

Centro de Investigación Científica y de Educación Superior de Ensenada



CAMPO DE VELOCIDAD DE MECANISMOS

**TESIS
MAESTRIA EN CIENCIAS**

JOSE CLEMENTE NAVARRETE GARCIA

ENSENADA BAJA CALIFORNIA AGOSTO DE 2003

TESIS DEFENDIDA POR

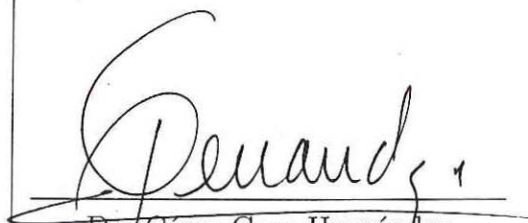
José Clemente Navarrete García

Y aprobada por el siguiente comité:



Dr. Rafael de Jesús Kelly Martínez

Director del Comité



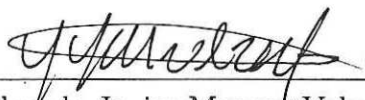
Dr. César Cruz Hernández

Miembro del Comité



Dr. Raúl Ramón Castro Escamilla

Miembro del Comité



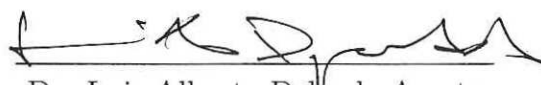
Dr. Eduardo Javier Moreno Valenzuela

Miembro del Comité



Dr. José Rosario Gallardo López

Jefe del Departamento



Dr. Luis Alberto Delgado Argote

Director de Estudios de Posgrado

Ensenada, B.C. a 28 de Agosto de 2003

CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN
SUPERIOR DE ENSENADA



DIVISIÓN DE FÍSICA APLICADA
DEPARTAMENTO DE ELECTRÓNICA Y
TELECOMUNICACIONES

Campo de velocidad de mecanismos

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el grado de

MAESTRO EN CIENCIAS

Presenta:

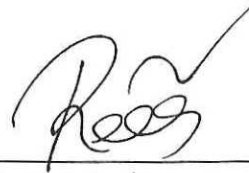
José Clemente Navarrete García

Ensenada, Baja California; a 28 de Agosto de 2003.

RESUMEN de la tesis de **José Clemente Navarrete García**, presentada como requisito parcial para obtener el grado de **MAESTRO EN CIENCIAS en ELECTRÓNICA Y TELECOMUNICACIONES**. Ensenada, Baja California, México. Agosto de 2003.

Campo de velocidad de mecanismos

Resumen aprobado por:



Dr. Rafael de Jesús Kelly Martínez

Director de Tesis

Esta tesis trata sobre el control de robots manipuladores mediante campo de velocidad en coordenadas cartesianas. El control mediante campo de velocidad es una filosofía de control novedosa que utiliza una referencia de movimiento definida de forma independiente al tiempo.

En esta tesis se estudian algoritmos que cumplen con el objetivo de control de campo de velocidad y se presentan pruebas de estabilidad utilizando el método directo de Lyapunov; también se diseña un campo vectorial con características particulares deseadas y se utiliza éste como referencia de movimiento.

Los resultados teóricos se respaldan con experimentos tras la prueba de los algoritmos en un mecanismo de laboratorio. Finalmente se presenta un estudio comparativo del desempeño experimental de los controladores.

Palabras clave: Robótica, Campo de Velocidad, Algoritmo de Control.

ABSTRACT of the thesis presented by **José Clemente Navarrete García**, as a partial requirement to obtain the **MASTER IN SCIENCE** degree in **ELECTRONICS AND TELECOMUNICATIONS**. Ensenada, Baja California, Mexico. August 2003.

Velocity field control for robot manipulators

Abstract approved by:



Dr. Rafael de Jesús Kelly Martínez

Thesis director

This thesis is about robot control via velocity field in Cartesian space. Velocity field control is a recent approach that uses a time independent defined motion reference.

In this thesis we study algorithms that achieve the velocity field control objective, and we perform stability analysis using Lyapunov's direct method; also we design a vector field with particular desired characteristics.

The theoretical results are backed up with experimental tests after the implementation of the algorithms in a laboratory mechanism. Finally a comparative study of the experimental performance of the controllers is presented.

Keywords: Robotics, Velocity Field, Control Algorithm.

A Eunice Martínez

Agradecimientos

Al Dr. Rafael Kelly Martínez por su confianza, enseñanzas y adecuada dirección en este trabajo de tesis.

Al Dr. Javier Moreno Valenzuela por su tiempo y por su muy importante e incondicional apoyo en esta tesis.

Al M.C. Ricardo Campa por su interés y apoyo en este proyecto.

A Rubén Hernández por su importante contribución en este trabajo.

Esta tesis no hubiese sido posible sin el apoyo del:

Centro de Investigación Científica y de Educación Superior de Ensenada

Consejo Nacional de Ciencia y Tecnología

Ensenada, México
28 de Agosto de 2003

José Clemente Navarrete García

Tabla de Contenido

Resumen	2
Abstract	3
Agradecimientos	6
Lista de Figuras	8
 CAPÍTULO	 PÁGINA
I Introducción	12
I.1 Robots manipuladores	14
I.1.1 Modelado dinámico	16
I.1.2 Cinemática	18
I.2 Control de robots	20
I.3 Campo de velocidad de mecanismos	21
I.4 Trabajo de tesis	22
 II Diseño de un campo de velocidad	23
II.1 Definiciones	23
II.2 Características deseadas	24
II.3 Metodología	25
II.4 Diseño	28
II.5 Consideraciones prácticas	33
 III Control basado en dinámica inversa	34
III.1 Algoritmo de control basado en dinámica inversa	34
III.1.1 Análisis de estabilidad	36
III.2 Simulaciones	37
III.3 Experimentos	44
 IV Control basado en estructura PD+	52
IV.1 Control PD+	52
IV.1.1 Análisis de estabilidad	53
IV.2 Control PD+ con compensación de fricción utilizando el modelo de Dahl	54
IV.2.1 Análisis de estabilidad	56
IV.3 Simulaciones	58
IV.4 Experimentos	63
 V Control de Li y Horowitz	72
V.1 Terminología	72
V.2 Ley de control	73
V.3 Simulaciones	75
V.4 Experimentos	83

Tabla de Contenido (Continuación)

CAPÍTULO	PÁGINA
VI Conclusiones	91
VI.1 Propuestas de trabajo a futuro	93
Bibliografía	95
Apéndices	96
A Espacios de funciones y estabilidad en el sentido de Lyapunov	97
A.1 Espacios de funciones	97
A.2 Estabilidad en el sentido de Lyapunov	98
B Códigos de Simnon	100
B.1 Control basado en dinámica inversa	100
B.1.1 Macro	100
B.1.2 Controlador	101
B.1.3 Robot Bugarín	105
B.1.4 Conexiones	107
B.2 Control en estructura PD+	107
B.2.1 Macro	107
B.2.2 Controlador	108
B.2.3 Conexiones	112
B.3 Control de Li y Horowitz	113
B.3.1 Macro	113
B.3.2 Controlador	113
B.3.3 Conexiones	120
C Codigos de WinMechLab	122
C.1 Control basado en dinámica inversa	122
C.2 Control en estructura PD+	129
C.3 Control de Li y Horowitz	137

Lista de Figuras

FIGURA		PÁGINA
1	Robot manipulador.	13
2	Cadena cinemática abierta.	14
3	Robot manipulador de 3 g.d.l.	15
4	Diagrama de bloques de un robot como sistema entrada/salida.	16
5	Campo escalar.	24
6	Ovalos de Cassini.	29
7	Ovalo de Cassini.	30
8	Componente normal del campo vectorial.	30
9	Componente tangencial del campo vectorial.	32
10	Campo vectorial que converge a un ovalo de Cassini.	32
11	Diagrama de bloques del controlador por dinámica inversa.	36
12	Trayectoria del extremo del robot correspondiente al eje cartesiano x_1 . [dinámica inversa - simulación]	37
13	Trayectoria del extremo del robot correspondiente al eje cartesiano x_2 . [dinámica inversa - simulación]	38
14	Trayectoria realizada por el extremo del robot en el espacio cartesiano x_1 - x_2 . [dinámica inversa - simulación]	39
15	Velocidad del extremo del robot correspondiente al eje cartesiano x_1 . [dinámica inversa - simulación]	39
16	Velocidad del extremo del robot correspondiente al eje cartesiano x_2 . [dinámica inversa - simulación]	40
17	Norma de la velocidad del extremo del robot. [dinámica inversa - simu- lación]	40
18	Error de campo de velocidad que corresponde al eje cartesiano x_1 . [dinámica inversa - simulación]	41
19	Error de campo de velocidad que corresponde al eje cartesiano x_2 . [dinámica inversa - simulación]	41
20	Norma del error de campo de velocidad. [dinámica inversa - simulación]	42
21	Par aplicado a la articulación 1 del robot. [dinámica inversa - simulación]	43
22	Par aplicado a la articulación 2 del robot. [dinámica inversa - simulación]	43
23	Robot Bugarín.	44
24	Trayectoria del extremo del robot correspondiente al eje cartesiano x_1 . [dinámica inversa - experimento]	45
25	Trayectoria del extremo del robot correspondiente al eje cartesiano x_2 . [dinámica inversa - experimento]	46
26	Trayectoria realizada por el extremo del robot en el espacio cartesiano x_1 - x_2 . [dinámica inversa - experimento]	46

Lista de Figuras (Continuación)

FIGURA		PÁGINA
27	Velocidad del extremo del robot correspondiente al eje cartesiano x_1 . [dinámica inversa - experimento]	47
28	Velocidad del extremo del robot correspondiente al eje cartesiano x_2 . [dinámica inversa - experimento]	48
29	Norma de la velocidad del extremo del robot. [dinámica inversa - experimento]	48
30	Error de campo de velocidad que corresponde al eje cartesiano x_1 . [dinámica inversa - experimento]	49
31	Error de campo de velocidad que corresponde al eje cartesiano x_2 . [dinámica inversa - experimento]	49
32	Norma del error de campo de velocidad. [dinámica inversa - experimento]	50
33	Par aplicado a la articulación 1 del robot. [dinámica inversa - experimento]	50
34	Par aplicado a la articulación 2 del robot. [dinámica inversa - experimento]	51
35	Diagrama de bloques del controlador PD+ para coordenadas cartesianas.	53
36	Traectoria del extremo del robot correspondiente al eje cartesiano x_1 . [PD+ - simulación]	59
37	Traectoria del extremo del robot correspondiente al eje cartesiano x_2 . [PD+ - simulación]	59
38	Traectoria del extremo del robot en el espacio cartesiano. [PD+ - simulación]	60
39	Velocidad del extremo del robot referente al eje cartesiano x_1 . [PD+ - simulación]	61
40	Velocidad del extremo del robot referente al eje cartesiano x_2 . [PD+ - simulación]	61
41	Norma de la velocidad del extremo del robot. [PD+ - simulación] . . .	62
42	Error de campo de velocidad que corresponde al eje cartesiano x_1 . [PD+ - simulación]	62
43	Error de campo de velocidad que corresponde al eje cartesiano x_2 . [PD+ - simulación]	63
44	Norma del error de campo de velocidad. [PD+ - simulación]	64
45	Par aplicado a la articulación 1 del robot. [PD+ - simulación]	64
46	Par aplicado a la articulación 2 del robot. [PD+ - simulación]	65
47	Traectoria del extremo del robot correspondiente al eje cartesiano x_1 . [PD+ - experimento]	66
48	Traectoria del extremo del robot correspondiente al eje cartesiano x_2 . [PD+ - experimento]	66
49	Traectoria del extremo del robot en el espacio cartesiano. [PD+ - experimento]	67

Lista de Figuras (Continuación)

FIGURA		PÁGINA
50	Velocidad del extremo del robot referente al eje cartesiano x_1 . [PD+ - experimento]	68
51	Velocidad del extremo del robot referente al eje cartesiano x_2 . [PD+ - experimento]	68
52	Norma de la velocidad del extremo del robot. [PD+ - experimento] . .	69
53	Error de campo de velocidad que corresponde al eje cartesiano x_1 . [PD+ - experimento]	69
54	Error de campo de velocidad que corresponde al eje cartesiano x_2 . [PD+ - experimento]	70
55	Norma del error de campo de velocidad. [PD+ - experimento]	70
56	Par aplicado a la articulación 1 del robot. [PD+ - experimento]	71
57	Par aplicado a la articulación 2 del robot. [PD+ - experimento]	71
58	Diagrama de bloques del controlador de Li y Horowitz.	75
59	Trayectoria del extremo del robot correspondiente al eje cartesiano x_1 . [Li y Horowitz - simulación]	77
60	Trayectoria del extremo del robot correspondiente al eje cartesiano x_2 . [Li y Horowitz - simulación]	78
61	Trayectoria del extremo del robot en el espacio cartesiano. [Li y Horowitz - simulación]	78
62	Velocidad del extremo del robot referente al eje cartesiano x_1 . [Li y Horowitz - simulación]	79
63	Velocidad del extremo del robot referente al eje cartesiano x_2 . [Li y Horowitz - simulación]	79
64	Norma de la velocidad del extremo del robot. [Li y Horowitz - simulación]	80
65	Error de campo de velocidad que corresponde al eje cartesiano x_1 . [Li y Horowitz - simulación]	80
66	Error de campo de velocidad que corresponde al eje cartesiano x_2 . [Li y Horowitz - simulación]	81
67	Norma del error de campo de velocidad. [Li y Horowitz - simulación] .	82
68	Par aplicado a la articulación 1 del robot. [Li y Horowitz - simulación]	82
69	Par aplicado a la articulación 2 del robot. [Li y Horowitz - simulación]	83
70	Trayectoria del extremo del robot correspondiente al eje cartesiano x_1 . [Li y Horowitz - experimento]	84
71	Trayectoria del extremo del robot correspondiente al eje cartesiano x_2 . [Li y Horowitz - experimento]	85
72	Trayectoria del extremo del robot en el espacio cartesiano x_1 - x_2 . [Li y Horowitz - experimento]	85

Lista de Figuras (Continuación)

FIGURA		PÁGINA
73	Velocidad del extremo del robot referente al eje cartesiano x_1 . [Li y Horowitz - experimento]	86
74	Velocidad del extremo del robot referente al eje cartesiano x_2 . [Li y Horowitz - experimento]	86
75	Norma de la velocidad del extremo del robot. [Li y Horowitz - experimento]	87
76	Error de campo de velocidad que corresponde al eje cartesiano x_1 . [Li y Horowitz - experimento]	87
77	Error de campo de velocidad que corresponde al eje cartesiano x_2 . [Li y Horowitz - experimento]	88
78	Norma del error de campo de velocidad. [Li y Horowitz - experimento]	89
79	Par aplicado a la articulación 1 del robot. [Li y Horowitz - experimento]	89
80	Par aplicado a la articulación 2 del robot. [Li y Horowitz - experimento]	90
81	Comparación de los índices de desempeño	92

Capítulo I

Introducción

El objeto de estudio del presente trabajo de tesis es el control de una clase particular de mecanismos: los robots manipuladores.

Una definición abstracta de un robot manipulador es (Kelly y Santibáñez, 2003):

- Un robot manipulador es un brazo mecánico articulado formado por eslabones conectados a través de uniones o articulaciones que permiten un movimiento relativo entre cada dos eslabones consecutivos.

Esta definición se refiere a la parte mecánica que forma un *sistema robótico*. Un sistema robótico consiste en un mecanismo (manipulador) provisto de sensores que le permitan medir las variables de interés, actuadores que proporcionen el movimiento relativo de sus partes y un sistema de control programable.

Un robot manipulador puede ser diseñado para ejecutar tareas en el espacio libre, o teniendo contacto con su medio ambiente. En el espacio libre puede ejecutar tareas tales como el posicionamiento del extremo libre del último eslabón, o el seguimiento de alguna trayectoria particular por este mismo extremo. Cuando tiene contacto con su medio ambiente puede ser el caso de tareas en las cuales el extremo del robot toma objetos para desplazarlos de un lugar a otro, o cargar algún objeto en general, bloquear el flujo en una línea de producción por citar algunos ejemplos.

Referente al movimiento de un robot, el concepto de *grados de libertad* (g.d.l.) se define como el número de movimientos distintos que las partes del mecanismo puede realizar. En general, en un robot manipulador los grados de libertad equivalen al número de articulaciones del mismo.

El movimiento del manipulador es posible debido a su articulaciones. Las articulaciones pueden tener actuadores eléctricos, neumáticos, o hidráulicos. En un robot manipulador las articulaciones pueden ser de tipo rotacional o de tipo traslacional.¹

¹También llamadas prismáticas.

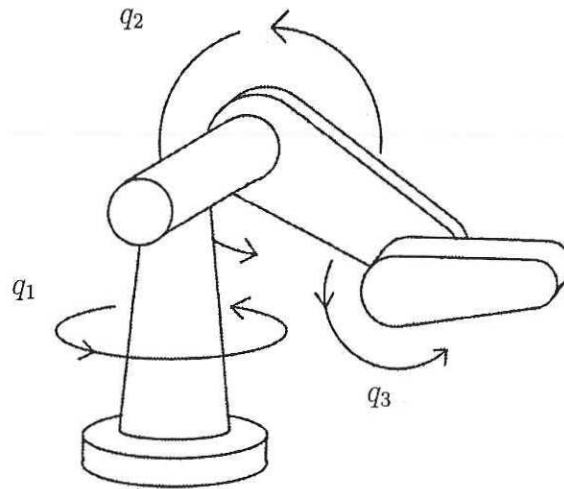


Figura 1: Robot manipulador.

En este trabajo de tesis se consideran mecanismos con articulaciones de tipo rotacional, tal como el que se muestra en la Figura 1.

Debido a consideraciones prácticas, los robots manipuladores se construyen con articulaciones que tengan únicamente un tipo de movimiento, un grado de libertad. Además la mayoría de los manipuladores tienen articulaciones rotacionales, o articulaciones traslacionales. En caso raro de que una articulación sea construida con n grados de libertad, ésta puede ser modelada como una cadena de eslabones, de n grados de libertad, conectados con $n - 1$ eslabones de longitud cero (Craig, 1989).

Así, sin perder generalidad, se considerará en nuestro estudio manipuladores que tengan articulaciones de un sólo grado de libertad cada una.

Esquemáticamente puede representarse un manipulador desde un punto de vista mecánico como una cadena de cuerpos rígidos (eslabones) conectados entre sí mediante uniones traslacionales o rotacionales, formando una cadena cinemática². La estructura común de un robot manipulador es una cadena cinemática abierta. Es decir, el primer eslabón fijo a un marco de referencia y el extremo final del último eslabón se encuentra libre en el espacio o dicho de otra forma, no tiene contacto con su medio ambiente. En la Figura 2 se muestra una cadena cinemática abierta con n eslabones en un marco de

²Cinemática: ciencia que estudia la geometría del movimiento.

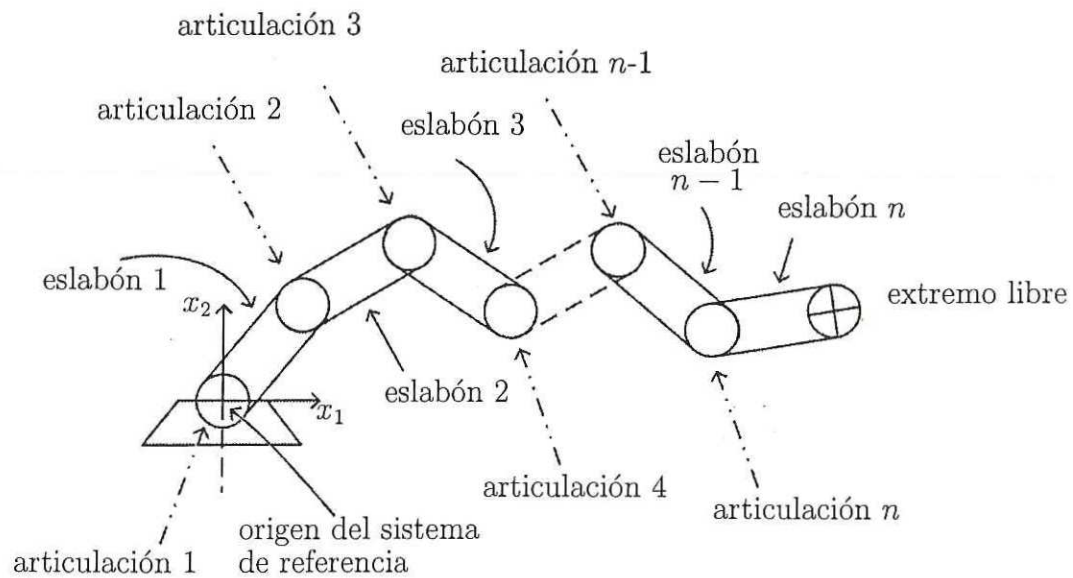


Figura 2: Cadena cinemática abierta.

referencia cartesiano.

En síntesis, en el presente documento se estudia el control de robots manipuladores con articulaciones rotacionales que realizan tareas sin tener contacto con su medio ambiente.

I.1 Robots manipuladores

Para fines del estudio analítico de las propiedades de un mecanismo manipulador se utiliza notación de vectores y matrices para agrupar las distintas variables de interés referentes al robot.

En un robot manipulador generalmente las variables de interés son las posiciones articulares, velocidades articulares, y los pares de control. Se denota por $\mathbf{q}$ el vector de posiciones articulares,

$$\mathbf{q} = [q_1 \quad q_2 \quad \dots \quad q_n]^T \quad (1)$$

por $\dot{\mathbf{q}}$ el vector de velocidades articulares,

$$\dot{\mathbf{q}} = [\dot{q}_1 \quad \dot{q}_2 \quad \dots \quad \dot{q}_n]^T \quad (2)$$

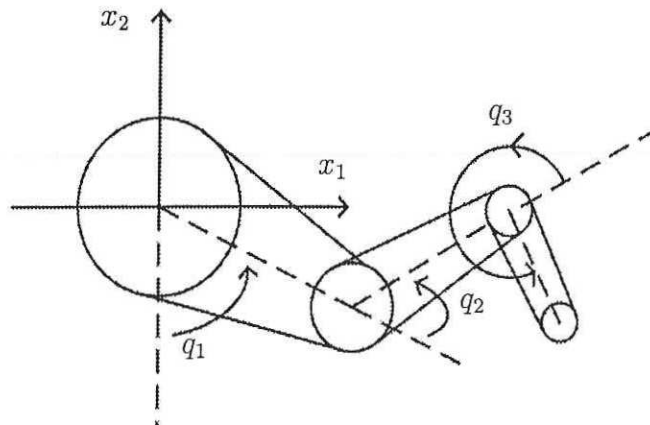


Figura 3: Robot manipulador de 3 g.d.l.

y por τ :

$$\tau = \begin{bmatrix} \tau_1 & \tau_2 & \dots & \tau_n \end{bmatrix}^T \quad (3)$$

al vector de pares de control.

De forma natural, surge la necesidad de conocer la posición y velocidad del extremo final del robot, el cual en términos prácticos es el ejecutor de las tareas a encomendar.

Hemos de expresar la posición y velocidad del extremo del robot utilizando también notación vectorial, denotando por $\mathbf{x}$ el vector de postura (posición y orientación),

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}^T \quad (4)$$

y por $\dot{\mathbf{x}}$ el vector de velocidad del extremo del robot manipulador:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 & \dot{x}_2 & \dots & \dot{x}_n \end{bmatrix}^T \quad (5)$$

En la Figura 3 se muestra un robot de tres grados de libertad. En la base del robot se muestra el origen del sistema de referencia cartesiana, indicándose mediante q_1, q_2, q_3 las correspondientes posiciones articulares o ángulos de desplazamiento de cada eslabón respecto al anterior. El primer eslabón tiene como referencia a la horizontal, y de ahí en adelante, el eslabón $n - 1$ es referencia angular del eslabón n .

Un robot manipulador puede ser visto como un sistema entrada/salida en un diagrama de bloques, donde la entrada al sistema es el vector de pares de control τ y las salidas son las variables vectoriales $\mathbf{q}$ y $\dot{\mathbf{q}}$, tal como se muestra en la Figura 4.

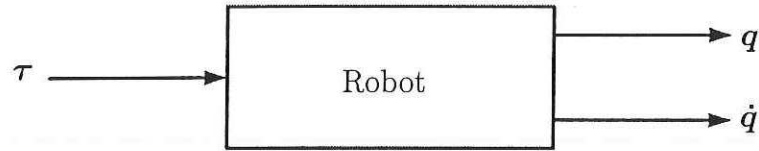


Figura 4: Diagrama de bloques de un robot como sistema entrada/salida.

I.1.1 Modelado dinámico

El contenido de esta sección proviene de (Kelly y Santibáñez, 2003).

El modelo dinámico de un robot consiste en un sistema de ecuaciones diferenciales ordinarias (no lineales en general), expresadas en conjunto a través de notación vectorial en función de las posiciones articulares $\mathbf{q}$ o sus posiciones cartesianas $\mathbf{x}$, generalmente de segundo orden, como:

$$f(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\tau}) = 0 \quad (6)$$

o

$$f(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}, \boldsymbol{\tau}) = 0 \quad (7)$$

La obtención de un modelo matemático que describa el comportamiento dinámico de un robot manipulador se puede obtener utilizando las ecuaciones de movimiento de Lagrange.³

El lagrangiano de un robot (denotado por $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$) es igual a la diferencia entre las funciones que modelan su energía cinética y las que modelan su energía potencial:

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) - \mathcal{U}(\mathbf{q}, \dot{\mathbf{q}}) \quad (8)$$

donde $\mathcal{K}$ denota energía cinética y $\mathcal{U}$ denota energía potencial. Las ecuaciones de movimiento de Lagrange para un manipulador de n grados de libertad, están dadas por:

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \right] - \frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} = \boldsymbol{\tau} \quad (9)$$

³En honor a Joseph Lagrange, matemático francés del siglo XVIII.

o de forma equivalente:

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{q}_i} \right] - \frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial q_i} = \tau_i; \quad i = 1, \dots, n \quad (10)$$

Los robots manipuladores tiene la importante propiedad de poseer un modelo dinámico en común. En ausencia de fuerzas de fricción, la ecuación dinámica general para un robot manipulador de n grados de libertad está dada por:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (11)$$

donde $M(\mathbf{q})$ es una matriz de dimensión $n \times n$ llamada matriz de inercia, $C(\mathbf{q}, \dot{\mathbf{q}})$ es una matriz de dimensión $n \times n$ llamada matriz de fuerzas centrífugas y de Coriolis, $\mathbf{g}(\mathbf{q})$ es el vector de dimensión $n \times 1$ llamado de pares gravitacionales y $\boldsymbol{\tau}$ es un vector de dimensión $n \times 1$ de pares de control.

En este modelo dinámico las matrices $M(\mathbf{q})$, $C(\mathbf{q}, \dot{\mathbf{q}})$ y el vector $\mathbf{g}(\mathbf{q})$ son características del mecanismo en particular, e independientes de los objetivos de control y tareas a ejecutar en general.

La matriz de inercia $M(\mathbf{q})$ es una matriz simétrica definida positiva de dimensión $n \times n$ cuyos elementos son funciones exclusivamente de $\mathbf{q}$.

Algunas propiedades del modelo dinámico de un robot de n g.d.l. son:

- La matriz $M(\mathbf{q})^{-1}$ existe y es también definida positiva.
- Siempre es posible seleccionar la matriz $C(\mathbf{q}, \dot{\mathbf{q}})$ de modo que esté relacionada con la matriz de inercia $M(\mathbf{q})$ mediante la expresión:

$$\mathbf{x}^T \left[\frac{1}{2} \dot{M}(\mathbf{q}) - C(\mathbf{q}, \dot{\mathbf{q}}) \right] \mathbf{x} = 0, \quad \forall \quad \mathbf{q}, \dot{\mathbf{q}}, \mathbf{x} \in \mathbb{R}^n \quad (12)$$

y de hecho $\frac{1}{2} \dot{M}(\mathbf{q}) - C(\mathbf{q}, \dot{\mathbf{q}})$ es una matriz antisimétrica.

- También resulta que:

$$\dot{M}(\mathbf{q}) = C(\mathbf{q}, \dot{\mathbf{q}}) + C(\mathbf{q}, \dot{\mathbf{q}})^T \quad (13)$$

Independientemente de la manera en que sea obtenida $C(\mathbf{q}, \dot{\mathbf{q}})$, siempre se satisface:

$$\dot{\mathbf{q}}^T \left[\frac{1}{2} \dot{M}(\mathbf{q}) - C(\mathbf{q}, \dot{\mathbf{q}}) \right] \dot{\mathbf{q}} = 0, \quad \forall \quad \mathbf{q}, \dot{\mathbf{q}} \in \mathbb{R}^n \quad (14)$$

I.1.2 Cinemática

El contenido de esta sección proviene de (Sciavico y Siciliano, 1996).

El problema de cinemática directa consiste en obtener las posiciones y velocidades cartesianas del extremo del robot a partir del conocimiento de las correspondientes posiciones y velocidades articulares. El problema se resuelve mediante la obtención del modelo cinemático directo, y a través del cálculo de un ente matemático útil en el ámbito de control de robots: la matriz jacobiana.

Modelo cinemático directo

Las coordenadas cartesianas del extremo de un robot no redundante se obtienen a través del modelo cinemático directo:

$$\mathbf{x} = \mathbf{h}(\mathbf{q}) \quad (15)$$

donde $\mathbf{h}(\mathbf{q})$ es un vector de dimensión $n \times 1$ que contiene la posición del extremo del robot en función de sus posiciones articulares. El modelo cinemático directo se obtiene a través de las relaciones trigonométricas entre los elabones del mecanismo.

La matriz jacobiana

Derivando el modelo cinemático directo (15) respecto al tiempo se obtiene:

$$\dot{\mathbf{x}} = \frac{\partial \mathbf{h}(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \quad (16)$$

donde $\mathbf{J}(\mathbf{q})$ es una matriz de dimensión $n \times n$ denominada la matriz jacobiana o simplemente el jacobiano del robot manipulador. El jacobiano es útil en control de robots en coordenadas cartesianas.

Cinemática inversa

El problema de cinemática inversa consiste en la determinación de todos los posibles arreglos de las posiciones articulares que correspondan a una posición y orientación particular del extremo del robot. Este problema es complejo ya que:

- Las ecuaciones a resolver son en general no lineales, por lo que existen casos en los que no es posible encontrar una solución algebraica.
- Pueden existir soluciones múltiples.
- Pueden existir un número infinito de soluciones; e.g. es el caso de manipuladores redundantes.⁴
- Dependiendo de la estructura cinemática del mecanismo, puede ser que no exista solución alguna.

Singularidades cinemáticas

El jacobiano define un mapeo entre el vector $\dot{\mathbf{q}}$ de velocidades articulares y el vector velocidad del extremo del robot $\dot{\mathbf{x}}$:

$$\dot{\mathbf{x}} = J(\mathbf{q})\dot{\mathbf{q}} \quad (17)$$

El jacobiano es en general una función del vector $\mathbf{q}$; aquellas configuraciones en las cuales $J(\mathbf{q})$ pierde rango son llamadas singularidades cinemáticas.

Es importante conocer las singularidades de un manipulador puesto que:

- Las singularidades representan configuraciones en las cuales la movilidad de la estructura se ve reducida, i.e., no es posible realizar movimientos arbitrarios del extremo del robot.
- Cuando la estructura se encuentra en alguna singularidad, puede haber un número infinito de soluciones al problema de cinemática inversa.
- En la vecindad de una singularidad, pequeñas velocidades en el espacio operacional, pueden ocasionar enormes velocidades en el espacio articular.

Las singularidades ocurren cuando:

- El brazo manipulador se encuentra totalmente extendido, en la frontera del espacio de trabajo.

⁴Un manipulador redundante tiene más grados de libertad que el número de variables requeridas para describir una tarea encomendada.

- Dos o mas articulaciones se encuentran alineadas, o en un mismo punto.

En ocasiones es necesario calcular el vector de velocidades articulares de un robot a partir del conocimiento del respectivo vector de velocidades cartesianas. En este caso se procede despejando $\dot{\mathbf{q}}$ de (17):

$$\dot{\mathbf{q}} = J(\mathbf{q})^{-1} \dot{\mathbf{x}} \quad (18)$$

A $J(\mathbf{q})^{-1}$ se le conoce como la matriz jacobiana inversa o simplemente el jacobiano inverso.

I.2 Control de robots

El control de un robot manipulador implica conseguir que el mecanismo a través de sus actuadores, ejecute una tarea encomendada, que además cumpla en general con requisitos impuestos que en conjunto llamaremos *objetivos de control*.

De forma tradicional los objetivos de control para robots manipuladores son definidos utilizando referencias temporales. Tales objetivos de control buscan que a medida que el tiempo avanza se anule la diferencia entre la referencia propuesta y la variable que la ha de seguir.

En el problema de regulación para robots manipuladores en coordenadas cartesianas se busca cumplir con:

$$\lim_{t \rightarrow \infty} [\mathbf{x}_d - \mathbf{x}(t)] = \mathbf{0} \quad (19)$$

donde $\mathbf{x}_d$ es un vector constante de dimensión $n \times 1$.

Para el problema de seguimiento de trayectoria en coordenadas cartesianas el objetivo de control es:

$$\lim_{t \rightarrow \infty} [\mathbf{x}_d(t) - \mathbf{x}(t)] = \mathbf{0} \quad (20)$$

donde $\mathbf{x}_d(t)$ es un vector de referencias temporales de dimensión $n \times 1$.

Los anteriores se pueden denominar objetivos de control clásicos para el caso de regulación y el caso de seguimiento de trayectoria, respectivamente.

La ley de control es una expresión matemática definida en función de las posiciones y velocidades articulares del robot y permite determinar las acciones de control τ . El modelo de un sistema dinámico en malla cerrada con un controlador se obtiene sustituyendo la ley de control en la expresión del modelo dinámico dado por (11). Al hacerlo se llega a una expresión del sistema que se denomina “en lazo cerrado”, que en general definirá un sistema con nuevas características dinámicas.

El objetivo del diseño de τ es conseguir modificar las características dinámicas del sistema de manera que cumplan con los objetivos de control propuestos.

I.3 Campo de velocidad de mecanismos

En este trabajo de tesis se aborda el estudio de una propuesta alterna de control para robots manipuladores llamada campo de velocidad. Esta propuesta tiene su origen en (Li y Horowitz, 1999).

Por campo de velocidad de mecanismos se debe entender una filosofía de control en la cual existe una referencia de velocidad definida mediante un campo vectorial en función de las coordenadas articulares o cartesianas del robot y además se considera implícito como objetivo de control:

$$\lim_{t \rightarrow \infty} e(\mathbf{x}(t), \dot{\mathbf{x}}(t)) = \lim_{t \rightarrow \infty} [\mathbf{v}(\mathbf{x}(t)) - \dot{\mathbf{x}}(t)] = \mathbf{0} \quad (21)$$

donde e es llamado error de campo de velocidad, y es definido como la diferencia entre el vector campo de velocidad $\mathbf{v}(\mathbf{x})$ definido en función de las coordenadas cartesianas (o articulares) del robot y el vector de velocidades $\dot{\mathbf{x}}$ (o $\dot{\mathbf{q}}$) del mismo. En esta tesis se supondrá que el jacobiano es de rango pleno y por lo tanto existe su inversa $J(\mathbf{q})^{-1}$.

En campo de velocidad de mecanismos, no existe una referencia de posición explícita y la referencia de velocidad es definida de forma independiente al tiempo.

En este trabajo de tesis se utiliza la metodología para el diseño de campos de velocidad con características particulares propuesta en (Moreno, 1999).

I.4 Trabajo de tesis

- En el presente trabajo de tesis se reportan tres algoritmos de control mediante campo de velocidad, uno de ellos es inédito y fruto de las investigaciones realizadas.
- Se reportan resultados experimentales con los tres controladores en un mecanismo de laboratorio de dos grados de libertad utilizando un mismo campo vectorial de referencia.
- Se presenta un estudio comparativo de prestaciones de los tres controladores.

Capítulo II

Diseño de un campo de velocidad

En este capítulo se utiliza la metodología propuesta en (Moreno, 1999) para el diseño de un campo de velocidad. Se explican los pasos a seguir y las características deseadas de un campo de vectorial que sirva como referencia de velocidad. Finalmente se define un campo de velocidad con características particulares. Para estudiar el diseño de un campo de velocidad es necesario tener presentes algunos conceptos básicos. A continuación se definen los conceptos de campo escalar, campo vectorial y gradiente.

II.1 Definiciones

Definición 1 (Campo escalar). Un campo escalar es un mapeo $f : \mathbb{R}^n \rightarrow \mathbb{R}$ que asigna un número real a cada vector $\mathbf{x} \in \mathbb{R}^n$.

Un campo escalar simplemente asigna un valor real a cada elemento de $\mathbb{R}^n$. Por ejemplo, para $\mathbf{x} \in \mathbb{R}^2$:

$$f(\mathbf{x}) = f(x_1, x_2) = x_1^2 + x_2^2 \quad (22)$$

En la Figura 5 se muestra un conjunto de curvas de nivel para $f(\mathbf{x})$. Las curvas de nivel se forman por todos los $\mathbf{x} \in \mathbb{R}^2$ de manera que $f(\mathbf{x}) = k_i$ donde $i = 1, 2, 3, \dots$ y cada $k_i \in \mathbb{R}$.

Definición 2 (Campo vectorial). Un campo vectorial es un mapeo $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ que asigna a cada punto $\mathbf{x} \in \mathbb{R}^n$ un vector $\mathbf{f}(\mathbf{x})$.

Definición 3 (Gradiente). El gradiente es un operador vectorial denotado por ∇ . Se aplica tanto a funciones escalares como a funciones vectoriales. El gradiente de una

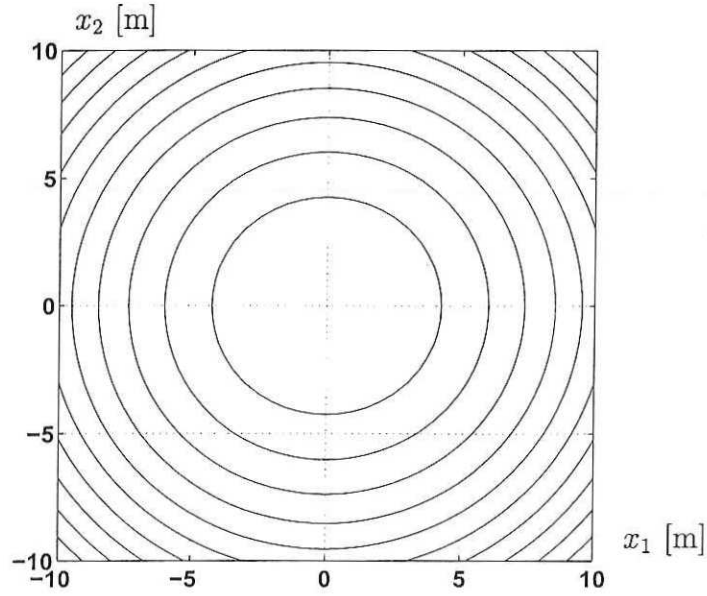


Figura 5: Campo escalar.

función escalar $f(\mathbf{x})$ donde $\mathbf{x} \in \mathbb{R}^2$, está dado por:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \end{bmatrix} \quad (23)$$

Teorema 1 (Gradiente). Para un campo escalar con primeras derivadas parciales continuas $f : \mathbb{R}^n \rightarrow \mathbb{R}$, el gradiente $\nabla f(\mathbf{x})$ existe; y si para un punto $\mathbf{x}_0 \in \mathbb{R}^n$ el gradiente es un vector no nulo, entonces éste indica la dirección de máximo crecimiento del campo escalar $f(\mathbf{x})$ en el punto $\mathbf{x} = \mathbf{x}_0$.

II.2 Características deseadas

El objetivo de control de campo de velocidad de mecanismos en coordenadas cartesianas se define como:

$$\lim_{t \rightarrow \infty} e(\mathbf{x}(t), \dot{\mathbf{x}}(t)) = \lim_{t \rightarrow \infty} [\mathbf{v}(\mathbf{x}(t)) - \dot{\mathbf{x}}(t)] = 0 \quad (24)$$

Para poder implementar un algoritmo de control mediante campo de velocidad, es necesario contar con un campo vectorial de referencia. El diseño de tal campo vectorial deberá contar con las siguientes características:

- El campo vectorial deberá estar en función de las posiciones cartesianas del extremo del manipulador x .
- Deberá estar definido en todo el espacio de trabajo del robot.
- Convendría que el campo de velocidad convergiera a una curva cerrada previamente definida.
- Lejos de la curva, el campo deberá apuntar hacia ésta.
- La convergencia hacia la curva deberá ser gradual.
- La magnitud de los vectores será tal que no se excedan los límites de par de los actuadores.
- En la curva, los vectores también deberán ser tangentes a ella.

II.3 Metodología

El diseño del campo de velocidad se lleva a cabo en tres partes (Moreno, 1999):

1. Definición de la curva a seguir.
2. Definición de un campo vectorial ortogonal a la curva.
3. Definición de un campo vectorial tangente a la curva.

Se define por Γ_x la curva a seguir dentro del espacio cartesiano. Las características con que debe contar Γ_x son (Moreno, 1999):

- Que se encuentre definida en coordenadas cartesianas.
- Que sea una curva simple, es decir, que no se cruce a sí misma.
- Que sea una curva cerrada (una elipse, una circunferencia).

A continuación se da una descripción del procedimiento propuesto en (Moreno, 1999) para el diseño de un campo vectorial.

1. Primeramente, para el caso de un robot manipulador de dos grados de libertad, es necesario definir una curva cerrada tal que¹:

$$\Gamma_{\mathbf{x}} = \{\mathbf{x} \in \mathbb{R}^2 : f(\mathbf{x}) = 0\} \quad (25)$$

donde el subíndice $\mathbf{x}$ de $\Gamma_{\mathbf{x}}$ indica que la definición de la curva es en coordenadas cartesianas.

2. En segunda instancia, se diseña una parte del campo vectorial que apunte hacia $\Gamma_{\mathbf{x}}$ desde cualquier posición dentro del espacio de trabajo.

En el diseño de esta parte del campo se considera el hecho de que:

- $f(\mathbf{x})$ evaluada adentro de la curva $\Gamma_{\mathbf{x}}$, da un valor negativo, es decir:

$$\Gamma_{-\mathbf{x}} = \{\mathbf{x} \in \mathbb{R}^2 : f(\mathbf{x}) < 0\} \quad (26)$$

donde $\Gamma_{-\mathbf{x}}$ es este conjunto de vectores $\mathbf{x}$.

- $f(\mathbf{x})$ evaluada fuera de la curva es positiva, es decir:

$$\Gamma_{+\mathbf{x}} = \{\mathbf{x} \in \mathbb{R}^2 : f(\mathbf{x}) > 0\} \quad (27)$$

donde $\Gamma_{+\mathbf{x}}$ es este conjunto de vectores $\mathbf{x}$.

Considerando que $\nabla f(\mathbf{x}) \neq 0$, $\nabla f(\mathbf{x})$ será un vector ortogonal a $f(\mathbf{x})$ y además apuntará hacia el exterior del espacio cartesiano a partir del centro de $f(\mathbf{x})$, pues es la dirección de mayor crecimiento de $f(\mathbf{x})$.

Un campo vectorial que apunte en todo momento hacia la curva cerrada dada por $\Gamma_{\mathbf{x}}$ es:

$$\hat{\mathbf{v}}(\mathbf{x}) = -k \frac{f(\mathbf{x})}{|f(\mathbf{x})|} \frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|} \quad (28)$$

donde k es una constante estrictamente positiva.

El comportamiento de $\mathbf{v}(\mathbf{x})$ es descrito a continuación:

¹En el presente trabajo de tesis, se implementan algoritmos en un robot de dos grados de libertad.

- Si $\mathbf{x} \in \Gamma_{-\mathbf{x}}$ entonces, $f(\mathbf{x}) < 0$ y por tanto habrá una doble inversión de signo, y los vectores apuntarán hacia la curva.
- Cuando $\mathbf{x} \in \Gamma_{\mathbf{x}}$, se tiene que $f(\mathbf{x}) = 0$, por lo tanto cuando $\mathbf{x} \in \Gamma_{\mathbf{x}}$ se tendrá que $\hat{\mathbf{v}}(\mathbf{x}) = 0$.
- Si $\mathbf{x} \in \Gamma_{+\mathbf{x}}$ entonces, $f(\mathbf{x}) > 0$ y por tanto habrá una sola inversión de signo, y los vectores apuntarán de nuevo hacia la curva.

La magnitud de los vectores de $\hat{\mathbf{v}}(\mathbf{x})$ dependerá de k ya que tanto $\frac{f(\mathbf{x})}{|f(\mathbf{x})|}$ como $\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}$ son vectores unitarios.

3. Finalmente es necesario diseñar una sección del campo vectorial que sea tangencial a la curva $\Gamma_{\mathbf{x}}$. Para obtener tal característica, se propone un vector $\mathbf{v}_{\mathcal{T}}(\mathbf{x})$ tal que satisfaga:

$$[\nabla f(\mathbf{x})]^T \mathbf{v}_{\mathcal{T}}(\mathbf{x}) = 0 \quad (29)$$

De modo que si $\mathbf{v}_{\mathcal{T}}(\mathbf{x})$ satisface la ecuación anterior es porque es ortogonal² a $\nabla f(\mathbf{x})$, y por tanto $\mathbf{v}_{\mathcal{T}}(\mathbf{x})$ será tangencial a $\Gamma_{\mathbf{x}}$.

De la misma forma que con $\hat{\mathbf{v}}(\mathbf{x})$, se desea tener un control sobre la magnitud de los vectores. En concreto, puede desearse que el desplazamiento sobre la curva $\Gamma_{\mathbf{x}}$ sea a velocidad constante. Por lo tanto el campo vectorial paralelo y tangencial a la trayectoria cerrada $\Gamma_{\mathbf{x}}$, con velocidad constante está dado por:

$$\tilde{\mathbf{v}}(\mathbf{x}) = c \frac{\mathbf{v}_{\mathcal{T}}(\mathbf{x})}{\|\mathbf{v}_{\mathcal{T}}(\mathbf{x})\|} \quad (30)$$

donde $c \in \mathbb{R}$.

De acuerdo con (Moreno, 1999), un campo de velocidad que converge a una curva deseada $\Gamma_{\mathbf{x}}$, está dado por:

$$\mathbf{v}(\mathbf{x}) = \hat{\mathbf{v}}(\mathbf{x}) + \tilde{\mathbf{v}}(\mathbf{x}) \quad (31)$$

donde $\hat{\mathbf{v}}(\mathbf{x})$ y $\tilde{\mathbf{v}}(\mathbf{x})$ son las componentes tangencial y normal del campo vectorial respectivamente.

²Dos vectores $\mathbf{u}$ y $\mathbf{v}$ distintos de cero, son ortogonales o perpendiculares si y sólo si $\mathbf{u}^T \mathbf{v} = 0$.

II.4 Diseño

La formulación del control de velocidad de mecanismos requiere contar con un campo vectorial deseado. De acuerdo a lo descrito en la sección II.3 es conveniente elegir una curva cerrada para tal diseño. La ecuación:

$$f(\mathbf{x}) = [(x_1 - x_{1c})^2 + (x_2 - x_{2c})^2 + a^2]^2 - 4a^2(x_1 - x_{1c})^2 - b^4 \quad (32)$$

define una familia de curvas cerradas conocidas como ovalos de Cassini, donde x_{1c} y x_{2c} son el centro de las curvas y los parámetros a y b definen la forma de éstas.

En la ecuación (32) que define los ovalos de Cassini, si a es mucho mayor que b , $f(\mathbf{x})$ dibujará una curva elíptica (Figura 6-a), a medida que a se acerque a b la forma se irá achatando hacia su centro por el eje horizontal (Figura 6-b). Si $a = b$, $f(\mathbf{x})$ define una lemniscata (Figura 6-c). Finalmente cuando $a < b$, $f(\mathbf{x})$ define un par de curvas cerradas, tal como se muestra en la Figura 6-d.

Para el diseño de un campo de velocidad convergente al ovalo de Cassini, definimos:

$$\Gamma_{\mathbf{x}} = \{\mathbf{x} \in \mathbb{R}^2 : f(\mathbf{x}) = 0\} \quad (33)$$

$$= \{\mathbf{x} \in \mathbb{R}^2 : [(x_1 - x_{1c})^2 + (x_2 - x_{2c})^2 + a^2]^2 - 4a^2(x_1 - x_{1c})^2 - b^4 = 0\} \quad (34)$$

con $a = 0.1125$, $b = 0.110$, $x_{1c} = 0.180$ y $x_{2c} = -0.180$. En la Figura 7 se observa tal curva.

La componente normal de un campo de velocidad se obtiene mediante:

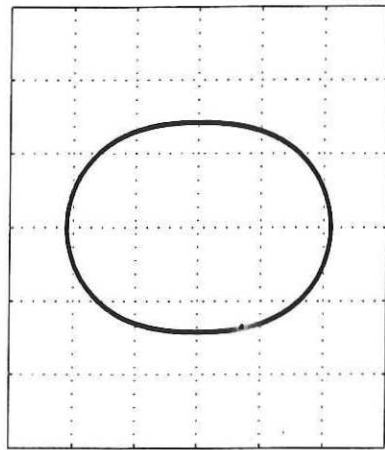
$$\hat{\mathbf{v}}(\mathbf{x}) = -k \frac{f(\mathbf{x})}{|f(\mathbf{x})|} \frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\| + \epsilon} \quad (35)$$

donde se agrega ϵ pequeño para evitar una división por cero.

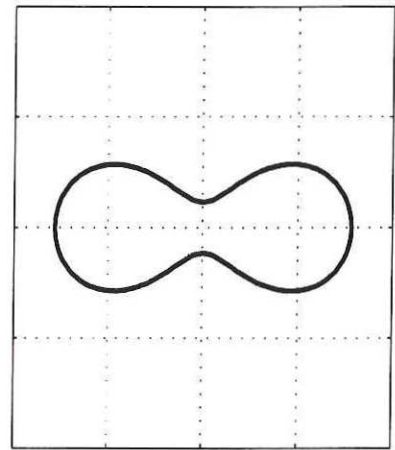
En la ecuación (35) se involucra el cálculo del gradiente de $f(\mathbf{x})$. Éste viene dado por:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} 4[x_1 - x_{1c}] [(x_1 - x_{1c})^2 + (x_2 - x_{2c})^2] - 4a^2[x_1 - x_{1c}] \\ 4[x_2 - x_{2c}] [(x_1 - x_{1c})^2 + (x_2 - x_{2c})^2] + 4a^2[x_2 - x_{2c}] \end{bmatrix} \quad (36)$$

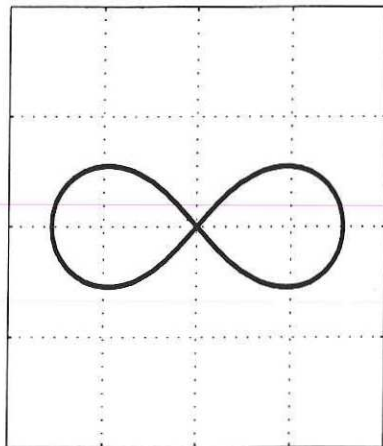
En la Figura 8 se muestra la componente normal del campo de velocidad para $k = 1.0$ y $\epsilon = 1 \times 10^{-6}$.



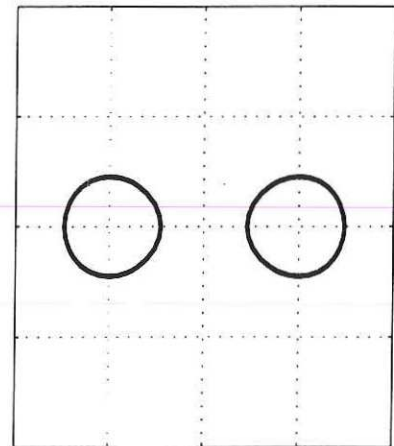
(a)



(b)



(c)



(d)

Figura 6: Ovalos de Cassini.

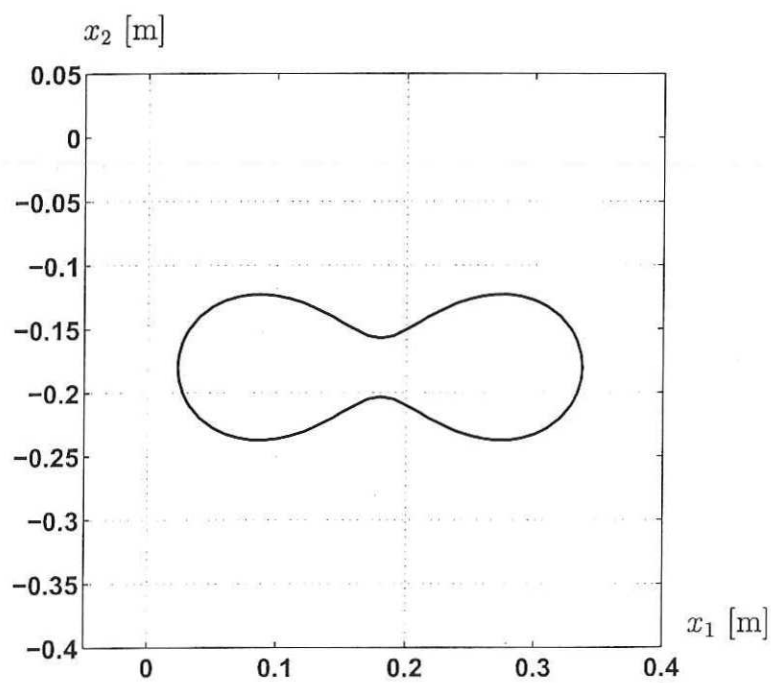


Figura 7: Ovalo de Cassini.

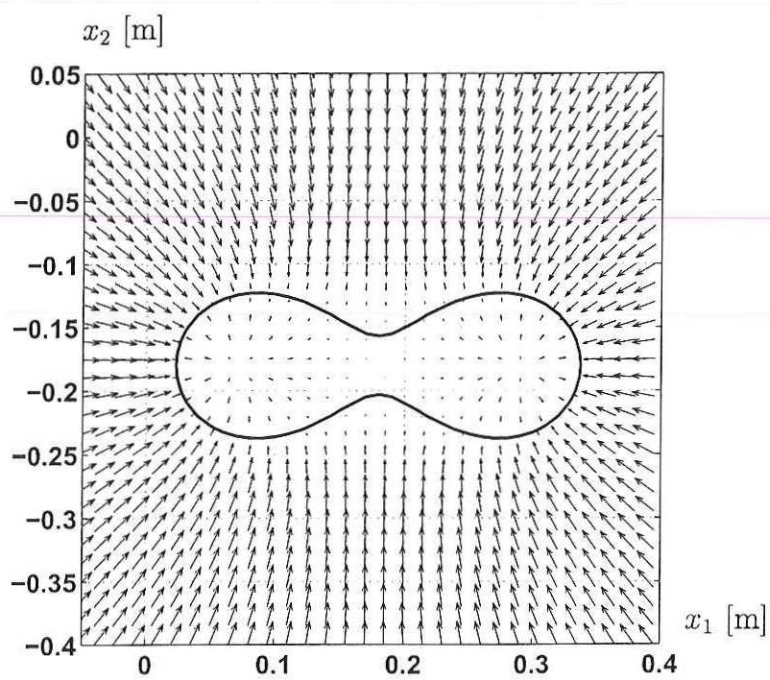


Figura 8: Componente normal del campo vectorial.

La componente tangencial del campo vectorial, se busca a partir de:

$$[\nabla f(\mathbf{x})]^T \mathbf{v}_T(\mathbf{x}) = 0 \quad (37)$$

Una posible solución de la ecuación anterior es:

$$\mathbf{v}_T(\mathbf{x}) = \begin{bmatrix} -4[x_2 - x_{2c}] [[x_1 - x_{1c}]^2 + [x_2 - x_{2c}]^2] - 4a^2[x_2 - x_{2c}] \\ 4[x_1 - x_{1c}] [[x_1 - x_{1c}]^2 + [x_2 - x_{2c}]^2] - 4a^2[x_1 - x_{1c}] \end{bmatrix} \quad (38)$$

Por tanto la componente tangencial de $\mathbf{v}(\mathbf{x})$ estará dada por;

$$\tilde{\mathbf{v}}(\mathbf{x}) = c \frac{\mathbf{v}_T(\mathbf{x})}{\|\mathbf{v}_T(\mathbf{x})\|} \quad (39)$$

en donde c será la velocidad de desplazamiento sobre la curva. Con fines de implementación se agregará una modificación a la componente tangencial de campo de velocidad diseñado. Se desea que la componente tangencial del campo se aproxime a cero cuando sea evaluada lejos de la curva a seguir. Para ello se define $\tilde{\mathbf{v}}(\mathbf{x})$ como:

$$\tilde{\mathbf{v}}(\mathbf{x}) = c \frac{\mathbf{v}_T(\mathbf{x})}{\|\mathbf{v}_T(\mathbf{x})\|} e^{-\alpha|f(\mathbf{x})|} \quad (40)$$

En la Figura 9 se muestra la componente tangencial de campo de velocidad. Se utiliza en el diseño los siguientes valores para los parámetros c y para α .

$$c = 0.35 \quad (41)$$

$$\alpha = 500 \quad (42)$$

De acuerdo con (Moreno, 1999), un campo de velocidad que converge a una curva deseada $\Gamma_{\mathbf{x}}$, está dado por:

$$\mathbf{v}(\mathbf{x}) = \hat{\mathbf{v}}(\mathbf{x}) + \tilde{\mathbf{v}}(\mathbf{x}) \quad (43)$$

o sea

$$\mathbf{v}(\mathbf{x}) = -k \frac{f(\mathbf{x})}{|f(\mathbf{x})|} \frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\| + \epsilon} + c \frac{\mathbf{v}_T(\mathbf{x})}{\|\mathbf{v}_T(\mathbf{x})\|} e^{-\alpha|f(\mathbf{x})|} \quad (44)$$

El diseño final de campo de velocidad se muestra en la Figura 10.

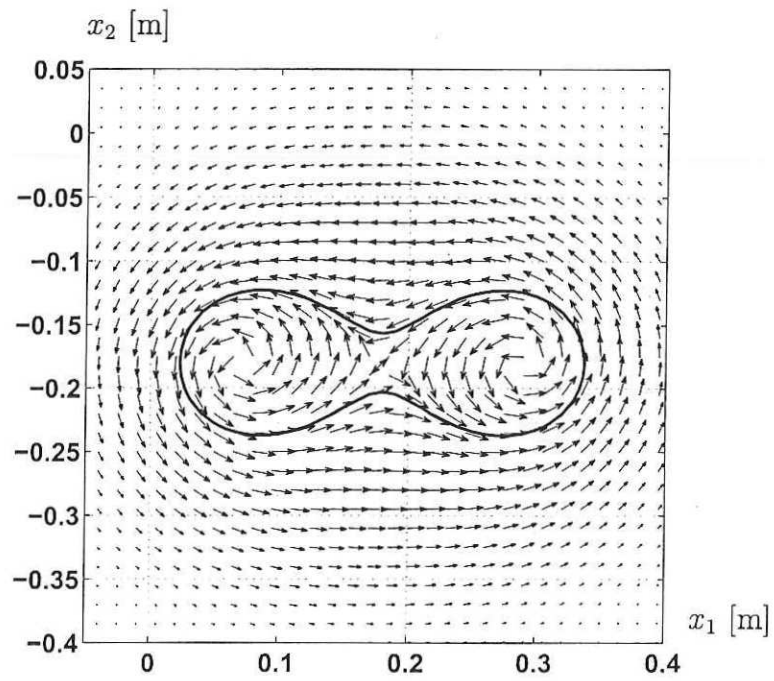


Figura 9: Componente tangencial del campo vectorial.

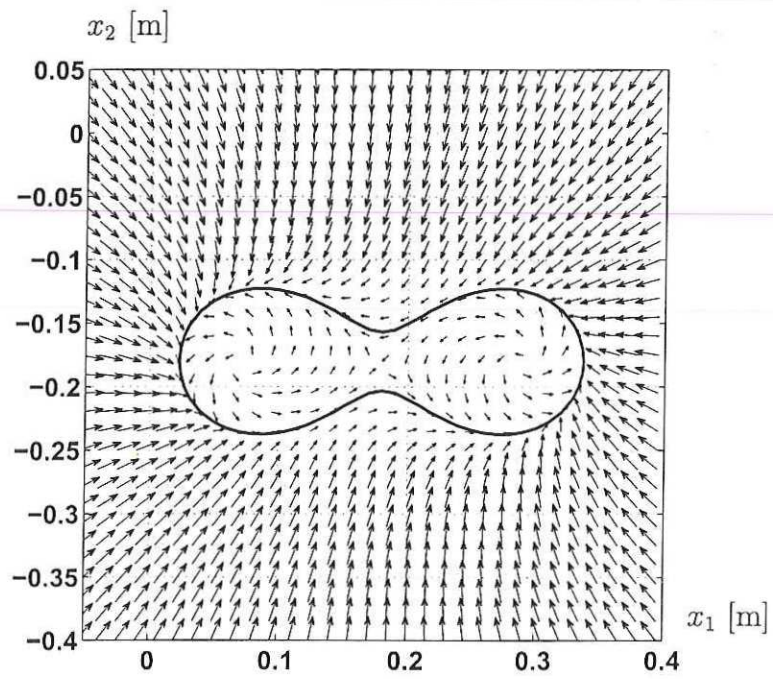


Figura 10: Campo vectorial que converge a un ovalo de Cassini.

II.5 Consideraciones prácticas

Para la programación de los algoritmos estudiados en los siguientes capítulos, es necesario calcular la derivada temporal del campo de velocidad definido por (44). La derivada temporal del campo de velocidad viene dada por:

$$\frac{d}{dt} [\mathbf{v}(\mathbf{x})] = \frac{d}{dt} [\hat{\mathbf{v}}(\mathbf{x})] + \frac{d}{dt} [\tilde{\mathbf{v}}(\mathbf{x})] \quad (45)$$

donde la derivada temporal de $\hat{\mathbf{v}}(\mathbf{x})$ es:

$$\frac{d}{dt} [\hat{\mathbf{v}}(\mathbf{x})] = \frac{d}{dt} \left[-k \frac{f(\mathbf{x})}{|f(\mathbf{x})|} \frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\| + \epsilon} \right] \quad (46)$$

$$= \frac{[|f(\mathbf{x})| \|\nabla f(\mathbf{x})\| + \epsilon] \frac{d}{dt} [-k f(\mathbf{x}) \nabla f(\mathbf{x})] - [k f(\mathbf{x}) \nabla f(\mathbf{x})] \frac{d}{dt} [|f(\mathbf{x})| \|\nabla f(\mathbf{x})\| + \epsilon]}{[|f(\mathbf{x})| \|\nabla f(\mathbf{x})\| + \epsilon]^2} \quad (47)$$

Se aproxima el cálculo de la derivada de $|f(\mathbf{x})|$ mediante la función tangente hiperbólica, es decir:

$$\frac{d}{dt} [|f(\mathbf{x})|] = \tanh(\lambda f(\mathbf{x})) \frac{d}{dt} [f(\mathbf{x})] \quad (48)$$

con λ un número real positivo grande.

Ahora calculamos la derivada temporal de $\tilde{\mathbf{v}}(\mathbf{x})$:

$$\frac{d}{dt} [\tilde{\mathbf{v}}(\mathbf{x})] = \frac{d}{dt} \left[c \frac{\mathbf{v}_{\mathcal{T}}(\mathbf{x})}{\|\mathbf{v}_{\mathcal{T}}(\mathbf{x})\|} e^{-\alpha |f(\mathbf{x})|} \right] \quad (49)$$

$$= c \left[\frac{[\|\mathbf{v}_{\mathcal{T}}(\mathbf{x})\| \frac{d}{dt} [\mathbf{v}_{\mathcal{T}}(\mathbf{x}) e^{-\alpha |f(\mathbf{x})|}]] - \mathbf{v}_{\mathcal{T}}(\mathbf{x}) e^{-\alpha |f(\mathbf{x})|} \frac{d}{dt} [\|\mathbf{v}_{\mathcal{T}}(\mathbf{x})\|]}{[\|\mathbf{v}_{\mathcal{T}}(\mathbf{x})\|]^2} \right] \quad (50)$$

donde se aproxima el cálculo de la derivada de $e^{-\alpha |f(\mathbf{x})|}$ mediante la función tangente hiperbólica:

$$\frac{d}{dt} [e^{-\alpha |f(\mathbf{x})|}] = -\alpha \tanh(\lambda f(\mathbf{x})) \frac{d}{dt} [f(\mathbf{x})] e^{-\alpha |f(\mathbf{x})|} \quad (51)$$

Capítulo III

Control basado en dinámica inversa

En este capítulo se presenta una modificación al algoritmo de control mediante campo de velocidad propuesto en (Moreno, 1999) llamado control por dinámica inversa. La modificación consiste en que el algoritmo a presentar carece de la acción integral de su antecesor. El modelo dinámico para un robot de n grados de libertad está dado por:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (52)$$

Se busca una ley de control que en lazo cerrado con (52) resulte en un nuevo sistema con propiedades dinámicas que satisfagan:

$$\lim_{t \rightarrow \infty} e(x(t), \dot{x}(t)) = \lim_{t \rightarrow \infty} [v(x(t)) - \dot{x}(t)] = 0. \quad (53)$$

III.1 Algoritmo de control basado en dinámica inversa

El concepto de dinámica inversa consiste en encontrar una ley de control $\tau(q, \dot{q})$ que al ser sustituida en la ecuación del modelo dinámico del robot manipulador (52) resulte en un sistema lineal en lazo cerrado (Spong y Vidyasagar, 1989). Se propone entonces el siguiente algoritmo de control modificado¹:

$$\tau = M(q)u_0 + C(q, \dot{q})\dot{q} + g(q) \quad (54)$$

con:

$$u_0 = J(q)^{-1} \left[\frac{\partial v(x)}{\partial x} \dot{x} + K e(x, \dot{x}) - \dot{J}(q)\dot{q} \right] \quad (55)$$

¹En (Moreno, 1999) se ha propuesto la versión original de este controlador por dinámica inversa.

donde $K = K^T > 0$, y definimos:

$$e(x, \dot{x}) = v(x) - \dot{x} \quad (56)$$

En la Figura 11 se muestra un diagrama de bloques del controlador por dinámica inversa modificado.

Igualando (52) y (54) se obtiene la expresión en lazo cerrado:

$$\ddot{q} = u_0 \quad (57)$$

Considerando el modelo cinemático directo:

$$\dot{x} = J(q)\dot{q} \quad (58)$$

y derivando con respecto al tiempo se obtiene:

$$\ddot{x} = J(q)\ddot{q} + \dot{J}(q)\dot{q} \quad (59)$$

Ahora despejamos $\ddot{q}$ de (59) y se obtiene:

$$J(q)\ddot{q} = \ddot{x} - \dot{J}(q)\dot{q}$$

$$\ddot{q} = J(q)^{-1} [\ddot{x} - \dot{J}(q)\dot{q}] \quad (60)$$

De acuerdo a (57) igualamos (55) y (60) para obtener:

$$J(q)^{-1} [\ddot{x} - \dot{J}(q)\dot{q}] = J(q)^{-1} \left[\frac{\partial v(x)}{\partial x} \dot{x} + K e(x, \dot{x}) - \dot{J}(q)\dot{q} \right] \quad (61)$$

Buscamos ahora reescribir en términos de e :

$$\frac{\partial v(x)}{\partial x} \dot{x} - \ddot{x} + K e(x, \dot{x}) = 0 \quad (62)$$

$$\dot{v}(x) - \ddot{x} + K e(x, \dot{x}) = 0 \quad (63)$$

$$\dot{e}(x, \dot{x}) + K e(x, \dot{x}) = 0 \quad (64)$$

quedando en la ecuación (64) un sistema de ecuaciones diferenciales de primer orden en términos de $e(x, \dot{x})$.

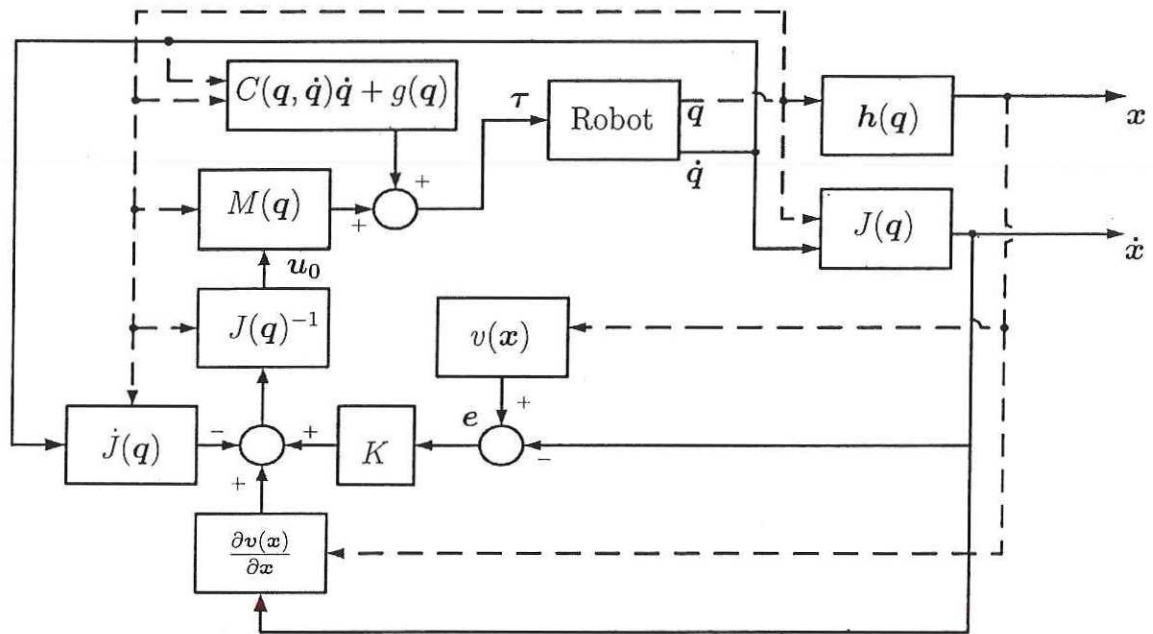


Figura 11: Diagrama de bloques del controlador por dinámica inversa.

III.1.1 Análisis de estabilidad

Para llevar a cabo el análisis formal de estabilidad se propone la siguiente función de Lyapunov:

$$V(e(x, \dot{x})) = \frac{1}{2} e(x, \dot{x})^T e(x, \dot{x}) \quad (65)$$

Calculando la derivada temporal de la función de Lyapunov se obtiene:

$$\dot{V}(e(x, \dot{x})) = e(x, \dot{x})^T \dot{e}(x, \dot{x}) \quad (66)$$

$$= e(x, \dot{x})^T (-K e(x, \dot{x})) \quad (67)$$

$$= -e(x, \dot{x})^T K e(x, \dot{x}) \quad (68)$$

La derivada de la función de candidata de Lyapunov a resultado negativa para todo $e(x, \dot{x}) \neq 0$ y $\dot{V}(0) = 0$ por lo que se ha probado estabilidad asintótica global del origen del sistema descrito en (64).

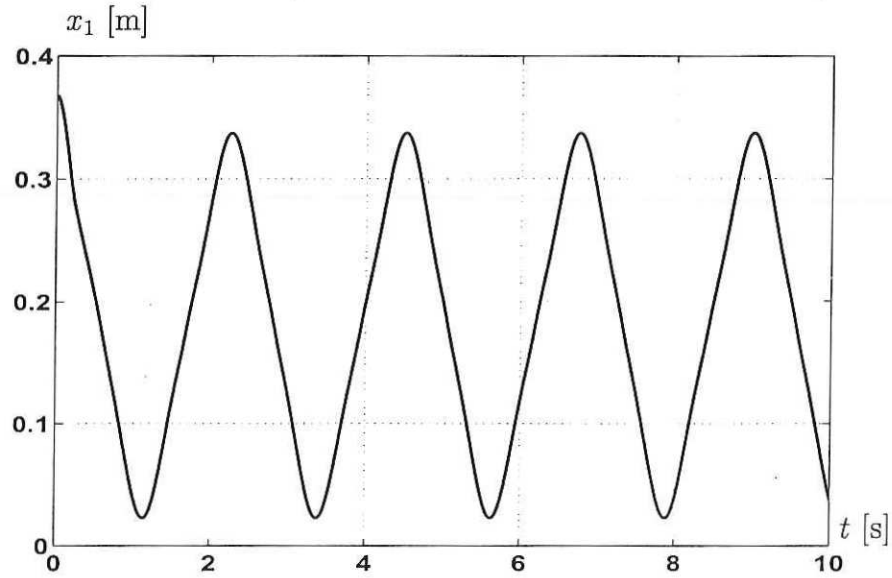


Figura 12: Trayectoria del extremo del robot correspondiente al eje cartesiano x_1 . [dinámica inversa - simulación]

III.2 Simulaciones

A continuación se muestran los resultados de las simulaciones hechas previas a la experimentación con el controlador por dinámica inversa modificado. El algoritmo de control por dinámica inversa modificado está dado por:

$$\tau = M(q)u_0 + C(q, \dot{q})\dot{q} + g(q) \quad (69)$$

con:

$$u_0 = J(q)^{-1} \left[\frac{\partial v(x)}{\partial x} \dot{x} + K e(x, \dot{x}) - \dot{J}(q)\dot{q} \right] \quad (70)$$

donde K es una matriz diagonal y definida positiva.

Se generaron simulaciones utilizando Simnon v.2.0 y el algoritmo de integración de Runge-Kutta-Felhberg 4/5, considerando el sistema dinámico como continuo en su totalidad².

Las condiciones iniciales del robot en coordenadas cartesianas son:

$$x_1(0) = 0.3677 [m] \quad (71)$$

$$x_2(0) = 0.0 [m] \quad (72)$$

²Las condiciones de compensación de fricción así como la sintonización de ganancias utilizadas en las simulaciones presentadas en esta tesis son las mismas que las utilizadas en los experimentos.

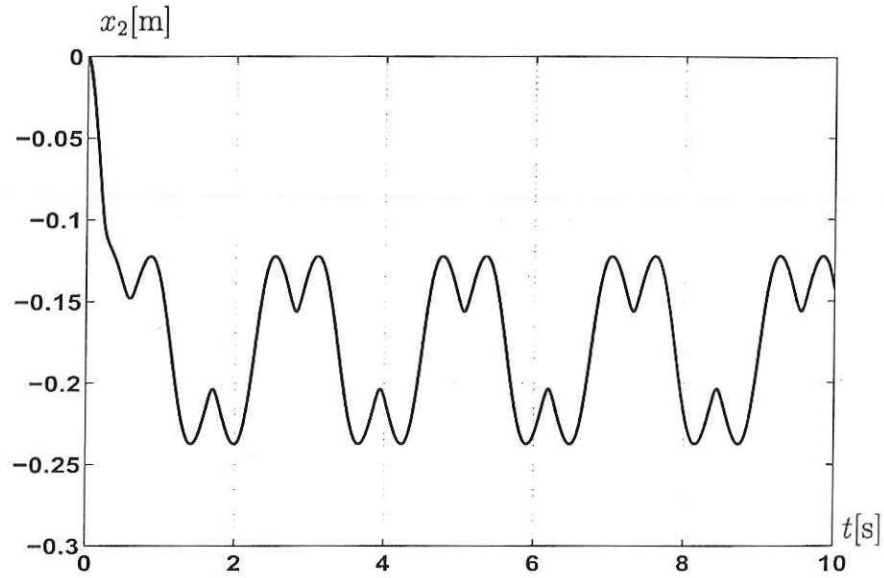


Figura 13: Trayectoria del extremo del robot correspondiente al eje cartesiano x_2 . [dinámica inversa - simulación]

o en coordenadas articulares:

$$q_1(0) = \pi/4 \text{ [rad]} \quad (73)$$

$$q_2(0) = \pi/2 \text{ [rad]} \quad (74)$$

En la Figura 12 se grafica la trayectoria del extremo del robot correspondiente al eje cartesiano x_1 así como en la Figura 13 se grafica la trayectoria del extremo del robot correspondiente al eje cartesiano x_2 .

El campo de velocidad $\mathbf{v}(\mathbf{x})$ fué diseñado para converger a la curva descrita por Γ_x , en este caso, al óvalo de Cassini. En la Figura 14 se grafica la trayectoria del extremo del robot en el espacio cartesiano x_1 - x_2 .

El campo de velocidad diseñado converge a Γ_x y sigue esta curva a una velocidad constante $c = 0.35 \text{ [m/s]}$.

En la Figura 15 se grafica la componente horizontal de la velocidad del extremo del robot así como en la Figura 16 se grafica la componente vertical de la velocidad del extremo del robot.

La norma $\|\dot{\mathbf{x}}\|$ se calcula a través de:

$$\|\dot{\mathbf{x}}\| = \sqrt{\dot{x}_1^2 + \dot{x}_2^2} \quad (75)$$

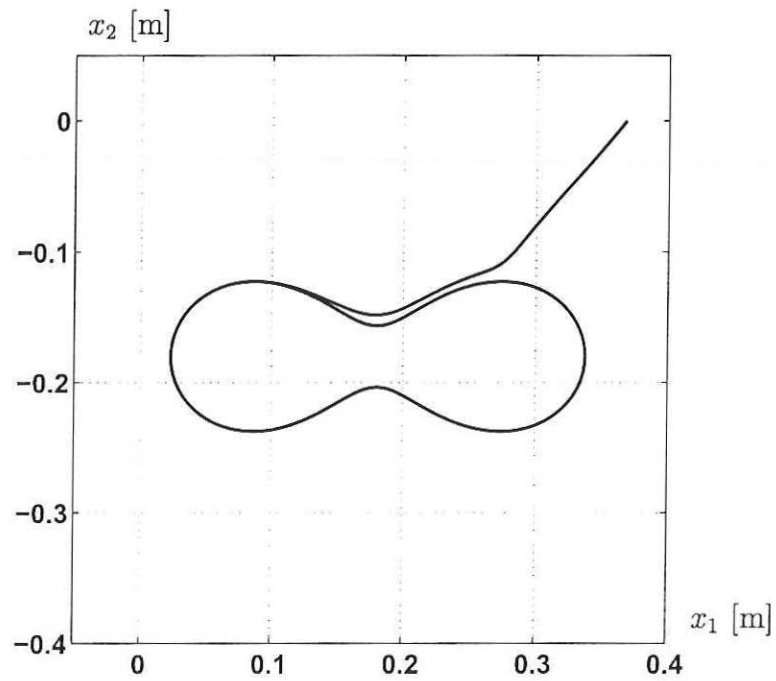


Figura 14: Trayectoria realizada por el extremo del robot en el espacio cartesiano x_1 - x_2 .
[dinámica inversa - simulación]

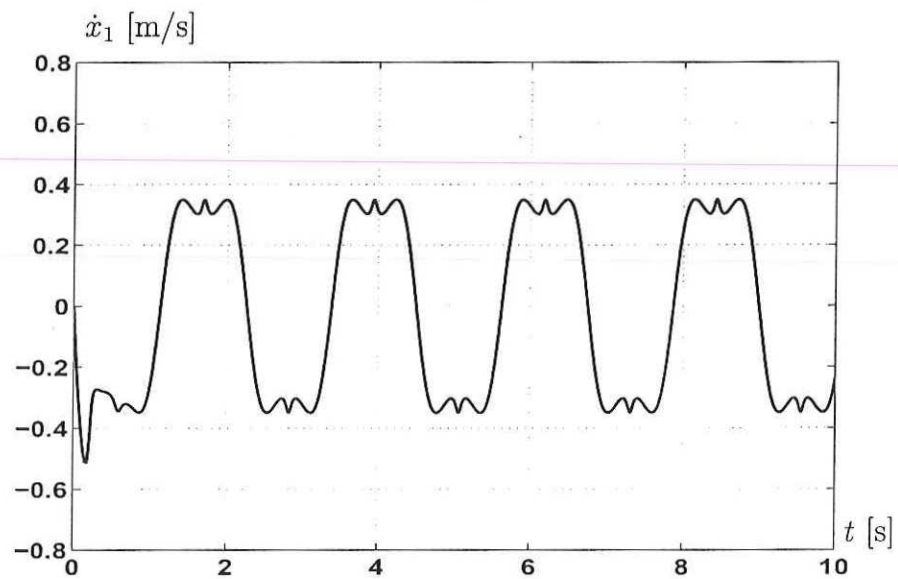


Figura 15: Velocidad del extremo del robot correspondiente al eje cartesiano x_1 .
[dinámica inversa - simulación]

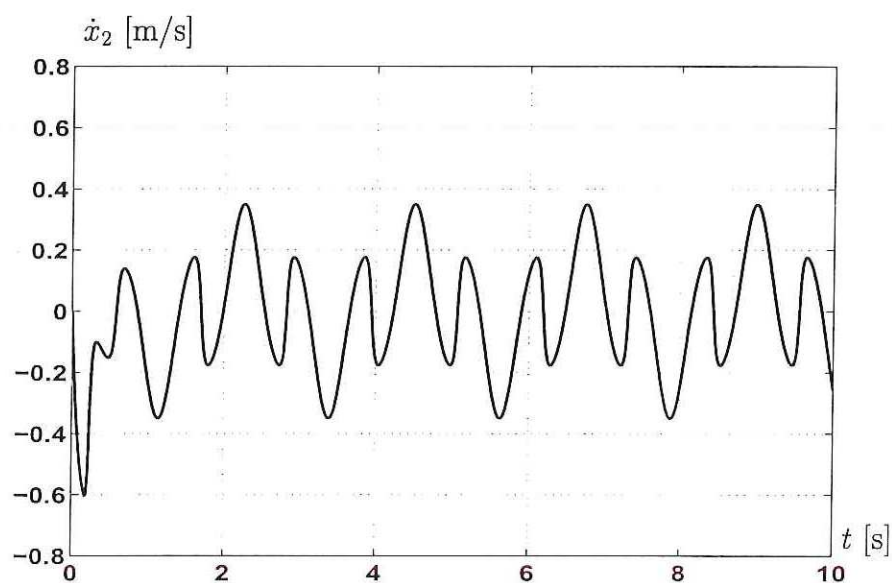


Figura 16: Velocidad del extremo del robot correspondiente al eje cartesiano x_2 . [dinámica inversa - simulación]

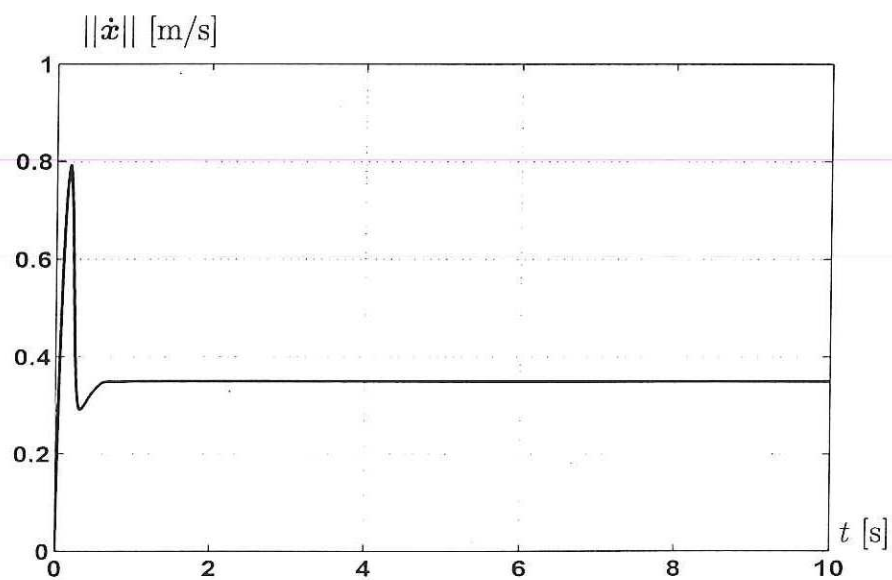


Figura 17: Norma de la velocidad del extremo del robot. [dinámica inversa - simulación]

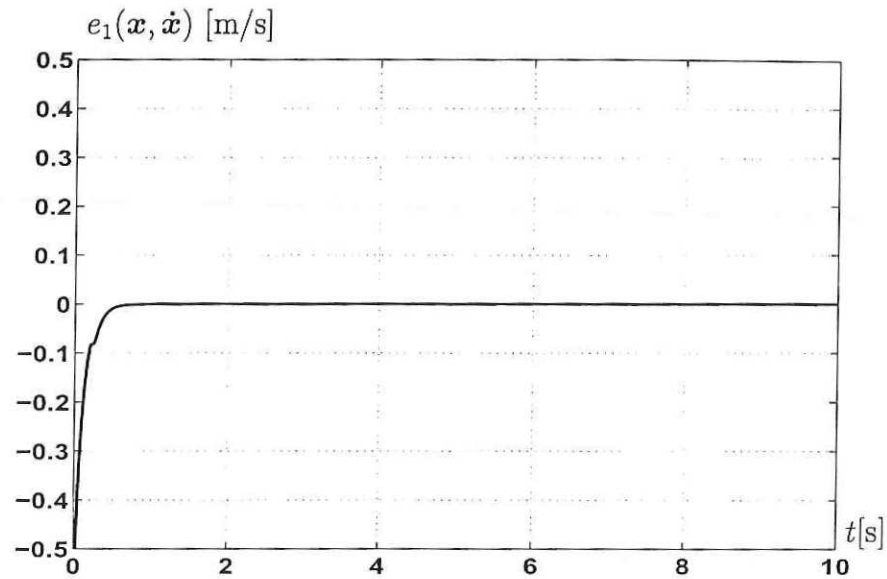


Figura 18: Error de campo de velocidad que corresponde al eje cartesiano x_1 . [dinámica inversa - simulación]

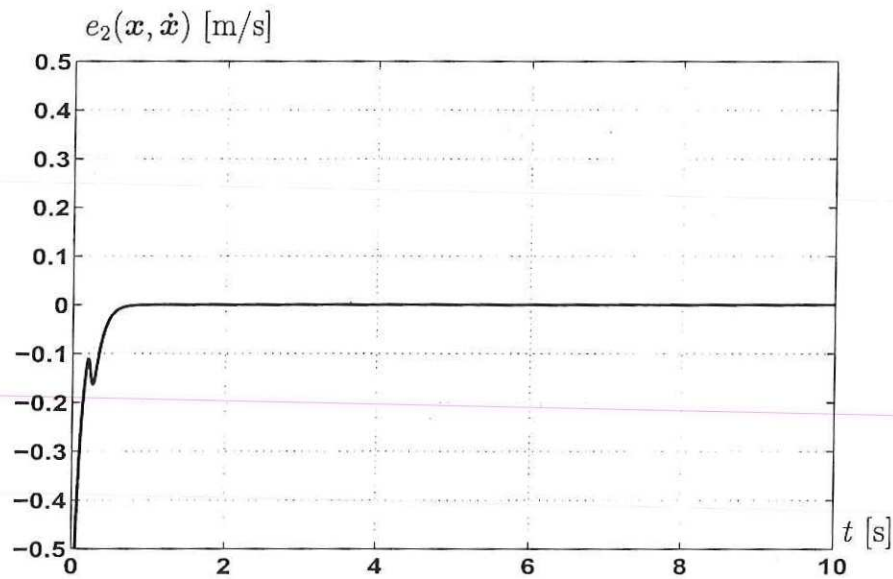


Figura 19: Error de campo de velocidad que corresponde al eje cartesiano x_2 . [dinámica inversa - simulación]

En la Figura 17 se grafica la norma $\|\dot{x}\|$ del extremo del robot; se puede observar como esta converge a 0.35 [m/s] como es deseado.

En la Figura 18 se observa el error de campo de velocidad correspondiente al eje

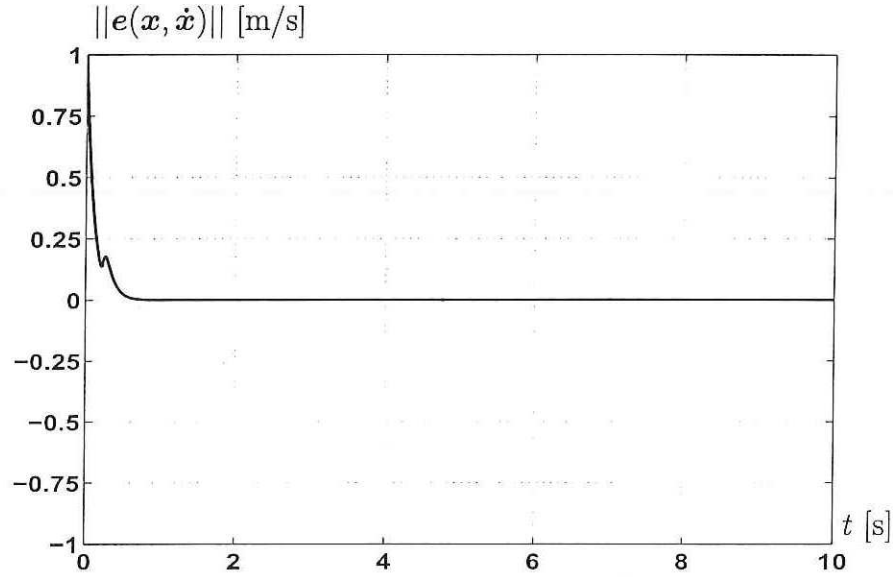


Figura 20: Norma del error de campo de velocidad. [dinámica inversa - simulación]

cartesiano x_1 :

$$e_1(\mathbf{x}, \dot{\mathbf{x}}) = v_1(\mathbf{x}) - \dot{x}_1 \quad (76)$$

y en la Figura 19 se observa el error de campo de velocidad correspondiente al eje cartesiano x_2 :

$$e_2(\mathbf{x}, \dot{\mathbf{x}}) = v_2(\mathbf{x}) - \dot{x}_2 \quad (77)$$

Se calcula la norma $\|e(\mathbf{x}, \dot{\mathbf{x}})\|$ utilizando:

$$\|e(\mathbf{x}, \dot{\mathbf{x}})\| = \sqrt{[e_1(\mathbf{x}, \dot{\mathbf{x}})]^2 + [e_2(\mathbf{x}, \dot{\mathbf{x}})]^2} \quad (78)$$

La norma $\|e(\mathbf{x}, \dot{\mathbf{x}})\|$ se muestra en la Figura 20, se observa convergencia asintótica a cero de la norma del error de campo de velocidad.

Por último se estudió el desempeño de los pares de control. Es importante mantener estos dentro de los límites físicos de los actuadores del robot.

Los límites de los actuadores en el robot Bugarín son:

$$\tau_1 = \pm 15 [Nm] \quad (79)$$

$$\tau_2 = \pm 4 [Nm] \quad (80)$$

$$\dot{q}_1 = 900 [^\circ/seg] \quad (81)$$

$$\dot{q}_2 = 720 [^\circ/seg] \quad (82)$$

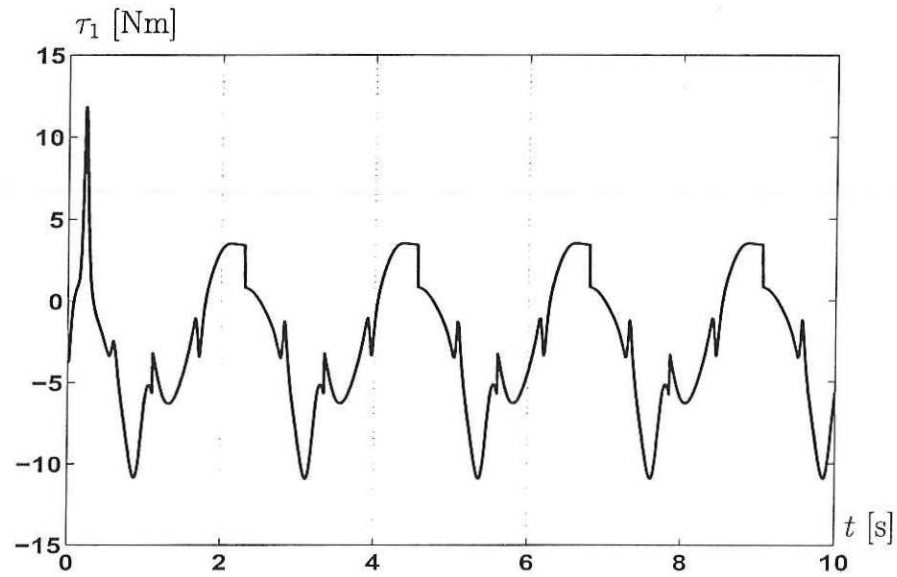


Figura 21: Par aplicado a la articulación 1 del robot. [dinámica inversa - simulación]

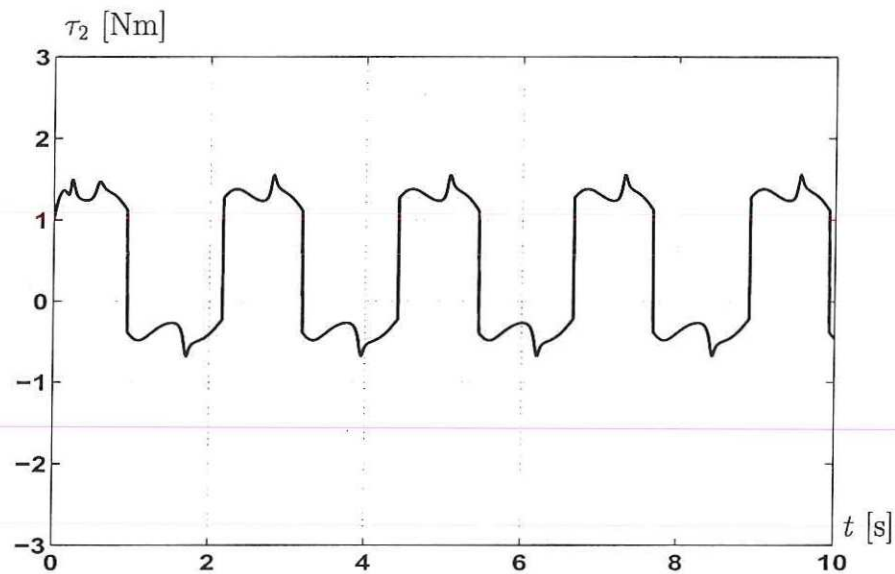


Figura 22: Par aplicado a la articulación 2 del robot. [dinámica inversa - simulación]

También es importante mantenerlos libres de componentes de alta frecuencia los cuales pudieran empobrecer el desempeño del controlador en experimentación.

En la Figura 21 se grafica el par correspondiente a la primera articulación y en la Figura 22 se observa el par correspondiente a la segunda articulación.

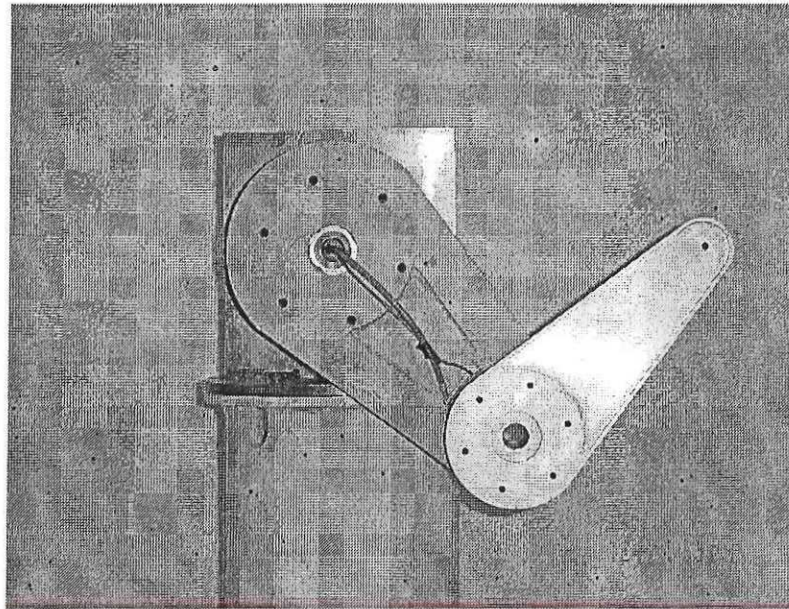


Figura 23: Robot Bugarín.

III.3 Experimentos

Se implementó el algoritmo por dinámica inversa modificado en el robot Bugarín disponible en el laboratorio de Robótica³. El robot de Bugarín descrito en (Moreno, 2002) es un mecanismo de dos grados de libertad, con articulaciones rotacionales. En la Figura 23 se presenta el robot de Bugarín.

El sistema robótico experimental nos permite acceder a mediciones de las posiciones articulares q_1 y q_2 del robot. Las velocidades articulares $\dot{q}_1$ y $\dot{q}_2$ son estimadas utilizando el algoritmo de Euler:

$$\dot{q}_k = \frac{q[k+1] - q[k]}{h} \quad (83)$$

donde h es el periodo de muestreo ($h = 0.001$ [s]).

En experimentación se utilizó compensación de fricción viscosa y de Coulomb, es decir, se consideró el modelo dinámico del robot:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + F_v\dot{q} + F_c \text{signo}(\dot{q}) = \tau \quad (84)$$

³División de Física Aplicada, CICESE

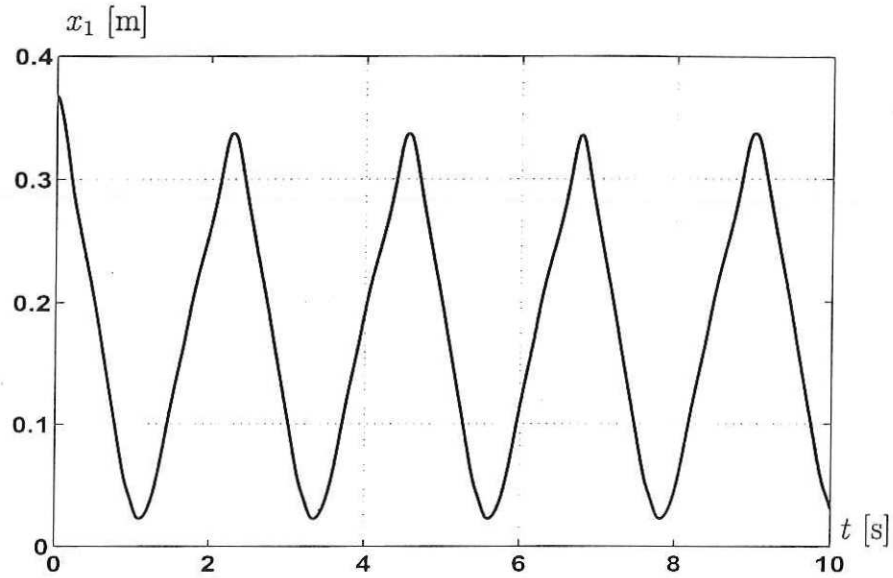


Figura 24: Trayectoria del extremo del robot correspondiente al eje cartesiano x_1 . [dinámica inversa - experimento]

donde:

$$F_v = \begin{bmatrix} 0.274 & 0 \\ 0 & 0.144 \end{bmatrix} \quad (85)$$

es la matriz de coeficientes de fricción viscosa y:

$$F_c = \begin{bmatrix} 1.290 & 0 \\ 0 & 0.745 \end{bmatrix} \quad (86)$$

es la matriz de coeficientes de fricción de Coulomb.

Las ganancias de la matriz K utilizadas en el experimento fueron:

$$K = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \quad (87)$$

y las condiciones iniciales fueron las mismas que las tomadas en simulación.

En las Figuras 24 y 25 se muestran las trayectorias realizadas por el extremo del robot correspondientes a los ejes cartesianos x_1 y x_2 , respectivamente. En la Figura 26 se muestra la trayectoria realizada por el extremo del robot en el espacio cartesiano x_1 - x_2 bajo control mediante campo de velocidad, utilizando el algoritmo de control mediante dinámica inversa modificado.

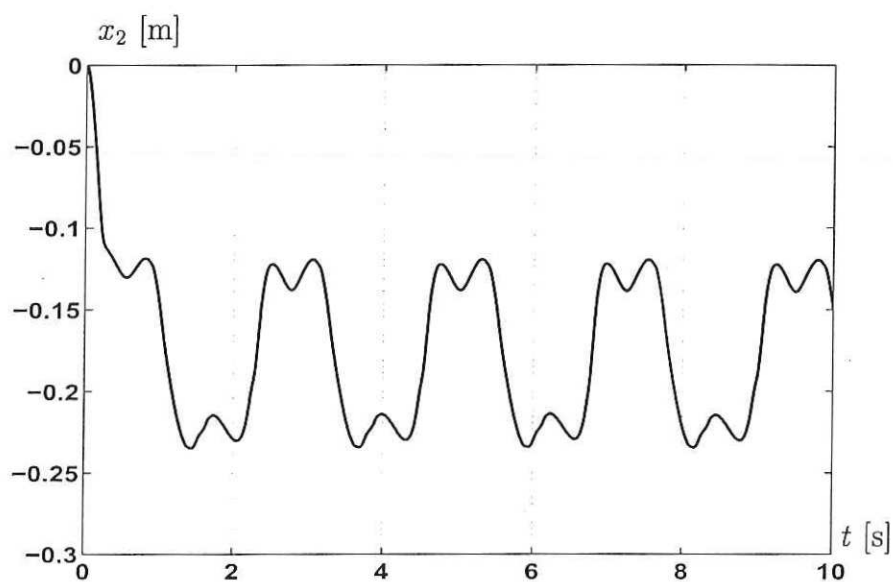


Figura 25: Trayectoria del extremo del robot correspondiente al eje cartesiano x_2 .
[dinámica inversa - experimento]

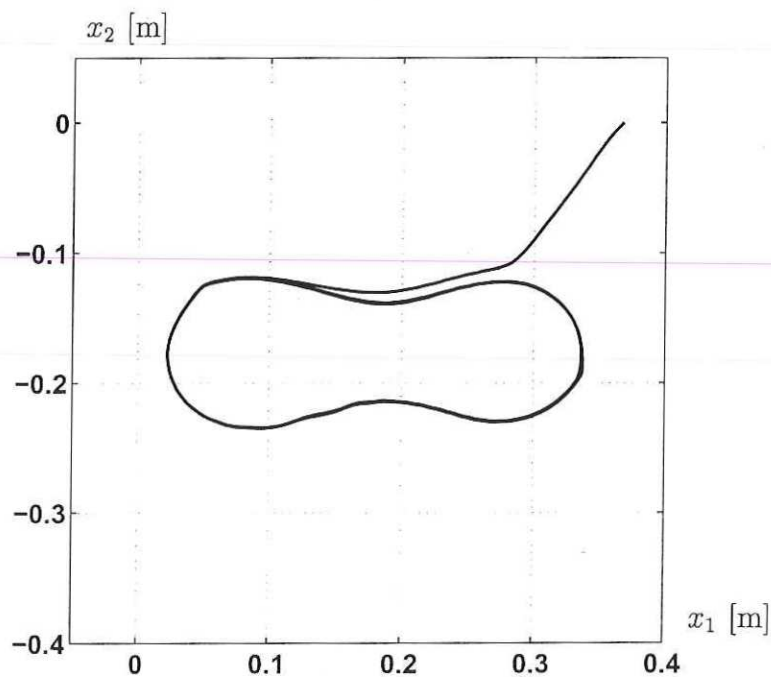


Figura 26: Trayectoria realizada por el extremo del robot en el espacio cartesiano x_1 - x_2 .
[dinámica inversa - experimento]

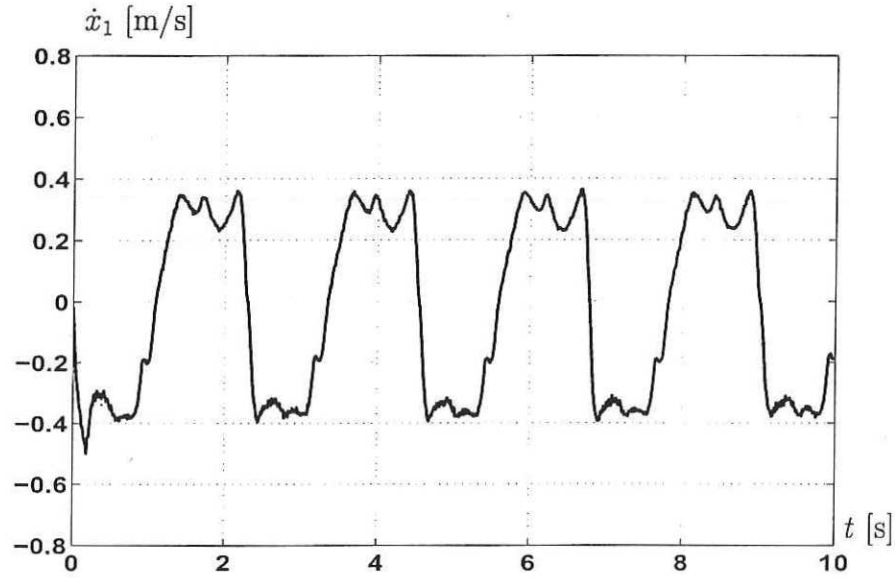


Figura 27: Velocidad del extremo del robot correspondiente al eje cartesiano x_1 . [dinámica inversa - experimento]

De acuerdo con el diseño de campo de velocidad, el extremo del robot debe converger a una curva definida por Γ_x a una velocidad constante de $c = 0.35$ [m/s].

En la Figura 27 se observa la velocidad correspondiente a eje cartesiano x_1 y en la Figura 28 se observa la velocidad correspondiente a eje cartesiano x_2 . En la Figura 29 se observa $\|\dot{\mathbf{x}}\|$. Se aprecian oscilaciones entre 0.2 y 0.65 [m/s].

En las Figuras 30 y 31 se presenta el error de campo de velocidad correspondiente a los ejes cartesianos x_1 y x_2 , respectivamente. En la Figura 32 se muestra la norma del error de campo de velocidad $\|e(\mathbf{x}, \dot{\mathbf{x}})\|$.

Finalmente en la Figura 33 se muestra el par aplicado a la primera articulación así como en la Figura 34 se muestra el par aplicado a la segunda articulación.

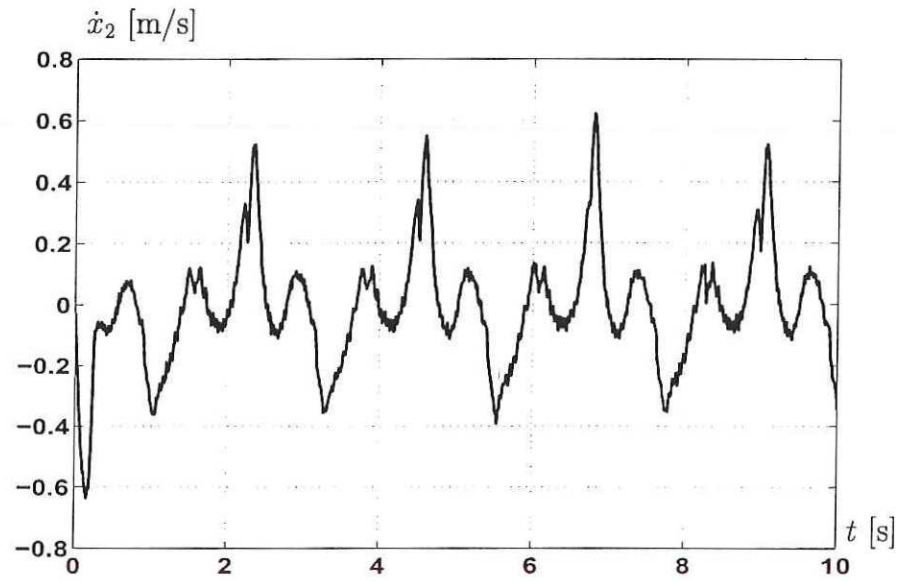


Figura 28: Velocidad del extremo del robot correspondiente al eje cartesiano x_2 . [dinámica inversa - experimento]

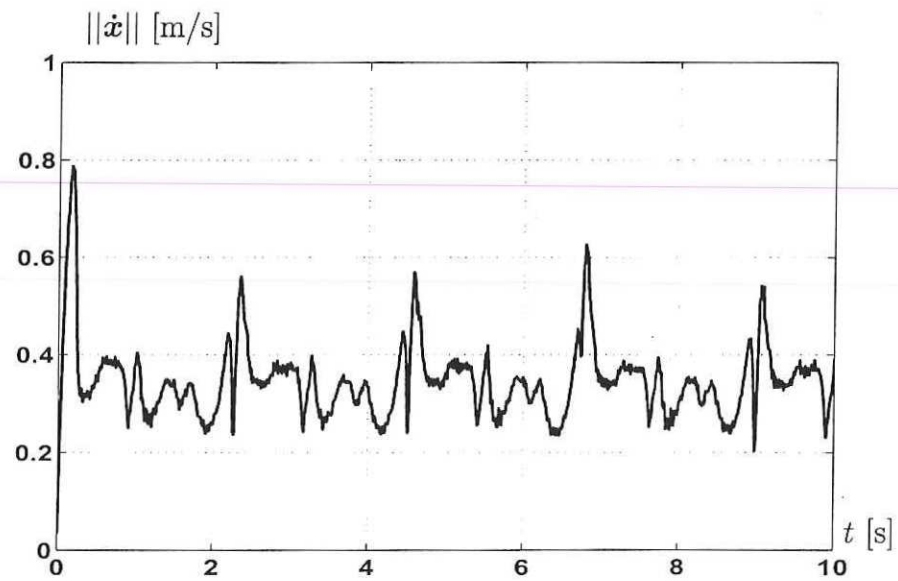


Figura 29: Norma de la velocidad del extremo del robot. [dinámica inversa - experimento]

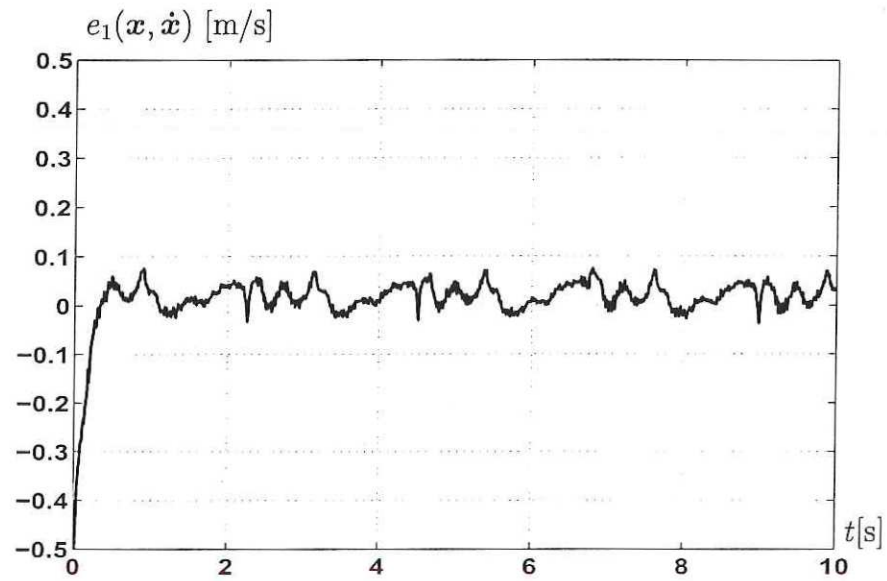


Figura 30: Error de campo de velocidad que corresponde al eje cartesiano x_1 . [dinámica inversa - experimento]

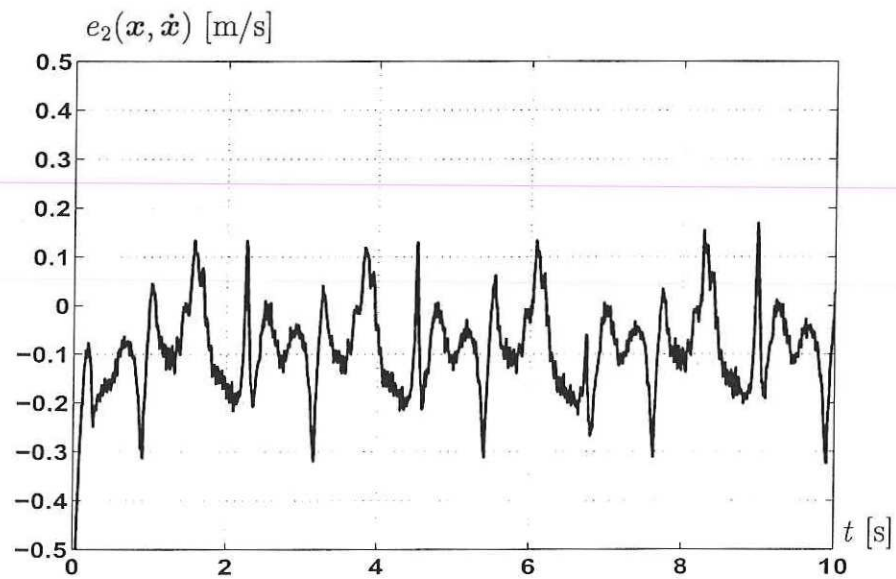


Figura 31: Error de campo de velocidad que corresponde al eje cartesiano x_2 . [dinámica inversa - experimento]

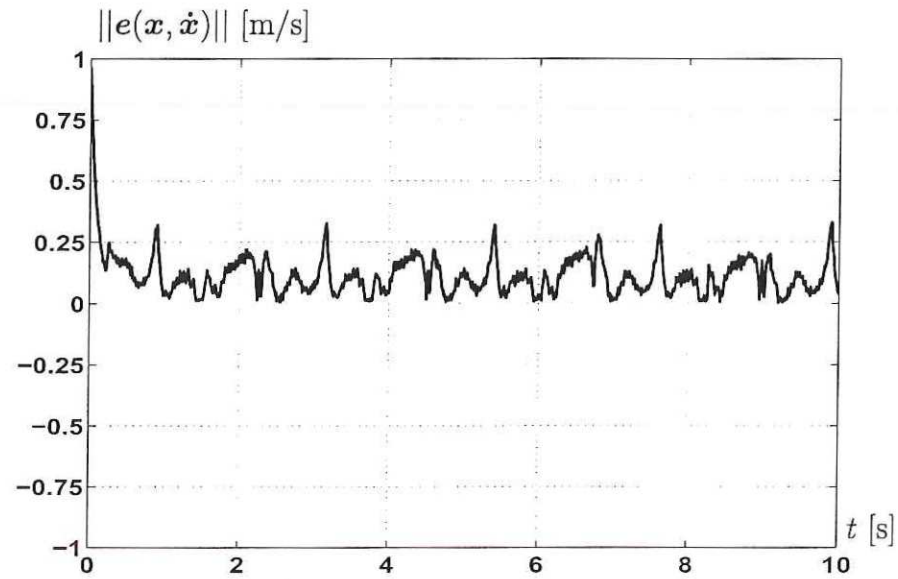


Figura 32: Norma del error de campo de velocidad. [dinámica inversa - experimento]

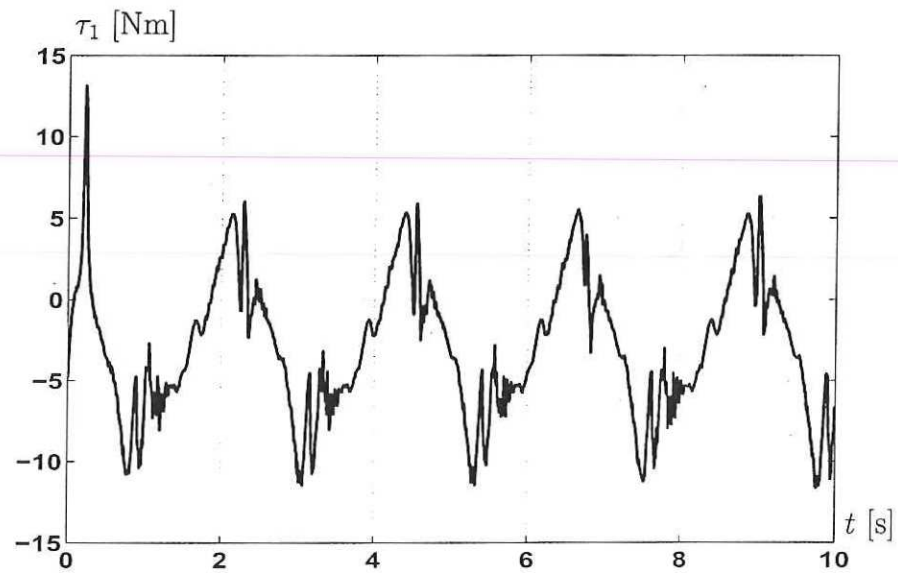


Figura 33: Par aplicado a la articulación 1 del robot. [dinámica inversa - experimento]

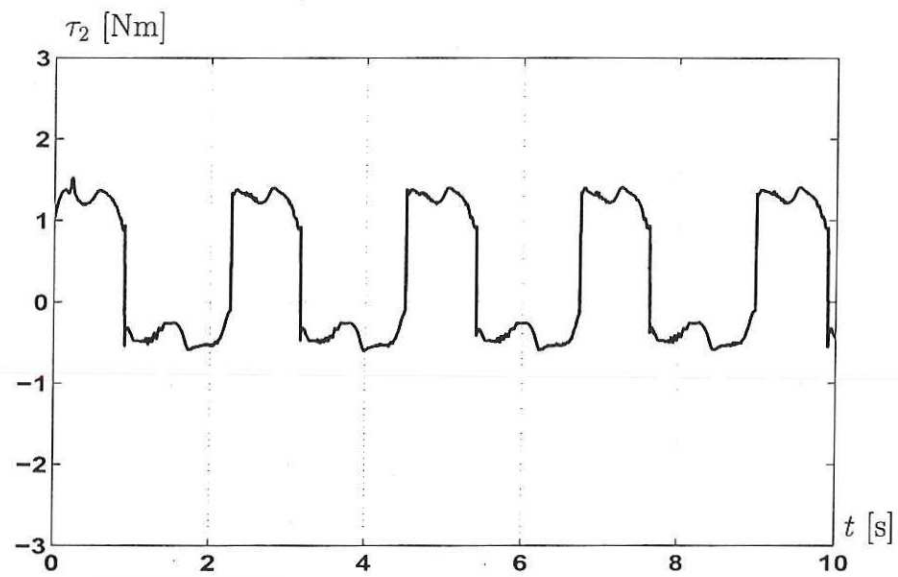


Figura 34: Par aplicado a la articulación 2 del robot. [dinámica inversa - experimento]

Capítulo IV

Control basado en estructura PD+

El siguiente controlador se inspira en el algoritmo propuesto por Koditschek (1984), definido en coordenadas articulares. El controlador PD+ original está dado por:

$$\tau = M(q)\ddot{q}_d + C(q, \dot{q})\dot{q}_d + g(q) + K_p \tilde{q} + K_v \dot{\tilde{q}} \quad (88)$$

donde $K_v, K_p \in \mathbb{R}^{n \times n}$ son matrices simétricas definidas positivas seleccionadas por el diseñador, q_d es una referencia temporal de posición y donde $\tilde{q} = q_d - q$ denota error de posición.

En este capítulo se presenta un controlador inspirado en (88), ahora para coordenadas cartesianas y mediante campo de velocidad. También se presenta un controlador en estructura PD+ para coordenadas cartesianas con compensación de fricción siguiendo el modelo de Dahl.

IV.1 Control PD+

Para resolver el problema de control mediante campo de velocidad se propone la siguiente ley de control:

$$\begin{aligned} \tau = M(q) \left[J(q)^{-1} \frac{d}{dt} [v(x)] + \frac{d}{dt} [J(q)^{-1}] v(x) \right] \\ + C(q, \dot{q}) [J(q)^{-1} v(x)] + g(q) + K [J(q)^{-1} v(x) - \dot{q}] \end{aligned} \quad (89)$$

donde K es una matriz, simétrica y definida positiva.

En la Figura 35 se muestra un diagrama de bloques del controlador PD+ mediante campo de velocidad para coordenadas cartesianas.

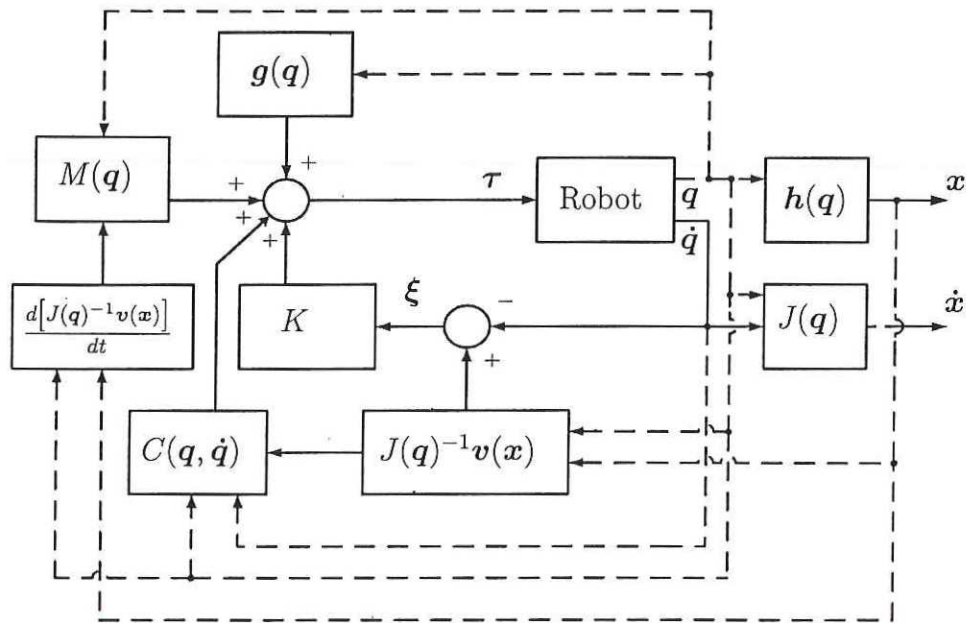


Figura 35: Diagrama de bloques del controlador PD+ para coordenadas cartesianas.

Se busca obtener la expresión en lazo cerrado igualando (52) y (89):

$$M(q) \left[J(q)^{-1} \frac{d}{dt} [v(x)] + \frac{d}{dt} [J(q)^{-1}] v(x) - \ddot{q} \right] + C(q, \dot{q}) [J(q)^{-1} v(x) - \dot{q}] + K [J(q)^{-1} v(x) - \dot{q}] = 0 \quad (90)$$

Si definimos:

$$\xi = J(q)^{-1} v(x) - \dot{q} \quad (91)$$

entonces el sistema en malla cerrada puede reescribirse como:

$$M(q)\dot{\xi} + [C(q, \dot{q}) + K]\xi = 0 \quad (92)$$

IV.1.1 Análisis de estabilidad.

Para el análisis de estabilidad del sistema descrito por (92) se propone la siguiente función candidata de Lyapunov:

$$V(\xi) = \frac{1}{2} \xi^T M(q) \xi \quad (93)$$

Calculando la derivada temporal de la función propuesta de Lyapunov se obtiene:

$$\dot{V}(\xi) = \xi^T M(q) \dot{\xi} + \frac{1}{2} \xi^T \dot{M}(q) \xi \quad (94)$$

$$= \xi^T [-K\xi - C(q, \dot{q})\xi] + \frac{1}{2} \xi^T \dot{M}(q) \xi \quad (95)$$

$$= -\xi^T K \xi \quad (96)$$

donde se ha utilizado la propiedad que establece la antisimetría de

$$\frac{1}{2} \dot{M}(q) - C(q, \dot{q}) \quad (97)$$

Por lo tanto el equilibrio $\xi = 0$ posee la propiedad de estabilidad asintótica global, y en consecuencia $J(q)\xi(t) \rightarrow 0$ cuando $t \rightarrow \infty$ ya que $J(q)$ es acotado. Esto significa que $[v(x(t)) - \dot{x}(t)] \rightarrow 0$ cuando $t \rightarrow \infty$, es decir, se satisface el objetivo de control de campo de velocidad.

IV.2 Control PD+ con compensación de fricción utilizando el modelo de Dahl

El modelo dinámico de un robot de n grados de libertad con articulaciones rotacionales en presencia de fricción puede ser descrito por:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + f_d(z, \dot{q}) = \tau \quad (98)$$

donde $f_d(z, \dot{q})$ es el modelo de fricción propuesto por [Dahl, 1976].

El modelo de Dahl está dado por:

$$\dot{z} = -\Sigma_0 \Psi(\dot{q})z + \dot{q} \quad (99)$$

$$f_d(z, \dot{q}) = \Sigma_0 z + F_v \dot{q} \quad (100)$$

donde Σ_0 es una matriz diagonal y definida positiva que contiene los coeficientes de rigidez asociados al modelo de fricción de Dahl correspondientes a las articulaciones del robot y F_v es la matriz de coeficientes de fricción viscosa. La matriz $\Psi(\dot{q})$ está definida por:

$$\Psi(\dot{q}) = \text{diag} \left\{ \frac{1}{f_{c1}} |\dot{q}_1|, \dots, \frac{1}{f_{cn}} |\dot{q}_n| \right\} \quad (101)$$

es diagonal y semidefinida positiva, donde $fc_1, \dots, fc_n$ son los coeficientes de fricción de Coulomb. El vector z definido por:

$$z = \begin{bmatrix} z_1 & z_2 & \dots & z_n \end{bmatrix}^T \quad (102)$$

contiene los estados internos del modelo de fricción de Dahl a los cuales no se puede acceder directamente.

Sustituyendo (100) en (98) se obtiene:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + \Sigma_0 z + F_v \dot{q} = \tau \quad (103)$$

Se propone la siguiente ley de control utilizando compensación de fricción mediante el modelo de Dahl:

$$\begin{aligned} \tau = M(q) & \left[J(q)^{-1} \frac{d}{dt} [v(x)] + \frac{d}{dt} [J(q)^{-1}] v(x) \right] \\ & + C(q, \dot{q}) [J(q)^{-1} v(x)] + g(q) + K [J(q)^{-1} v(x) - \dot{q}] + \Sigma_0 \hat{z} + F_v \dot{q} \end{aligned} \quad (104)$$

donde $\hat{z}$ son los estados internos observados del modelo de fricción de Dahl.

Para obtener una estimación de los estados internos del modelo de fricción se propone el siguiente observador:

$$\dot{\hat{z}} = -\Sigma_0 \Psi(\dot{q}) \hat{z} + K_0 [J(q)^{-1} v(x) - \dot{q}] + \dot{q} \quad (105)$$

Donde K_0 es una matriz de ganancias diagonal y definida positiva denominada *ganancia del observador*.

Igualando (104) con el modelo dado por (103) se obtiene:

$$M(q) \left[\frac{d}{dt} [J(q)^{-1} v(x) - \dot{q}] \right] + [C(q, \dot{q}) + K] [J(q)^{-1} v(x) - \dot{q}] + \Sigma_0 [\hat{z} - z] = 0 \quad (106)$$

Ahora definimos el error de observación de los estados internos del modelo de fricción:

$$\tilde{z} = \hat{z} - z \quad (107)$$

También conviene definir:

$$\xi = J(q)^{-1} v(x) - \dot{q} \quad (108)$$

Podemos reescribir ahora el sistema en términos de los estados $[\xi \ \tilde{z}]^T$:

$$M(q)\dot{\xi} + [C(q, \dot{q}) + K]\xi + \Sigma_0 \tilde{z} = 0 \quad (109)$$

$$\dot{\tilde{z}} = -\Sigma_0 \Psi(\dot{q})\tilde{z} + K_0 \xi \quad (110)$$

donde $\tilde{z}$ es el error de observación para el estado interno z .

El sistema en ecuaciones de estado del nuevo sistema está dado por:

$$\frac{d}{dt} \begin{bmatrix} \xi \\ \tilde{z} \end{bmatrix} = \begin{bmatrix} M(q)^{-1} [-C(q, \dot{q})\xi - K\xi - \Sigma_0 \tilde{z}] \\ -\Sigma_0 \Psi(\dot{q})\tilde{z} + K_0 \xi \end{bmatrix} \quad (111)$$

IV.2.1 Análisis de estabilidad

Para llevar a cabo el análisis de estabilidad del nuevo sistema se propone la siguiente función candidata de Lyapunov:

$$V(\xi, \tilde{z}) = \frac{1}{2} \xi^T M(q) \xi + \frac{1}{2} \tilde{z}^T \Sigma_0 K_0^{-1} \tilde{z} \quad (112)$$

Calculando la derivada temporal de la función candidata de Lyapunov se obtiene:

$$\dot{V}(\xi, \tilde{z}) = \xi^T M(q) \dot{\xi} + \frac{1}{2} \xi^T \dot{M}(q) \xi + \tilde{z}^T \Sigma_0 K_0^{-1} \dot{\tilde{z}} \quad (113)$$

Sustituyendo las derivadas temporales de los estados descritas en (111) se llega a:

$$\dot{V}(\xi, \tilde{z}) = -\xi^T K \xi - \tilde{z}^T \Sigma_0^2 K_0^{-1} \Psi(\dot{q}) \tilde{z} \quad (114)$$

Ya que $\Sigma_0^2 K_0^{-1} \Psi(\dot{q})$ es semidefinida positiva, se concluye que:

$$\dot{V}(\xi, \tilde{z}) \leq 0 \quad \forall \begin{bmatrix} \xi \\ \tilde{z} \end{bmatrix} \in \mathbb{R}^{2n} \quad (115)$$

es decir, la derivada temporal de la función candidata de Lyapunov es semidefinida negativa. Por lo tanto, de acuerdo con el Teorema 2 del Apéndice A, el origen del espacio de estado es estable y sus soluciones acotadas, i.e. $\xi, \tilde{z} \in L_\infty^n$ (Definición 2, Apéndice A). A partir de la ecuación (116):

$$\dot{\xi} = M(q)^{-1} [-C(q, \dot{q})\xi - K\xi - \Sigma_0 \tilde{z}] \quad (116)$$

y de la ecuación (108) de donde se desprende:

$$\dot{q} = J(q)^{-1}v(x) - \xi \quad (117)$$

se puede concluir que $\dot{q} \in L_{\infty}^n$ y por lo tanto:

$$\dot{\xi} \in L_{\infty}^n \quad (118)$$

A partir de la derivada temporal de la función candidata de Lyapunov se concluyó:

$$\dot{V}(\xi, \tilde{z}) = -\xi^T K \xi - \tilde{z}^T \Sigma_0^2 K_0^{-1} \Psi(\dot{q}) \tilde{z} \quad (119)$$

Igualmente podemos escribir:

$$\dot{V}(\xi, \tilde{z}) \leq -\xi^T K \xi \quad (120)$$

Utilizando el teorema de Raleygh-Ritz¹:

$$\dot{V}(\xi, \tilde{z}) \leq -\xi^T K \xi \leq -\lambda_{\min} \{K\} \|\xi\|^2 \quad (122)$$

Integrando ambos lados de (122) se obtiene:

$$\int_0^t \dot{V}(\xi, \tilde{z}) d\sigma \leq \int_0^t -\lambda_{\min} \{K\} \|\xi\|^2 d\sigma \quad (123)$$

$$V(\xi(t), \tilde{z}(t)) - V(\xi(0), \tilde{z}(0)) \leq -\int_0^t \lambda_{\min} \{K\} \|\xi\|^2 d\sigma \quad (124)$$

$$-V(\xi(t), \tilde{z}(t)) + V(\xi(0), \tilde{z}(0)) \geq \int_0^t \lambda_{\min} \{K\} \|\xi\|^2 d\sigma \quad (125)$$

$$\int_0^t \lambda_{\min} \{K\} \|\xi\|^2 d\sigma \leq V(\xi(0), \tilde{z}(0)) \quad (126)$$

$$\int_0^t \|\xi\|^2 d\sigma \leq \frac{V(\xi(0), \tilde{z}(0))}{\lambda_{\min} \{K\}} \quad (127)$$

¹El teorema de Raleygh-Ritz establece que:

$$\lambda_{Max} \{A\} \|x\|^2 \geq x^T A x \geq \lambda_{min} \{A\} \|x\|^2 \quad (121)$$

donde $\lambda_{Max} \{A\}$ y $\lambda_{min} \{A\}$ son los valores propios máximo y mínimo de A respectivamente.

Por lo tanto se concluye que $\xi \in L_2^n$ y de acuerdo el Lema 1 del Apéndice A se puede concluir entonces que:

$$\lim_{t \rightarrow \infty} \xi(t) = 0 \quad (128)$$

Es decir, se concluye convergencia al origen del error de campo de velocidad:

$$\lim_{t \rightarrow \infty} e(x(t), \dot{x}(t)) \Rightarrow \lim_{t \rightarrow \infty} [v(x(t)) - \dot{x}(t)] \Rightarrow \lim_{t \rightarrow \infty} J(q(t))\xi(t) = 0 \quad (129)$$

IV.3 Simulaciones

Se generaron simulaciones utilizando Simnon v.2.0 y el algoritmo de integración de Runge-Kutta-Felhberg 4/5, considerando el sistema dinámico como continuo en su totalidad.

Se presentan simulaciones del controlador mediante campo de velocidad en estructura PD+ propuesto en la sección IV.1.

El controlador mediante campo de velocidad en estructura PD+ está definido por:

$$\tau = M(q) \left[J(q)^{-1} \frac{d}{dt} [v(x)] + \frac{d}{dt} [J(q)^{-1}] v(x) \right] \quad (130)$$

$$+ C(q, \dot{q}) [J(q)^{-1} v(x)] + g(q) + K [J(q)^{-1} v(x) - \dot{q}]. \quad (131)$$

Se generaron simulaciones con las condiciones iniciales en coordenadas cartesianas siguientes:

$$x_1(0) = 0.3677 [m] \quad (132)$$

$$x_2(0) = 0.0 [m] \quad (133)$$

o en coordenadas articulares:

$$q_1(0) = \pi/4 [rad] \quad (134)$$

$$q_2(0) = \pi/2 [rad] \quad (135)$$

En la Figura 36 se muestra la trayectoria del extremo del robot referente al eje cartesiano x_1 . En la Figura 37 se muestra la trayectoria del extremo del robot referente al eje cartesiano x_2 .

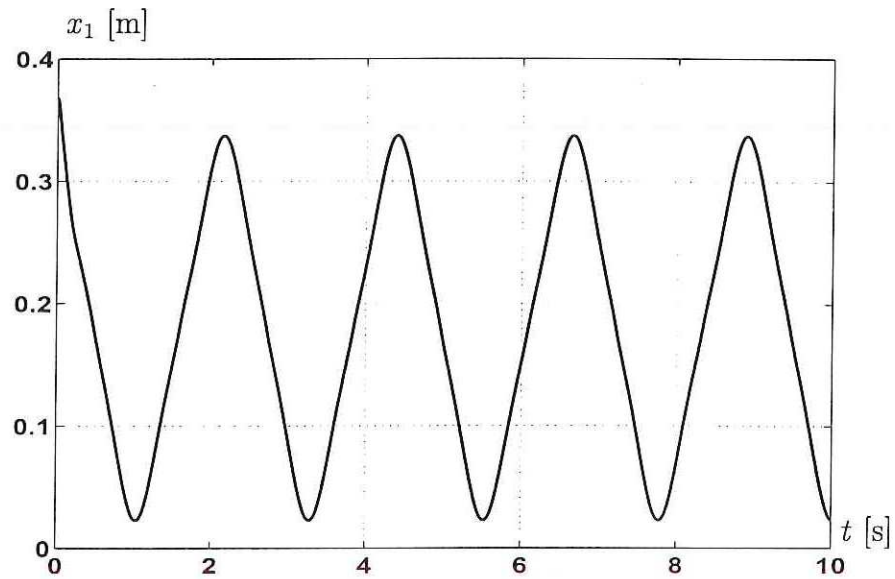


Figura 36: Trayectoria del extremo del robot correspondiente al eje cartesiano x_1 . [PD+ - simulación]

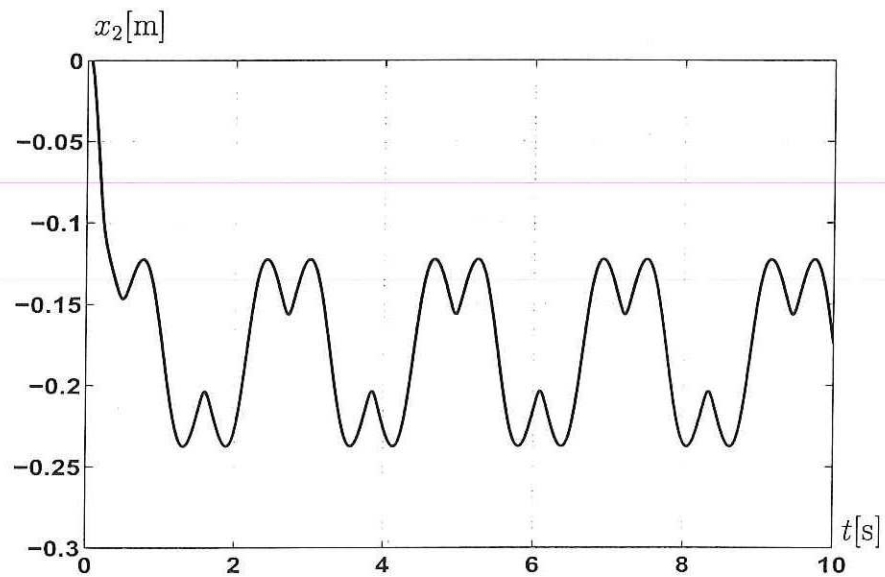


Figura 37: Trayectoria del extremo del robot correspondiente al eje cartesiano x_2 . [PD+ - simulación]

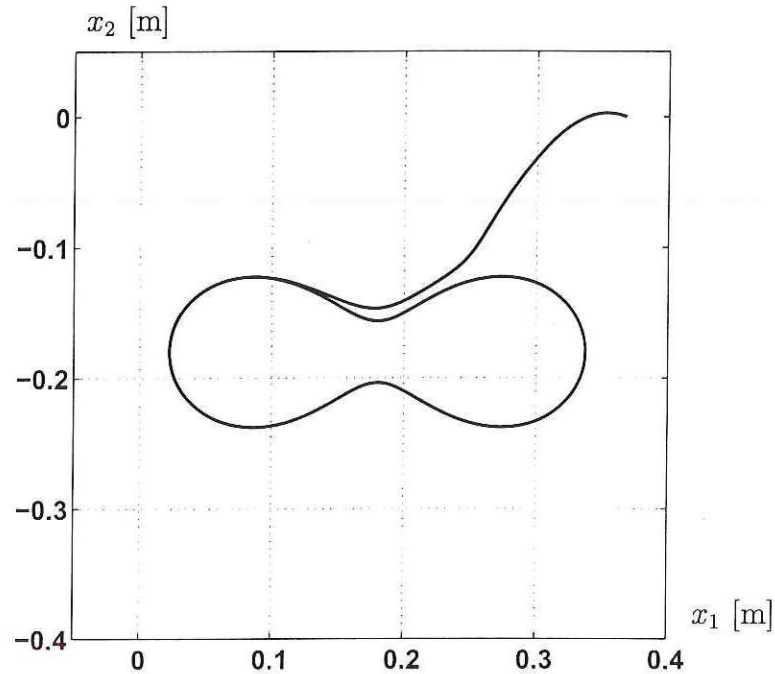


Figura 38: Trayectoria del extremo del robot en el espacio cartesiano. [PD+ - simulación]

El campo de velocidad de referencia fué diseñado para converger a una curva cerrada definida por Γ_x . En la Figura 38 se muestra la trayectoria del extremo del robot en el espacio cartesiano x_1 - x_2 .

El campo de velocidad se diseño para converger a Γ_x y para seguir esta curva a una velocidad constante de $c = 0.35$ [m/s].

En la Figura 39 se grafica la velocidad del extremo del robot que corresponde al eje cartesiano x_1 así como en la Figura 40 se grafica la velocidad del extremo del robot que corresponde al eje cartesiano x_2 .

La velocidad resultante del extremo del robot se calcula a través de la norma $\|\dot{\mathbf{x}}\|$:

$$\|\dot{\mathbf{x}}\| = \sqrt{\dot{x}_1^2 + \dot{x}_2^2} \quad (136)$$

En la Figura 41 se muestra la norma de la velocidad del extremo del robot en el espacio cartesiano x_1 - x_2 , se observa como esta converge a 0.35 [m/s].

En la Figura 42 se muestra el error de campo de velocidad correspondiente al eje cartesiano x_1 :

$$e_1(\mathbf{x}, \dot{\mathbf{x}}) = v_1(\mathbf{x}) - \dot{x}_1 \quad (137)$$

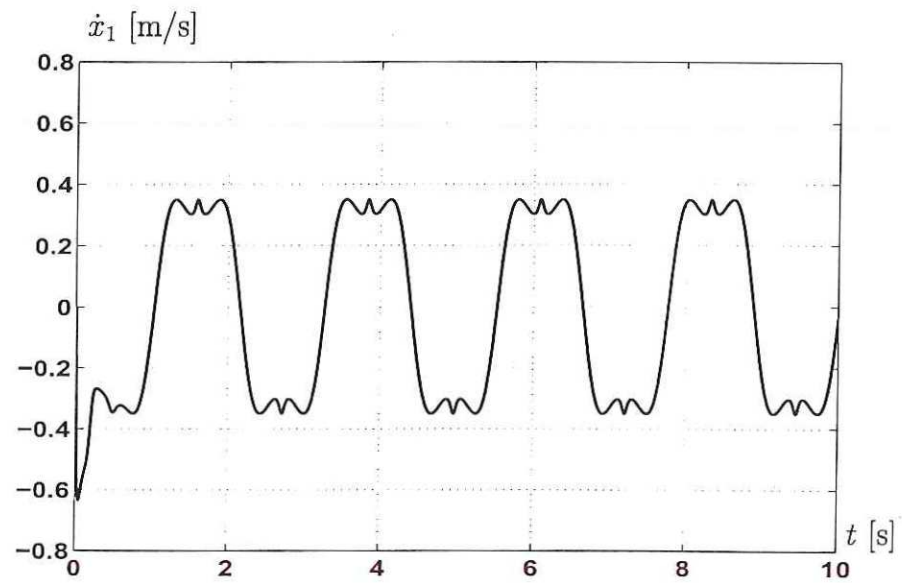


Figura 39: Velocidad del extremo del robot referente al eje cartesiano x_1 . [PD+ - simulación]

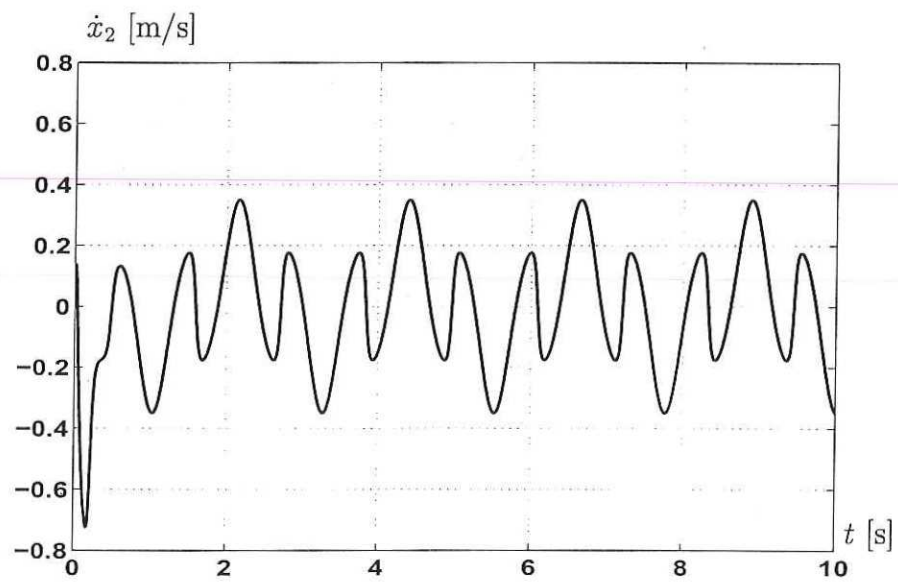


Figura 40: Velocidad del extremo del robot referente al eje cartesiano x_2 . [PD+ - simulación]

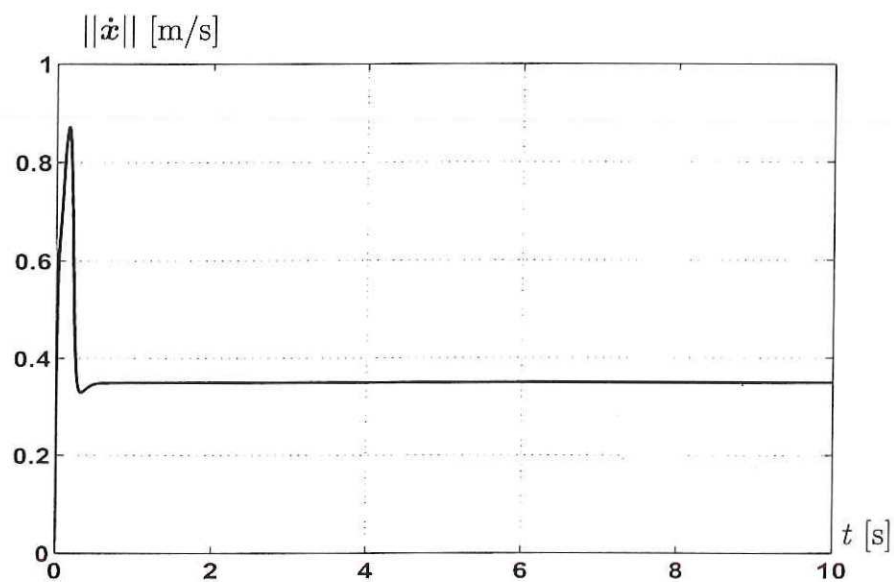


Figura 41: Norma de la velocidad del extremo del robot. [PD+ - simulación]

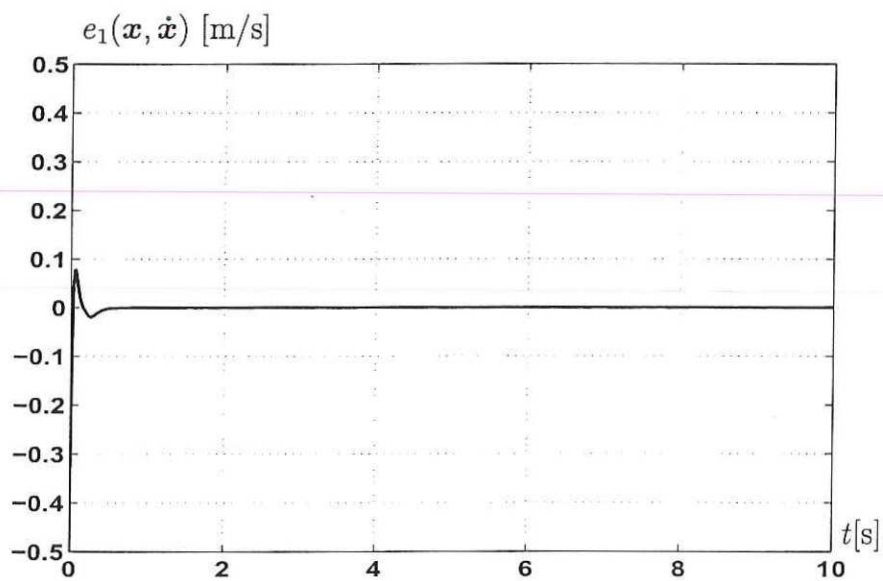


Figura 42: Error de campo de velocidad que corresponde al eje cartesiano x_1 . [PD+ - simulación]

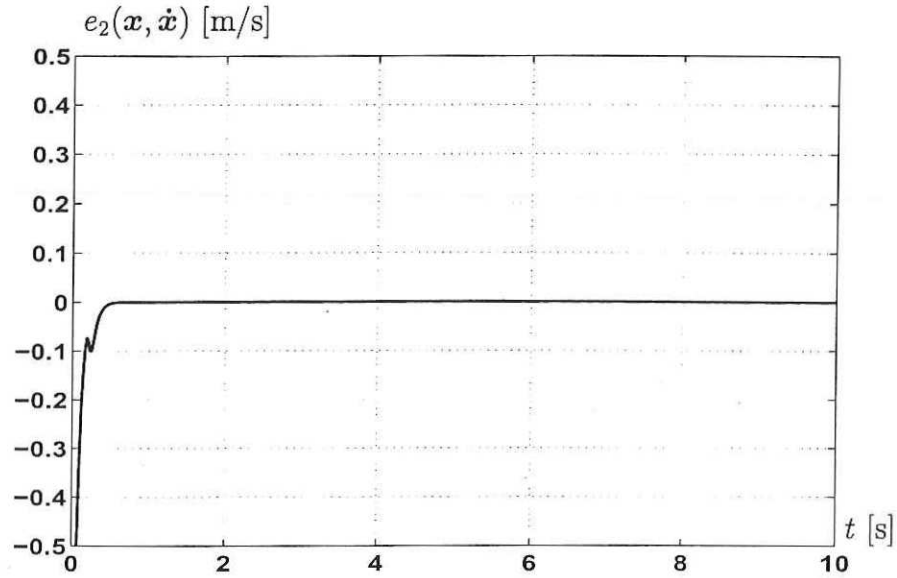


Figura 43: Error de campo de velocidad que corresponde al eje cartesiano x_2 . [PD+ - simulación]

así como en la Figura 43 se muestra el error de campo de velocidad correspondiente al eje cartesiano x_2 :

$$e_2(\mathbf{x}, \dot{\mathbf{x}}) = v_2(\mathbf{x}) - \dot{x}_2 \quad (138)$$

Se calcula la norma $\|e(\mathbf{x}, \dot{\mathbf{x}})\|$ a través de:

$$\|e(\mathbf{x}, \dot{\mathbf{x}})\| = \sqrt{[e_1(\mathbf{x}, \dot{\mathbf{x}})]^2 + [e_2(\mathbf{x}, \dot{\mathbf{x}})]^2} \quad (139)$$

En la Figura 44 se muestra la norma del error de campo de velocidad.

Finalmente en las Figuras 45 y 46 se muestran los pares de control τ_1 y τ_2 correspondientes a las articulaciones q_1 y q_2 .

IV.4 Experimentos

Se implementó el controlador en estructura PD+ mediante campo de velocidad para coordenadas cartesianas propuesto en la Sección IV.2 en el robot Bugarín.

La ley de control PD+ mediante campo de velocidad para coordenadas cartesianas y utilizando compensación dinámica de fricción siguiendo el modelo de Dahl viene dada

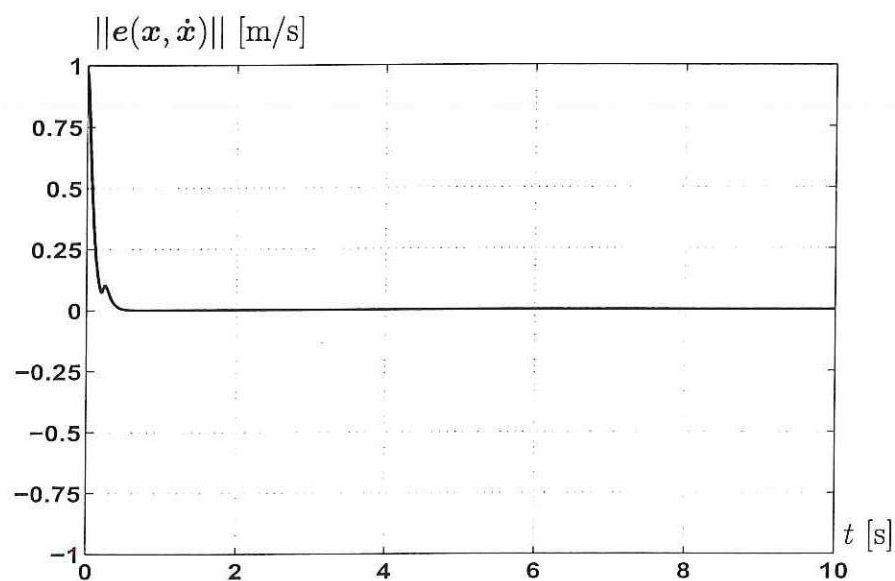


Figura 44: Norma del error de campo de velocidad. [PD+ - simulación]

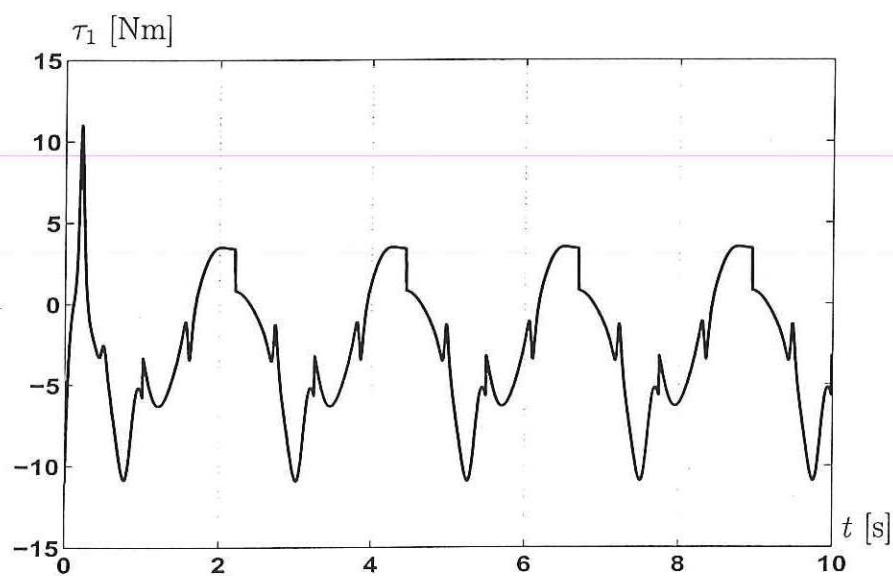


Figura 45: Par aplicado a la articulación 1 del robot. [PD+ - simulación]

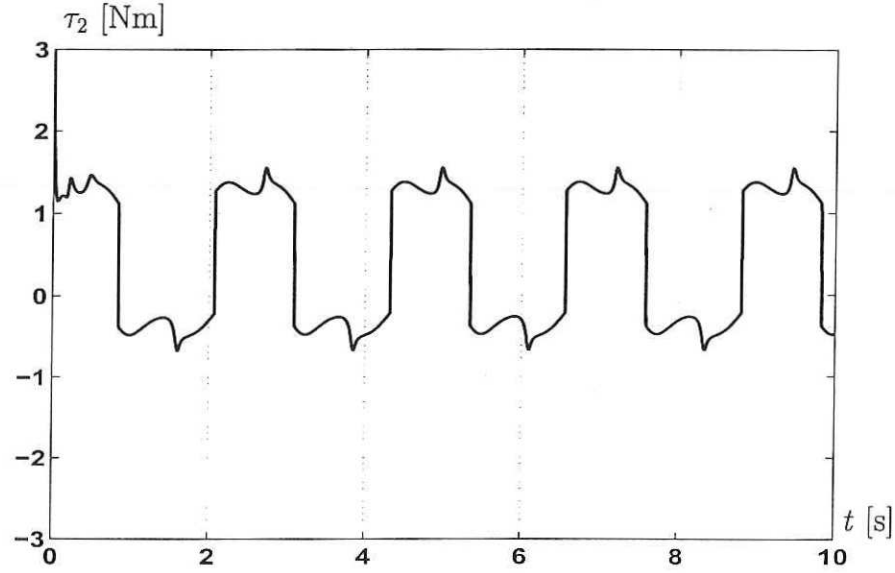


Figura 46: Par aplicado a la articulación 2 del robot. [PD+ - simulación]

por:

$$\begin{aligned} \tau = M(q) & \left[J(q)^{-1} \frac{d}{dt} [v(x)] + \frac{d}{dt} [J(q)^{-1}] v(x) \right] \\ & + C(q, \dot{q}) [J(q)^{-1} v(x)] + g(q) + K [J(q)^{-1} v(x) - \dot{q}] + \Sigma_0 \hat{z} + F_v \dot{q} \end{aligned} \quad (140)$$

Para la estimación de los estados internos del modelo de fricción (99)-(100) se utiliza un observador dado por:

$$\dot{\hat{z}} = -\Sigma_0 \Psi(\hat{q}) \hat{z} + K_0 [J(q)^{-1} v(x) - \dot{q}] + \dot{q} \quad (141)$$

La velocidades articulares son estimadas utilizando el algoritmo de Euler:

$$\dot{q}_k = \frac{q[k+1] - q[k]}{h} \quad (142)$$

donde h es el periodo de muestreo ($h = 0.001$ [s]).

La matriz de ganancias del observador implementada fué:

$$K_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (143)$$

y la matriz de ganancias K implementada fué:

$$K = \begin{bmatrix} 5.0 & 0 \\ 0 & 0.75 \end{bmatrix} \quad (144)$$

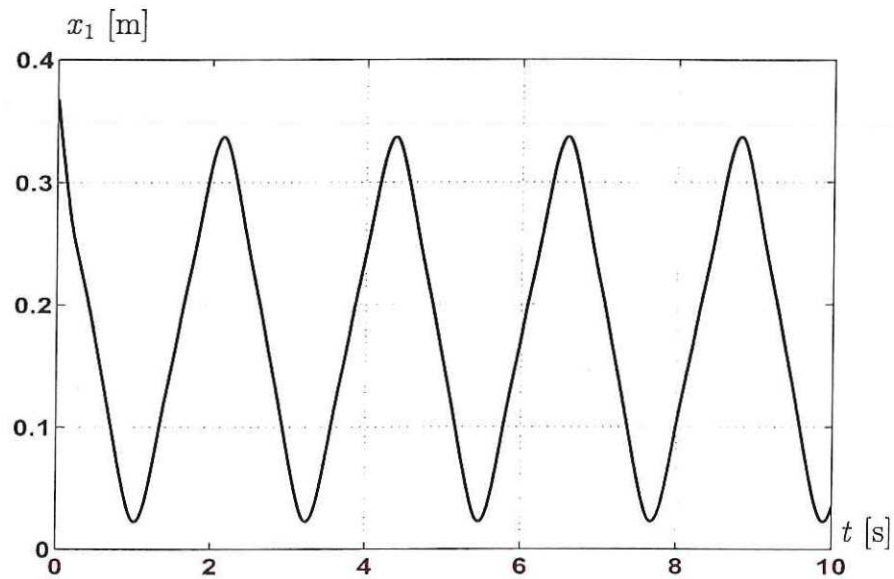


Figura 47: Trayectoria del extremo del robot correspondiente al eje cartesiano x_1 . [PD+ - experimento]

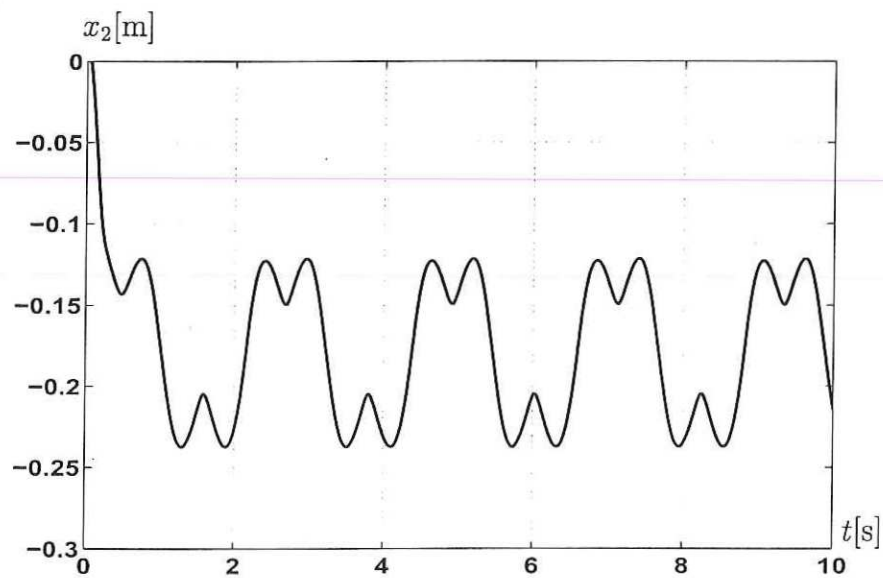


Figura 48: Trayectoria del extremo del robot correspondiente al eje cartesiano x_2 . [PD+ - experimento]

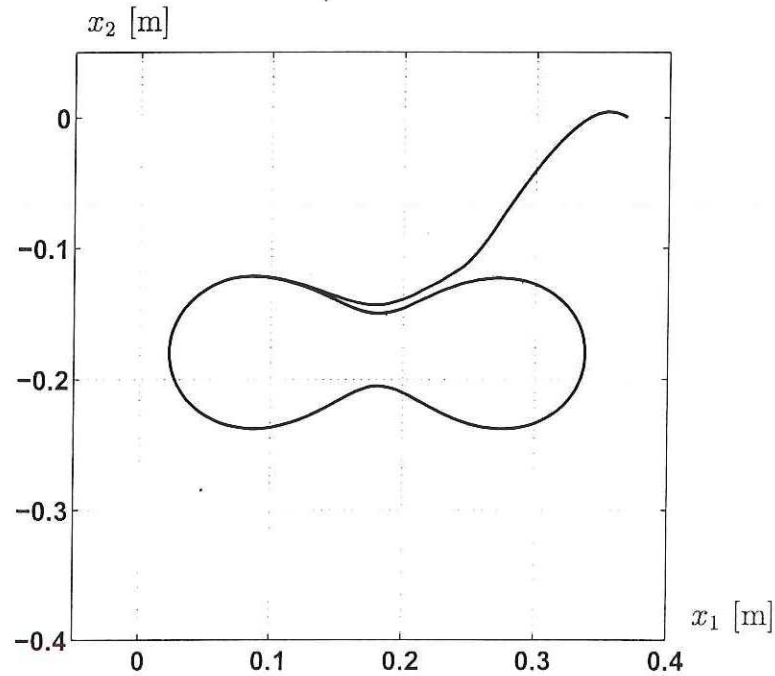


Figura 49: Trayectoria del extremo del robot en el espacio cartesiano. [PD+ - experimento]

La matriz de coeficientes de fricción de Dahl implementada fué:

$$\Sigma_0 = \begin{bmatrix} 1757.0 & 0 \\ 0 & 1450.0 \end{bmatrix} \quad (145)$$

la matriz de coeficientes de fricción de Coulomb F_c implementada fué:

$$F_c = \begin{bmatrix} 1.290 & 0 \\ 0 & 0.745 \end{bmatrix} \quad (146)$$

y finalmente la matriz de coeficientes de fricción viscosa implementada fué:

$$F_v = \begin{bmatrix} 0.274 & 0 \\ 0 & 0.144 \end{bmatrix} \quad (147)$$

En la Figura 47 se grafica la trayectoria realizada por el extremo del robot correspondiente a la coordenada cartesiana x_1 , así como en la Figura 48 se muestra la trayectoria realizada por el extremo del robot correspondiente a la coordenada cartesiana x_2 . En la Figura 49 se presenta la trayectoria realizada por el extremo del robot correspondiente en el espacio cartesiano x_1-x_2 , bajo control PD+ mediante campo de velocidad con compensación de fricción de Dahl.

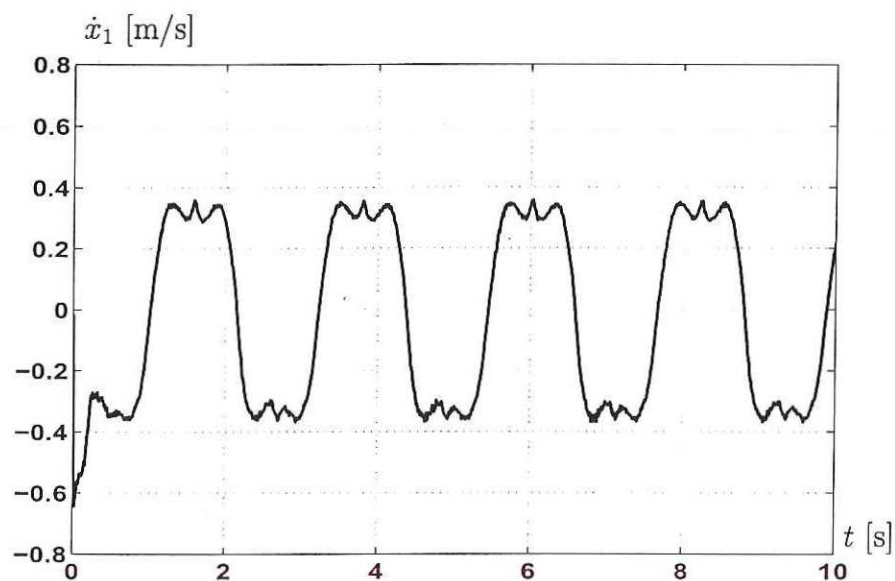


Figura 50: Velocidad del extremo del robot referente al eje cartesiano x_1 . [PD+ - experimento]

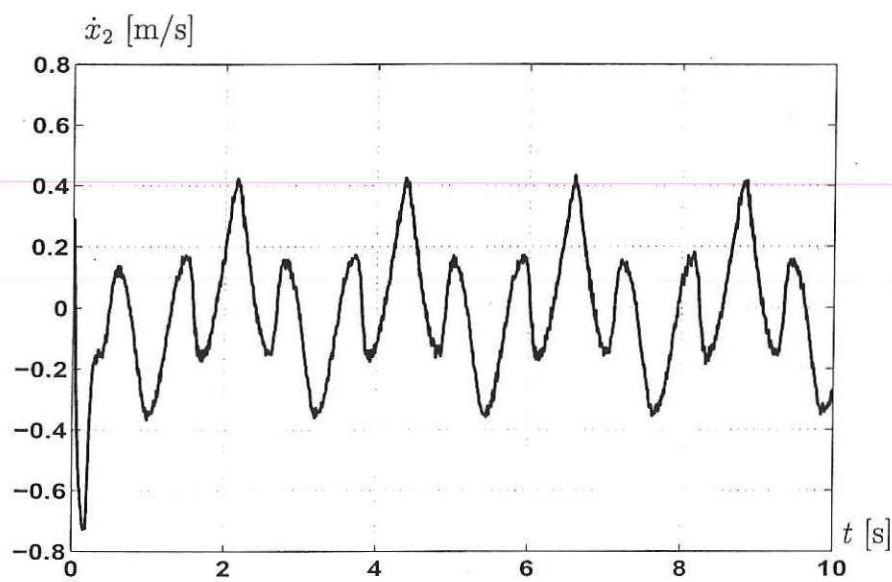


Figura 51: Velocidad del extremo del robot referente al eje cartesiano x_2 . [PD+ - experimento]

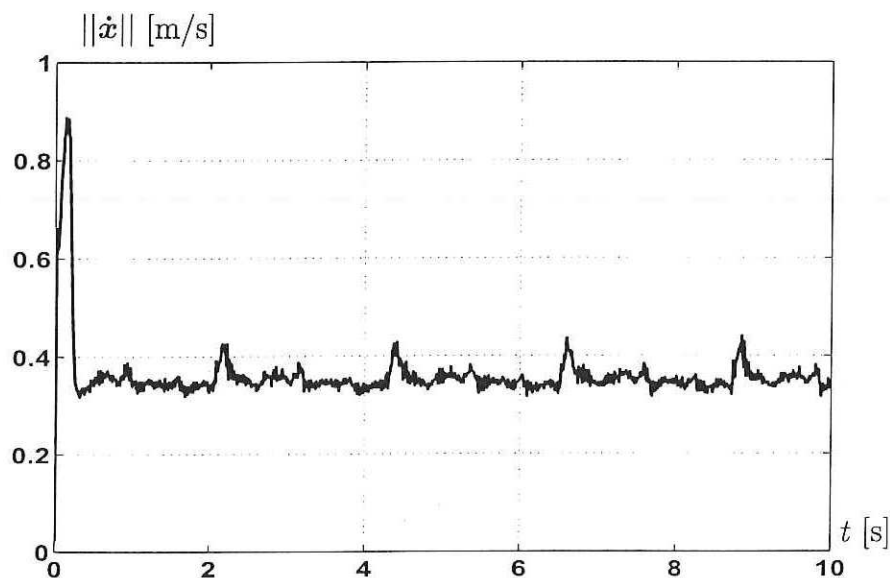


Figura 52: Norma de la velocidad del extremo del robot. [PD+ - experimento]

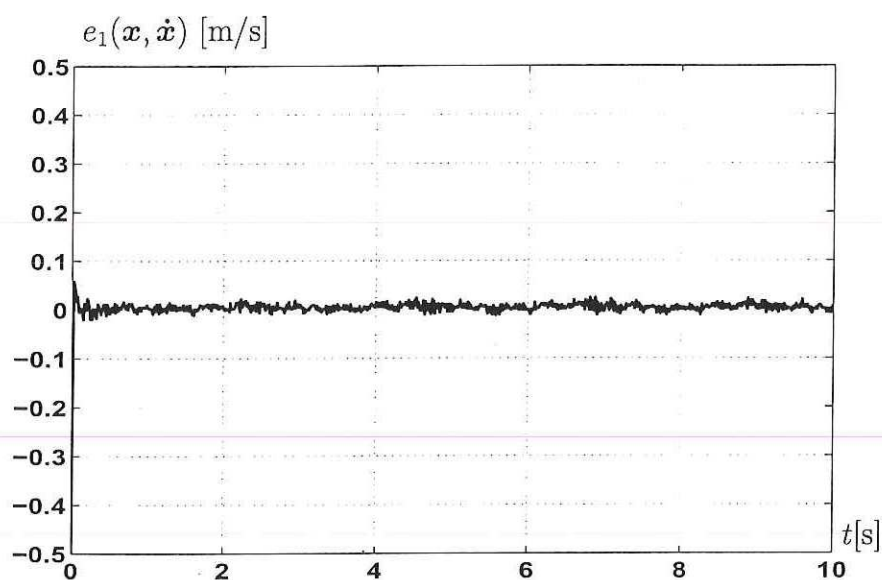


Figura 53: Error de campo de velocidad que corresponde al eje cartesiano x_1 . [PD+ - experimento]

En la Figura 50 se presenta la velocidad del extremo del robot correspondiente a la coordenada cartesiana x_1 , así como en la Figura 51 se muestra la velocidad del extremo del robot correspondiente a la coordenada cartesiana x_2 . En la Figura 52 se presenta la norma de la velocidad $||\dot{x}||$. Se observa como ésta oscila entre 0.3 y 0.45 [m/s].

En la Figura 53 se grafica el error de campo de velocidad correspondiente a la

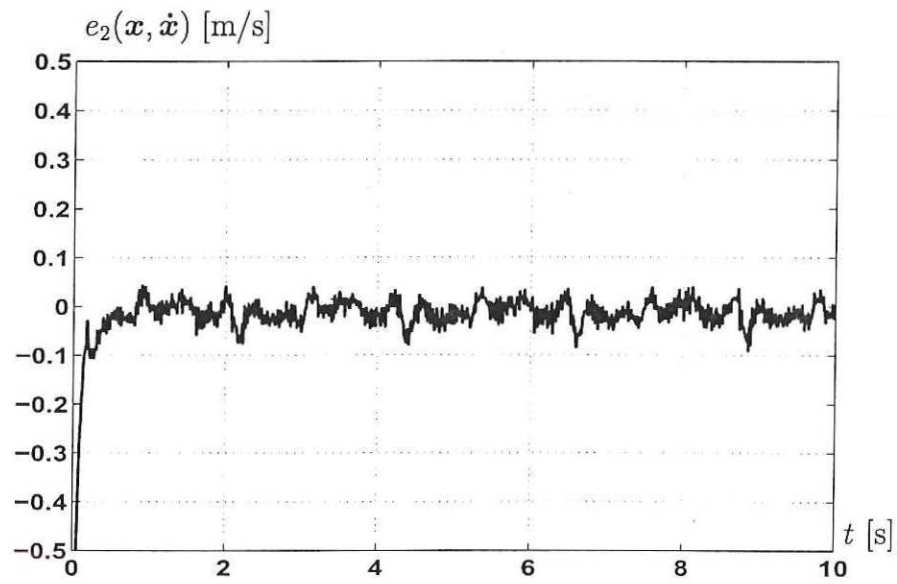


Figura 54: Error de campo de velocidad que corresponde al eje cartesiano x_2 . [PD+ - experimento]

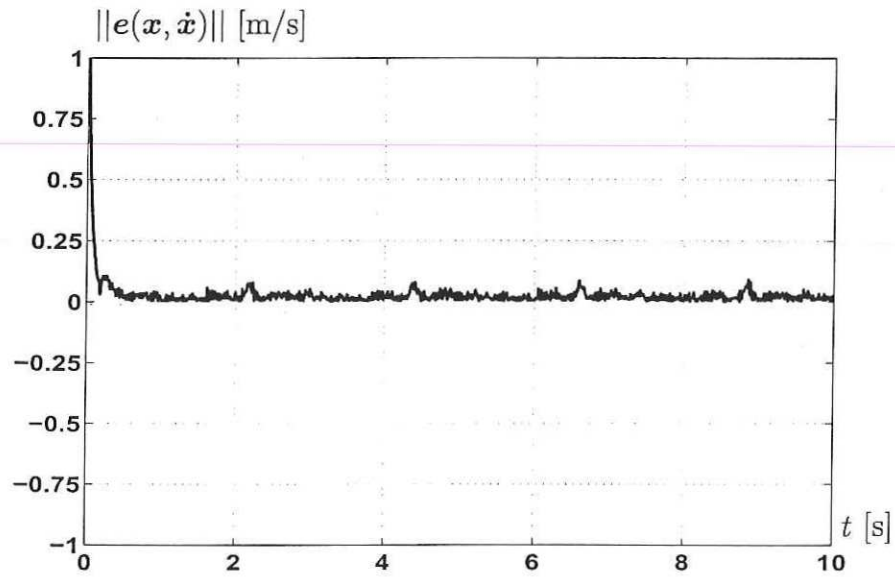


Figura 55: Norma del error de campo de velocidad. [PD+ - experimento]

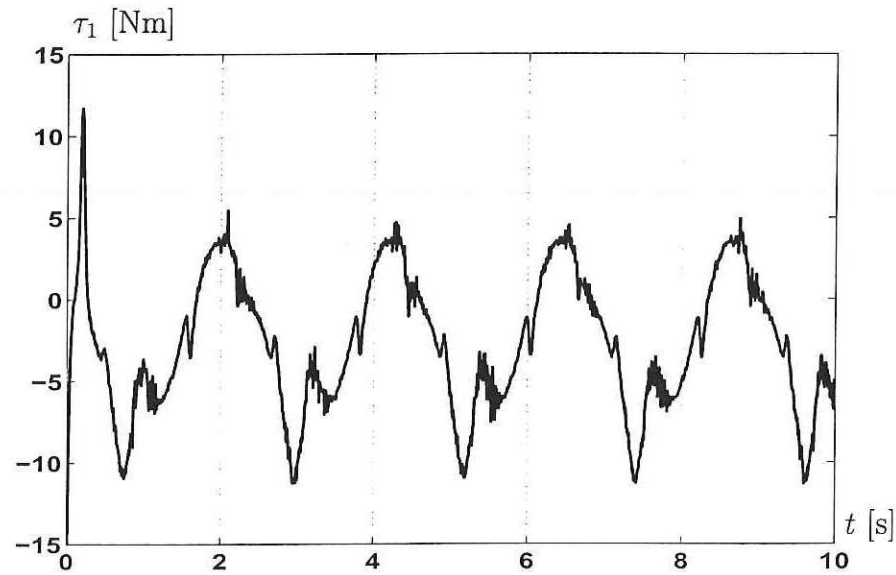


Figura 56: Par aplicado a la articulación 1 del robot. [PD+ - experimento]

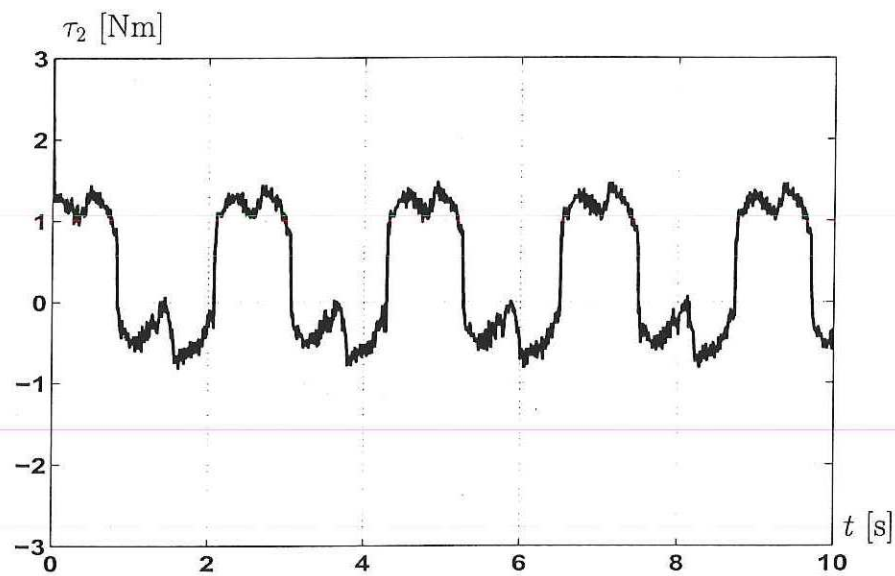


Figura 57: Par aplicado a la articulación 2 del robot. [PD+ - experimento]

coordenada cartesiana x_1 , así como en la Figura 51 se grafica el error de campo de velocidad correspondiente a la coordenada cartesiana x_2 . En la Figura 55 se muestra la norma del error $\|e(\mathbf{x}, \dot{\mathbf{x}})\|$.

Finalmente en la Figura 56 se grafica el par τ_1 aplicado a la articulación q_1 , así como en la Figura 57 se grafica el par τ_2 aplicado a la articulación q_2 .

Capítulo V

Control de Li y Horowitz

El algoritmo de control de Li y Horowitz es el antecedente de los algoritmos estudiados en el presente trabajo de tesis, en cuanto al uso de campos vectoriales para especificación de movimiento. Tal controlador cumple con el siguiente objetivo de control:

$$\lim_{t \rightarrow \infty} e^*(q(t), \dot{q}(t)) = \lim_{t \rightarrow \infty} [\dot{q}(t) - \omega_d v^*(q(t))] = 0 \quad (148)$$

donde $e^*(q, \dot{q})$ es el error de campo de velocidad definido en coordenadas articulares, $v^*(q)$ es la especificación de campo de velocidad en coordenadas articulares y ω_d es un número real positivo. Es decir, el vector velocidad converge a un múltiplo del campo de velocidad de referencia.

V.1 Terminología

El algoritmo de Li y Horowitz incluye dentro de su definición una articulación virtual, con dinámica definida por:

$$\tau_{n+1} = M_F \ddot{q}_{n+1} \quad (149)$$

donde M_F es una constante que define el momento de inercia del elemento virtual, n es el número de grados de libertad del robot, y $n + 1$ es simplemente un subíndice que situa en un orden a los elementos de ésta nueva dinámica.

El sistema dinámico aumentado en ausencia de fricción es definido por:

$$\bar{M}(q) \ddot{\bar{q}} + \bar{C}(q, \dot{q}) \dot{\bar{q}} + \bar{g}(q) = \bar{\tau} \quad (150)$$

donde:

$$\bar{q} = \begin{bmatrix} q \\ q_{n+1} \end{bmatrix}, \quad \dot{\bar{q}} = \begin{bmatrix} \dot{q} \\ \dot{q}_{n+1} \end{bmatrix}, \quad \ddot{\bar{q}} = \begin{bmatrix} \ddot{q} \\ \ddot{q}_{n+1} \end{bmatrix} \quad (151)$$

son los vectores columna de posiciones articulares aumentado, de velocidades articulares aumentado y de aceleraciones articulares aumentado respectivamente. Además, se definen:

$$\bar{M}(\mathbf{q}) = \begin{bmatrix} M(\mathbf{q}) & 0 \\ 0 & M_F \end{bmatrix}, \quad \bar{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} C(\mathbf{q}, \dot{\mathbf{q}}) & 0 \\ 0 & 0 \end{bmatrix}, \quad \bar{\mathbf{g}}(\mathbf{q}) = \begin{bmatrix} \mathbf{g}(\mathbf{q}) \\ 0 \end{bmatrix} \quad (152)$$

donde $\bar{M}(\mathbf{q})$ es la matriz de inercia aumentada, $\bar{C}(\mathbf{q}, \dot{\mathbf{q}})$ es la matriz de fuerzas centrífugas y de Coriolis del sistema aumentado, $\bar{\mathbf{g}}(\mathbf{q})$ es el vector de pares gravitacionales aumentado y:

$$\bar{\boldsymbol{\tau}} = \begin{bmatrix} \boldsymbol{\tau} \\ \tau_{n+1} \end{bmatrix} \quad (153)$$

donde $\bar{\boldsymbol{\tau}}$ es el vector de pares de control aumentado.

V.2 Ley de control

Para llevar a cabo el control mediante campo de velocidad, contando con el diseño de un campo vectorial $\mathbf{v}^*(\mathbf{q})$, es necesario calcular un elemento extra para definir un campo de velocidad aumentado que corresponda con el nuevo sistema.

Tal campo de velocidad aumentado es diseñado de modo que cumpla con las siguiente condición referente a la energía cinética del sistema:

Condición 1. La energía cinética total del sistema aumentado evaluada con el campo de velocidad aumentado deseado $\bar{\mathbf{v}}(\mathbf{q})$ deberá de ser constante, i.e.:

$$\bar{k}(\mathbf{q}, \bar{\mathbf{v}}(\mathbf{q})) = \frac{1}{2} \bar{\mathbf{v}}(\mathbf{q})^T \bar{M}(\mathbf{q}) \bar{\mathbf{v}}(\mathbf{q}) = e > 0 \quad (154)$$

donde $\bar{k}(\mathbf{q}, \bar{\mathbf{v}}(\mathbf{q}))$ es la energía cinética deseada del sistema aumentado, e es una constante mayor que cero y:

$$\bar{\mathbf{v}}(\mathbf{q}) = \begin{bmatrix} \mathbf{v}^*(\mathbf{q}) \\ v_{n+1}(\mathbf{q}) \end{bmatrix} \quad (155)$$

es el campo de velocidad aumentado.

Despejando de (154) se puede obtener la siguiente expresión para calcular v_{n+1} :

$$v_{n+1} = \sqrt{\frac{2}{M_F} \left(e - \frac{1}{2} \mathbf{v}^*(\mathbf{q})^T M(\mathbf{q}) \mathbf{v}^*(\mathbf{q}) \right)} \quad (156)$$

Se debe seleccionar e lo suficientemente grande de modo que (156) tenga una solución real.

Para facilitar la presentación de la ley de control, se definen los vectores:

$$\bar{\mathbf{p}} = \bar{M}(\mathbf{q}) \dot{\bar{\mathbf{q}}} \quad (157)$$

$$\bar{\mathbf{P}} = \bar{M}(\mathbf{q}) \bar{\mathbf{v}}(\mathbf{q}) \quad (158)$$

$$\bar{\mathbf{w}} = \bar{M}(\mathbf{q}) \dot{\bar{\mathbf{v}}} + \bar{C}(\mathbf{q}, \dot{\bar{\mathbf{q}}}) \bar{\mathbf{v}}(\bar{\mathbf{q}}) \quad (159)$$

donde $\bar{\mathbf{p}}$ es el momento de inercia del sistema dinámico aumentado, $\bar{\mathbf{P}}$ es el momento de inercia deseado del sistema dinámico aumentado y $\bar{\mathbf{w}}$ es la dinámica inversa necesaria para que el robot siga al campo de velocidad aumentado (y donde $\dot{\bar{\mathbf{v}}} = \frac{d}{dt} \bar{\mathbf{v}}$).

La ley de control viene dada por:

$$\boldsymbol{\tau} = \begin{bmatrix} I_{n \times n} & 0_{n \times 1} \end{bmatrix} \bar{\boldsymbol{\tau}} \quad (160)$$

$$\frac{d}{dt} \begin{bmatrix} q_{n+1} \\ \dot{q}_{n+1} \end{bmatrix} = \begin{bmatrix} \dot{q}_{n+1} \\ \frac{1}{M_F} \begin{bmatrix} 0_{1 \times n} & 1 \end{bmatrix} \bar{\boldsymbol{\tau}} \end{bmatrix} \quad (161)$$

donde:

$$\bar{\boldsymbol{\tau}} = \bar{\boldsymbol{\tau}}_c + \bar{\boldsymbol{\tau}}_{f_1} + \bar{\boldsymbol{\tau}}_{f_2} \quad (162)$$

$$\bar{\boldsymbol{\tau}}_c = \frac{1}{2e} \left(\bar{\mathbf{w}} \bar{\mathbf{P}}^T - \bar{\mathbf{P}} \bar{\mathbf{w}}^T \right) \dot{\bar{\mathbf{q}}} = \frac{1}{2e} G(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) \dot{\bar{\mathbf{q}}} \quad (163)$$

$$\bar{\boldsymbol{\tau}}_{f_1} = \gamma (\bar{\mathbf{P}} \bar{\mathbf{p}}^T - \bar{\mathbf{p}} \bar{\mathbf{P}}^T) \dot{\bar{\mathbf{q}}} = \gamma R(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) \dot{\bar{\mathbf{q}}} \quad (164)$$

$$\bar{\boldsymbol{\tau}}_{f_2} = \frac{\delta}{2e} \bar{\mathbf{P}} \bar{\mathbf{P}}^T \bar{\mathbf{e}}(\mathbf{q}, \bar{\mathbf{q}}) \quad (165)$$

que para los anteriores dos controladores:

$$x_1(0) = 0.367 [m] \quad (168)$$

$$x_2(0) = 0.0 [m] \quad (169)$$

o en coordenadas articulares:

$$q_1(0) = \pi/4 [rad] \quad (170)$$

$$q_2(0) = \pi/2 [rad] \quad (171)$$

La ley de control descrita por las ecuaciones (160)–(165) cumple con el siguiente objetivo de control:

$$\lim_{t \rightarrow \infty} e^*(q, \dot{q}) = \lim_{t \rightarrow \infty} [\dot{q}(t) - \omega_d v^*(q(t))] = 0 \quad (172)$$

Con la finalidad de poder comparar el desempeño del algoritmo de Li y Horowitz, con los algoritmos por dinámica inversa y en estructura PD+, elegimos:

$$\omega_d = 1 \quad (173)$$

De manera que el objetivo de control en coordenadas articulares viene dado por:

$$\lim_{t \rightarrow \infty} e^*(q(t), \dot{q}(t)) = \lim_{t \rightarrow \infty} [\dot{q}(t) - v^*(q(t))] = 0 \quad (174)$$

el cual es el que comúnmente se define para campo de velocidad.

En analogía con (16) que mapea velocidades en el espacio articular al espacio cartesiano, se puede escribir:

$$v(x) = J(q)v^*(q) \quad (175)$$

o en sentido inverso:

$$v^*(q) = J(q)^{-1}v(x) \quad (176)$$

La tarea encomendada al robot es definida en coordenadas cartesianas, así que se utiliza la ecuación (176) para mapear las velocidades del espacio cartesiano al espacio articular.

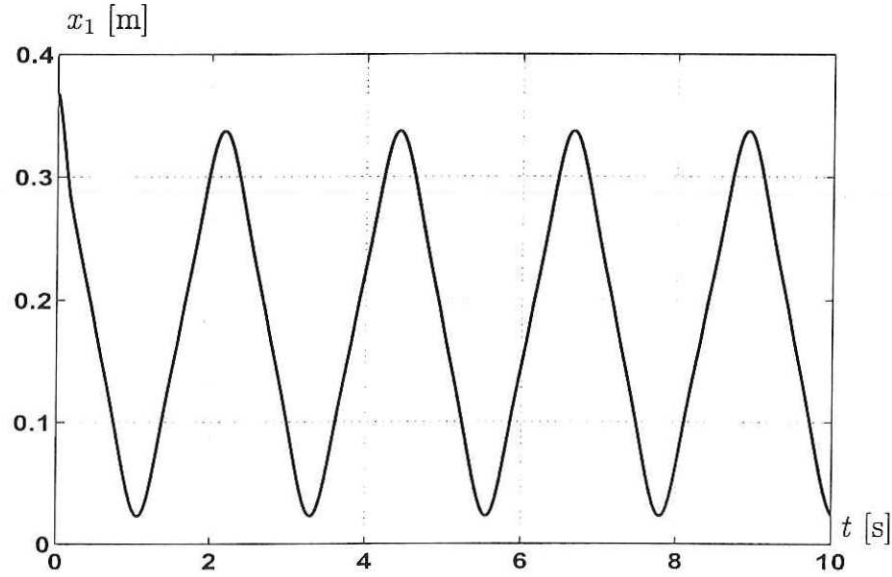


Figura 59: Trayectoria del extremo del robot correspondiente al eje cartesiano x_1 . [Li y Horowitz - simulación]

El error de campo de velocidad en coordenadas cartesianas viene dado por:

$$e(x, \dot{x}) = \dot{x} - v(x) \quad (177)$$

$$= J(q) [\dot{q} - v^*(q)] \quad (178)$$

$$= J(q) e^*(q, \dot{q}) \quad (179)$$

donde $e^*(q, \dot{q})$ es el error de campo de velocidad definido en coordenadas articulares. Por lo tanto se puede concluir que si se cumple el objetivo de control para coordenadas articulares contenido en (179) se cumple también con el objetivo de control en coordenadas cartesianas.

En la Figura 59 se muestra la trayectoria realizada por el extremo del robot correspondiente a la coordenada cartesiana x_1 , así como en la Figura 60 se muestra la trayectoria realizada por el extremo del robot correspondiente a la coordenada cartesiana x_2 . El campo de velocidad de referencia converge a la curva Γ_x lo cual se puede apreciar en la Figura 61 donde se presenta la trayectoria realizada por el extremo del robot correspondiente en el espacio cartesiano x_1-x_2 .

De la misma forma que con los anteriores controladores, el seguimiento del ovalo de Cassini se debe ejecutar a una velocidad constante de 0.35 [m/s]. En la Figura 62 se muestra la componente horizontal de la velocidad del extremo del robot, así como en

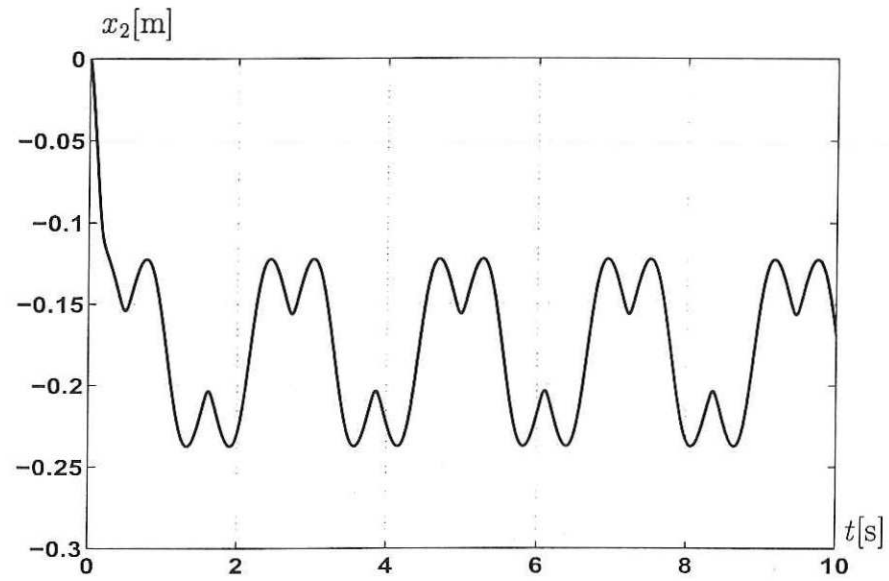


Figura 60: Trayectoria del extremo del robot correspondiente al eje cartesiano x_2 . [Li y Horowitz - simulación]

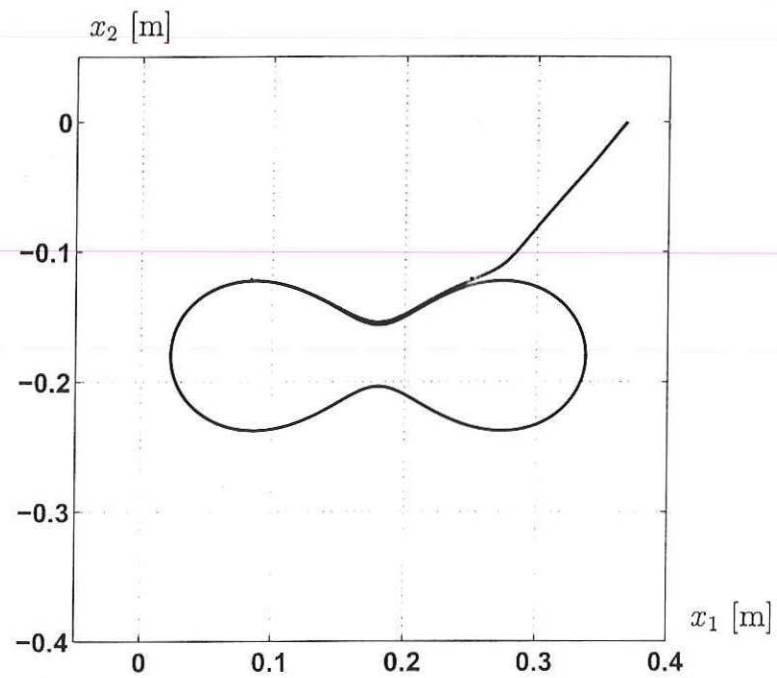


Figura 61: Trayectoria del extremo del robot en el espacio cartesiano. [Li y Horowitz - simulación]

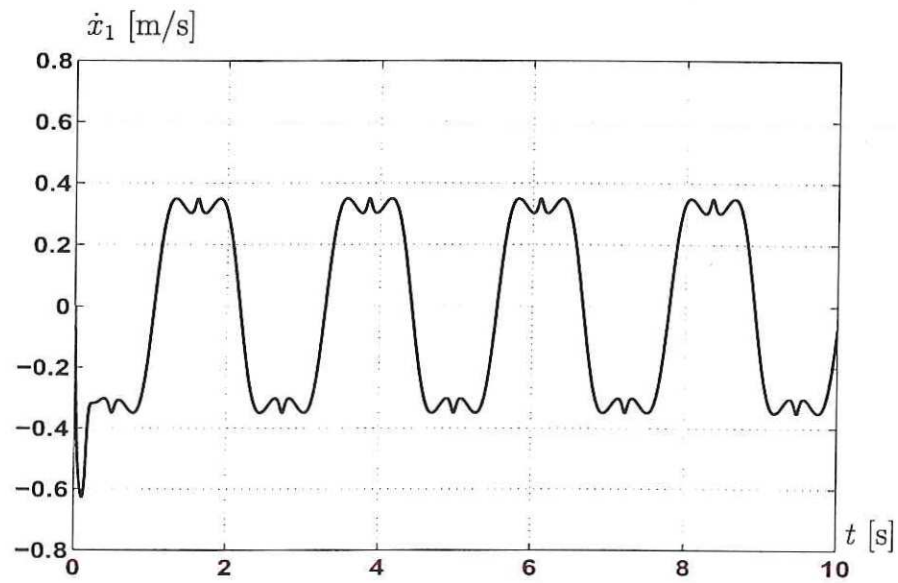


Figura 62: Velocidad del extremo del robot referente al eje cartesiano x_1 . [Li y Horowitz - simulación]

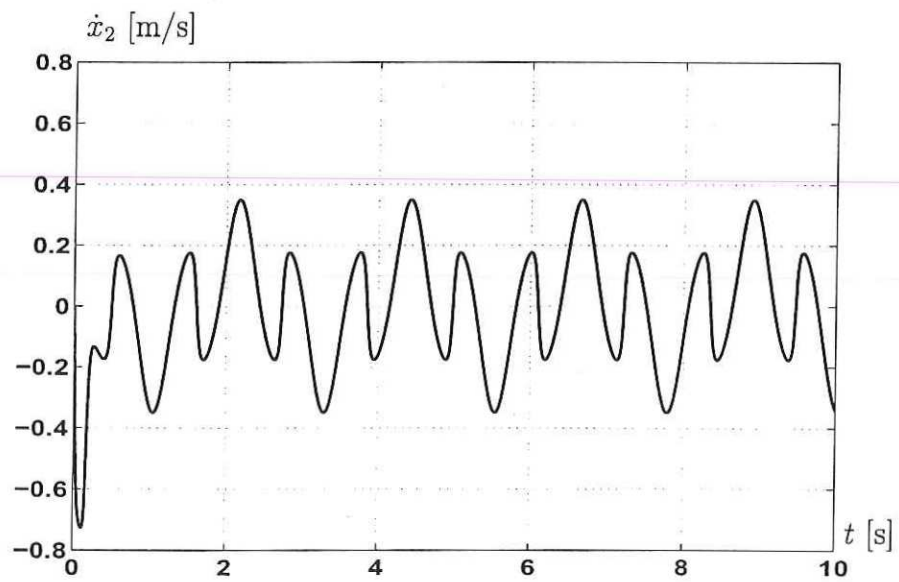


Figura 63: Velocidad del extremo del robot referente al eje cartesiano x_2 . [Li y Horowitz - simulación]

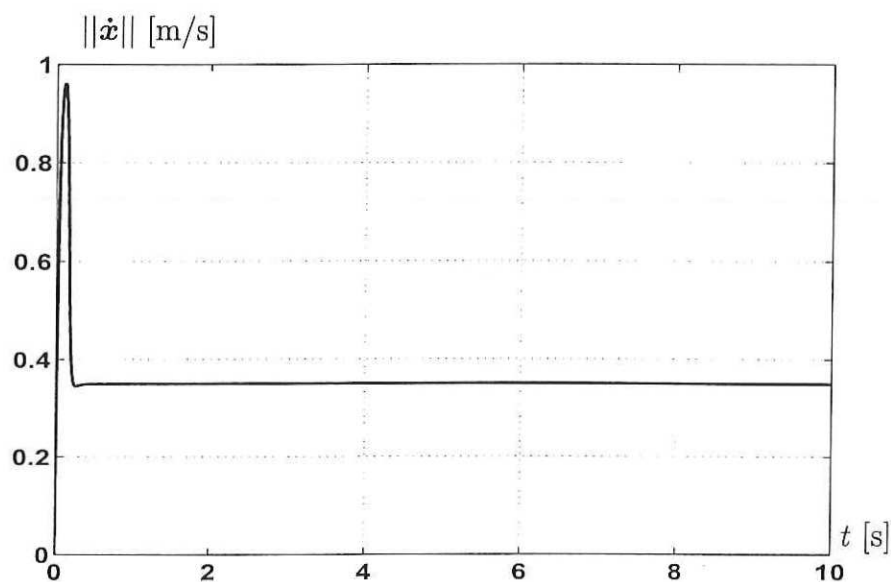


Figura 64: Norma de la velocidad del extremo del robot. [Li y Horowitz - simulación]

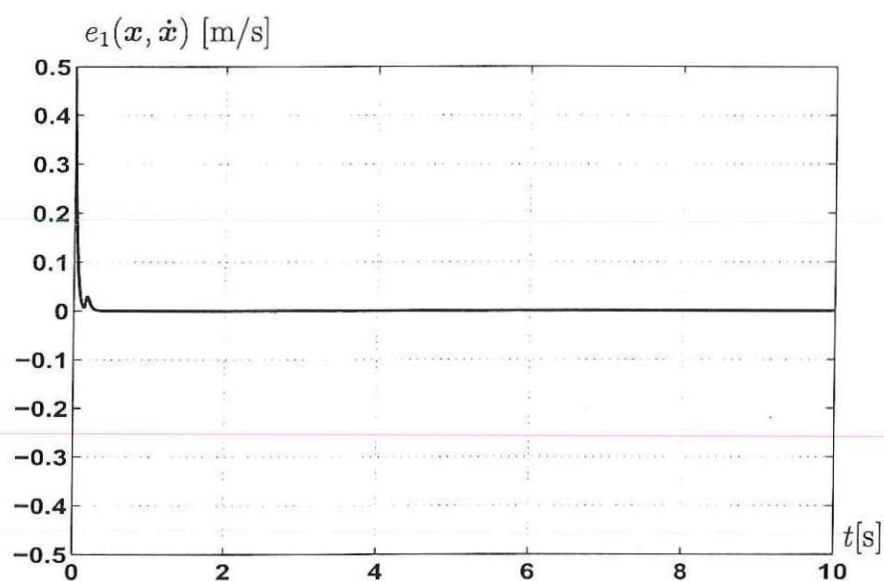


Figura 65: Error de campo de velocidad que corresponde al eje cartesiano x_1 . [Li y Horowitz - simulación]

la Figura 63 se muestra la componente vertical de la velocidad del extremo del robot. En la Figura 64 se presenta la norma de la velocidad $||\dot{x}||$.

En la Figura 65 se muestra el error de campo de velocidad correspondiente a la

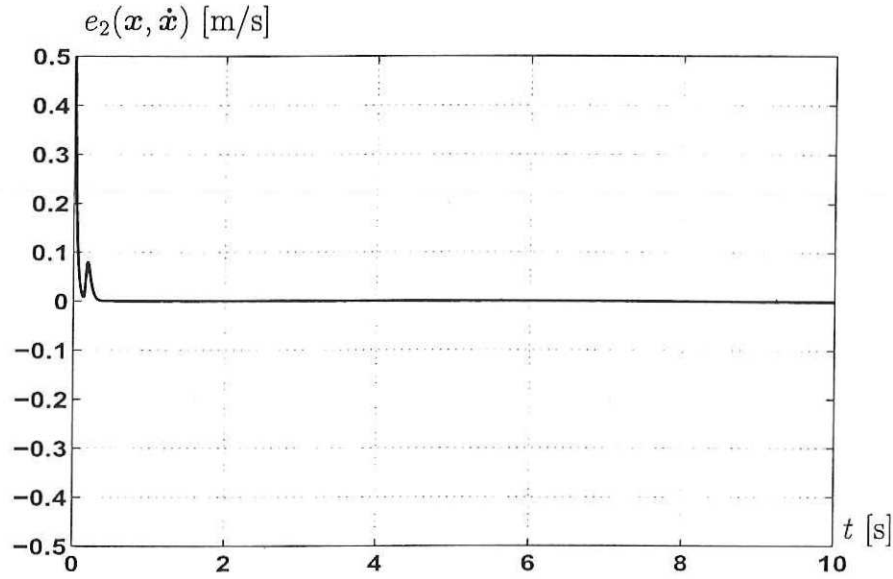


Figura 66: Error de campo de velocidad que corresponde al eje cartesiano x_2 . [Li y Horowitz - simulación]

coordenada cartesiana x_1 :

$$e_1(\mathbf{x}, \dot{\mathbf{x}}) = \dot{x}_1 - v_1(\mathbf{x}) = 0 \quad (180)$$

así como en la Figura 66 se muestra el error de campo de velocidad correspondiente a la coordenada cartesiana x_2 .

$$e_2(\mathbf{x}, \dot{\mathbf{x}}) = \dot{x}_2 - v_2(\mathbf{x}) = 0 \quad (181)$$

Se calcula el error resultante de campo de velocidad mediante:

$$\|e(\mathbf{x}, \dot{\mathbf{x}})\| = \sqrt{[e_1(\mathbf{x}, \dot{\mathbf{x}})]^2 + [e_2(\mathbf{x}, \dot{\mathbf{x}})]^2} \quad (182)$$

En la Figura 67 se presenta la norma del error $\|e(\mathbf{x}, \dot{\mathbf{x}})\|$. Se observa convergencia asintótica a cero del error de campo de velocidad.

En la Figura 68 se muestra el par aplicado a la primera articulación, así como en la Figura 69 se muestra el par aplicado a la segunda articulación.

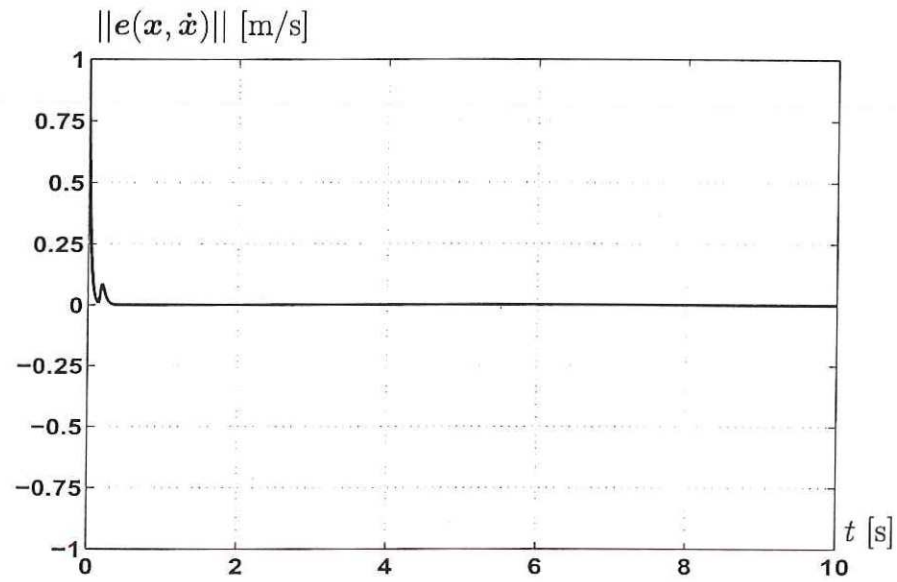


Figura 67: Norma del error de campo de velocidad. [Li y Horowitz - simulación]

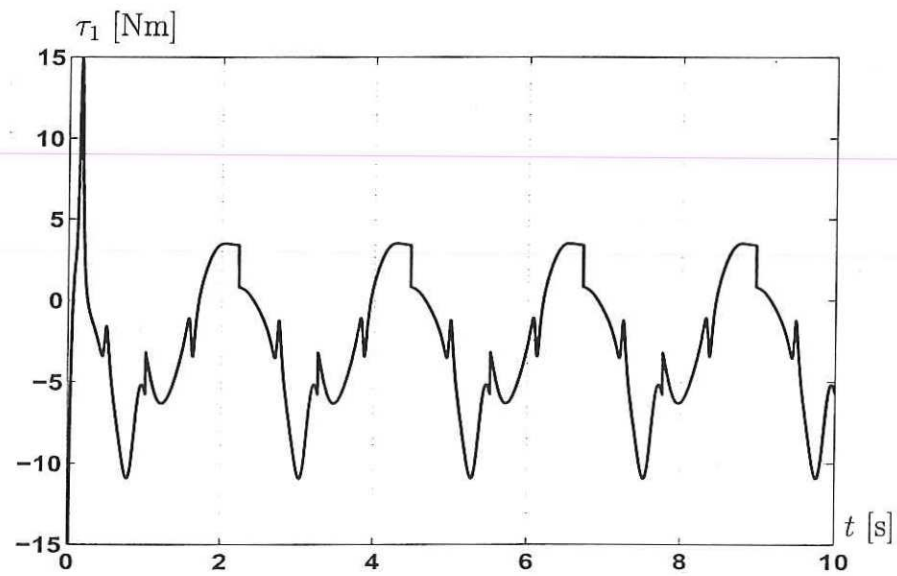


Figura 68: Par aplicado a la articulación 1 del robot. [Li y Horowitz - simulación]

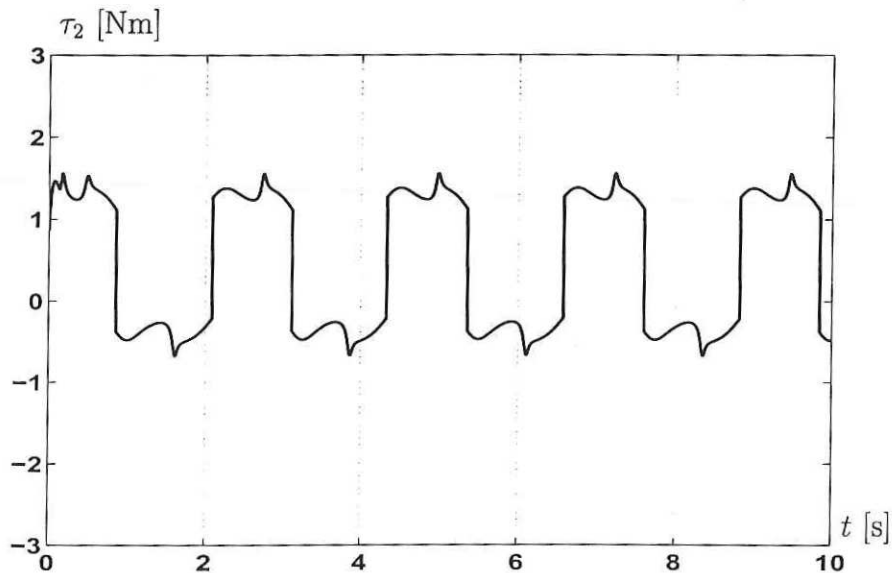


Figura 69: Par aplicado a la articulación 2 del robot. [Li y Horowitz - simulación]

V.4 Experimentos

Se implementó el algoritmo de Li y Horowitz descrito en las ecuaciones (160)–(165) en el robot Bugarín.

Se utilizó compensación de fricción viscosa y de Coulomb, es decir, se consideró el modelo del mecanismo manipulador:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + F_v\dot{q} + F_c \text{signo}(\dot{q}) = \tau \quad (183)$$

donde:

$$F_v = \begin{bmatrix} 0.274 & 0 \\ 0 & 0.144 \end{bmatrix} \quad (184)$$

es la matriz de coeficientes de fricción viscosa y:

$$F_c = \begin{bmatrix} 1.290 & 0 \\ 0 & 0.745 \end{bmatrix} \quad (185)$$

es la matriz de coeficientes de fricción de Coulomb.

De modo que δ cumpla con:

$$\delta > 2e\gamma\omega_d \quad (186)$$

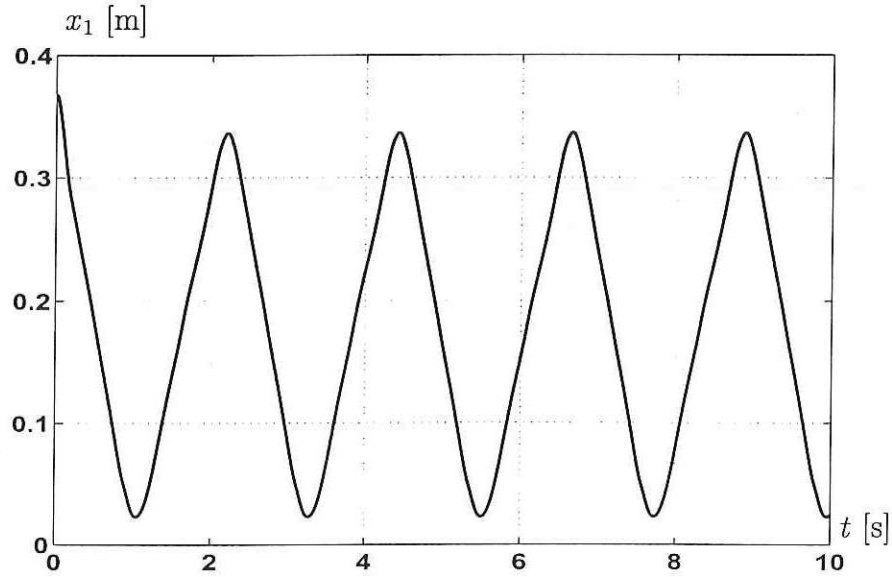


Figura 70: Trayectoria del extremo del robot correspondiente al eje cartesiano x_1 . [Li y Horowitz - experimento]

se designó:

$$\delta = 2e\gamma\omega_d + 1 \quad (187)$$

El valor de inercia del elemento virtual $n + 1$ se designó:

$$M_F = 2.5 \quad (188)$$

La energía cinética del sistema definida fué:

$$e = 2.42 \quad (189)$$

y la ganancia de control de $\bar{\tau}_{f_2}$:

$$\gamma = 7.0 \quad (190)$$

En las Figuras 70 y 71 se muestran las trayectorias realizadas por el extremo del robot en el espacio cartesiano, correspondientes a los ejes horizontal y vertical respectivamente. En la Figura 72 se presenta la trayectoria realizada por el extremo del robot en el espacio cartesiano x_1 y x_2 .

En las Figuras 73 y 74 se muestran los componentes de velocidad $\dot{x}_1$ y $\dot{x}_2$ correspondientes a las coordenadas cartesianas x_1 y x_2 respectivamente. La norma $\|\dot{\mathbf{x}}\|$ se

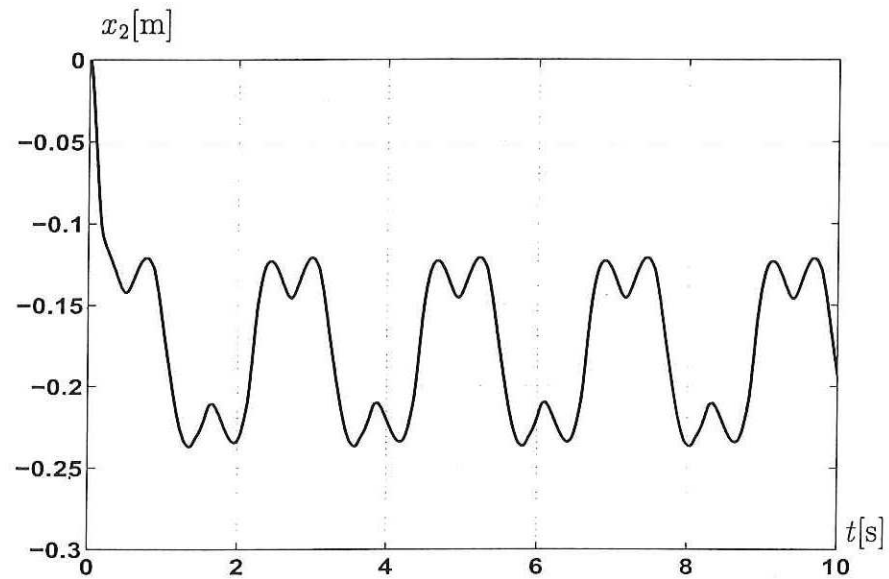


Figura 71: Trayectoria del extremo del robot correspondiente al eje cartesiano x_2 . [Li y Horowitz - experimento]

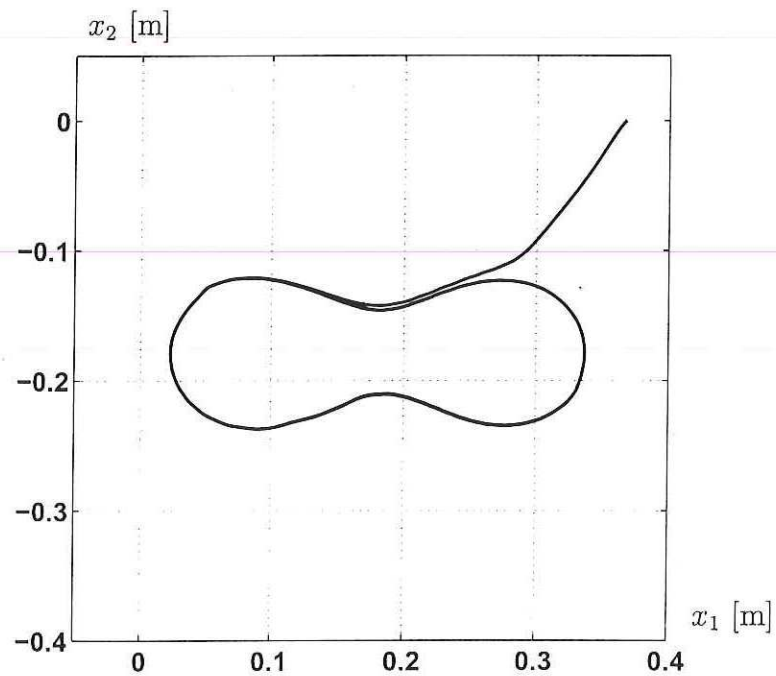


Figura 72: Trayectoria del extremo del robot en el espacio cartesiano x_1-x_2 . [Li y Horowitz - experimento]

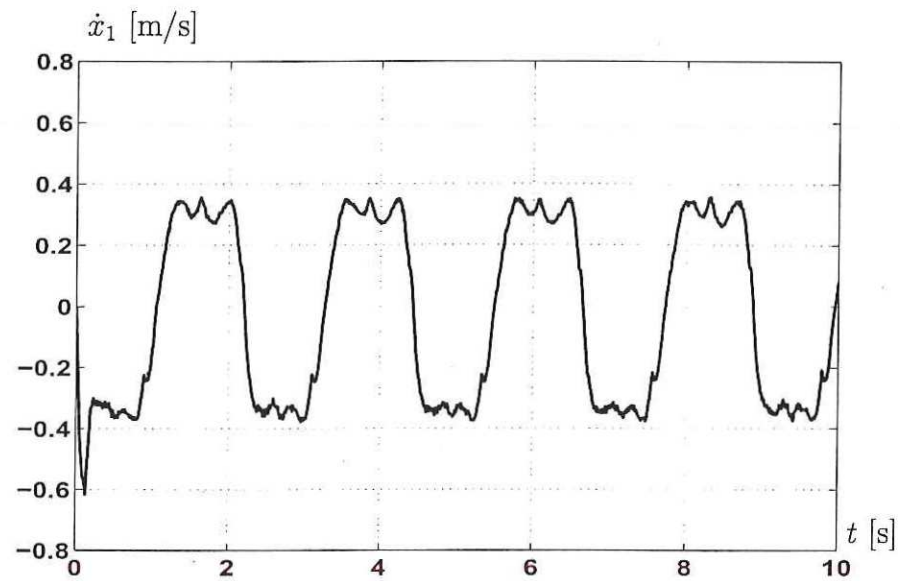


Figura 73: Velocidad del extremo del robot referente al eje cartesiano x_1 . [Li y Horowitz - experimento]

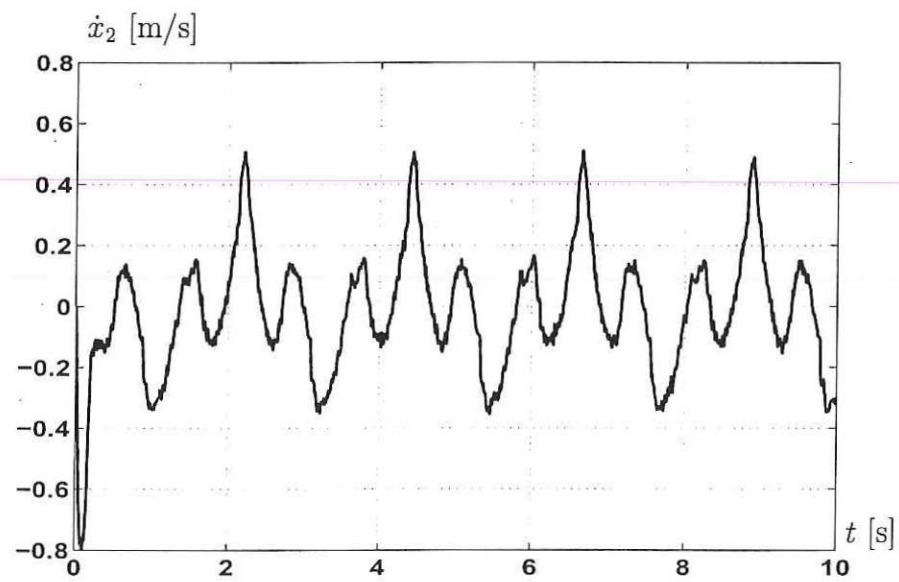


Figura 74: Velocidad del extremo del robot referente al eje cartesiano x_2 . [Li y Horowitz - experimento]

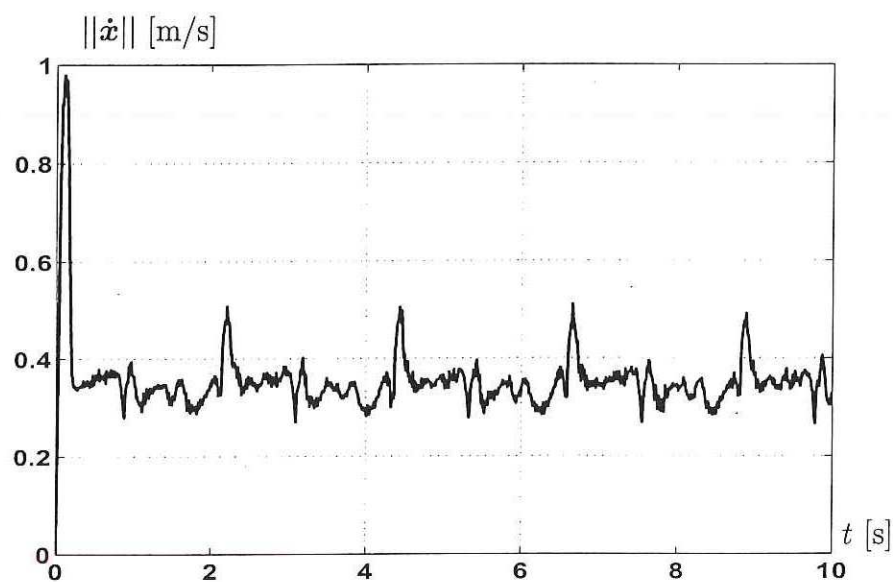


Figura 75: Norma de la velocidad del extremo del robot. [Li y Horowitz - experimento]

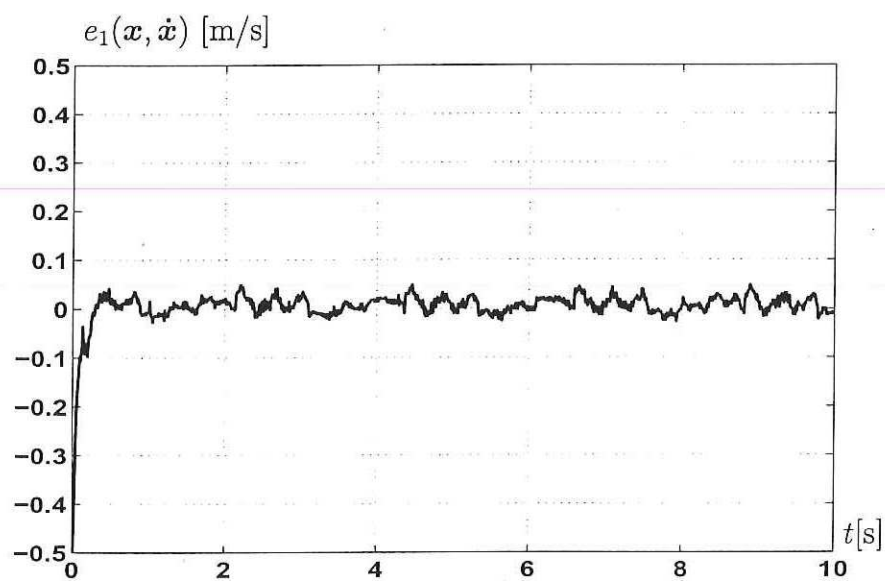


Figura 76: Error de campo de velocidad que corresponde al eje cartesiano x_1 . [Li y Horowitz - experimento]

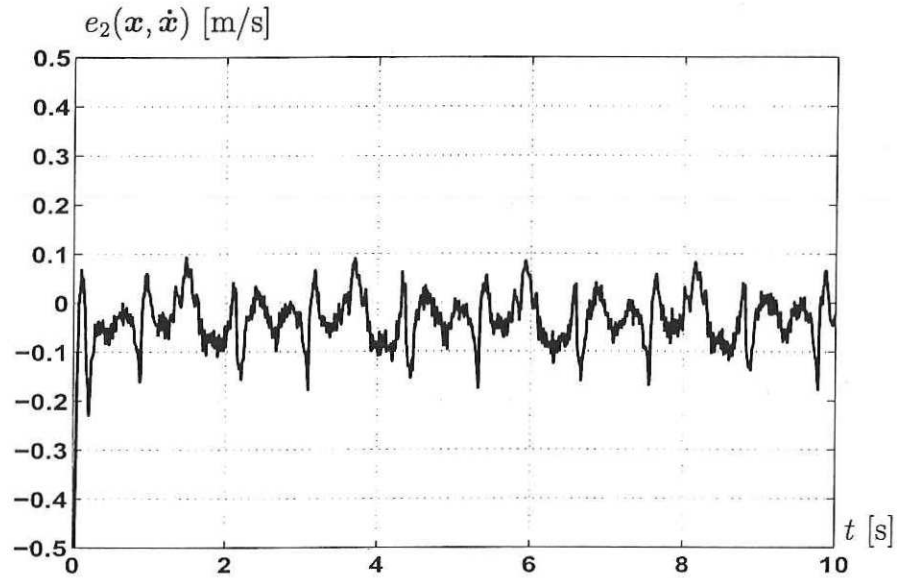


Figura 77: Error de campo de velocidad que corresponde al eje cartesiano x_2 . [Li y Horowitz - experimento]

calcula a través de:

$$\|\dot{\mathbf{x}}\| = \sqrt{\dot{x}_1^2 + \dot{x}_2^2} \quad (191)$$

En la Figura 75 se muestra la norma $\|\dot{\mathbf{x}}\|$ de las componentes de velocidad en el espacio cartesiano. Se observan oscilaciones entre 0.25 y 0.50 [m/s].

En las Figuras 76 y 77 se muestran los componentes de error de campo de velocidad e_1 y e_2 correspondientes a las coordenadas cartesianas x_1 y x_2 respectivamente. En la Figura 78 se muestra la norma $\|e(\mathbf{x}, \dot{\mathbf{x}})\|$ de las componentes de error de campo de velocidad en el espacio cartesiano.

Finalmente en las Figuras 79 y 80 se muestran los pares de control aplicados a la primera y segunda articulación respectivamente.

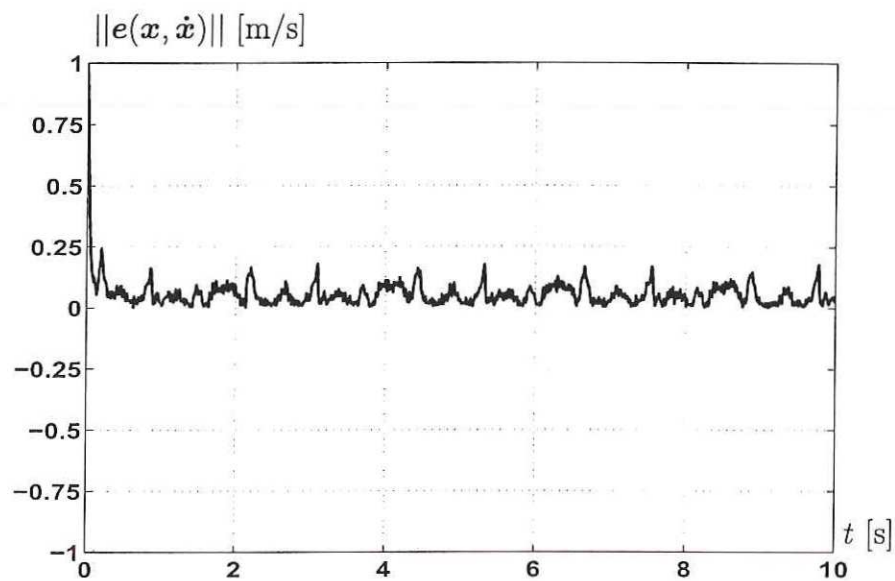


Figura 78: Norma del error de campo de velocidad. [Li y Horowitz - experimento]

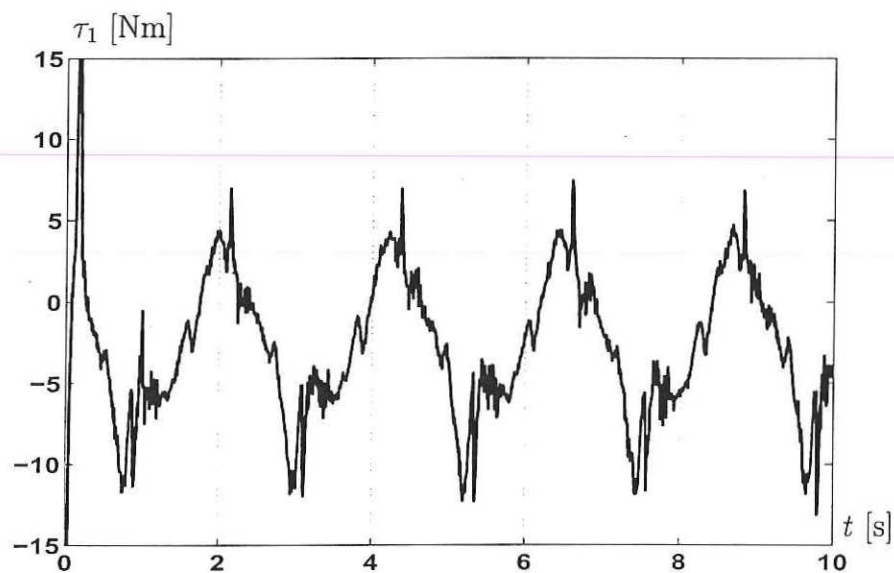


Figura 79: Par aplicado a la articulación 1 del robot. [Li y Horowitz - experimento]

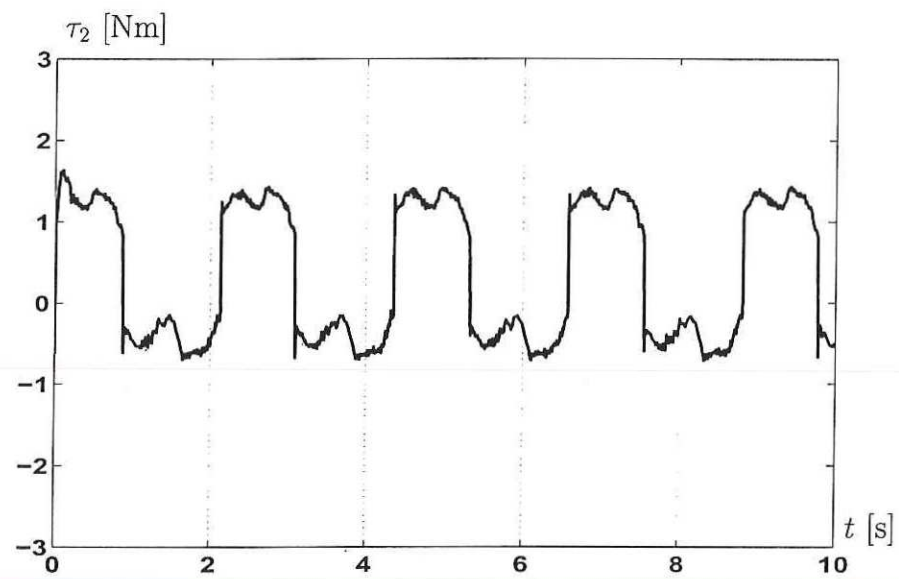


Figura 80: Par aplicado a la articulación 2 del robot. [Li y Horowitz - experimento]

Capítulo VI

Conclusiones

En los capítulos previos se estudiaron tres algoritmos que resuelven el problema de control de campo de velocidad. Es decir, en base a un campo vectorial deseado $\mathbf{v}(\mathbf{x})$ se presentan leyes de control que cumplen con:

$$\lim_{t \rightarrow \infty} \mathbf{e}(\mathbf{x}(t), \dot{\mathbf{x}}(t)) = \lim_{t \rightarrow \infty} [\dot{\mathbf{x}}(t) - \mathbf{v}(\mathbf{x}(t))] = \mathbf{0} \quad (192)$$

En este trabajo de tesis se llevaron a cabo experimentos con los tres algoritmos de control. De tales experimentos se puede concluir lo siguiente:

En el experimento llevado a cabo con el control por dinámica inversa, la norma del error $\|\mathbf{e}(\mathbf{x}, \dot{\mathbf{x}})\|$ presentó oscilaciones entre 0 y 0.3 [m/s]. Como se puede apreciar en la Figura 29, la norma $\|\dot{\mathbf{x}}\|$ oscila entre 0.2 y 0.65 [m/s]. Esto se puede calificar como un pobre desempeño, pues el seguimiento del campo de velocidad se hace a velocidades de aproximadamente de 85.0% por encima y 57.0% por debajo de 0.35 [m/s], lo indicado por la referencia. El deterioro en el seguimiento del campo de velocidad se atañe a la poca robustez del algoritmo en presencia de fuerzas de fricción así como a una deficiente compensación de los pares ocasionados por fricción.

En el experimento llevado a cabo con el control en estructura PD+ con compesación de fricción utilizando el modelo de Dahl, la norma del error $\|\mathbf{e}(\mathbf{x}, \dot{\mathbf{x}})\|$ presentó oscilaciones entre 0 y 0.1 [m/s]. Como se muestra en la Figura 52, la norma $\|\dot{\mathbf{x}}\|$ oscila entre 0.3 y 0.45 [m/s]. Esto es aproximadamente un 28.6% por encima de la referencia de velocidad y un 14.3% por debajo de la misma.

Finalmente en el experimento con el control de Li y Horowitz, la norma del error $\|\mathbf{e}(\mathbf{x}, \dot{\mathbf{x}})\|$ mostró oscilaciones entre 0 y 0.2 [m/s]. En la Figura 75 se aprecia la norma $\|\dot{\mathbf{x}}\|$ que presenta oscilaciones entre 0.25 y 0.50 [m/s], un 42% por encima de 0.35 [m/s] y un 28.6% por debajo aproximadamente.

La compensación de fricción utilizando el modelo de Coulomb utiliza dentro de su

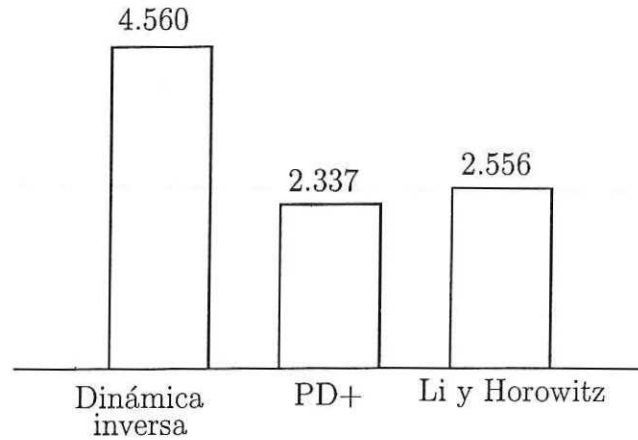


Figura 81: Comparación de los índices de desempeño

definición la función *signo*. El valor de *signo* cambia de -1 a $+1$ de acuerdo a los cruces por cero de la velocidad articular correspondiente. Es decir, la función *signo* implica cambios súbitos del par que debe compensar fricción para tal articulación, lo cual presenta complicaciones de implementación. Aún cuando el procesamiento del algoritmo de control indique a los actuadores que deberán modificar los pares aplicados, dependerá ya de la capacidad de respuesta de estos la compensación adecuada de fricción.

Por otro lado, el modelo de compensación de fricción propuesto por Dahl, describe de mejor forma las causas que ocasionan la aparición de fuerzas de fricción entre superficies. Este modelo predice de forma eficiente los efectos de fricción y además evita el uso de la función *signo*, sin embargo, es un modelo dinámico y presenta un mayor grado de complejidad en cuanto a diseño.

Para poder comparar el desempeño de los controladores en experimentación se utiliza el índice dado por el valor cuadrático medio del error de campo de velocidad:

$$indice = \sqrt{\left[\frac{1}{T} \int_0^T \|e(x(t))\|^2 dt \right]} \quad (193)$$

donde $T = 10$ [s].

En la Figura 81 se muestran los índices de desempeño que resultaron de los experimentos con los controladores por dinámica inversa, en estructura PD+ y de Li y Horowitz. Como se observa en la Figura 81, el controlador por dinámica inversa presentó el peor desempeño de los tres, con un índice de 4.560. El controlador de Li y

Horowitz presentó un índice de 2.556. El mejor desempeño lo tuvo el controlador PD+ con compensación de fricción de Dahl, con un índice de 2.337.

VI.1 Propuestas de trabajo a futuro

- Se propone la investigación y desarrollo de nuevos algoritmos de control mediante campo de velocidad. Los algoritmos estudiados en el presente trabajo, surgen como consecuencia de la revisión bibliográfica sobre el tema (control de Li y Horowitz), de modificaciones a propuestas existentes (control por dinámica inversa [Moreno, 1999]) así como del diseño de algoritmos inspirados en otros existentes para control de movimiento (control con estructura PD+).
- Considerando la susceptibilidad de los algoritmos de control mediante campo de velocidad a los efectos causados por fricción, se propone investigar las formas de compensación de manera general, concentrándose en el estudio de compensadores dinámicos, los cuales presentan un mayor reto de diseño a nivel analítico. El control con estructura PD+ presentado en esta tesis, utiliza un observador sumamente sencillo, el cual demostró atenuar los efectos de fricción de forma considerable. Se puede pensar en mejores resultados investigando y probando el uso observadores con estructuras más complejas.
- En el presente trabajo de tesis, se considera una metodología para el diseño de campos vectoriales que convergen a una curva cerrada. Como trabajo a futuro se propone investigar nuevos métodos para la definición de campos vectoriales con características arbitrarias, por ejemplo, para el caso cuando no se cuenta con una ecuación paramétrica que defina la curva a la que se desea converjan las flechas del campo vectorial. También representa un reto teórico el caso para cuando se trata de seguir una curva abierta en lugar de una curva cerrada.
- Se propone investigar la generalización de campo de velocidad de mecanismos para el caso de robots redundantes ($n > m$, donde n es el número de g.d.l. y m es la dimensión del espacio de trabajo). Los robots redundantes tienen aplicación en

la realización de tareas en presencia de obstáculos, actividades en que los robots no redundantes están limitados.

- Otra alternativa para la definición de tareas en el espacio cartesiano, puede ser inspirada en la metodología de control denominada punto-a-punto. Por ejemplo, se diseña un campo de velocidad que converja a un punto en el espacio de trabajo, y se modifica este en tiempo real, de manera que el robot corrija el error de campo de velocidad trasladando el extremo del robot al nuevo punto en el espacio, y así sucesivamente.
- También se propone estudiar la aplicación de algoritmos de control para el caso de trayectorias definidas en el espacio tridimensional. Esto involucra la definición campos vectoriales que converjan a curvas en el espacio $\mathbb{R}^3$, y la experimentación con mecanismos de 3 g.d.l.
- *Campo de velocidad de mecanismos* es una propuesta novedosa para el control de robots, por lo que se considera, existe un amplio espectro de posibilidades de aplicación dentro del contexto del control automático.

Bibliografía

Craig J. 1989. "Introduction to robotics". Addison-Wesley Publishing Company. Reading, MA. Segunda edición. 450 pp.

Kelly R. y Santibáñez V. 2003. "Control de movimiento de robots manipuladores". Pearson Educación / Prentice Hall. Primera Edición. Madrid. 323 pp.

Li P. y Horowitz R. 1999. "Passive velocity field control of mechanical manipulators". IEEE Transactions on Robotics and Automation, 15(4):751-763 p.

Moreno E. 1999. "Control de robots manipuladores mediante campo de velocidad". Tesis de maestría, CICESE, Ensenada, Baja California, México. 111 pp.

Moreno E. 2002. "Control jerárquico de robots manipuladores basado en lazos internos de velocidad". Tesis de doctorado, CICESE, Ensenada, Baja California, México. 337 pp.

Sciavico L. y Siciliano B. 1996. "Modeling and control of robot manipulators". The McGraw Hill companies, Inc. New York, NY. Primera Edición. 358 pp.

Spong M. y Vidyasagar M. 1989. "Robot dynamics and control". John Wiley & Sons, New York, NY. Primera Edición. 336 pp.

Apéndices

Apéndice A

Espacios de funciones y estabilidad en el sentido de Lyapunov

El contenido de este apéndice proviene de (Kelly y Santibáñez, 2003).

A.1 Espacios de funciones

Los siguientes conceptos matemáticos son sumamente útiles en el estudio de las propiedades dinámicas de sistemas descritos por ecuaciones diferenciales.

Definición 1 (Espacio L_2^n). El espacio L_2^n consiste de todas las funciones continuas $f : \mathbb{R}_+ \rightarrow \mathbb{R}^n$ tales que:

$$\int_0^\infty f(t)f(t)^T dt = \int_0^\infty \|f(t)\|^2 dt < \infty \quad (194)$$

Es decir, la función f pertenece al espacio L_2^n ($f \in L_2^n$) si la integral del cuadrado de su norma euclidiana está acotado superiormente.

Definición 2 (Espacio L_∞^n). El espacio L_∞^n consiste de todas las funciones continuas $f : \mathbb{R}_+ \rightarrow \mathbb{R}^n$ tales que sus normas euclidianas sean acotadas:

$$\sup_{t \geq 0} \|f(t)\| < \infty \quad (195)$$

Se debe hacer la observación en este momento que una función continua f perteneciente al espacio L_2^n puede no tener un límite asintótico. A continuación se presenta un resultado obtenido en la bibliografía de análisis funcional que proporciona condiciones suficientes sobre funciones pertenecientes al espacio L_2^n para que éstas tiendan asintóticamente a cero.

Lema 1. Considérese una función continua $f : \mathbb{R}_+ \rightarrow \mathbb{R}^n$. Supóngase que la función f satisface la siguientes hipótesis:

$$f, \dot{f} \in L_\infty^n \quad (196)$$

$$f \in L_2^n \quad (197)$$

Entonces necesariamente se tiene que:

$$\lim_{t \rightarrow \infty} f(t) = 0 \quad (198)$$

A.2 Estabilidad en el sentido de Lyapunov

En esta sección se presentan teoremas referentes a la teoría de Lyapunov, en concreto referentes al llamado *segundo método de Lyapunov* o *método directo de Lyapunov*.

La teoría de estabilidad de Lyapunov tiene como objetivo principal el estudio del comportamiento de sistemas dinámicos descritos por ecuaciones diferenciales de la forma:

$$\dot{x}(t) = f(t, x(t)) \quad x(0) \in \mathbb{R}^n \quad \forall t \geq 0 \quad (199)$$

donde el vector $x(t) \in \mathbb{R}^n$ se refiere al estado del sistema dinámico representado por (199) y $x(0) \in \mathbb{R}^n$ se denomina estado inicial. La función $f : \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ es una función continua en t y $x(t)$, y se supone que es tal que:

- La ecuación (199) tiene una solución única en el intervalo $[0, \infty)$ correspondiente a cada condición inicial de $x(0)$.
- Si $x(t)$ es la solución de (199) correspondiente a la condición inicial $x(0)$, entonces $x(t)$ depende de una manera continua del estado inicial $x(0)$.

Teorema 1 (Estabilidad). El origen $x = 0 \in \mathbb{R}^n$ es un estado de equilibrio estable de (199), si existe una función candidata de Lyapunov $V(t, x)$ tal que su derivada temporal satisfaga:

$$\dot{V}(t, x) \leq 0 \quad \forall t \geq 0 \quad \text{al menos para } \|x\| \text{ pequeña} \quad (200)$$

Teorema 2 (Estabilidad y acotamiento de las soluciones). El origen $\mathbf{x} = \mathbf{0} \in \mathbb{R}^n$ es un equilibrio estable, y las soluciones $\mathbf{x}(t)$ están acotadas para toda condición inicial $\mathbf{x}(0) \in \mathbb{R}^n$, si existe una función candidata de Lyapunov que sea definida positiva (globalmente), radialmente desacotada, $V(t, \mathbf{x})$, tal que su derivada temporal satisfaga:

$$\dot{V}(t, \mathbf{x}) \leq 0 \quad \forall t \geq 0 \quad \forall \mathbf{x} \in \mathbb{R}^n \quad (201)$$

Teorema 3 (Estabilidad asintótica global). El origen $\mathbf{x} = \mathbf{0}$ es un estado de equilibrio asintóticamente estable en forma global de (199), si existe una función candidata de Lyapunov $V(t, \mathbf{x})$ definida positiva (globalmente), radialmente desacotada, y menguante, tal que su derivada satisfaga:

- $\dot{V}(t, \mathbf{0}) = 0 \quad \forall t \geq 0$
- $\dot{V}(t, \mathbf{x}) < 0 \quad \forall t \geq 0, \quad \forall \mathbf{x} \neq \mathbf{0} \in \mathbb{R}^n$

Apéndice B

Códigos de Simnon

Como se mencionó anteriormente, las simulaciones se llevaron a cabo utilizando Simnon 2.0. En este apéndice se presentan los listados de código programados.

Para generar simulaciones en Simnon se definen básicamente cuatro bloques de código: el modelo del robot, el algoritmo de control, un medio de conexión de variables en común, y un macro que defina el sistema a simular.¹

B.1 Control basado en dinámica inversa

B.1.1 Macro

MACRO RDINV

```
" Descripción: Robot Bugarían bajo control por dinámica
"             inversa.
" Autor:      José C. Navarrete García
" Fecha:      10/9/02
```

```
" Comandos:
```

```
syst dinversa robobuga cdinv
```

```
"store x[dinversa] y[dinversa]
"store dx[dinversa] dy[dinversa]
store tau1[dinversa] tau2[dinversa]
"store ecvx[dinversa] ecvy[dinversa]
"store nrmvel[dinversa] nrmerr[dinversa]
```

```
" Inicializaciones:
```

```
init xq1[robobuga]:0.785398633
init xq2[robobuga]:1.570796326795
```

```
init xq3[robobuga]:0.0
init xq4[robobuga]:0.0
```

```
newplot
```

¹Las simulaciones se ejecutan con un incremento en tiempo máximo de 1.0 mseg.

```
axes v -1 1 h -1 1
plot y[robobuga](x[robobuga])
```

```
END
```

B.1.2 Controlador

CONTINUOUS SYSTEM DINVERSA

```
" Descripción: Controlador por dinámica inversa
"               mediante campo de velocidad.
```

```
" Autor:       José C. Navarrete García
```

```
" Fecha:       10/3/2002
```

```
" Entradas y salidas:
```

```
INPUT    q1 q2 dq1 dq2
```

```
OUTPUT tau1 tau2
```

```
" Tiempo:
```

```
TIME t
```

```
" Ecuaciones:
```

```
q1kn=q1
```

```
q2kn=q2
```

```
dq1kn=dq1
```

```
dq2kn=dq2
```

```
" Modelo dinámico del robot de Bugarín
```

```
" Matriz de inercia
```

```
m11=0.33530+0.02436*cos(q2kn)
```

```
m12=0.01267+0.01218*cos(q2kn)
```

```
m21=0.01267+0.01218*cos(q2kn)
```

```
m22=0.01267
```

```
" Matriz de fuerzas centrífugas y de Coriolis
```

```
c11=-0.02436*sin(q2kn)*dq2kn
```

```
c12=-0.01218*sin(q2kn)*dq2kn
```

```
c21= 0.01218*sin(q2kn)*dq1kn
```

```
c22= 0.0
```

```
" Vector de pares gravitacionales
```

```
g1=1.17310*g*sin(q1kn)+0.04685*g*sin(q1kn+q2kn)
```

```
g2=0.04685*g*sin(q1kn+q2kn)
```

```

" La posición en coordenadas de trabajo
" El modelo cinemático

x= 0.26*sin(q1kn)+0.26*sin(q1kn+q2kn)
y=-0.26*cos(q1kn)-0.26*cos(q1kn+q2kn)

" Derivada temporal de x e y

dx=0.26*cos(q1kn)*dq1kn+0.26*cos(q1kn+q2kn)*(dq1kn+dq2kn)
dy=0.26*sin(q1kn)*dq1kn+0.26*sin(q1kn+q2kn)*(dq1kn+dq2kn)

" Campo de Velocidad que dibuja
" el ovalo de Cassini

" Definicion de la curva a seguir:

f1=2*a*a*((x-xc)*(x-xc)-(y-yc)*(y-yc))+b*b*b*b
f=(r^2)+a*a*a*a-f1
r=(x-xc)*(x-xc)+(y-yc)*(y-yc)

" El gradiente de f

delfx=4*(x-xc)*r-4*a*a*(x-xc)
delfy=4*(y-yc)*r+4*a*a*(y-yc)

dist=(delfx*delfx+delfy*delfy)^0.5

" La derivada temporal de f

dtf=delfx*dx+delfy*dy

n1=2*(x-xc)*dx+2*(y-yc)*dy
n2=2*(x-xc)*dx-2*(y-yc)*dy

" Derivada temporal del gradiente

dtdelfx=4*(x-xc)*n1+4*dx*r-4*a*a*dx
dtdelfy=4*(y-yc)*n1+4*dy*r+4*a*a*dy

dtdist=(delfx*dtdelfx+delfy*dtdelfy)/dist

" Ganancia k1(q)

k1=ko/(abs(f)*dist+epsilon)

" Derivada temporal de k1(q)

dendtk1=((abs(f)*dist+epsilon)^2)
dtk1=-ko*(abs(f)*dtdist+tanh(lambda*f)*dtf*dist)/dendtk1

```

```

" Ganancia c1(q)
c1=co*exp(-alpha*abs(f))

" Derivada temporal de c1(q)
dtc1=-alpha*co*tanh(lambda*f)*dtf*exp(-alpha*abs(f))

" El campo de velocidad
vx=-k1*f*delfx-c1*delfy/dist
vy=-k1*f*delfy+c1*delfx/dist

" Variables auxiliares para calcular la derivada
temporal del campo de velocidad
ax=-dtk1*f*delfx-dtc1*delfy/(dist)
ay=-dtk1*f*delfy+dtc1*delfx/(dist)

terx=c1*(dist*dtdelfy-dtdist*delfy)/(dist*dist)
tery=c1*(dist*dtdelfx-dtdist*delfx)/(dist*dist)

" Derivada temporal del campo de velocidad
dvtvx=-k1*(dtf*delfx+f*dtdelfx)-terx+ax
dvtvy=-k1*(dtf*delfy+f*dtdelfy)+tery+ay

" Norma de la velocidad del extremo del robot
norma=sqrt(dx*dx+dy*dy)

" El Jacobiano J(q)

j11=0.26*cos(q1kn)+0.26*cos(q1kn+q2kn)
j12=0.26*cos(q1kn+q2kn)
j21=0.26*sin(q1kn)+0.26*sin(q1kn+q2kn)
j22=0.26*sin(q1kn+q2kn)

JJ=j11*j22-j12*j21

" Derivada temporal del Jacobiano
pj11dq1=-0.26*sin(q1kn)-0.26*sin(q1kn+q2kn)
pj12dq1=-0.26*sin(q1kn+q2kn)
pj21dq1= 0.26*cos(q1kn)+0.26*cos(q1kn+q2kn)
pj22dq1= 0.26*cos(q1kn+q2kn)

pj11dq2=-0.26*sin(q1kn+q2kn)
pj12dq2=-0.26*sin(q1kn+q2kn)
pj21dq2= 0.26*cos(q1kn+q2kn)
pj22dq2= 0.26*cos(q1kn+q2kn)

```



```

dtj11=pj11dq1*dq1kn+pj11dq2*dq2kn
dtj12=pj12dq1*dq1kn+pj12dq2*dq2kn
dtj21=pj21dq1*dq1kn+pj21dq2*dq2kn
dtj22=pj22dq1*dq1kn+pj22dq2*dq2kn

" Error de campo de velocidad

ecvx=vx-dx
ecvy=vy-dy

" Cálculo de la norma de la velocidad

nrmvel=sqrt(dx*dx+dy*dy)

"Cálculo de la norma del error

nrmerr=sqrt(ecvx*ecvx+ecvy*ecvy)

" Aqui se define la parte del controlador que tiene la forma
" dl=dt[v(x)]+kv[vx-dx]=kp*zeta-dtJ(q)vel

dl1=dtvx+(kk1*ecvx)-(dtj11*dq1kn+dtj12*dq2kn)
dl2=dtvy+(kk2*ecvy)-(dtj21*dq1kn+dtj22*dq2kn)

" Definición del vector uo donde
" uo=(J(q))^-1 * dl

uo1=( j22*dl1-j12*dl2)/JJ
uo2=(-j21*dl1+j11*dl2)/JJ

aa1=m11*uo1+m12*uo2+c11*dq1kn
aa2=m21*uo1+m22*uo2+c21*dq1kn

tau1=aa1+c12*dq2kn+g1+b1*dq1kn+fc1*sign(dq1kn)
tau2=aa2+c22*dq2kn+g2+b2*dq2kn+fc2*sign(dq2kn)

kk1:10
kk2:10

" Parámetros del Ovalo de Cassini

a:0.110
b:0.1125

xc:0.180
yc:-0.180

" Parámetros del campo de velocidad

ko:1.00

```

co:0.35

epsilon:0.000001 " 1E-6

lambda:5000

alpha:500

" Gravedad

g:9.81

" Constantes de fricción viscosa y de Coulomb

fc1:1.290

fc2:0.745

b1:0.274

b2:0.144

END

B.1.3 Robot Bugarín

CONTINUOUS SYSTEM ROBOBUGA

" Descripción: Modelo dinámico del Robot Bugarín

" Autor: José C. Navarrete García

" Fecha: 2/1/03

" Entradas y Salidas.

INPUT tau1 tau2

OUTPUT q1 q2 dq1 dq2

" Estados, derivadas y tiempo:

STATE xq1 xq2 xq3 xq4

DER dxq1 dxq2 dxq3 dxq4

TIME t

" Modelo dinámico del robot de Bugarín:

" Matriz de inercia

m11=0.33530+0.02436*cos(xq2)

m12=0.01267+0.01218*cos(xq2)

m21=0.01267+0.01218*cos(xq2)

m22=0.01267

detm=m11*m22-m12*m21

" Matriz de fuerzas centrífugas y de Coriolis

```

c11=-0.02436*sin(xq2)*xq4
c12=-0.01218*sin(xq2)*xq4
c21= 0.01218*sin(xq2)*xq3
c22= 0.0

" Vector de pares gravitacionales

g1=1.17310*g*sin(xq1)+0.04685*g*sin(xq1+xq2)
g2=0.04685*g*sin(xq1+xq2)

" El sistema en ecuaciones de estado:

h1=(c11*xq3)+(c12*xq4)+g1+b1*xq3+fc1*sign(xq3)
h2=(c21*xq3)+(c22*xq4)+g2+b2*xq4+fc2*sign(xq4)

s1=tau1-h1
s2=tau2-h2

dxq1=xq3
dxq2=xq4
dxq3=(m22*s1-m12*s2)/detm
dxq4=(-m12*s1+m11*s2)/detm

" Información de salida: posiciones y velocidades
" articulares.

q1=xq1
q2=xq2

dq1=xq3
dq2=xq4

" El modelo cinemático

x= 0.26*sin(xq1)+0.26*sin(xq1+xq2)
y=-0.26*cos(xq1)-0.26*cos(xq1+xq2)

" Valor de parámetros:

g:9.81

fc1:1.290
fc2:0.745

b1:0.274
b2:0.144

END

```

B.1.4 Conexiones

CONNECTING SYSTEM CDINV

" Autor: José C. Navarrete García

" Fecha: 10/3/2002

" Tiempo:

TIME t

" Conexiones:

tau1[robobuga]=tau1[dinversa]

tau2[robobuga]=tau2[dinversa]

q1[dinversa]=q1[robobuga]

q2[dinversa]=q2[robobuga]

dq1[dinversa]=dq1[robobuga]

dq2[dinversa]=dq2[robobuga]

END

B.2 Control en estructura PD+

B.2.1 Macro

MACRO VPDMAS

" Descripción: Robot Bugarín bajo control mediante
" campo de velocidad en estructura PD+
" para coordenadas cartesianas.

" Autor: José C. Navarrete García

" Fecha: 10/9/02

" Comandos:

syst pdmas1 robobuga cpdmas

"store x[pdmas1] y[pdmas1]

"store dx[pdmas1] dy[pdmas1]

"store tau1[pdmas1] tau2[pdmas1]

"store ecvx[pdmas1] ecvy[pdmas1]

store nrmvel[pdmas1] nrmerr[pdmas1]

" Inicializaciones:

init xq1[robobuga]:0.785398633

init xq2[robobuga]:1.570796326795

```

init xq3[robobuga]:0.0
init xq4[robobuga]:0.0

newplot
axes v -1 1 h -1 1
plot y[robobuga](x[robobuga])

END

```

B.2.2 Controlador

CONTINUOUS SYSTEM PDMAS1

```

" Descripción: Controlador mediante campo de velocidad
"              en estructura PD+ para coordenadas cartesianas.
" Autor:       José C.Navarrete García
" Fecha:       10/3/2002

```

" Entradas y salidas:

```

INPUT q1 q2 dq1 dq2
OUTPUT tau1 tau2

```

" Tiempo:

TIME t

" Ecuaciones:

```

q1kn=q1
q2kn=q2

```

```

dq1kn=dq1
dq2kn=dq2

```

" Matriz de inercia

```

m11=0.33530+0.02436*cos(q2kn)
m12=0.01267+0.01218*cos(q2kn)
m21=0.01267+0.01218*cos(q2kn)
m22=0.01267

```

" Matriz de fuerzas centrífugas y de Coriolis

```

c11=-0.02436*sin(q2kn)*dq2kn
c12=-0.01218*sin(q2kn)*dq2kn
c21= 0.01218*sin(q2kn)*dq1kn
c22= 0.0

```



```

" Vector de pares gravitacionales

g1=1.17310*g*sin(q1kn)+0.04685*g*sin(q1kn+q2kn)
g2=0.04685*g*sin(q1kn+q2kn)

" La posición en coordenadas de trabajo
" El modelo cinemático

x= 0.26*sin(q1kn)+0.26*sin(q1kn+q2kn)
y=-0.26*cos(q1kn)-0.26*cos(q1kn+q2kn)

" Derivada temporal de x e y

dx=0.26*cos(q1kn)*dq1kn+0.26*cos(q1kn+q2kn)*(dq1kn+dq2kn)
dy=0.26*sin(q1kn)*dq1kn+0.26*sin(q1kn+q2kn)*(dq1kn+dq2kn)

" Campo de Velocidad que dibuja
" el ovalo de Cassini

" Definición de la curva a seguir:

f1=2*a*a*((x-xc)*(x-xc)-(y-yc)*(y-yc))+b*b*b*b
f=(r^2)+a*a*a*a-f1
r=(x-xc)*(x-xc)+(y-yc)*(y-yc)

" El gradiente de f

delfx=4*(x-xc)*r-4*a*a*(x-xc)
delfy=4*(y-yc)*r+4*a*a*(y-yc)

dist=sqrt(delfx*delfx+delfy*delfy)

" La derivada temporal de f

dtf=delfx*dx+delfy*dy

n1=2*(x-xc)*dx+2*(y-yc)*dy
n2=2*(x-xc)*dx-2*(y-yc)*dy

" Derivada temporal del gradiente

dtdelfx=4*(x-xc)*n1+4*dx*r-4*a*a*dx
dtdelfy=4*(y-yc)*n1+4*dy*r+4*a*a*dy

dtdist=(delfx*dtdelfx+delfy*dtdelfy)/dist

" Ganancia k1(q)

k1=ko/(abs(f)*dist+epsilon)

" Derivada temporal de k1(q)

```

```

dendtk1=((abs(f)*dist+epsilon)^2)
dtk1=-ko*(abs(f)*dtdist+tanh(lambda*f)*dtf*dist)/dendtk1

" Ganancia c1(q)

c1=co*exp(-alpha*abs(f))

" Derivada temporal de c1(q)

dtk1=-alpha*co*tanh(lambda*f)*dtf*exp(-alpha*abs(f))

" El campo de velocidad

vx=-k1*f*delfx-c1*delfy/dist
vy=-k1*f*delfy+c1*delfx/dist

" Variables auxiliares para calcular la derivada
temporal del campo de velocidad

ax=-dtk1*f*delfx-dtc1*delfy/(dist)
ay=-dtk1*f*delfy+dtc1*delfx/(dist)

terx=c1*(dist*dtdelfy-dtdist*delfy)/(dist*dist)
tery=c1*(dist*dtdelfx-dtdist*delfx)/(dist*dist)

" Derivada temporal del campo de velocidad

dvtvx=-k1*(dtf*delfx+f*dtdelfx)-terx+ax
dvtvy=-k1*(dtf*delfy+f*dtdelfy)+tery+ay

" El Jacobiano J(q)

j11=0.26*cos(q1kn)+0.26*cos(q1kn+q2kn)
j12=0.26*cos(q1kn+q2kn)
j21=0.26*sin(q1kn)+0.26*sin(q1kn+q2kn)
j22=0.26*sin(q1kn+q2kn)

JJ=j11*j22-j12*j21

dtJJ=j11*dtj22+dtj11*j22-j12*dtj21-j21*dtj12

" Derivada temporal del Jacobiano

pj11dq1=-0.26*sin(q1kn)-0.26*sin(q1kn+q2kn)
pj12dq1=-0.26*sin(q1kn+q2kn)
pj21dq1= 0.26*cos(q1kn)+0.26*cos(q1kn+q2kn)
pj22dq1= 0.26*cos(q1kn+q2kn)

pj11dq2=-0.26*sin(q1kn+q2kn)
pj12dq2=-0.26*sin(q1kn+q2kn)

```

```

pj21dq2= 0.26*cos(q1kn+q2kn)
pj22dq2= 0.26*cos(q1kn+q2kn)

dtj11=pj11dq1*dq1kn+pj11dq2*dq2kn
dtj12=pj12dq1*dq1kn+pj12dq2*dq2kn
dtj21=pj21dq1*dq1kn+pj21dq2*dq2kn
dtj22=pj22dq1*dq1kn+pj22dq2*dq2kn

" Derivada temporal del jacobiano inverso

idtj11=(JJ*dtj11-j11*dtJJ)/(JJ*JJ)
idtj12=(JJ*dtj12-j12*dtJJ)/(JJ*JJ)
idtj21=(JJ*dtj21-j21*dtJJ)/(JJ*JJ)
idtj22=(JJ*dtj22-j22*dtJJ)/(JJ*JJ)

" El controlador

vq1=( j22*vx-j12*vy)/JJ
vq2=(-j21*vx+j11*vy)/JJ

" Error de campo de velocidad

ecvx=vx-dx
ecvy=vy-dy

" Cálculo de la norma de la velocidad

nrmvel=sqrt(dx*dx+dy*dy)

"Cálculo de la norma del error

nrmerr=sqrt(ecvx*ecvx+ecvy*ecvy)

" \xi

e1=vq1-dq1kn
e2=vq2-dq2kn

cntrl1= idtj22*vx-idtj12*vy+( j22*dtvx-j12*dtvy)/JJ
cntrl2=-idtj21*vx+idtj11*vy+(-j21*dtvx+j11*dtvy)/JJ

tau1=m11*cntrl1+m12*cntrl2+c11*vq1+c12*vq2+g1+kk1*e1+b1*dq1kn+fc1*sign(dq1kn)
tau2=m21*cntrl1+m22*cntrl2+c21*vq1+c22*vq2+g2+kk2*e2+b2*dq2kn+fc2*sign(dq2kn)

kk1:5.00
kk2:0.75

" Parámetros del Ovalo de Cassini

a:0.110

```

```

b:0.1125
xc:0.180
yc:-0.180

" Otras constantes del controlador

ko:1.00
co:0.35

epsilon:0.000001 "1E-6
lambda:5000
alpha:500

g:9.81

fc1:1.290
fc2:0.745

b1:0.274
b2:0.144

END

```

B.2.3 Conexiones

CONNECTING SYSTEM CPDMAS

```

" Autor: José C. Navarrete García
" Fecha: 10/3/2002

" Tiempo:

```

TIME t

" Conexiones:

```

q1[pdmas1]=q1[robobuga]
q2[pdmas1]=q2[robobuga]

tau1[robobuga]=tau1[pdmas1]
tau2[robobuga]=tau2[pdmas1]

dq1[pdmas1]=dq1[robobuga]
dq2[pdmas1]=dq2[robobuga]

END

```

B.3 Control de Li y Horowitz

B.3.1 Macro

MACRO RVFC

" Autor: José C. Navarrete García

" Fecha: 10/23/02

syst robobuga vfc cvfc

"store x[vfc] y[vfc]

"store dx[vfc] dy[vfc]

"store xtau1[vfc] xtau2[vfc]

"store ecvx[vfc] ecvy[vfc]

"store nrmvel[vfc] nrmerr[vfc]

" Condiciones iniciales

init xq1[robobuga]:0.785398633

init xq2[robobuga]:1.570796326795

init xq3[robobuga]:0.0

init xq4[robobuga]:0.0

" posicion y velocidad articular iniciales

" del flywheel

init x1[vfc]:0.0 " Condiciones iniciales cero

init x2[vfc]:0.0

newplot

axes v -1 1 h -1 1

plot y[vfc](x[vfc])

END

B.3.2 Controlador

CONTINUOUS SYSTEM VFC

" Descripción: Controlador de Li y Horowitz.

" Autor: José C. Navarrete García

" Fecha: 10/23/02

" Entradas y salidas:

INPUT q1 q2 dq1 dq2

OUTPUT tau1 tau2

" Variables de estado y la variable tiempo:

STATE x1 x2

DER dx1 dx2

TIME t

" Posiciones y velocidades articulares del tobot de Bugarín.

q1kn=q1

q2kn=q2

dq1kn=dq1

dq2kn=dq2

" Inertor/Inerciómetro rotacional

dx1=x2

dx2=(1/MF)*tau3

q3kn=x1

dq3kn=dx1

" Matriz de inercia.

" El modelo de Bugarin

m11=0.33530+0.02436*cos(q2kn)

m12=0.01267+0.01218*cos(q2kn)

m21=0.01267+0.01218*cos(q2kn)

m22=0.01267

" Matriz de fuerzas centrífugas y de Coriolis.

c11=-0.02436*sin(q2kn)*dq2kn

c12=-0.01218*sin(q2kn)*dq2kn

c21= 0.01218*sin(q2kn)*dq1kn

c22= 0.0

" Vector de pares gravitacionales.

g1=1.17310*g*sin(q1kn)+0.04685*g*sin(q1kn+q2kn)

g2=0.04685*g*sin(q1kn+q2kn)

" Cinemática directa.

x= 0.26*sin(q1kn)+0.26*sin(q1kn+q2kn)

y=-0.26*cos(q1kn)-0.26*cos(q1kn+q2kn)

" Derivada temporal de x e y.

```

dx=0.26*cos(q1kn)*dq1kn+0.26*cos(q1kn+q2kn)*(dq1kn+dq2kn)
dy=0.26*sin(q1kn)*dq1kn+0.26*sin(q1kn+q2kn)*(dq1kn+dq2kn)

```

```

" Campo de Velocidad que dibuja
" el ovalo de Cassini

```

```

"Definicion de la curva a seguir:

```

```

f1=2*a*a*((x-xc)*(x-xc)-(y-yc)*(y-yc))+b*b*b*b
f=(r^2)+a*a*a*a-f1
r=(x-xc)*(x-xc)+(y-yc)*(y-yc)

```

```

" El gradiente de f

```

```

delfx=4*(x-xc)*r-4*a*a*(x-xc)
delfy=4*(y-yc)*r+4*a*a*(y-yc)

```

```

dist=sqrt(delfx*delfx+delfy*delfy)

```

```

" La derivada temporal de f

```

```

dtf=delfx*dx+delfy*dy

```

```

n1=2*(x-xc)*dx+2*(y-yc)*dy
n2=2*(x-xc)*dx-2*(y-yc)*dy

```

```

" Derivada temporal del gradiente

```

```

dtdelfx=4*(x-xc)*n1+4*dx*r-4*a*a*dx
dtdelfy=4*(y-yc)*n1+4*dy*r+4*a*a*dy
dtdist=(delfx*dtdelfx+delfy*dtdelfy)/dist

```

```

" Ganancia k1(q)

```

```

k1=ko/(abs(f)*dist+epsilon)

```

```

" Derivada temporal de k1(q)

```

```

dendtk1=((abs(f)*dist+epsilon)^2)
dtk1=-ko*(abs(f)*dtdist+tanh(lambda*f)*dtf*dist)/dendtk1

```

```

" Ganancia c1(q)

```

```

c1=co*exp(-alpha*abs(f))

```

```

" Derivada temporal de c1(q)

```

```

dtc1=-alpha*co*tanh(lambda*f)*dtf*exp(-alpha*abs(f))

```

```

" El campo de velocidad

```

```

    vx=-k1*f*delfx-c1*delfy/dist
    vy=-k1*f*delfy+c1*delfx/dist

    " Variables auxiliares para calcular la derivada
    " temporal del campo de velocidad

    ax=-dtk1*f*delfx-dtc1*delfy/(dist)
    ay=-dtk1*f*delfy+dtc1*delfx/(dist)

    terx=c1*(dist*dtdelfy-dtdist*delfy)/(dist*dist)
    tery=c1*(dist*dtdelfx-dtdist*delfx)/(dist*dist)

    " Derivada temporal del campo de velocidad

    dtvx=-k1*(dtf*delfx+f*dtdelfx)-terx+ax
    dtvy=-k1*(dtf*delfy+f*dtdelfy)+tery+ay

    " Controlador NO Pasivo mediante campo de velocidad (incluyendo friccion viscosa)

    ecvx=dx-omega*vx
    ecvy=dy-omega*vy

    " Cálculo de la norma de la velocidad

    nrmvel=sqrt(dx*dx+dy*dy)

    "Cálculo de la norma del error

    nrmerr=sqrt(ecvx*ecvx+ecvy*ecvy)

    " El Jacobiano J(q)

    j11=0.26*cos(q1kn)+0.26*cos(q1kn+q2kn)
    j12=0.26*cos(q1kn+q2kn)
    j21=0.26*sin(q1kn)+0.26*sin(q1kn+q2kn)
    j22=0.26*sin(q1kn+q2kn)

    JJ=j11*j22-j12*j21

    "La matriz de inercia aumentada

    m13=0
    m23=0
    m31=0
    m32=0
    m33=MF

    " Matriz de fuerzas centrífugas y de Coriolis aumentada

    c13=0

```

```

c23=0
c31=0
c32=0
c33=0

" Cálculo del campo vectorial en función de coordenadas articulares
" a través del Jacobiano inverso.

v1q=( j22*vx-j12*vy)/JJ
v2q=(-j21*vx+j11*vy)/JJ

" Control Pasivo utilizando Campo Vectorial de Velocidad
" derivadas parciales del campo de velocidad aumentado respecto a
" cada posición x1 x2 x3 (x y z):
" pv1dx, pv1dy, pv2dx, pv2dy

ax1=v1q*m11+v2q*m21

ax2=v1q*m21+v2q*m22

VMV=ax1*v1q+ax2*v2q

"El campo vectorial extendido, elemento extra

v3q=sqrt((2/MF)*(E-(0.5)*VMV))

" Todas las parciales de los elementos de M(q)
" respecto a q1 son igual a cero. La pm22dq2=0 también.

pm11dq1=0
pm12dq1=0
pm21dq1=0
pm22dq1=0

pm11dq2=-0.02436*sin(q2kn)
pm12dq2=-0.01218*sin(q2kn)
pm21dq2=-0.01218*sin(q2kn)
pm22dq2=0

dtm11=pm11dq1*dq1kn+pm11dq2*dq2kn
dtm12=pm12dq1*dq1kn+pm12dq2*dq2kn
dtm21=pm21dq1*dq1kn+pm21dq2*dq2kn
dtm22=pm22dq1*dq1kn+pm22dq2*dq2kn

pj11dq1=-0.26*sin(q1kn)-0.26*sin(q1kn+q2kn)
pj12dq1=-0.26*sin(q1kn+q2kn)
pj21dq1= 0.26*cos(q1kn)+0.26*cos(q1kn+q2kn)
pj22dq1= 0.26*cos(q1kn+q2kn)

pj11dq2=-0.26*sin(q1kn+q2kn)
pj12dq2=-0.26*sin(q1kn+q2kn)

```

```

pj21dq2= 0.26*cos(q1kn+q2kn)
pj22dq2= 0.26*cos(q1kn+q2kn)

pJJdq1=j11*pj22dq1+j22*pj11dq1-j12*pj21dq1-j21*pj12dq1
pJJdq2=j11*pj22dq2+j22*pj11dq2-j12*pj21dq2-j21*pj12dq2
dtJJ=pJJdq1*dq1kn+pJJdq2*dq2kn

dtj22=pj22dq1*dq1kn+pj22dq2*dq2kn
dtj12=pj12dq1*dq1kn+pj12dq2*dq2kn
dtj21=pj21dq1*dq1kn+pj21dq2*dq2kn
dtj11=pj11dq1*dq1kn+pj11dq2*dq2kn

dtj22_JJ=(JJ*dtj22-j22*dtJJ)/(JJ*JJ)
dtj12_JJ=(JJ*dtj12-j12*dtJJ)/(JJ*JJ)
dtj21_JJ=(JJ*dtj21-j21*dtJJ)/(JJ*JJ)
dtj11_JJ=(JJ*dtj11-j11*dtJJ)/(JJ*JJ)

dtv1=( j22*dtvx-j12*dtvy)/JJ+dtj22_JJ*vx-dtj12_JJ*vy
dtv2=(-j21*dtvx+j11*dtvy)/JJ-dtj21_JJ*vx+dtj11_JJ*vy

MV_1=m11*v1q+m12*v2q
MV_2=m21*v1q+m22*v2q

dtMV_1=dtm11*v1q+m11*dtv1+dtm12*v2q+m12*dtv2
dtMV_2=dtm21*v1q+m21*dtv1+dtm22*v2q+m22*dtv2

dtVMV=v1q*dtMV_1+v2q*dtMV_2+dtv1*MV_1+dtv2*MV_2

dtv3=-((1/(2*MF))*dtVMV)/(sqrt((2/MF)*(E-(0.5)*VMV)))

" w=M(q)dV(q)+C(q,dq)dV(q) todos los elementso del sistema aumentado:

w11=m11*dtv1+m12*dtv2
w21=m21*dtv1+m22*dtv2
w31=MF*dtv3

w12=c11*v1q+c12*v2q
w22=c21*v1q+c22*v2q
w23=0

w1=w11+w12
w2=w21+w22
w3=MF*dtv3

p1=m11*dq1kn+m12*dq2kn
p2=m21*dq1kn+m22*dq2kn
p3=MF*dq3kn

PP1=m11*v1q+m12*v2q
PP2=m21*v1q+m22*v2q

```


PP3=MF*v3q

" El controlador

```
xx11=w1*PP1-PP1*w1
xx12=w1*PP2-PP1*w2
xx13=w1*PP3-PP1*w3
xx21=w2*PP1-PP2*w1
xx22=w2*PP2-PP2*w2
xx23=w2*PP3-PP2*w3
xx31=w3*PP1-PP3*w1
xx32=w3*PP2-PP3*w2
xx33=w3*PP3-PP3*w3
```

```
yy11=PP1*p1-p1*PP1
yy12=PP1*p2-p1*PP2
yy13=PP1*p3-p1*PP3
yy21=PP2*p1-p2*PP1
yy22=PP2*p2-p2*PP2
yy23=PP2*p3-p2*PP3
yy31=PP3*p1-p3*PP1
yy32=PP3*p2-p3*PP2
yy33=PP3*p3-p3*PP3
```

```
tau_c1=(1/(2*E))*(xx11*dq1kn+xx12*dq2kn+xx13*dq3kn)
tau_c2=(1/(2*E))*(xx21*dq1kn+xx22*dq2kn+xx23*dq3kn)
tau_c3=(1/(2*E))*(xx31*dq1kn+xx32*dq2kn+xx33*dq3kn)
```

```
tau_f1=gama*(yy11*dq1kn+yy12*dq2kn+yy13*dq3kn)
tau_f2=gama*(yy21*dq1kn+yy22*dq2kn+yy23*dq3kn)
tau_f3=gama*(yy31*dq1kn+yy32*dq2kn+yy33*dq3kn)
```

" Delta debe de ser mayor que 2*E*gama

delta=2*E*gama+1

```
tau_k1=delta*PP1*(omega-(PP1*dq1kn+PP2*dq2kn+PP3*dq3kn)/(2*E))
tau_k2=delta*PP2*(omega-(PP1*dq1kn+PP2*dq2kn+PP3*dq3kn)/(2*E))
tau_k3=delta*PP3*(omega-(PP1*dq1kn+PP2*dq2kn+PP3*dq3kn)/(2*E))
```

" Se asigna finalmente la información de control

```
xtau1=tau_c1+tau_f1+tau_k1+g1+b1*dq1kn+fc1*sign(dq1kn)
xtau2=tau_c2+tau_f2+tau_k2+g2+b2*dq2kn+fc2*sign(dq2kn)
xtau3=tau_c3+tau_f3+tau_k3
```

```
tau1=xtau1
tau2=xtau2
tau3=xtau3
```

"Parámetros

```

g:9.81

" Parámetros del Ovalo de Cassini

a:0.110
b:0.1125
xc:0.180
yc:-0.180

ko:1.00
epsilon:0.000001
lambda:5000
alpha:500
co:0.350

"parámetros del controlador

omega:1.0
MF:2.5

E:2.42
gama:7.0

"Parametros de friccion viscosa

fc1:1.290
fc2:0.745

b1:0.274
b2:0.144

END

```

B.3.3 Conexiones

CONNECTING SYSTEM CVFC

```

" Author:      José C. Navarrete García
" Created:     1/12/03

```

```

" Tiempo:

```

TIME t

```

" Conexiones:

```

```

tau1[robobuga]=tau1[vfc]
tau2[robobuga]=tau2[vfc]

```

```
q1[vfc]=q1[robobuga]  
q2[vfc]=q2[robobuga]  
dq1[vfc]=dq1[robobuga]  
dq2[vfc]=dq2[robobuga]
```

```
END
```

Apéndice C

Codigos de WinMechLab

Se utilizó el sistema WinMechLab para la implementación de los algoritmos de control estudiados. WinMechLab es un sistema de propósito general para el control en tiempo real de sistemas mecánicos que opera en Windows NT. WinMechlab permite la edición, compilación, simulación y ejecución de los algoritmos de control¹. La programación de código en WinMechLab se hace utilizando una sintáxis similar al estandar del lenguaje C.

C.1 Control basado en dinámica inversa

```

/*ENCABEZADO*/

/*dinversa_final.rob*/

/*Controlador por dinámica inversa sin accion integral*/
/*para coordenadas de trabajo*/
/*Control mediante campo de velocidad*/

/*Autor: José Clemente Navarrete García*/
/*Fecha:14 Marzo 2003*/

/*Revisiones*/
/*R.Campa: 13 marzo 2003*/
/*Kelly: 17 de marzo de 2003, 18 de marzo de 2003*/
/*Modificacion 09 Abril de 2003*/

/*Edición Final*/
/*Mayo 16 2003*/
/*J.Navarrete*/

/*DECLARACIONES*/
/*DEFINICION DE VARIABLES*/
/*Definición de variables obligatorias*/

float pos1,pos2;
float vel1,vel2;

```

¹WinMechLab es un sistema desarrollado en el laboratorio de Robótica del CICESE.

```

float tau1,tau2;

/*Matriz de inercia*/
/*Matriz de fuerzas centrifugas y de Coriolis*/
/*Vector de pares gravitacionales*/

float m11,m12,m21,m22;
float c11,c12,c21,c22;
float g1,g2;

/*El modelo cinematico directo*/

float x,y;
float dx,dy;

/*Variables utilizadas en la definicion del campo de velocidad*/

float f,f1,r;
float delfx,delfy,dist;
float dtf;

float n1,n2;
float dtdelfx,dtdelfy;
float dtdist;

/*Ganancia k1*/

float k1;
float dendtk1;
float dtk1;

/*Ganancia c1*/

float c1;
float dtc1;

/*El campo de velocidad*/

float vx,vy;
float ax,ay;
float terx,tery;
float dtvx,dtvy;

/*El Jacobiano J(q)*/

float j11,j12,j21,j22;
float JJ;

/*Derivada temporal del Jacobiano*/

float pj11dq1;

```



```

float pj12dq1;
float pj21dq1;
float pj22dq1;

float pj11dq2;
float pj12dq2;
float pj21dq2;
float pj22dq2;

float dtj11;
float dtj12;
float dtj21;
float dtj22;

/*Error de campo de velocidad*/

float ecvx;
float ecvy;

/*Variables utilizadas en la programación del algoritmo de control*/

float dl1;
float dl2;
float uo1;
float uo2;
float aa1;
float aa2;

/*Función Signo*/

float sign1, sign2;

/*Parte del controlador de posición tanh-d*/

float error1,error2;

/*Indice de desempeño*/

float norma;
float indice;

/*DEFINICION DE CONSTANTES*/
/*Parametros del ovalo de Cassini*/

float a=0.110;
float b=0.1125;
float xc=0.180;
float yc=-0.180;

/*Referentes al campo de velocidad*/

```

```

float ko=1.0; /*Velocidad de aproximacion */
float co=0.35; /*Velocidad tangencial */

/*Tambien del campo de velocidad*/

float epsilon=0.000001; /* epsilon=1E-6 */
float lambda=5000.0;
float alpha=500.0;

/*Ganancia del error de campo de velocidad*/

float kk1=10.0;
float kk2=10.0;

/*Gravedad*/

float g=9.81;

/*Parámetros del controlador de posición tanh-d*/

float kp1=8.0;
float kp2=4.0;
float kv1=0.16;
float kv2=0.01;

float lambda1=2.5;
float lambda2=1.8;

/*Posiciones articulares iniciales*/

float qd1=45.0;
float qd2=90.0;

/*Parámetros de fricción viscosa y de Coulomb*/

float b1=0.274, b2=0.144;
float fc1=1.290, fc2=0.745;

/*Factor de conversión de grados a radianes*/

float c_g_r = 0.01745329;

/*Tiempo de ejecución*/

float T_norma=10; //segundos

/*CUERPO DEL PROGRAMA*/
/* Matriz de inercia */

m11=0.33530+0.02436*cos(c_g_r*pos2);
m12=0.01267+0.01218*cos(c_g_r*pos2);

```

```

m21=0.01267+0.01218*cos(c_g_r*pos2);
m22=0.01267;

/*Matriz de fuerzas centrifugas y de Coriolis*/

c11=-0.02436*sin(c_g_r*pos2)*c_g_r*vel2;
c12=-0.01218*sin(c_g_r*pos2)*c_g_r*vel2;
c21= 0.01218*sin(c_g_r*pos2)*c_g_r*vel1;
c22= 0.0;

/*Vector de pares gravitacionales*/

g1=1.17310*g*sin(c_g_r*pos1)+0.04685*g*sin(c_g_r*(pos1+pos2));
g2=0.04685*g*sin(c_g_r*(pos1+pos2));

/*El modelo cinemático*/

x= 0.26*sin(c_g_r*pos1)+0.26*sin(c_g_r*(pos1+pos2));
y=-0.26*cos(c_g_r*pos1)-0.26*cos(c_g_r*(pos1+pos2));

/*Derivada temporal del modelo cinemático*/

dx=0.26*cos(c_g_r*pos1)*c_g_r*vel1+0.26*cos(c_g_r*(pos1+pos2))*c_g_r*(vel1+vel2);
dy=0.26*sin(c_g_r*pos1)*c_g_r*vel1+0.26*sin(c_g_r*(pos1+pos2))*c_g_r*(vel1+vel2);

/*Campo de Velocidad que converge a*/
/*un ovalo de Cassini*/

/*Definicion de la curva a seguir:*/

f1=2*a*a*((x-xc)*(x-xc)-(y-yc)*(y-yc))+b*b*b*b;
f=(r*r)+a*a*a*a-f1;
r=(x-xc)*(x-xc)+(y-yc)*(y-yc);

/*El gradiente de f*/

delfx=4*(x-xc)*r-4*a*a*(x-xc);
delfy=4*(y-yc)*r+4*a*a*(y-yc);
dist=sqrt(delfx*delfx+delfy*delfy);

/*La derivada temporal de f*/

dtf=delfx*dx+delfy*dy;
n1=2*(x-xc)*dx+2*(y-yc)*dy;
n2=2*(x-xc)*dx-2*(y-yc)*dy;

/*Derivada temporal del gradiente*/

dtdelfx=4*(x-xc)*n1+4*dx*r-4*a*a*dx;
dtdelfy=4*(y-yc)*n1+4*dy*r+4*a*a*dy;
dtdist=(delfx*dtdelfx+delfy*dtdelfy)/dist;

```

```

/*Ganancia k1(q)*/

k1=ko/(fabs(f)*dist+epsilon);

/*Derivada temporal de k1(q)*/

dendtk1=(fabs(f)*dist+epsilon)*(fabs(f)*dist+epsilon);
dtk1=-ko*(fabs(f)*dtdist+tanh(lambda*f)*dtf*dist)/dendtk1;

/*Ganancia c1(q)*/

c1=co*exp(-alpha*fabs(f));

/*Derivada temporal de c1(q)*/

detc1=-alpha*co*tanh(lambda*f)*dtf*exp(-alpha*fabs(f));

/*El campo de velocidad*/

vx=-k1*f*delfx-c1*delfy/dist;
vy=-k1*f*delfy+c1*delfx/dist;

/*Variables auxiliares para calcular la derivada*/
/*temporal del campo de velocidad*/

ax=-dtk1*f*delfx-dtc1*delfy/(dist);
ay=-dtk1*f*delfy+dtc1*delfx/(dist);

terx=c1*(dist*dtdelfy-dtdist*delfy)/(dist*dist);
tery=c1*(dist*dtdelfx-dtdist*delfx)/(dist*dist);

/*Derivada temporal del campo de velocidad*/

dtvx=-k1*(dtf*delfx+f*dtdelfx)-terx+ax;
dtvy=-k1*(dtf*delfy+f*dtdelfy)+tery+ay;

/*Si la variable Tiempo_real es menor que 5 segundos*/
/*entra en operacion un controlador que lleva al robot*/
/*a la posición inicial deseada \q(0) */

if(Tiempo_real<5.0)
{
/*Error en coordenadas articulares*/

error1=qd1-pos1;
error2=qd2-pos2;

/*Ley de control Tanh-d con compensacion de gravedad*/

tau1=kp1*tanh(lambda1*error1)-kv1*vel1+g1;

```

```

    tau2=kp2*tanh(lambda2*error2)-kv2*vel2+g2;

    norma=0;

    ecvx=0;
    ecvy=0;

}
else
{

/*Error de campo de velocidad*/

    ecvx=vx-dx;
    ecvy=vy-dy;

/*El Jacobiano J(q)*/

    j11=0.26*cos(c_g_r*pos1)+0.26*cos(c_g_r*(pos1+pos2));
    j12=0.26*cos(c_g_r*(pos1+pos2));
    j21=0.26*sin(c_g_r*pos1)+0.26*sin(c_g_r*(pos1+pos2));
    j22=0.26*sin(c_g_r*(pos1+pos2));

    JJ=j11*j22-j12*j21;

/*Derivada temporal del Jacobiano*/

    pj11dq1=-0.26*sin(c_g_r*pos1)-0.26*sin(c_g_r*(pos1+pos2));
    pj12dq1=-0.26*sin(c_g_r*(pos1+pos2));
    pj21dq1= 0.26*cos(c_g_r*pos1)+0.26*cos(c_g_r*(pos1+pos2));
    pj22dq1= 0.26*cos(c_g_r*(pos1+pos2));

    pj11dq2=-0.26*sin(c_g_r*(pos1+pos2));
    pj12dq2=-0.26*sin(c_g_r*(pos1+pos2));
    pj21dq2= 0.26*cos(c_g_r*(pos1+pos2));
    pj22dq2= 0.26*cos(c_g_r*(pos1+pos2));

    dtj11=c_g_r*(pj11dq1*vel1+pj11dq2*vel2);
    dtj12=c_g_r*(pj12dq1*vel1+pj12dq2*vel2);
    dtj21=c_g_r*(pj21dq1*vel1+pj21dq2*vel2);
    dtj22=c_g_r*(pj22dq1*vel1+pj22dq2*vel2);

// Aqui se define la parte del controlador que tiene la forma
//  $dl=dt[v(x)]+kk*e-d[J(q)]/dt*qp$ 

    dl1=dtvx+(kk1*ecvx)-c_g_r*(dtj11*vel1+dtj12*vel2);
    dl2=dtvy+(kk2*ecvy)-c_g_r*(dtj21*vel1+dtj22*vel2);

// Definición del vector uo donde
//  $uo=[J(q)]^{-1} * dl$ 

```



```

    uo1=( j22*d11-j12*d12)/JJ;
    uo2=(-j21*d11+j11*d12)/JJ;

    /*Definición de la función signo*/

    if (vel1>0)
        sign1=1;
    else
        sign1=-1;

    if (vel2>0)
        sign2=1;
    else
        sign2=-1;

    /*Calculo de acciones de control*/

    tau1=m11*uo1+m12*uo2+c_g_r*(c11*vel1+c12*vel2)+g1+b1*c_g_r*vel1+fc1*sign1;

    if(tau1>15) tau1=15;
    if(tau1<-15) tau1=-15;

    tau2=m21*uo1+m22*uo2+c_g_r*(c21*vel1+c22*vel2)+g2+b2*c_g_r*vel2+fc2*sign2;

    if(tau2>4) tau2=4;
    if(tau2<-4) tau2=-4;

    /*Acumulador de la norma*/

    if(Tiempo_real>=5.0 && Tiempo_real<=5+T_norma){
        norma=norma+ecvx*ecvx+ecvy*ecvy;
    }
    indice=sqrt(norma/T_norma);

}

/* FIN DEL PROGRAMA */

```

C.2 Control en estructura PD+

```

/*ENCABEZADO*/

/*pdmas_final.rob*/

/*Controlador en estructura PD+ para coordenadas cartesianas*/
/*Control mediante campo de velocidad*/

/*Autor: José Clemente Navarrete García*/
/*Fecha: 01 Abril 2003*/

```

```

/*Edición Final*/
/*Mayo 16 2003*/
/*J.Navarrete*/

/* Corrección en el calculo de la derivada temporal del jacobiano inverso*/
/* Junio 27 2003 */
/* J.Navarrete */

/*DECLARACIONES*/
/*DEFINICION DE VARIABLES*/
/*Definición de variables obligatorias*/

float pos1,pos2;
float vel1,vel2;
float tau1,tau2;

/*Matriz de inercia*/
/*Matriz de fuerzas centrifugas y de Coriolis*/
/*Vector de pares gravitacionales*/

float m11,m12,m21,m22;
float c11,c12,c21,c22;
float g1,g2;

/*El modelo cinematico directo*/

float x,y;
float dx,dy;

/*Variables utilizadas en la definicion del campo de velocidad*/

float f,f1,r;
float delfx,delfy,dist;
float dtf;

float n1,n2;
float dtdelfx,dtdelfy;
float dtdist;

/*Ganancia k1*/

float k1;
float dendtk1;
float dtk1;

/*Ganancia c1*/

float c1;
float dtc1;

/*El campo de velocidad*/

```

```

float vx,vy;
float ax,ay;
float terx,tery;
float dtvx,dtvy;

/*Elementos referentes al Jacobiano

float j11,j12,j21,j22;
float JJ;
float dtJJ;

/*Derivada temporal del Jacobiano*/

float pj11dq1;
float pj12dq1;
float pj21dq1;
float pj22dq1;

float pj11dq2;
float pj12dq2;
float pj21dq2;
float pj22dq2;

float dtj11;
float dtj12;
float dtj21;
float dtj22;

/*Derivada temporal del Jacobiano inverso*/

float idtj11;
float idtj12;
float idtj21;
float idtj22;

/*Parametros del controlador PD mas*/

float vq1,vq2;
float ecvx,ecvy;
float ji1,ji2;
float cntrl1,cntrl2;

/*Parte del controlador tanh-d*/

float error1;
float error2;

/*Estados del modelo de friccion de Dahl*/

float z1k,z2k;

```

```
float z1kn,z2kn;

/*Función signo*/

float sign1,sign2;

/*Indice de desempeño*/

float indice;
float norma;

/*DEFINICION DE CONSTANTES*/

/*Parametros del ovalo de Cassini*/

float a=0.110;
float b=0.1125;
float xc=0.180;
float yc=-0.180;

/*Referentes al campo de velocidad*/

float ko=1.0;
float co=0.35;

/*Tambien del campo de velocidad*/

float epsilon=0.000001;
float lambda=5000.0;
float alpha=500.0;

/*Ganancias de control*/

float kk1=5.00;
float kk2=0.75;

/*Gravedad*/

float g=9.81;

/*Ganancias controlador tanh-d*/

float kp1=8.0;
float kp2=4.0;
float kv1=0.16;
float kv2=0.01;

float lambda1=2.5;
float lambda2=1.8;

/*Posiciones articulares iniciales*/
```

```

float qd1=45.0;
float qd2=90.0;

/*Factor de conversión de grados a radianes*/

float c_g_r=0.01745329;

/*Constantes de fricción viscosa y de Coulomb*/

float b1=0.274, b2=0.144;
float fc1=1.290, fc2=0.745;

/*Parametros del modelo de fricción de Dahl*/

float sigma_1=1757.0;
float sigma_2=1450.0;

/*Periodo de integración*/

float h=0.0010; // 1 mseg

/*Tiempo de ejecución*/

float T_norma=10; //segundos

/*CUERPO DEL PROGRAMA*/
/* Matriz de inercia */

m11=0.33530+0.02436*cos(c_g_r*pos2);
m12=0.01267+0.01218*cos(c_g_r*pos2);
m21=0.01267+0.01218*cos(c_g_r*pos2);
m22=0.01267;

/*Matriz de fuerzas centrífugas y de Coriolis*/

c11=-0.02436*sin(c_g_r*pos2)*c_g_r*vel2;
c12=-0.01218*sin(c_g_r*pos2)*c_g_r*vel2;
c21= 0.01218*sin(c_g_r*pos2)*c_g_r*vel1;
c22= 0.0;

/*Vector de pares gravitacionales*/

g1=1.17310*g*sin(c_g_r*pos1)+0.04685*g*sin(c_g_r*(pos1+pos2));
g2=0.04685*g*sin(c_g_r*(pos1+pos2));

/*El modelo cinemático*/

x= 0.26*sin(c_g_r*pos1)+0.26*sin(c_g_r*(pos1+pos2));
y=-0.26*cos(c_g_r*pos1)-0.26*cos(c_g_r*(pos1+pos2));

```



```

/*Derivada temporal del modelo cinemático*/

dx=0.26*cos(c_g_r*pos1)*c_g_r*vel1+0.26*cos(c_g_r*(pos1+pos2))*c_g_r*(vel1+vel2);
dy=0.26*sin(c_g_r*pos1)*c_g_r*vel1+0.26*sin(c_g_r*(pos1+pos2))*c_g_r*(vel1+vel2);

/*Campo de Velocidad que converge*/
/*a un ovalo de Cassini*/

/*Definicion de la curva a seguir*/

f1=2*a*a*((x-xc)*(x-xc)-(y-yc)*(y-yc))+b*b*b*b;
f=(r*r)+a*a*a*a-f1;
r=(x-xc)*(x-xc)+(y-yc)*(y-yc);

/*El gradiente de f*/

delfx=4*(x-xc)*r-4*a*a*(x-xc);
delfy=4*(y-yc)*r+4*a*a*(y-yc);

dist=sqrt(delfx*delfx+delfy*delfy);

/*La derivada temporal de f*/

dtf=delfx*dx+delfy*dy;

n1=2*(x-xc)*dx+2*(y-yc)*dy;
n2=2*(x-xc)*dx-2*(y-yc)*dy;

/*Derivada temporal del gradiente*/

dtdelfx=4*(x-xc)*n1+4*dx*r-4*a*a*dx;
dtdelfy=4*(y-yc)*n1+4*dy*r+4*a*a*dy;

dtdist=(delfx*dtdelfx+delfy*dtdelfy)/dist;

/*Ganancia k1(q)*/

k1=ko/(fabs(f)*dist+epsilon);

/*Derivada temporal de k1(q)*/

dendtk1=((fabs(f)*dist+epsilon)*(fabs(f)*dist+epsilon));
dtk1=-ko*(fabs(f)*dtdist+tanh(lambda*f)*dtf*dist)/dendtk1;

/*Ganancia c1(q)*/

c1=co*exp(-alpha*fabs(f));

/*Derivada temporal de c1(q)*/

dtk1=-alpha*co*tanh(lambda*f)*dtf*exp(-alpha*fabs(f));

```

```

/* El campo de velocidad*/

vx=-k1*f*delfx-c1*delfy/dist;
vy=-k1*f*delfy+c1*delfx/dist;

/*Variables auxiliares para calcular la derivada*/
/*temporal del campo de velocidad*/

ax=-dtk1*f*delfx-dtc1*delfy/(dist);
ay=-dtk1*f*delfy+dtc1*delfx/(dist);

terx=c1*(dist*dtdelfy-dtdist*delfy)/(dist*dist);
tery=c1*(dist*dtdelfx-dtdist*delfx)/(dist*dist);

/*Derivada temporal del campo de velocidad*/

dtvx=-k1*(dtf*delfx+f*dtdelfx)-terx+ax;
dtvy=-k1*(dtf*delfy+f*dtdelfy)+tery+ay;

/*Si el tiempo real es menor que 5 segundos entra*/
/*en operacion un controlador de posición que lleva*/
/*al robot a las condición inicial q(0)*/

if(Tiempo_real<5.0)
{
/*Inicialización de los estados*/

z1k=0;
z2k=0;

z1kn=0;
z2kn=0;

/*Error en coordenadas articulares */

error1=qd1-pos1;
error2=qd2-pos2;

/*Ley de control Tanh-d con compensacion de gravedad*/

tau1=kp1*tanh(lambda1*error1)-kv1*vel1+g1;
tau2=kp2*tanh(lambda2*error2)-kv2*vel2+g2;

ecvx=0;
ecvy=0;
norma=0;

}
else
{

```

```

/*El Jacobiano J(q)*/

j11=0.26*cos(c_g_r*pos1)+0.26*cos(c_g_r*(pos1+pos2));
j12=0.26*cos(c_g_r*(pos1+pos2));
j21=0.26*sin(c_g_r*pos1)+0.26*sin(c_g_r*(pos1+pos2));
j22=0.26*sin(c_g_r*(pos1+pos2));

JJ=j11*j22-j12*j21;
dtJJ=j11*dtj22+dtj11*j22-j12*dtj21-j21*dtj12;

/*Derivada temporal del Jacobiano*/

pj11dq1=-0.26*sin(c_g_r*pos1)-0.26*sin(c_g_r*(pos1+pos2));
pj12dq1=-0.26*sin(c_g_r*(pos1+pos2));
pj21dq1= 0.26*cos(c_g_r*pos1)+0.26*cos(c_g_r*(pos1+pos2));
pj22dq1= 0.26*cos(c_g_r*(pos1+pos2));

pj11dq2=-0.26*sin(c_g_r*(pos1+pos2));
pj12dq2=-0.26*sin(c_g_r*(pos1+pos2));
pj21dq2= 0.26*cos(c_g_r*(pos1+pos2));
pj22dq2= 0.26*cos(c_g_r*(pos1+pos2));

dtj11=c_g_r*(pj11dq1*vel1+pj11dq2*vel2);
dtj12=c_g_r*(pj12dq1*vel1+pj12dq2*vel2);
dtj21=c_g_r*(pj21dq1*vel1+pj21dq2*vel2);
dtj22=c_g_r*(pj22dq1*vel1+pj22dq2*vel2);

/*Derivada temporal del jacobiano inverso*/

idtj11=(JJ*dtj11-j11*dtJJ)/(JJ*JJ);
idtj12=(JJ*dtj12-j12*dtJJ)/(JJ*JJ);
idtj21=(JJ*dtj21-j21*dtJJ)/(JJ*JJ);
idtj22=(JJ*dtj22-j22*dtJJ)/(JJ*JJ);

/*Controlador PD+*/

vq1=( j22*vx-j12*vy)/JJ;
vq2=(-j21*vx+j11*vy)/JJ;

ecvx=vx-dx;
ecvy=vy-dy;

ji1=vq1-c_g_r*vel1;
ji2=vq2-c_g_r*vel2;

cntrl1= idtj22*vx-idtj12*vy+( j22*dtvx-j12*dtvy)/JJ;
cntrl2=-idtj21*vx+idtj11*vy+(-j21*dtvx+j11*dtvy)/JJ;

/*Modelo de friccion de Dahl*/

```

```

z1kn=(z1k+vq1*h)/(1+sigma_1*h*fabs(vel1*c_g_r)/fc1);
z2kn=(z2k+vq2*h)/(1+sigma_2*h*fabs(vel2*c_g_r)/fc2);

/*Pares de control*/

tau1=m11*cntrl1+m12*cntrl2+c11*vq1+c12*vq2+g1+kk1*ji1+b1*c_g_r*vel1+sigma_1*z1k;

if(tau1>15) tau1=15;
if(tau1<-15) tau1=-15;

tau2=m21*cntrl1+m22*cntrl2+c21*vq1+c22*vq2+g2+kk2*ji2+b2*c_g_r*vel2+sigma_2*z2k;

if(tau2>4) tau2=4;
if(tau2<-4) tau2=-4;

/*Actualización de los estados del modelo de Dahl*/

z1k=z1kn;
z2k=z2kn;

/*Acumulador de la norma*/

if(Tiempo_real>=5.0 && Tiempo_real<=5+T_norma){
    norma=norma+ecvx*ecvx+ecvy*ecvy;
}
indice=sqrt(norma/T_norma);
}

/* FIN DEL PROGRAMA */

```

C.3 Control de Li y Horowitz

```

/*ENCABEZADO*/

/*vfc_final.rob*/

/*Control mediante campo de velocidad*/
/*Controlador propuesto por Li y Horowitz*/

/*Autor:José C.Navarrete García*/
/*Fecha: 01 Abril 2003 */

/*Revisiones:*/
/* Modificado: Mayo/01/2003 */

/* Modificado para experimento */
/* Kelly Mayo 02 2003 */

/* José C. Navarrete García */

```

```

/* Revisión Mayo 04,2003 */

/* Revision */
/* J.Navarrete Mayo 12 2003 */

/* Revisión*/
/* Mayo 19 2003 */
/* J.Navarrete / Javier Moreno */

/* Edición Final*/
/* Mayo 23 2003 */
/* J.Navarrete */

//DEFINICION DE VARIABLES
//Definición de variables obligatorias

float pos1,pos2; //posiciones articulares
float vel1,vel2; //velocidades articulares
float tau1,tau2; //pares de control
float tau3; //par ficticio

/*Posiciones articulares en radianes*/

float q1p;
float q2p;

/*Norma de velocidad*/

float norma1;
float norma2;
float norma_e;

//Definicion de variables

//Matriz de inercia
//Matriz de fuerzas centrifugas y de Coriolis
//Vector de pares gravitacionales

float m11,m12,m21,m22;
float c11,c12,c21,c22;
float g1,g2;

//El modelo cinematico directo

float x,y;
float dx,dy;

//Variables utilizadas en la definicion del campo de velocidad

float f,f1,r;
float delfx,delfy,dist;

```



```

//La derivada temporal de f

float dtf;
float n1,n2;

//Derivada temporal del gradiente

float dtdelfx;
float dtdelfy;
float dtdist;

//Ganancia k1 y derivada temporal

float k1,dendtk1,dtk1;

//Ganancia c1 y derivada temporal

float c1,dtc1;

//El campo de velocidad

float vx,vy;

//Referentes a la derivada temporal del campo de velocidad

float ax,ay;
float terx,tery;
float dtvx,dtvy;

//El Jacobiano J(q)

float j11,j12,j21,j22;
float JJ;

//Cálculo del campo vectorial en función de coordenadas articulares
//a través del Jacobiano inverso

float v1q;
float v2q;

//El campo vectorial extendido, elemento extra

float ax1,ax2;
float VMV;
float v3q;

//Parciales respecto a las posiciones articulares

float pm11dq1;
float pm12dq1;

```

```

float pm21dq1;
float pm22dq1;

float pm11dq2;
float pm12dq2;
float pm21dq2;
float pm22dq2;

//Derivada temporal de los elementos de la matriz de inercia

float dtm11;
float dtm12;
float dtm21;
float dtm22;

//Parciales de los elementos del jacobiano

float pj11dq1;
float pj12dq1;
float pj21dq1;
float pj22dq1;

float pj11dq2;
float pj12dq2;
float pj21dq2;
float pj22dq2;

float pJJdq1;
float pJJdq2;
float dtJJ;

//Las derivadas temporales de los elementos del jacobiano

float dtj22;
float dtj12;
float dtj21;
float dtj11;

//Las derivadas temporales de los elementos del jacobiano inverso

float dtj22_JJ;
float dtj12_JJ;
float dtj21_JJ;
float dtj11_JJ;

//Referentes a la derivada dtv3

float MV_1;
float MV_2;
float dtMV_1;
float dtMV_2;

```

```

float dtVMV;

//Derivadas temporales del campo de velocidad

float dtv1,dtv2,dtv3;

float w11,w21,w31;
float w12,w22,w23;

float w1,w2,w3;
float p1,p2,p3;

float PP1,PP2,PP3;

//El controlador

float xx11,xx12,xx13;
float xx21,xx22,xx23;
float xx31,xx32,xx33;

float yy11,yy12,yy13;
float yy21,yy22,yy23;
float yy31,yy32,yy33;

/*Pares de control*/

float tau_c1,tau_c2,tau_c3;
float tau_f1,tau_f2,tau_f3;
float tau_k1,tau_k2,tau_k3;

/*Error de campo de velocidad*/

float ecvx,ecvy;

/*Función signo*/

float sign1, sign2;

/*Errores para el controlador de posición*/

float error1,error2;

/*Ganancia de control */

float delta;

/*Flywheel */

float vel3;
float x1k,x2k;
float x1kn,x2kn;

```

```
/*Para el índice de desempeño*/  
  
float norma, indice;  
  
/*DEFINICION DE CONSTANTES*/  
  
//Parametros del controlador  
  
float m13=0;  
float m23=0;  
float m31=0;  
float m32=0;  
  
//Matriz de fuerzas centrifugas y de Coriolis aumentada  
  
float c13=0;  
float c23=0;  
float c31=0;  
float c32=0;  
float c33=0;  
  
/*Constantes del modelo de friccion */  
  
float b1=0.274, b2=0.144;  
float fc1=1.290, fc2=0.745;  
  
/*Parametros del ovalo de Cassini*/  
  
float a=0.110;  
float b=0.1125;  
float xc=0.180;  
float yc=-0.180;  
  
/*Referentes al campo de velocidad*/  
  
float ko=1.0;  
float co=0.350;  
  
/*Tambien del campo de velocidad*/  
  
float epsilon=0.000001; // 1E-6  
float alpha=500.0;  
float lambda=5000.0;  
  
/*Gravedad*/  
  
float g=9.81;  
  
/*Parametros */
```

```

float E=2.42;
float gama=7.0;
float m33=2.5;
float omega=1.0;

/*Ganacias del controlador de posición*/

float kp1=8.0;
float kp2=4.0;
float kv1=0.16;
float kv2=0.01;

float lambda1=2.5;
float lambda2=1.8;

/*Posiciones articulares iniciales*/

float qd1=45.0;
float qd2=90.0;

/*Factor de conversión de grados a radianes*/

float c_g_r=0.01745329;

/*Periodo de integración*/

float h=0.0010; // 1 mseg.

/* Tiempo de integracion de la norma */
float T_norma=10.0;

/*CUERPO DEL PROGRAMA*/

//PARTE I: Matriz de inercia, de fuerzas centrífugas y de Colriolis
//y vector de pares gravitacionales.

/*Matriz de inercia*/

m11=0.33530+0.02436*cos(c_g_r*pos2);
m12=0.01267+0.01218*cos(c_g_r*pos2);
m21=0.01267+0.01218*cos(c_g_r*pos2);
m22=0.01267;

/*Matriz de fuerzas centrífugas y de Coriolis*/

c11=-0.02436*sin(c_g_r*pos2)*c_g_r*vel2;
c12=-0.01218*sin(c_g_r*pos2)*c_g_r*vel2;
c21= 0.01218*sin(c_g_r*pos2)*c_g_r*vel1;
c22= 0.0;

/*Vector de pares gravitacionales*/

```



```

g1=1.17310*g*sin(c_g_r*pos1)+0.04685*g*sin(c_g_r*(pos1+pos2));
g2=0.04685*g*sin(c_g_r*(pos1+pos2));

//PARTE II: Cinemática directa

/*La posición en coordenadas de trabajo*/
/*El modelo cinemático*/

x= 0.26*sin(c_g_r*pos1)+0.26*sin(c_g_r*(pos1+pos2));
y=-0.26*cos(c_g_r*pos1)-0.26*cos(c_g_r*(pos1+pos2));

/*Derivada temporal del modelo cinemático*/

dx=0.26*cos(c_g_r*pos1)*c_g_r*vel1+0.26*cos(c_g_r*(pos1+pos2))*c_g_r*(vel1+vel2);
dy=0.26*sin(c_g_r*pos1)*c_g_r*vel1+0.26*sin(c_g_r*(pos1+pos2))*c_g_r*(vel1+vel2);

/* Calculo de la norma de la velocidad */
/* y de la magnitud de la norma */

normal=sqrt(dx*dx+dy*dy);
norma2=sqrt(c_g_r*vel1*c_g_r*vel1+c_g_r*vel2*c_g_r*vel2);
norma_e=sqrt(ecvx*ecvx+ecvy*ecvy);

/* Calculo del error de campo de velocidad*/

ecvx=vx-omega*dx;
ecvy=vy-omega*dy;

//PARTE III
//DEFINICION DEL CAMPO DE VELOCIDAD

/*Campo de Velocidad que dibuja*/
/*el ovalo de Cassini*/

/*Definicion de la curva a seguir:*/

f1=2*a*a*((x-xc)*(x-xc)-(y-yc)*(y-yc))+b*b*b*b;
f=(r*r)+a*a*a*a-f1;
r=(x-xc)*(x-xc)+(y-yc)*(y-yc);

/*El gradiente de f*/

delfx=4*(x-xc)*r-4*a*a*(x-xc);
delfy=4*(y-yc)*r+4*a*a*(y-yc);
dist=sqrt(delfx*delfx+delfy*delfy);

//La derivada temporal de f

dtf=delfx*dx+delfy*dy;
n1=2*(x-xc)*dx+2*(y-yc)*dy;

```

```

n2=2*(x-xc)*dx-2*(y-yc)*dy;

//Derivada temporal del gradiente

dtdelfx=4*(x-xc)*n1+4*dx*r-4*a*a*dx;
dtdelfy=4*(y-yc)*n1+4*dy*r+4*a*a*dy;
dtdist=(delfx*dtdelfx+delfy*dtdelfy)/dist;

/*Ganancia k1(q)*/

k1=ko/(fabs(f)*dist+epsilon);

//Derivada temporal de k1(q)

dendtk1=((fabs(f)*dist+epsilon)*(fabs(f)*dist+epsilon));
dtk1=-ko*(fabs(f)*dtdist+tanh(lambda*f)*dtf*dist)/dendtk1;

/*Ganancia c1(q)*/

c1=co*exp(-alpha*fabs(f));

//Derivada temporal de c1(q)

dtc1=-alpha*co*tanh(lambda*f)*dtf*exp(-alpha*fabs(f));

/*El campo de velocidad*/

vx=-k1*f*delfx-c1*delfy/dist;
vy=-k1*f*delfy+c1*delfx/dist;

//Variables auxiliares para calcular la derivada
//temporal del campo de velocidad

ax=-dtk1*f*delfx-dtc1*delfy/(dist);
ay=-dtk1*f*delfy+dtc1*delfx/(dist);

terx=c1*(dist*dtdelfy-dtdist*delfy)/(dist*dist);
tery=c1*(dist*dtdelfx-dtdist*delfx)/(dist*dist);

//Derivada temporal del campo de velocidad

dtvx=-k1*(dtf*delfx+f*dtdelfx)-terx+ax;
dtvy=-k1*(dtf*delfy+f*dtdelfy)+tery+ay;

/*Condicion de delta*/

delta=2*E*gama+1; /* Delta debe de ser mayor que 2*E*gama */

/*Si el tiempo real es menor que 5 segundos entra en operacion un*/
/*controlador que lleva al robot a las condiciones iniciales*/
/*deseadas q1(0) y q2(0)*/

```

```

if(Tiempo_real<5.0)
{
    x1k=0;
    x2k=0; // vel3

    /*Error en coordenadas articulares */

    error1=qd1-pos1;
    error2=qd2-pos2;

    /*Ley de control Tanh-d con compensacion de gravedad*/

    tau1=kp1*tanh(lambda1*error1)-kv1*vel1+g1;
    tau2=kp2*tanh(lambda2*error2)-kv2*vel2+g2;

    norma=0;
}
else
{
    q1p=c_g_r*vel1;
    q2p=c_g_r*vel2;

    //Flywheel

    x1kn=x1k+h*x2kn;
    x2kn=x2k+h*(1/m33)*tau3;
    vel3=x2k;

    /*El Jacobiano J(q)*/

    j11=0.26*cos(c_g_r*pos1)+0.26*cos(c_g_r*(pos1+pos2));
    j12=0.26*cos(c_g_r*(pos1+pos2));
    j21=0.26*sin(c_g_r*pos1)+0.26*sin(c_g_r*(pos1+pos2));
    j22=0.26*sin(c_g_r*(pos1+pos2));

    JJ=j11*j22-j12*j21;

    /*Cálculo del campo vectorial en función de coordenadas articulares*/
    /*a través del Jacobiano inverso.*/

    v1q=( j22*vx-j12*vy)/JJ;
    v2q=(-j21*vx+j11*vy)/JJ;

    /*Para el elemento extra del campo de velocidad*/

    ax1=v1q*m11+v2q*m21;
    ax2=v1q*m21+v2q*m22;
    VMV=ax1*v1q+ax2*v2q;

```

```

//El campo vectorial extendido, elemento extra

v3q=sqrt((2/m33)*(E-(0.5)*VMV));

//Todas las parciales de los elementos de M(q)
//respecto a q1 son igual a cero. La pm22dq2=0 tambien.

pm11dq1=0;
pm12dq1=0;
pm21dq1=0;
pm22dq1=0;

pm11dq2=-0.02436*sin((c_g_r*pos2));
pm12dq2=-0.01218*sin((c_g_r*pos2));
pm21dq2=-0.01218*sin((c_g_r*pos2));
pm22dq2=0;

dtm11=pm11dq1*c_g_r*vel1+pm11dq2*c_g_r*vel2;
dtm12=pm12dq1*c_g_r*vel1+pm12dq2*c_g_r*vel2;
dtm21=pm21dq1*c_g_r*vel1+pm21dq2*c_g_r*vel2;
dtm22=pm22dq1*c_g_r*vel1+pm22dq2*c_g_r*vel2;

pj11dq1=-0.26*sin(c_g_r*(pos1))-0.26*sin(c_g_r*(pos1+pos2));
pj12dq1=-0.26*sin(c_g_r*(pos1+pos2));
pj21dq1= 0.26*cos(c_g_r*(pos1))+0.26*cos(c_g_r*(pos1+pos2));
pj22dq1= 0.26*cos(c_g_r*(pos1+pos2));

pj11dq2=-0.26*sin(c_g_r*(pos1+pos2));
pj12dq2=-0.26*sin(c_g_r*(pos1+pos2));
pj21dq2=0.26*cos(c_g_r*(pos1+pos2));
pj22dq2=0.26*cos(c_g_r*(pos1+pos2));

pJJdq1=j11*pj22dq1+j22*pj11dq1-j12*pj21dq1-j21*pj12dq1;
pJJdq2=j11*pj22dq2+j22*pj11dq2-j12*pj21dq2-j21*pj12dq2;
dtJJ=pJJdq1*(c_g_r*vel1)+pJJdq2*(c_g_r*vel2);

//Las derivadas temporales de los elementos del jacobiano

dtj22=pj22dq1*(c_g_r*vel1)+pj22dq2*(c_g_r*vel2);
dtj12=pj12dq1*(c_g_r*vel1)+pj12dq2*(c_g_r*vel2);
dtj21=pj21dq1*(c_g_r*vel1)+pj21dq2*(c_g_r*vel2);
dtj11=pj11dq1*(c_g_r*vel1)+pj11dq2*(c_g_r*vel2);

//Las derivadas temporales de los elementos del jacobiano inverso

dtj22_JJ=(JJ*dtj22-j22*dtJJ)/(JJ*JJ);
dtj12_JJ=(JJ*dtj12-j12*dtJJ)/(JJ*JJ);
dtj21_JJ=(JJ*dtj21-j21*dtJJ)/(JJ*JJ);
dtj11_JJ=(JJ*dtj11-j11*dtJJ)/(JJ*JJ);

```



```

//Las derivadas temporales de el campo de velocidad aumentado

dtv1=( j22*dtvx-j12*dtvy)/JJ+dtj22_JJ*vx-dtj12_JJ*vy;
dtv2=(-j21*dtvx+j11*dtvy)/JJ-dtj21_JJ*vx+dtj11_JJ*vy;

MV_1=m11*v1q+m12*v2q;
MV_2=m21*v1q+m22*v2q;

dtMV_1=dtm11*v1q+m11*dtv1+dtm12*v2q+m12*dtv2;
dtMV_2=dtm21*v1q+m21*dtv1+dtm22*v2q+m22*dtv2;

dtVMV=v1q*dtMV_1+v2q*dtMV_2+dtv1*MV_1+dtv2*MV_2;

//La derivada temporal del elemento aumentado

dtv3=-((1/(m33))*dtVMV)/(sqrt((2/m33)*(E-(0.5)*VMV)));

//w=M(q)dV(q)+C(q,dq)dV(q) todos los elementos del sistema aumentado:

w11=m11*dtv1+m12*dtv2;
w21=m21*dtv1+m22*dtv2;
w31=m33*dtv3;

w12=c11*v1q+c12*v2q;
w22=c21*v1q+c22*v2q;
w23=0;

w1=w11+w12;
w2=w21+w22;
w3=m33*dtv3;

p1=m11*c_g_r*vel1+m12*c_g_r*vel2;
p2=m21*c_g_r*vel1+m22*c_g_r*vel2;
p3=m33*vel3;

PP1=m11*v1q+m12*v2q;
PP2=m21*v1q+m22*v2q;
PP3=m33*v3q;

// El controlador

xx11=w1*PP1-PP1*w1;
xx12=w1*PP2-PP1*w2;
xx13=w1*PP3-PP1*w3;
xx21=w2*PP1-PP2*w1;
xx22=w2*PP2-PP2*w2;
xx23=w2*PP3-PP2*w3;
xx31=w3*PP1-PP3*w1;
xx32=w3*PP2-PP3*w2;
xx33=w3*PP3-PP3*w3;

```



```

yy11=PP1*p1-p1*PP1;
yy12=PP1*p2-p1*PP2;
yy13=PP1*p3-p1*PP3;
yy21=PP2*p1-p2*PP1;
yy22=PP2*p2-p2*PP2;
yy23=PP2*p3-p2*PP3;
yy31=PP3*p1-p3*PP1;
yy32=PP3*p2-p3*PP2;
yy33=PP3*p3-p3*PP3;

tau_c1=(1/(2*E))*(xx11*c_g_r*vel1+xx12*c_g_r*vel2+xx13*vel3);
tau_c2=(1/(2*E))*(xx21*c_g_r*vel1+xx22*c_g_r*vel2+xx23*vel3);
tau_c3=(1/(2*E))*(xx31*c_g_r*vel1+xx32*c_g_r*vel2+xx33*vel3);

tau_f1=gama*(yy11*c_g_r*vel1+yy12*c_g_r*vel2+yy13*vel3);
tau_f2=gama*(yy21*c_g_r*vel1+yy22*c_g_r*vel2+yy23*vel3);
tau_f3=gama*(yy31*c_g_r*vel1+yy32*c_g_r*vel2+yy33*vel3);

/*Elemento extra*/

tau_k1=delta*PP1*(omega-(PP1*vel1*c_g_r+PP2*vel2*c_g_r+PP3*vel3)/(2*E));
tau_k2=delta*PP2*(omega-(PP1*vel1*c_g_r+PP2*vel2*c_g_r+PP3*vel3)/(2*E));
tau_k3=delta*PP3*(omega-(PP1*vel1*c_g_r+PP2*vel2*c_g_r+PP3*vel3)/(2*E));

/*Función Signo*/

if (vel1>0)
sign1=1;
else
sign1=-1;

if (vel2>0)
sign2=1;
else
sign2=-1;

/*Asignación de los pares de control*/

tau1=tau_c1+tau_f1+tau_k1+g1+b1*c_g_r*vel1+fc1*sign1;
if(tau1>15) tau1=15;
if(tau1<-15) tau1=-15;

tau2=tau_c2+tau_f2+tau_k2+g2+b2*c_g_r*vel2+fc2*sign2;
if(tau2>4) tau2=4;
if(tau2<-4) tau2=-4;

tau3=tau_c3+tau_f3+tau_k3; //Par ficticio

/*Actualización de los estados de flywheel*/

x1k=x1kn;

```

```
x2k=x2kn;  
  
/*Acumulador de la norma*/  
  
if(Tiempo_real>=5.0 && Tiempo_real<=5+T_norma){  
    norma=norma+ecvx*ecvx+ecvy*ecvy;  
}  
    indice=sqrt(norma/T_norma);  
}  
  
/* FIN DEL PROGRAMA */
```