles amplificadores suministra al actuador del nudo correspondiente del brazo robot. Los servos digitales están provistos de un procesador Rockwell 6503 a fin de interpolar las trayectorias proporcionadas por el procesador LSI-11/02. Los paneles analógicos, los cuales están conectados a los paneles digitales a través de convertidores digitales/analógicos (DAC), contienen un servo analógico, diseñado como un controlador de retroalimentación activa, para cada uno de los actuadores de los nudos. La interacción con el usuario tiene lugar a través del VAL-II, el cual constituye el sistema operativo del procesador LSI-11/02.

La arquitectura original del robot manipulador PUMA 560 es muy similar a la mostrada en la Figura 5.3 (véase Moreira et al. 1996).

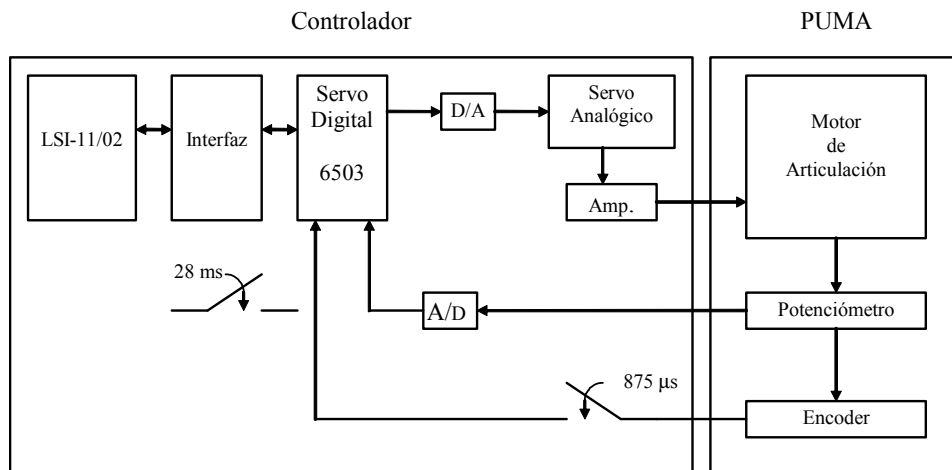


Figura 5-3. Esquema de la arquitectura original de un PUMA 560.

A continuación, se describen los componentes que constituyen dicha arquitectura:

- **Procesador LSI-11/02:** Interpreta los comandos procedentes del VAL-II y transmite los conjuntos de puntos de posición a los seis procesadores 6503 a un ratio de 35.7 muestras por segundo.
- **Servos Digitales:** Seis servos digitales, cada uno de ellos provisto de un procesador 6503, lee las posiciones y velocidades actuales de los encoders a razón de 1.2 kmuestras por segundo. Este alto ratio requiere de la interpolación del conjunto de puntos determinado por el procesador LSI-11/02 a fin de alcanzar un ratio de 1.2 ksets de puntos por segundo. Partiendo de los errores de posición y velocidad, el procesador 6503 calcula los pulsos a transmitir a los servos analógicos a través del DAC.

- **Servos Analógicos:** Los servos analógicos escanean los pulsos en el conjunto de puntos procedentes de los servos digitales a fin de controlar la acción de cada actuador de nudo.
- **Amplificadores:** Los amplificadores de potencia suministran la potencia requerida por los actuadores de los nudos.
- **Encoders Incrementales:** Los encoders proporcionan tres señales de salida: los canales A y B y el Pulso Indexado. Los canales A y B están en cuadratura, lo cual permite detectar el sentido de rotación. El pulso indexado permite detectar una revolución completa del encoder. Esta señal, en combinación con la lectura del potenciómetro, permite la calibración de las lecturas de los encoders.
- **Potenciómetros:** La posición angular de los nudos proporcionada por los encoders es relativa, no permitiendo por tanto la determinación de la posición angular absoluta del nudo. Así pues, la lectura inicial de los encoders, durante la inicialización del brazo robot, es corregida mediante la posición angular absoluta del nudo suministrada por los potenciómetros. Los potenciómetros son leídos a través del convertidor analógico/digital (ADC).

Tal como se ha indicado anteriormente, la arquitectura de control original incluye un procesador LS-11/02, el cual interpreta los comandos del VAL-II y genera las trayectorias a transmitir a los seis servos digitales constituidos por los procesadores Rockwell 6503, los cuales generan las señales analógicas a través de los DACs (módulo de control). Estas señales son transmitidas a los amplificadores, los cuales suministran, a su vez, la potencia requerida por el brazo robot (módulo de amplificadores).

El bucle de control se completa con los potenciómetros y los encoders (brazo robot). Cada nudo está equipado con un potenciómetro, el cual proporciona la lectura de la posición angular absoluta, y un encoder, el cual suministra la posición angular relativa y posibilita una determinación más precisa de la posición angular del nudo.

5.1.4. Arquitectura de Control Abierta

Uno de los principales problemas encontrados a la hora de establecer el control de un sistema robótico industrial radica en la unidad de control del robot, ya que ésta constituye en general un subsistema totalmente cerrado. Debido al hecho de que el sistema de control utiliza su propio sistema operativo y no resulta posible modificar el sistema de control, incluso los valores de las ganancias, los usuarios encuentran serios problemas de cara a implementar tanto estrategias de control convencionales como avanzadas por ejemplo control de fuerza o control cooperativo de varios robots. También es importante la posibilidad de programar generación automática de trayectorias, control basado en sensorización externa (por ejemplo, en visión), etc.

A fin de resolver estas dificultades, pueden encontrarse algunas soluciones tal como en (Moreira et al. 1996), donde el hardware de control es modificado, lo cual no resulta normalmente, sin embargo, una tarea trivial ni económica. En (Valera et al. 2003) y (Valera et al. 1998), se propone una solución a estos inconvenientes, presentando una arquitectura de control muy simple, económica y totalmente abierta. A continuación, se describe esta arquitectura de control abierta, donde se reemplaza la unidad de control original del robot manipulador PUMA 560 mediante un PC provisto de tarjetas de adquisición de datos.

5.1.5. Arquitectura de Control Propuesta

La arquitectura de control original presenta muchos inconvenientes para llevar a cabo tareas de alto nivel, ya que el sistema operativo VAL-II está basado, como se mencionaba anteriormente, en una arquitectura cerrada. Además, los controladores de los nudos, así como los algoritmos de generación de trayectorias, no pueden ser modificados.

A fin de evitar estas limitaciones, se propuso la sustitución de la unidad de control mediante un PC. De este modo, el PC accede a las posiciones angulares de los nudos y

genera las acciones de control requeridas para accionar los distintos elementos del brazo robot, tal como se especifica en la Figura 5-4.

Como puede observarse en la Figura 5-5, las lecturas de las posiciones angulares de los nudos del brazo robot son obtenidas a través de las tarjetas de adquisición de datos, la cuales recogen las señales de los potenciómetros y los encoders. Las acciones de control calculadas por medio de los algoritmos de control implementados en el PC son similarmente transmitidas a través de las tarjetas de adquisición de datos a los actuadores correspondientes.

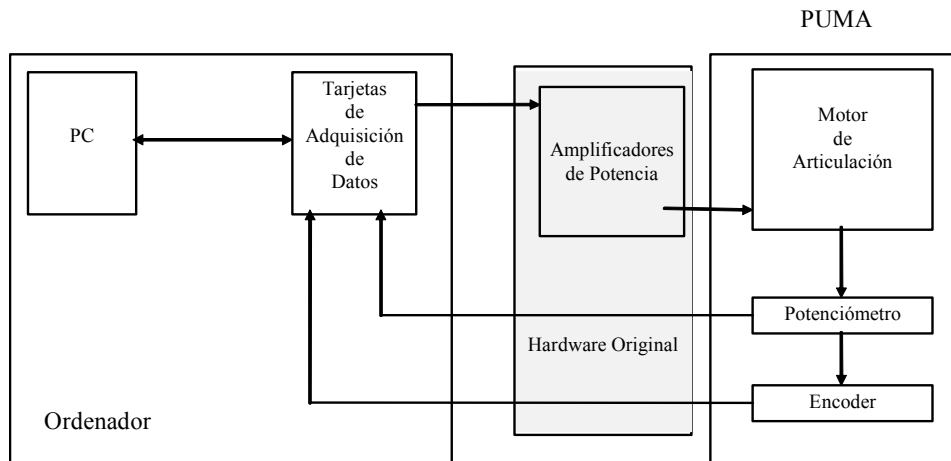


Figura 5-4. Arquitectura de control abierta propuesta.

Las distintas tarjetas de adquisición de datos utilizadas son las siguientes:

- **PCL – 818:** Tarjeta con 16 convertidores A / D, 1 convertidor D / A, 16 entradas y 16 salidas D / D. Es, por tanto, la encargada de leer los potenciómetros de las articulaciones.
- **PCL – 833:** Tarjeta con 3 lectores de encoders y 5 entradas D / D. Esta se encarga de la lectura de los encoders del brazo robot. El modo en el que situaremos la tarjeta para la lectura de los encoders es el de pulso-dirección (One Pulse In), al tener 6 articulaciones, usaremos dos de estas tarjetas.

- **PCL – 726:** Tarjeta con 6 convertidores D / A, 16 entradas y 16 salidas D / D. Encargada de la generación de las señales de referencia de la acción de control. Estas señales se generarán de forma bipolar y serán recogidas directamente por los amplificadores del armario de control. Además con sus salidas digitales generaremos la señal que controle el freno del robot.

Como se puede observar en la siguiente figura, a través de tarjetas de adquisición de datos con convertidores A/D podremos leer los potenciómetros, con tarjetas de adquisición de datos lectoras de encoders conseguiremos la lectura de los encoders, el PC se encargará de generar todo el control, y después las tarjetas de adquisición de datos con convertidores D/A generarán las acciones de control de referencia para los amplificadores de potencia del robot.

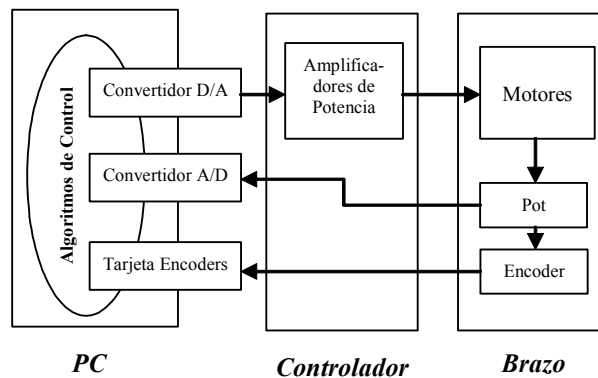


Figura 5-5. Esquema de entradas y salidas de la arquitectura de control propuesta.

Las siguientes figuras muestran la arquitectura de control propuesta ya aplicada al robot PUMA 560.

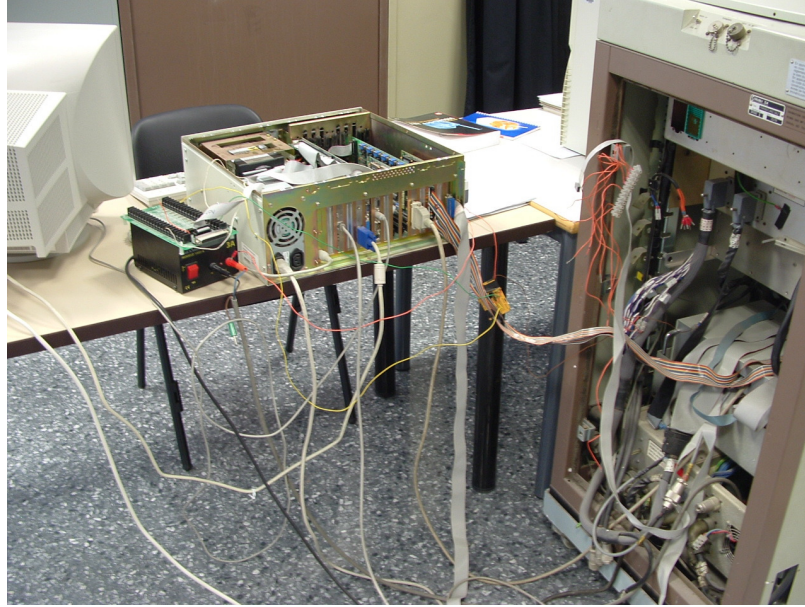


Figura 5-6 Conexión de la Unidad de Control del PUMA 560 al PC Industrial

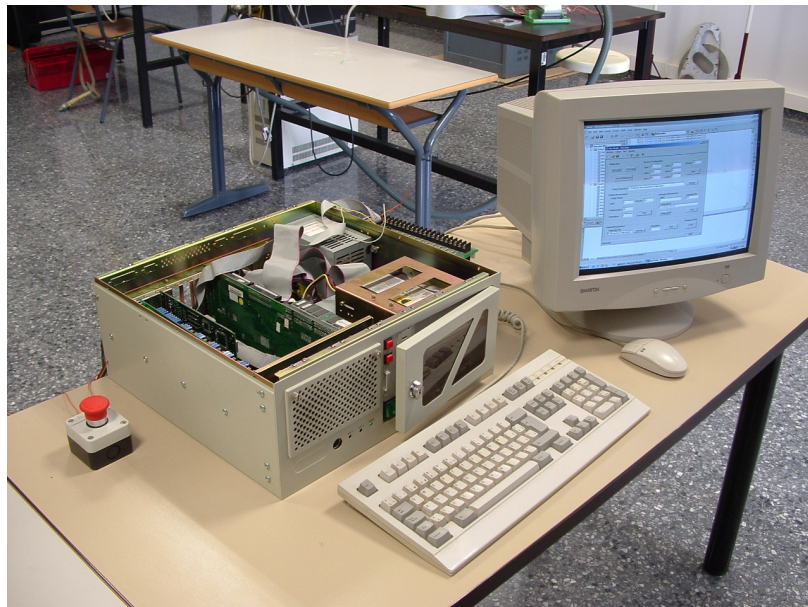


Figura 5-7 Imagen del PC Industrial con la aplicación de control ejecutando

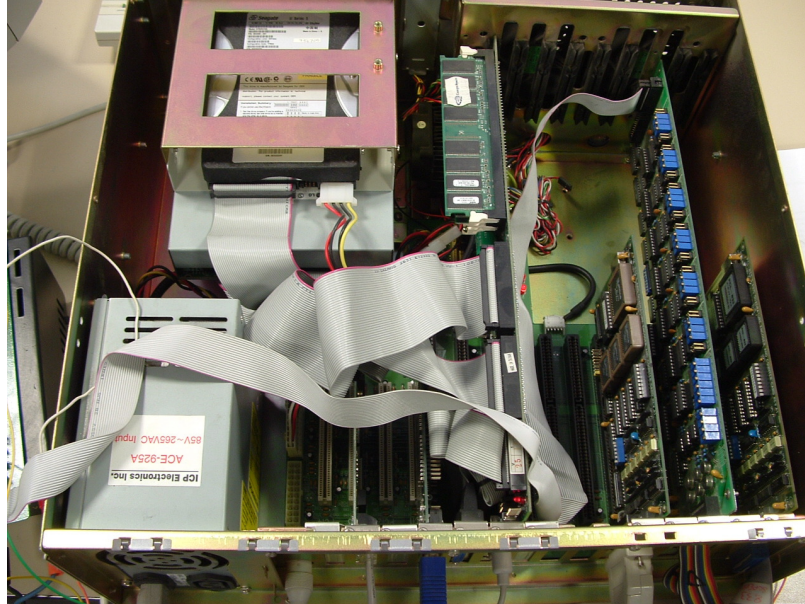


Figura 5-8 Imagen de las tarjetas de adquisición de datos

Con esta arquitectura se conseguirá un sistema totalmente abierto, consiguiendo las ventajas que este hecho conlleva.

5.1.6. Circuito de freno

En el punto anterior hemos visto la arquitectura que nos va a permitir realizar el control por computador, pero queda resolver como se realiza el frenado del robot.

Tenemos dos opciones, podemos dejar el sistema de frenado original el cual se encuentra en la unidad de control, o podemos diseñar un circuito que nos permita un control del freno por computador.

Con la primera opción se nos presentan varios problemas. En primer lugar, se necesitan dos personas para realizar el control, una que maneje el computador y otra que pulse el freno cada vez que finalice el control, por tanto se diseñó esta segunda opción.

Para ello se diseñó el siguiente circuito que permite el frenado automático, es decir, serán dos salidas digitales de la tarjeta PCL-726 las encargadas de enviar un 0 (freno) o un 1 (quitar freno) a la unidad de control. Además se añadió un pulsador del operador que le permita un frenado de emergencia al robot en caso de un comportamiento anómalo.

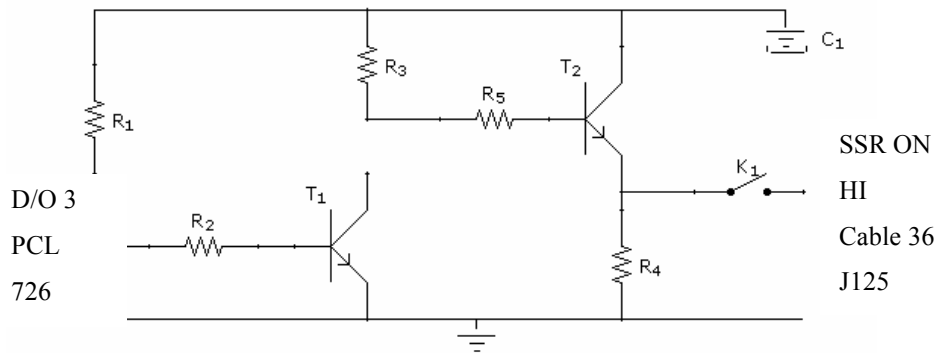


Figura 5-9 Circuito de Freno

Los elementos del circuito anterior serían los siguientes:

- R1. Resistencia 3K9 Ω
- R2. Resistencia 1K5 Ω
- R1. Resistencia 5K6 Ω
- T1. Transistor BC 337-25
- R5. Resistencia 3K3 Ω
- T2. Transistor BC 337-25

- R4. Resistencia 560 Ω
- K1. Pulsador de emergencia
- C1. Condensador 100 μF

5.2. ARQUITECTURA SOFTWARE DE CONTROL

Para poder implementar los algoritmos de control ya expuestos en capítulos anteriores sobre la arquitectura hardware propuesta y aplicada, también comentada en la sección anterior, se ha procedido a la implementación de una serie de algoritmos encargados del control del PUMA 560 en tiempo real.

Estos algoritmos han sido implementados y diseñados utilizando dos herramientas software muy diferentes, una de ellas ha sido el uso de la Real-Time Workshop de Matlab, mientras que la otra ha sido la implementación de dichos algoritmos en un lenguaje de alto nivel, en este caso Visual C++.

5.2.1. Introducción a la Real-Time Workshop

La Real-Time Workshop[®] permite generar código a partir de modelos Simulink[®] para crear aplicaciones autónomas de modelos que operan tanto en tiempo real como no, estas aplicaciones además se pueden ejecutar en diferentes entornos.

En la presente tesis el código se ejecuta en un PC Industrial conectado directamente al robot PUMA 560, siendo esta la arquitectura de control utilizada como se ha comentado en el apartado anterior.

La siguiente figura muestra el papel de la Real-Time Workshop® en el proceso del diseño del Software.

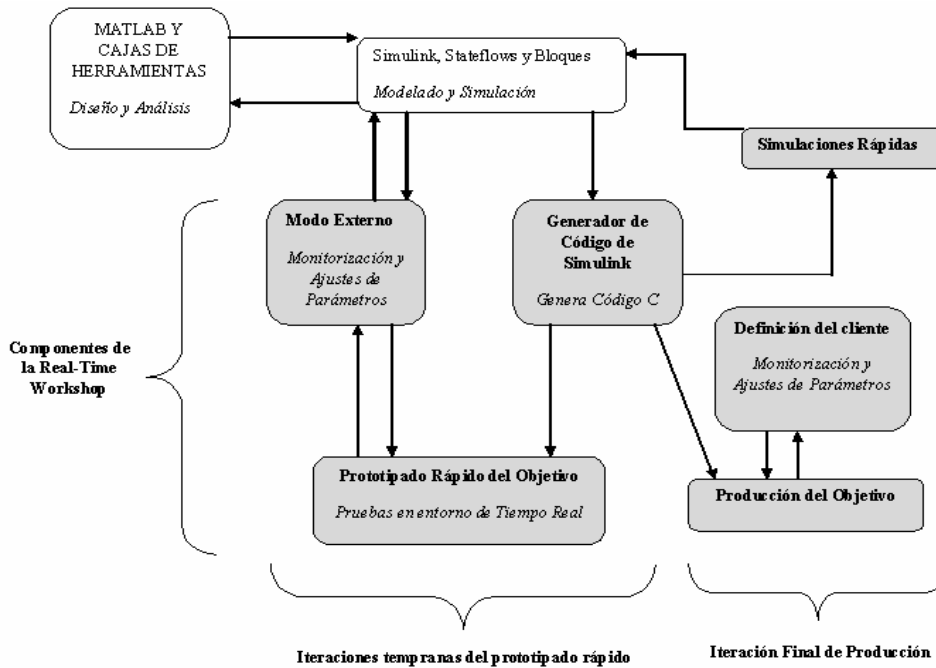


Figura 5-10 Papel de la RTW en la fase de Diseño

Otra de las características de la Real-Time Workshop® es que soporta un prototipado rápido, es decir, permite conceptualizar soluciones utilizando un entorno de modelado de diagramas de bloques y realiza una evaluación del sistema antes de ejecutarlo sobre el modelo físico.

5.2.2. La Arquitectura Abierta de la Real-Time Workshop

La Real-Time Workshop[®] como se ha comentado anteriormente es un sistema abierto diseñado para su uso sobre una amplia variedad de entornos operativos y tipos de hardware, por lo que es la herramienta idónea para la realización de la presente tesis.

La Figura 5-11 muestra los elementos claves de la Real-Time Workshop[®]

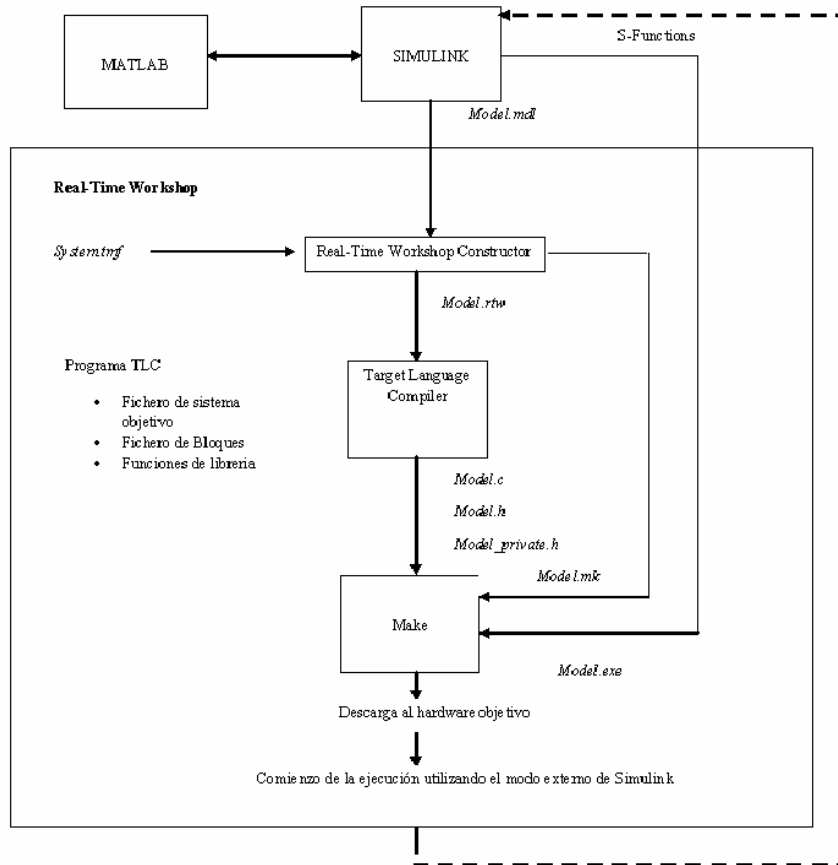


Figura 5-11 Arquitectura Abierta de la RTW

Se puede configurar el proceso de generación del programa adaptándose a las necesidades modificando un conjunto de componentes, tales como:

- Simulink[®] y el fichero modelo (model.mdl). En esta tesis se han utilizado S-functions y Embedded functions, que se comentarán más adelante para ampliar el VHLL de Simulink[®] añadiendo nuevos bloques de propósito general o bien incorporando código a un bloque.
- El modelo intermedio (model.rtw)
- Target Language Compiler (TLC)

5.2.3. S-Function

Una S-function se puede definir como una descripción escrita en un lenguaje de programación de un bloque de Simulink[®]. Las S-functions pueden ser escritas en MATLAB, C, C++, Ada, o Fortran. En la presente tesis las S-functions fueron escritas en C. Estas funciones son compiladas como los ficheros MEX utilizando la utilidad *mex*. Al igual que con otros ficheros MEX, son vinculados dinámicamente en MATLAB cuando sea necesario.

Las S-functions utilizan una sintaxis especial de invocaciones que le permiten interactuar con los solucionadores de ecuaciones de Simulink[®]. Esta interacción es muy similar a la interacción que tiene lugar entre los solver y los bloques de Simulink[®] construidos.

Se pueden utilizar S-functions en la Real-Time Workshop[®]. También se puede personalizar el código generado por la Real-Time Workshop[®] para S-functions escribiendo un fichero de TLC, que no ha sido en nuestro caso.

En este trabajo se ha utilizado las S-function para crear el bloque Simulink que nos ha permitido acceder a las tarjetas de adquisición de datos utilizadas en la arquitectura abierta de control, concretamente la PCL-833, puesto que, esta tarjeta no se encuentra entre las tarjetas de adquisición de datos que contiene Simulink.

Este bloque contiene el código necesario para interconectar con el dispositivo específico, como se ha comentado la PCL-833, gracias a las S-Functions tendremos el controlador de la tarjeta que se puede empotrar en el código del modelo que se va a generar con la RTW.

En la siguiente figura se muestra el bloque para la PCL-833.

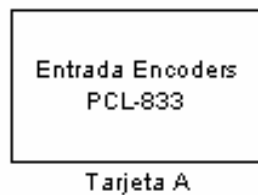


Figura 5-12 Bloque del controlador de la PCL-833

Para añadir este bloque, se hace exactamente igual que cualquier otro bloque Simulink. Se conecta a la entrada del modelo exactamente cualquier otro bloque.

Como se ha comentado este bloque de controlador de dispositivo ha sido implementado como una S-function escrito en C. El código C para para este bloque es compilado como un fichero MEX, por tanto, solo puede ser llamado por Simulink.

Ese mismo código C puede además de ser compilado y enlazado con el código generado, al igual que cualquier otro código en C que sea una S-Function. Sin embargo, usando el *target file (.tlc)* que corresponde a cada fichero C de la S-Function, el código del controlador será empotrado en el código generado.

Para este bloque se tendrá su fichero en C y su fichero TLC. La siguiente figura muestra el bloque de S-Function si se edita el bloque correspondiente al controlador.

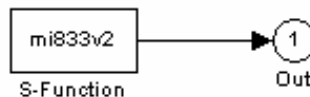


Figura 5-13 Bloque correspondiente a la S-Function del controlador de la tarjeta PCL-833

5.2.4. Modo Externo

El modo externo permite dos sistemas separados (un host y un target) puedan comunicarse. El host es el ordenador donde MATLAB® y Simulink® se están ejecutando. El target es el ordenador donde el ejecutable creado por la Real-Time Workshop® se ejecuta.

El host (Simulink®) transmite mensajes solicitando al target la aceptación de cambios o para subir los datos de la señal. El target responde ejecutando la solicitud. La comunicación del modo externo se basa en una arquitectura cliente/servidor, en el cual Simulink es el cliente y el target el servidor.

El modo externo permite:

- Modificar, o ajustar los parámetros de los bloques en tiempo real. En el modo externo, cualquier cambio de los parámetros en el diagrama de bloques Simulink® automáticamente se descarga en el programa destino que se está ejecutando. Esto permite sintonizar los parámetros del programa sin recompilar. En el modo externo, el modelo Simulink® se convierte en el modo gráfico del programa destino.

- Visualizar e inspeccionar las salidas de los bloques en la mayoría de los tipos de bloques y subsistemas. Se puede monitorizar y/o almacenar datos de la señal desde el programa que se está ejecutando en el target, sin tener que escribir ningún código. Además se puede definir las condiciones bajo las cuales los datos son subidos desde el target al host.

El modo externo trabaja estableciendo un canal de comunicación entre Simulink® y el código generado por la Real-Time Workshop®. Este canal se implementa mediante una capa de transporte a bajo nivel que maneja físicamente las transmisiones de los mensajes. Ambos, Simulink® y el código generado son independientes de esta capa.

La siguiente figura muestra la implementación del regulador pasivo PD+G mediante Simulink y la RTW que permite ejecutar el código directamente sobre el hardware, en este caso, se muestran los controladores de las tarjetas, incluyendo las dos PCL-833 realizados, así como la obtención del vector de gravedad necesario para este tipo de regulador, para ello se han utilizado las Embedded Matlab Functions que se describen a continuación.

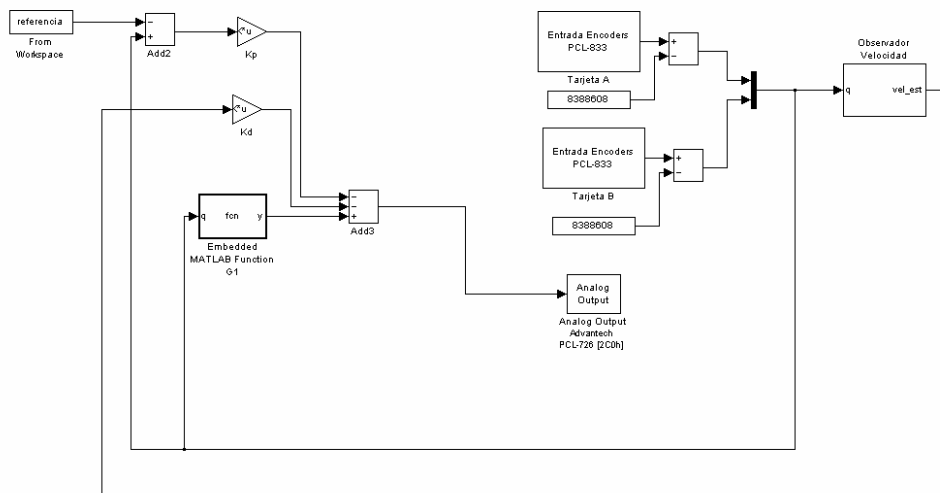


Figura 5-14 Esquema Simulink del Regulador PD+G sobre el Robot Real

5.2.5. Embedded MATLAB Function

Como se ha comentado anteriormente la Embedded MATLAB Function permite añadir funciones de MATLAB[®] para los modelos de Simulink[®]. Esta capacidad es útil para codificar algoritmos que están mejor en el lenguaje textual del software MATLAB que en el lenguaje gráfico de Simulink[®].

La función en Matlab que se crea se ha ejecutado tanto en simulaciones como también ha generado código para la Real-Time Workshop[®] target, por lo que se puede ejecutar en modo externo y controlar directamente el robot.

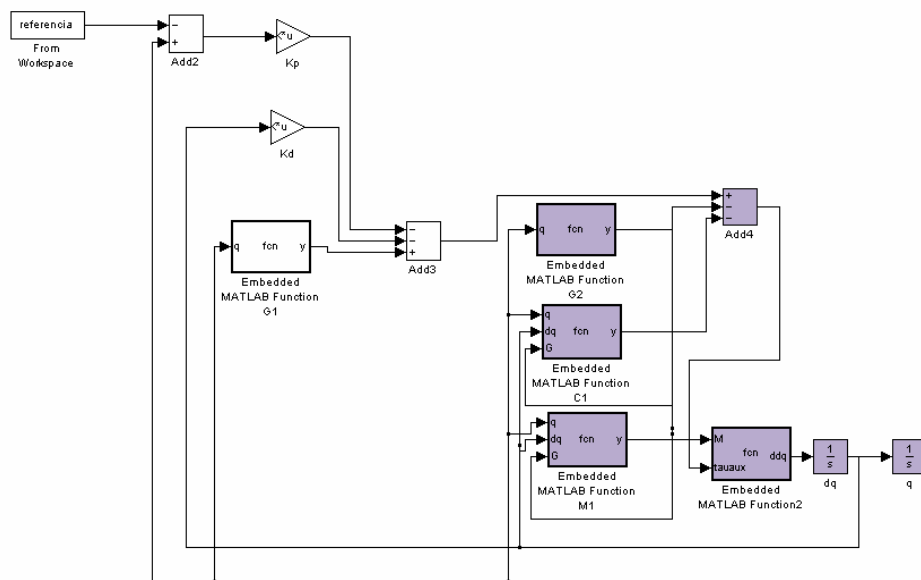
En la presente tesis se han utilizado para resolver tanto el Problema Dinámico Inverso como el Problema Dinámico Directo, por lo que se ha podido tanto simular el comportamiento del PUMA 560 gracias a su modelo dinámico, como realizar como control por dinámica inversa obteniendo los diferentes elementos de la ecuación general del movimiento, ya visto en el capítulo 2.

Este bloque trabaja con un subconjunto del lenguaje MATLAB llamado el subconjunto Embedded MATLAB, que apoya la generación de código eficiente para el despliegue en sistemas.

Además utiliza un motor de inferencia para hacer cumplir las limitaciones del lenguaje para la simulación y generación de código. El motor de inferencia trabaja con el software de generación de código de Real-Time Workshop[®] para convertir de un código dinámicamente tipado (MATLAB) a uno estáticamente tipado (C), sin necesidad de utilizar la asignación dinámica de memoria. Para convertir tipos de datos con precisión, el motor de inferencia de Embedded MATLAB requiere que se defina la clase, el tamaño y la complejidad de los datos en el código fuente para que pueda asignar correctamente los tipos de datos en tiempo de compilación.

La siguiente figura muestra un esquema Simulink que ha permitido simular el comportamiento del PUMA 560, los bloques correspondientes al robot están sombreados,

gracias a las Embedded Functions se ha obtenido la matriz de Inercias, el vector de las fuerzas de Coriolis y vector de la Gravedad, la obtención de estos elementos ya se ha explicado en el Capítulo 2 de la presente tesis, una cuestión muy importante de estos bloques es son reutilizables para cualquier esquema.



5.2.6. Diseño del Algoritmo de Control en tiempo real

A la hora de implementar los reguladores y aplicar el control al robot PUMA 560, se ha trabajado en una arquitectura software con las siguientes características:

- Uso del sistema operativo Windows 2000.
- Microsoft Visual Studio 6.0, el lenguaje de programación que se ha utilizado para la realización del algoritmo de control ha sido el Visual C++.

- Librerías propias del manejo de las tarjetas de adquisición de datos, ya comentadas en la sección anterior.

En la siguiente figura se muestra el Diagrama de Estructura asociado al algoritmo que se ha implementado en la presente tesis.

Este algoritmo se ha implementado en tiempo real, por lo que se han utilizados hilos de control o threads a la hora de programarlo, fijando para ello la máxima prioridad a la hora de ejecutarse.

Un esquema del algoritmo de control utilizado sería el siguiente:

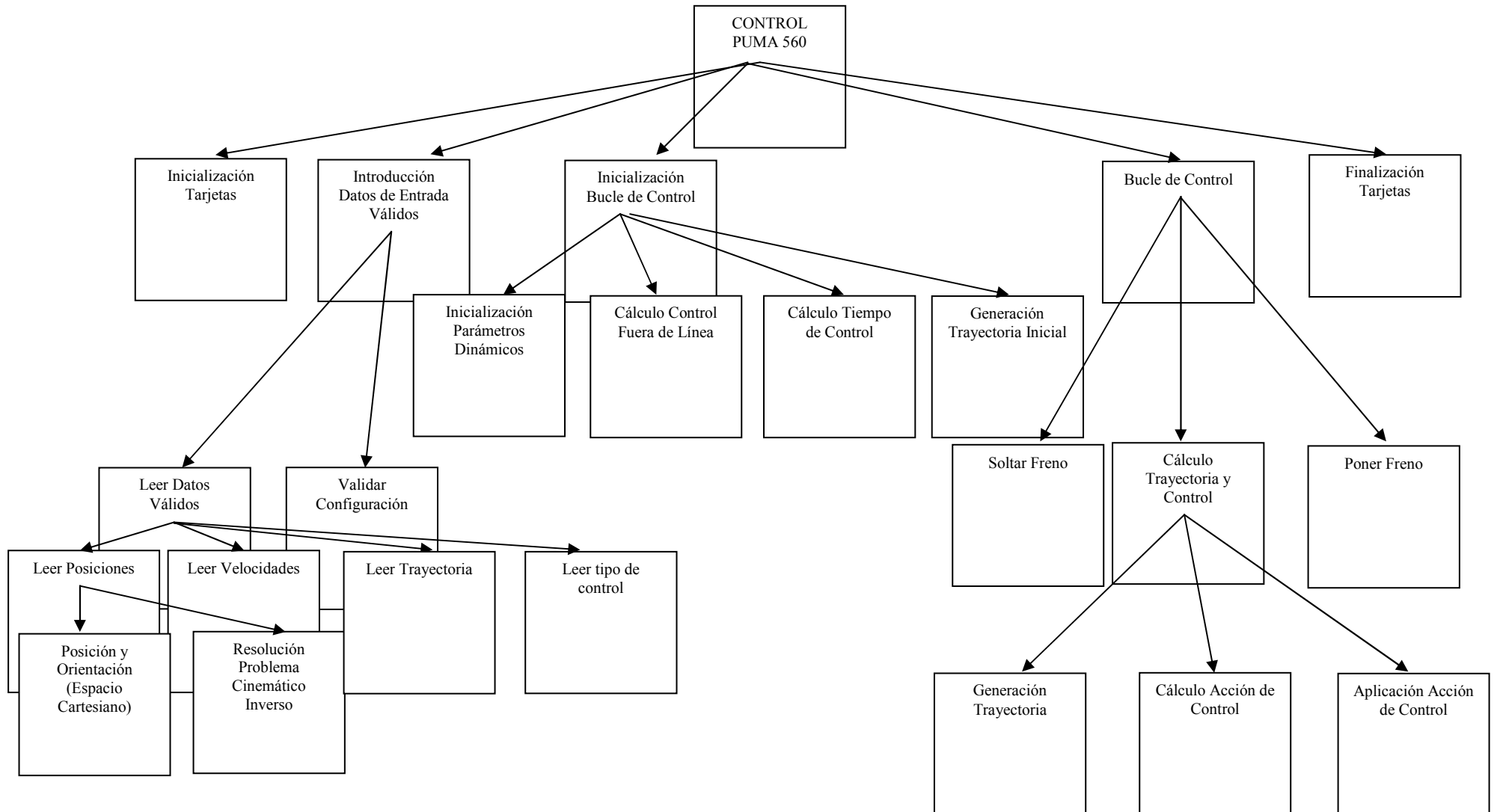
Una vez inicializadas las tarjetas, el usuario selecciona el tipo de trayectoria a realizar, en el caso de estar trabajando con el espacio cartesiano se resolvería el problema cinemático inverso para convertir esos puntos en el espacio de nudos. El usuario indica los puntos iniciales y puntos finales siempre y cuando seleccione trayectorias interpoladas, además también podría indicar un conjunto de puntos intermedios de paso, en el caso de seleccionar una trayectoria de Fourier, el usuario selecciona la localización del fichero con los coeficientes de dichas trayectoria. En ambos casos siempre se indica la velocidad a la que se podrá realizar el movimiento. Tanto las configuraciones con la velocidad siempre son validadas para trabajar dentro del espacio de trabajo del robot y evitar posibles accidentes.

Posteriormente se elegiría el tipo de control a utilizar, ya sea control por dinámica inversa o alguno de los controladores por pasividad implementados. Algunos controladores pueden realizar parte de su cálculo fuera de línea, como el caso de los controladores pasivos que cancelen el término de la gravedad en la posición deseada, pero todos son aplicados en tiempo real dentro del bucle de control.

Una vez se tiene los datos de entrada, el algoritmo se encarga de calcular al tiempo de control, sabiendo como se ha comentado en el capítulo 3 que el tiempo total viene determinado por la articulación más lenta.

En este momento ya puede comenzar el bucle de control, cuyo tiempo va determinado por el cálculo realizado anteriormente con una frecuencia de muestreo establecida o bien de 5ms o bien de 10 ms, normalmente esa frecuencia será de 5 ms. El freno se libera de forma automática gracias al circuito ya comentado en este capítulo y se calcula la trayectoria en tiempo real, además de leer directamente la posición de los encoders de los motores, así como el cálculo del par a suministrar a los motores que se aplica directamente a la etapa de potencia del armario de control del robot.

Una vez finalizado ese bucle se activa el freno y se almacenan los datos en ficheros que posteriormente pueden servir para realizar el estudio de los resultados obtenidos.



CAPÍTULO 6

CONCLUSIONES

En esta tesis, se ha abordado el modelado dinámico de robots y el desarrollo de controladores dinámicos y su implementación en entornos de tiempo real, utilizándose como banco de pruebas un robot industrial: el robot PUMA 560.

En el Capítulo 2 se ha llevado a cabo la descripción y la resolución del modelo cinemático de un robot con configuración de cadena cinemática abierta. La modelización se ha realizado a partir de la notación de Denavit-Hartenberg, implementada bajo la convención de Paul. Se han implementado las expresiones que permiten la obtención de las diversas soluciones del problema cinemático inverso en forma explícita.

Asimismo, se ha desarrollado un modelo dinámico de barras rígidas del brazo robot a partir del formalismo dinámico de Gibbs-Appell, resolviéndose tanto el Problema Dinámico Directo como el Problema Dinámico Inverso. La utilización de dicha formulación proporciona un conjunto de ecuaciones bien estructuradas donde es posible conocer cada uno de los términos dinámicos del robot (inerciales, Coriolis y gravitatorios), y cual es su aportación en la ecuación general del movimiento de éste. Así mismo se ha desarrollado e implementado un algoritmo recursivo de complejidad $O(n)$

para la resolución del Problema Dinámico Inverso, siendo esto sumamente interesante y necesario para el desarrollo de aplicaciones de tiempo real.

Como se sabe, para obtener el modelo dinámico del robot se necesita disponer de los parámetros físicos geométricos y dinámicos del robot. Los primeros no son difíciles de obtener puesto que suelen aparecer directamente en las especificaciones del fabricante de los robots. Sin embargo, no pasa lo mismo con los parámetros dinámicos, puesto que éstos generalmente no se encuentran disponibles y además en la práctica la medición directa de alguno de ellos no es fácil.

Los parámetros dinámicos utilizados en esta tesis se basan en los resultados obtenidos en (Benimeli, 2006) mediante técnicas de identificación de parámetros dinámicos. Sin embargo, hay que tener en cuenta que el conjunto de los parámetros identificados son combinaciones lineales de los parámetros dinámicos, por lo que no se pueden utilizar directamente en las ecuaciones de Gibbs-Appell. Por ello se ha realizado una adaptación basada en la denominada matriz de observación. Gracias de esta matriz de observación se puede resolver tanto el Problema Dinámico Inverso como el Directo, lo que permite obtener directamente los términos de la ecuación general del movimiento a partir del conjunto de parámetros base identificados.

En el Capítulo 3 se describe la implementación del control cinemático. Este control ha permitido establecer las trayectorias que deben seguir cada una de las articulaciones del robot a lo largo del tiempo para lograr los objetivos que se fijen (puntos de destino en el espacio de nudos, trayectorias cartesianas, identificación de los parámetros inerciales etc). Estas trayectorias se han seleccionado atendiendo a las restricciones físicas propias del robot industrial y a ciertos criterios de calidad como precisión o continuidad.

Para resolver el control cinemático se muestrea de la trayectoria cartesiana de movimiento, obteniendo así un número finito de puntos de dicha trayectoria. Posteriormente se resuelve el problema cinemático inverso, haciendo corresponder a cada

punto del espacio cartesiano su correspondiente valor como variable en el espacio de nudos. Luego, se interpolan estos puntos en el espacio de nudos y finalmente se muestrea la trayectoria articular en el espacio de nudos para generar las referencias de movimiento para el control dinámico.

Dependiendo del espacio se han generado distintas trayectorias entre dos puntos (punto inicial y final) directamente o mediante la especificación de un conjunto de puntos intermedios. En el espacio cartesiano se han implementado trayectorias rectilíneas y cúbicas. En el espacio de articulaciones se han implementado, además de las anteriores, trayectorias basadas en series de Fourier, siendo éstas últimas necesarias para poder generar referencias excitables para poder identificar los parámetros físicos del modelo dinámico del robot.

En el Capítulo 4 se describe la implementación del control dinámico realizado. Este control se basa en la resolución del Problema Dinámico Inverso del robot obtenido en el Capítulo 2 de la tesis. Las estrategias implementadas permiten establecer tanto el control punto a punto como el control de trayectoria. El primero únicamente considera el punto inicial y final del movimiento. Los controladores de trayectoria permiten especificar, además de una trayectoria continua, cual es la velocidad y aceleración deseadas para el movimiento.

Se han desarrollado dos tipos de controladores: los controladores por dinámica inversa o por cancelación y los controladores pasivos. Los primeros se basan en una cancelación de los términos dinámicos de la ecuación del robot. Por ello, para poder implementarlos se necesita calcular los términos inerciales, coriolis y gravitacionales, obteniéndose mediante la resolución del Problema Dinámico Inverso. Los controladores por pasividad no lineales implementados en esta tesis requieren realizar una cancelación de los términos gravitacionales, necesitando como en el caso anterior el cálculo de dichos términos mediante la resolución del Problemas Dinámico Inverso.

Para poder simular la respuesta de todos los controladores implementados y realizar la sintonización de las matrices constantes que conforman dichos controladores se han desarrollado un conjunto de simuladores. En éstos se tiene el controlador dinámico a validar y la ecuación dinámica del robot. Para ello se tiene que resolver tanto el Problema Dinámico Directo como el Inverso.

En el Capítulo 5 se describe la arquitectura abierta de control basada en PC utilizada para un robot industrial PUMA 560. Gracias a dicha unidad abierta de control es posible desarrollar e implementar cualquier estrategia de control cinemático y dinámico mediante lenguajes de alto nivel. En esta tesis se ha utilizado el lenguaje Visual C++ para la programación de las distintas tareas de la arquitectura de control. Además, se ha utilizado el entorno de Matlab/Simulink para realizar un prototipado rápido de los distintos controladores implementados gracias a la generación automática de código que proporciona la herramienta Real-Time Workshop.

6.1. APORTACIONES

Las aportaciones realizadas en la presente tesis han sido las siguientes:

- **Aportación 1:** se ha desarrollado un algoritmo para la resolución del PDI basado en las ecuaciones de Gibbs-Appell con una complejidad computacional de $O(n)$. Esto permite la implementación de controladores por dinámica inversa en tiempo real, tanto controladores dinámicos punto a punto como con control de la trayectoria. Así como controladores pasivos cancelando los términos gravitatorios que se obtienen también a partir de dicha resolución.
- **Aportación 2:** Se ha obtenido un algoritmo capaz de obtener cada uno de los elementos de la ecuación general del movimiento, basado en la

resolución del Problema Dinámico Directo, para su posterior utilización en diferentes estrategias de control realizadas en tiempo real.

- **Aportación 3:** Se han desarrollado, implementado y validado un conjunto de controladores no lineales de robots basados en estrategias de control por dinámica inversa y por pasividad
- **Aportación 4:** Se ha desarrollado y utilizado una arquitectura de control simple, económica y abierta para el robot PUMA 560. El control se implementa y aplica a través de un PC conectado a la unidad de potencia del robot. Así se pueden implementar cualquier estrategia de control sobre el sistema, tanto dinámico como cinemático.
- **Aportación 5:** Se ha utilizado herramientas de generación automática de código, la Real-Time Workshop de Matlab, que permite generar código a partir de modelos realizados en Simulink, tanto para la simulación de las estrategias de control sobre el modelo dinámico del robot manipulador, como para la ejecución de las estrategias de control sobre el robot real.
- **Aportación 6:** Se ha implementado en un entorno de programación de alto nivel, concretamente en C++, la resolución tanto del PDI como del PDD que permite implementar cualquier estrategia de control sobre el sistema abierto.

6.2. PUBLICACIONES

Las publicaciones a las que ha dado lugar esta tesis se detallan a continuación:
“Computed-torque Control of Industrial Robot Manipulators using the Gibbs-Appell

Formulation”, Vicente Mata, Ángel Valera, Francisco Valero, Marina Vallés y Nuria Rosillo, 5th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2001), Orlando, Florida (Estados Unidos), Julio de 2001. Las aportaciones realizadas son la 1, 3, 4 y 6.

“Modelado e identificación de parámetros dinámicos de robots. Aplicación a un PUMA 560”, Nuria Rosillo, Francesc Benimeli, Ángel Valera y Vicente Mata, XXVIII Jornadas de automática, Almería, España, Septiembre de 2006. Las aportaciones realizadas son la 1, 3, 4, 5 y 7.

“Solving the Inverse Dynamics Control for Low Cost Real-Time Industrial Robot Control Applications”, Ángel Valera, Vicente Mata, Marina Vallés, Francisco Valero, Nuria Rosillo y Francesc Benimeli, Revista Robotica (Cambridge Journals), nº 21. En este caso las aportaciones realizadas son la 1, 3, 4 y 6.

“Modelling and Identification of Robot Dynamic Parameters. Resolution of the Inverse Dynamic Problem in Real-Time”, Nuria Rosillo, Angel Valera, Francesc Benimeli, Vicente Mata y Francisco Valero, DAAAM International Scientific Book, 2008 capítulo 56. En este caso las aportaciones son la 1, 3, 4, 5 y 6.

6.2.1. Futuras líneas de Investigación

Como continuación lógica del trabajo, se plantea la mejora y extensión del modelo dinámico del robot manipulador, en esta tesis se ha utilizado únicamente el modelado de las barras, si se quiere incrementar la precisión con que pueden describirse el modelo dinámico del robot se podría ampliar las ecuaciones de la dinámica utilizadas en el modelo, para poder realizar la modelización del fenómeno de la fricción, la inclusión de los modelos de los rotores o incluso la inclusión de la elasticidad en las articulaciones, fenómeno éste que cobra una mayor importancia cuando aumentan los requerimientos

sobre prestaciones del robot como la velocidad de ejecución de una trayectoria o el grado de precisión y repetibilidad de un determinado movimiento.

Además, convendría el desarrollo y la implementación de otros tipos de estrategias de control, como por ejemplo controladores adaptativos, estos controladores son capaces de adaptarse por sí mismos a las incertidumbres del proceso de forma que se garantice un funcionamiento adecuado del control. Este tipo de controlador se podría diseñar sobre el sistema ya desarrollado, de esta forma se tendría nuevas estrategias para poder estudiar y comparar con las ya descritas.

No únicamente se podría diseñar e implementar nuevos tipos de controladores para desarrollar el control dinámico sino que también se pueden realizar nuevos tipos de trayectorias que modifique el control cinemático, ya sean en el espacio de nudos como en el espacio cartesiano, por ejemplo trayectorias interpoladas optimizadas, sobre el sistema desarrollado y utilizado en esta tesis.

Por último, hay que mencionar que las estrategias de control desarrolladas en esta tesis se han aplicado a sistemas mecánicos con configuración de cadena cinemática abierta y, en particular, a un robot PUMA 560. Se podría implementar el mismo sistema, es decir, la arquitectura abierta de control basada en PC y la programación en un lenguaje de alto nivel de las tareas asociadas a la arquitectura de control, sobre otros sistemas robotizados tales como robots paralelos o sistemas más actuales.

CAPÍTULO 7

REFERENCIAS BIBLIOGRÁFICAS

- [1] Denavit, J. y Hartenberg, R.S.” A Kinematic Notation for Lower Pair Mechanisms Based On Matrices.” J. Applied Mechanics, V 22, pp 215-221, 1955.
- [2] Craig, J.J. “Introduction to Robotics: Mechanics and Control.” Addison Wesley Publishing Company, Reading, Massachussets, 1989.
- [3] Pieper, D. L. “The Kinematics of Manipulators Unser Computer Control”. Tesis Doctoral, Stanford University, 1968.
- [4] Fu, K.S., Gonzalez, R. C., y Lee, C.S.G, ”Robotics: Control, Sensing, Vision, and Intelligence.”, McGraw-Hill, 1987.
- [5] Luh, J. Y. S., Walker, M. W. y Paul, R. P., “On-line computational scheme for mechanical manipulators,” Journal of Dynamica Systems, Measurement, and Control, Vol. 102, pp. 69-79, 1980.
- [6] Uicker, J. J., “On Dynamic analysis of spatial linkages using 4x4 matrices”, Ph. D. Dissertaton, Northwestern University, August, 1965.
- [7] Hooker, W. y Magulies, G., “The dynamical attitude equations for an n-body satellite, “Journal of Astronautical Science, Vol. 12, No. 4, pp. 123-128, 1965.
- [8] Kahn, M.E., “The near minimum-time control of open articulated kinematics chains”, Ph. D. Dissertaton, Stanford University, 1969.

- [9] Stepanenko, Y. y Vukobratovic, M. "Dynamics of articulated open-chain active mechanisms", *Mathematical Biosciences*, Vol. 28, 1976.
- [10] Orin, D. E., McGhee, R. B., Vukobratovic, M. y Hartoch, G., "Kinematic and Kinetic analysis of open-chain linkages using Newton-Euler methods", *Journal of Dynamic Systems, Measurements, and Control*, Vol. 102, pp. 69-79, 1980.
- [11] Hollerbach, J. M., "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity", *IEEE Transactions on Systems, Man and Cybernetics*, pp.730-736, 1980.
- [12] Renaud, M., "An efficient iterative analytical procedure for obtaining a robot manipulator dynamic model", *Proc. 1st Int. Symp. On Robotic Research*, Bretton Woods, New Hampshire, pp. 749-762, 1983.
- [13] Kane, T. y Levinson, D., "The Use of Kane's Dynamical Equations in Robotics", *Int. Journal of Robotic Research*, Vol. 2, No. 3, pp. 3-21, 1983.
- [14] Angeles, J, Ma, O. y Rojas, A. "An algorithm for Inverse Dynamics of n-Axis General Manipulators Using Kane's Equations". *Computer Math. Applic.*, V 17(12),pp. 1545-15561, 1989.
- [15] Neuman, C. P. and Khosla, P. K. "Identification of Robot Dynamics: An Application of Recursive Estimation.", *Proceedings of the Fourth Yale Workshop on Applications of Adaptive Systems Theory*, pp. 42-49. Yale University, NewHaven, 1985.
- [16] Murray, J.J. y Newman, C.P., "Organizing Customized Robot Dynamics Algorithms for Efficient Numerical Evaluation", *IEEE Trans. On Systems, Man and Cybernetics* 18 (1), pp. 115-125, 1988.
- [17] Balafoutis, C.A. y Patel, R.V. (1991) "Dynamic Analysis of Robot Manipulators: A Cartesian Tensor Approach", *Kluwer Academic Press*, Boston 1991.
- [18] Zomaya, A. Y., "Modelling and Simulation of Robot Manipulators: A Parallel Processing Approach", *World Scientific Publishing Co.*, Singapore, 1992.
- [19] Reanud, M., *Contribution a l'étude de la modélisation et de la commande des systèmes mécaniques articulés. Thèse doctoral. Université Paul Sabatier, Toulouse*, 1975.
- [20] Vukobratovic, M. y Kircanski, N., "Real Time Dynamics of Manipulation Robots", *Springer-Verlag*, Berlin, 1985.

- [21] Desoyer, K. y Lugner, P., "Recursive Formulation for the Analytical or Numerical Application of the Gibbs-Appell Method to the Dynamics of Robots", *Robotica*, Vol 7, pp. 343-347, 1989.
- [22] Vereshchagin, A. F., "Computer simulation of the dynamics of complicated mechanisms of robotic manipulator", *Engineering Cybernetics*, Vol. 5, pp. 65-70, 1974.
- [23] Armstrong, W. W., "Recursive solution to the equations of motion of an n-link manipulator," Fifth World Congress on Theory of Machines and Mechanisms, Montreal, 1979.
- [24] Walker, M. W. y Orin, D. E., "Efficient Dynamic Computer Simulation of Robotic Mechanisms", *Journal of Dynamics Systems, Measurement and Control*, Vol. 104, pp. 205-211, 1982.
- [25] Provenzano, S., Aplicación de las ecuaciones de Gibbs-Appell a la Dinámica de Robots. Departamento de Ingeniería Mecánica, Universidad Politécnica de Valencia, Valencia, 2001.
- [26] Besa, A. J., "Identificación de parámetros cinemáticos de un robot industrial mediante un sistema de rastreo láser", Tesis Doctoral, 2002
- [27] P. I. Corke y B. Armstrong-Hélouvy, "A Meta-Study of Puma 560 Dynamics: A Critical Appraisal of Literature Data," *ROBOTICA (Journal of the International Federation of Robotics)*, vol. 13, pp. 253-258, 1995.
- [28] Corke, P. I. y Armstrong-Hélouvy, B., "A Search for Consensus among Model Parameters Reported for the Puma-560 Robot," presented at IEEE Proceedings on the International Conference on Robotics and Automation, 1994.
- [29] Benimeli, F. "Estimación de Parámetros Dinámicos en Robots Manipuladores", Tesis doctoral, 2006.
- [30] Spong, M.W., y Vidyasagar, M. 1989. "Robot Dynamics and Control." John Wiley and Sons.
- [31] Torfs, D., De Schutter, J., and Swevers, J., 1992, "Extended Bandwidth Zero Phase Error Tracking Control of Nonminimum Phase Systems", *Journal of Dynamic Systems, Measurement, and Control*, Vol 114, pp 347-351.
- [32] Lawson, C.L., Hanson, R.J., 1974. "Solving Least Squares Problems." Prentice-Hall, Englewood Cliffs, NJ.

- [33] B. Armstrong, "On Finding Exciting Trajectories for Identification Experiments Involving Systems with Nonlinear Dynamics," *International Journal of Robotics Research*, vol. 8, pp. 28-48, 1989
- [34] M. Gautier, "Numerical-Calculation of the Base Inertial Parameters of Robots," *Journal of Robotic Systems*, vol. 8, pp. 485-506, 1991.
- [35] K. Otani y T. Kakizaki, "Motion Planning and Modeling for Accurately Identifying Dynamic Parameters of an Industrial Robotic Manipulator," presented at Proceedings of 24th ISIR, 1993.
- [36] Gautier, M., Janin, C., y Presse, C., "Dynamic identification of robots using least squares and extended Kalman filtering methods." Proceedings of ECC'93, pp. 2291-2297, 1993.
- [37] Swevers, J., Ganseman, C., Tukul, D. B., DeShutter, J. y VanBrussel, H., "Optimal Robot Excitation and Identification," *IEEE Transactions on Robotics and Automation*, vol. 13, pp. 730-740, 1997.
- [38] Swevers, J., Ganseman, C., DeShutter, J. y VanBrussel, H., "Generation of Periodics Trajectories for Optimal Robot Excitation," *Journal of Manufacturing Science and Engineering-Transactions of the Asme*, vol. 119, pp.611-615, 1997.
- [39] Grimm, W. M., "Robot non-linearity bounds evaluation techniques for robust control," *Int. J. of Adaptative Control and Signal Processing*, vol. 4, pp. 501-504, 1990.
- [40] Markiewicz, B. R. "Analysis of the Computed-Torque Drive Method and Comparision with the Conventional Position Servo for a Computer-Controlled Manipulator". Technical Memorandum 33-601, Jet Propulsion Laboratory, Pasadena, CA, March, 1973.
- [41] Aubin A., Canudas de Wit C. y Sidaoui H, "Dynamic Model simplification of industrial manipulators", Prepr. 3d IFAC Symp. On Robot Control, Viena, pp. 9-14, 1991.
- [42] Takegaki, M., Arimoto S., "A New Feedback Method for Dynamic Control of Manipulators" *ASME J. of Dynamic Systems, Measurement, and Ctrl*, Vol.102, pp. 119-125, 1981
- [43] Arimoto, S. y Miyazaki F., Stability and robustness of PID feedback control of robots manipulators, in *Robotic Research 1st Int. Symp.*, M. Brady y R.P. Paul (Eds.), MIT Press, Cambridge, MA, pp. 793-789, 1983.

- [44] Kelly R., "A Simple Set-Point Robot Controller by Using Only Position Measurements," Proc. 12th IFAC World Congr., 6, 173-176, 1993.
- [45] Berghuis, H. y Nijmeijer, H., "Global Regulation of Robots Using Only Position Measurements", Report n° 92R225, Control Lab. Electrical Engineering Dept., University of Twente, Holanda, 1992
- [46] Wen, J. T., Murphy, S., "PID Control for Robots Manipulators", CIRSSSE Document 54, Reusselaer Polytechnic Institute, 1990
- [47] Ortega, R, Loria, A. y Nelly, R., "A Semiglobally Stable Output Feedback PI²D Regulator for Robot Manipulators", Report interno. Heuristique et Diagnostic des Systemes Complexes, Ura cnrs 817, Univertié de Technologie de Compiègne, 1991.
- [48] Moreira, N., Alvito, P., y Lima, P., "First Steps Towards an Open Control Architecture for a Puma 560," presented at Proceedings of the 2nd Portuguese Conference on Automatic Control, Porto, Portugal, 1996.
- [49] Valera, A., Vergara, J., Tornero, J. y Garcia, E., "Control a Puma 500 Using a New Open Architecture," presented at 3th Portuguese Conference On Automatic Control, University of Coimbra, Portugal, 1998
- [50] Valera, A., Mata, V., Vallés, M., Valero, F., Rosillo, N. y Benimeli, F., "Solving the inverse dynamic control for low cost real-time industrial robot control applications", Robotica (Cambridge University Press), Vol 21, pp 261-269, 2003.
- [51] Valera, A., "Análisis Comparativo de Técnicas de Control de Robots Rígidos y Flexibles". Tesis Doctoral, 2000.

ANEXO A

ESTRATEGIAS DE CONTROL NO LINEAL DE ROBOTS

En esta sección se va a mostrar la demostración de la estabilidad de los controladores implementados en esta tesis, así como, resultados reales obtenidos del robot PUMA 560, utilizando para ello la arquitectura abierta de control comentada en el capítulo 5, programada directamente utilizando un lenguaje de alto de nivel, en concreto el Visual C++ y siguiendo el esquema ya comentado en ese mismo capítulo.

Estas demostraciones y los resultados reales obtenidos son de los dos tipos de controladores implementados, estos controladores están basados en el modelo del robot, es decir, tienen en cuenta la descripción no lineal del sistema robotizado., es decir, controladores por dinámica inversa o par calculado que como ya se ha comentado alcanzan el objetivo de seguimiento de trayectoria mediante la linealización por realimentación de la dinámica no lineal del robot, mientras que los controladores por

pasividad modifican la energía mecánica del sistema con el fin de alcanzar el objetivo de control.

Estos algoritmos se basan en el PDI usando la formulación de Gibbs-Appell con un coste computacional lineal por lo que están implementados en tiempo real.

A.1. CONTROL POR DINÁMICA INVERSA

A.1.1. Control Punto a Punto

Como se ha visto en capítulos anteriores la ecuación del controlador por dinámica inversa punto a punto es la siguiente:

$$\tau_c = M(q)(K_d \dot{q} - K_p e) + C(q, \dot{q})\dot{q} + G(q) \quad (\text{A.1})$$

La ecuación del sistema robot/controlador en bucle cerrado se puede escribir como:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau_c = M(q)(-K_d \dot{q} - K_p e) + C(q, \dot{q})\dot{q} + G(q) \quad (\text{A.2})$$

Si se cancelan los términos de Coriolis y los gravitatorios se obtiene:

$$M(q)\ddot{q} = M(q)(-K_d \dot{q} - K_p e) \quad (\text{A.3})$$

Puesto que $M(q)$ es simétrica y definida positiva tenemos

$$\ddot{q} = -K_d \dot{q} - K_p e = -K_d \dot{q} - K_p (q - q_d) \quad (\text{A.4})$$

Agrupando los términos nos queda:

$$\ddot{q} + K_d \dot{q} + K_p q = K_p q_d \quad (\text{A.5})$$

A partir de la transformada de Laplace se obtiene:

$$s^2 q(s) + K_d s q(s) + K_p q(s) = K_p q_d(s) \quad (\text{A.6})$$

Se puede calcular la función de transferencia en bucle cerrado como sigue:

$$\frac{q(s)}{q_d(s)} = \frac{K_p}{s^2 + K_d s + K_p} \quad (\text{A.7})$$

El denominador de dicha función de transferencia se corresponde con la ecuación característica del sistema. Para que ésta sea estable, bastará con que se cumpla que $K_d > 0$ y $K_p > 0$.

A.1.1.1. Resultados reales de las ejecuciones para una trayectoria de Fourier

Tabla A-1 Comparación de Referencia y Posición para las 6 articulaciones del controlador por dinámica inversa punto a punto
(abcisas: posición (rad); ordenadas: períodos de muestreo ($T_s=0.005s$))

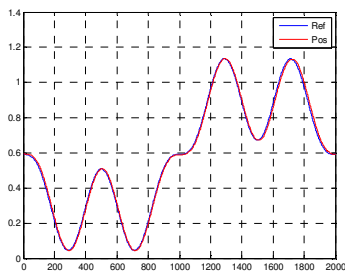


Figura A-1 Articulación 1

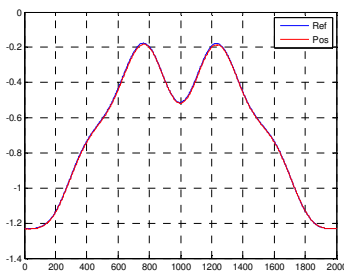


Figura A-2 Articulación 2

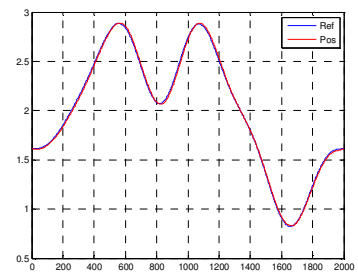


Figura A-3 Articulación 3

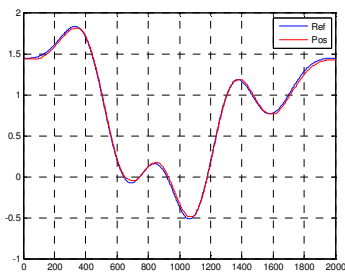


Figura A-4 Articulación 4

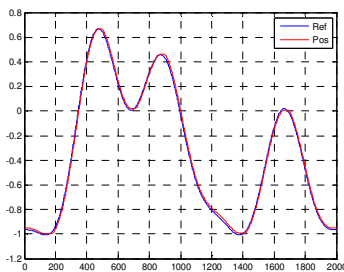


Figura A-5 Articulación 5

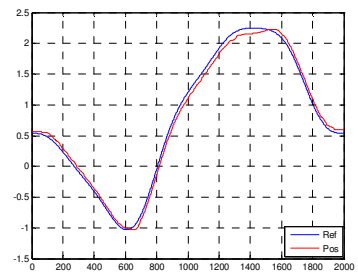


Figura A-6 Articulación 6

Tabla A-2 Acciones de Control del controlador por dinámica inversa punto a punto
(abcisas: acción de control (Nm); ordenadas: períodos de muestreo ($T_s=0.005s$))

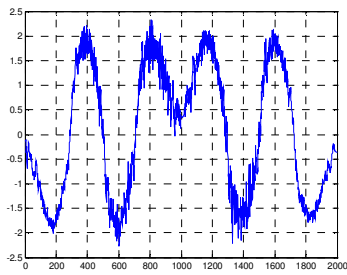


Figura A-7 Articulación 1

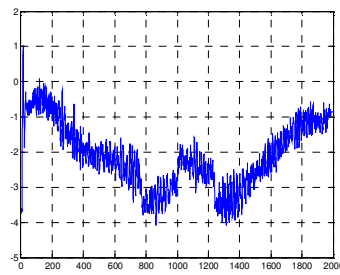


Figura A-8 Articulación 2

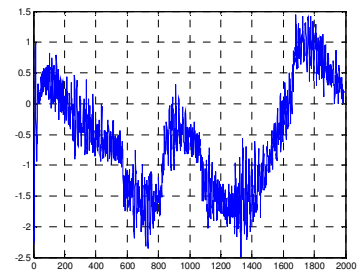


Figura A-9 Articulación 3

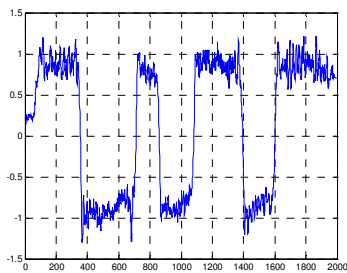


Figura A-10 Articulación 4

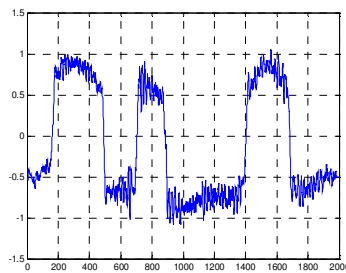


Figura A-11 Articulación 5

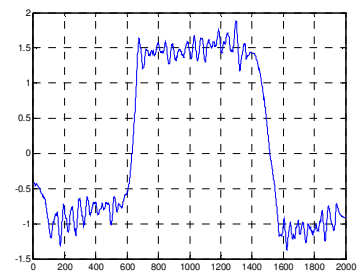


Figura A-12 Articulación 6

A.1.2. Seguimiento de la Trayectoria

El controlador por Dinámica Inversa con seguimiento de la trayectoria es el siguiente:

$$\tau_c = M(q)(\ddot{q}_d - K_d \dot{e} - K_p e) + C(q, \dot{q})\dot{q} + G(q) \quad (\text{A.8})$$

A partir de la ecuación del sistema robot/controlador en bucle cerrado y cancelando los términos gravitatorios y de Coriolis se obtiene la siguiente expresión:

$$\ddot{q} = \ddot{q}_d - K_d \dot{e} - K_p e \quad (\text{A.9})$$

Agrupando términos se obtiene:

$$\ddot{e} + K_d \dot{e} + K_p e = 0 \quad (\text{A.10})$$

siendo $\ddot{e} = \ddot{q} - \ddot{q}_d$

Para demostrar la estabilidad del sistema se puede hacer a partir de la representación en el espacio de estados:

$$\begin{bmatrix} \dot{e} \\ \ddot{e} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix}}_A \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \quad (\text{A.11})$$

Los valores propios de A se corresponden con la ecuación característica del sistema en bucle cerrado.

$$|sI - A| = 0 \quad (\text{A.12})$$

$$\left\| \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix} \right\| = 0 \quad (\text{A.13})$$

$$\begin{vmatrix} s & -I \\ K_p & s + K_d \end{vmatrix} = 0 \quad (\text{A.14})$$

$$s^2 + K_d s + K_p = 0 \quad (\text{A.15})$$

Como en el caso anterior, para que el sistema sea estable bastará con que $K_d > 0$ y $K_p > 0$.

A.1.2.1. Resultados reales de las ejecuciones para una trayectoria de Fourier

Tabla A-3 Comparación de Referencia y Posición para las 6 articulaciones del controlador por dinámica inversa con seguimiento de la trayectoria (abcisas: posición (rad); ordenadas: períodos de muestreo ($T_s=0.005s$))

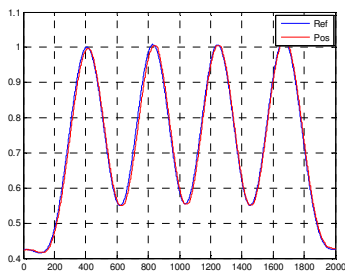


Figura A-13 Articulación 1

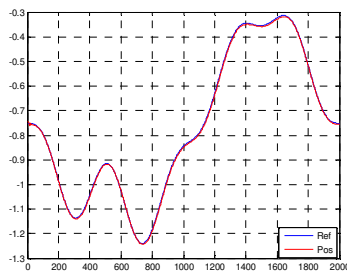


Figura A-14 Articulación 2

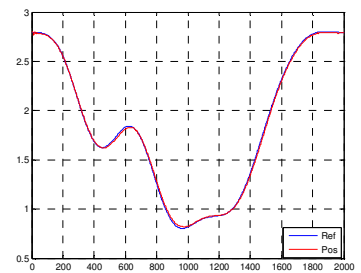


Figura A-15 Articulación 3

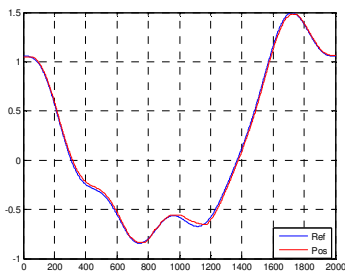


Figura A-16 Articulación 4

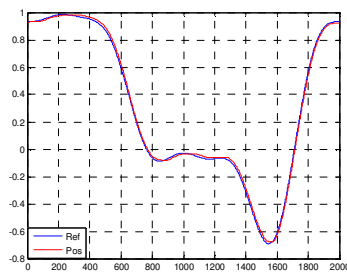


Figura A-17 Articulación 5

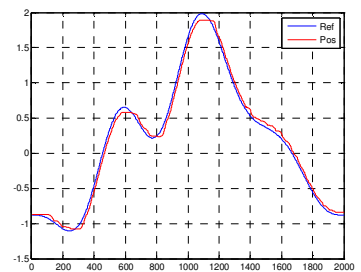


Figura A-18 Articulación 6

Tabla A-4 Acciones de Control del controlador por dinámica inversa con seguimiento de la trayectoria
(abcisas: acción de control (Nm); ordenadas: períodos de muestreo ($T_s=0.005s$))

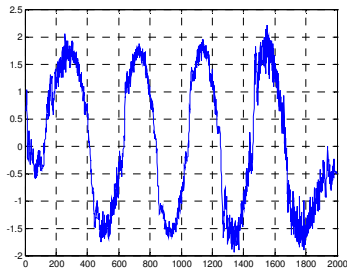


Figura A-19 Articulación 1

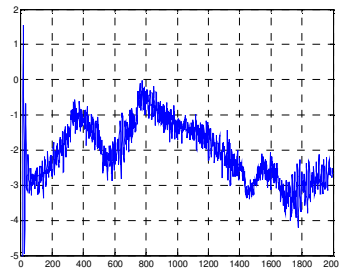


Figura A-20 Articulación 2

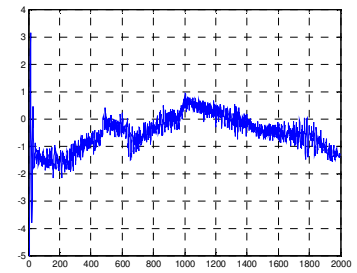


Figura A-21 Articulación 3

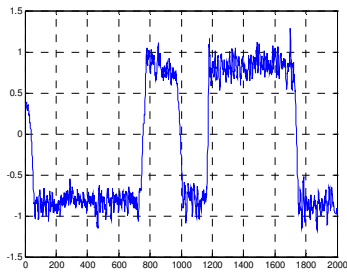


Figura A-22 Articulación 4

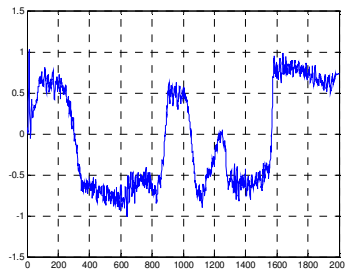


Figura A-23 Articulación 5

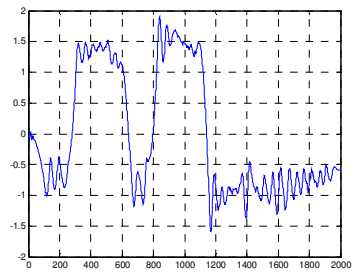


Figura A-24 Articulación 6

A.1.3. Seguimiento de la Trayectoria con Acción Integral

El controlador por dinámica inversa con seguimiento de la trayectoria con acción integral es el siguiente:

$$\tau_c = M(q)(\ddot{q}_d - K_d\dot{e} - K_p e - K_i \int e dt) + C(q, \dot{q})\dot{q} + G(q) \quad (\text{A.16})$$

A partir de la ecuación del sistema robot/controlador en bucle cerrado, cancelando los términos gravitatorios y de Coriolis y sabiendo que la matriz de Inercias es definida positiva se obtiene la siguiente expresión:

$$\ddot{q} = \ddot{q}_d - K_d\dot{e} - K_p e - K_i \int e dt = 0 \quad (\text{A.17})$$

Agrupando términos se obtiene:

$$\ddot{e} + K_d\dot{e} + K_p e + K_i \int e dt = 0 \quad (\text{A.18})$$

A partir de la transformada de Laplace podemos obtener la siguiente expresión:

$$\ddot{e} + K_d\dot{e} + K_p e + K_i e = 0 \quad (\text{A.19})$$

Obteniendo la representación en el espacio de estados:

$$\begin{bmatrix} \dot{e} \\ \ddot{e} \\ \ddot{e} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ -K_i & -K_p & -K_d \end{bmatrix}}_A \begin{bmatrix} e \\ \dot{e} \\ \ddot{e} \end{bmatrix} \quad (\text{A.20})$$

Calculando los valores propios de la matriz de estados del sistema:

$$|sI - A| = 0 \quad (\text{A.21})$$

$$\left[\begin{array}{ccc} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{array} \right] - \left[\begin{array}{ccc} 0 & I & 0 \\ 0 & 0 & I \\ -K_i & -K_p & -K_d \end{array} \right] = 0 \quad (\text{A.22})$$

$$\left| \begin{array}{ccc} s & -I & 0 \\ 0 & s & -I \\ K_i & K_p & s + K_d \end{array} \right| = 0 \quad (\text{A.23})$$

Resolviendo el determinante

$$s^2(s + K_d) + K_i - (-K_p Is) \quad (\text{A.24})$$

Agrupando términos se puede obtener la ecuación característica del sistema:

$$s^3 + K_d s^2 + K_p s + K_i = 0 \quad (\text{A.25})$$

Como se ha comentado en los casos anteriores, el sistema será estable si $K_d > 0$, $K_p > 0$ y $K_i > 0$.

A.1.3.1. Resultados reales de las ejecuciones para una trayectoria de Fourier

Tabla A-5 Comparación de Referencia y Posición para las 6 articulaciones del controlador por dinámica inversa con seguimiento de la trayectoria con acción integral (abcisas: posición (rad); ordenadas: períodos de muestreo ($T_s=0.005s$))

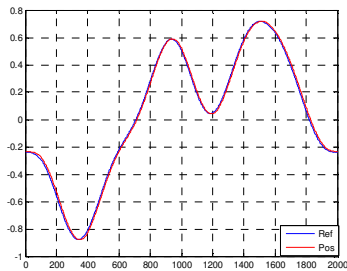


Figura A-25 Articulación 1

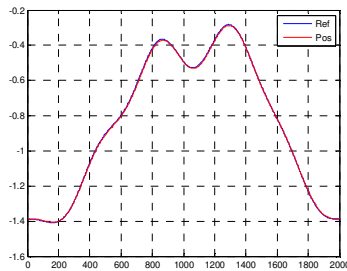


Figura A-26 Articulación 2

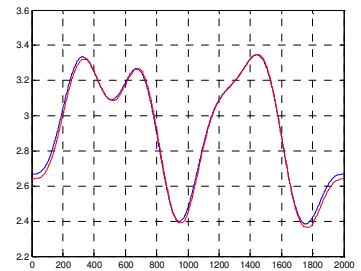


Figura A-27 Articulación 3

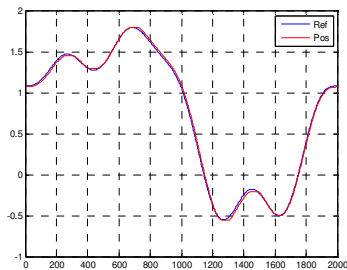


Figura A-28 Articulación 4

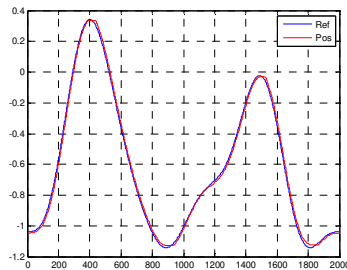


Figura A-29 Articulación 5

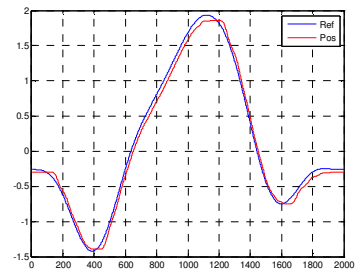


Figura A-30 Articulación 6

**Tabla A-6 Acciones de Control del controlador por dinámica inversa con seguimiento de la trayectoria
con acción integral
(abcisas: acción de control (Nm); ordenadas: períodos de muestreo ($T_s=0.005s$))**

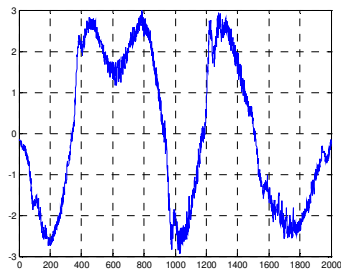


Figura A-31 Articulación 1

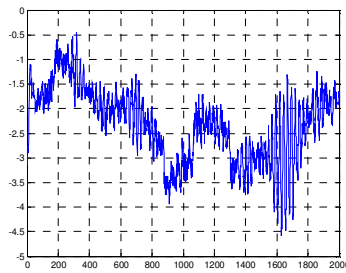


Figura A-32 Articulación 2

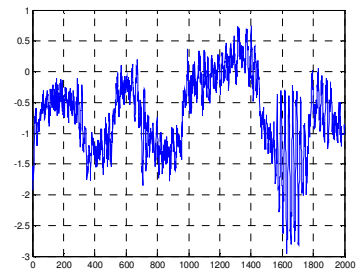


Figura A-33 Articulación 3

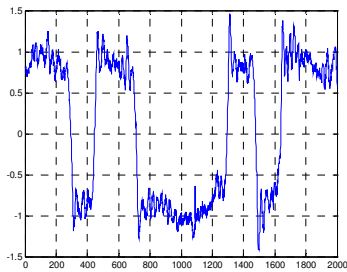


Figura A-34 Articulación 4

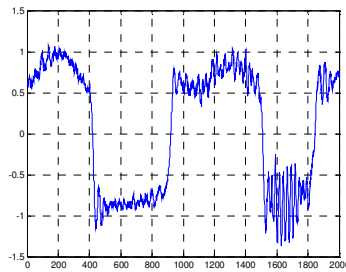


Figura A-35 Articulación 5

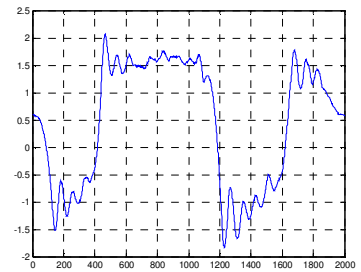


Figura A-36 Articulación 6

A.1.3.2. Resultados reales de las ejecuciones para una trayectoria en el espacio cartesiano del Robot con curvas polinomiales de tercer orden con interpolación en puntos intermedios

Tabla A-7 Comparación de Referencia y Posición para las 6 articulaciones del controlador por dinámica inversa con seguimiento de la trayectoria con acción integral (abscisas: posición (rad); ordenadas: períodos de muestreo (Ts=0.005s))

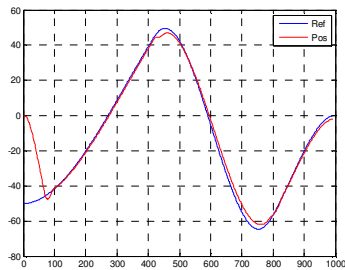


Figura A-37 Articulación 1

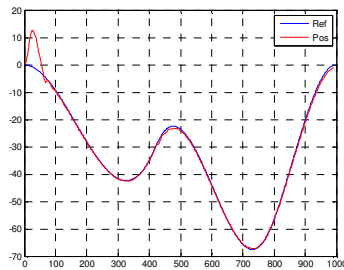


Figura A-38 Articulación 2

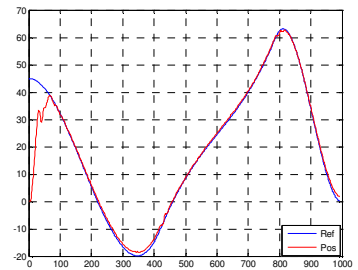


Figura A-39 Articulación 3

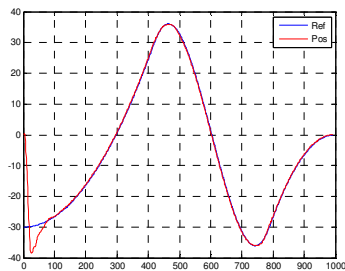


Figura A-40 Articulación 4

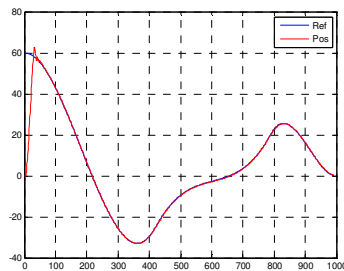


Figura A-41 Articulación 5

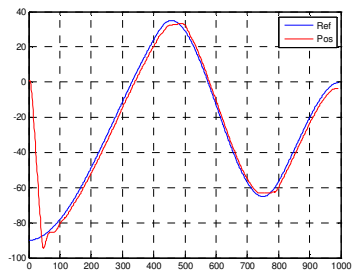


Figura A-42 Articulación 6

**Tabla A-8 Acciones de Control del controlador por dinámica inversa con seguimiento de la trayectoria
con acción integral
(abcisas: acción de control (Nm); ordenadas: períodos de muestreo ($T_s=0.005s$))**

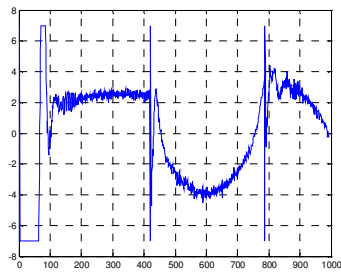


Figura A-43 Articulación 1

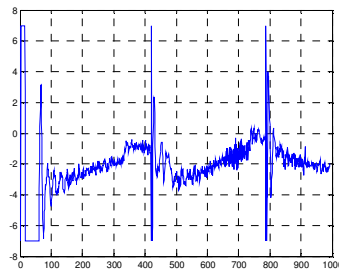


Figura A-44 Articulación 2

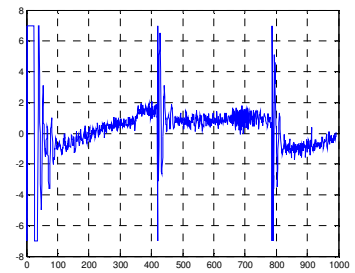


Figura A-45 Articulación 3

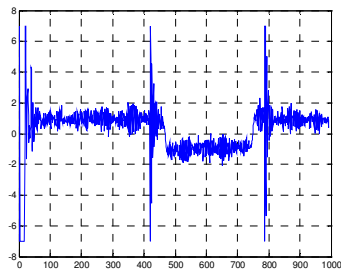


Figura A-46 Articulación 4

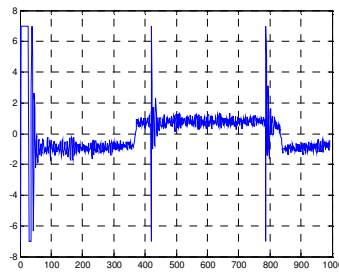


Figura A-47 Articulación 5

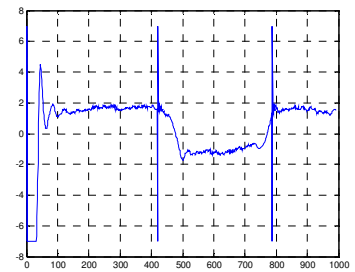


Figura A-48 Articulación 6

A.1.3.3. Resultados reales de las ejecuciones para una curva de Hermite sin puntos intermedios

Tabla A-9 Comparación de Referencia y Posición para las 6 articulaciones del controlador por dinámica inversa con seguimiento de la trayectoria con acción integral (abcisas: posición (rad); ordenadas: periodos de muestreo ($T_s=0.005s$))

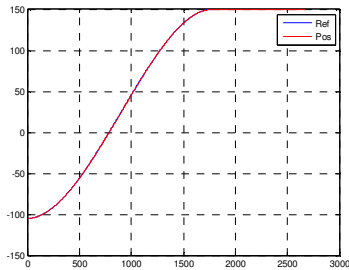


Figura A-49 Articulación 1

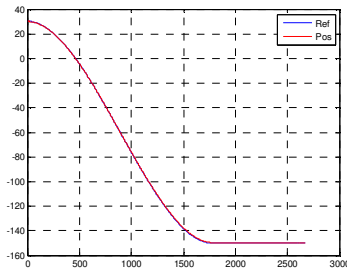


Figura A-50 Articulación 2

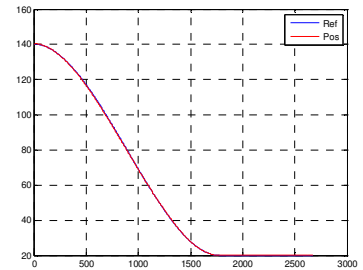


Figura A-51 Articulación 3

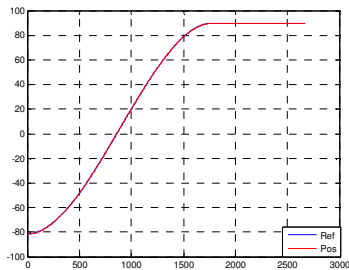


Figura A-52 Articulación 4

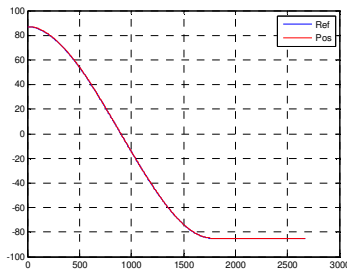


Figura A-53 Articulación 5

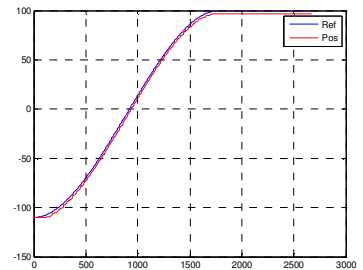


Figura A-54 Articulación 6

Tabla A-10 Acciones de Control del controlador por dinámica inversa con seguimiento de la trayectoria con acción integral
(abcisas: acción de control (Nm); ordenadas: períodos de muestreo ($T_s=0.005s$))

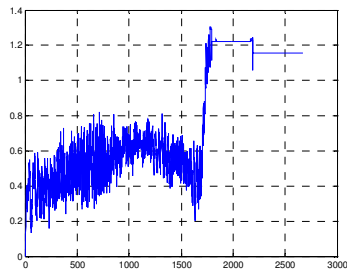


Figura A-55 Articulación 1

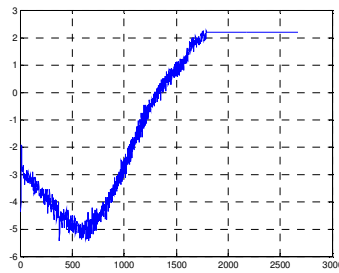


Figura A-56 Articulación 2

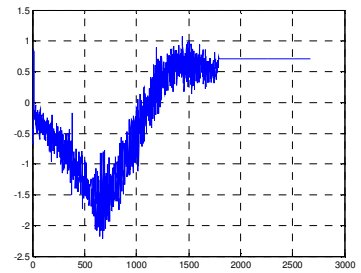


Figura A-57 Articulación 3

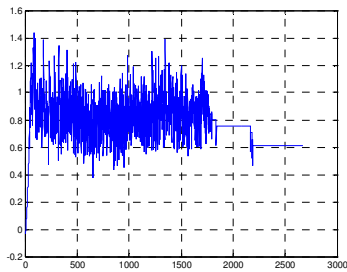


Figura A-58 Articulación 4

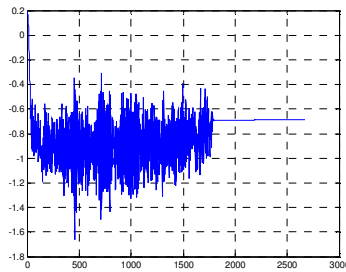


Figura A-59 Articulación 5

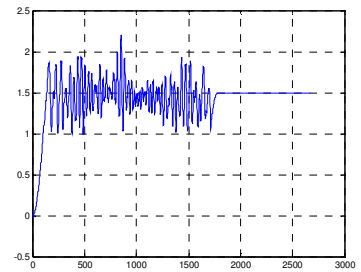


Figura A-60 Articulación 6

A.2. CONTROL POR PASIVIDAD

Como ya se ha comentado en el capítulo 4 de la tesis, los reguladores basados en la pasividad con estrategias punto a punto se pueden ver como casos particulares de la siguiente ley de control general:

$$\tau_e = -K_p e - K_d v - u \quad (\text{A.26})$$

donde:

$$e = q - q_d$$

u, v : varían según el tipo de controlador, tal como se mostrará posteriormente

En este apartado se va a presentar un conjunto de estrategias de control punto a punto basadas en técnicas de pasividad, estableciéndose para cada una de ellas la demostración de la estabilidad del sistema.

A.2.1. Controlador PD con compensación de la gravedad

El controlador propuesto por (Takegaki, Arimoto, 81) tiene la siguiente ley de control:

$$\tau_e = -K_p e - K_d \dot{q} + G(q, \theta) \quad (\text{A.27})$$

donde:

K_p y K_d : matrices diagonales definidas positivas

Teorema A.1.-

El controlador de posición (A A.2) estabiliza global y asintóticamente el sistema robot en el punto de equilibrio (e, \dot{q}) .

Demostración.-

A partir de la ecuación de la dinámica del robot y de la ecuación del controlador tenemos que el error en bucle cerrado es:

$$M(q, \theta)\ddot{q} + C(q, \dot{q}, \theta)\dot{q} + K_d\dot{q} + K_p e = 0 \quad (\text{A.28})$$

Consideramos la función modificada de la energía

$$H_0(\dot{q}, e) = \frac{1}{2} \dot{q}^T M(q, \theta) \dot{q} + \frac{1}{2} e^T K_p e \quad (\text{A.29})$$

como función de Lyapunov. La derivada de (A.4) respecto del tiempo viene dada por

$$\dot{H}_0(\dot{q}, e) = \dot{q}^T M(q, \theta) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{M}(q, \theta) \dot{q} + \dot{e}^T K_p e \quad (\text{A.30})$$

Combinando con la ecuación (A.28) queda

$$\dot{H}_0(\dot{q}, e) = \dot{q}^T (-C(q, \dot{q}, \theta)\dot{q} - K_p e - K_d \dot{q}) + \frac{1}{2} \dot{q}^T \dot{M}(q, \theta) \dot{q} + \dot{e}^T K_p e \quad (\text{A.31})$$

31)

Teniendo en cuenta la hemisimetría de $\dot{M}(q, \theta) - 2C(q, \dot{q}, \theta)$ y que $\dot{q} = \dot{e}$, agrupando términos nos queda:

$$\dot{H}_0(\dot{q}, e) = -\dot{q}^T K_d \dot{q} \quad (\text{A.32})$$

la cual es semidefinida negativa. Sin embargo, se puede verificar que el mayor conjunto invariante con $\dot{H}_0(\dot{q}, e) = 0$ es el estado de equilibrio $(e, \dot{q}) = (0, 0)$. Por lo tanto podemos probar la estabilidad asintótica en el equilibrio aplicando el principio de invarianza de LaSalle.

A.2.1.1. Resultados reales de las ejecuciones para una curva de Hermite sin puntos intermedios

**Tabla A-11 Comparación de Referencia y Posición para las 6 articulaciones del controlador PD+G(q)
(abcisas: posición (rad); ordenadas: períodos de muestreo (Ts=0.005s))**

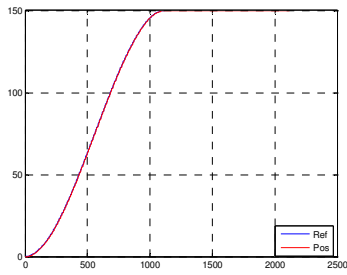


Figura A-61 Articulación 1

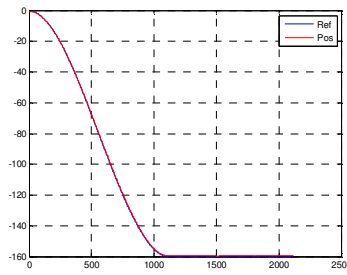


Figura A-62 Articulación 2

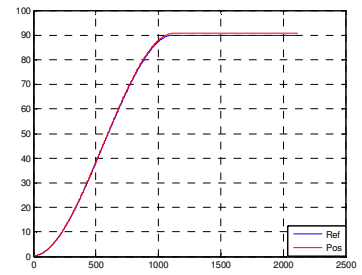


Figura A-63 Articulación 3

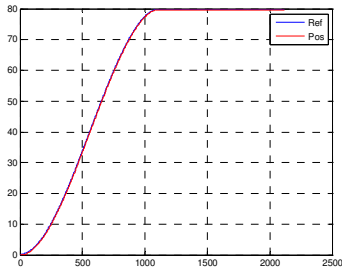


Figura A-64 Articulación 4

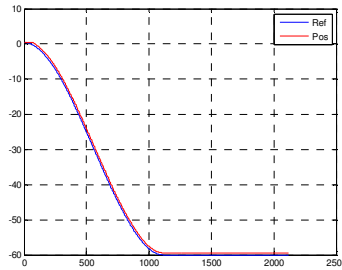


Figura A-65 Articulación 5

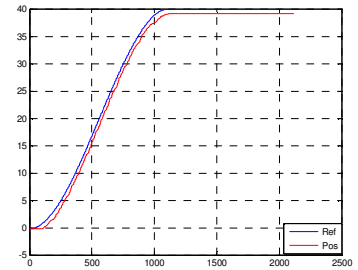


Figura A-66 Articulación 6

Tabla A-12 Acciones de Control del controlador PD+G(q)
(abcisas: acción de control (Nm); ordenadas: periodos de muestreo ($T_s=0.005s$))

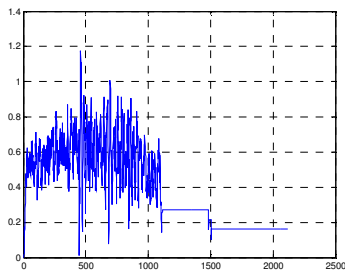


Figura A-67 Articulación 1

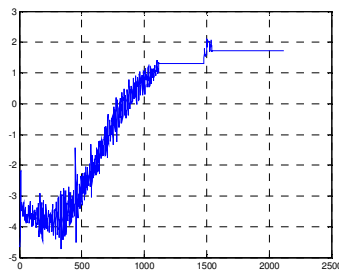


Figura A-68 Articulación 2

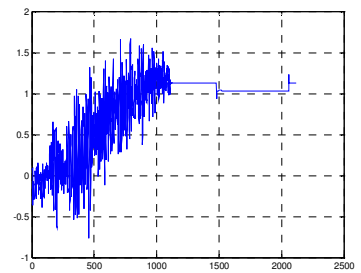


Figura A-69 Articulación 3

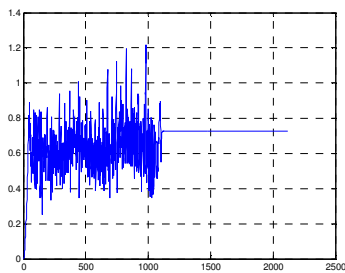


Figura A-70 Articulación 4

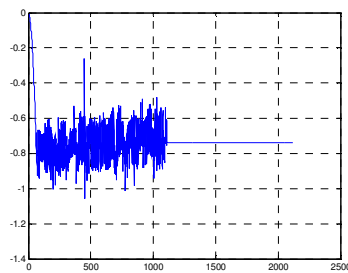


Figura A-71 Articulación 5

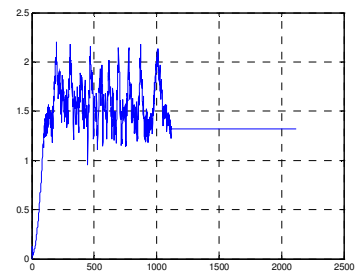


Figura A-72 Articulación 6

A.2.2. PD con compensación de la gravedad en la posición deseada

La ley de control también fue planteada en (Takegaki, Arimoto, 81), y es

$$\tau_e = -K_p e - K_d \dot{q} + G(q_d, \theta) \quad (\text{A.33})$$

Teorema A.2.-

El regulador (A.33) estabiliza global y asintóticamente el sistema robot en el punto de equilibrio $(e, \dot{q}) = (0, 0)$ bajo la condición

$$K_{p,m} > k_g \quad (\text{A.34})$$

donde:

$K_{p,m}$ mayor valor propio de K_p

$$\left\| \frac{\partial G(q, \theta)}{\partial q} \right\| \leq k_g \quad (\text{A.35})$$

Demostración.-

El error en bucle cerrado vendrá dado por la ecuación:

$$M(q, \theta) \ddot{q} + C(q, \dot{q}, \theta) \dot{q} + G(q, \theta) - G(q_d, \theta) + K_p e + K_d \dot{q} = 0 \quad (\text{A.36})$$

La función de la energía del sistema que vamos a utilizar como candidata de Lyapunov es:

$$H_2(e, \dot{q}) = \frac{1}{2} \dot{q}^T M(q, \theta) \dot{q} + V(q, q_d) - V(q_d, q_d) \quad (\text{A.37})$$

donde

$$V(q, q_d) = P(q, \theta) - q^T G(q_d, \theta) + \frac{1}{2} q^T K_p q - q^T K_p q_d \quad (\text{A.38})$$

Para poder aplicar el teorema de LaSalle la función $H_2(\cdot)$ debe ser definida positiva. El primer término de la parte derecha lo es, y la función $V(q, q_d)$ tiene un mínimo absoluto en q_d bajo la condición (A.34). De esta forma podemos calcular la derivada:

$$\dot{H}_2(e, \dot{q}) = \dot{q}^T M(q, \theta) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{M}(q, \theta) \dot{q} + \dot{V}(q, q_d) - \dot{V}(q_d, q_d) \quad (\text{A.39})$$

A partir de la ecuación del error y derivando la función V se obtiene:

$$\begin{aligned} \dot{H}_2(e, \dot{q}) = & \dot{q}^T (-C(q, \dot{q}, \theta) \dot{q} - G(q, \theta) + G(q_d, \theta) - K_p e - K_d \dot{q}) + \frac{1}{2} \dot{q}^T \dot{M}(q, \theta) \dot{q} + \\ & + \dot{P}(q, \theta) - \dot{q}^T G(q_d, \theta) + \dot{q}^T K_p q - \dot{q}^T K_p q_d \end{aligned} \quad (\text{A.40})$$

Agrupando términos, teniendo en cuenta la hemisimetría de $\dot{M}(q, \theta) - 2C(q, \dot{q}, \theta)$ y que $\dot{P}(q, \theta) = \dot{q}^T G(q, \theta)$, llegamos a:

$$\dot{H}_2(e, \dot{q}) = -\dot{q}^T K_d \dot{q} \quad (\text{A.41})$$

que como se puede apreciar es semidefinida negativa. Se puede verificar que el mayor conjunto invariante en el que $\dot{H}_2(e, \dot{q}) = 0$ es el punto de equilibrio $(e, \dot{q}) = (0, 0)$. Aplicando la teoría de invarianza de LaSalle el sistema es global y asintóticamente estable en el punto de equilibrio $(e, \dot{q}) = (0, 0)$. Además, la condición de K_p asegura la unicidad del equilibrio.

A.2.2.1. Resultados reales de las ejecuciones para una curva de Hermite sin puntos intermedios

Tabla A-13 Comparación de Referencia y Posición para las 6 articulaciones del controlador PD+G(qd)
(abcisas: posición (rad); ordenadas: períodos de muestreo ($T_s=0.005s$))

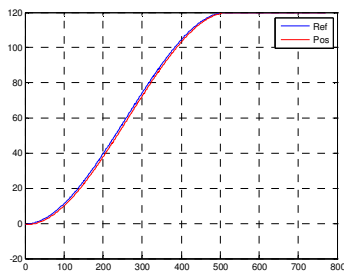


Figura A-73 Articulación 1

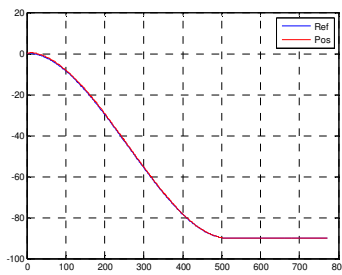


Figura A-74 Articulación 2

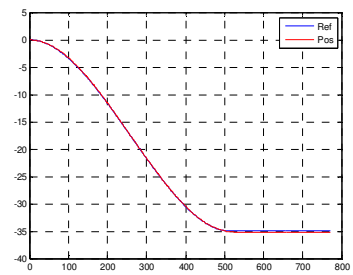


Figura A-75 Articulación 3

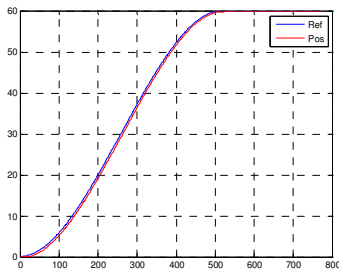


Figura A-76 Articulación 4

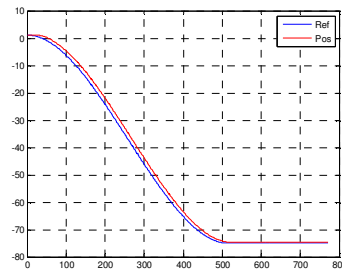


Figura A-77 Articulación 5

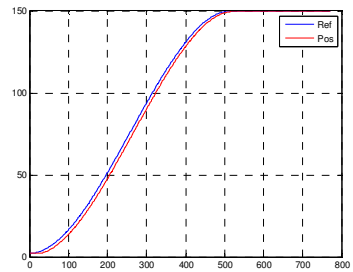


Figura A-78 Articulación 6

Tabla A-14 Acciones de Control del controlador PD+G(q)
(abcisas: acción de control (Nm); ordenadas: periodos de muestreo ($T_s=0.005s$))

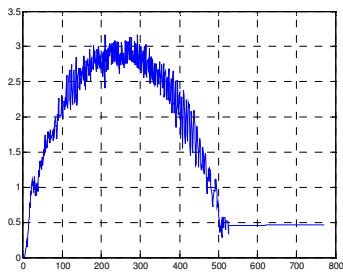


Figura A-79 Articulación 1

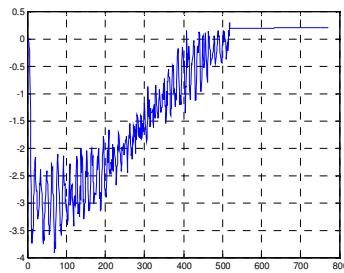


Figura A-80 Articulación 2

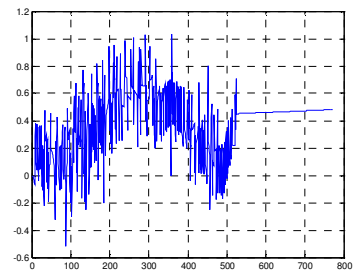


Figura A-81 Articulación 3

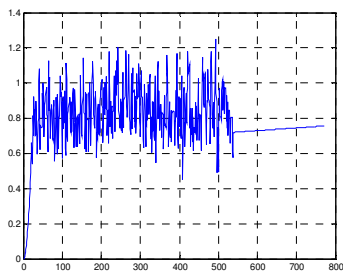


Figura A-82 Articulación 4

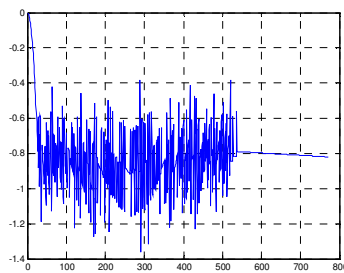


Figura A-83 Articulación 5

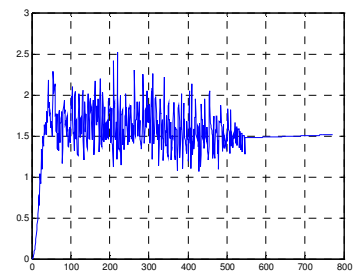


Figura A-84 Articulación 6

A.2.3. Regulador con sólo medida de la posición

En este controlador la ley de control que se propone en (Berghuis, Nijmeijer, 1992) es la siguiente:

$$\begin{aligned}\tau &= -K_p e - K_d \dot{x} + G(q, \theta) \\ \dot{x} &= -Lx + K_d e\end{aligned}\tag{A.42}$$

donde

L : matriz diagonal definida positiva.

Esta ley de control es similar a la del PD con compensación de la gravedad, compuesta por una parte de compensación de la gravedad y por un control proporcional derivativo que utiliza una estimación en lugar de la medida de la velocidad.

Teorema A.3.-

El controlador de posición (A.42) estabiliza el sistema robot global y asintóticamente en el punto de equilibrio $(\dot{q}, e, x) \equiv 0$.

Demostración.-

Calculamos primero el error en bucle cerrado. Nos viene dado por las siguientes ecuaciones:

$$M(q, \theta)\ddot{q} + C(q, \dot{q}, \theta)\dot{q} + K_d \dot{x} + K_p e = 0\tag{A.43a}$$

$$\dot{x} = -Lx + K_d e\tag{A.43b}$$

Tomamos como función de la energía la siguiente función candidata de Lyapunov:

$$H_2(\dot{q}, e, x) = \frac{1}{2} \dot{q}^T M(q, \theta) \dot{q} + \frac{1}{2} e^T K_p e + \frac{1}{2} (K_d e - Lx)^T (K_d e - Lx)\tag{A.44}$$

Calculamos su derivada en el tiempo y utilizando (A.43a) tenemos que:

$$\begin{aligned} \dot{H}_2(\dot{q}, e, x) = & \dot{q}^T M(q, \theta) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{M}(q, \theta) \dot{q} + \frac{1}{2} \dot{e}^T K_p e + \\ & + (K_d \dot{e} - L\dot{x})^T (K_d e - Lx) \end{aligned} \quad (\text{A.45})$$

Teniendo en cuenta la ecuación del error:

$$\begin{aligned} \dot{H}_2(\dot{q}, e, x) = & \dot{q}^T (-C(q, \dot{q}, \theta) \dot{q} - K_d \dot{x} - K_p e) + \frac{1}{2} \dot{q}^T \dot{M}(q, \theta) \dot{q} + \\ & + \frac{1}{2} \dot{e}^T K_p e + (K_d \dot{e})^T \dot{x} - (L(K_d e - Lx))^T (K_d e - Lx) \end{aligned} \quad (\text{A.46})$$

Agrupando términos queda:

$$\dot{H}_2(\dot{q}, e, x) = -(K_d e - Lx)^T L (K_d e - Lx) \quad (\text{A.47})$$

que teniendo en cuenta (A.43b):

$$\dot{H}_2(\dot{q}, e, x) = -\dot{x}^T L \dot{x} \quad (\text{A.48})$$

En la ecuación anterior se puede apreciar que $\dot{H}_2(\dot{q}, e, x)$ es sólo semidefinida negativa. Podemos aplicar el teorema de invarianza de LaSalle para completar la prueba de estabilidad. Con este fin debemos verificar que el mayor conjunto invariante en $\{(\dot{q}, e, x) | \dot{H}_2(\dot{q}, e, x) = 0\} \equiv \{(\dot{q}, e, x) | \dot{x} = 0\}$ es el estado de equilibrio $(\dot{q}, e, x) = 0$. Si $\dot{x} = 0$ entonces x es una constante. Si x es una constante y a partir de (A.43b) tenemos que e también es constante, lo que implica que $\dot{q} = 0$ y $\ddot{q} = 0$. Luego a partir de (A.43a) obtenemos que $e=0$ y utilizando (A.43b) $x=0$, lo cual implica la tesis.

A.1.3.2. Resultados reales de las ejecuciones para una trayectoria en el espacio cartesiano del Robot con curvas polinomiales de tercer orden con interpolación en puntos intermedios

**Tabla A-15 Comparación de Referencia y Posición para las 6 articulaciones del controlador PD+CL(q)
(abcisas: posición (rad); ordenadas: períodos de muestreo (Ts=0.005s))**

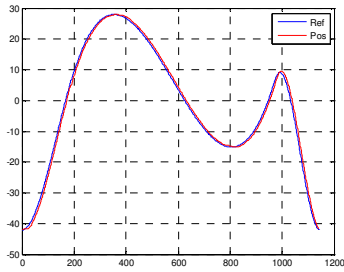


Figura A-85 Articulación 1

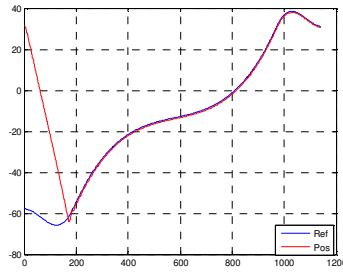


Figura A-86 Articulación 2

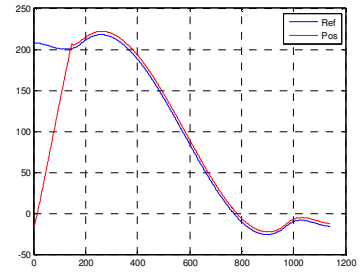


Figura A-87 Articulación 3

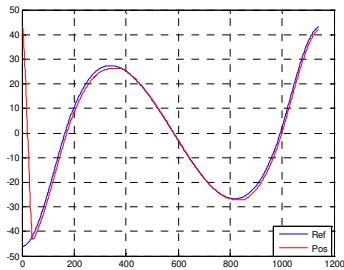


Figura A-88 Articulación 4

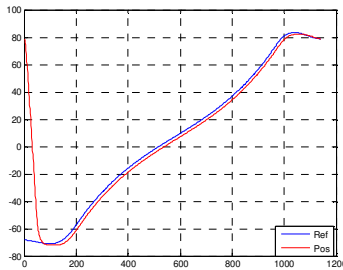


Figura A-89 Articulación 5

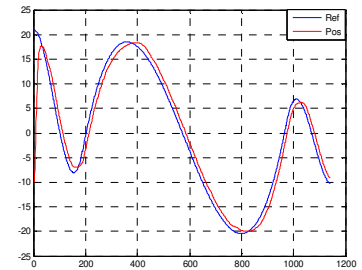


Figura A-90 Articulación 6

Tabla A-16 Acciones de Control controlador PD+CL(q)
(abcisas: acción de control (Nm); ordenadas: períodos de muestreo ($T_s=0.005s$))

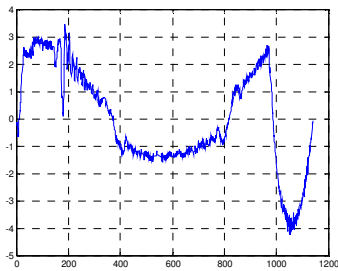


Figura A-91 Articulación 1

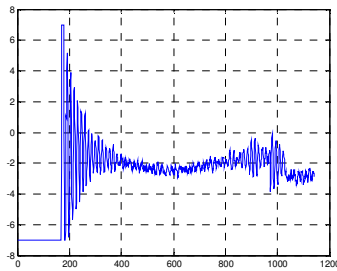


Figura A-92 Articulación 2

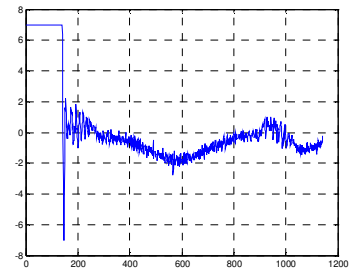


Figura A-93 Articulación 3

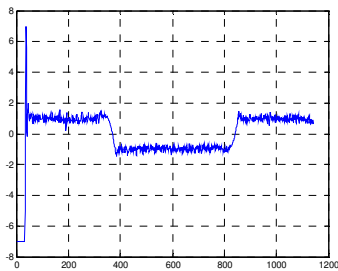


Figura A-94 Articulación 4

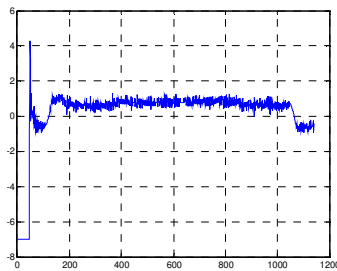


Figura A-95 Articulación 5

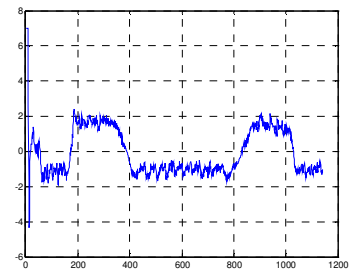


Figura A-96 Articulación 6

A.2.4. Controlador sin medida de la velocidad con compensación de las fuerzas de gravedad en la posición deseada

La ley de control de este controlador es:

$$\tau = -K_p e - K_d \dot{x} + G(q_d, \theta) \quad (\text{A.49a})$$

$$\dot{x} = -Lx + K_d e \quad (\text{A.49b})$$

donde:

K_p, K_d, L : matrices diagonales definidas positivas.

Teorema A.4.-

El controlador (A.49) estabiliza global y asintóticamente el sistema robot en el punto de equilibrio $(\dot{q}, e, \vartheta) = (0,0,0)$ si

$$K_{p,m} > k_g \quad (\text{A.50})$$

donde k_g está definida en (A.35).

Demostración.-

La ecuación del error en bucle cerrado viene dada por

$$M(q, \theta)\ddot{q} + C(q, \dot{q}, \theta)\dot{q} + G(q, \theta) - G(q_d, \theta) + K_d \dot{x} + K_p e = 0 \quad (\text{A.51a})$$

$$\dot{x} = -Lx + K_d e \quad (\text{A.51b})$$

La función de la energía que se propone como candidata de Lyapunov es:

$$\begin{aligned}
 H_1(\dot{q}, e, x) = & \frac{1}{2} \dot{q}^T M(q, \theta) \dot{q} + V(q, q_d) - \\
 & - V(q_d, q_d) + \frac{1}{2} (K_d e - Lx)^T (K_d e - Lx)
 \end{aligned} \tag{A.52}$$

donde la función $V(q, q_d)$ es la misma que se utilizó en el PD con compensación de la gravedad en la posición deseada. La función H_1 es definida positiva y tiene un mínimo absoluto en $(\dot{q}, e, \vartheta) = (0, 0, 0)$ si se cumple la condición (A.50).

La derivada respecto del tiempo de la función de la energía es:

$$\begin{aligned}
 \dot{H}_1(\dot{q}, e, x) = & \dot{q}^T M(q, \theta) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{M}(q, \theta) \dot{q} + \dot{V}(q, q_d) - \dot{V}(q_d, q_d) + \\
 & + \dot{q}^T K_d \dot{x} - (K_d e - Lx)^T L (K_d e - Lx)
 \end{aligned} \tag{A.53}$$

A partir de la ecuación del error, la hemisimetría de $M(q, \theta) - 2C(q, \theta)$, y teniendo en cuenta que

$$\dot{V}(q, q_d) - \dot{V}(q_d, q_d) = \dot{P}(q, \theta) - \dot{q}^T G(q_d, \theta) + \dot{q}^T K_p e \tag{A.54}$$

la derivada de la función de la energía queda

$$\begin{aligned}
 \dot{H}_1(\dot{q}, e, x) = & \dot{q}^T (-G(q, \theta) + G(q_d, \theta) - K_d \dot{x} - K_p e) + \dot{P}(q, \theta) - \\
 & - \dot{q}^T G(q_d, \theta) + \dot{q}^T K_p e + \dot{q}^T K_d \dot{x} - (K_d e - Lx)^T L (K_d e - Lx)
 \end{aligned} \tag{A.55}$$

Agrupando términos:

$$\dot{H}_1(\dot{q}, e, x) = -(K_d e - Lx)^T L (K_d e - Lx) \tag{A.56}$$

que teniendo en cuenta (A.51b) se puede reescribir como

$$\dot{H}_1(\dot{q}, e, x) = -\dot{x}^T L \dot{x} \quad (\text{A.57})$$

la cual es semidefinida negativa tal como vimos en el controlador anterior. De esta forma, si aplicamos la teoría de invarianza de LaSalle podemos demostrar que el sistema es global y asintóticamente estable.

A.2.4.1. Resultados reales de las ejecuciones para una trayectoria en el espacio cartesiano del Robot con curvas polinomiales de tercer orden con interpolación en puntos intermedios

Tabla A-17 Comparación de Referencia y Posición para las 6 articulaciones controlador PD+CL(qd)
(abcisas: posición (rad); ordenadas: períodos de muestreo ($T_s=0.005s$))

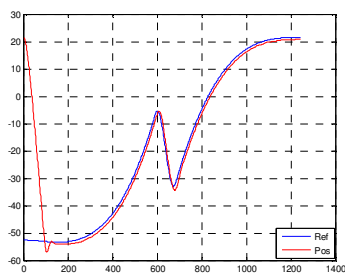


Figura A-97 Articulación 1

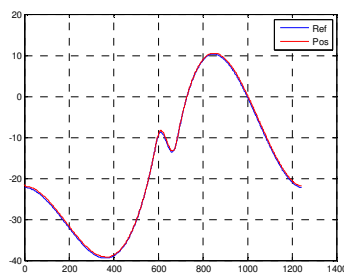


Figura A-98 Articulación 2

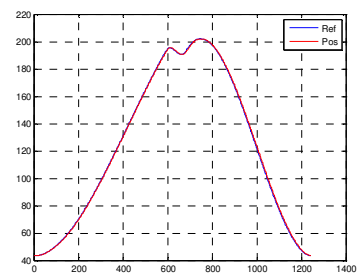


Figura A-99 Articulación 3

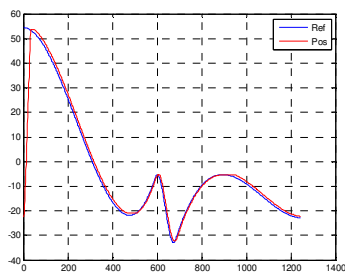


Figura A-100 Articulación 4

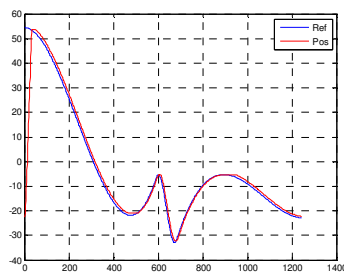


Figura A-101 Articulación 5

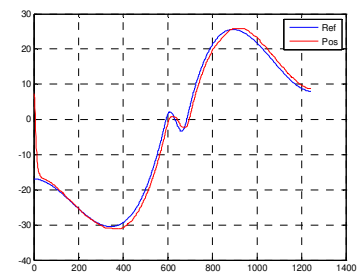


Figura A-102 Articulación 6

Tabla A-18 Acciones de Control controlador PD+CL(qd)
 (abcisas: acción de control (Nm); ordenadas: períodos de muestreo ($T_s=0.005s$))

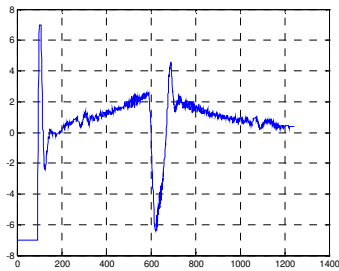


Figura A-103 Articulación 1

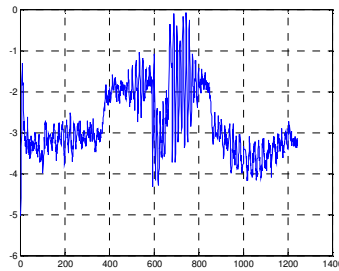


Figura A-104 Articulación 2

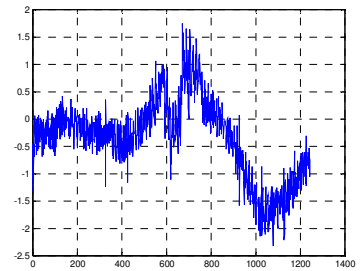


Figura A-105 Articulación 3

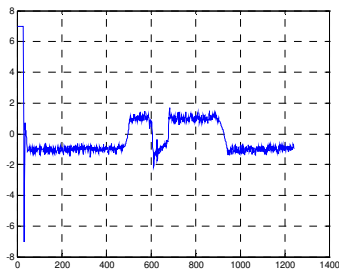


Figura A-106 Articulación 4

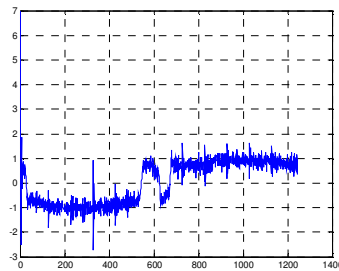


Figura A-107 Articulación 5

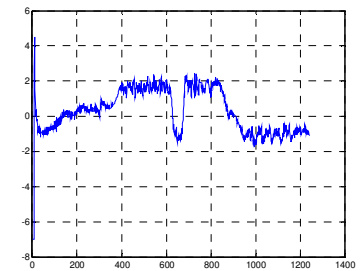


Figura A-108 Articulación 6

A.2.5. Controlador con diferenciación aproximativa

La ley de control de este controlador fue planteada por varios autores, como (Kelly, 93) o (Berghuis, Nijmeijer, 93), y es la siguiente:

$$\tau = G(q, \theta) - K_d \text{diag} \left\{ \frac{b_i p}{p + a_i} \right\} q - K_p e \quad (\text{A.58})$$

donde

a_i, b_i : constantes positivas

p : el operador diferencial $\frac{d}{dt}$

En forma matricial la ley de control anterior queda:

$$\tau = G(q, \theta) - K_d \vartheta - K_p e \quad (\text{A.59a})$$

$$\dot{\vartheta} = -A \vartheta + B \dot{q} \quad (\text{A.59b})$$

donde A, B : matrices diagonales definidas positivas.

Teorema A.5.-

El controlador de posición estabiliza el sistema robot global y asintóticamente en el punto de equilibrio $(\dot{q}, e, \vartheta) \equiv 0$.

Demostración.-

Calculamos primero el error en bucle cerrado. Nos viene dado por las siguientes ecuaciones:

$$M(q, \theta) \ddot{q} + C(q, \dot{q}, \theta) \dot{q} + K_d \vartheta + K_p e = 0 \quad (\text{A.60a})$$

$$\dot{\vartheta} = -A \vartheta + B \dot{q} \quad (\text{A.60b})$$

Tomamos como función de la energía la siguiente función candidata de Lyapunov:

$$H_2(\dot{q}, e, \vartheta) = \frac{1}{2} \dot{q}^T M(q, \theta) \dot{q} + \frac{1}{2} e^T K_p e + \frac{1}{2} (\vartheta^T K_d B^{-1} \vartheta) \quad (\text{A.61})$$

Cuya derivada en el tiempo viene dada por

$$\dot{H}_2(\dot{q}, e, \vartheta) = \dot{q}^T M(q, \theta) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{M}(q, \theta) \dot{q} + \dot{q}^T K_p e + \vartheta^T K_d B^{-1} \dot{\vartheta} \quad (\text{A.62})$$

Aplicando la ecuación del error:

$$\begin{aligned} \dot{H}_2(\dot{q}, e, \vartheta) = & \dot{q}^T (-C(q, \dot{q}, \theta) \dot{q} - K_d \vartheta - K_p e) + \\ & + \frac{1}{2} \dot{q}^T \dot{M}(q, \theta) \dot{q} + \dot{q}^T K_p e + \vartheta^T K_d B^{-1} (-A \vartheta + B \dot{q}) \end{aligned} \quad (\text{A.63})$$

Si se trabaja con la expresión anterior se llega a:

$$\dot{H}_2(\dot{q}, e, \vartheta) = -\vartheta^T K_d B^{-1} A \vartheta \quad (\text{A.64})$$

En la cual se puede ver que $\dot{H}_2(\dot{q}, e, \vartheta)$ es sólo semidefinida negativa. Se puede verificar que el mayor conjunto invariante en que $\dot{H}_2(\dot{q}, e, \vartheta) = 0$ es el estado de equilibrio $(\dot{q}, e, \vartheta) = 0$, y por lo tanto se comprueba la estabilidad.

A.2.5.1. Resultados reales de las ejecuciones para una trayectoria de Hermite

Tabla A-19 Comparación de Referencia y Posición para las 6 articulaciones del controlador PD+DA(q)
(abcisas: posición (rad); ordenadas: períodos de muestreo (Ts=0.005s))

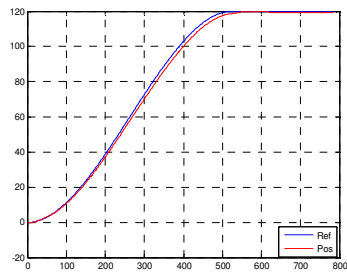


Figura A-109 Articulación 1

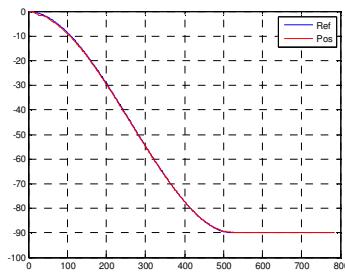


Figura A-110 Articulación 2

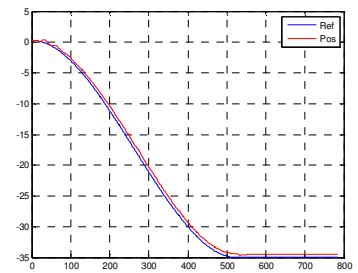


Figura A-111 Articulación 3

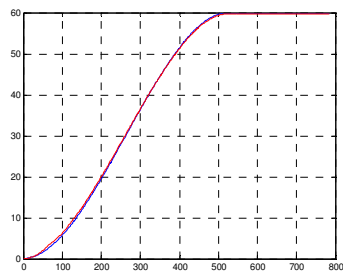


Figura A-112 Articulación 4

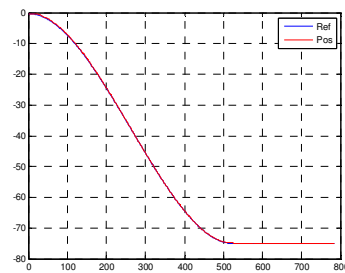


Figura A-113 Articulación 5

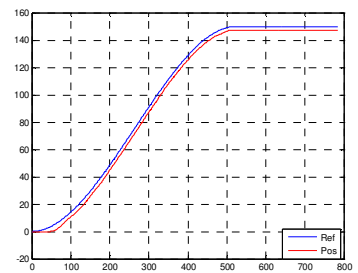


Figura A-114 Articulación 6

Tabla A-20 Acciones de Control del controlador PD+DA(q)
 (abcisas: acción de control (Nm); ordenadas: períodos de muestreo (Ts=0.005s))

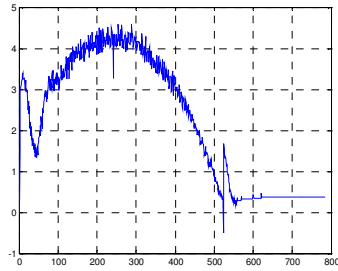


Figura A-115 Articulación 1

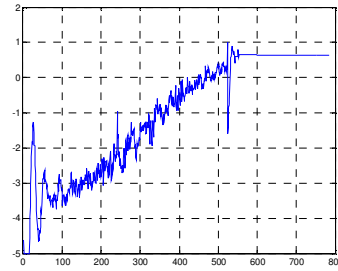


Figura A-116 Articulación 2

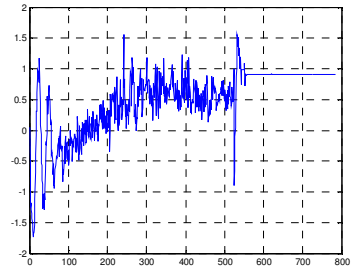


Figura A-117 Articulación 3

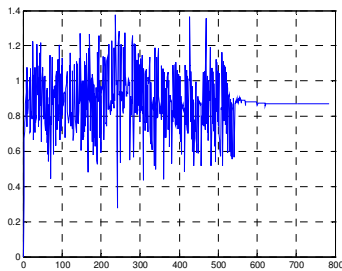


Figura A-118 Articulación 4

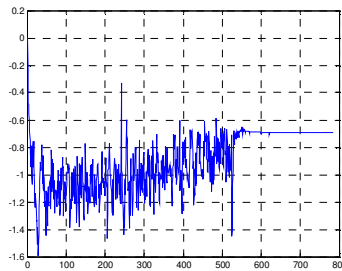


Figura A-119 Articulación 5

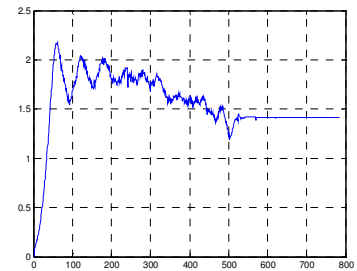


Figura A-120 Articulación 6

A.2.6. Controlador con diferenciación aproximativa con compensación de las fuerzas de gravedad en la posición deseada

La ley de control de este controlador es:

$$\tau = G(q_d, \theta) - K_d \dot{\vartheta} - K_p e \quad (\text{A.65a})$$

$$\dot{\vartheta} = -A\vartheta + B\dot{q} \quad (\text{A.65b})$$

donde

K_p, K_d, A, B : matrices diagonales definidas positivas.

Teorema A I.6.-

El controlador (A.65) estabiliza global y asintóticamente el sistema robot en el punto de equilibrio $(\dot{q}, e, \vartheta) = (0,0,0)$ si se cumple que

$$K_{p,m} > k_g \quad (\text{A.66})$$

con k_g definida en (A.35).

Demostración.-

La ecuación del error en bucle cerrado es:

$$M(q, \theta)\ddot{q} + C(q, \dot{q}, \theta)\dot{q} + G(q, \theta) - G(q_d, \theta) + K_d\vartheta + K_p e = 0 \quad (\text{A.67a})$$

$$\dot{\vartheta} = -A\vartheta + B\dot{q} \quad (\text{A.67b})$$

La función de la energía que se propone como candidata de Lyapunov es:

$$H_1(\dot{q}, e, \vartheta) = \frac{1}{2}\dot{q}^T M(q, \theta)\dot{q} + V(q, q_d) - V(q_d, q_d) + \frac{1}{2}(\vartheta^T K_d B^{-1} \vartheta) \quad (\text{A.68})$$

68)

donde la función $V(q, q_d)$ es la misma que se utilizó en el PD con compensación de la gravedad en la posición deseada, y la función H_1 es definida positiva y tiene un mínimo absoluto en $(\dot{q}, e, \vartheta) = (0,0,0)$ si se cumple la condición (A.66).

De esta forma, la derivada respecto del tiempo de la función de la energía es:

$$\begin{aligned} \dot{H}_1(\dot{q}, e, \vartheta) = & \dot{q}^T M(q, \theta) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{M}(q, \theta) \dot{q} + \dot{V}(q, q_d) - \\ & - \dot{V}(q_d, q_d) + \dot{q}^T K_d \vartheta - \vartheta^T K_d B^{-1} A \vartheta \end{aligned} \quad (\text{A.69})$$

A partir de la ecuación del error y teniendo en cuenta que

$$\dot{V}(q, q_d) - \dot{V}(q_d, q_d) = \dot{P}(q, \theta) - \dot{q}^T G(q_d, \theta) + \dot{q}^T K_p e \quad (\text{A.70})$$

y la hemisimetría de $M(q, \theta) - 2C(q, \theta)$, la derivada de la función de la energía queda

$$\begin{aligned} \dot{H}_1(\dot{q}, e, \vartheta) = & \dot{q}^T (-G(q, \theta) + G(q_d, \theta) - K_d \vartheta - K_p e) + \dot{P}(q, \theta) - \\ & - \dot{q}^T G(q_d, \theta) + \dot{q}^T K_p e + \dot{q}^T K_d \vartheta - \vartheta^T K_d B^{-1} A \vartheta \end{aligned} \quad (\text{A.71})$$

Agrupando términos:

$$\dot{H}_1(\dot{q}, e, \vartheta) = -\vartheta^T K_d B^{-1} A \vartheta \quad (\text{A.72})$$

La cual es semidefinida negativa tal como vimos en el controlador anterior. Se puede verificar también que el mayor conjunto invariante en el que $\dot{H}_1(\dot{q}, e, \vartheta) = 0$ es el estado de equilibrio $(\dot{q}, e, \vartheta) = 0$, y de esta forma podemos aplicar la teoría de invarianza de LaSalle con lo que el sistema es global y asintóticamente estable.

A.2.6.1. Resultados reales de las ejecuciones para una trayectoria de Hermite

Tabla A-21 Comparación de Referencia y Posición para las 6 articulaciones del controlador PD+DA(qd)
(abcisas: posición (rad); ordenadas: períodos de muestreo (Ts=0.005s))

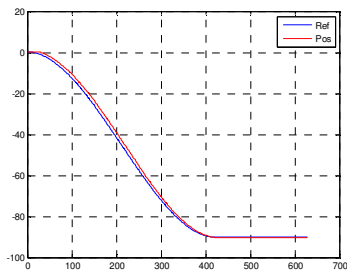


Figura A-121 Articulación 1

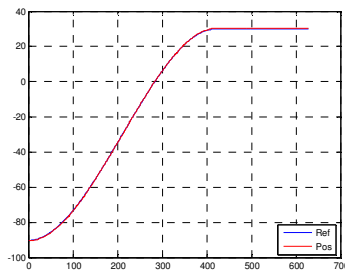


Figura A-122 Articulación 2

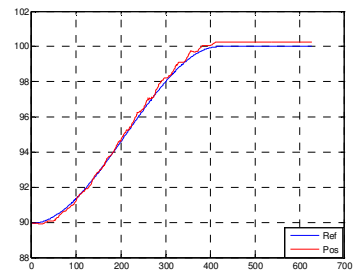


Figura A-123 Articulación 3

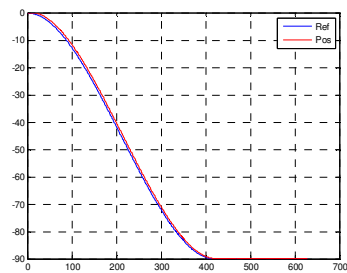


Figura A-124 Articulación 4

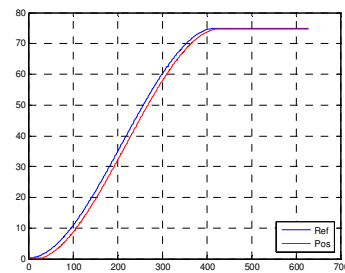


Figura A-125 Articulación 5

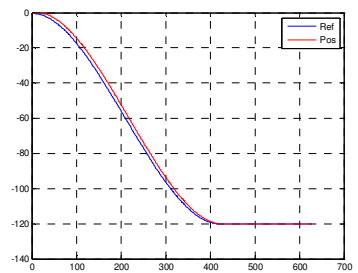


Figura A-126 Articulación 6

Tabla A-22 Acciones de Control del controlador PD+DA(qd)
(abcisas: acción de control (Nm); ordenadas: períodos de muestreo ($T_s=0.005s$))

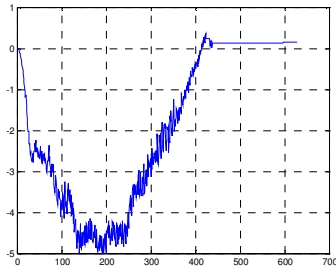


Figura A-127 Articulación 1

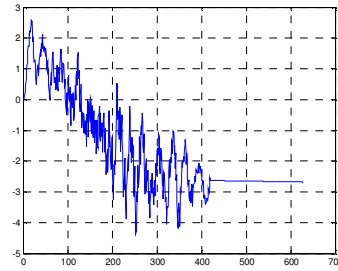


Figura A-128 Articulación 2

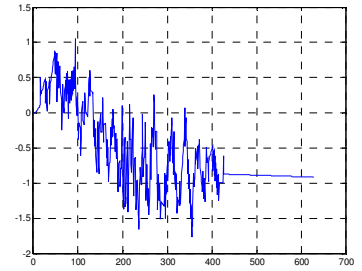


Figura A-129 Articulación 3

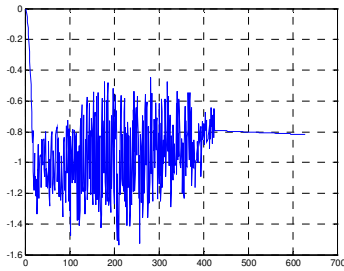


Figura A-130 Articulación 4

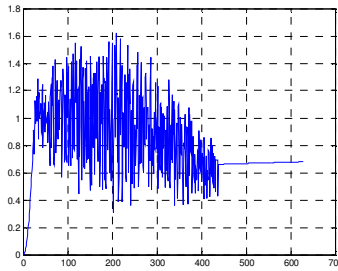


Figura A-131 Articulación 5

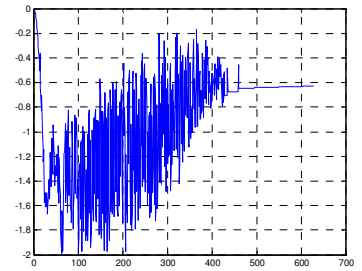


Figura A-132 Articulación 6

A.2.7. Controlador PID

La ley de control que se propone para este controlador es la siguiente (Wen, Murphy, 1990), (Qu, Dorsey, 1991):

$$\tau_e = -K_p e - K_d \dot{q} - K_i \xi \quad (\text{A.73a})$$

$$\dot{\xi} = e \quad (\text{A.73b})$$

donde

K_p, K_d, K_i : matrices diagonales definidas positivas.

Se puede apreciar que esta ley de control es básicamente la misma del PD con compensación de la gravedad. Únicamente se sustituye el término de compensación de la gravedad por la acción integral.

El problema de control consiste en dar valores a las matrices K_p, K_d, K_i de forma que

$$\lim_{t \rightarrow \infty} e(t) = 0$$

al menos para valores suficientemente pequeños en el instante inicial del error de posición $e(0)$ y de la velocidad $\dot{q}(0)$.

Teorema A I.7.-

El regulador (A.73) estabiliza local y asintóticamente el sistema robotizado en el punto de equilibrio si

$$K_{p,m} > \alpha^2 M_M + \alpha^{-1} K_{i,M} + k_g \quad (\text{A.74a})$$

$$K_{d,m} > M_M \quad (\text{A.74b})$$

Demostración.-

La ecuación del error en bucle cerrado a partir de la ecuación de la dinámica del robot y de la ecuación del controlador (A.73) viene dada por la expresión:

$$M(q, \theta) \ddot{q} + C(q, \dot{q}, \theta) \dot{q} + G(q, \theta) + K_p e + K_d \dot{q} + K_i \xi = 0 \quad (\text{A.75a})$$

$$\dot{\xi} = e \quad (\text{A.75b})$$

Podemos realizamos el cambio de variable $z = -\xi + K_i^{-1} G(q_d, \theta)$, con lo que la ecuación del error queda:

$$M(q, \theta) \ddot{q} + C(q, \dot{q}, \theta) \dot{q} + G(q, \theta) + K_p e + K_d \dot{q} - K_i z + G(q_d, \theta) = 0 \quad (\text{A.76a})$$

$$z = -\xi + K_i^{-1} G(q_d, \theta) \quad (\text{A.76b})$$

La función candidata de Lyapunov que se propone es la siguiente:

$$H_2(e, \dot{q}, z) = V_1(e, \dot{q}, z) + V_2(e) + \frac{1}{2} \alpha e^T K_d e + \frac{1}{2} e^T K_p e \quad (\text{A.77})$$

donde

α : constante positiva menor de la unidad $0 < \alpha < 1$

$$V_1(e, \dot{q}, z) = \frac{1}{2} \begin{bmatrix} \dot{q} & e & z \end{bmatrix} \begin{bmatrix} M(q, \theta) & \alpha M(q, \theta) & 0 \\ \alpha M(q, \theta) & 0 & -K_i \\ 0 & -K_i & \alpha K_i \end{bmatrix} \begin{bmatrix} \dot{q} \\ e \\ z \end{bmatrix} \quad (\text{A.78})$$

$$V_2(e) = P(q, \theta) - P(q_d, \theta) - e^T G(q_d, \theta) \quad (\text{A.79})$$

Nótese que la función $V_2(e)$ guarda la siguiente relación con la función $V(q, q_d)$ utilizada en los controladores que realizaban una compensación de las fuerzas gravitatorias en la posición deseada:

$$V_2(e) + \frac{1}{2} e^T K_p e = V(q, q_d) - V(q_d, q_d) \quad (\text{A.80})$$

Por tanto $V_2(e) + \frac{1}{2} e^T K_p e$ es definida positiva como se ha visto en otros controladores si K_p cumple la siguiente condición:

$$K_{p,m} > k_g \quad (\text{A.81})$$

El tercer término de $H_2(e, \dot{q}, z)$ es definido positivo al ser K_d una matriz diagonal definida positiva.

Aplicando el teorema A.2 y el lema A.1 de matrices particionadas obtendremos una serie de condiciones que nos garantizarán que $V_1(e, \dot{q}, z) + \frac{1}{2} e^T K_p e$ es definida positiva.

La matriz asociada a la forma cuadrática $V_1(e, \dot{q}, z) + \frac{1}{2} e^T K_p e$ a partir de (A.78) da

$$\frac{1}{2} \begin{bmatrix} M(q, \theta) & \alpha M(q, \theta) & 0 \\ \alpha M(q, \theta) & K_p & -K_i \\ 0 & -K_i & \alpha K_i \end{bmatrix} \quad (\text{A.82})$$

que puede ser reescrita en la forma particionada (A.2) escogiendo:

$$A = \frac{1}{2} M(q, \theta) \in R^{n \times n} \quad (\text{A.83})$$

$$B = \frac{1}{2} [\alpha M(q, \theta) \quad 0] \in R^{n \times 2n} \quad (\text{A.84})$$

$$C = \frac{1}{2} \begin{bmatrix} K_p & -K_i \\ -K_i & \alpha K_i \end{bmatrix} \quad (\text{A.85})$$

Aplicando el teorema A.2 la matriz (A.82) será definida positiva si y sólo si:

$$M(q, \theta) > 0 \quad (\text{A.86})$$

$$\begin{bmatrix} K_p - \alpha^2 M(q, \theta) & -K_i \\ -K_i & \alpha K_i \end{bmatrix} > 0 \quad (\text{A.87})$$

La condición (A.86) se cumple siempre puesto que la matriz de inercias es definida positiva para todo $q \in R^n$. La matriz (A.87) se puede reescribir como una nueva matriz particionada siendo:

$$A = K_p - \alpha^2 M(q, \theta) \quad (\text{A.88})$$

$$B = -K_i \quad (\text{A.89})$$

$$C = \alpha K_i \quad (\text{A.90})$$

Aplicando de nuevo el teorema A.2 la matriz (A.87) es definida positiva si y sólo si:

$$K_p - \alpha^2 M(q, \theta) > 0 \quad (\text{A.91})$$

$$\alpha K_i - K_i [K_p - \alpha^2 M(q, \theta)]^{-1} K_i > 0 \quad (\text{A.92})$$

Aplicando el lema A.1 con $N = K_p$, $P = \alpha^2 M(q, \theta)$, $Q = K_i$ y $\varepsilon = \alpha$, se obtienen las siguientes condiciones para que se cumplan (A.91) y (A.92)

$$K_i > 0 \quad (\text{A.93})$$

$$K_p > \alpha^2 M(q, \theta) + \alpha^{-1} K_i \quad (\text{A.94})$$

Por tanto para que $H_2(e, \dot{q}, z)$ sea definida positiva se debe cumplir la condición (A.93) y además:

$$K_p > \alpha^2 M(q, \theta) + \alpha^{-1} K_i + k_g I \quad (\text{A.95})$$

Calculamos ahora la derivada con respecto al tiempo de la función $H_2(e, \dot{q}, z)$

$$\begin{aligned} \dot{H}_2(e, \dot{q}, z) = & \frac{1}{2} \dot{q}^T \dot{M}(q, \theta) \dot{q} + \dot{q}^T M(q, \theta) \ddot{q} + \alpha e^T \dot{M}(q, \theta) \dot{q} + \\ & + \alpha e^T M(q, \theta) \ddot{q} + \alpha \dot{q}^T M(q, \theta) \dot{q} + e^T K_i e - \dot{q}^T K_i z - \\ & - \alpha z^T K_i e + \dot{P}(q, \theta) - \dot{q}^T G(q_d, \theta) + \dot{q}^T K_p e + \alpha \dot{q}^T K_d e \end{aligned} \quad (\text{A.96})$$

Sustituyendo en la ecuación anterior la ecuación del error (A.75) y teniendo en cuenta la hemisimetría de $\dot{M}(q, \theta) - 2C(q, \dot{q}, \theta)$

$$\begin{aligned} \dot{H}_2(e, \dot{q}, z) = & -\dot{q}^T K_d \dot{q} + \alpha e^T C^T(q, \dot{q}, \theta) \dot{q} + \\ & + \alpha e^T [-G(q, \theta) - K_p e + G(q_d, \theta)] + \alpha \dot{q}^T M(q, \theta) \dot{q} + e^T K_i e \end{aligned} \quad (\text{A.97})$$

Agrupando términos

$$\begin{aligned} \dot{H}_2(e, \dot{q}, z) = & -\dot{q}^T [K_d + \alpha M(q, \theta)] \dot{q} + \alpha e^T C^T(q, \dot{q}, \theta) \dot{q} - \\ & - e^T [K_p e - K_i] e - \alpha e^T [G(q_d, \theta) - G(q, \theta)] \end{aligned} \quad (\text{A.98})$$

Utilizando la propiedad 2.4. tenemos que

$$\alpha e^T C^T(q, \dot{q}, \theta) \dot{q} \leq \alpha C_M \|e\| \|\dot{q}\|^2 \quad (\text{A.99})$$

$$-\alpha e^T [G(q_d, \theta) - G(q, \theta)] \leq \alpha k_g \|e\|^2 = \alpha k_g e^T e \quad (\text{A.100})$$

Por tanto

$$\begin{aligned} \dot{H}_2(e, \dot{q}, z) \leq & - \begin{bmatrix} e^T & \dot{q}^T \end{bmatrix} \underbrace{\begin{bmatrix} \alpha K_p - K_i - \alpha k_g I & 0 \\ 0 & K_p - \alpha M(q, \theta) \end{bmatrix}}_Q \begin{bmatrix} e \\ \dot{q} \end{bmatrix} + \\ & + \alpha C_M \|e\| \|\dot{q}\|^2 \end{aligned} \quad (\text{A.101})$$

La matriz Q es definida positiva si y sólo si

$$\alpha K_p - K_i - \alpha k_g I > 0 \quad (\text{A.102})$$

$$K_d - \alpha M(q, \theta) > 0 \quad (\text{A.103})$$

La condición (A.102) se cumple si se cumple (A.95). Puesto que $0 < \alpha < 1$ la condición (A.103) se cumple si

$$K_d > M(q, \theta) \quad (\text{A.104})$$

En conclusión si K_p , K_d y K_i satisfacen las condiciones (A.93), (A.95) y (A.104) con $0 < \alpha < 1$ entonces $H_2(e, \dot{q}, z)$ es globalmente definida positiva mientras que $\dot{H}_2(e, \dot{q}, z)$ es localmente semidefinida negativa.

Con el fin de aplicar el teorema de invarianza de LaSalle debemos verificar que el mayor conjunto invariante que verifica que $\dot{H}_2(e, \dot{q}, z) = 0$ es el estado de equilibrio $(\dot{q}, e, z) = (0, 0, 0)$. El origen es la única solución a la ecuación del error (A.75) que cumple que $\dot{H}_2(e, \dot{q}, z) = 0$.

Aplicando el teorema de invarianza de LaSalle el origen es un estado de equilibrio localmente asintóticamente estable.

A.2.7.1. Resultados reales de las ejecuciones para una curva de Hermite sin puntos intermedios

Tabla A-23 Comparación de Referencia y Posición para las 6 articulaciones del controlador PID (abcisas: posición (rad); ordenadas: períodos de muestreo ($T_s=0.005s$))

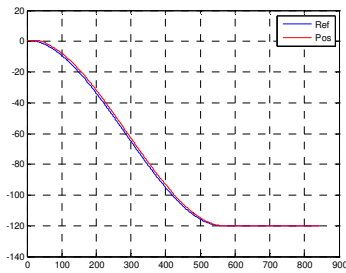


Figura A-133 Articulación 1

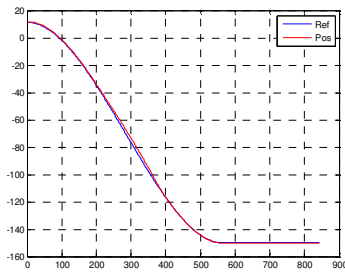


Figura A-134 Articulación 2

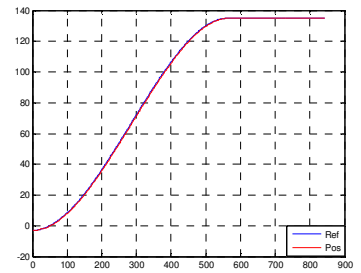


Figura A-135 Articulación 3

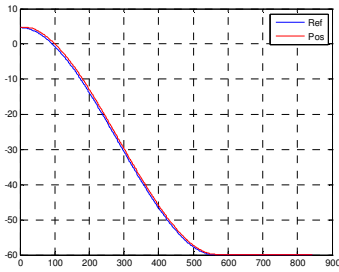


Figura A-136 Articulación 4

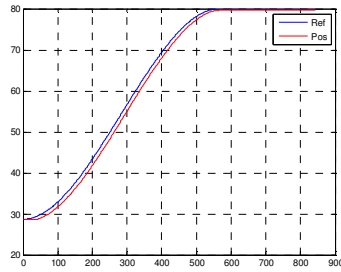


Figura A-137 Articulación 5

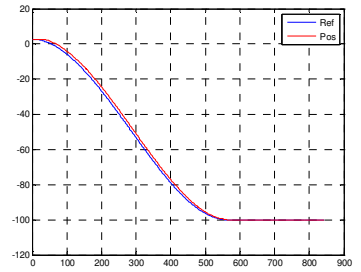


Figura A-138 Articulación 6

Tabla A-24 Acciones de Control del controlador PID
(abcisas: acción de control (Nm); ordenadas: períodos de muestreo ($T_s=0.005s$))

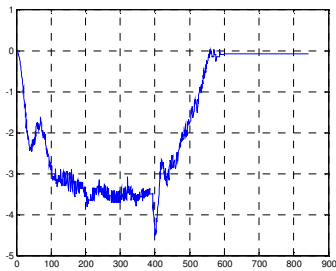


Figura A-139 Articulación 1

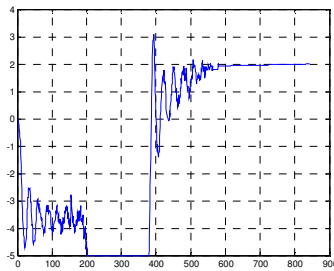


Figura A-140 Articulación 2

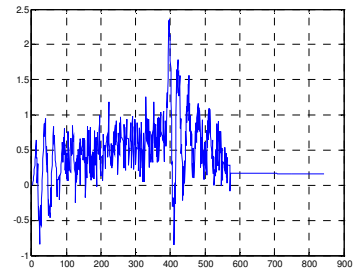


Figura A-141 Articulación 3

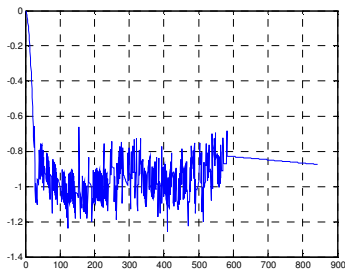


Figura A-142 Articulación 4

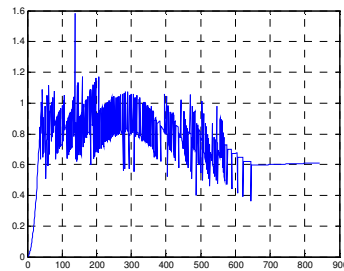


Figura A-143 Articulación 5

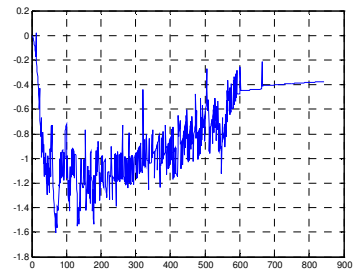


Figura A-144 Articulación 6

A.2.7.2. Resultados reales de las ejecuciones para una trayectoria en el espacio cartesiano del Robot con curvas polinomiales de tercer orden con interpolación en puntos intermedios

Tabla A-25 Comparación de Referencia y Posición para las 6 articulaciones del controlador PID (abcisas: posición (rad); ordenadas: periodos de muestreo ($T_s=0.005s$))

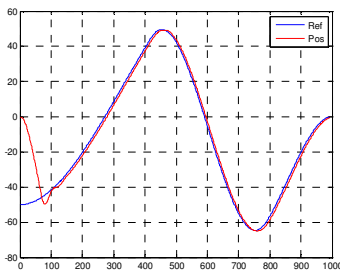


Figura A-145 Articulación 1

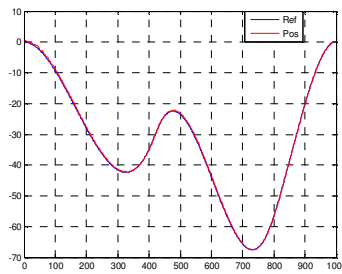


Figura A-146 Articulación 2

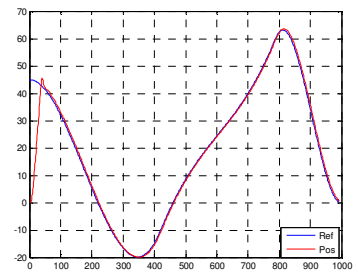


Figura A-147 Articulación 3

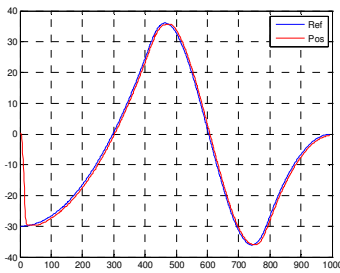


Figura A-148 Articulación 4

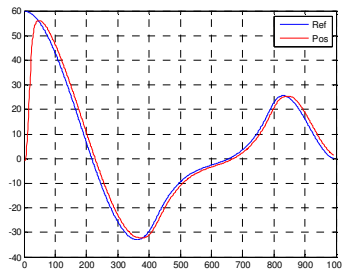


Figura A-149 Articulación 5

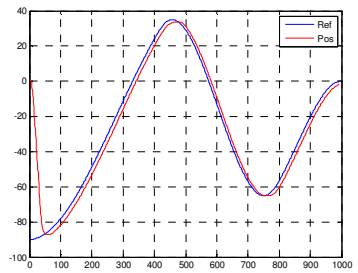


Figura A-150 Articulación 6

Tabla A-26 Acciones de Control del controlador PID
(abcisas: acción de control (Nm); ordenadas: periodos de muestreo ($T_s=0.005s$))

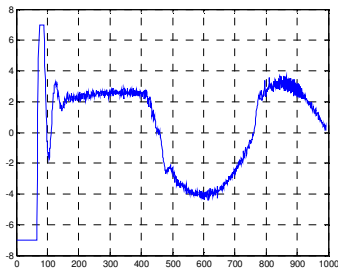


Figura A-151 Articulación 1

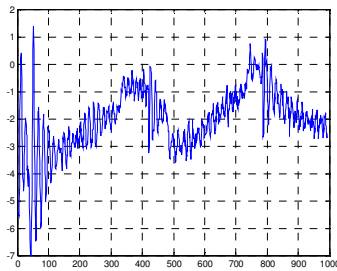


Figura A-152 Articulación 2

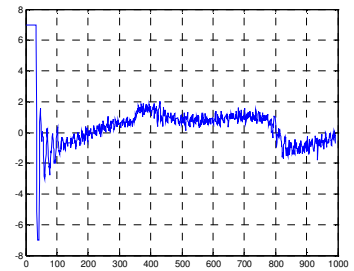


Figura A-153 Articulación 3

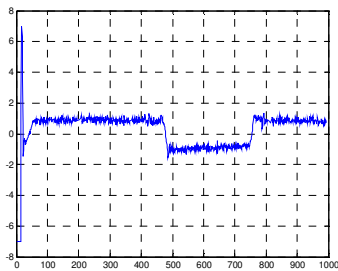


Figura A-154 Articulación 4

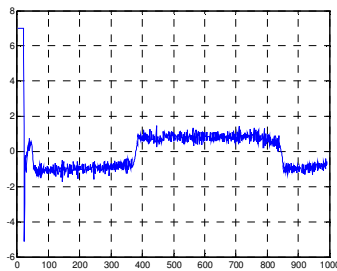


Figura A-155 Articulación 5

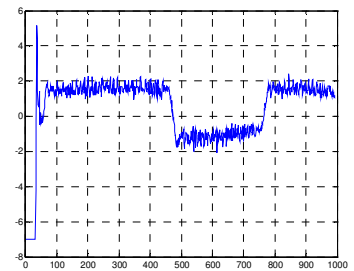


Figura A-156 Articulación 6

A.2.7.3. Resultados reales de las ejecuciones para una trayectoria de Fourier

Tabla A-27 Comparación de Referencia y Posición para las 6 articulaciones del controlador PID
(abcisas: posición (rad); ordenadas: períodos de muestreo ($T_s=0.005s$))

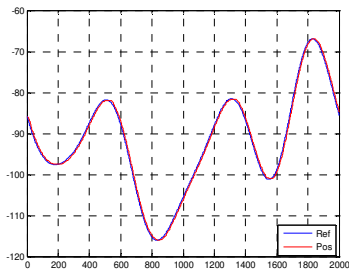


Figura A-157 Articulación 1

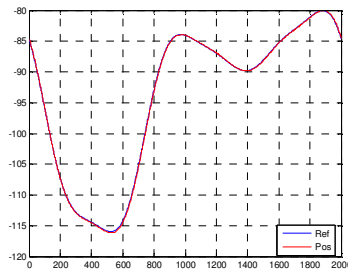


Figura A-158 Articulación 2

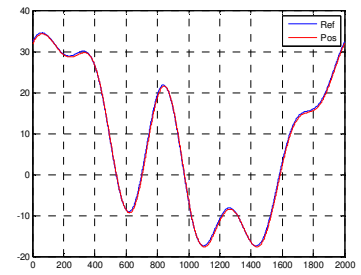


Figura A-159 Articulación 3

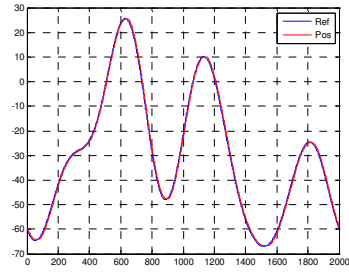


Figura A-160 Articulación 4

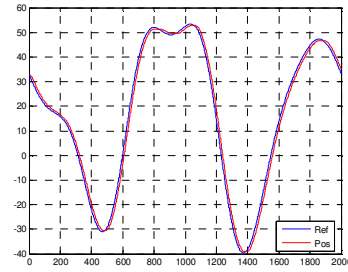


Figura A-161 Articulación 5

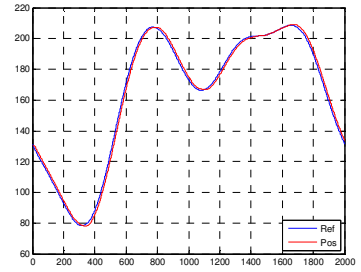


Figura A-162 Articulación 6

Tabla A-28 Acciones de Control del controlador PID
(abcisas: acción de control (Nm); ordenadas: periodos de muestreo ($T_s=0.005s$))

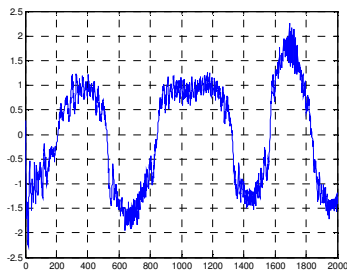


Figura A-163 Articulación 1

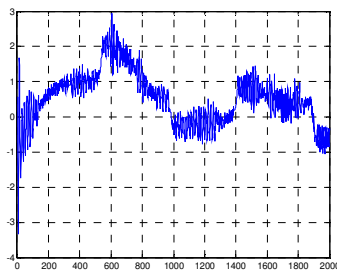


Figura A-164 Articulación 2

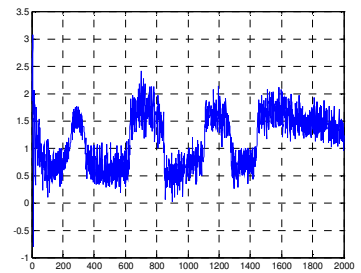


Figura A-165 Articulación 3

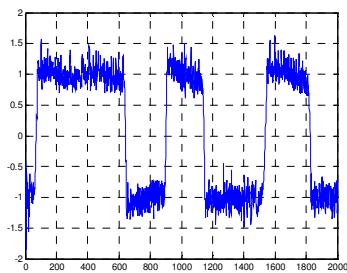


Figura A-166 Articulación 4

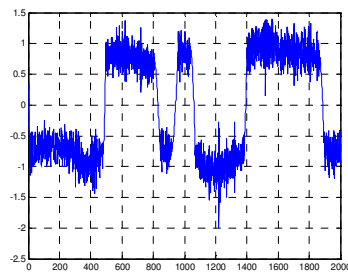


Figura A-167 Articulación 5

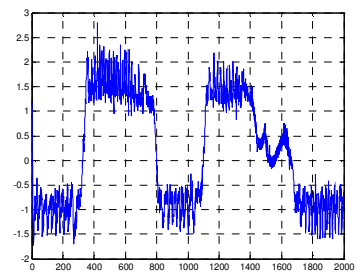


Figura A-168 Articulación 6

A.2.8. Controlador PI²D

En esta última estrategia de control (Ortega, Loria, Kelly, 93) presentan un controlador que incorpora un segundo término integral en el filtro de la velocidad, el cual motiva el nombre de PI²D.

La ley de control propuesta es la siguiente:

$$\tau = -K_p e + v - K_d \vartheta \quad (\text{A.105a})$$

$$\dot{v} = -K_i (e + \vartheta) \quad (\text{A.105b})$$

$$\dot{\vartheta} = -A \vartheta + B \dot{q} \quad (\text{A.105c})$$

que se puede reescribir como

$$\tau = -K_p e - K_i \int_0^t e \, dt - K_d \vartheta - K_i \int_0^t \vartheta \, dt \quad (\text{A.106a})$$

$$\dot{\vartheta} = -A \vartheta + B \dot{q} \quad (\text{A.106b})$$

Podemos observar que, como se ha comentado al principio de este apartado, el controlador está constituido de cuatro partes: el término proporcional, el derivativo y los dos integrales: el de compensación de los efectos de la gravedad y el del filtro de velocidad.

Antes de comenzar la demostración de la estabilidad asintótica, definimos:

$$K'_p \equiv K_p - \frac{1}{\varepsilon} K_i \quad (\text{A.107})$$

donde, para una K_i dada, $\varepsilon > 0$ se escoge de forma que

$$I > \frac{1}{\varepsilon} K_i \quad (\text{A.108})$$

Teorema A.8.-

El regulador (A.105) estabiliza asintóticamente semiglobalmente el sistema robotizado bajo las condiciones:

$$K_{p,m} > 4k_g + 1 \quad (\text{A.109a})$$

$$B_m > 2M_M / M_m \quad (\text{A.109b})$$

$$K'_{p,m} > 4k_g \quad (\text{A.109c})$$

$$K_{i,M} < \varepsilon \quad (\text{A.109d})$$

$$\varepsilon < \frac{1}{4} \sqrt{\frac{K'_{p,m}}{M_M}} \quad (\text{A.109e})$$

$$\varepsilon < \sqrt{\frac{\lambda_m(K_d B^{-1})}{8M_M}} \quad (\text{A.109f})$$

$$\varepsilon < \frac{(2K'_{p,m} - k_g) \lambda_m(K_d B^{-1})}{2[K'_{p,M} + K_{d,M} + k_g]^2} \quad (\text{A.109g})$$

$$\varepsilon < \frac{\lambda_m(K_d A) M_m}{2[A_M M_M]^2} \quad (\text{A.109h})$$

$$\|x\| < \frac{1}{2C_M} \left[\frac{1}{2} B_m M_m - M_M \right] \quad (\text{A.109i})$$

$$\varepsilon < \frac{\lambda_m(K_d B^{-1} A)}{2K_{d,M}} \quad (\text{A.109j})$$

Una región de atracción viene dada por

$$B = \left\{ x \in R^{4n} \mid \|x\| < \frac{1}{2C_M} \left[\frac{1}{2} B_m M_m - M_M \right] \sqrt{\frac{P_m}{P_M}} \right\} \quad (\text{A.110})$$

donde

$$x^T \equiv [e^T \quad \dot{q}^T \quad \vartheta^T \quad v^T]$$

$$P_m = \frac{1}{4} \min \left\{ K'_{p,m} + \left[2 \frac{K_{i,m}^2}{\varepsilon K_{i,M}} - 1 \right], \lambda_m(K_d B^{-1}), M_m, 2 \left[\frac{\varepsilon}{K_{i,M}} - 1 \right] \right\}$$

$$P_M \leq \max \left\{ \frac{1}{4} \left[K'_{p,m} + 2k_g \right] + \frac{1}{2} \left[\frac{\varepsilon}{2} M_M + K'_{p,M} \right] + \left[\frac{1}{2} \left(\frac{K_{i,M}}{\varepsilon} + 1 \right) \right], \left(\varepsilon + \frac{1}{2} \right) M_M, \right. \\ \left. \frac{1}{2} \left[\varepsilon M_M + \lambda_m(K_d B^{-1}) \right], \frac{1}{2} (\varepsilon K_{i,M}^{-1} + 1) \right\}$$

Demostración.-

La prueba de la estabilidad se va a basar en la teoría de Lyapunov. Se puede dividir en 5 partes:

- Definir una ecuación adecuada del error para el sistema en bucle cerrado.
- Proponer una función candidata de Lyapunov.
- Probar que bajo las condiciones del teorema la función propuesta es una función de Lyapunov y establecer la estabilidad asintótica en el equilibrio invocando el principio de invarianza de LaSalle.
- Definir el dominio de atracción.
- Probar que se puede aumentar arbitrariamente.

Ecuación del error

Descomponemos el término proporcional del controlador en:

$$K_p e = K'_p (q - \delta) + \frac{1}{\varepsilon} K_i e + G(q_d, \theta) \quad (\text{A.111})$$

donde δ se define como:

$$\delta \equiv q_d + (K'_p)^{-1} G(q_d, \theta) \quad (\text{A.112})$$

A partir de la ecuación de la dinámica del robot y de la ley de control obtenemos la ecuación del error en bucle cerrado:

$$M(q, \theta)\ddot{q} + C(q, \dot{q}, \theta)\dot{q} + G(q, \theta) + K'_p(q - \delta) + K_d x = -\frac{1}{\varepsilon} K_i e + y \quad (\text{A.113a})$$

$$\dot{\vartheta} = -A\vartheta + B\dot{q} \quad (\text{A.113b})$$

$$\dot{y} = -K_i(e + \vartheta) \quad (\text{A.113c})$$

donde

$$y = v - G(q_d, \theta) \quad (\text{A.114})$$

Para simplificar la derivación de la función de Lyapunov es conveniente introducir el cambio de coordenadas:

$$z = -\frac{1}{\varepsilon} K_i e + y \quad (\text{A.115})$$

La nueva ecuación del error será:

$$M(q, \theta)\ddot{q} + C(q, \dot{q}, \theta)\dot{q} + G(q, \theta) + K'_p(q - \delta) + K_d \vartheta = z \quad (\text{A.116a})$$

$$\dot{\vartheta} = -A\vartheta + B\dot{q} \quad (\text{A.116b})$$

$$\dot{z} = -K_i(e + \frac{1}{\varepsilon}\dot{q} + \vartheta) \quad (\text{A.116c})$$

Función candidata de Lyapunov

En (Ortega, Loría, Kelly, 93) se propone la siguiente función como candidata de Lyapunov:

$$H(\dot{q}, e, \vartheta) = H_1 + H_2 + \varepsilon(H_3 + H_4) \quad (\text{A.117})$$

siendo

$$H_1(e, \dot{q}, \vartheta) = \frac{1}{2} \dot{q}^T M(q, \theta) \dot{q} + P(q) + \frac{1}{2} (q - \delta) K_p' (q - \delta) + \frac{1}{2} \vartheta^T K_d B^{-1} \vartheta + c_1 \quad (\text{A.118})$$

donde $c_1 \equiv -\frac{1}{2} G(q_d, \theta)^T (K_p')^{-1} G(q_d, \theta) - P(q_d, \theta)$ se añade con el fin de forzar a que la energía total del sistema cumpla que $H_i(0,0,0)=0$.

$$H_2(z) \equiv \frac{\varepsilon}{2} z^T K_i^{-1} z \quad (\text{A.119})$$

$$H_3(e, \dot{q}) = e^T M(q, \theta) \dot{q} \quad (\text{A.120})$$

$$H_4(e, \dot{q}, \vartheta) = \vartheta^T M(q, \theta) \dot{q} \quad (\text{A.121})$$

Las derivadas respecto del tiempo son:

$$\dot{H}_1(e, \dot{q}, \vartheta) = \frac{1}{2} \vartheta^T K_d B^{-1} A \vartheta + \dot{q}^T z \quad (\text{A.122})$$

$$\dot{H}_2 = \varepsilon^T K_i^{-1} \dot{z} = -\varepsilon^T K_i^{-1} K_i (e + \frac{1}{\varepsilon} \dot{q} + \vartheta) = -\varepsilon^T (e + \frac{1}{\varepsilon} \dot{q} + \vartheta) \quad (\text{A.123})$$

$$\dot{H}_1 + \dot{H}_2 = \frac{1}{2} \vartheta^T K_d B^{-1} A \vartheta - \varepsilon^T (e + \vartheta) \quad (\text{A.124})$$

$$\begin{aligned}
\dot{H}_3 &= e^T M(q, \theta) \ddot{q} + \dot{q}^T M(q, \theta) \dot{q} + e^T \dot{M}(q, \theta) \dot{q} = \\
&= e^T \left[C^T(q, \dot{q}, \theta) \dot{q} - G(q, \theta) + G(q_d, \theta) - K_p e - K_d \vartheta + z \right] + \dot{q}^T M(q, \theta) \dot{q}
\end{aligned}
\tag{A.125}$$

$$\begin{aligned}
\dot{H}_4 &= \vartheta^T M(q, \theta) \ddot{q} + \dot{\vartheta}^T M(q, \theta) \dot{q} + \vartheta^T \dot{M}(q, \theta) \dot{q} = \\
&= \vartheta^T \left[C^T(q, \dot{q}, \theta) \dot{q} - G(q, \theta) + G(q_d, \theta) - K_p e - K_d \vartheta + z \right] + \dot{\vartheta}^T M(q, \theta) \dot{q}
\end{aligned}
\tag{A.126}$$

Aplicando cotas a \dot{H}_3 y \dot{H}_4 quedan las siguientes desigualdades

$$\dot{H}_3 \leq C_M \|e\| \|\dot{q}\|^2 + k_g \|e\|^2 - K'_{p,m} \|e\|^2 + K_{d,M} \|e\| \|\vartheta\| + e^T z + M_M \|\dot{q}\|^2
\tag{A.127}$$

$$\begin{aligned}
\dot{H}_4 &\leq C_M \|\vartheta\| \|\dot{q}\|^2 + k_g \|e\| \|\vartheta\| - K'_{p,m} \|e\| \|\vartheta\| + K_{d,M} \|\vartheta\|^2 + \vartheta^T z - \\
&\quad - M_M A_M \|\dot{q}\| \|\vartheta\| + B_m M_m \|\dot{q}\|^2
\end{aligned}
\tag{A.128}$$

Tras una serie de cálculos y con las particiones adecuadas queda:

$$\begin{aligned}
 \dot{H} \leq & -\frac{\varepsilon}{2} \begin{bmatrix} \|e\| \\ \|\vartheta\| \end{bmatrix}^T \overbrace{\begin{bmatrix} 2K'_{p,m} - k_g & -(K'_{p,M} - K_{d,M} - k_g) \\ -K'_{p,M} - K_{d,M} - k_g & \frac{1}{2\varepsilon} \lambda_m(K_d B^{-1} A) \end{bmatrix}}^{Q_1} \begin{bmatrix} \|e\| \\ \|\vartheta\| \end{bmatrix} - \\
 & -\frac{\varepsilon}{2} \begin{bmatrix} \|\vartheta\| \\ \|\dot{q}\| \end{bmatrix}^T \overbrace{\begin{bmatrix} \frac{1}{2\varepsilon} \lambda_m(K_d B^{-1} A) & -A_M M_M \\ -A_M M_M & B_m M_m \end{bmatrix}}^{Q_2} \begin{bmatrix} \|\vartheta\| \\ \|\dot{q}\| \end{bmatrix} - \\
 & -\left[\frac{1}{2} B_m M_m - M_M - C_M \|\vartheta\| - C_M \|e\| \right] \|\dot{q}\|^2 - \\
 & -\left[\frac{1}{2} \lambda_m(K_d B^{-1} A) - \varepsilon K'_{p,M} \right] \|\vartheta\|^2
 \end{aligned} \tag{A.129}$$

Estabilidad Asintótica

Lo primero es exponer las condiciones suficientes para que H sea definida positiva. Con este fin redefinimos H como

$$H = W_1 + W_2 + W_3 + W_4 \tag{A.130}$$

donde:

$$W_1 = \frac{1}{8} \dot{q}^T M(q, \theta) \dot{q} + \frac{1}{8} e^T K'_p e + \varepsilon e^T M(q, \theta) \dot{q} \tag{A.131}$$

$$W_2 = \frac{1}{8} e^T K'_p e + P(q, \theta) + e^T K'_p (q_d - \delta) + \frac{1}{2} (q_d - \delta)^T K'_p (q_d - \delta) + c_1 \tag{A.132}$$

$$W_3 = \frac{1}{8} \dot{q}^T M(q, \theta) \dot{q} + \frac{1}{4} \vartheta^T K_d B^{-1} \vartheta + \varepsilon \vartheta^T M(q, \theta) \dot{q} \tag{A.133}$$

$$W_4 = \frac{1}{4} e^T K'_p e + \frac{\varepsilon}{2} z^T K_i^{-1} z^T + \frac{1}{4} \vartheta^T K_d B^{-1} \vartheta + \frac{1}{4} \dot{q}^T M(q, \theta) \dot{q} \quad (\text{A.134})$$

W_1 es definida positiva si se cumple la condición (A.109e). La condición (A.109f) garantiza que W_3 es definida positiva. W_4 es definida positiva. Queda ver si W_2 es definida positiva. Para ello sustituimos δ y c_I :

$$W_2 = \frac{1}{8} e^T K'_p e + P(q, \theta) - e^T G(q_d, \theta) - P(q_d, \theta) \quad (\text{A.135})$$

Podemos ver ahora que guarda un cierto parecido con

$$V(q, q_d) - V(q_d, q_d) = \frac{1}{2} e^T K'_p e + P(q, \theta) - e^T G(q_d, \theta) - P(q_d, \theta) \quad (\text{A.136})$$

Basándonos en ello definimos la función auxiliar

$$V_1(q, q_d) = P(q, \theta) - q^T G(q_d, \theta) + \frac{1}{8} q^T K'_p q - \frac{1}{4} q^T K'_p q_d \quad (\text{A.137})$$

que, siguiendo un análisis similar al realizado para la función V , tiene un mínimo en q_d si se cumple la condición (A.109c). Por tanto, W_2 es definida positiva bajo la condición (A.109c).

Ahora veremos condiciones suficientes para que \dot{H} sea localmente semidefinida negativa para e, \dot{q}, ϑ . Por (A.109g) y (A.109h) $Q_1 > 0$ y $Q_2 > 0$ respectivamente, y por tanto los dos primeros términos de (A.129) son semidefinidos negativos. El tercer término es negativo si se cumple (A.109i) y el cuarto cuando se cumple (A.109h). Por tanto (A.129) es semidefinida negativa localmente. Aplicando el teorema de invarianza de Lasalle se obtiene estabilidad asintótica local.

Dominio

Para definir el dominio tendremos que encontrar algunas constantes positivas P_m, P_M tal que se cumpla la siguiente desigualdad:

$$P_m \|x\|^2 \leq H(x) \leq P_M \|x\|^2 \quad (\text{A.138})$$

Se puede ver que

$$H \geq W_4 \geq \frac{1}{4} \left[K'_{p,m} \|e\|^2 + \lambda_m(K_d B^{-1}) \|\vartheta\|^2 + M_m \|\dot{q}\|^2 + 2 \varepsilon K_{i,m}^{-1} \|z\|^2 \right] \quad (\text{A.139})$$

Para obtener una cota inferior en términos de x necesitamos la siguiente desigualdad:

$$\left[\frac{1 - K_{i,M}}{\varepsilon} \right] \|y\|^2 + \left[\frac{K_{i,m}^2}{\varepsilon^2} - \frac{K_{i,M}}{\varepsilon} \right] \|e\|^2 \leq \|z\|^2 \quad (\text{A.140})$$

Lo cual nos lleva a lo siguiente:

$$V \geq \frac{1}{4} \left\{ \left[K'_{p,m} + 2 \frac{K_{i,m}^2}{\varepsilon K_{i,M}} - 2 \right] \|e\|^2 + \lambda_m(K_d B^{-1}) \|\vartheta\|^2 \right\} + \frac{1}{4} \left\{ M_m \|\dot{q}\|^2 + 2 \left[\frac{\varepsilon}{K_{i,M}} - 1 \right] \|y\|^2 \right\} \quad (\text{A.141})$$

de esta forma definiremos P_m como:

$$P_m = \frac{1}{4} \min \left\{ K'_{p,m} + \left[2 \frac{K_{i,m}^2}{\varepsilon K_{i,M}} - 1 \right], \lambda_m(K_d B^{-1}), M_m, 2 \left[\frac{\varepsilon}{K_{i,M}} - 1 \right] \right\} \quad (\text{A.142})$$

De forma similar, podemos dar una cota superior a H:

$$H \leq \left\{ \frac{1}{4} [K'_{p,m} + 2k_g] + \frac{1}{2} \left[\frac{\varepsilon}{2} M_M + K'_{p,M} \right] \right\} \|e\|^2 + \left[\left(\varepsilon + \frac{1}{2} \right) M_M \right] \|\dot{q}\|^2 + \frac{1}{2} \left[\varepsilon M_M + \lambda_M(K_d B^{-1}) \right] \|\vartheta\|^2 + \frac{\varepsilon}{2} K_{i,M}^{-1} \|z\|^2 \quad (\text{A.143})$$

y utilizando la expresión:

$$\|z\|^2 \leq \left[1 + \frac{K_{i,M}}{\varepsilon} \right] \left[\|y\|^2 + \frac{K_{i,M}}{\varepsilon} \|e\|^2 \right] \quad (\text{A.144})$$

podremos definir P_M como:

$$P_M \leq \max \left\{ \frac{1}{4} [K'_{p,m} + 2k_g] + \frac{1}{2} \left[\frac{\varepsilon}{2} M_M + K'_{p,M} \right] + \left[\frac{1}{2} \left(\frac{K_{i,M}}{\varepsilon} + 1 \right) \right], \left(\varepsilon + \frac{1}{2} \right) M_M, \frac{1}{2} [\varepsilon M_M + \lambda_M(K_d B^{-1})], \frac{1}{2} (\varepsilon K_{i,M}^{-1} + 1) \right\} \quad (\text{A.145})$$

de las ecuaciones (A.122i) y (A.151) concluimos que el dominio viene dado por:

$$\|x\| \leq c_3 \frac{1}{2C_M} \left[\frac{1}{2} B_m M_m - M_M \right] \sqrt{\frac{P_m}{P_M}} \quad (\text{A.146})$$

Estabilidad Semiglobal

Para establecer la estabilidad semiglobal, deberemos probar primero que con una adecuada elección de las ganancias del controlador, podemos ampliar arbitrariamente el dominio. Para ello, se propone incrementar B_m teniendo en cuenta las relaciones entre B y \mathcal{E} impuestas por las condiciones de estabilidad.

Estas condiciones de estabilidad fuerzan las siguientes relaciones de orden:

$$\mathcal{E}(B_m) = O(B_m^{-1/2}) \quad (\text{A.147})$$

$$\mathcal{E}(B_m) = O(B_m^{-1}) \quad (\text{A.148})$$

si tenemos en cuenta el hecho de que:

$$\lambda_m(K_d B^{-1}) \geq K_{d,m} B_M^{-1} \quad (\text{A.149})$$

Por otro lado para un B_m suficientemente grande, podemos encontrar $\mathcal{E} > 0$ tal

que $P_m = \frac{c_4}{B_m}$ y $P_M = \frac{c_5}{\mathcal{E}}$ donde c_4 y c_5 son constantes independientes de B .

Reemplazando lo anterior en la ecuación (A.146) obtendremos:

$$\lim_{B_m \rightarrow \infty} c_3^2 = c_6 \left[\mathcal{E} \sqrt{B_m} \right] \sqrt{B_m} \quad (\text{A.150})$$

donde c_6 es también independiente de B .

Con esto probamos que las condiciones de estabilidad se cumplen verificando $\mathcal{E} = O(B_m^{-1/2})$ y además que el dominio puede ser arbitrariamente ampliado, es decir:

$$\lim_{B_m \rightarrow \infty} c_3 = \infty.$$

La demostración se completa escogiendo para un ε dado, $K_{i,M} = O(\varepsilon)$ de forma que

$$I > \frac{1}{\varepsilon} K_i$$

A.2.8.1. Resultados reales de las ejecuciones para una trayectoria de Fourier

Tabla A-29 Comparación de Referencia y Posición para las 6 articulaciones del controlador PI²D
(abcisas: posición (rad); ordenadas: períodos de muestreo (Ts=0.005s))

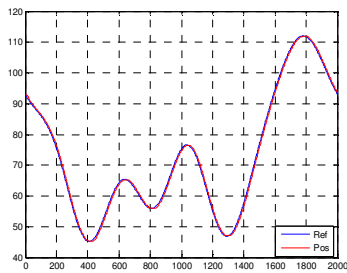


Figura A-169 Articulación 1

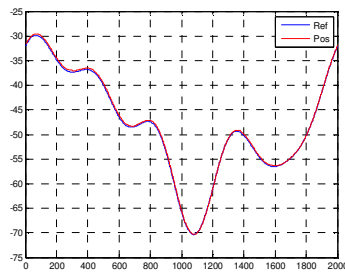


Figura A-170 Articulación 2

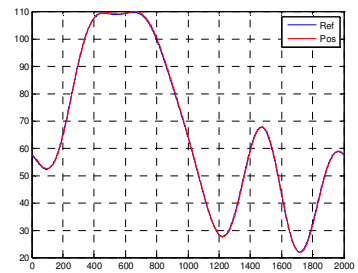


Figura A-171 Articulación 3

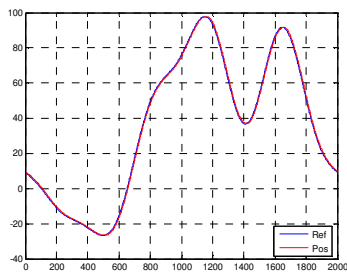


Figura A-172 Articulación 4

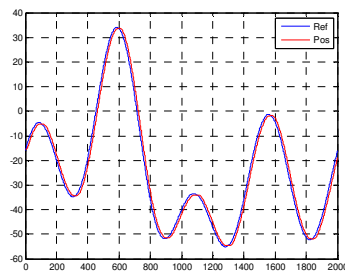


Figura A-173 Articulación 5

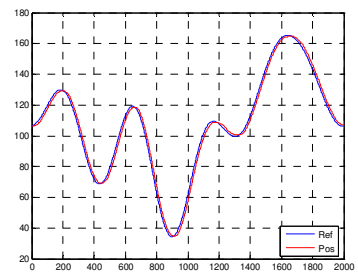


Figura A-174 Articulación 6

Tabla A-30 Acciones de Control del controlador PI²D
(abcisas: acción de control (Nm); ordenadas: períodos de muestreo (Ts=0.005s))

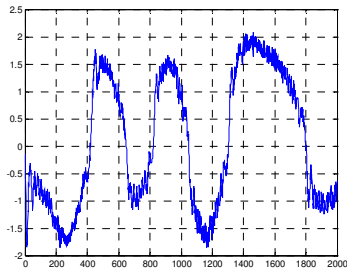


Figura A-175 Articulación 1

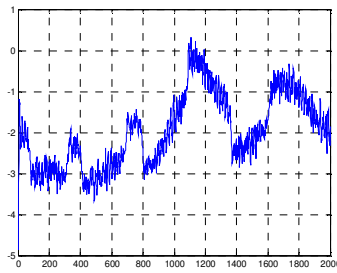


Figura A-176 Articulación 2

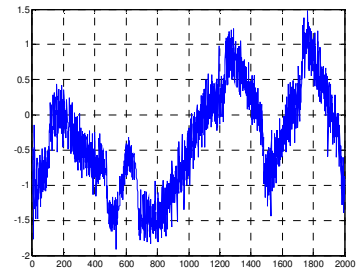


Figura A-177 Articulación 3

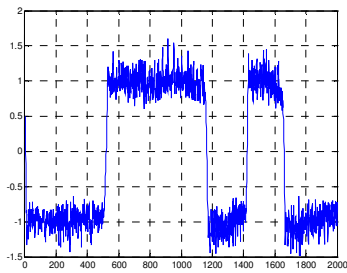


Figura A-178 Articulación 4

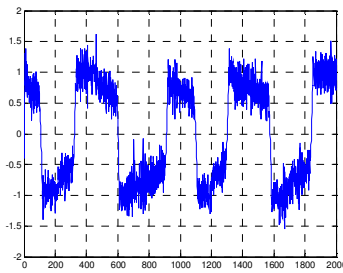


Figura A-179 Articulación 5

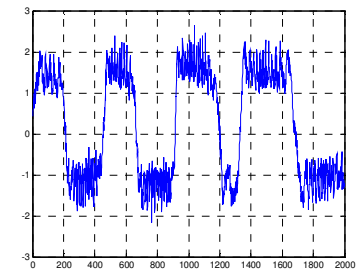


Figura A-180 Articulación 6

A.3. HERRAMIENTAS PARA EL ANÁLISIS DE LA ESTABILIDAD

Corilario A.1

Si $g \in L_\infty$, $\dot{g} \in L_\infty$ y $g \in L_p$ para algún $p \in [1, \infty)$, entonces

$$\lim_{t \rightarrow \infty} g(t) = 0$$

Teorema A.1

Sea $H(s) \in R^{n \times n}(s)$ una función de transferencia exponencialmente estable y estrictamente propia. Sea $x=H(s)u$. Entonces

Si $u \in L_2$, entonces $x \in L_2 \cap L_\infty$, x es continua, $x(t) \rightarrow 0$ y $t \rightarrow \infty$

Si $u \in L_\infty$, entonces $x \in L_\infty$, $\dot{x} \in L_\infty$, x es uniformemente continua

(Demostración en (Desoer, Vidyasagar, 75

Teorema A.2

Supongamos que una matriz simétrica está particionada como:

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$$

donde A y C son matrices cuadradas

Esta matriz es definida positiva si y sólo si

$$A > 0$$

$$C-B^T A^{-1} B > 0$$

Teorema A.3 (Matrosov)

Sea $\Omega \in \mathbb{R}^n$ un abierto conexo en \mathbb{R}^n que contiene el origen. Si existen dos funciones C^1 $V: [t_0, \infty) \times \Omega \rightarrow \mathbb{R}$, $W: [t_0, \infty) \times \Omega \rightarrow \mathbb{R}$; una función C^0 $V^*: \Omega \rightarrow \mathbb{R}$; tres funciones a, b, c de clase K tales que para todo $(x, t) \in [t_0, \infty) \times \Omega$,

$$(1) \quad a(\|x\|) \leq V(t, x) \leq b(\|x\|)$$

$$(2) \quad \dot{V}(t, x) \leq V^*(x) \leq 0. \text{ Se define } E = \{x \in \Omega \mid V^*(x) = 0\}$$

$$(3) \quad |W(t, x)| \text{ está acotada}$$

$$(4) \quad \max(d(x, E), |\dot{W}(t, x)|) \geq c(\|x\|)$$

$$(5) \quad \|f(t, x)\| \text{ está acotada.}$$

Escogiendo $\alpha > 0$ tal que $\bar{B}_\alpha \subset \Omega$, se define para todo $t \in [t_0, \infty)$

$$V_{t, \alpha}^{-1} = \{x \in \Omega: V(t, x) \leq a(\alpha)\}$$

Entonces

Para todo $x_0 \in V_{t_0}^{-1}$, $x(t)$ tiende a cero uniformemente en t_0 , x_0 tiende a ∞

El origen es uniforme y asintóticamente estable

La demostración de este teorema se puede encontrar en (Rouche, Habets, Laloy, 77)

Teorema A.4 (Teoremas del Conjunto Invariante de LaSalle)

Definición

Un conjunto G es un *conjunto invariante* para un sistema dinámico si cada trayectoria del sistema que comienza en un punto de G permanece en G para todo el tiempo futuro.

Teorema del Conjunto Invariante Local

Consideremos el sistema autónomo de la forma:

$$\dot{x} = f(x)$$

con f continua, y sea $V(x)$ una función escalar con primeras derivadas parciales continuas. Asumimos que para cualquier $r > 0$, la región Ω_r definida por $V(x) < r$ está acotada

$$\dot{V}(x) \leq 0 \text{ para todo } x \text{ en } \Omega_r.$$

Sea R el conjunto de todos los puntos de Ω_r donde $\dot{V}(x) = 0$, y M el conjunto invariante más grande de R . Entonces, cada solución $x(t)$ originada en Ω_r tiende a M cuando t tiende a infinito.

Corolario

Consideremos el sistema autónomo $\dot{x} = f(x)$, con f continua, y sea $V(x)$ una función escalar con derivadas parciales continuas. Asumimos que en un cierto entorno Ω del origen se cumple:

$V(x)$ es localmente positiva definida

\dot{V} es negativa semidefinida

El conjunto R definido por $\dot{V}(x) = 0$ no contiene trayectorias de $\dot{x} = f(x)$ que no sea la trayectoria trivial $x = 0$

Entonces, el punto de equilibrio 0 es asintóticamente estable. Además, la región conectada más grande de la forma Ω_r (definida por $V(x) < r$) dentro de Ω es un dominio de atracción del punto de equilibrio.

Este corolario implica que el más grande conjunto M en R contiene sólo el punto de equilibrio 0 . Notemos que:

El corolario anterior reemplaza la condición de negativa definida de \dot{V} en el teorema de estabilidad asintótica local de Lyapunov por una condición de negativa semidefinida combinada con la tercera condición de las trayectorias de R .

La región conectada más grande de la forma Ω_r dentro de Ω es un dominio de atracción del punto de equilibrio, pero no necesariamente la totalidad del espacio de estados ya que V no es única.

El conjunto Ω en sí mismo no es un dominio de atracción. De momento, el teorema anterior no nos garantiza que Ω sea invariante: algunas trayectorias que comienzan en Ω pero que salen de Ω_r pueden terminar fuera de Ω .

Teorema del Conjunto Invariante Global

Consideremos el sistema autónomo $\dot{x} = f(x)$, con f continua, y sea $V(x)$ una función escalar con primeras derivadas parciales continuas. Asumimos que

$$\dot{V}(x) \leq 0 \text{ para todo el espacio de estados}$$

$$V(x) \rightarrow \infty \text{ cuando } \|x\| \rightarrow \infty$$

Sea R un conjunto de puntos donde $\dot{V}(x) = 0$, y M el más grande conjunto invariante de R . Entonces, todas las soluciones globalmente asintóticamente estables convergen a M cuando t tiende a infinito.

Para un sistema dado pueden existir varias funciones de Lyapunov y por tanto, varios conjuntos invariantes. El sistema entonces converge a la intersección de los conjuntos invariantes M_i que nos da un resultado más preciso que los conjuntos por separado. De forma equivalente, la suma de dos funciones de Lyapunov es otra función de Lyapunov cuyo conjunto R es la intersección de los R_i individuales.

Lema A.1

Sean N , P y Q matrices simétricas definidas positivas y ε un número real positivo.

Si

$$N - P - \frac{1}{\varepsilon} Q > 0 \tag{A.151}$$

entonces las matrices

$$1 \ N - P$$

$$2 \ \varepsilon Q - Q(N - P) - I Q$$

son definidas positivas.

Lema A.2 (Barbalat)

Para sistemas autónomos, los teoremas del conjunto invariante son una herramienta poderosa para estudiar la estabilidad ya que nos permite establecer conclusiones sobre la estabilidad asintótica incluso cuando la derivada de V es sólo negativa semidefinida. Sin embargo, los teoremas del conjunto invariante no son aplicables a sistemas no autónomos. Para solucionar esto utilizaremos el lema de Barbalat. Este lema es un resultado puramente matemático referente a las propiedades asintóticas de las funciones y sus derivadas.

Lema de Barbalat.

Si una función diferenciable $f(t)$ tiene límite finito cuando $t \rightarrow \infty$, y \dot{f} es uniformemente continua, entonces $\dot{f}(t) \rightarrow 0$ cuando $t \rightarrow \infty$.

Corolario

Si la función diferenciable $f(t)$ tiene límite finito cuando $t \rightarrow \infty$, y además \ddot{f} existe y está acotada, entonces, $\dot{f}(t) \rightarrow 0$ cuando $t \rightarrow \infty$.

Para aplicar el lema de Barbalat al análisis de sistemas dinámicos, generalmente utiliza el siguiente lema que se asemeja bastante al teorema del conjunto invariante del análisis de Lyapunov:

Lema

Si una función escalar $V(x,t)$ satisface las siguientes conclusiones:

$V(x,t)$ está acotada inferiormente.

$\dot{V}(x,t)$ es negativa semidefinida.

- 244 - *Estudio e implementación de controladores dinámicos de robots industriales en tiempo real*

$\dot{V}(x, t)$ es uniformemente continua en el tiempo,

entonces $\dot{V}(x, t) \rightarrow 0$ cuando $t \rightarrow \infty$.

ANEXO B

GENERACIÓN AUTOMÁTICA DE ALGORITMOS DE CONTROL DE ROBOTS

B.1. GENERACIÓN AUTOMÁTICA DE CÓDIGO DE ESQUEMAS DE SIMULACIÓN

Como se ha comentado en el capítulo 5 de la presente tesis, a la hora de realizar las pruebas sobre el modelo del PUMA 560 o bien sobre el propio PUMA, se ha utilizado la herramienta Real-Time Workshop de MATLAB, en concreto, se han implementado Embedded MATLAB Functions que nos ha permitido añadir funciones de MATLAB[®] para los modelos de Simulink[®].

La función en Matlab que se crea se ha ejecutado en simulaciones y ha permitido resolver tanto el Problema Dinámico Inverso como el Problema Dinámico Directo, por lo

que se ha podido tanto simular el comportamiento del PUMA 560 gracias a su modelo dinámico ya visto en capítulos anteriores.

Las siguientes figuras muestran diferentes esquemas Simulink que ha permitido simular el comportamiento del PUMA 560 con los distintos reguladores implementados en la presente tesis. En estos esquemas se pueden observar los bloques correspondientes a la obtención de los elementos de la ecuación general del movimiento tal y como se explicó en el capítulo 2.

Si se observa más detenidamente como se ha simulado el comportamiento del PUMA 560, vemos que se resuelto el problema dinámico directo, de forma que se obtiene la aceleración que posteriormente se integra para obtener la velocidad y volviéndose a integrar se obtendría la posición que permite realimentar el bucle de control. A continuación se muestra esta parte de los esquemas con más detalle y ejemplos realizados con ellos.

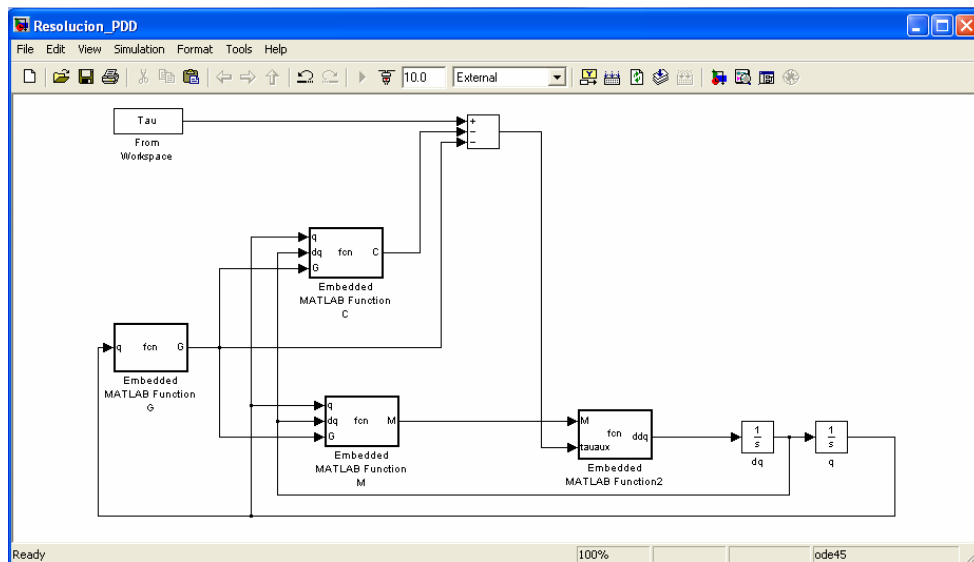


Figura B-1 Resolución del Problema Dinámico Directo en Simulink

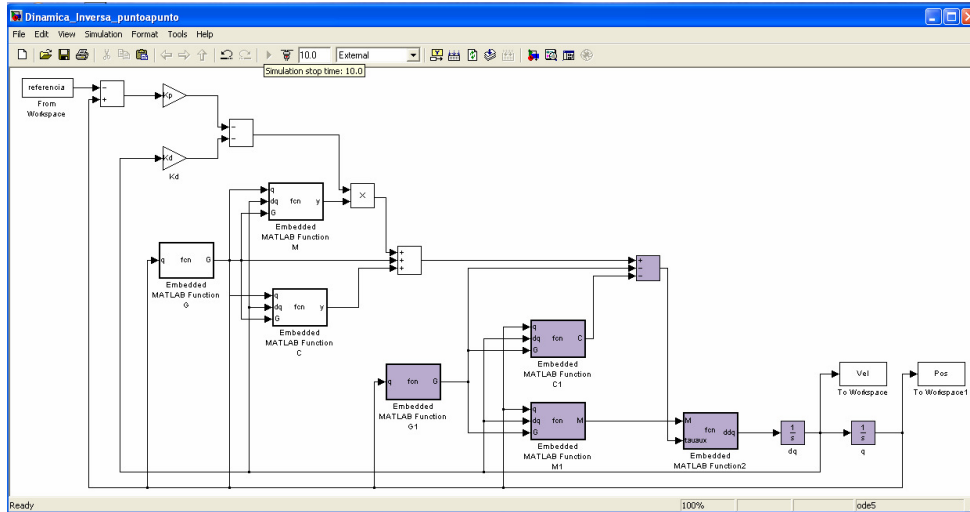


Figura B-2 Control por Dinámica Inversa Punto a Punto

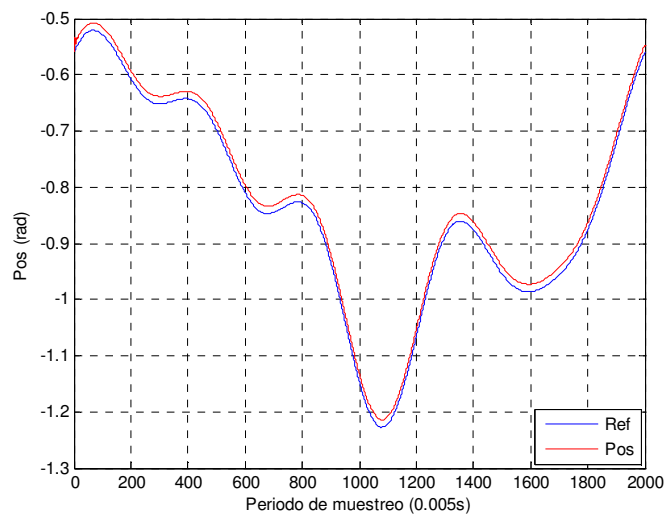


Figura B-3 Simulación del controlador por dinámica inversa punto a punto en la Articulación 1

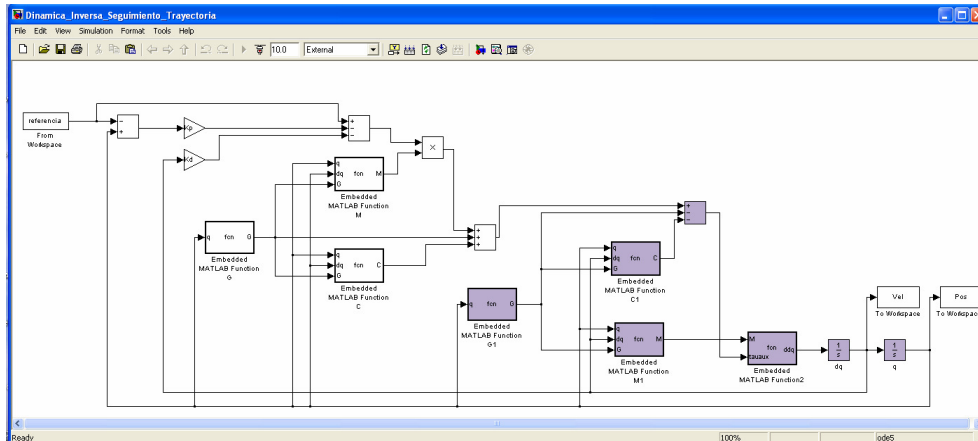


Figura B-4 Control por Dinámica Inversa con Seguimiento de la trayectoria

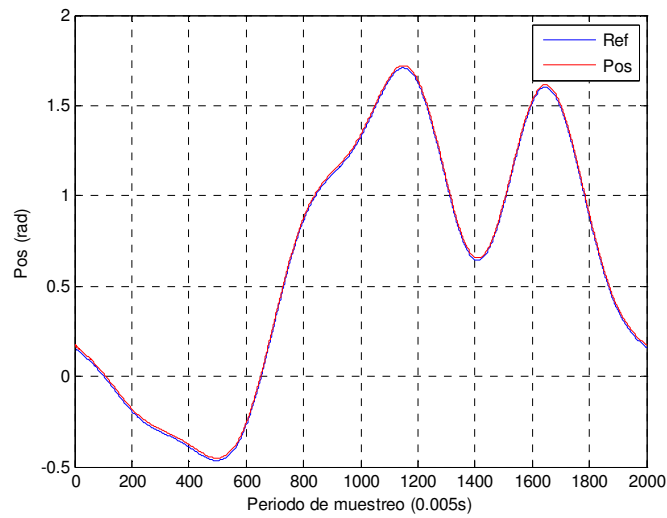


Figura B-5 Simulación del controlador por dinámica inversa con seguimiento de la trayectoria en la Articulación 1

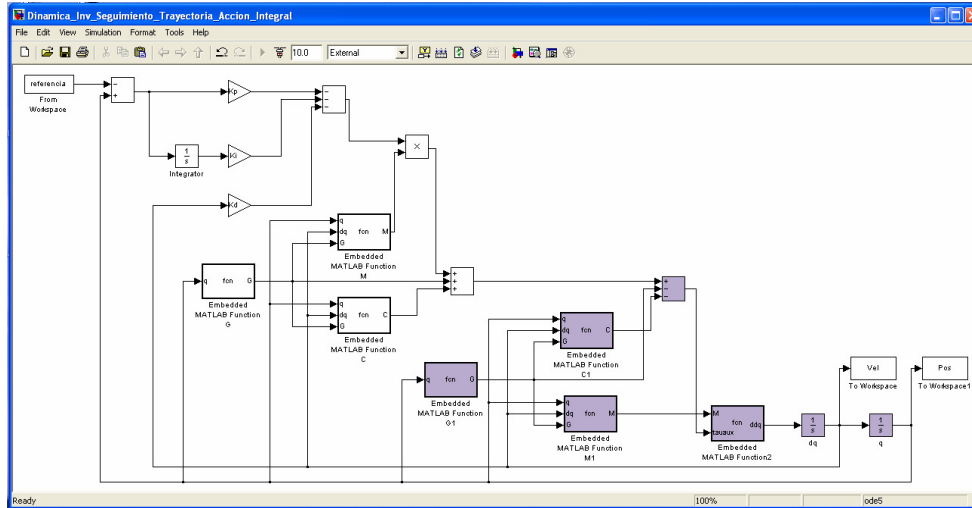


Figura B-6 Control por Dinámica Inversa con Seguimiento de la trayectoria con acción integral

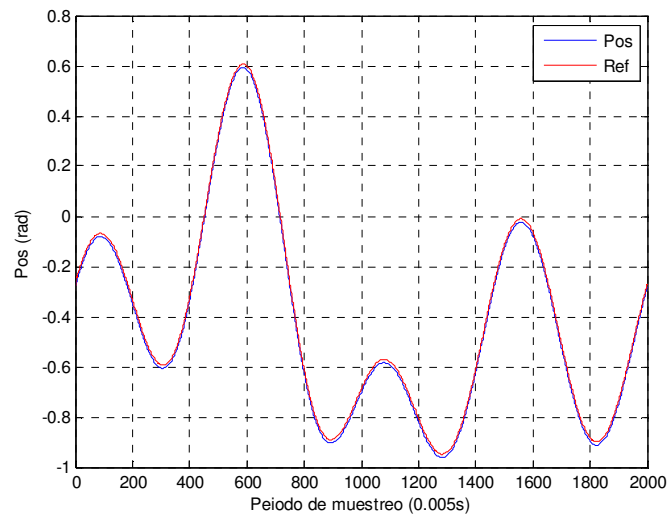


Figura B-7 Simulación del controlador por dinámica inversa con seguimiento de la trayectoria con acción integral en la Articulación 1

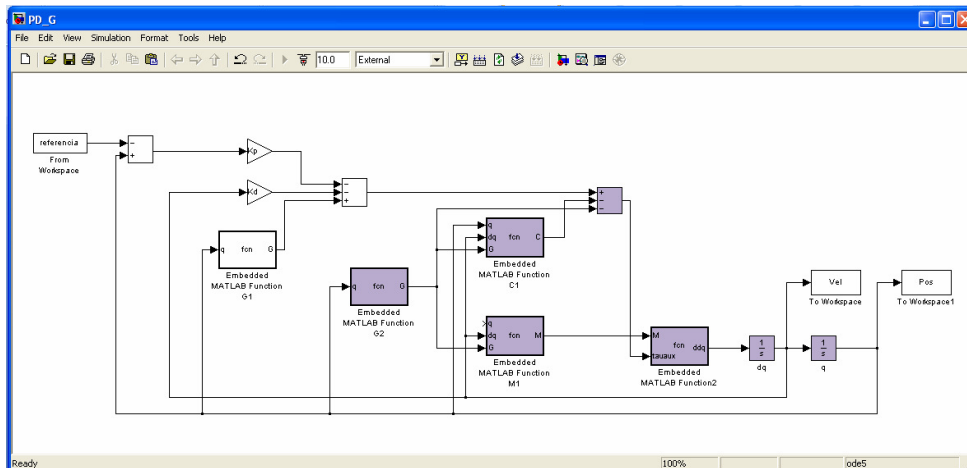


Figura B-8 Control por Pasividad. PD+G(q)

La siguiente figura muestra un ejemplo de simulación aplicando este controlador.

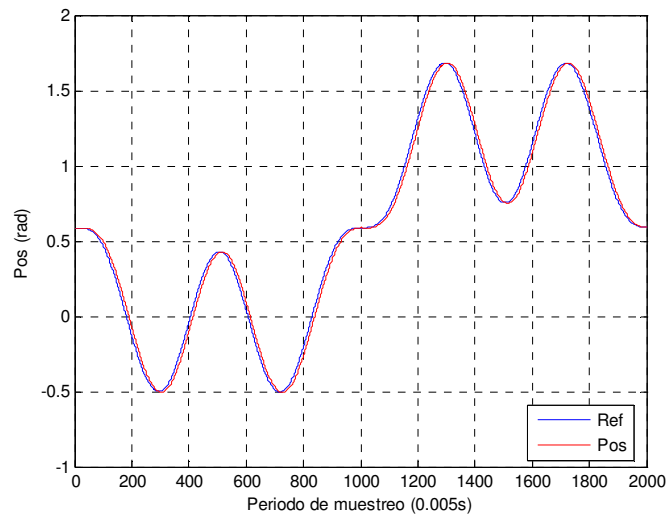


Figura B-9 Simulación del controlador PD+G(q) en la Articulación 1

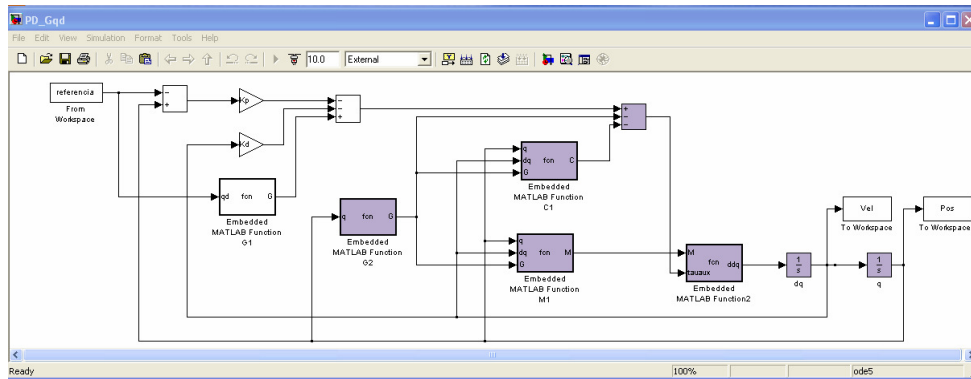


Figura B-10 Control por Pasividad. PD+G(qd)

Ejemplo de la simulación del controlador PD+G(qd)

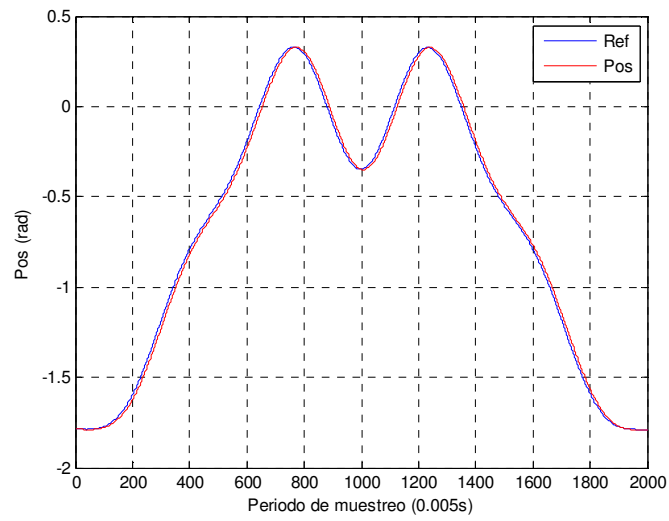


Figura B-181 Simulación del controlador PD+G(qd) en la Articulación 1

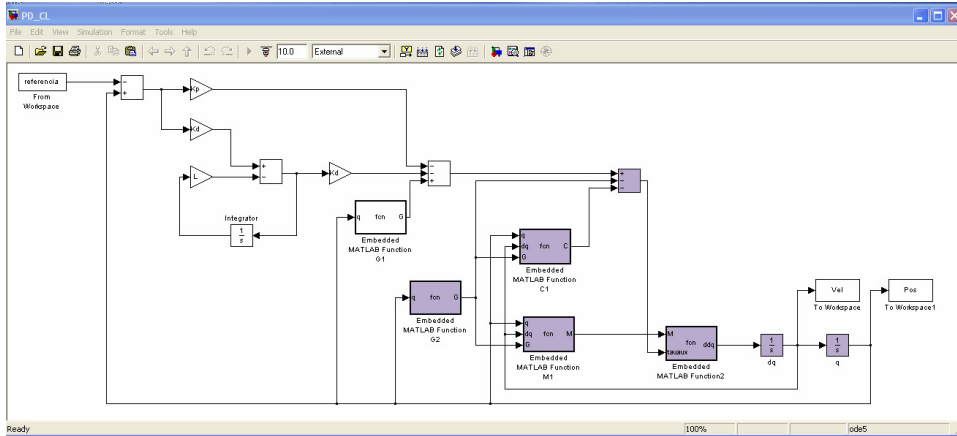


Figura B-12 Control por Pasividad. PD+CL(q)

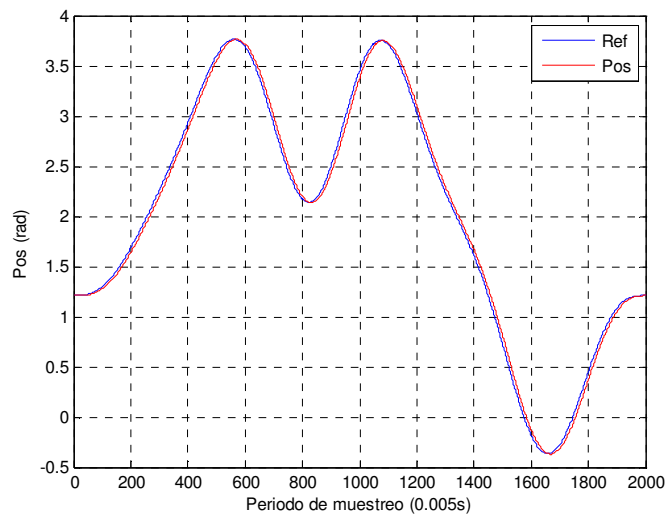


Figura B-13 Simulación del controlador PD CL(q) en la Articulación 1

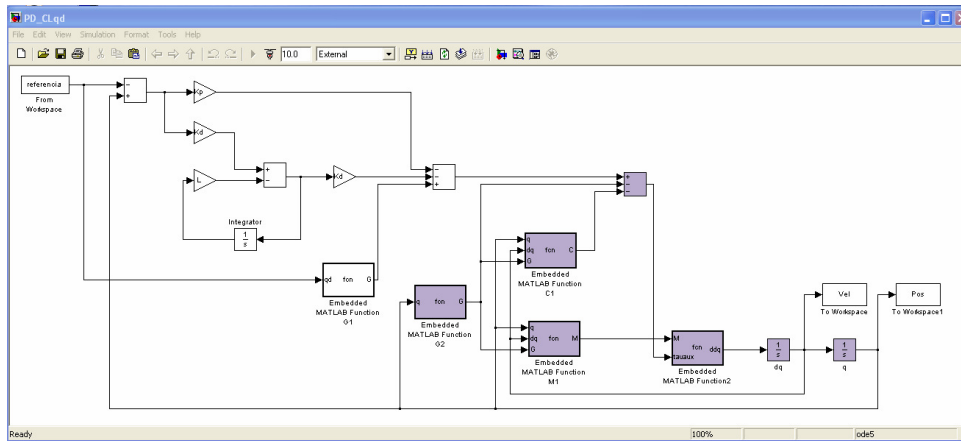


Figura B-14 Control por Pasividad. PD+CL(qd)

A continuación se muestra un ejemplo realizado sobre la Articulación 1.

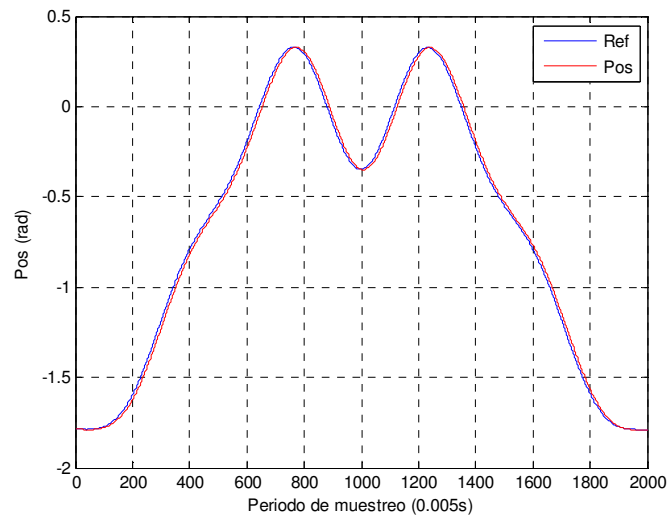


Figura B-15 Simulación del controlador PD CL(qd) en la Articulación 1

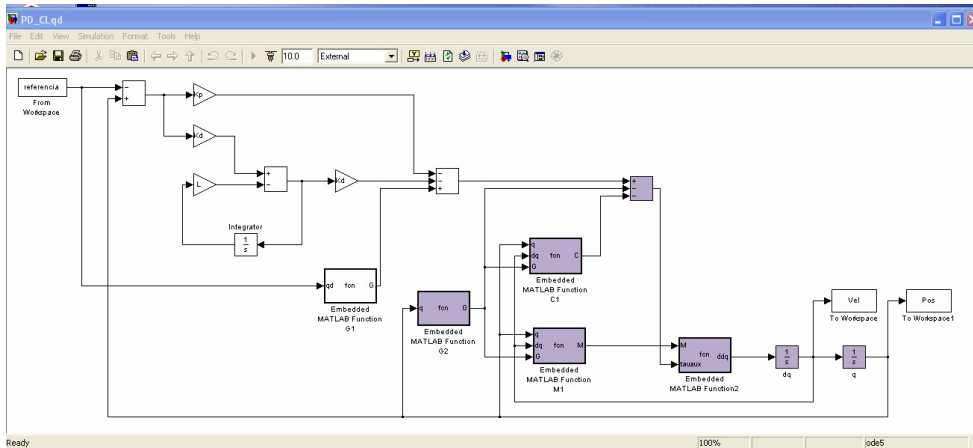


Figura B-16 Control por Pasividad. PD+DA(q)

Ejemplo de la articulación 1

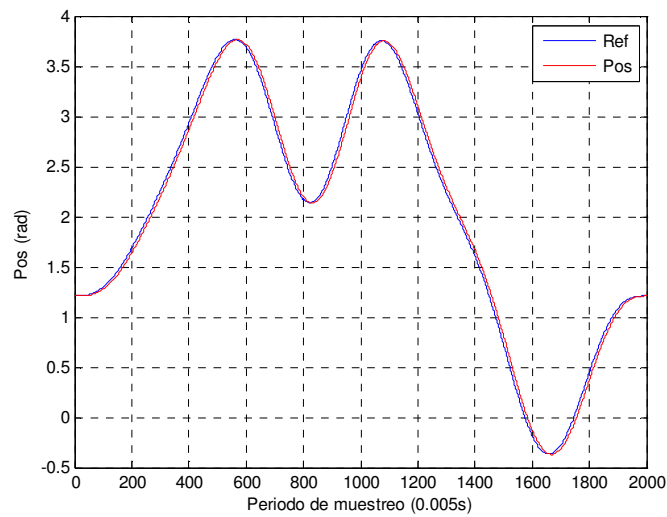


Figura B-17 Simulación del controlador PD DA(q) en la Articulación 1

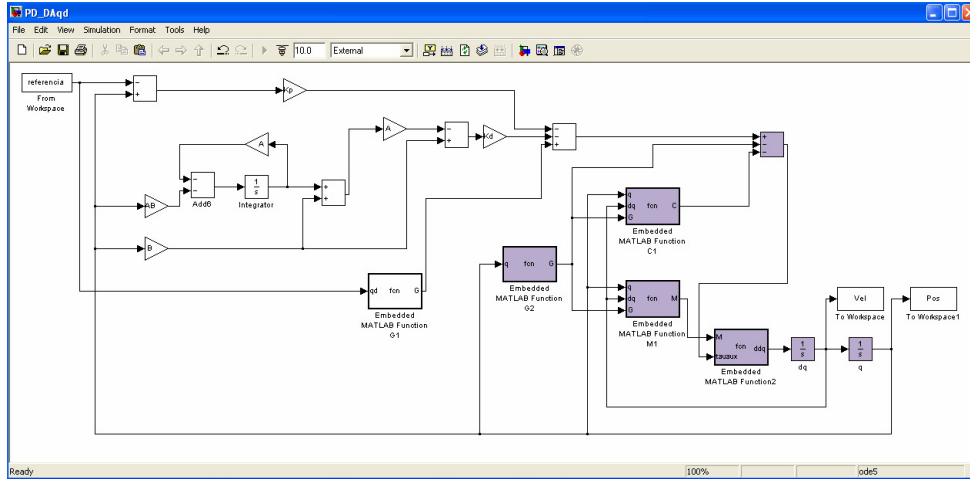


Figura B-18 Control por Pasividad. PD+DA(qd)

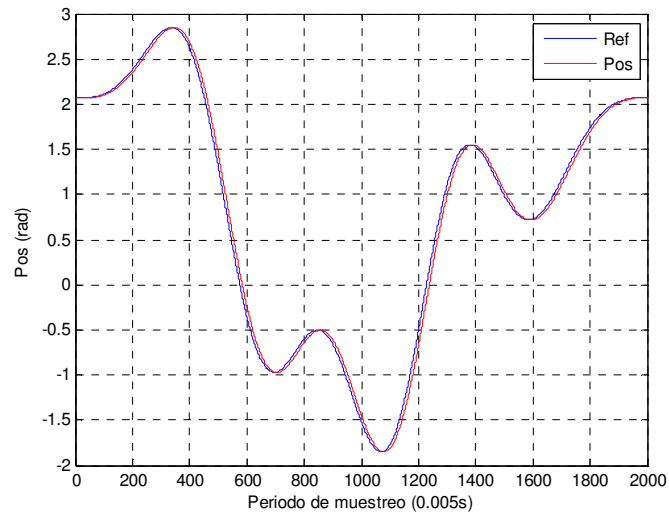


Figura B-182 Simulación del controlador PD DA(qd) en la Articulación 1

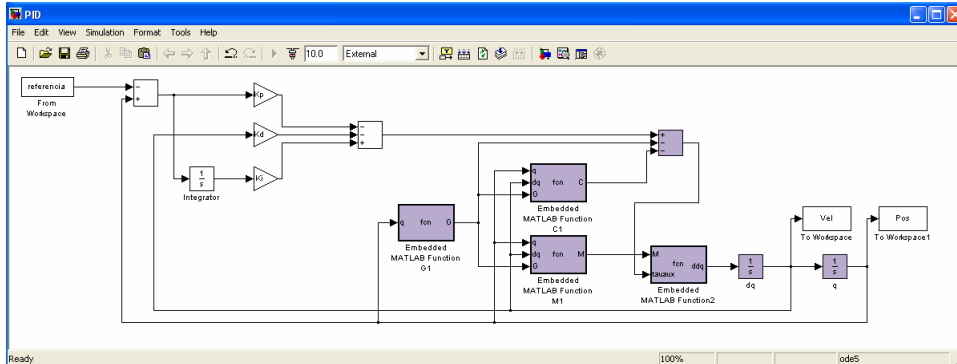


Figura B-20 Control por Pasividad. PID

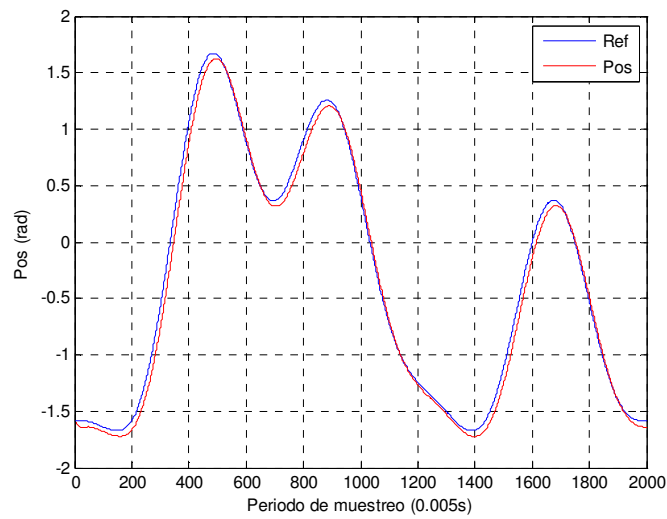


Figura B-1831 Simulación del controlador PID en la Articulación 1

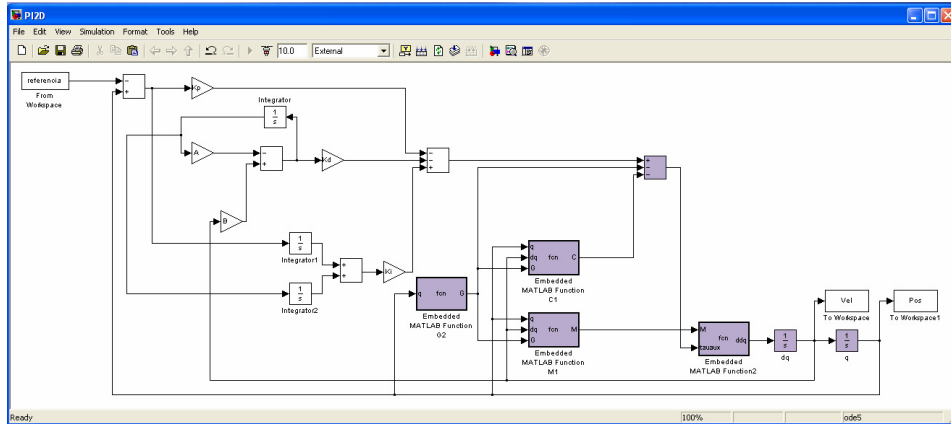


Figura B-1842 Control por Pasividad. PI^2D

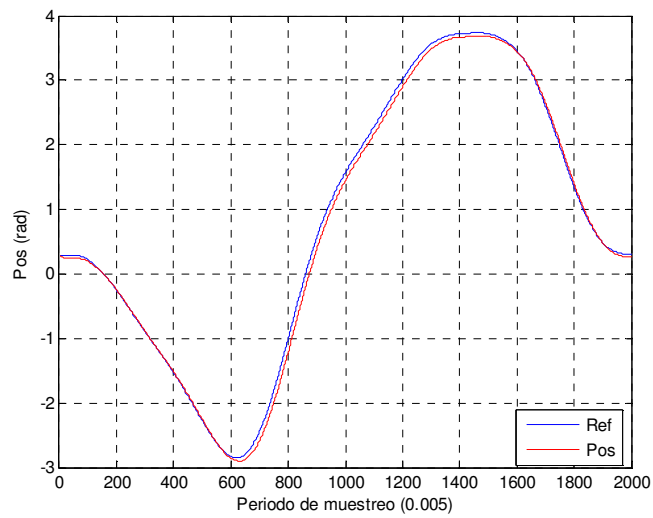


Figura B-1853 Simulación del controlador PI^2D en la Articulación 1

B.2. GENERACIÓN AUTOMÁTICA DE CÓDIGO EN LOS ESQUEMAS DE CONTROL

Una de las características de la Real-Time Workshop es su modo externo. Como ya se ha comentado en el capítulo 5, este modo externo permite que dos sistemas separados (un host y un target) puedan comunicarse. El host es el ordenador donde MATLAB® y Simulink® se están ejecutando. El target es el ordenador donde el ejecutable creado por la Real-Time Workshop® se ejecuta.

El modo externo ha permitido:

- Modificar, o ajustar los parámetros de los bloques en tiempo real.
- Visualizar e inspeccionar las salidas de los bloques en la mayoría de los tipos de bloques y subsistemas.

El modo externo trabaja estableciendo un canal de comunicación entre Simulink® y el código generado por la Real-Time Workshop®. Este canal se implementa mediante una capa de transporte a bajo nivel que maneja físicamente las transmisiones de los mensajes. Ambos, Simulink® y el código generado son independientes de esta capa.

Además también se han implementado S-functions para realizar el controlador de la tarjeta de adquisición de datos PCL-833, ya que, esta tarjeta no se encuentra entre las tarjetas predefinidas en las librerías de Simulink. Para realizar este nuevo bloque se ha implementado la S-function en lenguaje C que se empotra en el código del modelo que se va a generar con la RTW.

A continuación se muestran los diferentes modelos que se han realizado para poder realizar el control, con los distintos tipos de controladores implementados, sobre el PUMA 560. Si se observa la parte correspondiente al robot real, se puede ver el controlador de la tarjeta PCL-726 que permite aplicar las acciones de control y también

los bloques correspondientes a las dos tarjetas PCL-833 que leen los encoders del PUMA, lo que permite leer su posición que posteriormente se puede derivar para obtener la velocidad.

A continuación se muestra el detalle del acceso a las tarjetas de adquisición de datos que están conectadas al PUMA 560.

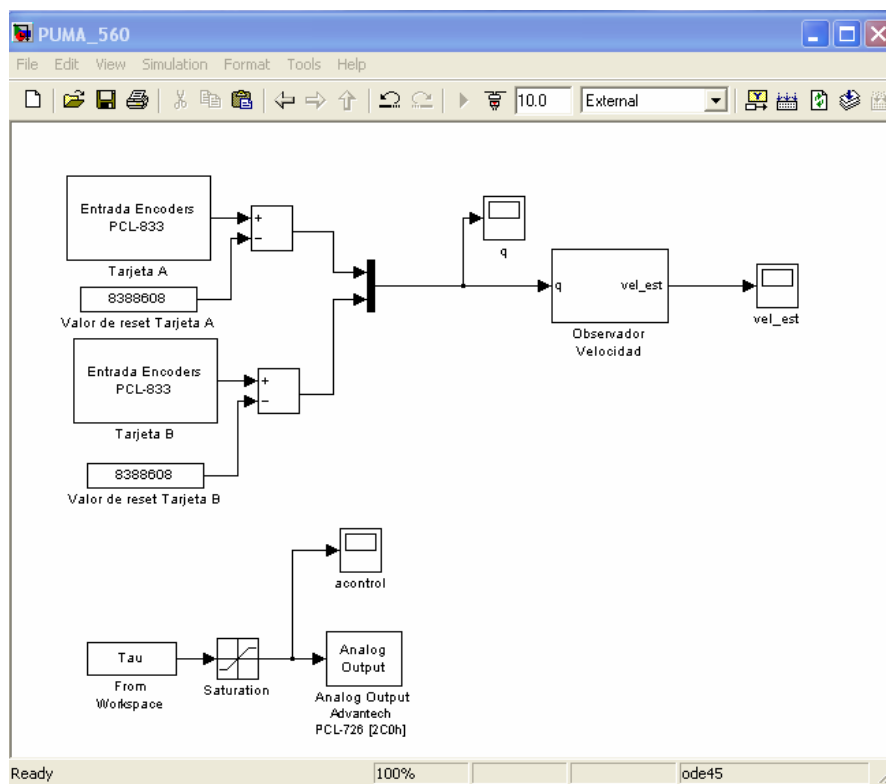


Figura B-24 Acceso al PUMA 560 desde Simulink

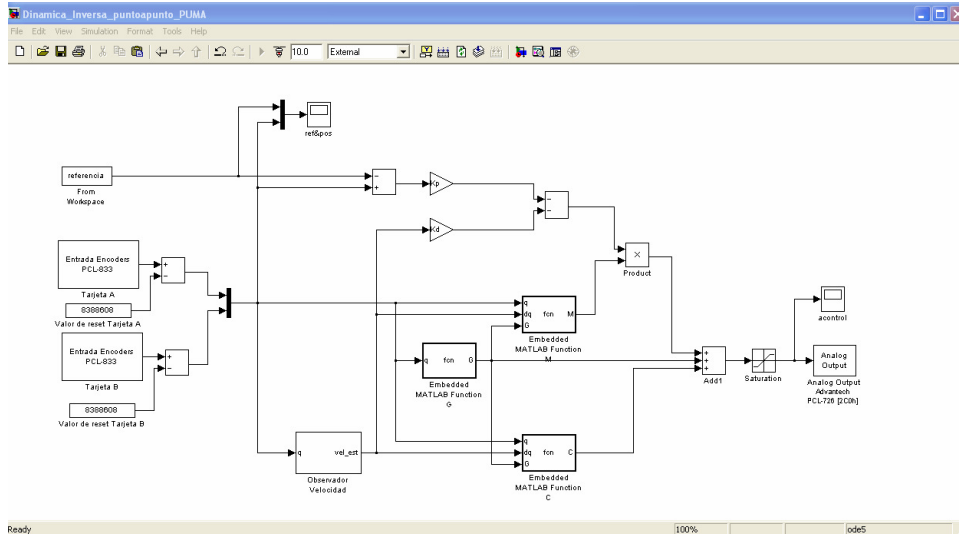


Figura B-1865 Control por dinámica inversa punto a punto sobre el PUMA 560

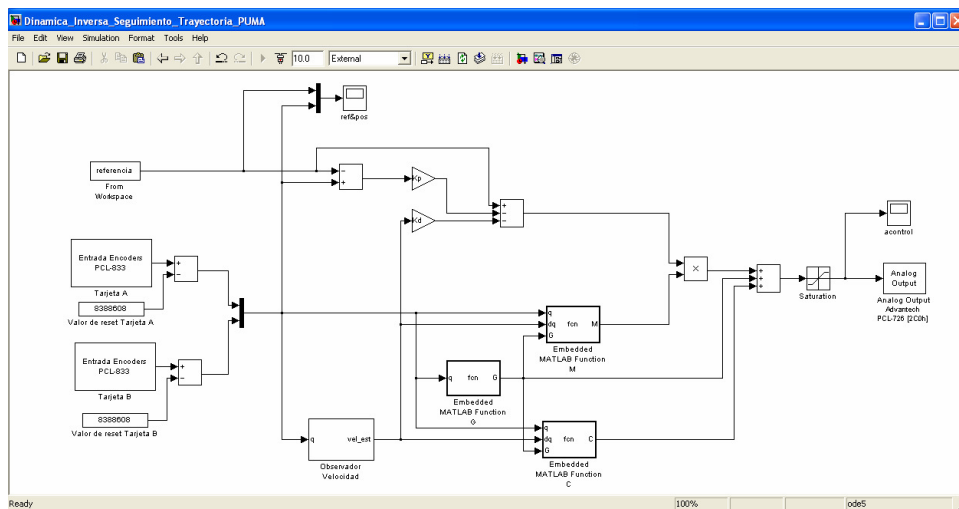


Figura B-1876 Control por dinámica inversa con seguimiento de la trayectoria sobre el PUMA 560

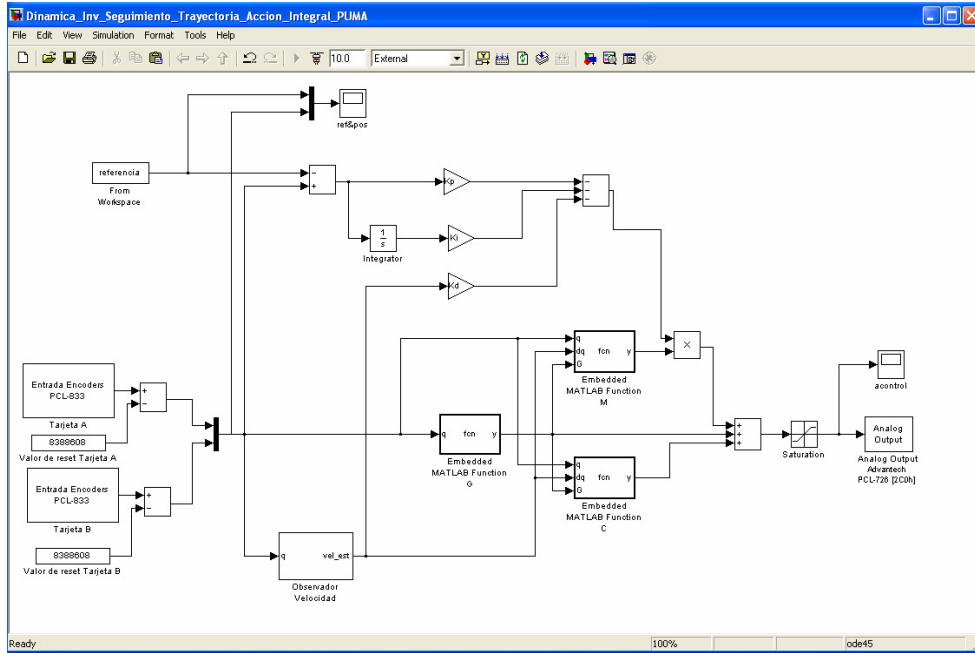


Figura B-1887 Control por dinámica inversa con seguimiento de la trayectoria con acción integral sobre el PUMA 560

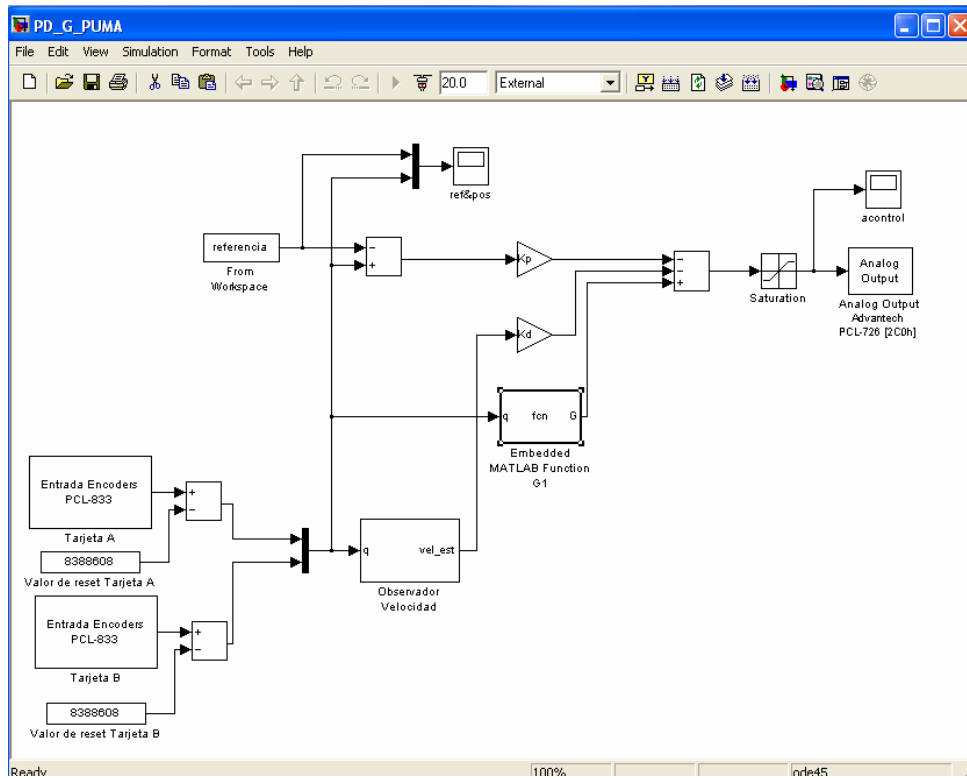


Figura B-1898 Control por pasividad PD+G(q) sobre el PUMA 560

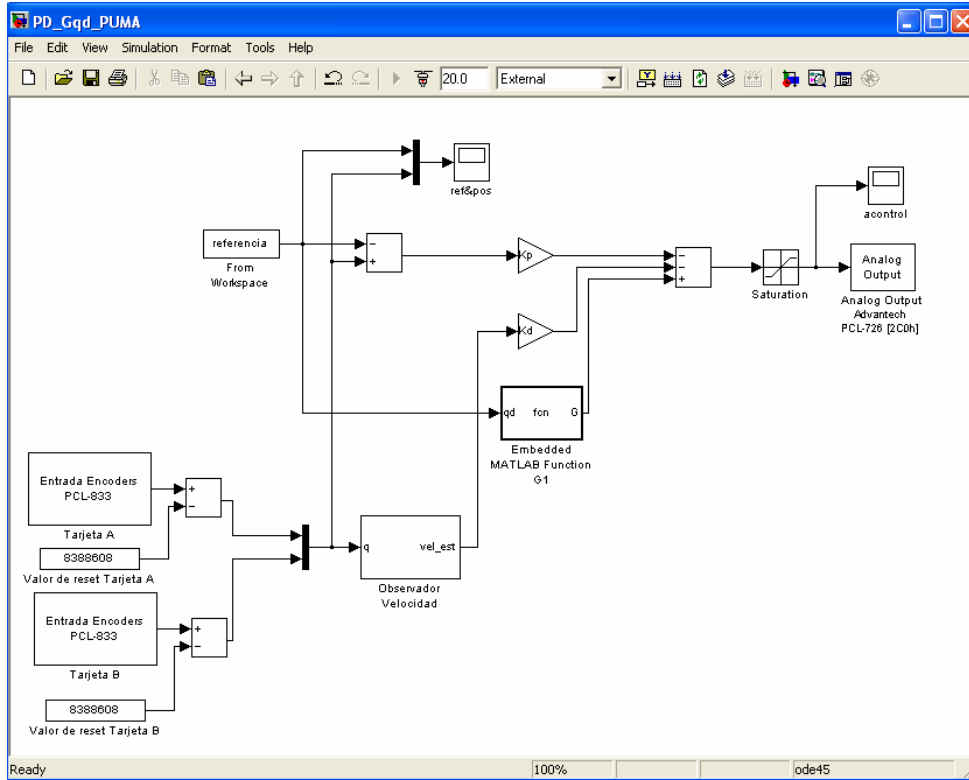


Figura B-1909 Control por pasividad PD+G(qd) sobre el PUMA 560

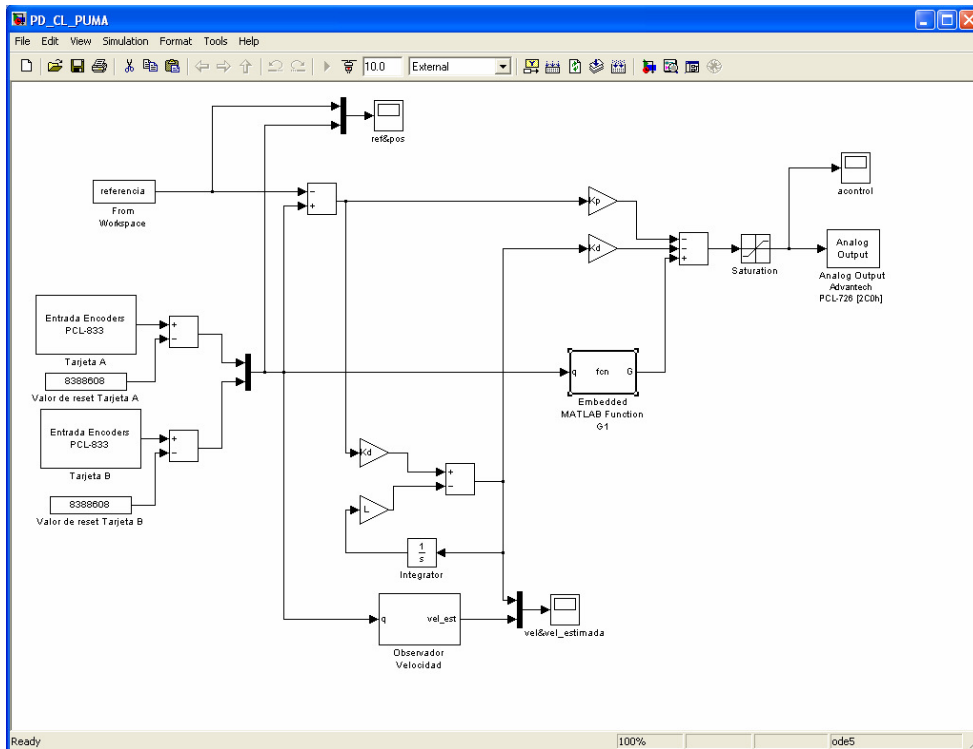


Figura B-30 Control por pasividad PD+CL(q) sobre el PUMA 560

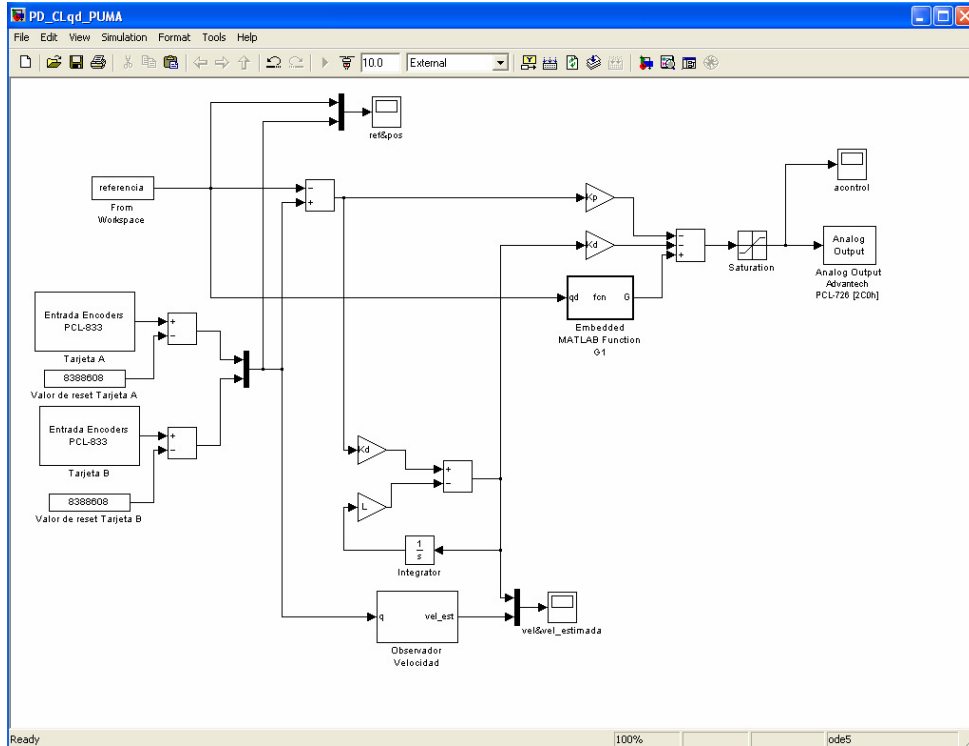


Figura B-31 Control por pasividad PD+CL(qd) sobre el PUMA 560

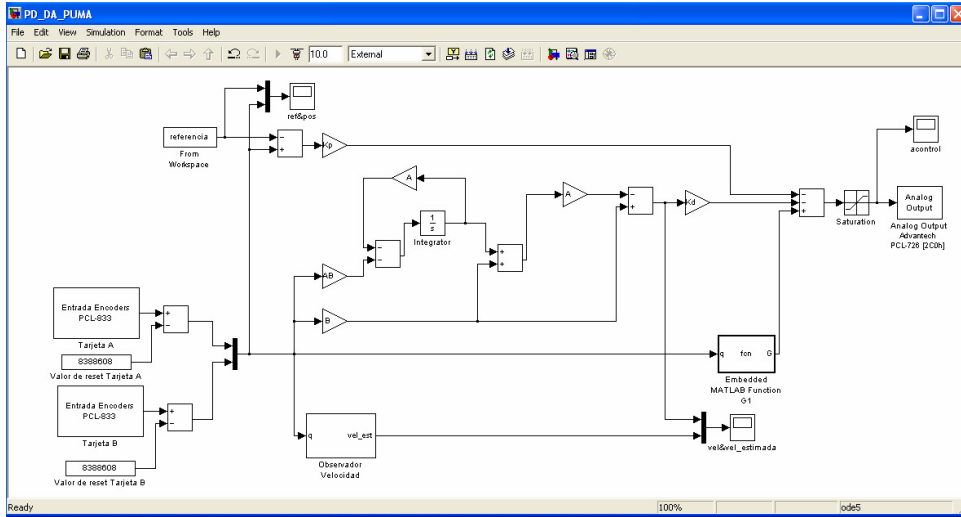


Figura B-32 Control por pasividad PD+DA(q) sobre el PUMA 560

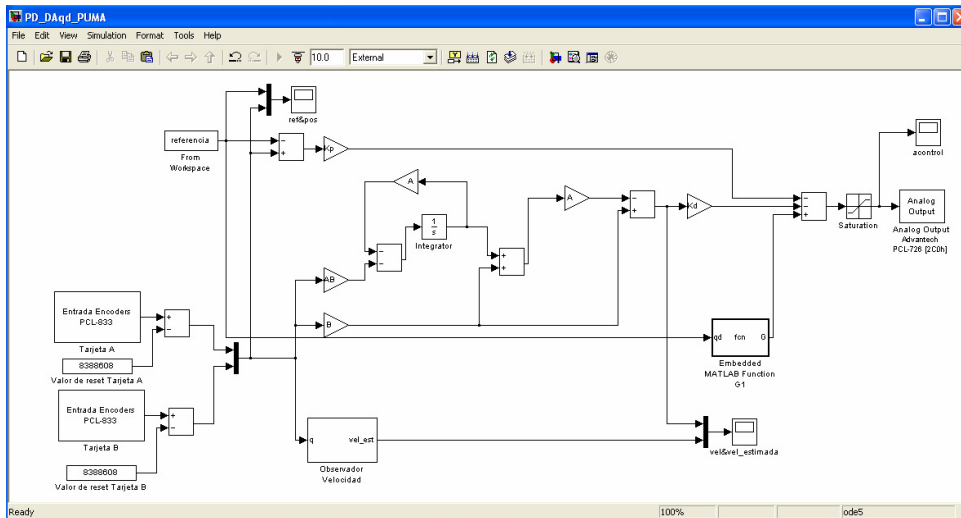


Figura B-33 Control por pasividad PD+DA(qd) sobre el PUMA 560

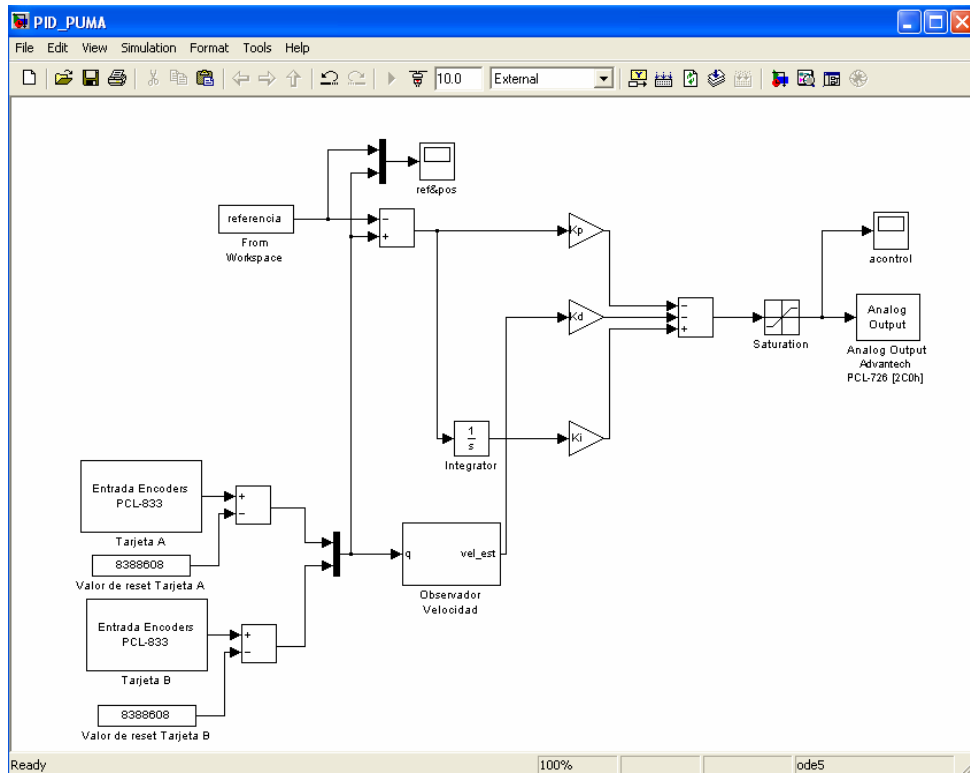


Figura B-34 Control por pasividad PID sobre el PUMA 560

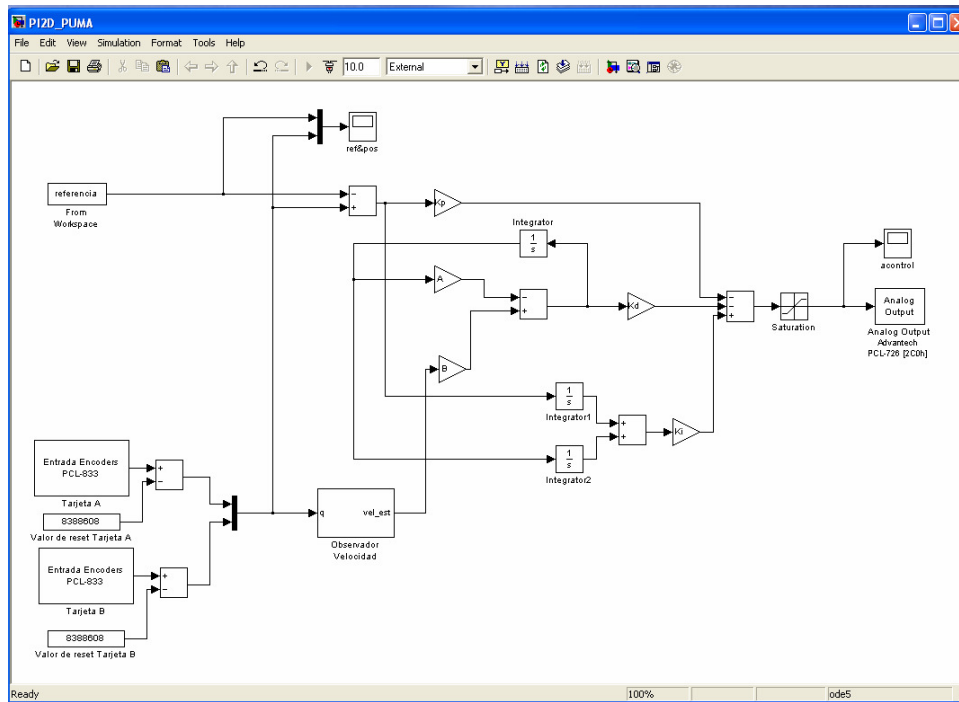


Figura B-35 Control por pasividad PI²D sobre el PUMA 560

Como se puede observar en las figuras anteriores una de las grandes ventajas a la hora de utilizar estos bloques es que permiten su reutilización en diferentes modelos de Simulink.

ÍNDICE DE FIGURAS

Figura 2-1 Problemas cinemáticos	12
Figura 2-2 Asignación de sistemas de referencia según D-H	13
Figura 3-1 Funcionamiento del control cinemático	48
Figura 3-2 Curva mediante Interpolación de Hermite	57
Figura 3-3 Curva con Interpolación cúbica con puntos intermedios	63
Figura 4-1 Estructura General Control por Linealización	73
Figura 4-3 Esquema de control del controlador por dinámica inversa con control punto a punto	78
Figura 4-40 Esquema del controlador por dinámica inversa con seguimiento de la trayectoria	85
Figura 4-77 Esquema del controlador por dinámica inversa con seguimiento de la trayectoria con acción integral	93
Figura 5-1 PUMA 560	130
Figura 5-2 Imagen de la Unidad de Control del PUMA 560	132
Figura 5-3. Esquema de la arquitectura original de un PUMA 560.	133
Figura 5-4. Arquitectura de control abierta propuesta.	136
Figura 5-5. Esquema de entradas y salidas de la arquitectura de control propuesta.	137
Figura 5-6 Conexión de la Unidad de Control del PUMA 560 al PC Industrial	138
Figura 5-7 Imagen del PC Industrial con la aplicación de control ejecutando	138
Figura 5-8 Imagen de las tarjetas de adquisición de datos	139
Figura 5-9 Circuito de Freno	140
Figura 5-10 Papel de la RTW en la fase de Diseño	142
Figura 5-11 Arquitectura Abierta de la RTW	143
Figura 5-12 Bloque del controlador de la PCL-833	145
Figura 5-13 Bloque correspondiente a la S-Function del controlador de la tarjeta PCL-833	146
Figura 5-14 Esquema Simulink del Regulador PD+G sobre el Robot Real	147

Figura B-1 Resolución del Problema Dinámico Directo en Simulink	246
Figura B-2 Control por Dinámica Inversa Punto a Punto	
Figura B-3 Simulación del controlador por dinámica inversa punto a punto en la Articulación 1	247
Figura B-4 Control por Dinámica Inversa con Seguimiento de la trayectoria	248
Figura B-5 Simulación del controlador por dinámica inversa con seguimiento de la trayectoria en la Articulación 1	248
Figura B-6 Control por Dinámica Inversa con Seguimiento de la trayectoria con acción integral	249
Figura B-7 Simulación del controlador por dinámica inversa con seguimiento de la trayectoria con acción integral en la Articulación 1	249
Figura B-8 Control por Pasividad. PD+G(q)	250
Figura B-9 Simulación del controlador PD+G(q) en la Articulación 1	250
Figura B-10 Control por Pasividad. PD+G(qd)	251
Figura B-11 Simulación del controlador PD+G(qd) en la Articulación 1	251
Figura B-12 Control por Pasividad. PD+CL(q)	252
Figura B-13 Simulación del controlador PD CL(q) en la Articulación 1	252
Figura B-14 Control por Pasividad. PD+CL(qd)	253
Figura B-15 Simulación del controlador PD CL(qd) en la Articulación 1	253
Figura B-16 Control por Pasividad. PD+DA(q)	254
Figura B-17 Simulación del controlador PD DA(q) en la Articulación 1	254
Figura B-18 Control por Pasividad. PD+DA(qd)	255
Figura B-19 Simulación del controlador PD DA(qd) en la Articulación 1	255
Figura B-20 Control por Pasividad. PID	256
Figura B-21 Simulación del controlador PID en la Articulación 1	256
Figura B-22 Control por Pasividad. PI ² D	257
Figura B-23 Simulación del controlador PI ² D en la Articulación 1	257
Figura B-24 Acceso al PUMA 560 desde Simulink	259
Figura B-25 Control por dinámica inversa punto a punto sobre el PUMA 560	260
Figura B-26 Control por dinámica inversa con seguimiento de la trayectoria sobre el PUMA 560	260

Figura B-27 Control por dinámica inversa con seguimiento de la trayectoria con acción integral sobre el PUMA 560	261
Figura B-28 Control por pasividad PD+G(q) sobre el PUMA 560	262
Figura B-29 Control por pasividad PD+G(qd) sobre el PUMA 560	263
Figura B-30 Control por pasividad PD+CL(q) sobre el PUMA 560	264
Figura B-31 Control por pasividad PD+CL(qd) sobre el PUMA 560	265
Figura B-32 Control por pasividad PD+DA(q) sobre el PUMA 560	266
Figura B-33 Control por pasividad PD+DA(qd) sobre el PUMA 560	266
Figura B-34 Control por pasividad PID sobre el PUMA 560	267
Figura B-35 Control por pasividad PI ² D sobre el PUMA 560	268

ÍNDICE DE TABLAS

Tabla 2-1 Algunos algoritmos para resolver el PDI	18
Tabla 2-2 Complejidad computacional del algoritmo propuesto	32
Tabla 2-3 Complejidad computacional para robots con $n \geq 3$	32
Tabla 3-1 Ejemplo de interpolación cúbica con puntos intermedios	63
Tabla 3-2 Implementación para cada una de las articulaciones	68
Tabla 4-1 Control sobre la trayectoria	77
Tabla 4-2 Comparación de referencia y posición para las 6 articulaciones del controlador por dinámica inversa punto a punto (abcisas: posición (grados); ordenadas: período de muestreo ($T_s=0.005s$))	79
Tabla 4-3 Acciones de Control del controlador por dinámica inversa punto a punto (abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))	80
Tabla 4-4 Comparación de referencia y posición para las 6 articulaciones del controlador por dinámica inversa punto a punto (abcisas: posición (grados); ordenadas: período de muestreo ($T_s=0.005s$))	81
Tabla 4-5 Acciones de Control del controlador por dinámica inversa punto a punto (abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))	82
Tabla 4-6 Comparación de referencia y posición para las 6 articulaciones del controlador por dinámica inversa punto a punto (abcisas: posición (rad); ordenadas: período de muestreo ($T_s=0.005s$))	83
Tabla 4-7 Acciones de Control del controlador por dinámica inversa punto a punto (abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))	84
Tabla 4-8 Comparación de referencia y posición para las 6 articulaciones del controlador por dinámica inversa con seguimiento de la trayectoria (abcisas: posición (grados); ordenadas: período de muestreo ($T_s=0.005s$))	86
Tabla 4-9 Acciones de control del controlador por dinámica inversa con seguimiento de la trayectoria (abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))	87
Tabla 4-10 Comparación de referencia y posición para las 6 articulaciones del controlador por dinámica inversa con seguimiento de la trayectoria (abcisas: posición (grados); ordenadas: período de muestreo ($T_s=0.005s$))	88
Tabla 4-11 Acciones de control del controlador por dinámica inversa con seguimiento de la trayectoria (abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))	89

Tabla 4-12 Comparación de referencia y posición para las 6 articulaciones del controlador por dinámica inversa con seguimiento de la trayectoria (abcisas: posición (rad); ordenadas: período de muestreo ($T_s=0.005s$))	90
Tabla 4-13 Acciones de control del controlador por dinámica inversa con seguimiento de la trayectoria (abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))	91
Tabla 4-14 Comparación de referencia y posición para las 6 articulaciones del controlador por dinámica inversa con seguimiento de la trayectoria con acción intergral (abcisas: posición (grados); ordenadas: período de muestreo ($T_s=0.005s$))	94
Tabla 4-15 Acciones de Control del controlador por dinámica inversa con seguimiento de la trayectoria con acción intergral (abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))	94
Tabla 4-16 Controladores Pasivos Punto a Punto para Robots Rígidos	98
Tabla 4-17 Comparación de referencia y posición para las 6 articulaciones del controlador PD+G(q) (abcisas: posición (grados); ordenadas: período de muestreo ($T_s=0.005s$))	103
Tabla 4-18 Acciones de control del controlador PD+G(q) (abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))	104
Tabla 4-19 Comparación de referencia y posición para las 6 articulaciones del controlador PD+G(q) (abcisas: posición (grados); ordenadas: período de muestreo ($T_s=0.005s$))	105
Tabla 4-20 Acciones de control del controlador PD+G(q) (abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))	106
Tabla 4-21 Comparación de referencia y posición para las 6 articulaciones del controlador PD+G(q) (abcisas: posición (rad); ordenadas: período de muestreo ($T_s=0.005s$))	107
Tabla 4-22 Acciones de control del controlador PD+G(q) (abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))	108
Tabla 4-23 Comparación de referencia y posición para las 6 articulaciones del controlador PD+G(qd) (abcisas: posición (grados); ordenadas: período de muestreo ($T_s=0.005s$))	110
Tabla 4-24 Acciones de control del controlador PD+G(qd) (abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))	111
Tabla 4-25 Comparación de referencia y posición para las 6 articulaciones del controlador PD+CL(q) (abcisas: posición (grados); ordenadas: período de muestreo ($T_s=0.005s$))	114
Tabla 4-26 Acciones de control del controlador PD+CL(q) (abcisas: acción de control (Nm); ordenadas: período de muestreo ($T_s=0.005s$))	114

Tabla 4-27 Comparación de referencia y posición para las 6 articulaciones del controlador PD+CL(qd) (abcisas: posición (grados); ordenadas: período de muestreo (Ts=0.005s))	116
Tabla 4-28 Acciones de control del controlador PD+CL(qd) (abcisas: acción de control (Nm); ordenadas: período de muestreo (Ts=0.005s))	117
Tabla 4-29 Comparación de referencia y posición para las 6 articulaciones del controlador PD+DA(q) (abcisas: posición (grados); ordenadas: período de muestreo (Ts=0.005s))	120
Tabla 4-30 Acciones de control del controlador PD+DA(q) (abcisas: acción de control (Nm); ordenadas: período de muestreo (Ts=0.005s))	120
Tabla 4-31 Comparación de referencia y posición para las 6 articulaciones del controlador PD+DA(qd) (abcisas: posición (rad); ordenadas: período de muestreo (Ts=0.005s))	122
Tabla 4-32 Acciones de control del controlador PD+DA(qd) (abcisas: acción de control (Nm); ordenadas: período de muestreo (Ts=0.005s))	122
Tabla 4-33 Comparación de referencia y posición para las 6 articulaciones del controlador PID (abcisas: posición (grados); ordenadas: período de muestreo (Ts=0.005s))	124
Tabla 4-34 Acciones de Control del controlador PID (abcisas: acción de control (Nm); ordenadas: período de muestreo (Ts=0.005s))	124
Tabla 4-35 Comparación de referencia y posición para las 6 articulaciones del controlador PI ² D (abcisas: posición (rad); ordenadas: período de muestreo (Ts=0.005s))	128
Tabla 4-36 Acciones de control del controlador PI ² D (abcisas: acción de control (Nm); ordenadas: período de muestreo (Ts=0.005s))	128
Tabla A-1 Comparación de Referencia y Posición para las 6 articulaciones del controlador por dinámica inversa punto a punto (abcisas: posición (rad); ordenadas: períodos de muestreo (Ts=0.005s))	170
Tabla A-2 Acciones de Control del controlador por dinámica inversa punto a punto (abcisas: acción de control (Nm); ordenadas: períodos de muestreo (Ts=0.005s))	171
Tabla A-3 Comparación de Referencia y Posición para las 6 articulaciones del controlador por dinámica inversa con seguimiento de la trayectoria (abcisas: posición (rad); ordenadas: períodos de muestreo (Ts=0.005s))	174
Tabla A-4 Acciones de Control del controlador por dinámica inversa con seguimiento de la trayectoria (abcisas: acción de control (Nm); ordenadas: períodos de muestreo (Ts=0.005s))	175

Tabla A-5 Comparación de Referencia y Posición para las 6 articulaciones del controlador por dinámica inversa con seguimiento de la trayectoria con acción integral (abcisas: posición (rad); ordenadas: períodos de muestreo ($T_s=0.005s$))	178
Tabla A-6 Acciones de Control del controlador por dinámica inversa con seguimiento de la trayectoria con acción integral (abcisas: acción de control (Nm); ordenadas: períodos de muestreo ($T_s=0.005s$))	179
Tabla A-7 Comparación de Referencia y Posición para las 6 articulaciones del controlador por dinámica inversa con seguimiento de la trayectoria con acción integral (abcisas: posición (rad); ordenadas: períodos de muestreo ($T_s=0.005s$))	180
Tabla A-8 Acciones de Control del controlador por dinámica inversa con seguimiento de la trayectoria con acción integral (abcisas: acción de control (Nm); ordenadas: períodos de muestreo ($T_s=0.005s$))	181
Tabla A-9 Comparación de Referencia y Posición para las 6 articulaciones del controlador por dinámica inversa con seguimiento de la trayectoria con acción integral (abcisas: posición (rad); ordenadas: períodos de muestreo ($T_s=0.005s$))	182
Tabla A-10 Acciones de Control del controlador por dinámica inversa con seguimiento de la trayectoria con acción integral (abcisas: acción de control (Nm); ordenadas: períodos de muestreo ($T_s=0.005s$))	183
Tabla A-11 Comparación de Referencia y Posición para las 6 articulaciones del controlador PD+G(q) (abcisas: posición (rad); ordenadas: períodos de muestreo ($T_s=0.005s$))	187
Tabla A-12 Acciones de Control del controlador PD+G(q) (abcisas: acción de control (Nm); ordenadas: períodos de muestreo ($T_s=0.005s$))	188
Tabla A-13 Comparación de Referencia y Posición para las 6 articulaciones del controlador PD+G(qd) (abcisas: posición (rad); ordenadas: períodos de muestreo ($T_s=0.005s$))	191
Tabla A-14 Acciones de Control del controlador PD+G(q) (abcisas: acción de control (Nm); ordenadas: períodos de muestreo ($T_s=0.005s$))	192
Tabla A-15 Comparación de Referencia y Posición para las 6 articulaciones del controlador PD+CL(q) (abcisas: posición (rad); ordenadas: períodos de muestreo ($T_s=0.005s$))	195
Tabla A-16 Acciones de Control controlador PD+CL(q) (abcisas: acción de control (Nm); ordenadas: períodos de muestreo ($T_s=0.005s$))	196
Tabla A-17 Comparación de Referencia y Posición para las 6 articulaciones controlador PD+CL(qd) (abcisas: posición (rad); ordenadas: períodos de muestreo ($T_s=0.005s$))	200
Tabla A-18 Acciones de Control controlador PD+CL(qd) (abcisas: acción de control (Nm); ordenadas: períodos de muestreo ($T_s=0.005s$))	201

Tabla A-19 Comparación de Referencia y Posición para las 6 articulaciones del controlador PD+DA(q) (abcisas: posición (rad); ordenadas: períodos de muestreo (Ts=0.005s))	204
Tabla A-20 Acciones de Control del controlador PD+DA(q) (abcisas: acción de control (Nm); ordenadas: períodos de muestreo (Ts=0.005s))	205
Tabla A-21 Comparación de Referencia y Posición para las 6 articulaciones del controlador PD+DA(qd) (abcisas: posición (rad); ordenadas: períodos de muestreo (Ts=0.005s))	208
Tabla A-22 Acciones de Control del controlador PD+DA(qd) (abcisas: acción de control (Nm); ordenadas: períodos de muestreo (Ts=0.005s))	209
Tabla A-23 Comparación de Referencia y Posición para las 6 articulaciones del controlador PID (abcisas: posición (rad); ordenadas: períodos de muestreo (Ts=0.005s))	217
Tabla A-24 Acciones de Control del controlador PID (abcisas: acción de control (Nm); ordenadas: períodos de muestreo (Ts=0.005s))	218
Tabla A-25 Comparación de Referencia y Posición para las 6 articulaciones del controlador PID (abcisas: posición (rad); ordenadas: períodos de muestreo (Ts=0.005s))	219
Tabla A-26 Acciones de Control del controlador PID (abcisas: acción de control (Nm); ordenadas: períodos de muestreo (Ts=0.005s))	220
Tabla A-27 Comparación de Referencia y Posición para las 6 articulaciones del controlador PID (abcisas: posición (rad); ordenadas: períodos de muestreo (Ts=0.005s))	221
Tabla A-28 Acciones de Control del controlador PID (abcisas: acción de control (Nm); ordenadas: períodos de muestreo (Ts=0.005s))	222
Tabla A-29 Comparación de Referencia y Posición para las 6 articulaciones del controlador PI ² D (abcisas: posición (rad); ordenadas: períodos de muestreo (Ts=0.005s))	236
Tabla A-30 Acciones de Control del controlador PI ² D (abcisas: acción de control (Nm); ordenadas: períodos de muestreo (Ts=0.005s))	237

