

Centro Nacional de Investigación y Desarrollo Tecnológico

Subdirección Académica

Departamento de Ingeniería Electrónica

TESIS DE MAESTRÍA EN CIENCIAS

Control de un Robot SCARA Basado en Pasividad

presentada por

Ing. José Gabriel Zúñiga Tapia

como requisito para la obtención del grado de
Maestro en Ciencias en Ingeniería Electrónica

Director de tesis
Dr. Carlos Daniel García Beltrán

Codirector de tesis
Dr. Gerardo Vicente Guerrero Ramírez

Cuernavaca, Morelos, México. Enero de 2016.



"2015, Año del Generalísimo José María Morelos y Pavón"

Cuernavaca, Mor., 11 de enero de 2016

OFICIO No. DIE/208/2016

Asunto: Aceptación de documento de tesis

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial del C. José Gabriel Zúñiga Tapia, con número de control **M13CE088** de la Maestría en Ciencias en Ingeniería Electrónica, le informamos que hemos revisado el trabajo de tesis profesional titulado "**CONTROL DE UN ROBOT SCARA BASADO EN PASIVIDAD**" y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTOR DE TESIS

Dr. Carlos Daniel García Beltrán
Doctor en Ciencias en Ingeniería Electrónica
Cédula profesional 8699605

CODIRECTOR DE TESIS

Dr. Gerardo Vicente Guerrero Ramírez
Doctor en Ingeniería
Cédula profesional 3431842

REVISOR 1

Dr. Luis Gerardo Vela Valdés
Doctor en Ciencias en Ingeniería Electrónica
Cédula profesional 7980044

REVISOR 2

Dr. Ricardo Fabricio Escobar Jiménez
Doctor en Ciencias en Ingeniería Electrónica
Cédula profesional 7534115

C.p. Lic. Guadalupe Garrido Rivera.- Jefa del Departamento de Servicios Escolares.
Estudiante
Expediente

CMAZ/lrr



Cuernavaca, Mor., 12 de enero de 2016
OFICIO No. SAC/002/2016

Asunto: Autorización de impresión de tesis

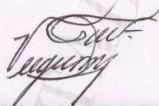
ING. JOSÉ GABRIEL ZÚÑIGA TAPIA
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS
EN INGENIERÍA ELECTRÓNICA
PRESENTE

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **"Control de un Robot Scara Basado en Pasividad"**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE

"CONOCIMIENTO Y TECNOLOGIA AL SERVICIO DE MEXICO"



DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO



CENTRO NACIONAL DE
INVESTIGACIÓN Y
DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA

C.p. Lic. Guadalupe Garrido Rivera.- Jefa del Departamento de Servicios Escolares.
Expediente

GVGR/mcr

HOJA EN BLANCO

Dedicatoria

Quiero dedicar este trabajo a mi familia, especialmente a mis padres que, gracias a ellos, por su dedicación, amor y apoyo que me han brindado durante toda mi vida, me motivan para seguir superándome día con día.

"Una buena familia comienza con un buen ejemplo de los padres".
Frase popular.

Agradecimientos

Quiero agradecer de una manera muy especial a todas las personas que me brindaron su apoyo y amistad durante mi estancia en la ciudad de Cuernavaca, especialmente a mis compañeros de generación, así también agradecer a las personas que, confiaron en mí y me dieron su apoyo para venir a esta ciudad.

A mi director de tesis el Dr. Carlos Daniel por haberme brindado su amistad, y al personal docente y administrativo del departamento de Electrónica por darme su apoyo en mi preparación.

Al CENIDET por haber confiado y dado la oportunidad de ingresar a su programa de postgrado, y coadyuvar a mi desarrollo personal y profesional.

Agradezco también al CONACYT por haberme brindado el apoyo necesario durante el desarrollo del postgrado.

"Lo que jamás ha sido cuestionado no puede ser de ningún modo probado, el escepticismo, es pues, el primer paso hacia la verdad".

Denis Diderot, extracto de: *Pensées Philosophiques* (1746)

Resumen

Los robots son sistemas mecánicos conformados por una gran variedad de componentes entre los que destacan los estructurales, los componentes de control, comunicación y la parte de cómputo que realiza el procesamiento de datos; a estos le son asignadas tareas que los humanos no podrían realizar, debido a sus limitaciones de precisión, velocidad y de ambiente.

La robótica industrial ha detonado la implementación de sistemas de control en sistemas mecánicos que demandan rapidez en sus algoritmos, una alta precisión y eficiencia en sus tareas.

Actualmente el grado de automatización en los procesos y el incremento en el nivel de seguridad, ha convertido la robótica en el primer elemento en consideración a la hora de diseñar un proceso flexible, eficiente y seguro. Podemos observar su implementación desde la robótica de servicios, robótica de exploración, hasta la robótica en medicina, militar y aeroespacial.

Para que un robot funcione se debe de diseñar un sistema de control, el cual es el encargado de los movimientos del mismo mediante la solución de complejas operaciones que se llevan a cabo dentro del controlador, el cual requiere conocer los parámetros físicos que definen la estructura y la dinámica del robot. A partir de estos parámetros, se obtienen ciertos patrones tales como posición y velocidad, y se realizan las tareas programadas de acuerdo a un objetivo de trabajo.

El robot industrial es un robot reprogramable multifuncional cuyo objetivo principal es maximizar la precisión, repetibilidad y velocidad de ejecución de tareas, todo esto sin importar las perturbaciones o dinámicas no modeladas que pudieran tener.

Se realiza un análisis del robot rígido de cuatro grados de libertad conocido como robot SCARA para obtener su modelo dinámico y de igual forma se obtiene el modelo cinemático inverso y directo.

El diseño del controlador propuesto se basa en una técnica de pasividad llamada IDA-PBC, enfocado en el *moldeo de energía* y el *balance energético* del sistema, eso se le conoce como *pasivar* el sistema.

Mediante pruebas de simulación, se realizan pruebas de robustez y estabilidad del robot, mediante una variación paramétrica en el modelo dinámico y fuerzas externas que actúan sobre los eslabones, y se determina bajo una perturbación acotada su comportamiento y el error.

Abstract

Robots are mechanical systems conformed by a lot of variety of components, among them, the structural ones, the components of control, communication and computing systems which performs data processing and programming, to which they are assigned tasks that humans not perform due to limitations of accuracy, speed and environment.

Industrial robotics has detonated the implementation of control system in mechanical systems which requires speed in their algorithms, a high precision and efficiency in their tasks.

Actually, the degree of automatization in process and the increasing of level of security, have become robotics in the first element in consideration in the develop of a flexible, efficient and safe process. We can observe the implementation of robotics from the robotics of service and exploration, medical up to the aeroespacial robotics.

To make a functional robot is necessary to design a control system which is responsible of its movements through the solution of complex operations which are developed in the controller and need to know the physical parameters that defines the structure and dynamic of our robot. From there, there are obtained certain parameters for example, position and speed and it perform the programmed tasks according with a work objective.

The industrial robot it's a multifunctional programmable robot with the main goal of maximize the precision, repeatability and execution speed of its task, no matter the not designed dynamics or perturbations.

We analyze the rigid four degrees of freedom robot, known as "SCARA" (Selective Compliant Articulated Robot Arm) to obtain its dynamic model and at the same time to obtain the inverse kinematic model.

The design of the controller its performed by a passivity technic called IDA-PBC (Interconnection and damping Assignment Passivity-Based Control) which is based on the *energy shaping* and obtaining an *energetic balance* of our system it is known as *passivate* the system.

We make an analysis based on simulation in terms of robot's robustness and stability based on a parametric variation in the dynamical model and external strengths which affect the chains, and the behavior and error are determined under a bounded perturbation.

Contenido

1	INTRODUCCIÓN	1
1.1	ANTECEDENTES	1
1.2	PLANTEAMIENTO DEL PROBLEMA	5
1.3	JUSTIFICACIÓN	6
1.4	OBJETIVOS	7
1.4.1	<i>Objetivo general</i>	7
1.4.2	<i>Objetivos específicos</i>	7
1.5	ESTADO DEL ARTE	7
1.6	MÉTODO DE SOLUCIÓN	8
1.7	METODOLOGÍA	8
1.8	ALCANCES	9
1.9	ESTRUCTURA DEL DOCUMENTO	9
2	MODELADO CINEMÁTICO	11
2.1	ESTRUCTURA DENAVIT-HARTENBERG	13
2.2	ALGORITMO DE D-H PARA LA OBTENCIÓN DEL MODELO CINEMÁTICO	16
2.3	OBTENCIÓN DE LOS PARÁMETROS D-H PARA EL ROBOT SCARA	18
2.4	OBTENCIÓN DEL MODELO DE CINEMÁTICA DIRECTA	20
2.5	OBTENCIÓN DEL MODELO DE CINEMÁTICA INVERSA	21
2.6	COMPROBACIÓN DE CINEMÁTICA INVERSA-DIRECTA	27
2.7	CONCLUSIÓN DEL CAPÍTULO	28
3	MODELADO DINÁMICO	29
3.1	ESTRUCTURA DEL ROBOT	29
3.2	OBTENCIÓN DEL MODELO EN EULER-LAGRANGE	31
3.3	PROPIEDADES DE LA ESTRUCTURA EULER-LAGRANGE	41
	<i>Propiedades de la matriz de Inercia $M(q)$</i>	41
	<i>Matriz de Coriolis $C(q, \dot{q})$</i>	41
	<i>Propiedad de la matriz de gravedad $gv(q)$</i>	42
	<i>Propiedades de la matriz de fuerzas disipativas $B(\dot{q})$</i>	42
	<i>Propiedad de la matriz de entrada u</i>	43
3.4	COMPARACIÓN DEL MODELO OBTENIDO CON ARTÍCULO DE REFERENCIA (VOGLEWEDE, SMITH, & MONTI, 2009)	43
3.5	REPRESENTACIÓN DEL MODELO EN HAMILTONIANO	44
3.6	CONCLUSIÓN DEL CAPÍTULO	47
4	PASIVIDAD Y DISEÑO DEL CONTROLADOR IDA-PBC	49
4.1	PRELIMINARES	51
4.1.1	<i>Balance de energía</i>	52
4.1.2	<i>Conceptos y estabilidad de Lyapunov (Kelly & Santibañez, 2003)</i>	53
4.2	DESCRIPCIÓN DEL MÉTODO IDA-PBC	57
4.3	OBTENCIÓN DEL CONTROLADOR	62
4.4	SINTONIZACIÓN DE GANANCIAS POR MEDIANTE LA OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS (PSO)	64
4.4.1	<i>Pasos del algoritmo</i>	67
4.4.2	<i>Ganancias obtenidas para la ley de control</i>	68
4.5	CONCLUSIÓN DEL CAPÍTULO	69

5	SIMULACIONES Y RESULTADOS.....	71
5.1	DESCRIPCIÓN DE LAS PRUEBAS.	71
5.2	REGULACIÓN.....	72
5.2.1	<i>Escenario 1: regulación sin perturbación.....</i>	<i>73</i>
5.2.2	<i>Escenario 2: Regulación con perturbación.....</i>	<i>76</i>
5.3	SEGUIMIENTO.	78
5.3.1	<i>Escenario 3: seguimiento sin perturbación.....</i>	<i>79</i>
5.3.2	<i>Escenario 4: Seguimiento con perturbación.....</i>	<i>81</i>
5.4	ESCENARIO 5: PERTURBACIÓN EN MASAS Y CONDICIONES INICIALES DIFERENTES.	84
5.5	ESCENARIO 6: TAREA PICK & PLACE.	87
5.6	CONCLUSIÓN DEL CAPÍTULO.	92
6	CONCLUSIONES Y TRABAJOS FUTUROS.....	93
6.1	TRABAJOS FUTUROS.....	94
	BIBLIOGRAFÍA	96
	APÉNDICE A	98
	APÉNDICE B	100
	APÉNDICE C.....	104
	APÉNDICE D	107

Índice de figuras.

Fig. 1.1: Robot Unimate fabricado en Japón (Barrientos, 2007).....	3
Fig. 1.2: Esquema general de control de un robot.....	5
Fig. 1.3: Unidades de robot industriales por sector de los últimos 3 años. (IFR Statistical department, 2015)	6
Fig. 2.1: Relación cinemática inversa y directa.	11
Fig. 2.2: Interpretación de los vectores de orientación y posición.	12
Fig. 2.3: Estructura de la matriz de transformación homogénea.	15
Fig. 2.4: Parámetros D-H para un eslabón giratorio (Barrientos, 2007).	17
Fig. 2.5: Obtención de parámetros D-H en Robotic Arm Kinematic GUI.	19
Fig. 2.6: Vista del robot desde el plano XY.	25
Fig. 2.7: Diagramas de bloque de cinemática inversa y directa.....	27
Fig. 3.1: Espacio de trabajo del robot SCARA 4GDL.	30
Fig. 3.2: Estructura física del robot SCARA.....	30
Fig. 3.3: Comparación del modelo obtenido vs artículo.	44
Fig. 4.1: Sistema visto como transformador de energía.	51
Fig. 4.2: Estabilidad de un sistema.	54
Fig. 4.3: Estabilidad asintótica.....	55
Fig. 4.4: Oscilador de Van der Pol con el origen del sistema como equilibrio inestable.	56
Fig. 5.1: Esquema de control del sistema en lazo cerrado.....	72
Fig. 5.2: Comparación del punto deseado vs. real en regulación sin perturbación.....	74
Fig. 5.3: Convergencia del error de posición en regulación sin perturbación.....	74
Fig. 5.4: Valores de las proyecciones n_x y o_y en regulación sin perturbación.	75
Fig. 5.5: Pares aplicados a los actuadores en regulación sin perturbación.....	75
Fig. 5.6: Comparación del punto deseado vs real en regulación con perturbación.....	76
Fig. 5.7: Valores de las proyecciones n_x y o_y en regulación con perturbación.....	77
Fig. 5.8: Pares aplicados a los actuadores en regulación con perturbación.	77
Fig. 5.9: Trayectoria deseada en el plano XY para seguimiento.....	78
Fig. 5.10: Comparación de la trayectoria deseada vs real en el plano XYZ.....	79
Fig. 5.11: Convergencia a cero en el error de posición del efector en seguimiento sin perturbación. .	80
Fig. 5.12: Valores de las proyecciones n_x y o_y en regulación sin perturbación.	80
Fig. 5.13: Pares aplicados a los actuadores en regulación sin perturbación.....	81
Fig. 5.14: Comparación de la trayectoria deseada vs real en seguimiento con perturbación.	82
Fig. 5.15: Error de posición en la trayectoria deseada vs real en seguimiento con perturbación.....	82
Fig. 5.16: Valores de las proyecciones n_x y o_y en seguimiento con perturbación.....	83
Fig. 5.17: Pares aplicados a los actuadores en seguimiento con perturbación.	83
Fig. 5.18: Convergencia de las posiciones al punto deseado con diferentes condiciones iniciales y masas.....	85
Fig. 5.19: Regulación ante una fuerza externa.....	86
Fig. 5.20: Definición gráfica de exactitud vs repetibilidad.	87

Fig. 5.21: Vista de la trayectoria descrita en las 3 simulaciones con 0%, 2.5% y 5% de aumento en masa 4.	89
Fig. 5.22: Vista del plano XY de la trayectoria con 0%, 2.5% y 5% de aumento en masa 4.	89
Fig. 5.23: Comparación de punto deseado y puntos obtenidos en pick & place plano XY.	90
Fig. 5.24: Comparación de punto deseado y puntos obtenidos en pick & place plano XZ.	91
Fig. A. 1: Bloque cinemática directa en SIMULINK.	98
Fig. A. 2: Bloques internos en cinemática directa.	98
Fig. A. 3: Bloque cinemática inversa en SIMULINK.	99
Fig. A. 4: Bloques internos en cinemática inversa.	99
Fig. C. 1: Controlador IDA-PBC en SIMULINK.	104
Fig. C. 1 Esquema de control con cinemáticas acopladas.	105
Fig. D. 1: Bloques para obtener el valor de fitness.	107
Fig. D. 2: Construcción de la señal de referencia para q_1	108
Fig. D. 3: Obtención de las áreas de error fuera de las referencias deseadas.	109
Fig. D. 4: Interior del bloque de comparación para obtener valor de fitness.	109
Fig. D. 5: Población inicial de partículas en PSO.	112
Fig. D. 6: Convergencia de los valores al finalizar el algoritmo PSO.	113

Lista de Tablas.

Tabla 2.1: Parámetros D-H del manipulador robótico.	19
Tabla 2.2: Evaluación cinemática inversa.....	28
Tabla 2.3: Evaluación cinemática directa.	28
Tabla 3.1: Parámetros del Robot SCARA.....	31
Tabla 4.1: Definición de términos del algoritmo PSO.	66
Tabla 4.2: Ganancias del controlador encontrados mediante PSO para el robot SCARA.....	69
Tabla 5.1: Punto deseado a regular en regulación.....	73
Tabla 5.2: Variación en las masas del modelo para simulación con perturbación.	84
Tabla 5.3: Intervalos de las condiciones iniciales para simulación con perturbación.	84
Tabla 5.4: Valores de simulación para perturbación externa.	86
Tabla 5.5: Puntos deseados para tarea pick & place.	88
Tabla 5.6: Puntos obtenidos en pick & place con 0% de aumento en masa 4.....	91
Tabla 5.7: Puntos obtenidos en pick & place con 2.5% de aumento en masa 4.....	91
Tabla 5.8: Puntos obtenidos en pick & place con 5% de aumento en masa 4.....	92
Tabla B. 1: Código en MATLAB para simulación del sistema SCARA.....	103
Tabla C. 1: Código en MATLAB de parámetros del sistema en lazo cerrado y cinemática.	106
Tabla D. 1: Código en MATLAB con algoritmo PSO de sintonización.....	111

Nomenclatura.

Por orden de aparición:

q_i	i -ésima coordenada generalizada (articulación del robot)
x	Coordenada x
y	Coordenada y
z	Coordenada z
n_i	Vector unitario de orientación proyección i
o_i	Vector unitario de orientación proyección i
a_i	Vector unitario de orientación proyección i
Px	Coordenada x del punto P
Py	Coordenada y del punto P
Pz	Coordenada z del punto P
X	Vector del espacio de trabajo
Y	Vector del espacio de trabajo
Z	Vector del espacio de trabajo
O	Origen del espacio de trabajo
$\{S_i\}$	i -ésimo sistema de coordenadas
θ_i	Rotación alrededor del eje Z_{i-1}
d_i	Traslación a lo largo de Z_{i-1}
a_i	Traslación a lo largo de X_i
α_i	Rotación alrededor del eje X_i
A_i	i -ésima matriz de transformación
T_i^{i-1}	Matriz de transformación relativa entre los sistemas asociados
$C_{1\dots i}$	$\cos(1+2+\dots+i)$
$S_{1\dots i}$	$\sin(1+2+\dots+i)$
Δ	Determinante
m_i	parámetro de masa del eslabón i
I_i	Parámetro de inercia del eslabón i

l_i	Parámetro de longitud del eslabón i
lc_i	Parámetro de centro de gravedad del eslabón
h_i	i -ésimo parámetro de longitud a través de Z del eslabón
$K_i(q, \dot{q})$	Energía cinética del eslabón i
$U_i(q)$	Energía potencial del eslabón i
$L(q, \dot{q})$	Ecuación de Lagrange
u_i	Par para el eslabón i
F	Función de disipación de Rayleigh
v_i	Velocidad del eslabón i
g	Constante de gravedad
h	Altura
B_i	Constante de fricción para el eslabón i
$M(q)$	Matriz de inercias
$C(q, \dot{q})$	Matriz de efectos coriolis
$B(\dot{q})$	Matriz de efectos de disipación
$gv(q)$	Matriz de efectos gravitatorios
I	Matriz identidad
μ_i	i -ésimo escalar
J	Matriz antisimétrica
τ_i	Par del eslabón i
H	Función Hamiltoniana
p	Coordenada generalizada de momento
J	Matriz de interconexión del sistema
x	Vector de estados del sistema
D	Matriz de amortiguamiento
G	Matriz de entrada al sistema
$g(x)$	Función de entrada del sistema
R	Matriz de disipación del sistema (B)
J_d	Matriz de interconexión deseada

H_d	Función Hamiltoniana deseada
R_d	Matriz de disipación deseada
$y(t)$	Salida pasiva del sistema
$d(t)$	Función de disipación simplificada del sistema
t	Variable tiempo
K_{vi}	i -ésima ganancia de retroalimentación de amortiguamiento
$A(t)$	Matriz que define la dinámica de un sistema lineal
δ	Constante positiva
ε	Constante positiva
α	Constante positiva
β	Constante positiva
M_d	Matriz de inercia deseada
u_{es}	Moldeo de energía
u_{di}	Inyección de amortiguamiento
K_{pi}	i -ésima ganancia del controlador
q_{di}	i -ésima coordenada generalizada deseada
\tilde{q}	Señal de error

Abreviaturas y acrónimos.

AMF	American Machine and Foundry
ANSI	American National Standards Institute
ASEA	Allmänna Svenska Elektriska Aktiebolaget, (English translation: General Swedish Electric Company)
D-H	Denavit-Hartenberg
E-L	Euler-Lagrange
Fig.	Figura
GDL	Grados De Libertad
IDA-PBC	Interconnection and Damping Assignment Passivity-Based Control
IFR	International Federation of Robotics
ISO	International Organization for Standardization
JIRA	Japanese Industrial Robot Association
MATLAB	Matrix Laboratory
MIT	Massachusetts Institute of Technology
MLD	Matriz Lado Derecho
PBC	Passivity-Based Control
PCH	Port-Controlled Hamiltonian
PD	Proporcional Derivativo
PDE	Partial Differential Equation
PID	Proporcional Integral Derivativo
PSO	Particle Swarm Optimization
RIA	Robotics Industry Association
RRPR	Rotativo Rotativo Prismático Rotativo
RRRP	Rotativo Rotativo Rotativo Prismático
SCARA	Selective Compliant Articulated Robot Arm
UNIMATE	Universal Automation

1 Introducción.

En este capítulo se presentan los antecedentes del proyecto, los objetivos y la justificación del proyecto. Después se presenta una revisión del estado del arte y la estructura del documento.

1.1 Antecedentes

La palabra robot fue usada por primera vez en el año 1921, cuando el escritor checo Karel Capek (1890-1938) estrena en el teatro nacional de Praga su obra *Rossum's Universal Robot (R.U.R.)*. Su origen es de la palabra eslava *robot* que se refiere al trabajo realizado de manera forzada (Barrientos, 2007).

Las principales ventajas del uso de la robótica son:

- Seguridad en los procesos.
- Una alta precisión y fuerza.
- Altos niveles de automatización en una planta.
- Reducción de espacios.

La robótica se puede clasificar en 2 grandes grupos:

- Robótica fija.
- Robótica móvil.

Dentro de la robótica fija se encuentra la robótica industrial de manipuladores, la cual se puede clasificar por su tipo de configuración articular:

- Robot cartesiano.
- Robot cilíndrico.
 - **SCARA**

- Robot polar.
- Eslabones unidos (paralelos).
- Configuraciones mixtas.

Con el objetivo de diseñar una máquina flexible, adaptable al entorno y de fácil manejo, George Devol, pionero de la robótica industrial, patentó en 1948, un manipulador programable que fue el germen del robot industrial. Después en 1948 R.C. Goertz del *Argonne National Laboratory*, desarrolló el primer robot teleoperado mediante un esquema esclavo-maestro, con el fin de replicar los movimientos de un manipulador a otro para manejar elementos radioactivos.

Años más tarde Goertz hizo uso de la electrónica y servocontrol, para mejorar su versión anterior y crear el primer telemanipulador con servocontrol. Otro pionero fue Ralph Mosher del grupo *General Electric* que en 1958 desarrolló el dispositivo conocido como *Handy-Man* parecido a un exoesqueleto.

Años más tarde, se sustituyó el operador por un programa de ordenador que controlaba los movimientos del manipulador, y esto dio paso al concepto de robot tal y como se le conoce hoy en día. La primera patente de un dispositivo robótico, fue solicitada en marzo de 1954 por el inventor británico C.W. Kenward y fue emitida en el Reino Unido en 1957, sin embargo fue George C. Devol, un ingeniero norteamericano quien estableció las bases del robot industrial moderno y en 1954 concibió la idea de un dispositivo de transferencia de datos programada que se patentó en Estados Unidos en 1961.

En 1956 Joseph F. Engelberger, director de ingeniería de la división aeroespacial de la empresa Manning Maxwell y Moore en Stanford, Connecticut junto con Devol, comenzaron a trabajar en la utilización industrial de sus máquinas, fundando la Consolidated Controls Corporation, que más tarde se convierte en *Unimation* (Universal Automation), e instalando su primera máquina *Unimate* (Fig. 1.1) en la fábrica de General Motors de Trenton, Nueva Jersey, en una aplicación de fundición por inyección. Otras grandes empresas como AMF, emprendieron la construcción de máquinas similares.

En 1968 J.F. Engelberger visitó Japón y poco más tarde se firmaron acuerdos con Kawasaki para la construcción de robots tipo *Unimate*. El crecimiento de la robótica en Japón aventaja en breve a los Estados Unidos gracias a Nissan, que formó la primera

asociación robótica del mundo, la Asociación de Robótica industrial de Japón (JIRA) en 1972. Dos años más tarde se formó el Instituto de Robótica de América (RIA), que en 1984 cambió su nombre por el de Asociación de Industrias Robóticas, manteniendo las mismas siglas (RIA).

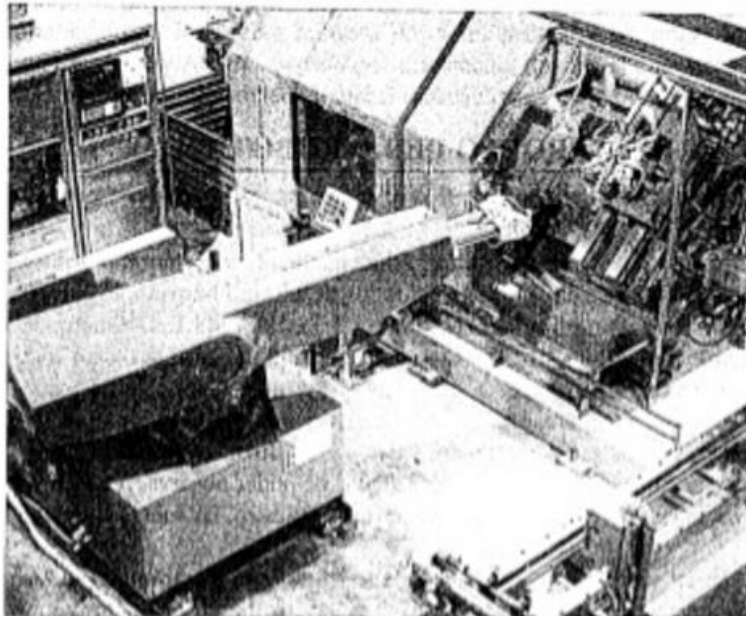


Fig. 1.1: Robot Unimate fabricado en Japón (Barrientos, 2007).

Por su parte Europa tuvo un despertar más tardío. En 1973 la firma sueca ASEA construyó el primer robot con accionamiento totalmente eléctrico, en 1980 se fundó la Federación Internacional de Robótica (IFR) con sede en Estocolmo Suecia.

La configuración de los primeros robots respondía a las denominadas configuraciones esférica y antropomórfica, de uso especialmente válido para la manipulación. En 1982, el profesor Makino de la Universidad Yamanashi de Japón, desarrolla el concepto de robot SCARA (*Selective Compliance Assembly Robot Arm*) que busca un robot con un número reducido en grados de libertad (3 o 4), un coste limitado y una configuración orientada al ensamblado de piezas.

Se pueden distinguir cinco fases relevantes en el desarrollo de la Robótica Industrial:

1. Argonne Laboratory diseña, en 1950, manipuladores amo-esclavo para manejar material radioactivo.

2. Unimation, fundada en 1958 por Engelberger, realiza los primeros proyectos de robots a principios de la década de los sesentas de nuestro siglo, instalando el primero en 1961 y posteriormente en 1967, un conjunto de ellos en una fábrica de *General Motors*. Tres años después, se inicia la implantación de los robots en Europa, especialmente en el área de fabricación de automóviles. Japón comienza a implementar esta tecnología hasta 1968.
3. Los laboratorios de la Universidad de Stanford y del MIT acometen, en 1970, la tarea de controlar un robot mediante computador.
4. En el año de 1975, la aplicación del microprocesador, transforma la imagen y las características del robot, hasta entonces grande y costoso.
5. A partir de 1980, el fuerte impulso en la investigación, por parte de las empresas fabricantes de robots, otros auxiliares y diversos departamentos de Universidades de todo el mundo, potencian la configuración del robot inteligente capaz de adaptarse al ambiente, tomar decisiones en tiempo real y adecuarlas para cada situación.

En esta fase de 1975 a 1980, la conjunción de los efectos de la revolución de la microelectrónica y la revitalización de las empresas automovilísticas, produjo un crecimiento acumulativo del parque de robots, cercano al 25%.

La evolución de los robots industriales desde sus principios ha sido vertiginosa. En poco más de 30 años las investigaciones y desarrollos sobre robótica industrial han permitido que los robots tomen posiciones en casi todas las áreas productivas y tipos de industria. En pequeñas o grandes fábricas, los robots pueden sustituir al hombre en aquellas áreas repetitivas y hostiles, adaptándose inmediatamente a los cambios de producción solicitados por la demanda variable.

El esquema de control de un robot como se conoce hoy en día se basa en un dispositivo electrónico conocido como controlador, que se encarga de adquirir y generar las señales apropiadas para realizar una tarea previamente programada por un operador de manera autónoma. (Barrientos, 2007).

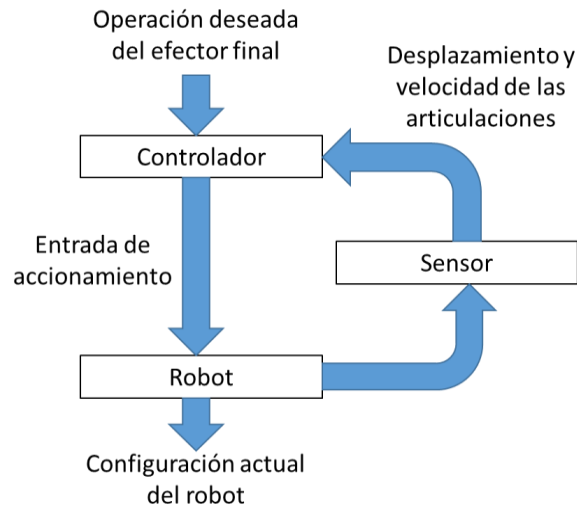


Fig. 1.2: Esquema general de control de un robot.

1.2 Planteamiento del problema.

Se debe mantener una posición y trayectoria del robot deseada, así como también estabilidad y robustez en un punto de operación de un robot tipo SCARA.

El robot industrial es un robot reprogramable multifuncional que consta de una estructura rígida sujeta a ciertos casos de flexibilidad por parte de sus elementos actuadores, esta flexibilidad se presenta en la transmisión de movimiento y se deben de considerar en términos de robustez y precisión.

En el modelado de robots manipuladores, generalmente encontramos que desconocemos algunos de sus parámetros. También existe el caso de las diferentes cargas, en el que no conocemos su valor de inercia.

Dicho esto se debe mantener una estabilidad en el punto deseado, presentar cierto grado de robustez a diferentes cargas de trabajo, precisión y repetitividad a pesar de la presencia de incertidumbres y comportamientos dinámicos no modelados.

En este trabajo se busca mejorar el desempeño de un control clásico con las nuevas técnicas de control enfocadas en energía, como son las de pasividad. Esta técnica de pasividad está basada en el control robusto con lo cual se espera que el controlador sea una buena alternativa para las diferentes perturbaciones que entren al sistema y resolver problemas tales como las diferentes cargas de trabajo.

1.3 Justificación.

Debido a las exigencias en los sistemas de producción, se está introduciendo una gran variedad de avances tecnológicos en las plantas industriales de todos los niveles. La industria automotriz, es la más demandante de equipos robóticos donde su principal aplicación es la soldadura, pintura y manipulación de partes.

Las capacidades de un robot manipulador están determinadas por la precisión del movimiento, repetitividad, capacidad de carga, el efector final, los sensores, grados de libertad del robot y la confiabilidad del controlador. Es por eso que es importante que nuestros sistemas de control sean más confiables y precisos.

La actual demanda en la robótica industrial ha hecho que esta sea un campo muy atractivo que en los últimos años ha tenido un crecimiento notable (Fig. 1.3).

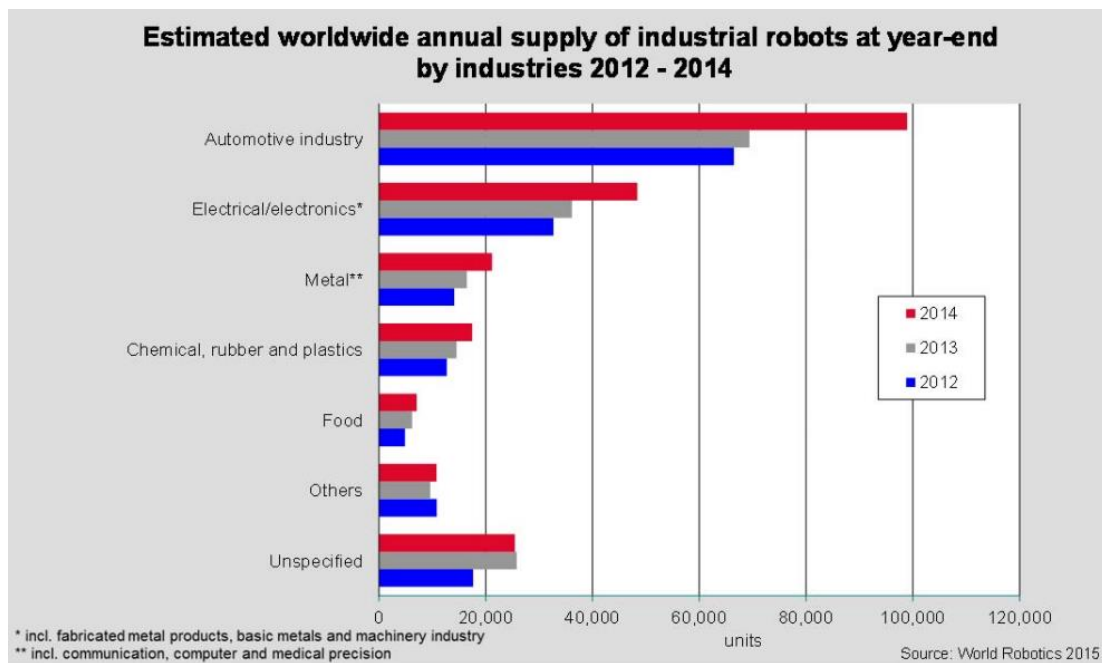


Fig. 1.3: Unidades de robot industriales por sector de los últimos 3 años. (IFR Statistical department, 2015)

El estudio de controladores sencillos es la solución para la pequeña y mediana industria, la cual tiene necesidad de automatizar sus procesos, pero también adquirir aplicaciones que se adapten exclusivamente a su proceso, es decir llevar la robótica industrial de una manera accesible a la industria de menor escala y recursos.

La técnica PBC desde su aparición en (Ortega, 1989), ha sido de mucho interés en la comunidad científica, de tal forma que ha sido uno de los artículos más citados en la revista *Automatica*.

1.4 Objetivos.

1.4.1 Objetivo general.

El Objetivo principal es controlar un robot tipo SCARA de cuatro grados de libertad aplicando una ley de control basada en el enfoque IDA-PBC.

Se debe mantener estabilidad en el punto deseado y presentar cierto grado de robustez a diferentes cargas de trabajo, precisión y repetitividad a pesar de la presencia de incertidumbres y comportamientos dinámicos no modelados.

1.4.2 Objetivos específicos.

- Desarrollar un modelo cinemático
- Desarrollar un modelo dinámico desde el enfoque Euler-Lagrange y obtener su representación en Hamiltoniano Controlado por Puertos..
- Diseñar un Control basado en IDA-PBC.
- Analizar la robustez del sistema de control ante cambios en algunos parámetros del sistema.

1.5 Estado del arte.

Han sido mucho los estudios relacionados al control de robots manipuladores utilizando diferentes métodos tales como el control clásico, robusto y adaptable. Primero surgieron los controladores tradicionales conocidos como PD y PID que son bastante usados actualmente, su descripción se puede observar en (Takegaki & Arimoto, 1981) y en (Kelly & Santibañez, 2003), después los controladores adaptables, en (Ortega, 1989) y en (Slotine & Li, 1988), siendo Ortega el que introdujo el termino PBC (*Passivity-Based control*), finalmente aparecieron los controladores del tipo robusto, como se documenta en (Sotirov, 1992), (Slotine, 1985) y (Lu & Erlic, 1989).

Se han realizado diversas comparativas entre controladores como en (Yao & Tomizuka, 1994), donde reporta como se obtienen diseños de controladores tipo

robusto y adaptable. A partir del análisis se determina que los controladores adaptables presentan mejor seguimiento de trayectoria que los del tipo robusto.

En (Loizidis & Doulgeri, 1995) se realiza la comparativa entre un controlador de modos deslizantes y un controlador PD, donde se demuestra que el desempeño del controlador de modos deslizante presenta un mejor desempeño.

Fue en 1989 cuando se introdujo el termino pasividad, y en (Ortega, van der Schaft, Maschke, & Escobar, 2002) donde se introduce la metodología IDA-PBC, y en (Ortega & Garcia-Canseco, 2004) se realiza un caso de aplicación a un péndulo con una rueda inercial, existen muchas aplicaciones a sistemas mecánicos para resolver problemas de estabilidad tales como (Piña & Arteaga, 2007) utilizado en un sistema sub actuado conocido como *acrobot*, (Banavar, Kazi, Ortega, & Manjarekar, 2006) aplicado a una grúa viajera sub-actuada, (Acosta & Rios-Bolívar, 2010) aplicada a un sistema *pendubot*, y en (Paim, Neves, Moreno, & de Pieri, 2011) aplicado a un robot SCARA de tres grados de libertad.

1.6 Método de solución.

Se realiza un análisis mediante simulación de la operación del robot rígido de cuatro grados de libertad serial RRRP conocido como tipo SCARA, y se establece una ley de control de regulación con la metodología IDA-PBC para hacer un análisis de robustez ante perturbaciones en sus masas y fuerzas que actúan en los actuadores, mediante una variación del $\pm 5\%$ del valor de las masas de sus eslabones y la simulación de fuerzas externas.

1.7 Metodología.

El presente trabajo se realiza mediante la siguiente metodología:

1. Modelado matemático del robot.
2. Análisis de la operación en lazo abierto.
3. Diseño del controlador.
4. Sintonización de las ganancias.
5. Simulación del sistema en lazo cerrado.
6. Análisis de operación en lazo cerrado.
7. Resultados.

1.8 Alcances.

Los alcances de este proyecto de investigación fueron, la obtención del modelo del robot rígido de cuatro grados de libertad mediante la estructura E-L, para posteriormente encontrar la representación equivalente en Hamilton, la obtención del modelo cinemático, una ley de control de regulación, sintonización de las ganancias por medio de un método heurístico y sus respectivos programas en MATLAB para realizar las simulaciones

1.9 Estructura del documento.

En este capítulo se introduce a una breve historia del desarrollo de la robótica industrial, la justificación, los objetivos y los trabajos realizados en el área de control de sistemas mecánicos robóticos.

En el capítulo dos se muestra el desarrollo del modelado dinámico mediante la estructura E-L, para conocer el comportamiento dinámico del robot y conocer su grado de actuación y grados de libertad.

En el capítulo tres se desarrollan los modelos cinemáticos inverso y directo, los cuales son necesarios para conocer el comportamiento espacial del robot en toda su estructura y saber en todo momento la posición de los eslabones y del efector final.

En el capítulo 4 se introducen los conceptos relacionados con pasividad y el diseño del controlador bajo la estructura Hamiltoniana, y se obtiene una ley de control para regulación; también se sintoniza las ganancias del controlador mediante un método heurístico conocido como PSO.

En el capítulo 5 se exponen una serie de simulaciones en lazo cerrado y se resuelve el problema de regulación, se realiza una tarea *pick & place* y un análisis de robustez.

Por último en el capítulo 6 se presentan las conclusiones, los objetivos cubiertos, aportación y los trabajos futuros derivados de este trabajo de investigación.

2 Modelado cinemático

La cinemática se encarga de estudiar los movimientos de los cuerpos sin considerar las fuerzas y las masas, solamente se limita a sus movimientos. El desarrollo de la cinemática del robot nos permitirá conocer las relaciones de posición y orientación de cada uno de los eslabones del robot y poder realizar los movimientos deseados para cumplir las tareas definidas.

Existen dos problemas fundamentales a resolver en la cinemática del robot. El primero de ellos se conoce como el problema de la cinemática directa, y consiste en determinar cuál es la posición y orientación del extremo final del robot con respecto a un sistema de coordenadas que se toma como referencia, conocido como los valores de articulaciones y los parámetros geométricos de los elementos del robot. El segundo denominado problema cinemático inverso, resuelve la configuración que debe adoptar el robot para una posición y orientación del extremo conocidas.

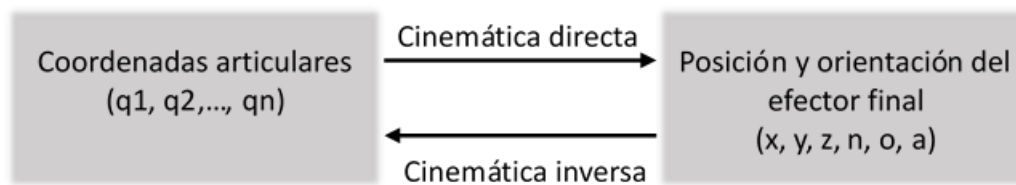


Fig. 2.1: Relación cinemática inversa y directa.

La metodología utilizada para este desarrollo es la llamada “metodología Denavit-Hartenberg” (Barrientos, 2007), la cual se realiza por medio de matrices homogéneas en las que se puede encontrar la posición y orientación de cada uno de los eslabones partiendo de una referencia global, así pues existe un marco de referencia para cada uno de los eslabones, la cual va tomando como referencia el marco anterior.

Para definir un punto u objeto en el espacio es necesario conocer su posición y su orientación, tomando en cuenta una referencia la cual es fija en el caso de los robots

manipuladores. Esta posición se define mediante los puntos P_x , P_y y P_z , y la orientación mediante los vectores unitarios de orientación noa , para interpretarlos se muestra la siguiente figura:

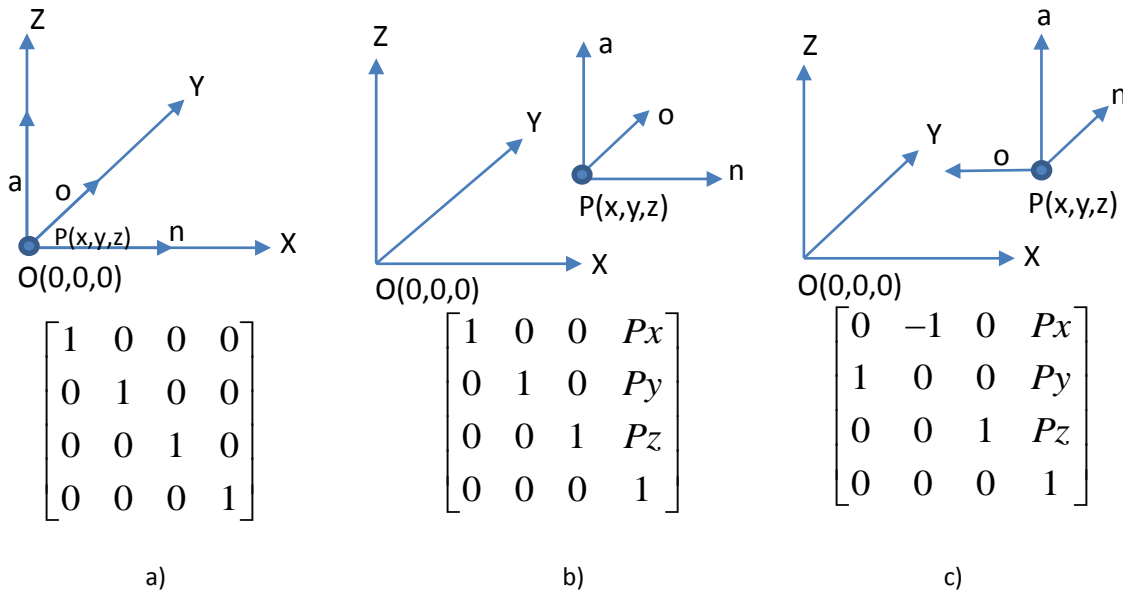


Fig. 2.2: Interpretación de los vectores de orientación y posición.

De acuerdo con Fig. 2.2, el vector de posición $P(x,y,z)$ describe la localización en el espacio del objeto a partir del punto de origen O , teniendo como unidad distancia, y la matriz de orientación nos da la proyección del vector unitario noa sobre cada uno de los ejes del origen, por ejemplo en a) se muestra que el punto P es coincidente con el origen O , y las proyecciones del vector noa son unitarias sobre los ejes XYZ , en b) el punto P ha sido desplazado una distancia a través del espacio, pero su orientación sigue siendo colineal con cada uno de los ejes así que su proyecciones siguen siendo unitarias, en cambio en c) el punto P ha sido desplazado cierta distancia y ha sido girado, por lo cual su orientación ha cambiado, siendo la proyección de n sobre x cero (n_x), pero n_y unitario, y o_x unitario pero en sentido contrario, así que en la matriz de rotación se representa con un signo negativo.

2.1 Estructura Denavit-Hartenberg

Denavit y Hartenberg propusieron un método sistemático para describir y representar la geometría espacial de los elementos de una cadena cinemática, en particular de un robot con respecto a un sistema de referencia fijo. Este método utiliza una matriz de transformación homogénea para describir la relación espacial entre dos elementos rígidos adyacentes, reduciéndose el problema cinemático directo a encontrar una matriz de transformación homogénea de 4x4 que relacione la localización espacial del extremo del robot con respecto al sistema de coordenadas de su base. Esta matriz estará en función de las coordenadas articulares.

En la descripción de la relación que existe entre dos elementos contiguos, se puede hacer uso de cualquier sistema de referencia ligado a cada elemento, sin embargo la forma habitual que se suele utilizar en robótica es la representación de D-H. Denavit y Hartenberg, propusieron en 1995 un método matricial que permite establecer de manera sistemática un sistema de coordenadas $\{S_i\}$ ligado a cada eslabón i de una cadena articulada, pudiéndose determinar a continuación las ecuaciones cinemáticas de la cadena completa (Barrientos, 2007).

Según la representación de D-H, escogiendo adecuadamente los sistemas de coordenadas asociados a cada eslabón, será posible pasar de uno al siguiente mediante cuatro transformaciones básicas que dependen exclusivamente de las características geométricas de cada eslabón.

Estas transformaciones básicas consisten en una sucesión de rotaciones y traslaciones que permiten relacionar el sistema de referencia del elemento i con el elemento $i-1$. Las transformaciones son las siguientes:

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

Dado que el producto de matrices no es conmutativo, las transformaciones se deben de realizar en el orden indicado. De este modo se obtiene:

$$A_i = T_i^{i-1} = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \quad (2.1.1)$$

donde las matrices (2.1.2),(2.1.3),(2.1.4) y (2.1.5) representan las rotaciones y traslaciones del marco de referencia.

$$Rot_{z,\theta_i} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1.2)$$

$$Trans_{z,d_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1.3)$$

$$Trans_{x,a_i} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1.4)$$

$$Rot_{x,\alpha_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1.5)$$

Los parámetros θ_i , a_i , d_i y α_i , son los parámetros D-H del eslabón i . De este modo, basta con identificar estos parámetros para obtener las matrices A_i y así poder relacionar a todos y cada uno de los eslabones del robot.

Realizando las debidas multiplicaciones tenemos la matriz (2.1.6) que corresponde a la forma simplificada de cada uno de los marcos de referencia.

$$A_i = T_i^{i-1} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1.6)$$

La estructura de la matriz de transformación homogénea simplificada para cada uno de los eslabones se presenta en Fig. 2.3.

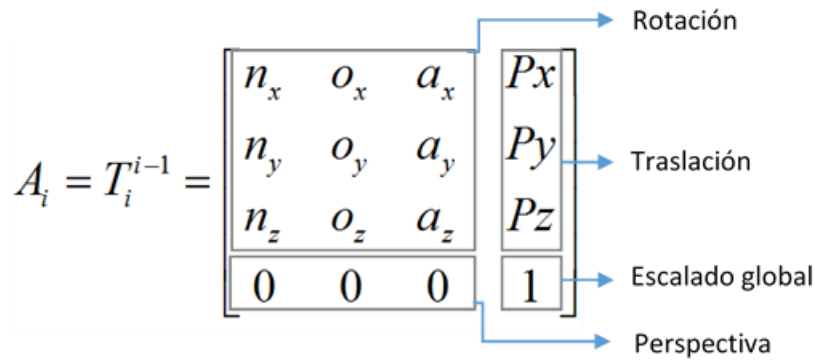


Fig. 2.3: Estructura de la matriz de transformación homogénea.

En general un robot de n grados de libertad está formado por eslabones unidos por n articulaciones, de forma que cada par articulación-eslabón constituye un grado de libertad. A cada eslabón se le puede asociar un sistema de referencia solidario a él y, utilizando las transformaciones homogéneas, es posible representar las rotaciones y traslaciones relativas entre los distintos eslabones que componen el robot. Normalmente, la matriz de transformación homogénea, que representa la posición y orientación relativa entre los sistemas asociados a dos eslabones consecutivos del robot, se suele denominar matriz T_i^{i-1} . De esta forma, T_1^0 describe la posición y orientación del sistema de referencia solidario al primer eslabón, con respecto al sistema de referencia solidario a la base, T_2^1 describe la posición y orientación del segundo eslabón respecto del primero, etc. Del mismo modo, denominando T_n^0 o matriz de transformación total, a las matriz resultante del producto de las matrices T_i^{i-1} con i desde 1 hasta n , con el que se puede representar de forma total o parcial la cadena cinemática que forma al robot.

2.2 Algoritmo de D-H para la obtención del modelo cinemático.

El algoritmo se trata de un procedimiento sistemático para obtener los parámetros D-H del robot y se describen por los siguientes pasos (Barrientos, 2007):

1. Numerar los eslabones comenzando con 1 (el primer eslabón móvil de la cadena) y acabando con n (último eslabón móvil). Se numera como eslabón 0 a la base fija del robot.
2. Numerar cada articulación comenzando por 1 (la correspondiente al primer grado de libertad) y acabando en n .
3. Localizar el eje de cada articulación. Si esta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento.
4. Para i de 0 a $n-1$ situar el eje Z_i sobre el eje de la articulación $i+1$.
5. Situar el origen del sistema de la base $\{S_i\}$ en cualquier punto del eje Z_0 . Los ejes X_0 e Y_0 se situaran de modo que formen un sistema dextrógiro con Z_0 .
6. Para i de 1 a $n-1$, situar el sistema $\{S_i\}$ (solidario al eslabón i) en la intersección del eje Z_i con la línea normal común a Z_{i-1} y Z_i . Si ambos ejes se cortasen se situaría $\{S_i\}$ en el punto de corte. Si fuesen paralelos $\{S_i\}$ se situaría en la articulación $i+1$.
7. Situar X_i en la línea normal común a Z_{i-1} y Z_i .
8. Situar Y_i de modo que forme un sistema dextrógiro con X_i y Z_i .
9. Situar el sistema $\{S_n\}$ en el extremo del robot de modo que Z_n coincida con la dirección de Z_{n-1} y X_n sea normal a Z_{n-1} y Z_n .
10. Obtener θ_i como el ángulo que hay que girar entorno a Z_{i-1} para que X_{i-1} y X_i queden paralelos.

11. Obtener d_i como la distancia medida a lo largo de Z_{i-1} , que habría que desplazar $\{S_{i-1}\}$ para que X_i y X_{i-1} quedasen alineados.
12. Obtener a_i como la distancia medida a lo largo de X_i (que ahora coincidiría con X_{i-1}) que habría que desplazar el nuevo $\{S_{i-1}\}$ para que su origen coincidiese con $\{S_i\}$.
13. Obtener α_i como el ángulo que habría que girar entorno a X_i (que ahora coincidiría con X_{i-1}), para que el nuevo $\{S_{i-1}\}$ coincidiese totalmente con $\{S_i\}$.
14. Obtener las matrices de transformación T_i^{i-1} definida en (2.1.6).
15. Obtener las matrices de transformación que relaciona al sistema de la base con el del extremo del robot $T_n^0 = T_1^0 T_2^1 \dots T_n^{n-1}$.
16. La matriz T_n^0 define la orientación (submatriz de rotación) y posición (submatriz de traslación) del extremo referido a la base en función de las n coordenadas articulares.

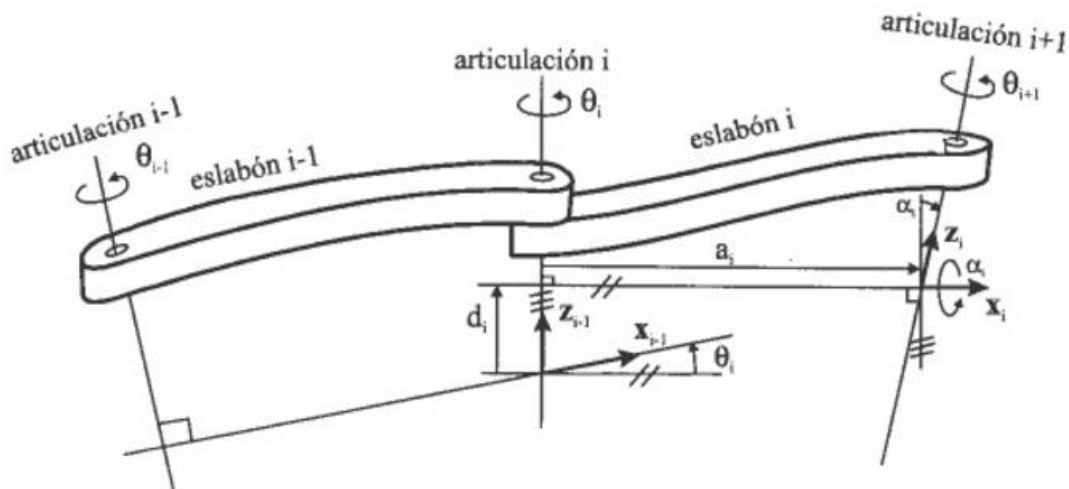


Fig. 2.4: Parámetros D-H para un eslabón giratorio (Barrientos, 2007).

Los cuatro parámetros de D-H $(\theta_i, d_i, a_i, \alpha_i)$ dependen únicamente de las características geométricas de cada eslabón y de las articulaciones que le unen con el anterior y siguiente. En concreto estos representan la Fig. 2.4.

El parámetro θ_i , es el ángulo que forman los ejes X_{i-1} y X_i medido en un plano perpendicular al eje Z_{i-1} , utilizando la regla de la mano derecha. Se trata de un parámetro variable de articulaciones giratoria.

El parámetro d_i , es la distancia a lo largo del eje Z_{i-1} desde el origen del sistema de coordenadas $(i-1)$ -ésimo hasta la intersección del eje Z_{i-1} con el eje X_i . Se trata de un parámetro variable en articulaciones prismáticas.

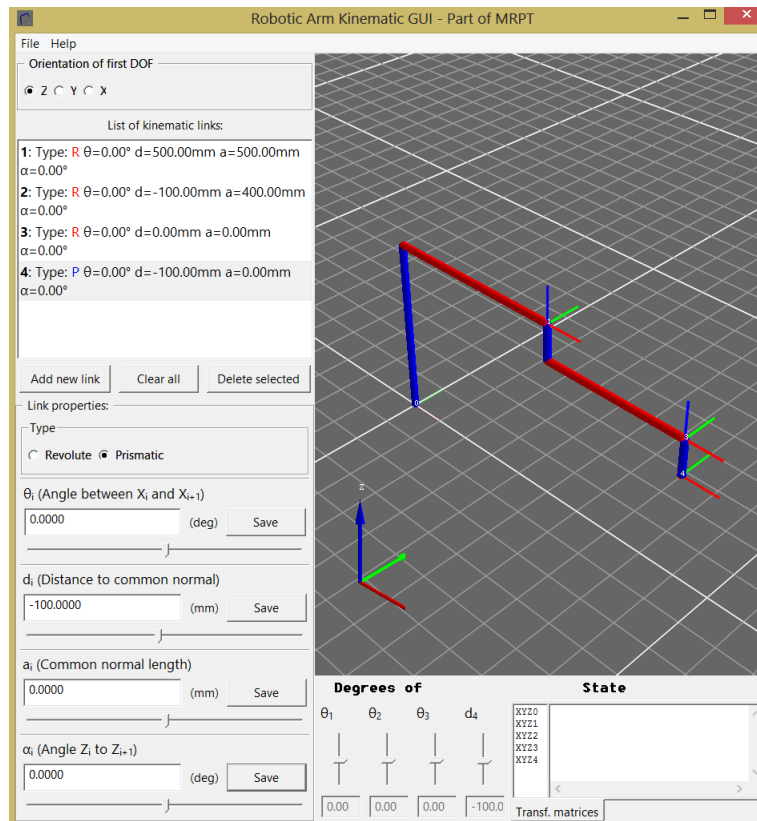
El parámetro a_i , es la distancia a lo largo del eje X_i que va desde la intersección del eje Z_{i-1} con el eje X_i hasta el origen del sistema i -ésimo, en el caso de articulaciones giratorias. En el caso de articulaciones prismáticas, se calcula como la distancia más corta entre los ejes Z_{i-1} y Z_i .

El parámetro α_i , es el ángulo de separación del eje Z_{i-1} y Z_i , medido en un plano perpendicular al eje X_i , utilizando la regla de la mano derecha.

Una vez obtenidos los parámetros D-H, el cálculo de las relaciones entre los eslabones del robot es inmediato, ya que vienen dadas por las matrices T , que se calculan según la expresión general (2.1.6). Se puede obtener la relación de entre cada uno de los eslabones, desarrollando la matriz correspondiente y haciendo la pre-multiplicación del marco de referencia anterior, para así obtener la ubicación y orientación de cada uno de los eslabones respecto a la base.

2.3 Obtención de los parámetros D-H para el robot SCARA

De acuerdo al algoritmo de D-H, es necesario encontrar los parámetros que nos definen el modelo cinemático del robot en cuestión, para esto se utilizó el *software Robotic Arm Kinematic GUI* (Mobile Robot Programming Toolkit, 2015). Este programa hace uso del algoritmo anterior, se construye la articulación de forma gráfica y se obtienen los parámetros D-H. Teniendo esto, insertamos en el programa los tres primero eslabones rotativos y el eslabón prismático, luego damos las distancias que existen entre los diferentes marcos de referencia, y el brazo robótico queda como se muestra en Fig. 2.5.

Fig. 2.5: Obtención de parámetros D-H en *Robotic Arm Kinematic GUI*.

Dando como resultado la siguiente tabla:

Tabla 2.1: Parámetros D-H del manipulador robótico.

# i	θ Rot. en Z [°]	d Desp. en z [mm]	a Desp. en x [mm]	α Rot. En z [°]
1	θ_1	$d_1 = 500$	$a_1 = 500$	$\alpha_1 = 0$
2	θ_2	$d_2 = -100$	$a_2 = 400$	$\alpha_2 = 0$
3	θ_3	$d_3 = 0$	$a_3 = 0$	$\alpha_3 = 0$
4	$\theta_4 = 0$	$d_4 = d - 100$	$a_4 = 0$	$\alpha_4 = 0$

Para cada renglón de Tabla 2.1 se establece la matriz de transformación según (2.1.6) , y así obtener la cinemática directa e inversa de nuestro robot como se muestra en la sección 2.4.

2.4 Obtención del modelo de cinemática directa.

La cinemática directa del robot se realiza sustituyendo para cada i hasta n , la matriz de transformación T , para después realizar una serie de multiplicaciones y encontrar la matriz de transformación total, es decir desde la base de nuestro manipulador hasta el extremo final del robot queda de la siguiente forma:

$$A_4 = T_4^0 = T_1^0 T_2^1 T_3^2 T_4^3 = \begin{bmatrix} n_x & o_x & a_x & Px \\ n_y & o_y & a_y & Py \\ n_z & o_z & a_z & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4.1)$$

donde T_1^0 , T_2^1 , T_3^2 y T_4^3 son las matrices para cada marco de referencia.

$$T_1^0 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & a_1 \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & a_1 \sin \theta_1 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4.2)$$

$$T_2^1 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & a_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & a_2 \sin \theta_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4.3)$$

$$T_3^2 = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & 0 \\ \sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4.4)$$

$$T_4^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4.5)$$

Entonces realizando las correspondientes operaciones tenemos:

$$T_4^0 = \begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 & a_2 \cos(\theta_1 + \theta_2) + a_1 \cos(\theta_1) \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 & a_2 \sin(\theta_1 + \theta_2) + a_1 \sin(\theta_1) \\ 0 & 0 & 1 & d_1 + d_2 + d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4.6)$$

y de acuerdo con (2.4.1) la ecuación (2.4.7) corresponde a la cinemática directa de nuestro manipulador para obtener la posición y orientación del efector final.

$$\begin{bmatrix} n_x & o_x & a_x & Px \\ n_y & o_y & a_y & Py \\ n_z & o_z & a_z & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 & a_2 \cos(\theta_1 + \theta_2) + a_1 \cos(\theta_1) \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 & a_2 \sin(\theta_1 + \theta_2) + a_1 \sin(\theta_1) \\ 0 & 0 & 1 & d_1 + d_2 + d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4.7)$$

2.5 Obtención del modelo de cinemática inversa.

La solución de la cinemática inversa implica conocer los valores de cada articulación para que el robot alcance una posición y orientación deseada. Se deben encontrar las ecuaciones que definen la cinemática inversa para la posición final, la cual es donde se colocará el efector del robot.

De la cinemática directa dada en (2.4.7) se observan acoplamientos angulares, esto hace muy complicado encontrar expresiones explícitas para encontrar los valores articulares a partir de posición y orientación deseadas. Para encontrar expresiones adicionales que permitan resolver acoplamientos entre ángulos, se pre-multiplica la expresión T_n^0 por matrices inversas para encontrar las ecuaciones de la cinemática inversa. Esto hace que la parte derecha de la ecuación esté libre de ángulos individuales, permitiendo encontrar elementos en términos de funciones trigonométricas, y de ahí la posibilidad de encontrar ángulos.

Entonces definimos como matriz del lado derecho:

$$MLD = T_4^0 = T_1^0 T_2^1 T_3^2 T_4^3 = \begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 & a_2 \cos(\theta_1 + \theta_2) + a_1 \cos(\theta_1) \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 & a_2 \sin(\theta_1 + \theta_2) + a_1 \sin(\theta_1) \\ 0 & 0 & 1 & d_1 + d_2 + d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5.1)$$

la cual corresponde a la matriz de transformación total.

La matriz de orientación y posición deseada del robot es,

$$T_d = \begin{bmatrix} n_x & o_x & a_x & Px \\ n_y & o_y & a_y & Py \\ n_z & o_z & a_z & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5.2)$$

la cual contiene la matriz de orientación, posición y perspectiva. Se igualan las matrices y se pre-multiplican por las matrices inversas para deducir las matrices del lado derecho de acuerdo con (2.5.3).

$$\begin{aligned} T_d &= MLD &= T_1^0 \times T_2^1 \times T_3^2 \times T_4^3 & (2.5.3) \\ (T_1^{0-1}) \times T_d &= (T_1^{0-1}) \times (T_1^0 \times T_2^1 \times T_3^2 \times T_4^3) = T_2^1 \times T_3^2 \times T_4^3 \\ (T_2^{1-1}) \times (T_1^{0-1} \times T_d) &= (T_2^{1-1}) \times (T_2^1 \times T_3^2 \times T_4^3) = T_3^2 \times T_4^3 \\ (T_3^{2-1}) \times (T_2^{1-1} \times T_1^{0-1} \times T_d) &= (T_3^{2-1}) \times (T_3^2 \times T_4^3) = T_4^3 \\ (T_4^{3-1}) \times (T_3^{2-1} \times T_2^{1-1} \times T_1^{0-1} \times T_d) &= T_4^{3-1} \times T_4^3 = I \end{aligned}$$

Se obtienen todas las matrices del lado derecho de la ecuación (2.5.3):

$$T_4^0 = \begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 & a_2 \cos(\theta_1 + \theta_2) + a_1 \cos(\theta_1) \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 & a_2 \sin(\theta_1 + \theta_2) + a_1 \sin(\theta_1) \\ 0 & 0 & 1 & d_1 + d_2 + d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5.4)$$

$$T_4^1 = \begin{bmatrix} \cos(\theta_2 + \theta_3) & -\sin(\theta_2 + \theta_3) & 0 & a_2 \cos(\theta_2) \\ \sin(\theta_2 + \theta_3) & \cos(\theta_2 + \theta_3) & 0 & a_2 \sin(\theta_2) \\ 0 & 0 & 1 & d_2 + d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5.5)$$

$$T_4^2 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5.6)$$

$$T_4^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5.7)$$

$$T_4^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5.8)$$

después se calculan las matrices del lado izquierdo,

$$T_d = \begin{bmatrix} n_x & o_x & a_x & Px \\ n_y & o_y & a_y & Py \\ n_z & o_z & a_z & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5.9)$$

$$T_d^1 = \begin{bmatrix} n_x C_1 + n_y S_1 & o_x C_1 + o_y S_1 & a_x C_1 + a_y S_1 & Px.C_1 + Py.S_1 - a_1 \\ n_y C_1 - n_x S_1 & o_y C_1 - o_x S_1 & a_y C_1 - a_x S_1 & Py.C_1 - Px.S_1 \\ n_z & o_z & a_z & Pz - d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5.10)$$

$$T_d^2 = \begin{bmatrix} n_x C_{12} + n_y S_{12} & o_x C_{12} + o_y S_{12} & a_x C_{12} + a_y S_{12} & Px.C_{12} + Py.S_{12} - a_1 C_2 - a_2 \\ n_y C_{12} - n_x S_{12} & o_y C_{12} - o_x S_{12} & a_y C_{12} - a_x S_{12} & Py.C_{12} - Px.S_{12} + a_1 S_2 \\ n_z & o_z & a_z & Pz - d_1 - d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5.11)$$

$$T_d^{3T} = \begin{bmatrix} n_x C_{123} + n_y S_{123} & n_y C_{123} - n_x S_{123} & n_z & 0 \\ o_x C_{123} + o_y S_{123} & o_y C_{123} - o_x S_{123} & o_z & 0 \\ a_x C_{123} + a_y S_{123} & a_x C_{123} + a_y S_{123} & a_z & 0 \\ Px.C_{123} + Py.S_{123} - a_1 C_{23} - a_2 C_3 & Py.C_{123} - Px.S_{123} + a_1 S_{23} + a_2 S_3 & Pz - d_1 - d_2 & 1 \end{bmatrix} \quad (2.5.12)$$

$$T_d^{4T} = \begin{bmatrix} n_x C_{123} + n_y S_{123} & n_y C_{123} - n_x S_{123} & n_z & 0 \\ o_x C_{123} + o_y S_{123} & o_y C_{123} - o_x S_{123} & o_z & 0 \\ a_x C_{123} + a_y S_{123} & a_y C_{123} - a_x S_{123} & a_z & 0 \\ Px.C_{123} + Py.S_{123} - a_1 C_{23} - a_2 C_3 & Py.C_{123} - Px.S_{123} + a_1 S_{23} + a_2 S_3 & Pz - d_1 - d_2 - d_4 & 1 \end{bmatrix} \quad (2.5.13)$$

y se igualan las ecuaciones de acuerdo con (2.5.3).

$$T_d = T_4^0 \quad (2.5.14)$$

$$T_d^1 = T_4^1 \quad (2.5.15)$$

$$T_d^2 = T_4^2 \quad (2.5.16)$$

$$T_d^3 = T_4^3 \quad (2.5.17)$$

$$T_d^4 = T_4^4 \quad (2.5.18)$$

De (2.5.14) los elementos (1,4) y (2,4),

$$Px = a_2 \cos(\theta_1 + \theta_2) + a_1 \cos(\theta_1) \quad (2.5.19)$$

$$Py = a_2 \sin(\theta_1 + \theta_2) + a_1 \sin(\theta_1) \quad (2.5.20)$$

se elevan al cuadrado y se suman de acuerdo con el teorema de Pitágoras:

$$Px^2 = (a_2 \cos(\theta_1 + \theta_2) + a_1 \cos(\theta_1))^2 \quad (2.5.21)$$

$$Py^2 = (a_2 \sin(\theta_1 + \theta_2) + a_1 \sin(\theta_1))^2 \quad (2.5.22)$$

$$Px^2 + Py^2 = a_1^2 (\cos^2 \theta_1 + \sin^2 \theta_1) + 2a_1 a_2 (\cos \theta_1 \cos(\theta_1 + \theta_2) + \sin \theta_1 \sin(\theta_1 + \theta_2)) + a_2^2 (\cos^2(\theta_1 + \theta_2) + \sin^2(\theta_1 + \theta_2)) \quad (2.5.23)$$

Utilizando las identidades trigonométricas (2.5.24), (2.5.25) y (2.5.26) se simplifica (2.5.23) y se obtiene (2.5.27).

$$\cos^2 \theta + \sin^2 \theta = 1 \quad (2.5.24)$$

$$\cos A \cos B = \frac{1}{2} (\cos(A + B) + \cos(A - B)) \quad (2.5.25)$$

$$\sin A \sin B = -\frac{1}{2} (\cos(A + B) - \cos(A - B)) \quad (2.5.26)$$

$$\cos \theta_2 = \frac{1}{2a_1a_2} (Px^2 + Py^2 - a_1^2 - a_2^2) \quad (2.5.27)$$

Se despeja el término $\sin \theta_2$ de (2.5.24):

$$\sin \theta_2 = \pm \sqrt{1 - \cos^2 \theta_2} \quad (2.5.28)$$

y se obtiene la ecuación de cinemática inversa para el ángulo θ_2 .

$$\theta_2 = \tan^{-1} \frac{\sin \theta_2}{\cos \theta_2} \quad (2.5.29)$$

Para encontrar el ángulo θ_1 se considera la Fig. 2.6, que es una vista desde arriba, en el plano XY de los dos eslabones de longitud a_1 y a_2 :

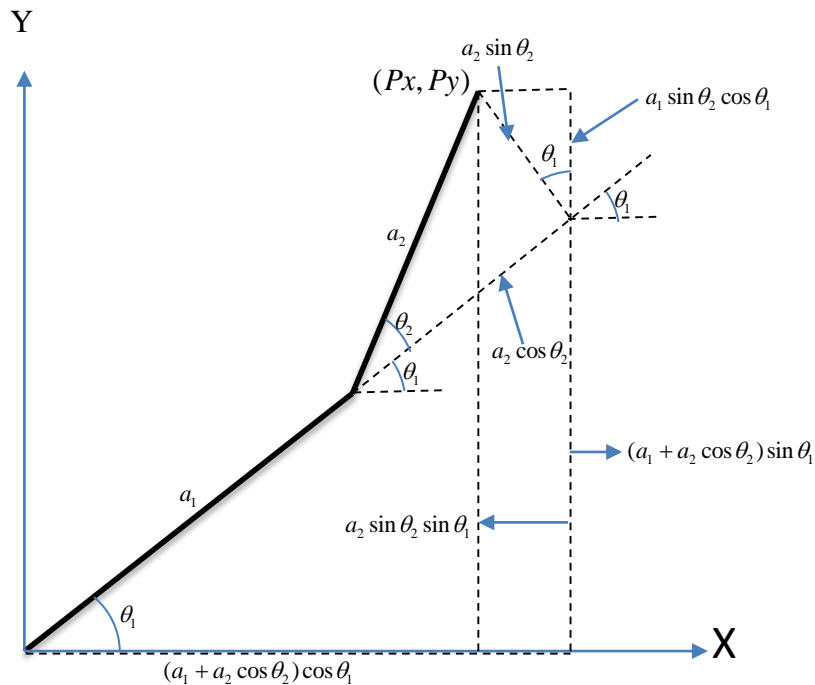


Fig. 2.6: Vista del robot desde el plano XY.

de las igualdades (2.5.21) y (2.5.22), y con las ecuaciones obtenidas de Fig. 2.6, se resuelve mediante la regla de *Cramer*.

$$Px = (a_1 + a_2 \cos \theta_2) \cos \theta_1 - a_2 \sin \theta_2 \sin \theta_1 \quad (2.5.30)$$

$$Py = a_2 \sin \theta_2 \cos \theta_1 + (a_1 + a_2 \cos \theta_2) \sin \theta_1 \quad (2.5.31)$$

Dicho procedimiento se soluciona mediante la obtención de determinantes de matrices:

$$\Delta = \begin{bmatrix} (a_1 + a_2 \cos \theta_2) & -a_2 \sin \theta_2 \\ a_2 \sin \theta_2 & (a_1 + a_2 \cos \theta_2) \end{bmatrix} = (a_1 + a_2 \cos \theta_2)^2 + (a_2 \sin \theta_2)^2 \quad (2.5.32)$$

$$\Delta \sin \theta_1 = \begin{bmatrix} (a_1 + a_2 \cos \theta_2) & Px \\ a_2 \sin \theta_2 & Py \end{bmatrix} = (a_1 + a_2 \cos \theta_2)Py - a_2 \sin \theta_2 Px \quad (2.5.33)$$

$$\Delta \cos \theta_1 = \begin{bmatrix} Px & -a_2 \sin \theta_2 \\ Py & (a_1 + a_2 \cos \theta_2) \end{bmatrix} = (a_1 + a_2 \cos \theta_2)Px + a_2 \sin \theta_2 Py \quad (2.5.34)$$

$$\sin \theta_1 = \frac{\Delta \sin \theta_1}{\Delta} = \frac{(a_1 + a_2 \cos \theta_2)Py - a_2 \sin \theta_2 Px}{Px^2 + Py^2} \quad (2.5.35)$$

$$\cos \theta_1 = \frac{\Delta \cos \theta_1}{\Delta} = \frac{(a_1 + a_2 \cos \theta_2)Px + a_2 \sin \theta_2 Py}{Px^2 + Py^2} \quad (2.5.36)$$

entonces se obtiene la ecuación de cinemática inversa para el ángulo θ_1 .

$$\theta_1 = \tan^{-1} \frac{(a_1 + a_2 \cos \theta_2)Py - a_2 \sin \theta_2 Px}{(a_1 + a_2 \cos \theta_2)Px + a_2 \sin \theta_2 Py} \quad (2.5.37)$$

Para encontrar el ángulo θ_3 de la igualdad (2.5.16) se toman los elementos (1,1) y (2,1):

$$\cos(\theta_1 + \theta_2)n_x + \sin(\theta_1 + \theta_2)n_y = \cos \theta_3 \quad (2.5.38)$$

$$\cos(\theta_1 + \theta_2)n_y - \sin(\theta_1 + \theta_2)n_x = \sin \theta_3 \quad (2.5.39)$$

mediante propiedades trigonométricas se obtiene la ecuación de cinemática inversa par el ángulo θ_3 .

$$\theta_3 = \tan^{-1} \frac{\sin \theta_3}{\cos \theta_3} = \frac{\cos(\theta_1 + \theta_2)n_x + \sin(\theta_1 + \theta_2)n_y}{\cos(\theta_1 + \theta_2)n_y - \sin(\theta_1 + \theta_2)n_x} \quad (2.5.40)$$

Por último de (2.5.18) y del elemento (3,4) se obtiene la ecuación de cinemática inversa para el movimiento prismático del último eslabón:

$$d = P_z - d_1 - d_2 + 100 \quad (2.5.41)$$

2.6 Comprobación de cinemática inversa-directa.

Para comprobar la cinemática inversa y directa se utilizan los valores D-H de la Tabla 2.1, después se realiza la cinemática directa a partir de valores articulares, el resultado se evalúa en la cinemática inversa para ver que sean coincidentes, tal y como lo explica la Fig. 2.1.

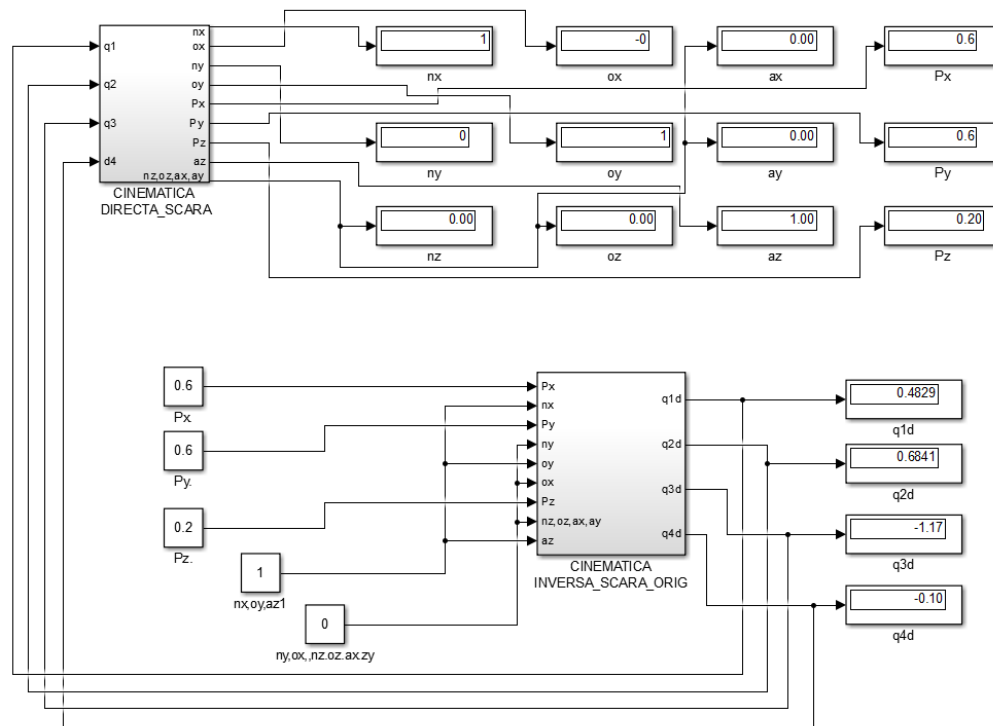


Fig. 2.7: Diagramas de bloque de cinemática inversa y directa.

En Fig. 2.7 se observa la evaluación con un punto en el espacio, y los resultados en la cinemática directa e inversa son coincidentes (Véase Apéndice A para el detalle de las

cinemáticas). En la Tabla 2.2 y Tabla 2.3 se muestra los resultados para dos coordenadas diferentes.

Tabla 2.2: Evaluación cinemática inversa.

Posición y orientación (Px,Py,Pz,nx,ox,ax,ny,oy,ay,nz,oz,az)	Valor articular (q1,q2,q3,q4)
(0.6,0.6,0.2,1,0,0,0,1,0,0,0,1)	(0.04829,0.6841,-1.17,-0.10)
(0.4,0.2,0.1,1,0,0,0,1,0,0,0,1)	(-0.4016, 2.124,-1.72,-0.20)

Ahora, el resultado de la cinemática inversa se evalúa en la cinemática directa.

Tabla 2.3: Evaluación cinemática directa.

Valor articular (q1,q2,q3,q4)	Posición y orientación (Px,Py,Pz,nx,ox,ax,ny,oy,ay,nz,oz,az)
(0.04829,0.6841,-1.17,-0.10)	(0.6,0.6,0.2,1,0,0,0,1,0,0,0,1)
(-0.4016, 2.124,-1.72,-0.20)	(0.4,0.2,0.1,1,0,0,0,1,0,0,0,1)

Se puede observar que los modelos cinemáticos son coincidentes y correctos.

2.7 Conclusión del capítulo.

En este capítulo se detalló la estructura D-H y se encontraron sus parámetros mediante el algoritmo propuesto con ayuda del software *Robotic Arm Kinematic GUI*, las ecuaciones que definen la cinemática directa (2.4.7), y las ecuaciones que definen la cinemática inversa (2.5.29) ,(2.5.37) ,(2.5.40) y (2.5.41), se comprobaron mediante una simulación en la que se obtuvo el mismo valor de referencia al pasar de cinemática inversa-directa-inversa.

3 Modelado dinámico.

En este capítulo se deduce el modelo dinámico del robot SCARA mediante la formulación Euler-Lagrange, se detalla su estructura y sus propiedades, y se lleva a cabo una simulación en lazo abierto para analizar su comportamiento.

3.1 Estructura del robot.

La configuración SCARA es un robot manipulador industrial normalmente de cuatro grados de libertad (GDL). Los robots SCARA se conocen por su velocidad, excelente repetitividad y su aplicación para tareas muy específicas tal como *pick & place* y ensamblaje de piezas. En general es un robot horizontal y es usado para el montaje de circuitos electrónicos y para el ensamblaje de productos. Su desplazamiento es a través de los tres ejes XYZ. Su base tiene un actuador rotativo en la parte superior, los dos siguientes eslabones están conectados a través de otro actuador rotativo y el último elemento es un desplazamiento lineal en el eje Z, el efector final tiene un giro para dar orientación al objeto manipulado y este es un actuador rotativo. Se le puede nombrar también como un robot serial RRRP. Se muestra en la Fig. 3.2 su configuración estructural. La principal ventaja de este tipo de robot, es que la fuerza de gravedad solo se ejerce en el desplazamiento vertical, es por eso que su principal aplicación son las tareas de *pick & place*.

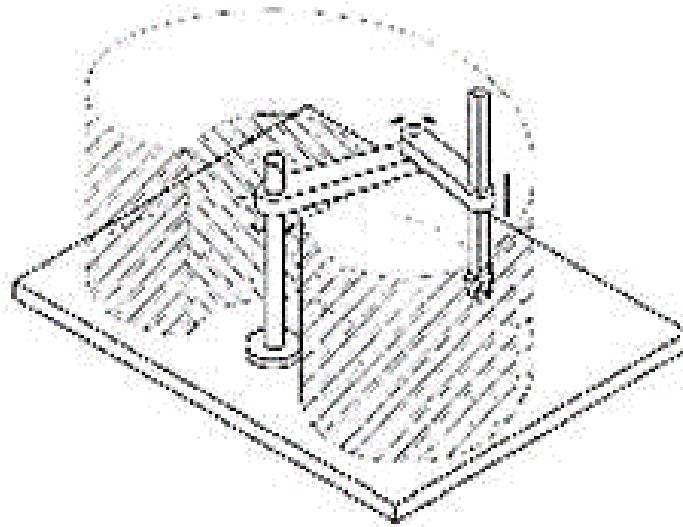


Fig. 3.1: Espacio de trabajo del robot SCARA 4GDL.

El área de trabajo del robot SCARA, describe la forma de un cilindro como lo muestra la Fig. 3.1.

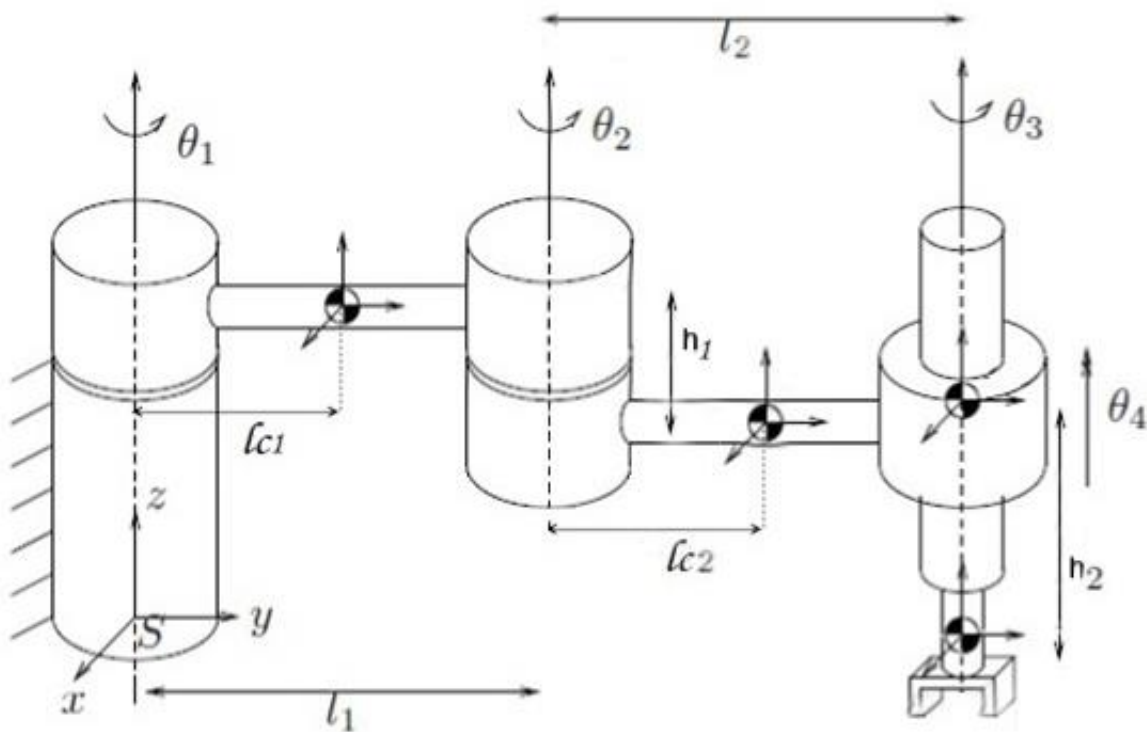


Fig. 3.2: Estructura física del robot SCARA.

El modelado del robot se basa en la Fig. 3.2 y se hacen las siguientes suposiciones:

1. Los movimientos θ_1 , θ_2 , θ_3 , se desarrollan en el plano horizontal, por lo cual se desprecia la energía potencial U en estos elementos.
2. En el análisis del nodo 3, se considera que esta articulación mueve la masa 3 y la masa 4, así como también existe una suma de las inercias I_3 e I_4 .
3. No existe disipación en los elementos actuadores.
4. El sistema es completamente actuado.

Los parámetros para este robot son:

Tabla 3.1: Parámetros del Robot SCARA

m_1	15kg
m_2	12kg
m_3	3kg
m_4	3kg
I_1	$0.02087 m_1 \text{ m}^2$
I_2	$0.08 m_2 \text{ m}^2$
I_3	0.05 kg m^2
I_4	$0.02 m_4 \text{ m}^2$
l_1	0.50m
l_2	0.40m
lc_1	0.25m
lc_2	0.20m
h_1	0.10m
h_2	0.10m

3.2 Obtención del modelo en Euler-Lagrange.

El modelado bajo el formalismo de E-L se basa en definir la energía total del sistema, haciendo uso de los conceptos de la energía potencial y la energía cinética. Cada elemento o cadena cinemática del robot se define mediante su energía, y al final se realiza una suma para obtener la energía total del sistema. Se ha establecido una serie de pasos para obtener el modelo bajo este formalismo.

Primeramente se establecen las coordenadas generalizadas q para cada uno de los grados de libertad y se obtiene la energía cinética de todo el sistema,

$$K(q, \dot{q}) = K_1(q, \dot{q}) + K_2(q, \dot{q}) + \dots + K_i(q, \dot{q}) \quad (3.2.1)$$

y la energía potencial,

$$U(q) = U_1(q) + U_n(q) + \dots + U_n(q) \quad (3.2.2)$$

se obtiene el Lagrangiano el cual está definido por

$$L(q, \dot{q}) = K(q, \dot{q}) - U(q) \quad (3.2.3)$$

el cuál es la diferencia entre la energía cinética menos la energía potencial. Por último se desarrolla la ecuación de movimiento de E-L la cual está dada por:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_i} \right) - \frac{\partial L}{\partial \theta_i} + \frac{\partial F}{\partial \dot{\theta}_i} = u_i \quad (3.2.4)$$

donde $i=1,2,\dots, n$, u_i es el par, y $\frac{\partial F}{\partial \dot{\theta}_i}$, es la ecuación de disipación de Rayleigh.

Se desarrollan las n ecuaciones diferenciales que representan a cada una de las cadenas cinemáticas del robot, en este caso $n = 4$.

La energía cinética se establece para cada uno de los eslabones con la siguiente fórmula:

$$K_i(q, \dot{q}) = \frac{1}{2} m_i v_i^2 + \frac{1}{2} I_i \dot{q}_i^2 \quad (3.2.5)$$

donde: m = masa, v = velocidad, I = inercia y q = coordenada generalizada.

El primer sumando representa la energía cinética traslacional y el segundo corresponde a la energía cinética rotacional. La energía potencial para cada eslabón se define como:

$$U_i(q) = m_i g h_i \quad (3.2.6)$$

donde: g = gravedad y h = altura.

Dicho esto, se procede a realizar los pasos antes descritos para cada eslabón.

Eslabón 1:

Se establece la energía cinética rotacional y traslacional,

$$K_1 = \frac{1}{2} m_1 v_1^2 + \frac{1}{2} I_1 \dot{\theta}_1^2 \quad (3.2.7)$$

se requiere la velocidad traslacional definida por:

$$v_1^2 = \dot{x}_1^2 + \dot{y}_1^2 \quad (3.2.8)$$

donde,

$$x_1 = l c_1 \cos \theta_1 \quad (3.2.9)$$

$$y_1 = l c_1 \sin \theta_1 \quad (3.2.10)$$

se derivan cada una y se elevan al cuadrado,

$$\dot{x}_1 = -l c_1 \sin \theta_1 \dot{\theta}_1 \quad (3.2.11)$$

$$\dot{y}_1 = l c_1 \cos \theta_1 \dot{\theta}_1 \quad (3.2.12)$$

$$\dot{x}_1^2 = l c_1^2 \sin^2 \theta_1 \dot{\theta}_1^2 \quad (3.2.13)$$

$$\dot{y}_1^2 = l c_1^2 \cos^2 \theta_1 \dot{\theta}_1^2 \quad (3.2.14)$$

$$v_1^2 = \dot{x}_1^2 + \dot{y}_1^2 = l c_1^2 \sin^2 \theta_1 \dot{\theta}_1^2 + l c_1^2 \cos^2 \theta_1 \dot{\theta}_1^2 = l c_1^2 (\sin^2 \theta_1 + \cos^2 \theta_1) \dot{\theta}_1^2 \quad (3.2.15)$$

recordar que $(\sin^2 \theta_1 + \cos^2 \theta_1) = 1$ para simplificar (3.2.15).

$$v_1^2 = l c_1^2 \dot{\theta}_1^2 \quad (3.2.16)$$

Por lo que al sustituir (3.2.16) en (3.2.7) se obtiene la energía cinética del eslabón 1:

$$K_1 = \frac{1}{2} m_1 l c_1^2 \dot{\theta}_1^2 + \frac{1}{2} I_1 \dot{\theta}_1^2 \quad (3.2.17)$$

Eslabón 2:

Se define la energía cinética como:

$$K_2 = \frac{1}{2} m_2 v_2^2 + \frac{1}{2} I_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \quad (3.2.18)$$

se obtiene la velocidad de traslación por medio de,

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 \quad (3.2.19)$$

$$x_2 = l_1 \cos \theta_1 + lc_2 \cos(\theta_1 + \theta_2) \quad (3.2.20)$$

$$y_2 = l_1 \sin \theta_1 + lc_2 \sin(\theta_1 + \theta_2) \quad (3.2.21)$$

se deriva (3.2.20) y (3.2.21) con respecto al tiempo y se obtiene,

$$\dot{x}_2 = -l_1 \sin \theta_1 \dot{\theta}_1 - lc_2 \sin(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2) \quad (3.2.22)$$

$$\dot{y}_2 = l_1 \cos \theta_1 \dot{\theta}_1 + lc_2 \cos(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2) \quad (3.2.23)$$

se eleva al cuadrado las dos ecuaciones anteriores,

$$\dot{x}_2^2 = l_1^2 \sin^2 \theta_1 \dot{\theta}_1^2 + 2l_1lc_2 \sin \theta_1 \sin(\theta_1 + \theta_2) \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \quad (3.2.24)$$

$$+lc_2^2 \sin^2(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2)^2$$

$$\dot{y}_2^2 = l_1^2 \cos^2 \theta_1 \dot{\theta}_1^2 + 2l_1lc_2 \cos \theta_1 \cos(\theta_1 + \theta_2) \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \quad (3.2.25)$$

$$+lc_2^2 \cos^2(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2)^2$$

se sustituye (3.2.24) y (3.2.25) en (3.2.19),

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 \quad (3.2.26)$$

$$= l_1^2 (\sin^2 \theta_1 + \cos^2 \theta_1) \dot{\theta}_1^2 + lc_2^2 [\sin^2(\theta_1 + \theta_2) + \cos^2(\theta_1 + \theta_2)] (\dot{\theta}_1 + \dot{\theta}_2)^2$$

$$+ 2l_1lc_2 [\sin \theta_1 \sin(\theta_1 + \theta_2) + \cos \theta_1 \cos(\theta_1 + \theta_2)] \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2)$$

recordamos que $(\sin^2 \theta_1 + \cos^2 \theta_1) = 1$, y se toman las identidades trigonométricas

$\sin(a+b) = \sin a \cos b + \cos a \sin b$ y $\cos(a+b) = \cos a \cos b - \sin a \sin b$ para sustituir en (3.2.26).

$$\sin \theta_1 \sin(\theta_1 + \theta_2) + \cos \theta_1 \cos(\theta_1 + \theta_2) \quad (3.2.27)$$

$$= \sin \theta_1 [\sin \theta_1 \cos \theta_2 + \cos \theta_1 \sin \theta_2] + \cos \theta_1 [\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2]$$

$$= \sin^2 \theta_1 \cos \theta_2 + \sin \theta_1 \cos \theta_1 \sin \theta_2 + \cos^2 \theta_1 \cos \theta_2 - \sin \theta_1 \cos \theta_1 \sin \theta_2$$

$$= \cos \theta_2 (\sin^2 \theta_1 + \cos^2 \theta_1)$$

$$= \cos \theta_2$$

Por lo tanto,

$$v_2^2 = l_1^2 \dot{\theta}_1^2 + lc_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2l_1lc_2 \cos \theta_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \quad (3.2.28)$$

la energía cinética del eslabón dos se define como:

$$K_2 = \frac{1}{2} m_2 \left[l_1^2 \dot{\theta}_1^2 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2l_1 l_2 \cos \theta_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \right] + \frac{1}{2} I_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \quad (3.2.29)$$

Eslabón 3:

Se define la energía cinética del eslabón 3 como,

$$K_3 = \frac{1}{2} (m_3 + m_4) v_3^2 + \frac{1}{2} (I_3 + I_4) (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^2 \quad (3.2.30)$$

se obtiene la velocidad de traslación del eslabón 3 encontrando las componentes de velocidad,

$$v_3^2 = \dot{x}_3^2 + \dot{y}_3^2 \quad (3.2.31)$$

$$(3.2.32)$$

$$x_3 = l_1 \cos \theta_1 + l_2 \cos (\theta_1 + \theta_2)$$

$$y_3 = l_1 \sin \theta_1 + l_2 \sin (\theta_1 + \theta_2) \quad (3.2.33)$$

se deriva y eleva al cuadrado las ecuaciones (3.2.32) y (3.2.33),

$$\dot{x}_3 = -l_1 \sin \theta_1 \dot{\theta}_1 - l_2 \sin (\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2) \quad (3.2.34)$$

$$\dot{y}_3 = l_1 \cos \theta_1 \dot{\theta}_1 + l_2 \cos (\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2) \quad (3.2.35)$$

$$\dot{x}_3^2 = l_1^2 \sin^2 \theta_1 \dot{\theta}_1^2 + 2l_1 l_2 \sin \theta_1 \sin (\theta_1 + \theta_2) \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) + l_2^2 \sin^2 (\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2)^2 \quad (3.2.36)$$

$$\dot{y}_3^2 = l_1^2 \cos^2 \theta_1 \dot{\theta}_1^2 + 2l_1 l_2 \cos \theta_1 \cos (\theta_1 + \theta_2) \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \quad (3.2.37)$$

$$+ l_2^2 \cos^2 (\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2)^2$$

se sustituye (3.2.36) y (3.2.37) en (3.2.31) obtenemos,

$$v_3^2 = \dot{x}_3^2 + \dot{y}_3^2 \quad (3.2.38)$$

$$= l_1^2 (\sin^2 \theta_1 + \cos^2 \theta_1) \dot{\theta}_1^2 + l_2^2 [\sin^2 (\theta_1 + \theta_2) + \cos^2 (\theta_1 + \theta_2)] (\dot{\theta}_1 + \dot{\theta}_2)^2 \\ + 2l_1 l_2 [\sin \theta_1 \sin (\theta_1 + \theta_2) + \cos \theta_1 \cos (\theta_1 + \theta_2)] \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2)$$

se utilizan las identidades trigonométricas aplicadas en el análisis del eslabón 2 para simplificar (3.2.38),

$$v_3^2 = l_1^2 \dot{\theta}_1^2 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2l_1 l_2 \cos \theta_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \quad (3.2.39)$$

por lo tanto, se sustituye (3.2.39) en (3.2.30) y se obtiene la energía cinética del eslabón 3:

$$K_3 = \frac{1}{2}(m_3 + m_4) \left[l_1^2 \dot{\theta}_1^2 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2l_1 l_2 \cos \theta_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \right] \quad (3.2.40)$$

$$+ \frac{1}{2}(I_3 + I_4) (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^2$$

Eslabón 4:

Se define la energía cinética del eslabón 4 como:

$$K_4 = \frac{1}{2} m_4 v_4^2 = \frac{1}{2} m_4 \dot{\theta}_4^2 \quad (3.2.41)$$

Ahora se obtiene la energía potencial para cada uno de los eslabones, tomando en cuenta que en las tres primeras coordenadas no actúa la energía potencial.

$$U_1 = U_2 = U_3 = 0 \quad (3.2.42)$$

$$U_4 = -m_4 g \theta_4 \quad (3.2.43)$$

De acuerdo con la ecuación de Lagrange (3.2.3), se obtiene la suma de las energías cinéticas de cada eslabón (3.2.1), y la suma de las energías potencial es (3.2.2), por lo que se obtiene,

$$K = K_1 + K_2 + K_3 + K_4 \quad (3.2.44)$$

$$= \frac{1}{2} m_1 l c_1^2 \dot{\theta}_1^2 + \frac{1}{2} I_1 \dot{\theta}_1^2$$

$$+ \frac{1}{2} m_2 \left[l_1^2 \dot{\theta}_1^2 + l c_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2l_1 l c_2 \cos \theta_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \right] + \frac{1}{2} I_2 (\dot{\theta}_1 + \dot{\theta}_2)^2$$

$$+ \frac{1}{2} (m_3 + m_4) \left[l_1^2 \dot{\theta}_1^2 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2l_1 l_2 \cos \theta_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \right]$$

$$+ \frac{1}{2} (I_3 + I_4) (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^2 + \frac{1}{2} m_4 \dot{\theta}_4^2$$

$$U = U_1 + U_2 + U_3 + U_4 = -m_4 g \theta_4 \quad (3.2.45)$$

dado lo anterior, se obtiene la ecuación de Lagrange,

$$L = K - U \quad (3.2.46)$$

$$= \frac{1}{2} m_1 l c_1^2 \dot{\theta}_1^2 + \frac{1}{2} I_1 \dot{\theta}_1^2 + \frac{1}{2} m_2 \left[l_1^2 \dot{\theta}_1^2 + l c_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2l_1 l c_2 \cos \theta_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \right]$$

$$+ \frac{1}{2} I_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + \frac{1}{2} (m_3 + m_4) \left[l_1^2 \dot{\theta}_1^2 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2l_1 l_2 \cos \theta_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \right]$$

$$+ \frac{1}{2} (I_3 + I_4) (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^2 + \frac{1}{2} m_4 \dot{\theta}_4^2 + m_4 g \theta_4$$

y la función F, la cual corresponde a la disipación en cada uno de los elementos.

$$F = \frac{1}{2}B_1\dot{\theta}_1^2 + \frac{1}{2}B_2\dot{\theta}_2^2 + \frac{1}{2}B_3\dot{\theta}_3^2 + \frac{1}{2}B_4\dot{\theta}_4^2 \quad (3.2.47)$$

Se realizan las derivadas parciales correspondientes, en base al formulismo de Euler-Lagrange (3.2.4).

Para $i=1$:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} + \frac{\partial F}{\partial \dot{\theta}_1} = u_1 \quad (3.2.48)$$

Se obtienen cada uno de los elementos de la ecuación anterior,

$$\begin{aligned} \frac{\partial L}{\partial \dot{\theta}_1} = & m_1 l c_1^2 \dot{\theta}_1 + I_1 \dot{\theta}_1 + m_2 \left[l_1^2 \dot{\theta}_1 + l c_2^2 (\dot{\theta}_1 + \dot{\theta}_2) + l_1 l c_2 \cos \theta_2 \dot{\theta}_1 + l_1 l c_2 \cos \theta_2 (\dot{\theta}_1 + \dot{\theta}_2) \right] \quad (3.2.49) \\ & + I_2 (\dot{\theta}_1 + \dot{\theta}_2) + (m_3 + m_4) \left[l_1^2 \dot{\theta}_1 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2) + l_1 l_2 \cos \theta_2 \dot{\theta}_1 + l_1 l_2 \cos \theta_2 (\dot{\theta}_1 + \dot{\theta}_2) \right] \\ & + (I_3 + I_4) (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \end{aligned}$$

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_1} \right) = & m_1 l c_1^2 \ddot{\theta}_1 + I_1 \ddot{\theta}_1 + m_2 l_1^2 \ddot{\theta}_1 + m_2 l c_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l c_2 \cos \theta_2 \ddot{\theta}_1 \quad (3.2.50) \\ & - m_2 l_1 l c_2 \sin \theta_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 l_1 l c_2 \cos \theta_2 (\ddot{\theta}_1 + \ddot{\theta}_2) - m_2 l_1 l c_2 \sin \theta_2 (\dot{\theta}_1 + \dot{\theta}_2) \dot{\theta}_2 \\ & + I_2 (\ddot{\theta}_1 + \ddot{\theta}_2) + (m_3 + m_4) l_1^2 \ddot{\theta}_1 + (m_3 + m_4) l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + (m_3 + m_4) l_1 l_2 \cos \theta_2 \ddot{\theta}_1 \\ & - (m_3 + m_4) l_1 l_2 \sin \theta_2 \dot{\theta}_1 \dot{\theta}_2 + (m_3 + m_4) l_1 l_2 \cos \theta_2 (\ddot{\theta}_1 + \ddot{\theta}_2) \\ & - (m_3 + m_4) l_1 l_2 \sin \theta_2 (\dot{\theta}_1 + \dot{\theta}_2) \dot{\theta}_2 + (I_3 + I_4) (\ddot{\theta}_1 + \ddot{\theta}_2 + \ddot{\theta}_3) \end{aligned}$$

$$\frac{\partial L}{\partial \theta_1} = 0 \quad (3.2.51)$$

$$\frac{\partial F}{\partial \dot{\theta}_1} = B_1 \dot{\theta}_1 \quad (3.2.52)$$

se sustituyen cada una de ellas en la ecuación (3.2.48), y se factoriza obteniendo convenientemente,

$$\begin{aligned}
 & \ddot{\theta}_1 \left[I_a + m_1 l c_1^2 + m_2 l_1^2 + m_2 (l c_2^2 + 2 l_1 l c_2 \cos \theta_2) + (m_3 + m_4) (l_1^2 + l_2^2 + 2 l_1 l_2 \cos \theta_2) \right] \\
 & + \ddot{\theta}_2 \left[I_2 + I_3 + I_4 + m_2 (l c_2^2 + l_1 l c_2 \cos \theta_2) + (m_3 + m_4) (l_2^2 + l_1 l_2 \cos \theta_2) \right] \\
 & + \ddot{\theta}_3 (I_3 + I_4) + \dot{\theta}_1 \left[-m_2 l_1 l c_2 \sin \theta_2 \dot{\theta}_2 - (m_3 + m_4) l_1 l_2 \sin \theta_2 \dot{\theta}_2 \right] \\
 & + \dot{\theta}_2 \left[-m_2 l_1 l c_2 \sin \theta_2 (\dot{\theta}_1 + \dot{\theta}_2) - (m_3 + m_4) l_1 l_2 \sin \theta_2 (\dot{\theta}_1 + \dot{\theta}_2) \right] + B_1 \dot{\theta}_1 = u_1
 \end{aligned} \tag{3.2.53}$$

donde $I_a = I_1 + I_2 + I_3 + I_4$.

Para $i=2$:

Se realiza el mismo procedimiento que para $i=1$,

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} + \frac{\partial F}{\partial \dot{\theta}_2} = u_2 \tag{3.2.54}$$

$$\begin{aligned}
 \frac{\partial L}{\partial \dot{\theta}_2} &= m_2 \left[l c_2^2 (\dot{\theta}_1 + \dot{\theta}_2) + l_1 l c_2 \cos \theta_2 \dot{\theta}_1 \right] + I_2 (\dot{\theta}_1 + \dot{\theta}_2) \\
 &+ (m_3 + m_4) \left[l_2^2 (\dot{\theta}_1 + \dot{\theta}_2) + l_1 l_2 \cos \theta_2 \dot{\theta}_1 \right] + (I_3 + I_4) (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)
 \end{aligned} \tag{3.2.55}$$

$$\begin{aligned}
 \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_2} \right) &= m_2 l c_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l c_2 \cos \theta_2 \ddot{\theta}_1 - m_2 l_1 l c_2 \sin \theta_2 \dot{\theta}_1 \dot{\theta}_2 + I_2 (\ddot{\theta}_1 + \ddot{\theta}_2) \\
 &+ (m_3 + m_4) l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + (m_3 + m_4) l_1 l_2 \cos \theta_2 \ddot{\theta}_1 - (m_3 + m_4) l_1 l_2 \sin \theta_2 \dot{\theta}_1 \dot{\theta}_2 \\
 &+ (I_3 + I_4) (\ddot{\theta}_1 + \ddot{\theta}_2 + \ddot{\theta}_3)
 \end{aligned} \tag{3.2.56}$$

$$\frac{\partial L}{\partial \theta_2} = -m_2 l_1 l c_2 \sin \theta_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) - (m_3 + m_4) l_1 l_2 \sin \theta_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \tag{3.2.57}$$

$$\frac{\partial F}{\partial \dot{\theta}_2} = B_2 \dot{\theta}_2 \tag{3.2.58}$$

se factoriza para cada uno de los elementos,

$$\begin{aligned}
 & \ddot{\theta}_1 \left[I_2 + I_3 + I_4 + m_2 (l c_2^2 + l_1 l c_2 \cos \theta_2) + (m_3 + m_4) (l_2^2 + l_1 l_2 \cos \theta_2) \right] \\
 & + \ddot{\theta}_2 \left[I_2 + I_3 + I_4 + m_2 l c_2^2 + (m_3 + m_4) l_2^2 \right] + \ddot{\theta}_3 (I_3 + I_4) \\
 & + \dot{\theta}_1 \left[\begin{aligned} & -m_2 l_1 l c_2 \sin \theta_2 \dot{\theta}_2 - (m_3 + m_4) l_1 l_2 \sin \theta_2 \dot{\theta}_2 + m_2 l_1 l c_2 \sin \theta_2 \dot{\theta}_1 \\ & + m_2 l_1 l c_2 \sin \theta_2 \dot{\theta}_2 + (m_3 + m_4) l_1 l_2 \sin \theta_2 \dot{\theta}_1 + (m_3 + m_4) l_1 l_2 \sin \theta_2 \dot{\theta}_2 \end{aligned} \right] \\
 & + B_2 \dot{\theta}_2 = u_2
 \end{aligned} \tag{3.2.59}$$

Para $i=3$:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_3} \right) - \frac{\partial L}{\partial \theta_3} + \frac{\partial F}{\partial \dot{\theta}_3} = u_3 \quad (3.2.60)$$

$$\frac{\partial L}{\partial \dot{\theta}_3} = (I_3 + I_4) (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \quad (3.2.61)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_3} \right) = (I_3 + I_4) (\ddot{\theta}_1 + \ddot{\theta}_2 + \ddot{\theta}_3) \quad (3.2.62)$$

$$\frac{\partial L}{\partial \theta_3} = 0 \quad (3.2.63)$$

$$\frac{\partial F}{\partial \dot{\theta}_3} = B_3 \dot{\theta}_3 \quad (3.2.64)$$

se factoriza para cada uno de los elementos,

$$\ddot{\theta}_1 (I_3 + I_4) + \ddot{\theta}_2 (I_3 + I_4) + \ddot{\theta}_3 (I_3 + I_4) + B_3 \dot{\theta}_3 = u_3 \quad (3.2.65)$$

Para $i=4$:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_4} \right) - \frac{\partial L}{\partial \theta_4} + \frac{\partial F}{\partial \dot{\theta}_4} = u_4 \quad (3.2.66)$$

$$\frac{\partial L}{\partial \dot{\theta}_4} = m_4 \dot{\theta}_4 \quad (3.2.67)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_4} \right) = m_4 \ddot{\theta}_4 \quad (3.2.68)$$

$$\frac{\partial L}{\partial \theta_4} = m_4 g \quad (3.2.69)$$

$$\frac{\partial F}{\partial \dot{\theta}_4} = B_4 \dot{\theta}_4 \quad (3.2.70)$$

se factoriza para cada uno de los elementos,

$$\ddot{\theta}_4 m_4 + m_4 g \theta_4 + B_4 \dot{\theta}_4 = u_4 \quad (3.2.71)$$

de esta forma se obtienen, todas las ecuaciones diferenciales de E-L para el sistema.

Las ecuaciones (3.2.53), (3.2.59), (3.2.65) y (3.2.71) son las ecuaciones de movimiento que describen el comportamiento del robot SCARA. Representaremos el sistema en la forma matricial de E-L el cual tiene la siguiente estructura:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + B(\dot{q}) + gv(q) = u \quad (3.2.72)$$

donde $M(q)$ es la matriz de inercia de $n \times n$ dimensión, $C(q, \dot{q})$ es la matriz de coriolis de $n \times n$ dimensión, $B(\dot{q})$ es la matriz de disipación, $gv(q)$ es la matriz de gravedad y es de $n \times 1$ dimensión y u es el vector de entrada de $n \times 1$ dimensión (también conocido como vector τ). Recordemos que las coordenadas generalizadas están dadas por $q_1 = \theta_1$, $q_2 = \theta_2$, $q_3 = \theta_3$ y $q_4 = \theta_4$ el cual es un movimiento lineal.

Dicho lo anterior se llega a la representación matricial de las ecuaciones de movimiento de E-L del robot SCARA de 4 GDL:

$$\begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{12} & M_{22} & M_{23} & M_{24} \\ M_{13} & M_{23} & M_{33} & M_{34} \\ M_{14} & M_{24} & M_{34} & M_{44} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \\ C_{41} & C_{42} & C_{43} & C_{44} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} + \begin{bmatrix} B_1 \dot{\theta}_1 \\ B_2 \dot{\theta}_2 \\ B_3 \dot{\theta}_3 \\ B_4 \dot{\theta}_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ m_4 g \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \quad (3.2.73)$$

donde:

$$M_{11} = I_1 + I_2 + I_3 + I_4 + m_1 l_1^2 + m_2 l_1^2 + m_2 (l_2^2 + 2l_1 l_2 \cos \theta_2) \quad (3.2.74)$$

$$+(m_3 + m_4)(l_1^2 + l_2^2 + 2l_1 l_2 \cos \theta_2) \quad (3.2.75)$$

$$M_{12} = I_2 + I_3 + I_4 + m_2 (l_2^2 + l_1 l_2 \cos \theta_2) + (m_3 + m_4)(l_2^2 + l_1 l_2 \cos \theta_2)$$

$$M_{13} = M_{23} = M_{33} = I_3 + I_4 \quad (3.2.76)$$

$$M_{14} = M_{24} = M_{34} = 0 \quad (3.2.77)$$

$$M_{22} = I_2 + I_3 + I_4 + m_2 l_2^2 + (m_3 + m_4) l_2^2 \quad (3.2.78)$$

$$M_{44} = m_4 \quad (3.2.79)$$

$$C_{11} = -m_2 l_1 l_2 \sin \theta_2 \dot{\theta}_2 - (m_3 + m_4) l_1 l_2 \sin \theta_2 \dot{\theta}_2 \quad (3.2.80)$$

$$C_{12} = -m_2 l_1 l_2 \sin \theta_2 (\dot{\theta}_1 + \dot{\theta}_2) - (m_3 + m_4) l_1 l_2 \sin \theta_2 (\dot{\theta}_1 + \dot{\theta}_2) \quad (3.2.81)$$

$$C_{21} = m_2 l_1 l_2 \sin \theta_2 \dot{\theta}_1 + (m_3 + m_4) l_1 l_2 \sin \theta_2 \dot{\theta}_1 \quad (3.2.82)$$

$$C_{13} = C_{14} = C_{22} = C_{23} = C_{24} = C_{31} = C_{32} = C_{33} = C_{34} = C_{41} = C_{42} = C_{43} = C_{44} = 0 \quad (3.2.83)$$

3.3 Propiedades de la estructura Euler-Lagrange.

Las propiedades que se presentan a continuación, son en general aplicables para los modelos en E-L que describen a los robots seriales completamente actuados, mediante la forma de la ecuación (3.2.72).

La ecuación dinámica (3.2.72) es no lineal respecto al vector de estado $\begin{bmatrix} q^T & \dot{q}^T \end{bmatrix}^T$ y sus componentes matriciales tienen las siguientes propiedades:

Propiedades de la matriz de Inercia $M(q)$.

La matriz de inercia $M(q) \in \mathbb{R}^n$ cumple con las siguientes propiedades:

Propiedad 1. Es simétrica y definida positiva.

Propiedad 2. Es acotada superior e inferiormente. Es decir existen dos escalares tal que:

$$\mu_1 I < M(q) < \mu_2 I \quad (3.3.1)$$

Donde μ_1 y μ_2 son escalares, e I es la matriz identidad. Para su cálculo se requiere del uso de los valores propios máximo y mínimo de la matriz de inercia.

Propiedad 3. Es continuamente diferenciable.

Propiedad 4. Su inversa existe, es definida positiva y también acotada.

Matriz de Coriolis $C(q, \dot{q})$.

La matriz de fuerzas centrípetas o matriz de coriolis, junto con la matriz de inercia $M(q)$, cumple con la siguiente propiedad:

Propiedad 6. La matriz:

$$\dot{M}(q) - 2C(q, \dot{q}) = J \quad (3.3.2)$$

Es una matriz *skew-symmetric* (antisimétrica).

Y análogamente $\frac{1}{2}\dot{M}(q) - C(q, \dot{q})$ es una matriz antisimétrica, también resulta que $\dot{M}(q) = C(q, \dot{q}) + C(q, \dot{q})^T$. Independientemente de la manera en la que se obtenga $C(q, \dot{q})$, esta siempre satisface que,

$$\dot{q}^T \left[\frac{1}{2}\dot{M}(q) - C(q, \dot{q}) \right] \dot{q} = 0, \quad \forall q, \dot{q} \in \mathbb{R}^n \quad (3.3.3)$$

Propiedad de la matriz de gravedad $g_v(q)$.

El vector de pares gravitacionales $g_v(q)$ de $n \times 1$, depende solo de las posiciones articulares q . El vector está acotado si q también lo está, es decir existe una cota superior y una cota inferior si el sistema únicamente si está provisto de articulaciones rotatorias.

Propiedades de la matriz de fuerzas disipativas $B(\dot{q})$.

El término de fricción, dependiente de \dot{q} , tiene tres propiedades:

Propiedad 7. La fricción es un efecto local, así que los elementos de $B(\dot{q})$ son $f_i(\dot{q}_i); i = 1, \dots, n$.

Propiedad 8. Existe una función lineal para cada uno de los elementos de $B(\dot{q})$.

Propiedad 9. La fuerza de fricción es disipativa, es decir $\dot{q}^T B(\dot{q}) \geq 0$.

Esto da por resultado que la matriz

$$B(\dot{q}) = \begin{bmatrix} f_1 \dot{q}_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & f_n \dot{q}_n \end{bmatrix} \quad (3.3.4)$$

Es una matriz diagonal de $n \times n$ elementos. Cabe señalar que en el presente trabajo, no se toma en cuenta la fricción del sistema, por lo que $B_i = 0$, donde $i = 1, \dots, n$.

Propiedad de la matriz de entrada u .

El vector de entrada u , es la matriz la cual describe la forma en la que ingresa nuestro sistema, y si se cumple que $\dim(q) = \dim(u)$, entonces el sistema es completamente actuado.

3.4 Comparación del modelo obtenido con artículo de referencia (Voglewede, Smith, & Monti, 2009).

Se comparó el modelo obtenido en el capítulo 3.2 con el modelo que se presenta en el artículo (Voglewede, Smith, & Monti, 2009), siendo robots SCARA de diferentes parámetros, aun así se compara los movimientos del modelo y se presenta el movimiento del efector final.

Los parámetros de dicha simulación fueron los descritos en Tabla 3.1, donde los parámetros $m_i, I_i, i = 1, \dots, n$ son las masas e inercias, l_1 y l_2 son las longitudes del brazo 1 y del brazo 2, los parámetros lc_1 y lc_2 son las distancias de los centros de masa de los eslabones. El vector de entrada tiene escalones de diferentes magnitudes,

$$u = \tau = \begin{bmatrix} 100 \\ 100 \\ 10 \\ 10 \end{bmatrix} \quad (3.4.1)$$

como resultado tenemos la Fig. 3.3.

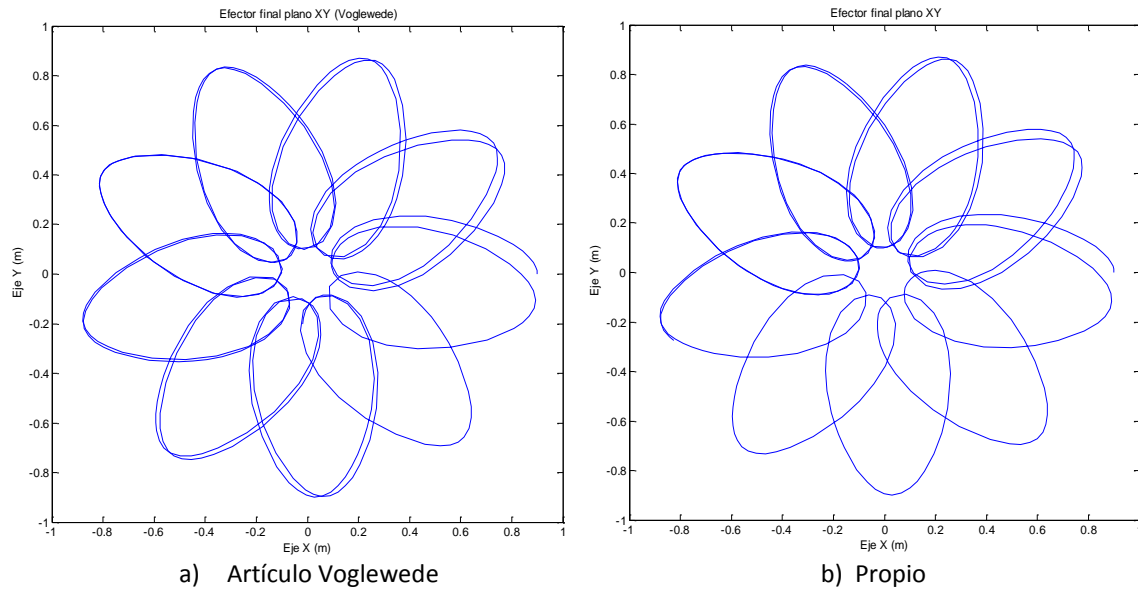


Fig. 3.3: Comparación del modelo obtenido vs artículo.

Cabe señalar que las consideraciones en estos modelos difieren, como por ejemplo el robot en el artículo es un RRPR y la forma en que las inercias tienen efecto en el sistema. Aun así, el comportamiento de estos dos modelos es muy similar y es válido para nuestros objetivos.

3.5 Representación del modelo en Hamiltoniano.

El modelo del robot mediante el formulismo de Hamilton es representado mediante ecuaciones diferenciales parciales, y se basa en la energía cinética y potencial al igual que el formulismo Lagrangiano. La principal diferencia es que, en Hamilton las variables generalizadas son q y p , las cuales corresponden al desplazamiento y momento generalizado, y existe una suma de las energías del sistema.

La expresión de Hamilton está dada por,

$$H(q, \dot{q}) = K(q, \dot{q}) + U(q) \quad (3.5.1)$$

donde $K(q, \dot{q})$ y $U(q)$ son la energía cinética y potencial respectivamente.

La energía cinética del sistema se representa por medio de (3.5.2).

$$K(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (3.5.2)$$

Al realizar el cambio de variables a las coordenadas generalizadas q y p , se tiene la relación,

$$\dot{q} = M^{-1}(q)p \quad (3.5.3)$$

entonces la ecuación (3.5.1) puede reescribirse como,

$$H(q, p) = \frac{1}{2} p^T M^{-1}(q)p + U(q) \quad (3.5.4)$$

donde $M(q)$ es la matriz de inercia del manipulador.

De acuerdo con (Ortega & Schaft, 2001), para caracterizar una clase de sistemas estabilizables con PBC y simplificar la solución de ecuaciones diferenciales parciales, necesitamos agregar más estructura a la dinámica del sistema, es decir, hacer explícitos los términos de amortiguamiento y dependencia de la función de energía. Es por eso que los sistemas PCH, son adecuados debido a que cubren un amplio rango de sistemas no lineales.

La forma estándar es,

$$\Sigma: \begin{cases} \dot{x} = [J(x) - R(x)] \frac{\partial H}{\partial x}(x) + g(x)u \\ y = g^T(x) \frac{\partial H}{\partial x}(x) \end{cases} \quad (3.5.5)$$

donde $H(x) \in \mathbb{R}$ es la función de energía del sistema en Hamilton dada por (3.5.4), $J(x) = -J^T(x) \in \mathbb{R}^{2n \times 2n}$ representa la estructura de interconexión, $R(x) \in \mathbb{R}^{2n \times 2n}$ es la matriz de disipación, $u, y \in \mathbb{R}^n$ son las entradas y salidas del sistema respectivamente, $g(x)$ es la función de acoplamiento entre el control y la dinámica del sistema. Nótese que el sistema es completamente actuado, así que u (en este caso corresponde a la entrada del manipulador τ) tiene la misma dimensión que y . Para sistemas mecánicos, el modelo PCH es descrito en términos de las coordenadas generalizadas:

$$x = [q \quad p]^T \quad (3.5.6)$$

El cual está compuesto por los términos generalizados $q = [q_1 \dots q_n]^T$ y las coordenadas generalizadas del momento $p = [p_1 \dots p_n]^T$. Por lo tanto el modelo PCH del manipulador toma la siguiente forma,

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \left\{ \begin{bmatrix} 0 & I_n \\ -I_n & D \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & R_n \end{bmatrix} \right\} \begin{bmatrix} \nabla_q H \\ \nabla_p H \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u \quad (3.5.7)$$

$$y = g^T(x) \nabla H \quad (3.5.8)$$

donde G es la matriz de entrada del sistema, $\nabla_q H = \frac{\partial H}{\partial q}$ y $\nabla_p H = \frac{\partial H}{\partial p}$, I_n es la matriz identidad, $D \in \mathbb{R}^{n \times n}$ es la matriz de amortiguamiento y R_n es la matriz de disipación del sistema (B).

Para nuestra aplicación, el sistema es completamente actuado, entonces $G = I_n$ y $D = 0$. Nótese que el sistema es afín al control porque $g(x) = [0 \ I_n]^T$. Por lo tanto el vector de salida contiene las velocidades generalizadas $y = \dot{q}$, la cual es llamada la nueva salida pasiva.

Para demostrar la equivalencia de un sistema PCH y el modelo dinámico E-L, realizamos unas sencillas operaciones en (3.5.7) obteniendo,

$$\begin{cases} \dot{q} = \nabla_p H = M^{-1}(q)p \\ \dot{p} = -\nabla_q H - R(\dot{q})\nabla_p H + u \end{cases} \quad (3.5.9)$$

e introducimos la derivada de (3.5.3),

$$\dot{p} = M(q)\ddot{q} + \dot{M}(q)\dot{q} \quad (3.5.10)$$

sustituyendo en la segunda línea de (3.5.9) y realizando algunas sustituciones tenemos,

$$M(q)\ddot{q} + \dot{M}(q)\dot{q} + \frac{\partial K}{\partial q}(q, \dot{q}) + \frac{\partial U}{\partial q}(q) + R(\dot{q})\dot{q} = u \quad (3.5.11)$$

donde podemos encontrar las siguientes relaciones,

$\dot{M}(q)\dot{q} + \frac{\partial K}{\partial q}(q, \dot{q}) = C(q, \dot{q})$ es la matriz de coriolis, $\frac{\partial U}{\partial q}(q) = gv(q)$ la matriz de gravedad y $R(\dot{q})\dot{q} = B(\dot{q})$ el vector de fricción (Paim, Neves, Moreno, & de Pieri, 2011).

Para ver en detalle el resultado de las ecuaciones y la función-S del modelo en PCH véase Anexo B.

3.6 Conclusión del capítulo.

En este capítulo se detalló la estructura física del robot, así como sus parámetros y consideraciones para la obtención del modelo dinámico en la estructura E-L dada por (3.2.73) y se mostraron algunas propiedades de dicho modelo. Se comparó el modelo obtenido con el modelo del artículo (Voglewede, Smith, & Monti, 2009), y se obtuvo la representación del modelo por medio de la estructura PCH (3.2.57).

4 Pasividad y diseño del controlador IDA-PBC.

La pasividad es una propiedad fundamental de los sistemas físicos en términos de disipación y transformación de la energía. Es una propiedad entrada-salida, y se refiere a la calidad y cantidad del balance de energía de un sistema, estimulado por entradas externas para generar alguna salida.

El término control basado en pasividad, fue introducido en (Ortega, 1989), para definir una metodología para el diseño de controladores, el cual logra la estabilización por pasividad. El objetivo de control es *pasivar* el sistema con una función de almacenamiento de energía, el cual tiene un mínimo en el punto de equilibrio deseado.

El control basado en pasividad, es una técnica bien conocida para sistemas mecánicos, ha demostrado ser muy fuerte en el diseño de controladores robustos y sistemas físicos descritos por la estructura E-L.

Esta idea ha sido exitosa para sistemas mecánicos simples que pueden ser estabilizados solamente moldeando la energía potencial. En este caso, el lazo cerrado sigue manteniendo el sistema en la estructura E-L, siendo la energía total la diferencia entre la energía almacenada y la energía suministrada, por lo tanto la estabilización puede ser explicada en términos de balance energético (Ortega & Schaft, 2001).

El PBC ha sido aplicado a sistemas físicos descritos por las ecuaciones de movimiento de E-L, el cual incluye sistemas electrónicos, mecánicos y electromecánicos. Desafortunadamente, este método no se puede utilizar adecuadamente cuando se necesita modificar la energía total del sistema o energía

cinética, en lazo cerrado se pierde la estructura de E-L y la función de almacenamiento de energía no tiene una interpretación lógica de la energía total.

Es por eso que se introduce una nueva metodología llamada *Interconnection and Damping Assignment Passivity-Based Control*, (Ortega, van der Schaft, Maschke, & Escobar, 2002), el cual se extiende a un amplio rango de sistemas que requieran un moldeo de la energía total.

Las principales características de este método son (Ortega, van der Schaft, Maschke, & Escobar, 2002):

- La estructura Hamiltoniana, se conserva en lazo cerrado.
- La función de almacenamiento es precisamente la función de la energía total del sistema en lazo cerrado.
- Además podemos dar condiciones para la disipación de la energía.

Hasta ahora, se parte de la formulación de E-L y lo sustituimos por la formulación PCH para sistemas no lineales.

Las principales distinciones entre el PBC clásico y el IDA-PBC (Ortega, van der Schaft, Maschke, & Escobar, 2002), es que en el primero seleccionamos la función de almacenamiento de energía y después diseñamos el controlador, asegurando el objetivo. Por otra parte, en IDA-PBC la función de energía en lazo cerrado es obtenida como resultado de nuestra elección de la interconexión y amortiguamiento deseado entre los subsistemas.

Los problemas de diseños de controladores son tradicionalmente vistos desde el punto de vista procesamiento de señales. Los objetivos de control son expresados en mantener una señal de error pequeña y reducir el efecto de las perturbaciones en el sistema, a pesar de la presencia de algunas dinámicas no modeladas.

De acuerdo con (Ortega, van der Schaft, Maschke, & Escobar, 2002), el análisis y diseño de controladores para sistemas no lineales es mejor cuando la planta a ser controlada y el controlador son vistos como dispositivos transformadores de energía (Fig. 4.1), descomponiéndolos en subsistemas en los cuales sus energías se suman después de la interconexión y determinan el comportamiento del sistema. El problema de control puede ser reformulado en orden de encontrar una dinámica del sistema y una interconexión estándar, tal que la función de la energía tome una

forma deseada. Este enfoque llamado *moldeo de energía* es la esencia del control basado en pasividad.

Como nuestro sistema es completamente actuado, el uso de la metodología PBC tiene un procedimiento natural para moldear solamente la energía potencial del sistema, resultando en un controlador con una interpretación física clara.

En el enfoque IDA-PBC se siguen dos pasos básicos (Ortega, van der Schaft, Maschke, & Escobar, 2002):

- Moldeo de energía: donde se modifica la función de energía total del sistema para asignar el punto de equilibrio deseado $(x^*, 0)$.
- Inyección de amortiguamiento: para alcanzar estabilidad asintótica y adicionalmente, para preservar la interpretación de energía del mecanismo de estabilización, se requiere que el sistema en lazo cerrado mantenga la forma Hamiltoniana.

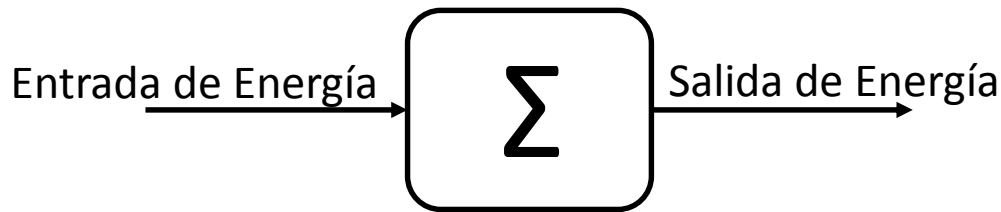


Fig. 4.1: Sistema visto como transformador de energía.

4.1 Preliminares

Considere el sistema no lineal afín al control,

$$\dot{x} = f(x) + g(x)u \quad (4.1.1)$$

$$y = h(x)$$

se asume la existencia de $g^\perp(x)$ tal que $g^\perp(x)g(x) = 0$, se define a $J_d(x) = -J_d^T(x)$, $R_d(x) = R_d^T(x) \geq 0$ como la matriz de interconexión deseada y la matriz de disipación deseada respectivamente, la función $H_d : \mathbb{R}^{2n} \rightarrow \mathbb{R}$ como la función Hamiltoniana deseada, e igualamos la primera línea de (4.1.1) con la primera línea de (3.5.5) tal que satisface la PDE,

$$g^\perp(x)f(x) = g^\perp(x)[J_d(x) - R_d(x)]\nabla H_d \quad (4.1.2)$$

y $H_d(x)$ es tal que,

$$x^* = \arg \min H_d(x) \quad (4.1.3)$$

con $x^* \in \mathbb{R}^{2n}$ un punto de equilibrio a ser estabilizado, entonces la dinámica en lazo cerrado del sistema (4.1.1), utilizando,

$$u(x) = [g^T(x)g(x)]^{-1}g^T(x)\{[J_d(x) - R_d(x)]\nabla H_d - f(x)\} \quad (4.1.4)$$

permite obtener la forma del sistema PCH deseado,

$$\dot{x} = [J_d(x) - R_d(x)]\nabla H_d \quad (4.1.5)$$

donde $x = [q \quad p]^T$.

4.1.1 Balance de energía.

Como se explicó en el capítulo 3, la función $H(q, p)$ (3.5.4) es la función de energía total del sistema y satisface lo siguiente (Ortega, van der Schaft, Maschke, & Escobar, 2002):

El sistema representado en (4.1.1) es pasivo desde $u \rightarrow y$, donde la salida pasiva y en sistemas mecánicos corresponde a \dot{q} , satisface la ecuación de balance de energía:

$$H[x(t)] - H[x(0)] = \int_0^t u^T(t)y(t)dt - d(t) \quad (4.1.6)$$

La cual se mantiene para todo $t \geq 0$. Esto quiere decir que si el sistema es acotado desde abajo define un operador pasivo que tiene como función $H[x(t)]$. En ese caso la ecuación anterior expresa el hecho de que un sistema no puede almacenar más energía de la que se suministra desde el exterior, siendo la diferencia la energía disipada.

Es fácil observar que la energía del sistema no forzado es no creciente, esto es $H[x(t)] \leq H[x(0)]$, y puede decrecer en presencia de disipación. Si el sistema es acotado desde abajo, esto significa que eventualmente el sistema podrá detenerse en un punto de mínima energía. Además la velocidad de convergencia de la función de energía es aumentada si extraemos energía del sistema, por ejemplo,

estableciendo $u = -K_v y$, con $K_v = K_v^T > 0$, también llamado *inyección de amortiguamiento*. El punto de interés aquí es que la energía en lazo abierto tiene un mínimo en un punto que usualmente no es de interés, entonces se introduce una ley de control alrededor de un punto de equilibrio distinto de cero, que nombramos x^* .

4.1.2 Conceptos y estabilidad de Lyapunov (Kelly & Santibañez, 2003).

La teoría de estabilidad de Lyapunov tiene como principal objetivo estudiar el comportamiento de sistemas dinámicos descritos por ecuaciones diferenciales de la forma,

$$\dot{x}(t) = f(t, x(t)), \quad x(0) \in \mathbb{R}^n \quad \forall t \geq 0 \quad (4.1.7)$$

donde el vector $x(t) \in \mathbb{R}^n$ se refiere al estado del sistema dinámico representado por la ecuación antes descrita y $x(0) \in \mathbb{R}^n$ se denomina la condición inicial o 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 (4.1.7) tiene una solución única en el intervalo inicial $[0, \infty)$ correspondiente a cada condición inicial de $x(0)$.
- Si $x(t)$ es la solución de (sistema diferencial) correspondiente a la condición inicial $x(0)$, entonces $x(t)$ depende de una manera continua del estado inicial $x(0)$.

Si la función f no depende explícitamente del tiempo, i.e. $f(t, x(t)) = f(x(t))$, luego la ecuación (sistema diferencial) se denomina autónoma. Si $f(t, x(t)) = A(t)x(t) + u(t)$ con $A(t)$ una matriz cuadrada de dimensión n y siendo $A(t)$ y $u(t)$ funciones únicamente de t (o constantes), entonces (4.1.7) es lineal. Caso contrario es no lineal.

Entre los conceptos básicos de la teoría de Lyapunov destacan los siguientes: equilibrio, estabilidad, estabilidad asintótica y estabilidad exponencial. Cada uno de ellos será tratado a continuación. Primeramente se presenta el concepto de equilibrio, el cual, a pesar de su sencillez, juega un papel clave en la teoría de estabilidad de Lyapunov.

Definición 1. Estado de equilibrio.

Un vector constante $x_e \in \mathbb{R}^n$ es un equilibrio o estado de equilibrio del sistema (4.1.7) si:

$$f(t, x_e) = 0 \quad \forall t \geq 0$$

Como consecuencias inmediatas de la definición inicial $x(0) \in \mathbb{R}^n$ es justamente un equilibrio $x(0) = x_e \in \mathbb{R}^n$, entonces se satisface:

- $x(t) = x_e \quad \forall t \geq 0$
- $\dot{x}(t) = 0 \quad \forall t \geq 0$

Definición 2. Estabilidad.

El origen $x(t) = 0 \in \mathbb{R}^n$ es un equilibrio estable (en el sentido de Lyapunov) de (4.1.7) si para cada número $\varepsilon > 0$ se puede encontrar un número $\delta > 0$, tal que:

$$\|x(0)\| < \delta \Rightarrow \|x(t)\| < \varepsilon \quad \forall t \geq 0$$

En la definición anterior la constante $\delta (\leq \varepsilon)$ no es única. Si existe una constante δ que satisfaga la condición de la definición, entonces cualquier $\alpha\delta$ con $\alpha \leq 1$ también la satisface.

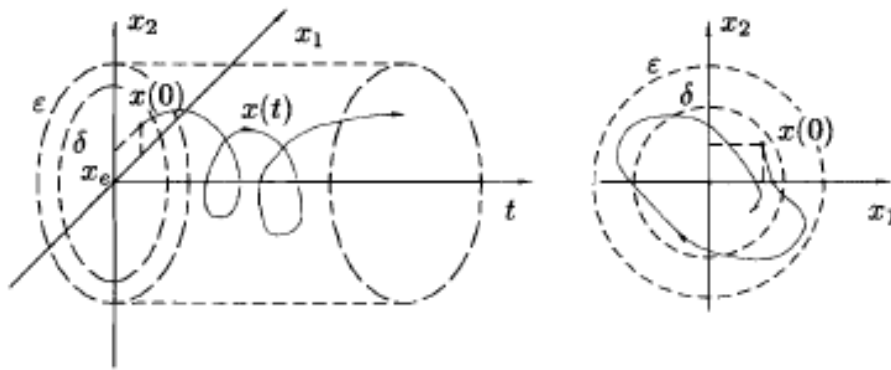


Fig. 4.2: Estabilidad de un sistema.

A modo de ilustración en Fig. 4.2 se muestra un caso para $x(0) \in \mathbb{R}^2$ donde el origen $x = 0 \in \mathbb{R}^2$ es un equilibrio estable y, por simplicidad, solo se muestra la trayectoria $x(t)$ correspondiente a una condición inicial $x(0)$. Se muestra también un ε y un δ que satisfacen la condición de la definición de estabilidad.

Definición 3. Estabilidad asintótica.

El origen $x = 0 \in \mathbb{R}^n$ es un equilibrio asintóticamente estable de (4.1.7) si:

- El origen es estable.
- El origen es atractivo, es decir, existe un número $\delta' > 0$ tal que: $\|x(0)\| < \delta' \Rightarrow \|x(t)\| \rightarrow 0$ cuando $t \rightarrow \infty$, como se observa en Fig. 4.3:

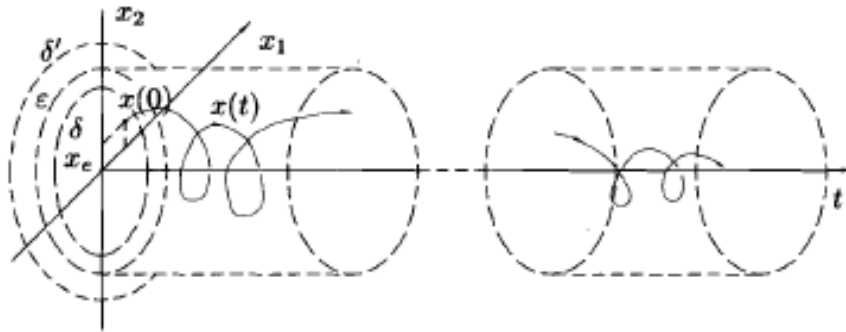


Fig. 4.3: Estabilidad asintótica.

La Fig. 4.3 ilustra el concepto de estabilidad asintótica para el caso de $x(0) \in \mathbb{R}^2$.

Definición 4. Estabilidad asintótica global.

El origen $x = 0 \in \mathbb{R}^n$ es un equilibrio asintóticamente estable en forma global de (4.1.7) si:

- El origen es estable.
- El origen es atractivo globalmente, es decir: $\|x(t)\| \rightarrow 0$ cuando $t \rightarrow \infty$, $\forall x(0) \in \mathbb{R}^n$

De la definición anterior se deduce que un equilibrio asintóticamente estable globalmente implica que dicho equilibrio es asintóticamente estable, pero lo contrario es falso.

Definición 5. Estabilidad exponencial global.

El origen $x = 0 \in \mathbb{R}^n$ es un equilibrio exponencialmente estable en forma global de (4.1.7) si existen constantes positivas α y β tales que:

$$\|x(t)\| < \alpha \|x(0)\| e^{-\beta t}, \quad \forall t \geq 0, \quad \forall x(0) \in \mathbb{R}^n$$

De acuerdo con las definiciones anteriores, un equilibrio exponencialmente estable en forma global es un equilibrio asintóticamente estable en forma global. Lo contrario, sin embargo, no es necesariamente verdadero.

Definición 6. Inestabilidad.

El origen $x = 0 \in \mathbb{R}^n$ es un equilibrio inestable de (4.1.7) si este no es estable. El concepto de equilibrio (origen) inestable es equivalente a enunciar que existe al menos un $\varepsilon > 0$ para el cual no es posible encontrar un $\delta > 0$, tal que:

$$\|x(0)\| < \delta \Rightarrow \|x(t)\| < \varepsilon, \quad \forall t \geq 0$$

En otras palabras, si existe al menos un $\varepsilon > 0$, el cual se desea que acote la norma de la solución $\|x(t)\|$ y no existe ninguna condición inicial $x(0) \neq 0 \in \mathbb{R}^n$ cuya solución $x(t)$ lo satisfaga ($\|x(0)\| < \delta \Rightarrow \|x(t)\| < \varepsilon \quad \forall t \geq 0$), entonces el origen es inestable.

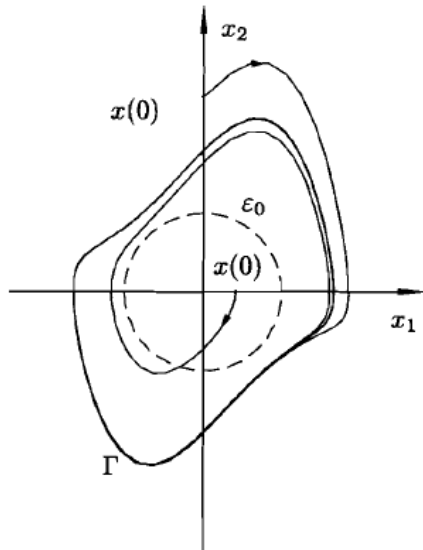


Fig. 4.4: Oscilador de Van der Pol con el origen del sistema como equilibrio inestable.

El origen del sistema Van der Pol es un equilibrio inestable. Para demostrarlo basta con tomar $\varepsilon = \varepsilon_0$ y claramente no existe un $\delta > 0$ tal que:

$$\|x(0)\| < \delta \Rightarrow \|x(t)\| < \varepsilon_0, \quad \forall t \geq 0.$$

Para el análisis de estabilidad del sistema no lineal se utiliza la definición de Lyapunov a continuación:

Definición 7. Sea $H : D \rightarrow \mathbb{R}$ un campo escalar continuamente diferenciable definido en un dominio $D \subset \mathbb{R}^n$ que contiene a x^* entonces:

- $H(x)$ es una función definida positiva si $H(x^*) = 0$ y $H(x) > 0$ en $D - x^*$.
- $H(x)$ es una función semidefinida positiva si $H(x^*) = 0$ y $H(x) \geq 0$ en D .
- $H(x)$ es una función definida negativa si $-H(x^*)$ es definida positiva.
- $H(x)$ es una función semidefinida negativa si $-H(x^*)$ es semidefinida positiva.

Definición 8. Una función $H(x)$ definida positiva con $\dot{H}(x)$ semidefinida negativa se llama **función de Lyapunov** para x^* ; si además se cumple que $\dot{H}(x)$ es definida negativa, se le llama **Lyapunov estricta** con relación a x^* .

Para estudiar la estabilidad de los sistemas no lineales en una determinada región a partir de las funciones de Lyapunov, se tiene el siguiente teorema, también conocido como método directo de Lyapunov.

Teorema 1. (Segundo método de Lyapunov): si existe una función de Lyapunov $H(x)$ para el punto de equilibrio x^* del sistema (4.1.1), entonces x^* es estable. Además, si esta función es Lyapunov estricta entonces el punto de equilibrio es asintóticamente estable.

Las definiciones anteriores se utilizarán en la siguiente sección y se demostrará que la función Hamiltoniana es una función de Lyapunov.

4.2 Descripción del método IDA-PBC.

El método usado en este trabajo no considera las fuerzas de fricción disipativas en los actuadores del sistema. Debido a esto el modelo a utilizar será (3.5.7) estableciendo el sistema sin amortiguamiento $D = 0, R_n = 0$ donde este último es la matriz que representa la fricción. Por lo tanto el sistema PCH toma la siguiente forma:

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}}_{f(x)} \begin{bmatrix} \nabla_q H \\ \nabla_p H \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ G \end{bmatrix}}_{g(x)} u \quad (4.2.1)$$

donde H es la función de energía del sistema dada por (3.5.4).

Similarmente podemos definir la función de energía deseada para el sistema de la misma forma:

$$H_d(q, p) = \frac{1}{2} p^T M_d^{-1}(q) p + U_d(q) \quad (4.2.2)$$

donde el subíndice d , indica “deseable” es decir la referencia, $M_d = M_d^T > 0$ es la matriz de inercia deseada y U_d la función de energía potencial deseada.

En el enfoque de PBC la señal de control es naturalmente descompuesta en la forma:

$$u = u_{es} + u_{di} \quad (4.2.3)$$

donde el primer término se diseña para lograr el moldeo de energía, y el segundo corresponde a la inyección de amortiguamiento. En (Ortega, van der Schaft, Maschke, & Escobar, 2002) se justifica la elección de la dinámica Hamiltoniana deseada de la forma:

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = [J_d(q, p) - R_d(q, p)] \begin{bmatrix} \nabla_q H_d \\ \nabla_p H_d \end{bmatrix} \quad (4.2.4)$$

donde:

$$J_d(q, p) = \begin{bmatrix} 0 & M^{-1}M_d \\ -M_d M^{-1} & J_2(q, p) \end{bmatrix} \quad (4.2.5)$$

$$R_d = R_d^T = \begin{bmatrix} 0 & 0 \\ 0 & GK_v G^T \end{bmatrix} \quad (4.2.6)$$

representan las estructuras deseadas de interconexión y amortiguamiento.

Lo siguiente a realizar es igualar el sistema (4.2.1) con la dinámica deseada en lazo cerrado (4.2.4), utilizando (4.2.2):

$$\begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H \\ \nabla_p H \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} (u_{es} + u_{di}) = \begin{bmatrix} 0 & M^{-1}M_d \\ -M_dM^{-1} & J_2(q, p) \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & GK_vG^T \end{bmatrix} \begin{bmatrix} \nabla_q H_d \\ \nabla_p H_d \end{bmatrix} \quad (4.2.7)$$

$$\begin{bmatrix} \nabla_p H \\ -\nabla_q H \end{bmatrix} + \begin{bmatrix} 0 \\ Gu_{es} + Gu_{di} \end{bmatrix} = \begin{bmatrix} 0 & M^{-1}M_d \\ -M_dM^{-1} & J_2(q, p) - GK_vG^T \end{bmatrix} \begin{bmatrix} \nabla_q H_d \\ \nabla_p H_d \end{bmatrix} \quad (4.2.8)$$

de la ecuación anterior, obtenemos dos ecuaciones diferenciales parciales,

$$\nabla_p H = M^{-1}M_d \nabla_p H_d \quad (4.2.9)$$

$$-\nabla_q H + Gu_{es} + Gu_{di} = -M_dM^{-1}\nabla_q H_d + J_2(q, p)\nabla_p H_d - GK_vG^T\nabla_p H_d \quad (4.2.10)$$

tomando en cuenta que u_{di} es la retroalimentación de la nueva entrada pasiva y como nuestro sistema es completamente actuado, entonces G es una matriz identidad y es invertible,

$$u_{di} = -K_vG^T\nabla_p H_d \quad (4.2.11)$$

equivalente también,

$$u_{di} = -K_vg^T(x)\nabla H_d \quad (4.2.12)$$

de (4.2.10) tenemos la siguiente ecuación (Ortega,2004),

$$u_{es} = \nabla_q H - M_dM^{-1}\nabla_q H_d \quad (4.2.13)$$

equivalente también a,

$$u_{es} = [g^T(x)g(x)]^{-1}g^T(x)\{[J_d(x)]\nabla H_d - f(x)\} \quad (4.2.14)$$

ahora haremos las siguientes suposiciones para simplificar la ley de control y obtener una más simple, pero no necesariamente menos eficiente.

- Si se realiza moldeo de la energía potencial, esto implica naturalmente que las matrices de inercias deben de ser iguales, esto es $M_d(q) = M(q)$ y el parámetro $J_2 = 0$.
- La simplificación arriba hecha nos permite hacer que $J_d = J$, esto es la matriz de interconexión no se modifica y es igual a la matriz de interconexión del sistema.

Con estas suposiciones entonces resolvemos (4.2.11) y obtenemos la siguiente ley de control:

$$u_{es} = \nabla_q U - \nabla_q U_d \quad (4.2.15)$$

donde sabemos que $\nabla_q U$ es el vector de gravedad $gv(q)$.

Ahora podemos hacer una adecuada selección de U_d para encontrar una ley de control. Una buena elección de U_d puede ser la siguiente (Paim, Neves, Moreno, & de Pieri, 2011):

$$U_d(q) = \frac{1}{2}(q - q_d)^T K_p (q - q_d) \quad (4.2.16)$$

Siendo q_d la posición de referencia del actuador. Como ya conocemos la energía potencial deseada del sistema así como también la función de energía deseada. Podemos reescribir (4.2.15),

$$u_{es} = \nabla_q U - K_p (q - q_d) \quad (4.2.17)$$

de la misma manera la acción de control responsable del amortiguamiento puede ser calculada, tomando en cuenta que G^T es una identidad, $\nabla_p H_d = M_d^{-1} p$ siendo $M_d(q) = M(q)$ y de (3.5.9) ,

$$u_{di} = -K_v \dot{q} \quad (4.2.18)$$

por lo tanto nuestra ley de control u está dada por:

$$u = \nabla_q U - K_p (q - q_d) - K_v \dot{q} \quad (4.2.19)$$

el cual es una ley de control conocida como controlador PD con compensación de gravedad (Takegaki & Arimoto, 1981), (Paim, Neves, Moreno, & de Pieri, 2011), ampliamente utilizado en control de estabilidad en sistemas mecánicos.

De acuerdo con la ley de control obtenida, la cual hace que nuestro punto de interés en lazo cerrado sea asintóticamente estable, realizamos la siguiente comprobación de la función Hamiltoniana y el por qué la selección de ella.

Considere la función candidata de Lyapunov (3.5.1) , donde $K(q, \dot{q})$ denota la energía cinética del robot. Claramente la función $H(q, \dot{q})$ es una función definida positiva en forma global, ya que la energía cinética $K(q, \dot{q})$ es definida positiva en \dot{q} y por otro lado tomamos la energía potencial análogamente como:

$$U(q) = \frac{1}{2} \tilde{q}^T K_p \tilde{q} \quad (4.2.20)$$

$$\tilde{q} = q - q_d \quad (4.2.21)$$

la cual es una función cuadrática, donde K_p es una matriz definida positiva, entonces también la forma cuadrática $\tilde{q}^T K_p \tilde{q}$ es una función definida positiva en \tilde{q} . Entonces (3.5.1) puede representarse como,

$$H(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} + \frac{1}{2} \tilde{q}^T K_p \tilde{q} \quad (4.2.22)$$

si la derivamos con respecto al tiempo tenemos,

$$\dot{H}(q, \dot{q}) = \dot{q}^T M(q) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{M}(q) \dot{q} + \tilde{q}^T K_p \dot{\tilde{q}} \quad (4.2.23)$$

donde conocemos el termino $M(q) \ddot{q}$ de acuerdo con (3.2.72) ,

$$M(q) \ddot{q} = -C(q, \dot{q}) \dot{q} - B(\dot{q}) - gv(q) + u \quad (4.2.24)$$

sustituyendo la ecuación anterior en (4.2.23), tomando en cuenta que $\nabla_q U = gv(q)$ y utilizando (4.2.19) ,

$$\begin{aligned} \dot{H}(q, \dot{q}) = \dot{q}^T [& -C(q, \dot{q}) \dot{q} - B(\dot{q}) - gv(q) + gv(q) - K_p(q - q_d) - K_v \dot{q}] \\ & + \frac{1}{2} \dot{q}^T \dot{M}(q) \dot{q} + \tilde{q}^T K_p \dot{\tilde{q}} \end{aligned} \quad (4.2.25)$$

donde por consideraciones del modelo $B(\dot{q}) = 0$ y $\tilde{q} = q - q_d$, se reduce a la forma,

$$\dot{H}(q, \dot{q}) = -\dot{q}^T C(q, \dot{q}) \dot{q} - \dot{q}^T K_p(\tilde{q}) - \dot{q}^T K_v \dot{q} + \frac{1}{2} \dot{q}^T \dot{M}(q) \dot{q} + \tilde{q}^T K_p \dot{\tilde{q}} \quad (4.2.26)$$

donde podemos eliminar el primero y cuarto termino de acuerdo a la propiedad (3.3.3) y como q_d es una constante, entonces $\dot{\tilde{q}} = \dot{q}$, y obtenemos la siguiente ecuación:

$$\dot{H}(q, \dot{q}) = -\dot{q}^T K_v \dot{q} \quad (4.2.27)$$

entonces la función $H(q, \dot{q})$ es una función de Lyapunov en vista de que cumple con $\dot{H}(q, \dot{q}) \leq 0$, ya que \dot{q} no puede ser negativo ni tampoco cero puesto que alguna articulación esta en movimiento. La función $H(q, \dot{q})$ es una función de Lyapunov para el sistema (3.2.72), y debido a que el análisis no se hizo para un único punto de equilibrio, sino que fue de forma general. Por lo tanto se dice que el sistema es globalmente asintóticamente estable (Yepez, 2013).

4.3 Obtención del controlador.

Se utiliza la metodología de la sección 4.2 para encontrar la ley de control que satisface nuestro objetivo de control. Se considera el sistema de la forma (4.1.1) y se pasa a la forma PCH con $D = 0$ y $R_n = 0$ (3.5.7).

Tenemos 4 GDL ($n = 4$) y un sistema completamente actuado, esto hace que G sea invertible:

$$q = [q_1 \quad q_2 \quad q_3 \quad q_4]^T \quad (4.3.1)$$

$$p = [p_1 \quad p_2 \quad p_3 \quad p_4]^T \quad (4.3.2)$$

$$u = [\tau_1 \quad \tau_2 \quad \tau_3 \quad \tau_4]^T \quad (4.3.3)$$

la función de energía del sistema es:

$$H(q, p) = \frac{1}{2} p^T M^{-1} p + U(q) \quad (4.3.4)$$

donde:

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{12} & M_{22} & M_{23} & M_{24} \\ M_{13} & M_{23} & M_{33} & M_{34} \\ M_{14} & M_{24} & M_{34} & M_{44} \end{bmatrix} \quad (4.3.5)$$

$$M_{12} = I_2 + I_3 + I_4 + m_2 (lc_2^2 + l_1 lc_2 \cos q_2) + (m_3 + m_4) (l_2^2 + l_1 l_2 \cos q_2) \quad (4.3.6)$$

$$M_{13} = M_{23} = M_{33} = I_3 + I_4 \quad (4.3.7)$$

$$M_{14} = M_{24} = M_{34} = 0 \quad (4.3.8)$$

$$M_{22} = I_2 + I_3 + I_4 + m_2 lc_2^2 + (m_3 + m_4) l_2^2 \quad (4.3.9)$$

$$M_{44} = m_4 \quad (4.3.10)$$

$$U(q) = -m_4 g q_4 \quad (4.3.11)$$

La función Hamiltoniana deseada es la siguiente:

$$H_d(q, p) = \frac{1}{2} p^T M^{-1} p + U_d(q, q_d) \quad (4.3.12)$$

donde,

$$U_d(q, p) = \frac{1}{2} \begin{bmatrix} q_1 - q_{1d} \\ q_2 - q_{2d} \\ q_3 - q_{3d} \\ q_4 - q_{4d} \end{bmatrix}^T \begin{bmatrix} K_{v1} & 0 & 0 & 0 \\ 0 & K_{v2} & 0 & 0 \\ 0 & 0 & K_{v3} & 0 \\ 0 & 0 & 0 & K_{v4} \end{bmatrix} \begin{bmatrix} q_1 - q_{1d} \\ q_2 - q_{2d} \\ q_3 - q_{3d} \\ q_4 - q_{4d} \end{bmatrix} \quad (4.3.13)$$

se calcula la ley de control (4.2.3),

$$u_{es} = \begin{bmatrix} -\nabla_{q_1} H_d + \nabla_{q_1} H \\ -\nabla_{q_2} H_d + \nabla_{q_2} H \\ -\nabla_{q_3} H_d + \nabla_{q_3} H \\ -\nabla_{q_4} H_d + \nabla_{q_4} H \end{bmatrix} \quad (4.3.14)$$

$$u_{di} = \begin{bmatrix} -K_{v1} \nabla_{p_1} H_d \\ -K_{v2} \nabla_{p_2} H_d \\ -K_{v3} \nabla_{p_3} H_d \\ -K_{v4} \nabla_{p_4} H_d \end{bmatrix} \quad (4.3.15)$$

se resolvieron las ecuaciones anteriores mediante el software MAPLE 18 y se obtiene la siguiente ley de control:

$$u_{es}(1) = -K_{p1}(q_1 - q_{1d}) \quad (4.3.16)$$

$$u_{es}(2) = -K_{p2}(q_2 - q_{2d}) \quad (4.3.17)$$

$$u_{es}(3) = -K_{p3}(q_3 - q_{3d}) \quad (4.3.18)$$

$$u_{es}(4) = -m_4 * g - K_{p4}(q_4 - q_{4d}) \quad (4.3.19)$$

$$u_{di}(1) = -K_{v1} * (((m_3 + m_4)l_2 + m_2lc_2)(p_2 - p_3)l_1 \cos q_2 - (p_2 - p_1)((m_3 + m_4)l_2^2 + m_2lc_2^2 + I_2)) / (-((m_3 + m_4)l_2 + m_2lc_2)^2 l_1^2 \cos^2 q_2 + ((m_2 + m_3 + m_4)l_1^2 + m_1lc_1^2 + I_1)((m_3 + m_4)l_2^2 + m_2lc_2^2 + I_2))) \quad (4.3.20)$$

$$u_{di}(2) = -K_{v2} * ((2((m_3 + m_4)l_2 + m_2lc_2)(p_2 - \frac{1}{2}p_3 - \frac{1}{2}p_1)l_1 \cos q_2 \quad (4.3.21)$$

$$\begin{aligned} &+ (m_2 + m_3 + m_4)(p_2 - p_3)l_1^2 + (m_3 + m_4)(p_2 - p_1)l_2^2 \\ &+ lc_2^2(p_2 - p_1)m_2 + (lc_1^2m_1 + I_1 + I_2)p_2 \\ &+ (-lc_1^2m_1 - I_1)p_3 - I_2p_1) / (-((m_3 + m_4)l_2 + m_2lc_2)^2l_1^2 \cos^2 q_2 \\ &+ ((m_2 + m_3 + m_4)l_1^2 + m_1lc_1^2 + I_1)((m_3 + m_4)l_2^2 + m_2lc_2^2 + I_2))) \end{aligned}$$

$$u_{di}(3) = -K_{v3} * ((-p_3((m_3 + m_4)l_2 + m_2lc_2)^2l_1^2 \cos^2 q_2 \quad (4.3.22)$$

$$\begin{aligned} &+ (I_3 + I_4)((m_3 + m_4)l_2 + m_2lc_2)(p_1 - p_2)l_1 \cos q_2 \\ &+ ((m_2 + m_3 + m_4)l_1^2 + m_1lc_1^2 + I_1)(p_3(m_3 + m_4)l_2^2 \\ &+ (lc_2m_2 + I_2 + I_3 + I_4)p_3 - p_2(I_3 + I_4))) / ((I_3 + I_4)(-((m_3 + m_4)l_2 \\ &+ m_2lc_2)^2l_1^2 \cos^2 q_2 + ((m_2 + m_3 + m_4)l_1^2 + m_1lc_1^2 + I_1)((m_3 + m_4)l_2^2 \\ &+ m_2lc_2^2 + I_2))) \end{aligned}$$

$$u_{di}(4) = -K_{v4} * \left(\frac{p_4}{m_4}\right) \quad (4.3.23)$$

4.4 Sintonización de ganancias por mediante la Optimización por Enjambre de Partículas (PSO).

La sintonización de las ganancias para el controlador propuesto debe cumplir con una dinámica deseada considerando la capacidad física de los actuadores, esto debido a la respuesta no suave que puede no producir el actuador.

Actualmente, aún existe diversas problemas para la sintonización de los controladores de robots industriales, pero se pueden encontrar diferentes propuestas para la búsqueda y selección de las ganancias proporcionales (Hernandez, Santibañez, Carrillo, Molina, & López, 2008), (Tomei, 1991) las cuales garantizan estabilidad y robustez, pero no hay un método definitivo para su correcta selección.

Al ser este un sistema altamente no lineal, se pueden encontrar diferentes regiones, las cuales pueden dar una dinámica similar en el actuador, es por eso que se propone el método PSO el cual es capaz de buscar en una región para encontrar la mejor solución, a partir de la minimización de un error llamado *fitness*.

El valor *fitness* puede ser seleccionado a conveniencia del diseñador, penalizando diferentes aspectos, en este caso se restringe el sobretiro de la salida de posición del sistema, tratando de asemejar una respuesta de primer orden para el control de posición.

Dado lo anterior, las ganancias tanto proporcionales como derivativas, se seleccionan mediante el método PSO, el cual hace una búsqueda dentro de una región de posibles soluciones y mediante el *fitness*, se ajusta la búsqueda para encontrar la solución para alcanzar un comportamiento deseado.

PSO es un método de optimización heurístico, evolutivo y de búsqueda (Glover & Kochenberger, 2003), inspirado en el comportamiento social de las bandadas de pájaros o cardúmenes de peces. Las soluciones, llamadas partículas se “echan a volar” en el espacio de búsqueda, guiadas por la partícula que mejor solución ha encontrado hasta el momento y que hace de líder de la bandada. Cada partícula evoluciona teniendo en cuenta la mejor solución encontrada en su recorrido y al líder. En cada iteración, las partículas modifican su velocidad hacia la mejor solución de su entorno teniendo en cuenta la información del líder. Existen dos versiones diferentes de PSO, el algoritmo normal y una versión binaria (Dorigo & Gambardella, 1997). En el segundo, las partículas solo pueden tomar el valor 0 o 1, para este caso se utilizó el PSO normal.

La ventaja del PSO es que es fácil de implementar y contiene muy pocos parámetros a modificar. Otra ventaja es que podemos penalizar ciertas variables para hacer que nuestra solución se ajuste a un *fitness* específico, por ejemplo, en este caso se penaliza el sobretiro en la salida q , se desea que la trayectoria que describa el eslabón tenga una respuesta suave similar a la respuesta de los sistemas de primer orden, después de eso se verifica que el esfuerzo de control no exceda de cierto par debido a que no sería posible realizar por el actuador. A continuación se establecen las siguientes definiciones:

Tabla 4.1: Definición de términos del algoritmo PSO.

R^D	El espacio de búsqueda designado, con D la cantidad de dimensiones que cuenta dicho espacio.
$X_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD})$	La posición de la i -ésima partícula en R^D .
$P_i = (p_{i1}, p_{i2}, p_{i3}, \dots, p_{iD})$	La mejor posición anterior (la posición que dio el mejor resultado) de la particular i -ésima.
$V_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{iD})$	La velocidad de la particular i -ésima en R^D .
P_{best}	El valor de la mejor <i>fitness</i> hasta este momento encontrado. Es el valor que se obtiene al aplicar la función de <i>fitness</i> a cada partícula.
P_{id}	Es la mejor posición que ha obtenido la particular i -ésima en la dimensión d .
P_{gd}	Es la mejor posición del grupo de partículas en la dimensión d .
$v_{id}(t)$	Velocidad de la i -ésima partícula en la dimensión d .
$x_{id}(t)$	Posición actual de la particular i -ésima en la dimensión d .
φ_1	Es un número aleatorio que se encuentra en el rango $[0, Qmax]$.
φ_2	Es un número aleatorio que se encuentra en el rango $[0, Qmax]$.
$Vmax$	Velocidad máxima que puede alcanzar una partícula.
$-Vmax$	Velocidad mínima que puede alcanzar una partícula.
ω	Coefficiente de inercia: valor aleatorio en el rango $[0.5, 1]$

4.4.1 Pasos del algoritmo.

La posición de una partícula i cualquiera la denotaremos por X_i , donde X_i es un vector que almacena cada una de las posiciones que tiene la partícula en cada una de las dimensiones que comprende el espacio de búsqueda. Además, denotaremos por V_i a la velocidad de la partícula i , la cual también es un vector, el cual contiene cada una de las velocidades que tiene la partícula en cada dimensión. Esta velocidad es añadida a la posición de la partícula, para mover la partícula desde un tiempo $t-1$ a un tiempo t (Kennedy & Eberhart, 2001).

A continuación se describen cada uno de los pasos que conlleva este algoritmo:

Paso 1:

Inicializar una población de partículas con posiciones y velocidades aleatorias en un espacio D -dimensional.

Paso 2:

Para cada partícula, evaluar la función de *fitness* en D variables.

Paso 3:

Comparar el *fitness* actual de cada partícula con el *fitness* de su mejor posición anterior, p_{best} . Si el valor actual es mejor que el de p_{best} , entonces asignarle a p_{best} el valor actual, y $P_i=X_i$, es decir, la localización actual pasa a ser la mejor hasta el momento.

Paso 4:

Identificar la partícula en la vecindad (puede ser todo el conjunto de partículas, o un grupo de ellos) con la mejor tasa de éxito hasta el momento, y asignar su índice a la variable g .

Paso 5:

Ajustar la posición y velocidad de la partícula de acuerdo a las ecuaciones (para cada dimensión) siguientes:

$$v_{id}(t) = \omega * v_{id}(t-1) + \varphi_1(p_{id} - x_{id}(t-1)) + \varphi_2(p_{gd} - x_{id}(t-1)) \quad (4.4.1)$$

$$x_{id}(t) = x_{id}(t-1) + v_{id}(t) \quad (4.4.2)$$

Paso 6:

Verificar si se cumple el criterio de detención (máximo número de iteraciones o valor de *fitness* alcanzado). Si no ir al paso 2.

La velocidad es una función que está compuesta por la suma de tres partes. La primera parte es la velocidad anterior de la partícula. A esta parte se le conoce como inercia. La segunda parte es la diferencia entre la mejor posición encontrada por la partícula, con la actual posición. Esta es la parte cognitiva que representa el aprendizaje de su propia experiencia. Y la última parte es la diferencia entre la mejor posición alcanzada por un vecino, con la posición actual de la partícula. Esta parte es la social, que representa el aprendizaje del grupo.

Lo anterior descrito se programó en un archivo *.m* en MATLAB con el que se encuentra los valores de las ganancias cercanos a los valores que, empíricamente, tenían un comportamiento deseado y se limita a que los valores de iteración de las ganancias no sean negativos, el valor de *fitness* se obtiene realizando una suma de los sobretiros que existe en cada uno de los actuadores, es decir, el área que describe el sobretiro se toma como un error y se suma a los de los demás, siendo este un *fitness* ponderado el cual es la suma de todos los sobretiros de los actuadores, entre más pequeño sea el valor de *fitness*, significa que el error ponderado en sobretiro es menor.

4.4.2 Ganancias obtenidas para la ley de control.

Las ganancias para nuestro controlador dadas por $(Kp1, \dots, Kp4)$ y $(Kv1, \dots, Kv4)$, deben de ajustarse para obtener una dinámica deseada dentro de los parámetros reales y posibles que el actuador puede realizar, es decir, que los pares necesarios para el movimiento articular puedan ser proporcionados por el actuador. Se propuso la sintonización de parámetros del controlador mediante el algoritmo de búsqueda de la sección anterior y las ganancias encontradas para esta simulación fueron:

Tabla 4.2: Ganancias del controlador encontrados mediante PSO para el robot SCARA.

$K_{p1} = 2460$
$K_{p2} = 1503$
$K_{p3} = 1593$
$K_{p4} = 1986$
$K_{v1} = 378$
$K_{p2} = 220$
$K_{p3} = 108$
$K_{p4} = 139$

Se detalla en Apéndice D la simulación en la que se encontraron dichos valores.

4.5 Conclusión del capítulo.

En esta sección se presentó los conceptos relacionados con pasividad y con estabilidad de Lyapunov, se describió el método por el cual se obtuvo la ley de control dada por (4.3.14) y (4.3.15), la cual se conoce como controlador PD más compensación de gravedad, y se demostró que con la elección de la función H_d y nuestra ley de control obtenida, el sistema se vuelve global asintóticamente estable.

5 Simulaciones y resultados.

En esta sección se muestran los resultados de la simulación del sistema robótico obtenido en el capítulo 3, con la ley de control obtenida en el capítulo 4. Se presentan diversas pruebas tales como regulación, seguimiento y repetición de tareas, todas ellas con y sin perturbación en sus masas, también se muestra las gráficas del error para cada una de las pruebas.

5.1 Descripción de las pruebas.

Se realizaron diversas simulaciones bajo el ambiente *SIMULINK*, la cual contiene el modelo del robot en la forma PCH, y se tiene como salidas las posiciones y momentos generalizados. La ley de control es la presentada en (4.3.14) y (4.3.15), y se utilizan los parámetros de Tabla 3.1 .Se utilizan bloques de funciones definidas por el usuario para facilitar el ingreso de las ecuaciones y se acoplan a la simulación los bloques de cinemática directa e inversa (Fig. 2.7) obtenidos en el capítulo 2.

La primera parte de las pruebas de regulación y seguimiento es sin perturbación en sus masas, y la segunda se realiza añadiendo una masa a m_4 , simulando un *payload* en su efector para poder observar la trayectoria y el error que describe en cada una de ellas. Después se realizan una prueba en la que se realizan 10 simulaciones con un rango de variación paramétrica en sus 4 masas con condiciones iniciales diferentes para observar la convergencia a un punto de regulación. Por último se realiza una tarea *pick & place* y se observan sus gráficas de comportamiento. El esquema en *SIMULINK* es el siguiente (véase también el Apéndice C):

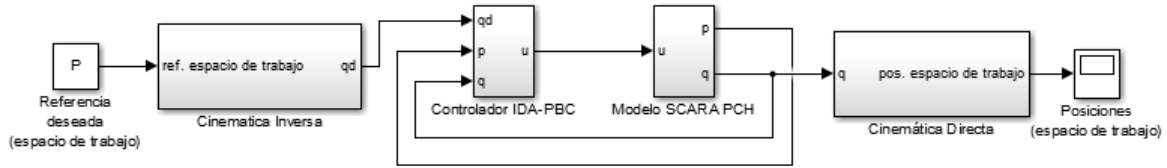


Fig. 5.1: Esquema de control del sistema en lazo cerrado

5.2 Regulación

El problema de control de posición o regulación consiste en mover el extremo final del robot manipulador hacia una posición deseada constante q_d ; independientemente de su posición inicial $q(0)$. Los puntos intermedios entre la posición inicial y la referencia deseada no son controlados como tal, forman parte de la etapa transitoria de la respuesta del robot incluido en la señal de error \tilde{q} , definida como la diferencia entre la posición deseada y la posición actual del robot. El punto final q_d es la posición clave donde se realiza el control. El objetivo del problema de control de posición está determinado en encontrar una ley de control u que proporcione los pares aplicados a las articulaciones o servomotores del robot, de tal forma que la posición actual del robot $q(t)$, y la velocidad articular de movimiento $\dot{q}(t)$, tienda asintóticamente hacia la posición deseada q_d y velocidad cero, respectivamente, sin importar las condiciones iniciales, es decir:

$$\lim_{t \rightarrow \infty} \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} q_d \\ 0 \end{bmatrix}.$$

El objetivo de control se cumple cuando:

$$\lim_{t \rightarrow \infty} \begin{bmatrix} \dot{q}(t) \\ \tilde{q}(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Evidentemente para propósitos industriales, una vez que el extremo final del robot ha alcanzado el punto deseado, en el siguiente periodo de muestreo el punto deseado cambiará su valor, entonces el robot se moverá al siguiente punto; el anterior punto deseado representa la condición inicial y el extremo final del robot se estará posicionando en la siguiente coordenada; así sucesivamente el robot sigue

una curva compuesta de puntos cercanos entre sí. Esta forma de controlar a un robot se denomina **control punto a punto**. Es importante aclarar que en control punto a punto no se controla la velocidad de movimiento como en el caso de control de movimiento, donde el error y la velocidad son controlados simultáneamente. La gran mayoría de las aplicaciones actuales de la industria son realizadas a través de control punto a punto (Reyes, 2013).

Las condiciones iniciales y parámetros para las simulaciones de regulación y seguimiento son:

$$\begin{aligned} q(0) &= 0 \\ p(0) &= 0 \end{aligned} \tag{5.2.1}$$

5.2.1 Escenario 1: regulación sin perturbación.

Se analiza el siguiente punto de regulación:

Tabla 5.1: Punto deseado a regular en regulación.

Punto deseado $P(x, y, z, nx, ox, ax, ny, oy, ay, nz, oz, az)$	Cin. Inversa (q_1, q_2, q_3, q_4)	% perturbación en m_4
(0.45, 0.3, 0.15, 1, 0, 0, 0, 1, 0, 0, 0, 1)	(-0.197, 1.868, -1.621, -0.15)	0%

Es decir, el robot partirá del reposo $q(0) = 0$, con estas coordenadas articulares el efector está desplegado a lo largo del eje X en las coordenadas $P(x,y,z)=(0.9,0,0.3)$ y la orientación es colineal a los ejes XYZ. Se observan las siguientes gráficas donde se muestran la posición deseada del efector comparada con la posición real:

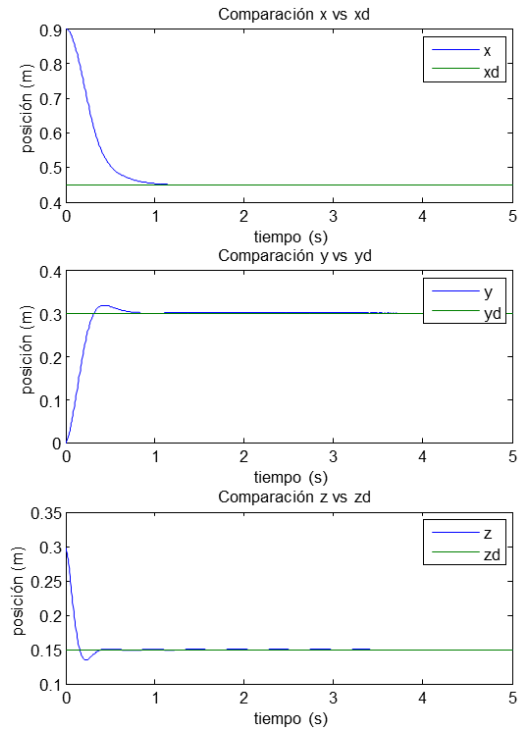


Fig. 5.2: Comparación del punto deseado vs. real en regulación sin perturbación.

La siguiente gráfica muestra la convergencia a cero los errores de posición:

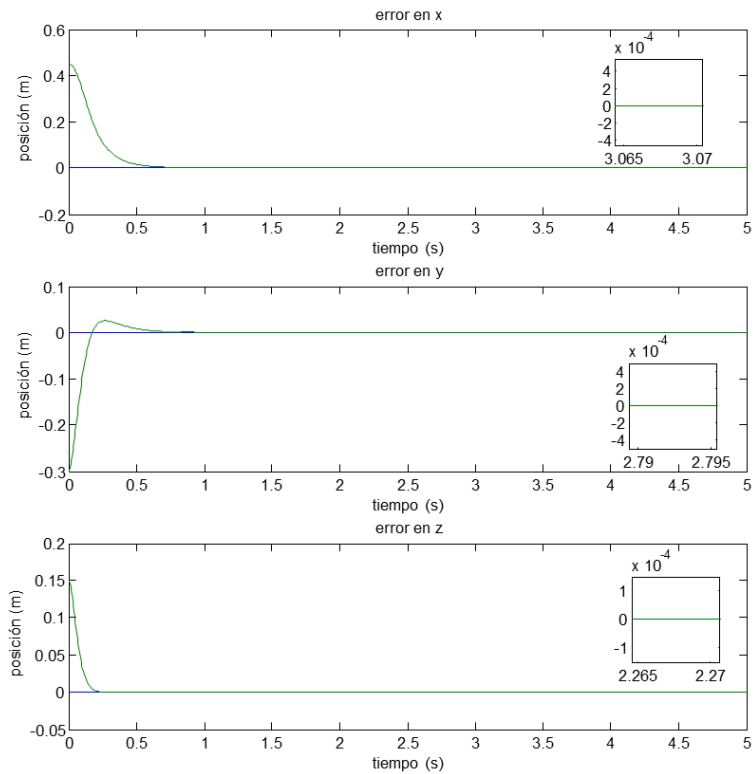


Fig. 5.3: Convergencia del error de posición en regulación sin perturbación.

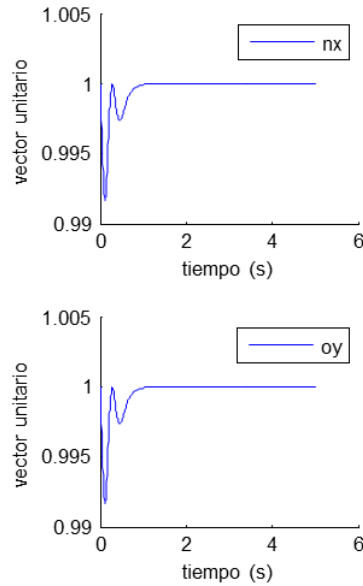


Fig. 5.4: Valores de las proyecciones n_x y o_y en regulación sin perturbación.

Se puede observar en Fig. 5.3 que el error es prácticamente cero respecto a las posiciones deseadas con las reales. Ahora se muestra los pares aplicados a cada uno de los actuadores:

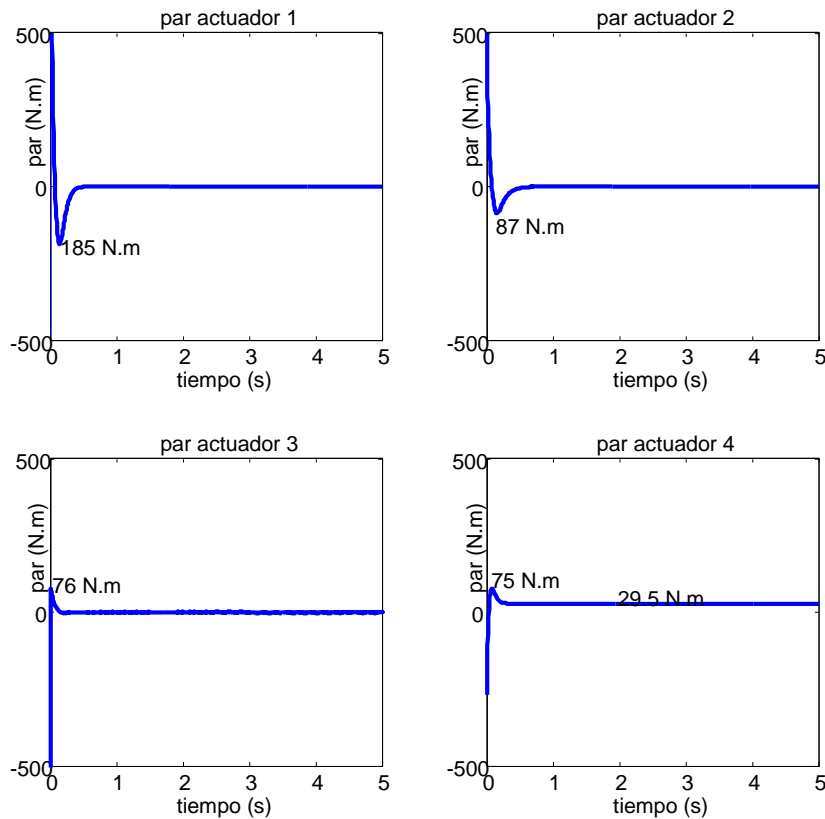


Fig. 5.5: Pares aplicados a los actuadores en regulación sin perturbación.

Como se puede observar en Fig. 5.3, la posición deseada en el punto P y la orientación es alcanzada correctamente en un tiempo menor a 1s, y el error de

posición es 0, debido a que no hay perturbación en el sistema. De la Fig. 5.5 se observa que el esfuerzo aplicado en el actuador 4 tiene un par constante de aproximadamente 29.5 N.m lo cual corresponde al par necesario para mantener el peso de las masas 3 y 4 y el efector en la posición deseada.

5.2.2 Escenario 2: Regulación con perturbación.

En la siguiente simulación se realiza un aumento de masa de un 5% (0.150kg) en m_4 , y se analiza el comportamiento ante una regulación. El punto a regular es el mismo de Tabla 5.1, y se muestran las gráficas de comparación en las siguientes figuras.

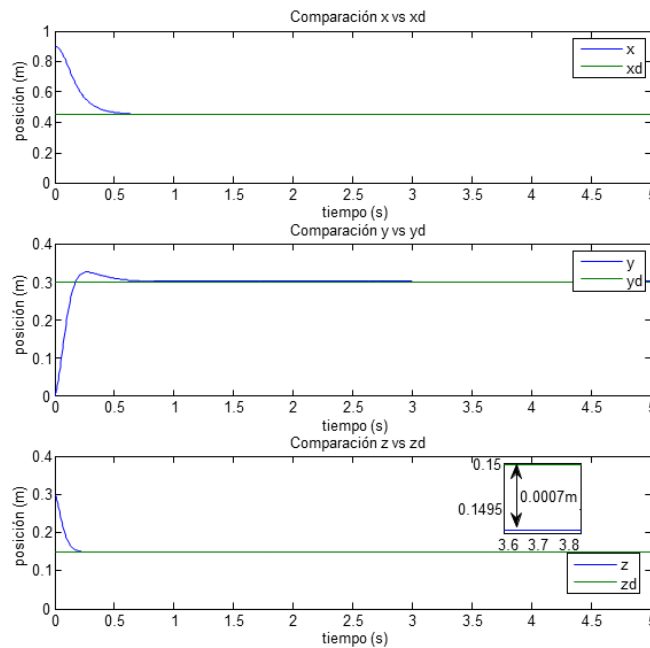


Fig. 5.6: Comparación del punto deseado vs real en regulación con perturbación.

En la comparación de X e Y, el error que se tiene es muy pequeño, aproximadamente de $5.9 \mu\text{m}$ y $5.7 \mu\text{m}$, respectivamente y la orientación del vector noa :

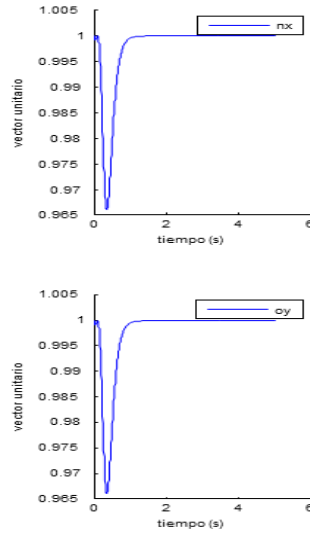


Fig. 5.7: Valores de las proyecciones n_x y o_y en regulación con perturbación.

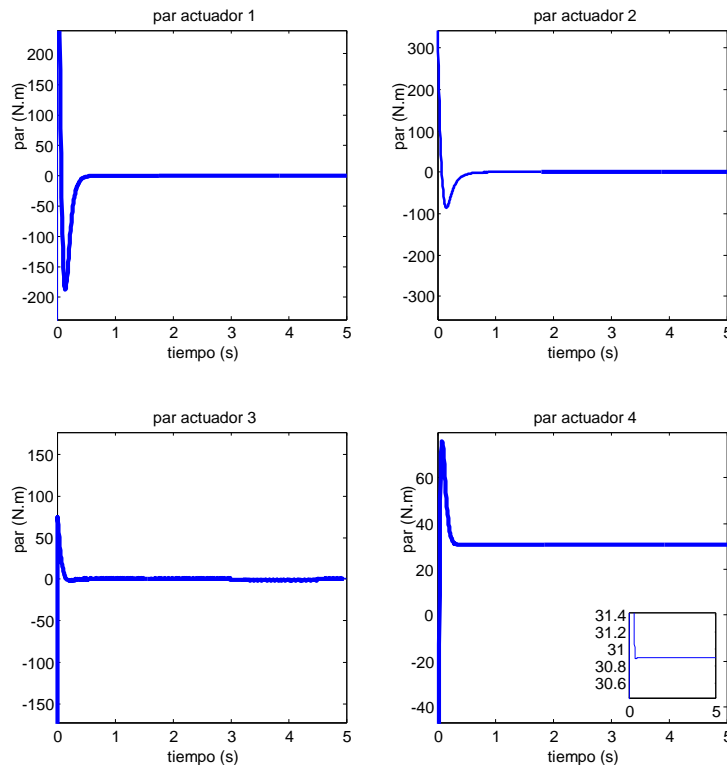


Fig. 5.8: Pares aplicados a los actuadores en regulación con perturbación.

Se puede observar en Fig. 5.6 que existe un error en estado estacionario de 0.0007m (0.7mm) en el eje Z, debido a que este eje es el único que tiene efecto directo de la carga m_4 , aun así, al cambiar el valor de esta masa, modifica la

dinámica de la estructura ya que afecta al comportamiento de los otros eslabones aumentando sus inercias y haciendo un poco más de esfuerzo para alcanzar el punto deseado. En Fig. 5.7 las proyecciones de los vectores unitarios se mantienen en 1, así que son colineales a los ejes XY.

En la Fig. 5.8, los pares de los actuadores 1, 2 y 3 tienden a cero, puesto que han alcanzado la posición deseada y al no actuar ninguna fuerza externa su posición no cambia, el par en el actuador 4 es de aproximadamente 30.9Nm lo cual es mayor a cuando no se tenía la perturbación, esto representa un esfuerzo mayor para mantener una posición estable.

5.3 Seguimiento.

El siguiente escenario de prueba permite analizar el comportamiento del sistema utilizando como referencia una trayectoria, es decir, una sucesión de puntos en el espacio de trabajo conocido como *control punto a punto*, se genera una trayectoria que cambie en el plano XYZ y la orientación es el vector *noa* colineal a los eje XYZ.

La trayectoria deseada es una línea recta en XY y en Z es una altura constante de 0.15m (Fig. 5.9):

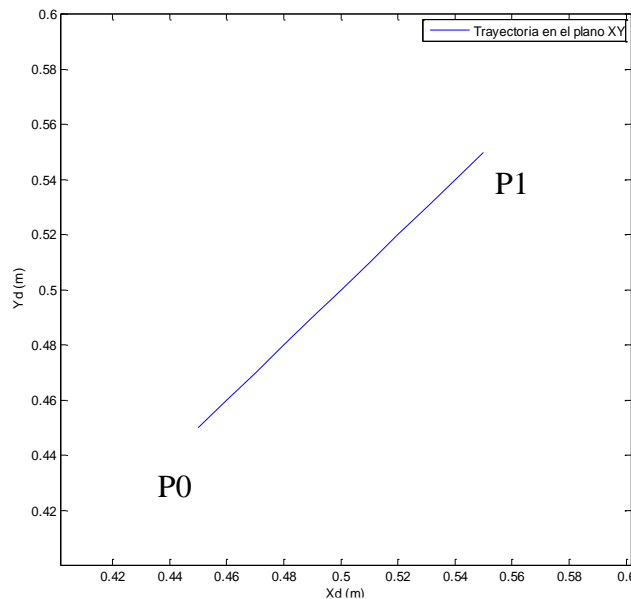


Fig. 5.9: Trayectoria deseada en el plano XY para seguimiento.

La función que genera esta trayectoria es la siguiente:

$$x = 0.45 + 0.02 * t \quad (5.3.1)$$

$$y = 0.45 + 0.02 * t$$

$$z = 0.15$$

Por lo que el desplazamiento en el eje X e Y son de 0.10m en 5s de simulación (2.82 cm/s), es una línea recta que empieza en las coordenadas $P0(x,y,z)=(0.45,0.45,0.15)$ y termina en el punto $P1(x,y,z)=(0.55,0.55,0.15)$, la distancia de recorrido del efector es de 14.14 cm.

5.3.1 Escenario 3: seguimiento sin perturbación.

Se muestran las gráficas de posición y de error en XYZ, para un seguimiento de trayectoria dadas las condiciones iniciales $q(0)=0$, la trayectoria dada por (5.3.1) y un *payload* de 5% (0.150kg):

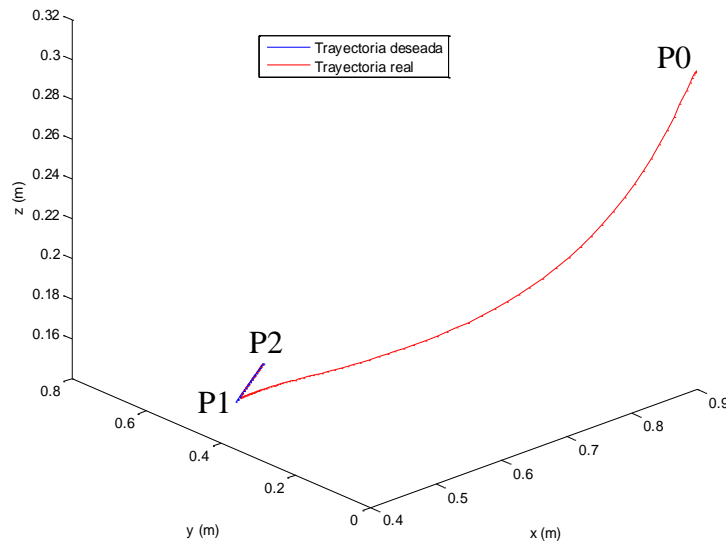


Fig. 5.10: Comparación de la trayectoria deseada vs real en el plano XYZ

Como puede observarse en Fig. 5.10, las condiciones iniciales corresponden a que el efector se encuentre en la posición $P0(0.9,0,0.3)$ y se lleve al punto inicial de la recta deseada, la cual inicia en $P1(0.45,0.45,0.15)$ y después a lo largo de la recta que termina en $P2(0.55,0.55,0.15)$, se muestra la gráfica del error en los respectivos ejes:

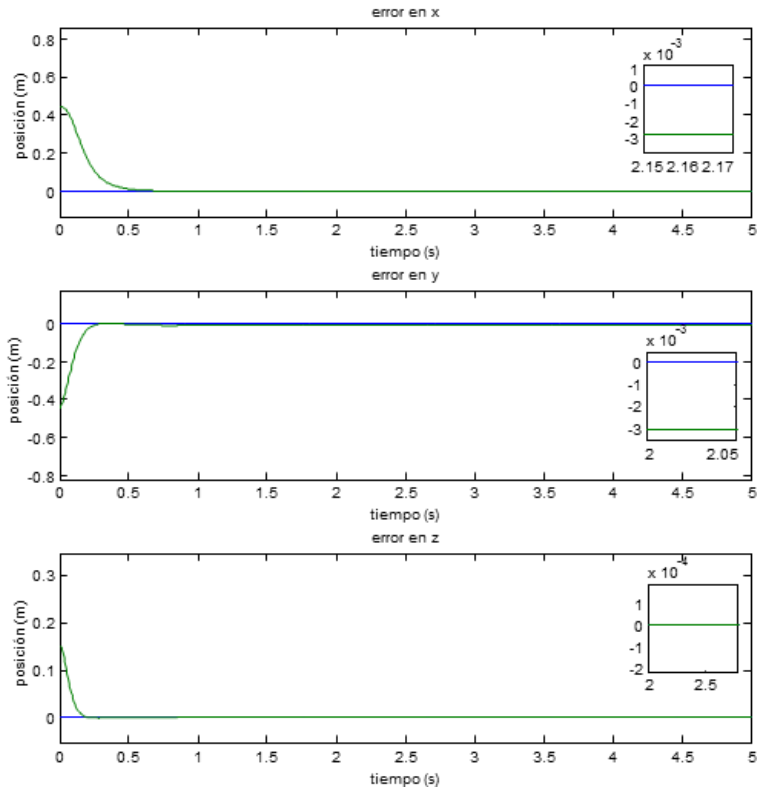


Fig. 5.11: Convergencia a cero en el error de posición del efector en seguimiento sin perturbación.

Los vectores n_x , o_y siguen siendo colineales a los ejes X e Y:

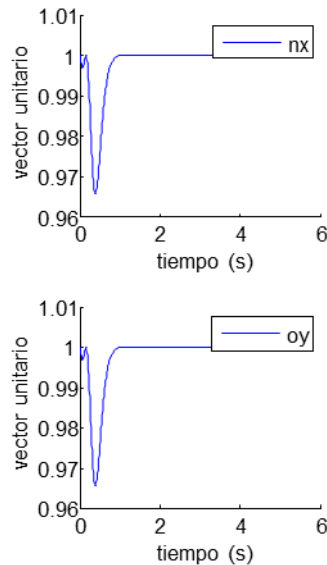


Fig. 5.12: Valores de las proyecciones n_x y o_y en regulación sin perturbación.

Se puede observar en la Fig. 5.11 que la posición del efector sigue la trayectoria con un error en estado estacionario menor a 0.003m en X e Y, y al no tener

perturbación, la posición en Z se mantiene correcta. Ahora se muestran los pares aplicados a los eslabones:

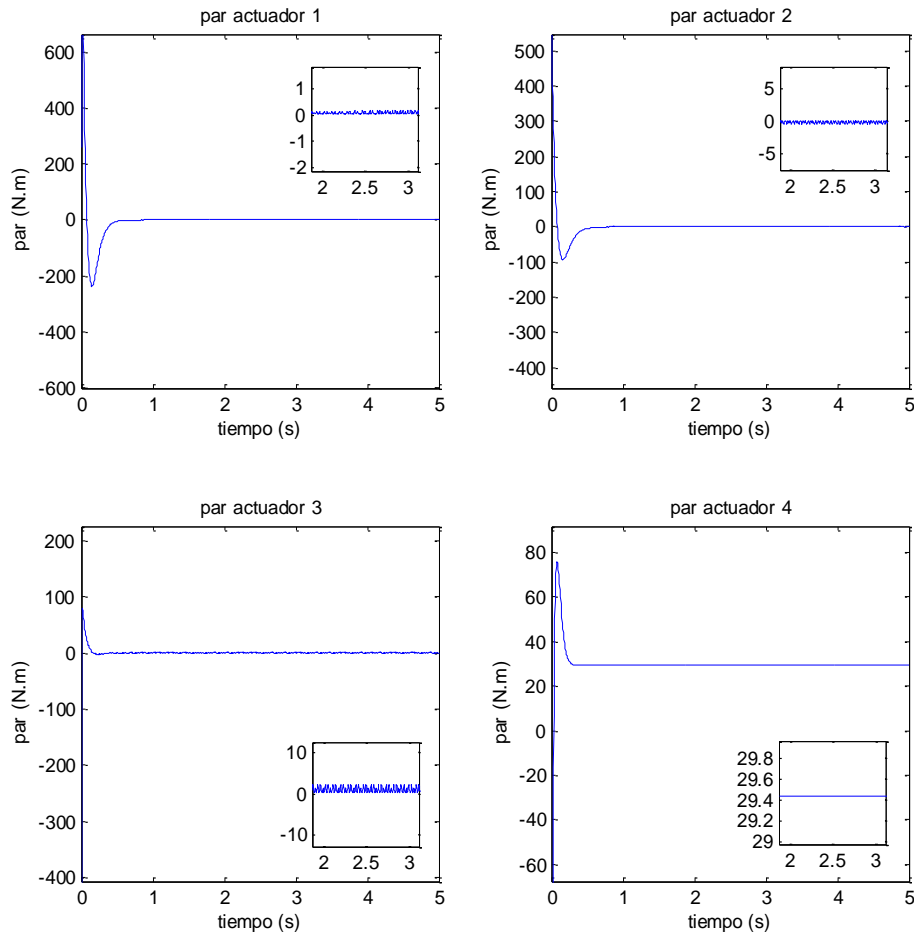


Fig. 5.13: Pares aplicados a los actuadores en regulación sin perturbación.

Se observa que después de un transitorio, existe una oscilación alrededor de cero, lo cual implica que están en movimiento los eslabones, y el par en el actuador cuatro es de 29.5N.m lo cual se relaciona con la fuerza necesaria para mantener el eslabón suspendido en la posición deseada.

5.3.2 Escenario 4: Seguimiento con perturbación.

Se muestran las gráficas de comparación de un seguimiento de trayectoria igual que la sección anterior, pero ahora con un aumento en la masa del eslabón 4, implicando un *payload* del 5%, la comparación se hace de forma individual para cada uno de los ejes con el fin de ser más ilustrativo.

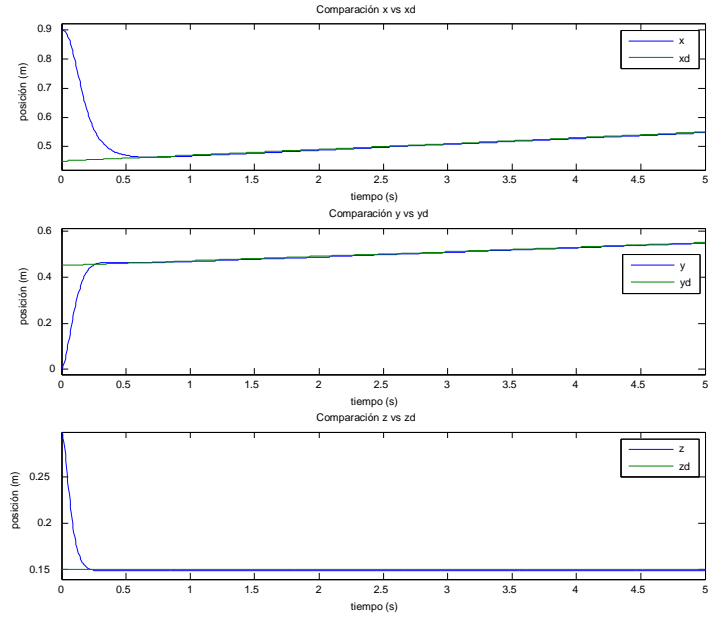


Fig. 5.14: Comparación de la trayectoria deseada vs real en seguimiento con perturbación.

En la Fig. 5.14 se compara la trayectoria deseada y la real, se puede observar una pequeña diferencia en los tres ejes, las cuales se aprecia en la Fig. 5.15.

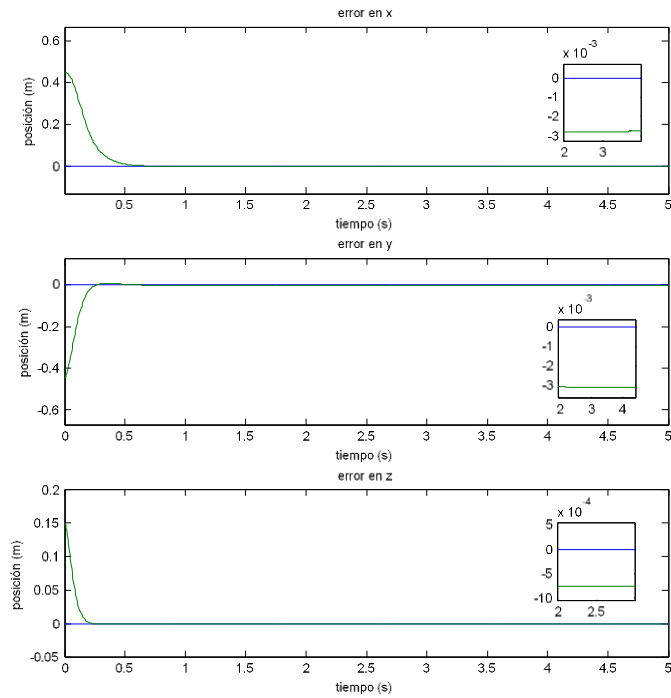


Fig. 5.15: Error de posición en la trayectoria deseada vs real en seguimiento con perturbación.

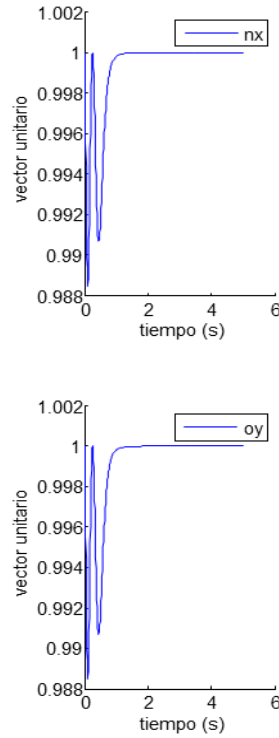


Fig. 5.16: Valores de las proyecciones n_x y o_y en seguimiento con perturbación.

La orientación del efector final sigue siendo unitaria por lo tanto, los vectores no_a son colineales a los ejes XYZ.

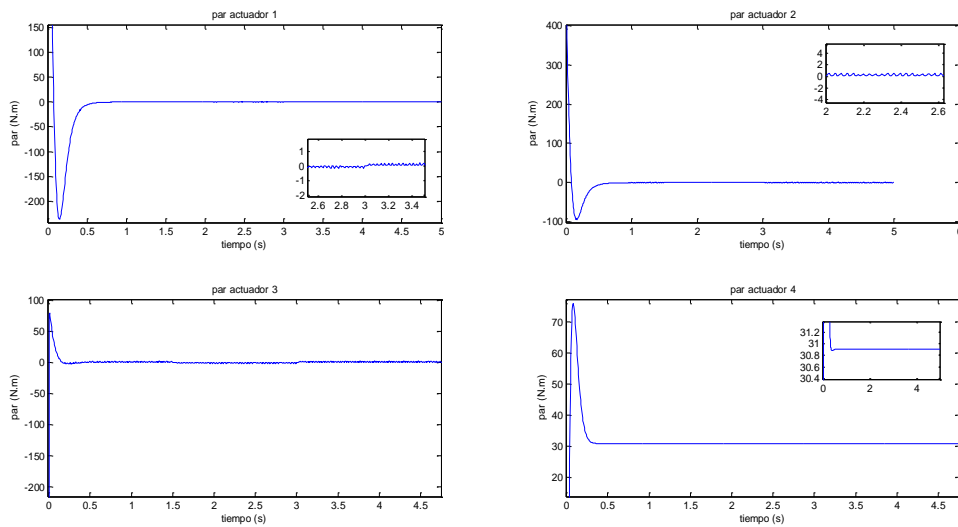


Fig. 5.17: Pares aplicados a los actuadores en seguimiento con perturbación.

En Fig. 5.17 se puede observar un error en estado estacionario en la posición del efector. En X un error de 0.0027 m, en Y de 0.0031 m y en Z de 0.0007 m., cuando

se le da una trayectoria como referencia, esta es seguida con un error y se observa que lo mismo sucedió con el seguimiento sin perturbación.

5.4 Escenario 5: perturbación en masas y condiciones iniciales diferentes.

Se realizó una serie de simulaciones del sistema con una variación paramétrica del $\pm 5\%$ en sus 4 masas y en las condiciones iniciales para observar su comportamiento ante una regulación en un punto fijo.

Se realizaron 10 simulaciones de las cuales se muestran los rangos de las variaciones paramétricas en las masas.

Tabla 5.2: Variación en las masas del modelo para simulación con perturbación.

$$\begin{aligned} 14.25 \text{ kg} &< m_1 < 15.75 \text{ kg} \\ 11.4 \text{ kg} &< m_2 < 12.6 \text{ kg} \\ 2.85 \text{ kg} &< m_3 < 3.15 \text{ kg} \\ 2.85 \text{ kg} &< m_4 < 3.15 \text{ kg} \end{aligned}$$

Las condiciones iniciales fueron tomadas en los siguientes rangos:

Tabla 5.3: Intervalos de las condiciones iniciales para simulación con perturbación.

$$\begin{aligned} 0 \text{ rad} &< q_1(0) < \pi/4 \text{ rad} \\ 0 \text{ rad} &< q_2(0) < \pi/4 \text{ rad} \\ 0 \text{ rad} &< q_3(0) < \pi/4 \text{ rad} \\ -0.3 \text{ m} &< q_4(0) < 0 \text{ m} \end{aligned}$$

El punto a regular es $P(x,y,z)=(0.7,0.4,0.1)$ para las 10 simulaciones, se muestran los resultados en Fig. 5.18.

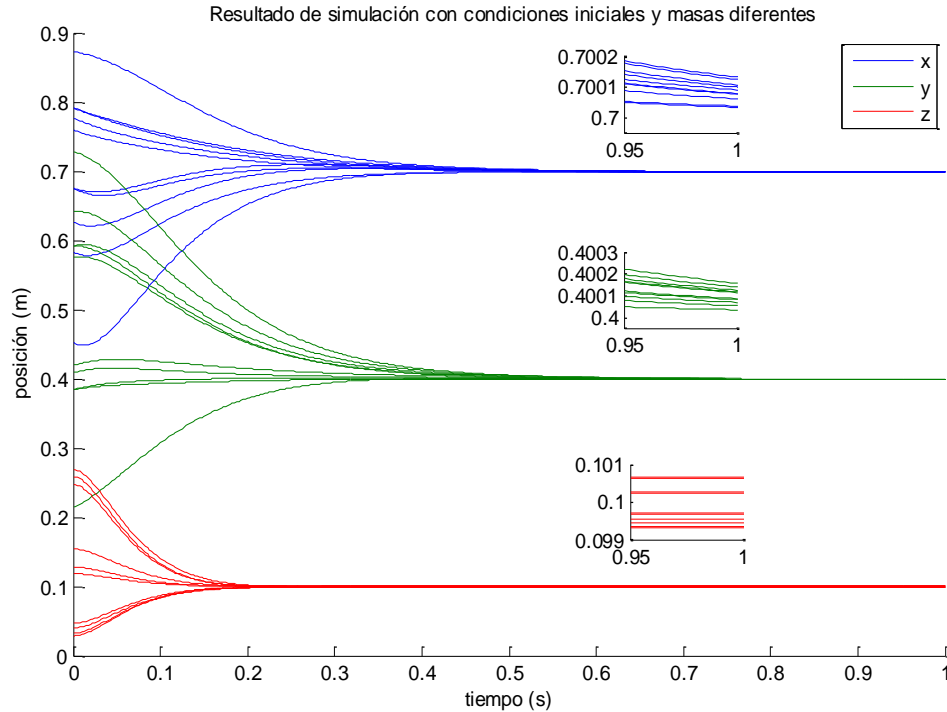


Fig. 5.18: Convergencia de las posiciones al punto deseado con diferentes condiciones iniciales y masas.

Se puede observar que partiendo de diferentes condiciones iniciales con perturbación en todas las masas, al llegar a 1s de simulación el error en el eje X es de 0.0002m, en Y es de 0.00025m y en Z es de 0.0009m, lo cual nos da una buena regulación ante la variación de masas en la estructura del robot, en base a esto y en las simulaciones anteriores se debe crear una región de tolerancia para que nuestro sistema realice las tareas *pick & place* cambiando de *setpoint* cuando el efector se encuentre en la región de tolerancia, es decir, se haga el *control punto a punto* cuando el controlador detecte que el punto ha sido alcanzado con un error máximo de 1mm.

Para comprobar su comportamiento ante una perturbación externa, se ingresa un escalón de magnitud 20 a los 2.5s de simulación con una duración de 0.1s a la retroalimentación de nuestro controlador, esto representa una fuerza de 20 N.m a los actuadores 1, 2 y 4, simulando un empuje de los eslabones.

Los datos de esta simulación están contenidos en Tabla 5.4.

Tabla 5.4: Valores de simulación para perturbación externa.

$m_1 = 15.1284 \text{ kg}$
 $m_2 = 12.3155 \text{ kg}$
 $m_3 = 2.8749 \text{ kg}$
 $m_4 = 3.04 \text{ kg}$
 $q_1(0) = 0.4060 \text{ rad}$
 $q_2(0) = 0.1343 \text{ rad}$
 $q_3(0) = 0.7371 \text{ rad}$
 $q_4(0) = -0.1229 \text{ m}$

El punto a regular es $P(0.7,0.4,0.10)$, se muestra la gráfica donde se ve el efecto de la fuerza externa:

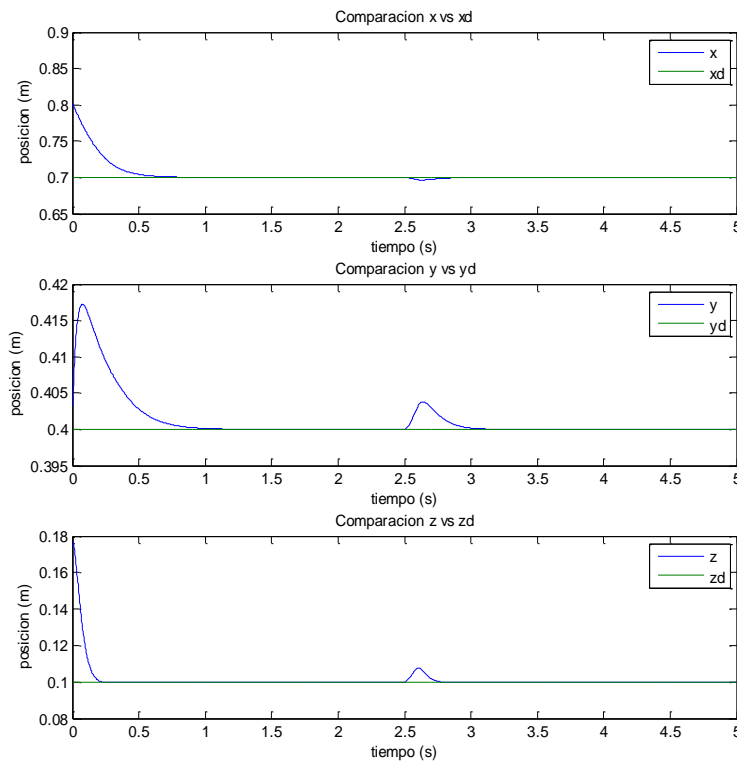


Fig. 5.19: Regulación ante una fuerza externa.

Como se puede observar, hasta los 2.5 s el sistema se encontraba en estado estable, cuando se ingresa el escalón el efector final cambia de posición, pero la acción de control lo regula de tal manera que vuelve a su posición deseada.

5.5 Escenario 6: Tarea pick & place.

Una de las principales barreras tecnológicas en la robótica industrial ha sido la reducción del error entre el efector final y el punto a alcanzar, las fuentes que producen este error han sido bien identificadas, los parámetros que tienen más influencia en la productividad de un robot son la exactitud y la repetibilidad.

Estos dos conceptos son con frecuencia confundidos, por lo que se proporcionan las siguientes definiciones (Conrad, Shiakolas, & Yih, 2000):

Repetibilidad es una medida de la capacidad del robot para volver a la misma posición y orientación; la exactitud es definida como la capacidad del robot de moverse precisamente a una posición deseada en un plano 3-D (Fig. 5.20).

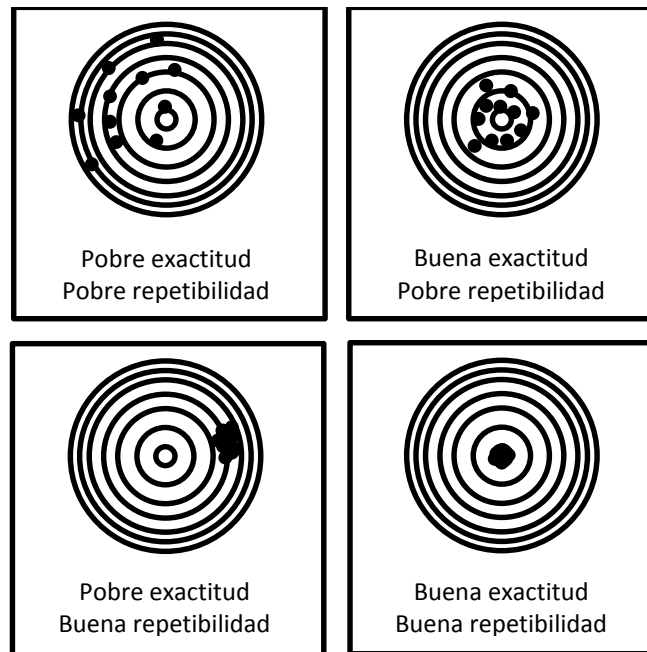


Fig. 5.20: Definición gráfica de exactitud vs repetibilidad.

El mayor efecto en la exactitud de un robot viene dada por la longitud de los eslabones del robot, en la manufactura es inevitable no tener variación en las dimensiones de los eslabones, y este error se propaga a través de los siguientes eslabones. La diferencia que existe entre la posición cero que el controlador detecta y la posición cero del robot real, tienen repercusiones en la exactitud de la localización del efector final.

Los procedimientos y tolerancias para la caracterización del comportamiento de los robots (exactitud y repetibilidad) están dadas en la norma ISO 9283:1998 (ISO, 1998), esta norma internacional es la más utilizada, también existe la norma ANSI/RIA R15.05-1 para describir estos parámetros. La norma ISO aplica para todo tipo de robots manipuladores y se especifica en la tabla de características básicas del robot.

Las pruebas asociadas a esta norma se realizan con ciertas características, tales como que el robot este completamente ensamblado y operacional, este debidamente nivelado y alineado, las condiciones ambientales de operación deben realizarse a $20^{\circ}\text{C} \pm 2^{\circ}\text{C}$, la carga de operación debe realizarse al 100% y a la máxima velocidad, entre otras características físicas. Después se trazan 5 puntos dentro de un plano inclinado que está situado en el interior de un cubo, dicho cubo está en la región de operación del robot, y se realizan 30 ciclos para poder determinar los valores de exactitud y repetibilidad.

Al ser este trabajo una simulación, la exactitud y repetibilidad son ideales, es decir la reproducibilidad de las simulaciones siempre va a dar los mismos puntos, así pues, no se puede obtener cuanta precisión o exactitud tiene en los diferentes puntos de operación.

Lo que se muestra en las siguientes figuras y tablas son tres simulaciones en la que se determina el error que existe entre un punto deseado y el punto obtenido, teniendo un aumento del valor de la masa 4 de 0%, después 2.5% y 5%, cada punto de referencia se mantiene durante 1s y luego cambia al siguiente, los puntos deseados se muestran en la siguiente tabla:

Tabla 5.5: Puntos deseados para tarea pick & place.

	Puntos deseados		
	X	Y	Z
P1	0.9	0	0.3
P2	0.8	0.4	0.2
P3	0.2	0.6	0.2
P4	0.7	0.1	0

En la Fig. 5.21 se muestran las trayectorias que describen las 3 simulaciones con los aumentos de masa, y se puede observar que las trayectorias no son las mismas, debido a la variación paramétrica, y como sabemos, solamente controlamos el punto final.

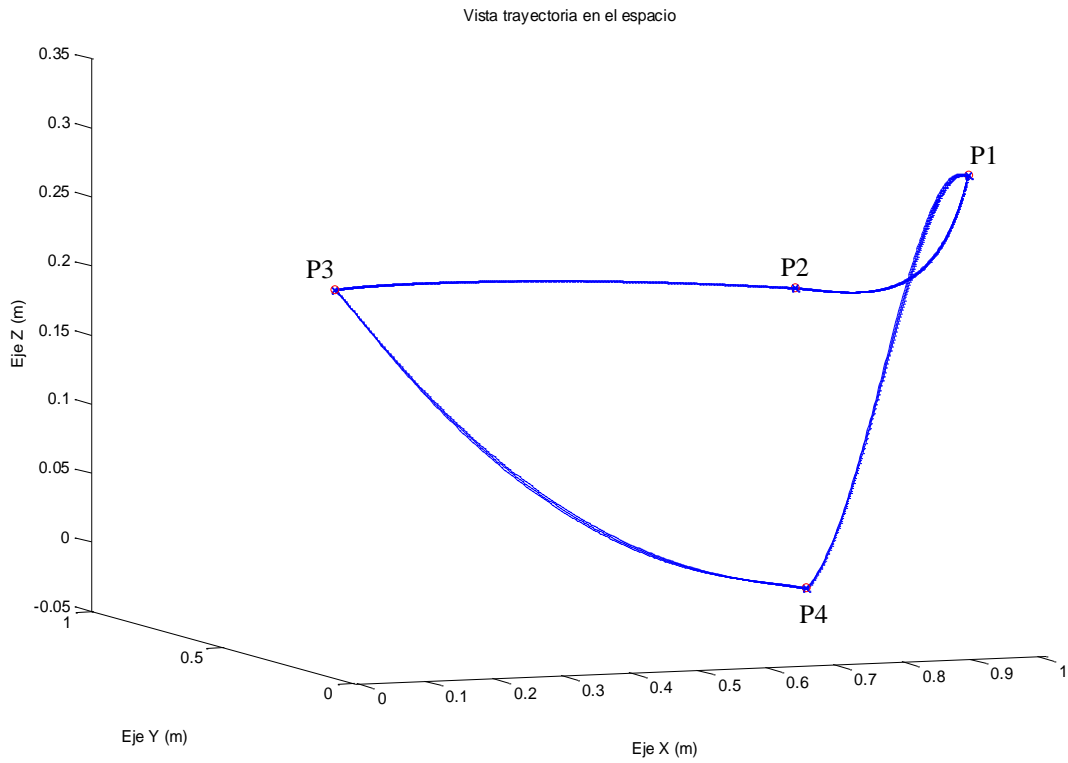


Fig. 5.21: Vista de la trayectoria descrita en las 3 simulaciones con 0%, 2.5% y 5% de aumento en masa 4.

Se muestra la vista del plano XY para fines ilustrativos:

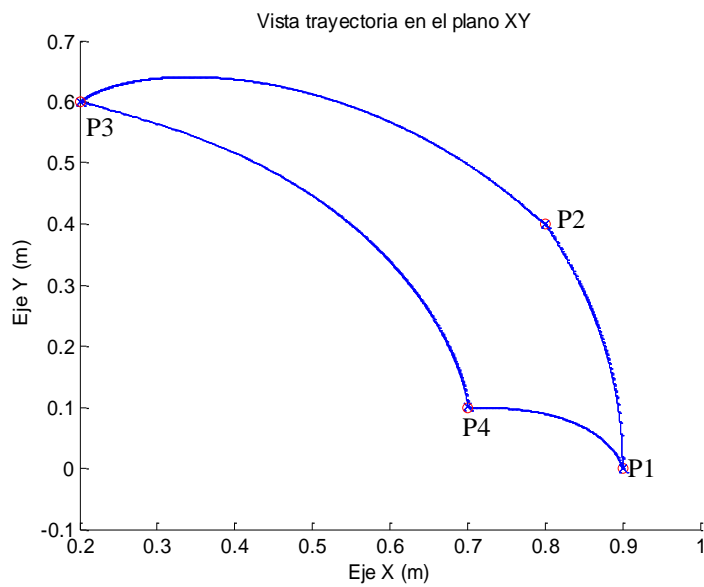


Fig. 5.22: Vista del plano XY de la trayectoria con 0%, 2.5% y 5% de aumento en masa 4.

En Fig. 5.23 se tiene una vista sobre el plano XY y se inserta una etiqueta que muestra la coordenada del punto más lejano del punto deseado, señalado con un círculo rojo.

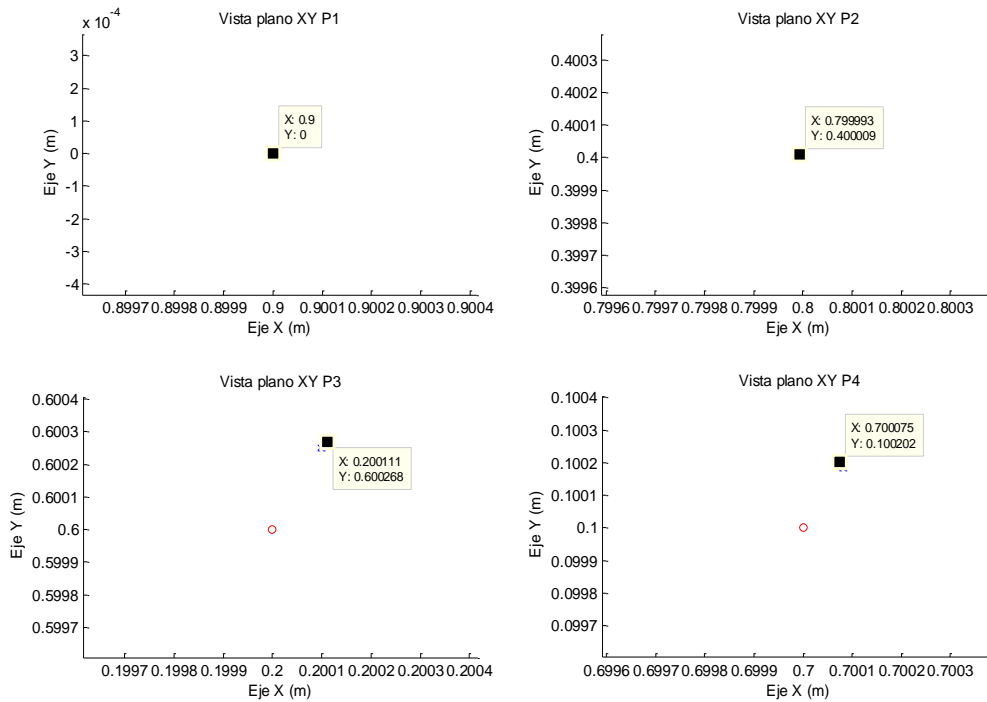


Fig. 5.23: Comparación de punto deseado y puntos obtenidos en *pick & place* plano XY.

En Fig. 5.24 se muestra el plano XZ en donde se aprecia un error en el eje Z de 0.7mm el cual corresponde al aumento de 5%, y se observa que en todos los puntos de mantiene el mismo valor de error, siendo estos especificados en *Tabla 5.8*.

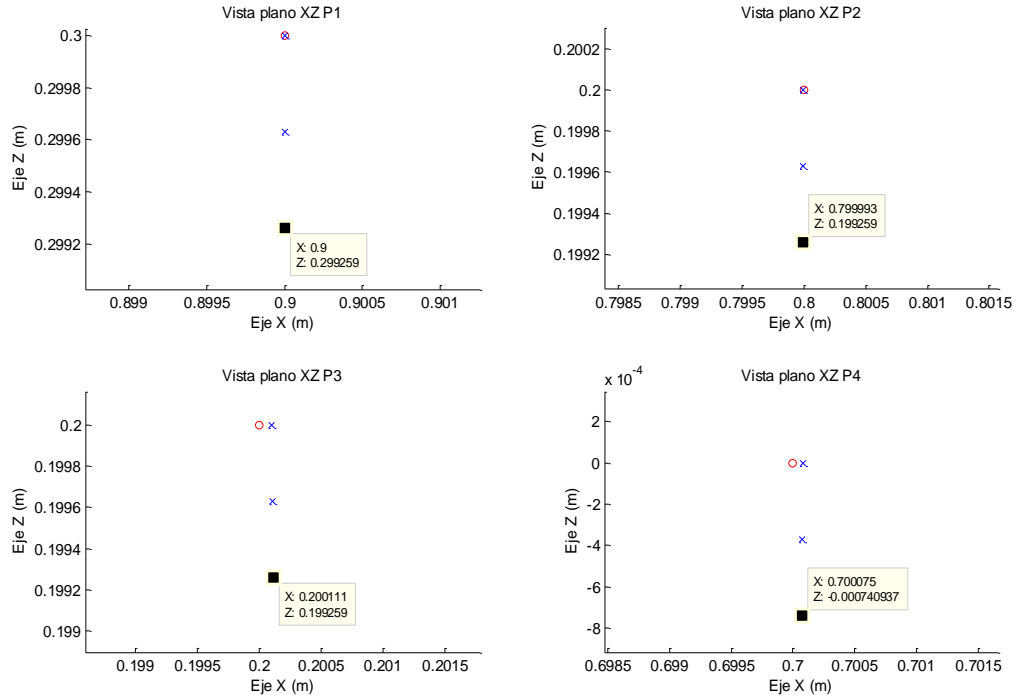


Fig. 5.24: Comparación de punto deseado y puntos obtenidos en *pick & place* plano XZ.

Tabla 5.6: Puntos obtenidos en *pick & place* con 0% de aumento en masa 4.

	0%		
	X	Y	Z
P1	0.9	0	0.3
P2	0.799992969736397	0.400008037481065	0.199999999983449
P3	0.200098321883228	0.600252965120260	0.2
P4	0.700079994244973	0.100187395046686	-3.182382135591411e-11

En Tabla 5.6 y Tabla 5.7, se muestran las posiciones obtenidas por el efector, con 0% de aumento en masa 4 y 2.5% respectivamente, para los 4 puntos deseados en la tarea *pick & place*.

Tabla 5.7: Puntos obtenidos en *pick & place* con 2.5% de aumento en masa 4.

	2.5%		
	X	Y	Z
P1	0.9	0	0.299629531721975
P2	0.799993070721611	0.400008655125634	0.199629531700508
P3	0.200105061041819	0.600261738263134	0.199629531722054
P4	0.700077891339060	0.100195474601231	-3.704683210380977e-04

Tabla 5.8: Puntos obtenidos en pick & place con 5% de aumento en masa 4.

	5%		
	X	Y	Z
P1	0.9	0	0.299259063444006
P2	0.799993224763570	0.400009199803286	0.199259063430204
P3	0.200112101557039	0.600270748660038	0.199259063444109
P4	0.700075590098217	0.100203877335917	7.409365837009463e-04

De acuerdo con la información anterior, se puede deducir que el error en los ejes X e Y es menor a 0.3mm y en el eje Z menor a 0.7mm, mientras que la carga máxima no exceda los 0.150g en el efector final como lo muestra la Tabla 5.8 , teniendo en cuenta que cada setpoint dura 1 segundo antes de cambiar a la nueva posición.

5.6 Conclusión del capítulo.

En este capítulo se desarrolló una simulación incluyendo el modelo no lineal del robot y la ley de control propuesta en el capítulo 4, pudiendo obtener un lazo cerrado, se realizaron pruebas de regulación y seguimiento, también se hizo pruebas de robustez ante perturbaciones en las masas, con fuerzas externas y con diferentes condiciones iniciales. Se determinó el error en cada una de las pruebas y se cuantifico el error máximo cuando la carga de trabajo no exceda de 0.150g.

6 Conclusiones y trabajos futuros.

En el presente trabajo se desarrolló un esquema de control basado en el método IDA-PBC para el sistema robótico SCARA de 4 GDL. Se modeló mediante el algoritmo de D-H la cinemática inversa y directa para conocer su geometría y los movimientos del robot. Se obtuvo el modelo dinámico en la estructura E-L para después obtener su representación en PCH. Se definió su función de energía Hamiltoniana la cual es necesaria en la metodología propuesta. Se definieron conceptos de pasividad, energía y estabilidad para comprender mejor la metodología IDA-PBC.

La principal característica de este método es que toda la información necesaria para entender el comportamiento dinámico del sistema está contenida en las funciones de energía y disipación, en consecuencia el controlador tiene un enfoque energético y el sistema es visto como un elemento transformador de energía.

Se comprobó que el sistema en lazo cerrado tiene una estabilidad global asintótica, es decir, el sistema se puede llevar a cualquier punto de operación a partir de cualquier condición inicial.

El controlador IDA-PBC propuesto demuestra una estabilidad robusta ante pequeños cambios de variación de parámetros de masa y condiciones iniciales tal como se muestra en la Fig. 5.19, la ley de control compuesta por (4.2.19), se compone por el moldeo de energía para alcanzar el punto deseado y la inyección de amortiguamiento para alcanzar estabilidad asintótica, siendo muy similar al comportamiento de un controlador PD + compensación de gravedad.

La velocidad de respuesta del controlador dependerá del valor de las ganancias, las cuales son las pendientes de la función del moldeo de energía y la inyección de amortiguamiento.

Las ganancias obtenidas mediante la técnica PSO son suficientes para tener una dinámica deseada por el diseñador y se comprueba que el esfuerzo de control necesario para esa dinámica, no exceda los valores reales que un actuador pueda proporcionar.

Este controlador tiene un buen desempeño ante problemas de regulación con un robot SCARA de 4 GDL como se demostró en el capítulo 5, que por su forma estructural permite minimizar las perturbaciones por la acción de la gravedad, es por eso que dicho robot sirve para tareas de *pick & place* donde es importante ubicar el efector final dentro de un plano XY y después realizar el movimiento de Z, que es el único elemento en el que se encuentra el efecto de la gravedad y la compensación por el payload.

Esta metodología IDA-PBC no fue desarrollada para el control de movimiento del robot, es decir controlar posición y velocidad al mismo tiempo, solo regulación, y no se puede tener un control del transitorio lo cual es un problema propio de los robots, debido a los sobretiros que existen al llegar a un punto de manera rápida. Aun así, en un seguimiento de trayectoria y al ser un robot para *pick & place*, se puede construir una trayectoria deseada, y como se mostró en el capítulo 5, existe un seguimiento con un error de estado estacionario, pero al llegar al punto final y siendo este un punto constante, se puede alcanzar a regular de manera satisfactoria.

Las salidas del sistema en la representación PCH son las coordenadas generalizadas de posición y momento, en la que el momento no es una variable física que se pueda obtener mediante sensores, por lo que se tiene que realizar una transformación para su obtención a partir de la velocidad.

Este trabajo es un caso de estudio por lo cual su aportación es la aplicación de la metodología propuesta en un sistema robótico, y analizar su desempeño, así como ver sus ventajas y desventajas que existen en él.

6.1 Trabajos futuros

A continuación se enlistan los trabajos que se proponen para ser realizados a futuro:

1. Validar el modelo obtenido respecto a un robot real.
2. Probar otros métodos de sintonización de ganancias.
3. Realizar el control de movimiento del robot.
4. Implementación del controlador en tiempo discreto.
5. Comprobar su desempeño respecto a otros controladores.
6. Realizar las pruebas de acuerdo a la norma ISO 9283:1998.



Bibliografía

- Acosta, V., & Rios-Bolívar, M. (2010). Application of the IDA-PBC in the stabilization of the system Pendubot. *Revista Ciencia e Ingeniería*, 31(1), 3-12.
- Banavar, R., Kazi, F., Ortega, R., & Manjarekar, N. (2006). The IDA-PBC methodology applied to a gantry crane. *17th International Symposium on Mathematical Theory of Networks and Systems*. Kyoto, Japan.
- Barrientos, A. (2007). *Fundamentos de robótica*. España: MCGRAW HILL.
- Conrad, K., Shiakolas, P., & Yih, T. (2000). Robotic calibration issues: accuracy, repeatability and calibration. *Proceedings of the 8th Mediterranean Conference on Control and Calibration*, (págs. 17-19). Rio Patras, Grecia.
- Dorigo, M., & Gambardella, L. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1).
- Glover, F., & Kochenberger, G. (2003). *Handbook of metaheuristics*. Colorado, USA: Springer-US.
- Hernandez, V., Santibañez, V., Carrillo, R., Molina, J., & López, J. (2008). Control PD de robots: Dinámica de actuadores y nueva sintonía. *Revista Iberoamericana de Automática e Informática Industrial*, 5(4), 62-68.
- IFR Statistical department. (1 de Noviembre de 2015). *IFR Statistical department*. Obtenido de World Robotics: http://www.worldrobotics.org/uploads/media/Executive_Summary__WR_2015.pdf
- ISO. (23 de 04 de 1998). ISO 9283:1998. *Manipulating industrial robots - Performance criteria and related test methods*. International Organization for Standardization.
- Kelly, R., & Santibañez, V. (2003). *Control de movimiento de robots manipuladores*. Madrid, España: Pearson Educación.
- Kennedy, J., & Eberhart, R. (2001). *Swarm intelligence*. Morgan Kaufmann Publishers.
- Loizidis, I., & Doulgeri, Z. (1995). Performance of two kinds of non-linear robust controllers for robot manipulators: a comparative study. *IEEE International Symposium of Industrial Electronics*. Atenas: IEEE.
- Lu, W.-S., & Erlic, M. (1989). Robust motion control of manipulators: Design and experimental evaluation. *Canadian Conference on Electrical & Computer Engineering*. Montreal, Canada.
- Mobile Robot Programming Toolkit*. (1 de Noviembre de 2015). Obtenido de MRPT: <http://www.mrpt.org/list-of-mrpt-apps/application-robotic-arm-kinematics/>
- Ortega, R. (1989). Adaptive motion control of rigid robots: a tutorial. *Automatica*, 25(6), 877-888.
- Ortega, R., & Garcia-Canseco, E. (2004). Interconnection and damping assignment passivity-based control: A survey. *European Journal of Control*, 432-450.

- Ortega, R., & Schaft, A. (2001). Putting energy back in control. *IEEE Transactions on Automatic Control*, 48(10), 18-33.
- Ortega, R., van der Schaft, A., Maschke, B., & Escobar, G. (2002). Interconnection and damping assignment passivity-based control of port-controlled Hamiltonian system. *Automatica*, 38(4), 585-596.
- Paim, G., Neves, L., Moreno, U., & de Pieri, E. (2011). Passivity and power based control of a scara robot manipulator. *X SBAI- Simpósio Brasileiro de Automação Inteligente*. São João del Rei: SBAI.
- Piña, A., & Arteaga, F. (2007). Estabilización del sistema Acrobot usando el enfoque IDA-PBC. *Revista Ingenieria UC*, 14, 30-40.
- Reyes, F. (2013). *MATLAB aplicado a robotica y mecatrónica*. México: Alfaomega Grupo Editor.
- Slotine, J.-J. E. (1985). The robust control of robot manipulators. *International Journal of Robotics Research*, 4, 49-63.
- Slotine, J.-J. E., & Li, W. (1988). Adaptive manipulator control: A case study. *IEEE Transactions on Automatic Control*, 33(11), 995-1003.
- Sotirov, Z. (1992). A non linear robust controller for robot manipulators. (págs. 2134-2139). Nice: 1992 IEEE International Conference on Robotics and Automation.
- Takegaki, M., & Arimoto, S. (1981). A new feedback method for dynamic control of manipulators. *ASME Journal of Dynamic Systems, Measurement, and Control*, 103(2), 119-125.
- Tomei, P. (1991). Adaptive PD controller for robot manipulators. *IEEE Transactions on Robotics and Automation*, 7, 565-570.
- Voglewede, P., Smith, A., & Monti, A. (2009). Dynamic performance of a SCARA robot manipulator with uncertainty using polynomial chaos theory. *IEEE Transactions on Robotics*, 25(1), 206-210.
- Yao, B., & Tomizuka, M. (1994). Comparative experiments of robust and adaptative control with new robust adaptative controllers for robot manipulators. *Proceedings of 33rd IEEE Conference on Decision and Control* (págs. 1290-1295). Lake Buena Vista: IEEE.
- Yepez, M. (2013). *Funciones de Lyapunov y algunas aplicaciones(Tesis de pregrado)*. Veracruz: Universidad Veracruzana.

Apéndice A

Se detalla los bloques de cinemática directa e inversa desarrollados en SIMULINK, está compuesto por funciones definidas por el usuario para facilitar el ingreso de las ecuaciones, cada bloque requiere de ciertas entradas que son necesarias para evaluar las funciones, se muestra el bloque de cinemática directa y su contenido, así como sus funciones definidas:

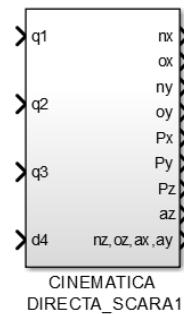


Fig. A. 1: Bloque cinemática directa en SIMULINK.

Al acceder al bloque encontramos lo siguiente:

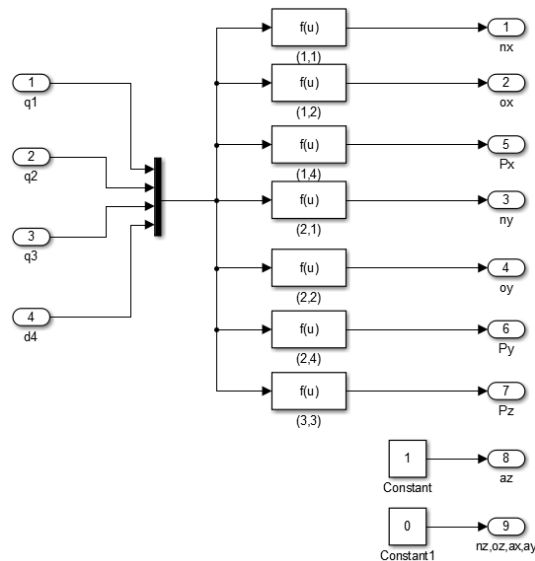


Fig. A. 2: Bloques internos en cinemática directa.

Las funciones están definidas por las relaciones dadas en (2.4.7), cabe resaltar que a_x , a_y , n_z y o_z son cero y que a_z siempre va a ser 1, debido a las limitaciones que tiene el robot, en su orientación (su orientación no cambia en el eje Z).

El bloque de cinemática inversa, tiene como entrada la matriz de orientación y de posición, y se obtiene las coordenadas generalizadas de posición tal como lo muestra la siguiente figura:

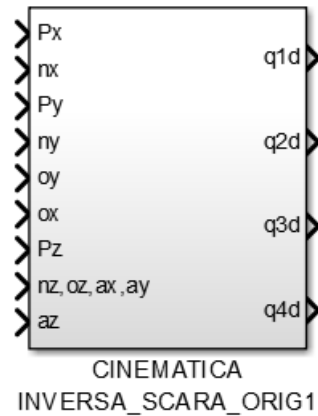


Fig. A. 3: Bloque cinemática inversa en SIMULINK.

Dentro del bloque se encuentran funciones definidas por el usuario de igual forma que la cinemática directa:

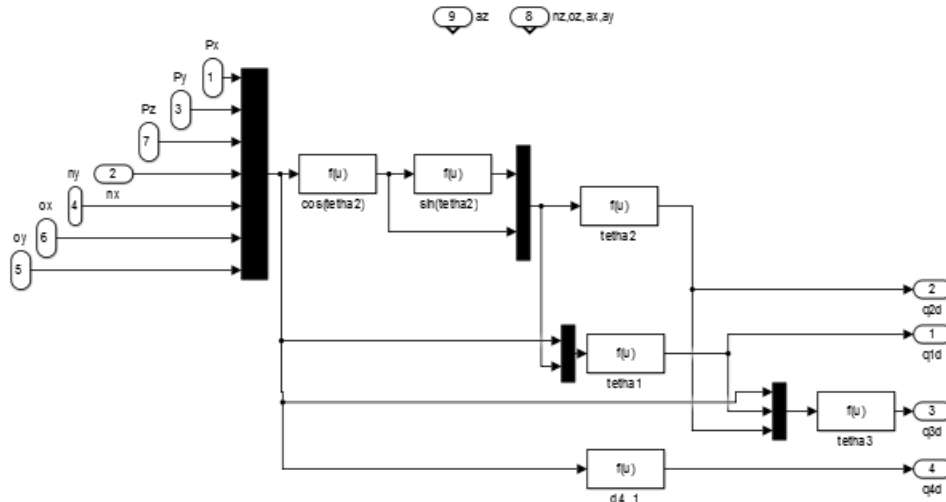


Fig. A. 4: Bloques internos en cinemática inversa.

Las ecuaciones dada por (2.5.29), (2.5.37) (2.5.40) y (2.5.41) son las definidas en sus respectivos bloques y también se hace uso de las ecuaciones (2.5.27) y (2.5.28)

Apéndice B

El modelo descrito en el subcapítulo 3.5, se programó en una *S-Function* de SIMULINK, las ecuaciones dadas por (3.5.7) se desarrollan y obtenemos:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{p}_1 \\ \dot{p}_2 \\ \dot{p}_3 \\ \dot{p}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial q_1} \\ \frac{\partial H}{\partial q_2} \\ \frac{\partial H}{\partial q_3} \\ \frac{\partial H}{\partial q_4} \\ \frac{\partial H}{\partial p_1} \\ \frac{\partial H}{\partial p_2} \\ \frac{\partial H}{\partial p_3} \\ \frac{\partial H}{\partial p_4} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \quad (\text{B.1})$$

Individualmente, tenemos las ecuaciones siguientes:

$$\dot{q}_1 = \frac{\partial H}{\partial p_1} \quad (\text{B.2})$$

$$\dot{q}_2 = \frac{\partial H}{\partial p_2} \quad (\text{B.3})$$

$$\dot{q}_3 = \frac{\partial H}{\partial p_3} \quad (\text{B. 4})$$

$$\dot{q}_4 = \frac{\partial H}{\partial p_4} \quad (\text{B. 5})$$

$$\dot{p}_1 = \frac{\partial H}{\partial q_1} + u_1 \quad (\text{B. 6})$$

$$\dot{p}_2 = \frac{\partial H}{\partial p_2} + u_2 \quad (\text{B. 7})$$

$$\dot{p}_3 = \frac{\partial H}{\partial p_3} + u_3 \quad (\text{B. 8})$$

$$\dot{p}_4 = \frac{\partial H}{\partial p_4} + u_4 \quad (\text{B. 9})$$

En donde la función H es la definida por (3.5.4), y su representación es la siguiente:

$$H(q, p) = \frac{1}{2} [p_1 \quad p_2 \quad p_3 \quad p_4] M^{-1} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} - m_4 \cdot g \cdot q_4 \quad (\text{B. 10})$$

Se capturo todas las variables en el software MAPLE 18 para facilitar su solución y se usaron las funciones *simplify* y *diff*, para obtener las derivadas parciales, su solución están escritas en el siguiente código:

```
function [sys,x0,str,ts] = SCARAIDAPBC_fs(t,x,u,flag,q10,q20,q30,q40,p10,p20,p30,p40)
%PARAMETROS DEL ROBOT SCARA
l1 = 0.50;          % Longitud del eslabon 1
l2 = 0.40;          % Longitud del eslabon 2
lc1 = 0.25;        % Longitud del centro de masa del eslabón 1
lc2 = 0.20;        % Longitud del centro de masa del eslabón 2
m1 = 15;           % Masa del eslabon 1
m2 = 12;           % Masa del eslabon 2
```

```

m3 = 3; % Masa del eslabón 3
m4 = 3; % Masa del eslabón 4 %%debe de ser 3
I1 = 0.02087*m1; % Inercia en la articulación 1
I2 = 0.08*m2; % Inercia en la articulación 2
I3 = 0.05*m3; % Inercia en la articulación 3
I4 = 0.02*m4; % Inercia en la articulación 4
g = -9.81; % Gravedad

switch flag

    %%%%%%%%%%%%%%%%%%%%%%%%%
    % Inicialización %
    %%%%%%%%%%%%%%%%%%%%%%%%%
    case 0

[sys,x0,str,ts]=mdlInitializeSizes(l1,l2,lc1,lc2,m1,m2,m3,m4,I1,I2,I3,I4,g,q10,q20,q30,q40,p10,p20,p30
,p40);

    %%%%%%%%%%%%%%%%%%%%%%%%%
    % Derivadas %
    %%%%%%%%%%%%%%%%%%%%%%%%%
    case 1
        sys=mdlDerivatives(t,x,u,l1,l2,lc1,lc2,m1,m2,m3,m4,I1,I2,I3,I4,g);

    %%%%%%%%%%%%%%%%%%%%%%%%%
    % Salidas %
    %%%%%%%%%%%%%%%%%%%%%%%%%
    case 3
        sys=mdlOutputs(t,x,u);

    %%%%%%%%%%%%%%%%%%%%%%%%%
    % Unhandled flags %
    %%%%%%%%%%%%%%%%%%%%%%%%%
    case {2 4 9}
        sys = [];

    %%%%%%%%%%%%%%%%%%%%%%%%%
    % Unexpected flags %
    %%%%%%%%%%%%%%%%%%%%%%%%%
    otherwise
        error(['Unhandled flag = ',num2str(flag)]);

end

%=====
% mdlInitializeSizes
% Tamaños del paso, condiciones iniciales, y tiempos de muestreo.
%=====

function
[sys,x0,str,ts]=mdlInitializeSizes(l1,l2,lc1,lc2,m1,m2,m3,m4,I1,I2,I3,I4,g,q10,q20,q30,q40,p10,p20,p30
,p40)

sizes = simsizes;
sizes.NumContStates = 8; %Número de variables de estado sizes.NumDiscStates = 0; %No hay estados
discretos
sizes.NumOutputs = 9; %Se contemplan 8 salidas
sizes.NumInputs = 4; %Número de entradas
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
x0 = [q10,q20,q30,q40,p10,p20,p30,p40]'; %Las condiciones iniciales se especifican por medio de
estas variables
str = [];
ts = [0 0];

%=====
% mdlDerivatives
% Serivadas para los estados continuos.
%=====

function sys=mdlDerivatives(t,x,u,l1,l2,lc1,lc2,m1,m2,m3,m4,I1,I2,I3,I4,g)

%Derivada parcial de H respecto a q1
dHq1=0;

%Derivada parcial de H respecto a q2
dHq2=-((m3+m4)*l2+lc2*m2)*(x(6)-x(7))*l1*cos(x(2))+(x(5)-
x(6))*(m3+m4)*l2^2+m2*lc2^2+I2)*(m3+m4)*l2+lc2*m2)*sin(x(2))*l1*((x(5)-
x(6))*(m3+m4)*l2+lc2*m2)*l1*cos(x(2))-...

```



```

((m3+m4+m2)*l1^2+m1*lc1^2+I1)*(x(6)-x(7))/(((m3+m4)*l2+l2*m2)^2*l1^2*cos(x(2))^2-
((m3+m4+m2)*l1^2+m1*lc1^2+I1)*((m3+m4)*l2^2+m2*lc2^2+I2))^2;

%Derivada parcial de H respecto a q3
dHq3=0;

%Derivada parcial de H respecto a q4
dHq4=-g*m4;

%Derivada parcial de H respecto a p1
dHp1=(-((m3+m4)*l2+l2*m2)*(x(6)-x(7))*l1*cos(x(2))+(x(5)-x(6))*((m3+m4)*l2^2+m2*lc2^2+I2))/(-
((m3+m4)*l2+l2*m2)^2*l1^2*cos(x(2))^2+(m3+m4+m2)*l1^2+m1*lc1^2+I1)*((m3+m4)*l2^2+m2*lc2^2+I2));

%Derivada parcial de H respecto a p2
dHp2=(-((m3+m4)*l2+l2*m2)*l1*(x(5)-2*x(6)+x(7))*cos(x(2))+(x(6)-x(7))*(m3+m4+m2)*l1^2-(x(5)-
x(6))*(m3+m4)*l2^2-l2^2*(x(5)-x(6))*m2+(lc1^2*m1+I1+I2)*x(6)+(-lc1^2*m1-I1)*x(7)-I2*x(5))/...
(-
((m3+m4)*l2+l2*m2)^2*l1^2*cos(x(2))^2+(m3+m4+m2)*l1^2+m1*lc1^2+I1)*((m3+m4)*l2^2+m2*lc2^2+I2));

%Derivada parcial de H respecto a p3
dHp3=(-((m3+m4)*l2+l2*m2)^2*l1^2*x(7)*cos(x(2))^2+(m3+m4)*l2+l2*m2)*l1*(x(5)-
x(6))*(I3+I4)*cos(x(2))+x(7)*(m3+m4)*l2^2+(l2^2*m2+I2+I3+I4)*x(7)-
x(6)*(I3+I4))*((m3+m4+m2)*l1^2+m1*lc1^2+I1))/...
((-
((m3+m4)*l2+l2*m2)^2*l1^2*cos(x(2))^2+(m3+m4+m2)*l1^2+m1*lc1^2+I1)*((m3+m4)*l2^2+m2*lc2^2+I2))*(I3+I
4));

%Derivada parcial de H respecto a p4
dHp4=(x(8)/m4);

sys(1) = dHp1;           %q1 prima
sys(2) = dHp2;           %q2 prima
sys(3) = dHp3;           %q3 prima
sys(4) = dHp4;           %q4 prima
sys(5) = -dHq1 + u(1);   %p1 prima
sys(6) = -dHq2 + u(2);   %p2 prima
sys(7) = -dHq3 + u(3);   %p3 prima
sys(8) = -dHq4 + u(4);   %p4 prima

%=====
% mdlOutputs
% Return -the block outputs.
%=====

function sys=mdlOutputs(t,x,u)
sys = [x(1),x(2),x(3),x(4),x(5),x(6),x(7),x(8),t]';%Vector de salidas del sistema
q1,q2,q3,q4,p1,p2,p3,p4,tiempo

```

Tabla B. 1: Código en MATLAB para simulación del sistema SCARA.

Apéndice C

El controlador IDA-PBC está contenido en los siguientes bloques de Fig. 5.1

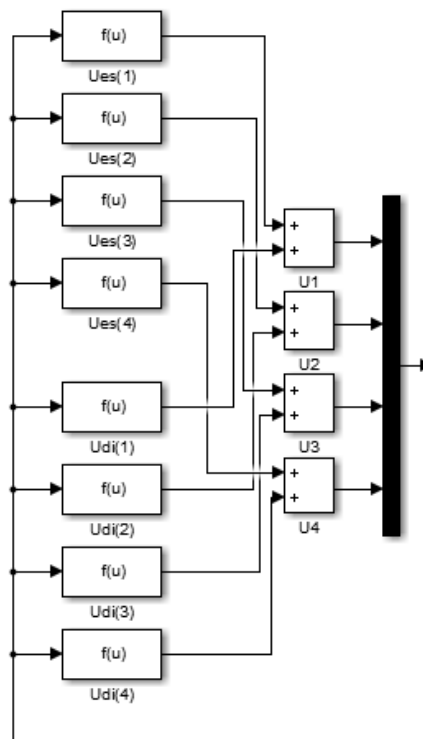


Fig. C. 1: Controlador IDA-PBC en SIMULINK.

Como se detalla en (4.2.3), cada entrada al sistema se compone de dos términos, el *energy shaping* y el *damping injection*, se observa en la figura anterior que existe un punto de suma para cada una de las entradas, en estos bloques, se encuentra la ley de control correspondiente de acuerdo a las ecuaciones (4.3.16) al (4.3.23), el diagrama completo utilizando los bloques de cinemática, el controlador y la planta es el siguiente:

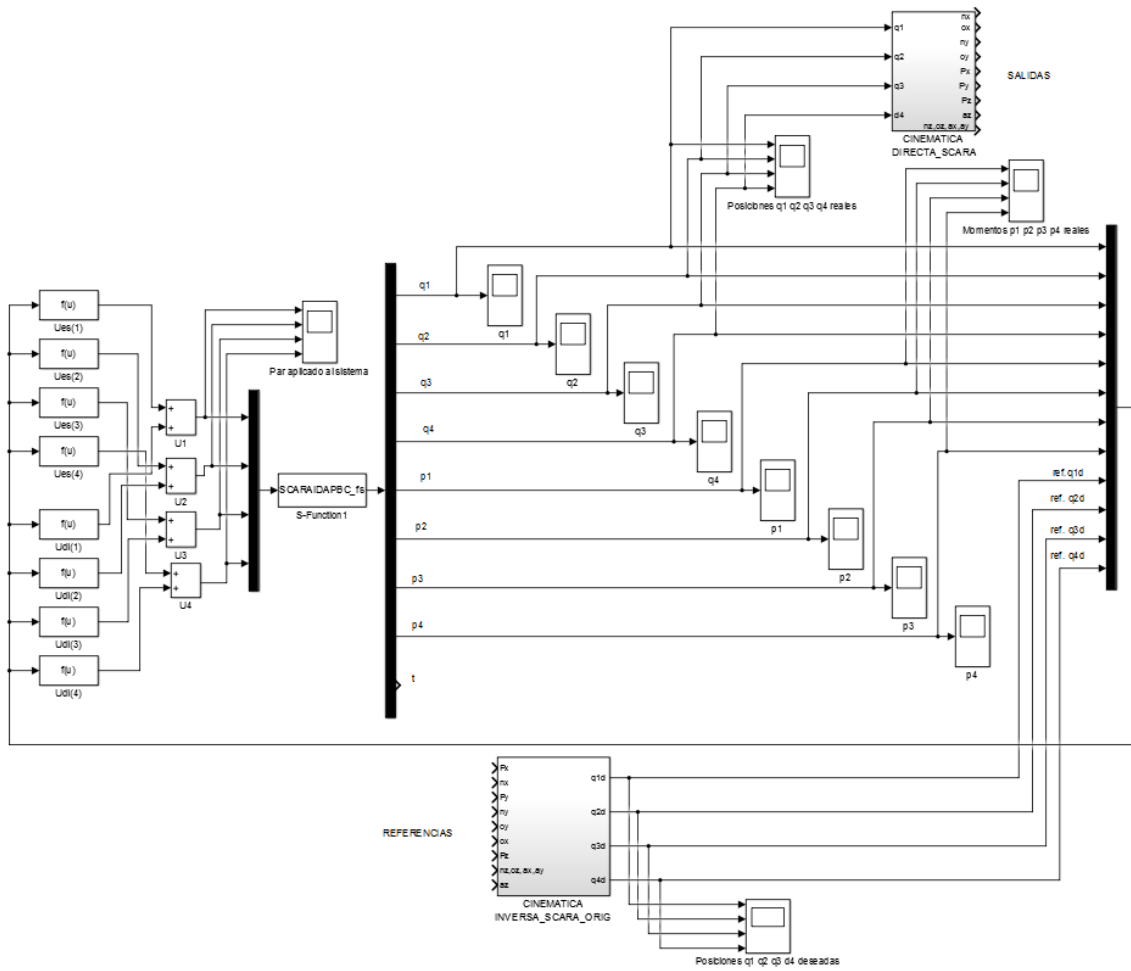


Fig. C. 2 Esquema de control con cinemáticas acopladas.

En donde la entrada de la cinemática inversa es donde colocamos nuestra posición y orientación deseada y esta ingresa al lazo cerrado en forma de coordenadas generalizadas, y la salida del bloque de cinemática directa nos da la posición del efector final y su orientación en la matriz de orientación.

El código *.m el cual contiene los parámetros de simulación del sistema y del controlador es el siguiente:

```

%parametros del robot SCARA
clear all
l1 = 0.50;          % Longitud del eslabon 1
l2 = 0.40;          % Longitud del eslabon 2
lc1 = 0.25;        % Longitud del centro de masa del eslabón 1
lc2 = 0.20;        % Longitud del centro de masa del eslabón 2
m1 = 15;           % Masa del eslabon 1
m2 = 12;           % Masa del eslabon 2
m3 = 3;            % Masa del eslabón 3
m4 = 3;            % Masa del eslabón 4 //aquí se agrega el payload
I1 = 0.02087*m1;  % Inercia en la articulación 1
    
```

```
I2 = 0.08*m2;      % Inercia en la articulación 2
I3 = 0.05*m3;      % Inercia en la articulación 3
I4 = 0.02*m4;      % Inercia en la articulación 4
g  = -9.81;% Gravedad

%parametros de cinematica
d1=.5;
d2=-.100;
d4c=-.100;
a1=l1;
a2=l2;

%parametros del controlador
Kp1=2460;
Kp2=1503;
Kp3=1593;
Kp4=1986;

Kv1=378;
Kv2=220;
Kv3=108;
Kv4=139;
```

Tabla C. 1: Código en MATLAB de parámetros del sistema en lazo cerrado y cinemática.

Apéndice D

La sintonización de ganancias por medio de PSO se realizó con el mismo esquema de control de la sección anterior, teniendo las condiciones iniciales iguales a cero y llevando a un punto fijo a regular, la diferencia es que las ganancias proporcionales se cambian en cada simulación, y existe un valor llamado *fitness*, el cual es un criterio de error que se establece por el diseñador, en este caso nuestro criterio es el sobretiro que existe en el punto deseado, se utilizó el siguiente esquema:

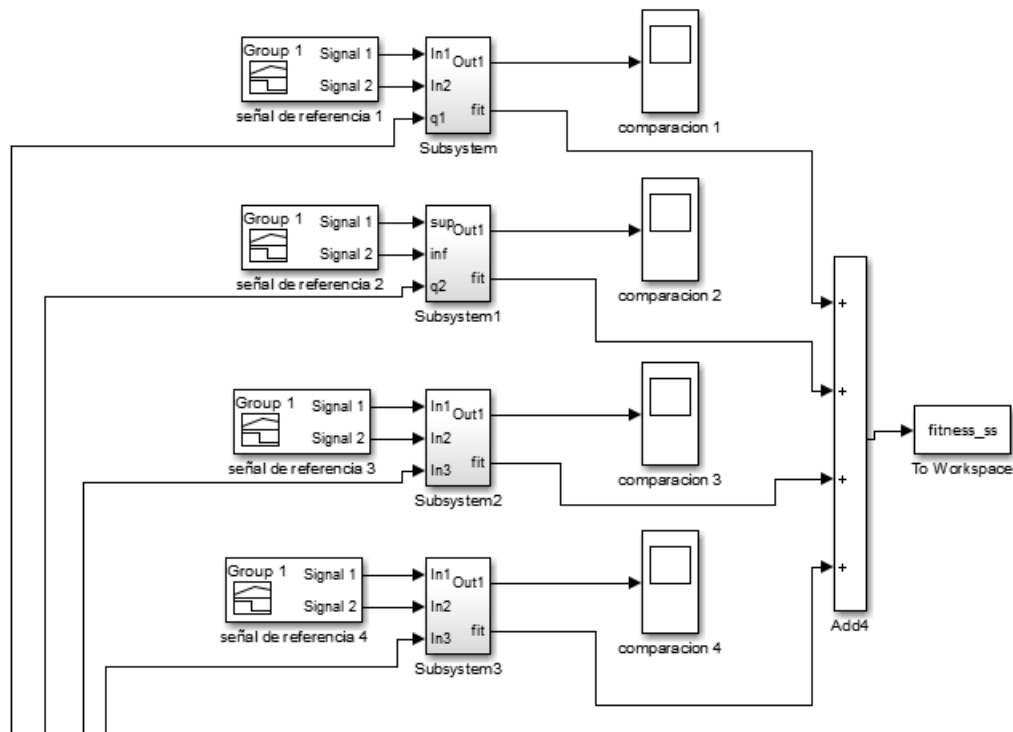


Fig. D. 1: Bloques para obtener el valor de *fitness*.

Los valores articulares para la regulación en la simulación de sintonización son: $q_{1d} = \pi/4\text{rad}$, $q_{2d} = \pi/4\text{rad}$, $q_{3d} = \pi/2\text{rad}$ y $q_{4d} = 0.1\text{m}$. Para la obtención del error, se establece una función de referencia la cual sirve para establecer los límites superior e inferior de nuestra salida deseada, la señal se construye mediante el bloque *Signal Builder*, y establecemos los valores deseados, en este caso nuestro valor será también el mismo de referencia:

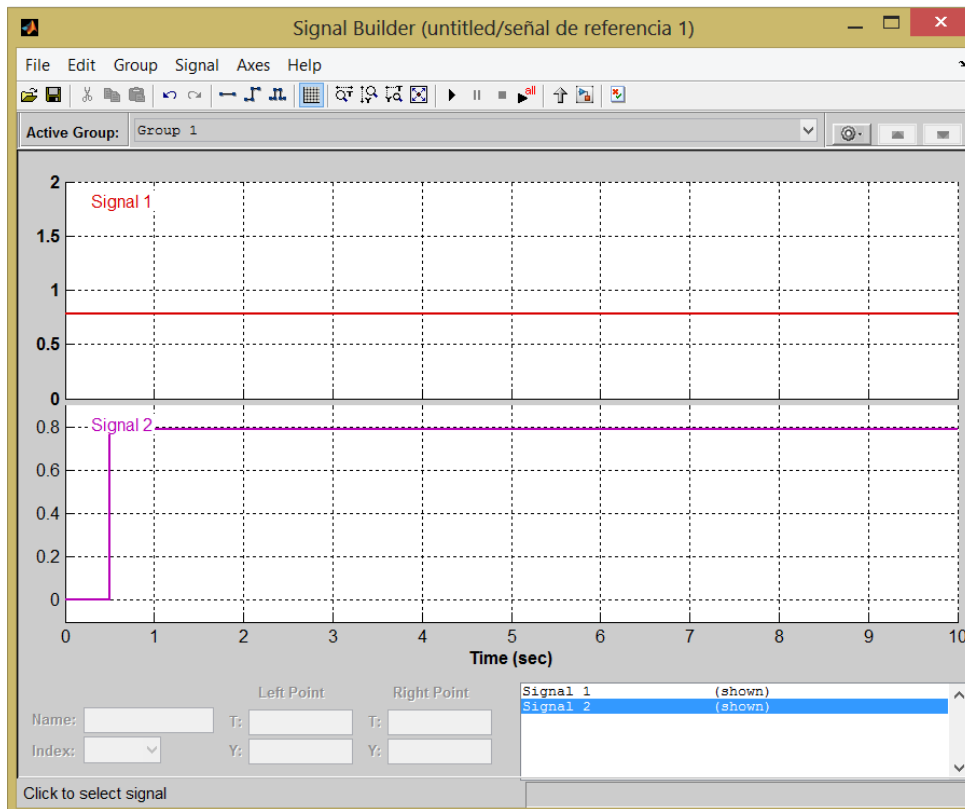


Fig. D. 2: Construcción de la señal de referencia para q_1 .

La siguiente figura muestra el ejemplo de la articulación 1, y señala las áreas que quedan fuera de los límites de la referencia, estas se suman, y ese valor es nuestro criterio *fitness*. Para cada uno de los eslabones se realiza el mismo procedimiento y al final se suman los 4 valores de *fitness*, siendo este el criterio de error para las ganancias que la búsqueda este realizando.

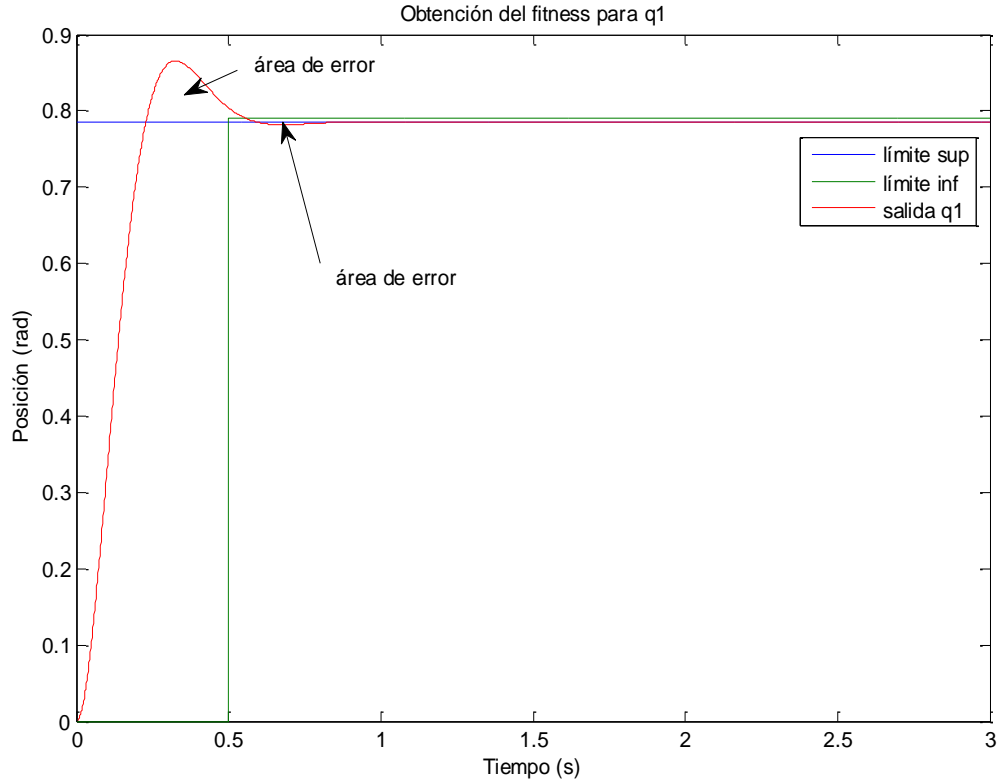


Fig. D. 3: Obtención de las áreas de error fuera de las referencias deseadas.

El bloque que realiza la comparación tiene en su interior el siguiente esquema, y en él se hace la suma de las áreas de error:

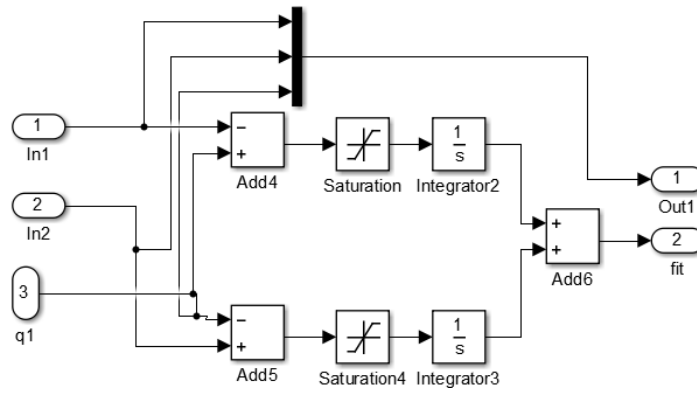


Fig. D. 4: Interior del bloque de comparación para obtener valor de *fitness*.

Se muestra a continuación el archivo *.m* en MATLAB, el cual realiza el algoritmo PSO tal como se describe en el capítulo 4.4, en él se realizan las iteraciones deseadas y se ejecuta el archivo en SIMULINK, obteniendo como retroalimentación el criterio de error.

```

clear all;
clc;
warning off;
flag=0;

C1=0.25;
C2=0.25;
W=random('unif',0.5,1);% coeficientes de inercia
par=50; %numero de particulas y creacion de las 100 particulas, acomodarlas en la matriz P
%y crear la matriz Vid con las velocidades asociadas a cada particulas
for i=1:par

Kp1(i)=random('unif',550,2500);
Kp2(i)=random('unif',500,2500);
Kp3(i)=random('unif',500,2500);
Kp4(i)=random('unif',500,2500);
Kv1(i)=random('unif',100,450);
Kv2(i)=random('unif',100,300);
Kv3(i)=random('unif',50,500);
Kv4(i)=random('unif',100,500);
Ka4(i)=random('unif',0,1000);

end
for i=1:par
    P(i,:)= [Kp1(i) Kp2(i) Kp3(i) Kp4(i) Kv1(i) Kv2(i) Kv3(i) Kv4(i) Ka4(i)];
end;
for i=1:par
    Vid(i,:)= [random('unif',-10,10) random('unif',-10,10) random('unif',-10,10) random('unif',-10,10)
random('unif',-10,10) random('unif',-10,10) random('unif',-10,10) random('unif',-10,10)
random('unif',-10,10)];
end
%hasta aqui la creacion de 50 particulas con posiciones y velocidades aleatorias P y Vid

for itera=1:1000%hacer mil iteraciones

subplot(2,2,1);
plot(P(:,1),P(:,5),'o');
axis square;
axis ([-0 3000 -0 1000]);
title('Ganancias Kp1 y Kv1');
xlabel('Kp1');
ylabel('Kv1');
drawnow;
hold off;

subplot(2,2,2);
plot(P(:,2),P(:,6),'o');
axis square;
axis ([-0 3000 -0 1000]);
title('Ganancias Kp2 y Kv2');
xlabel('Kp2');
ylabel('Kv2');
drawnow;
hold off;

subplot(2,2,3);
plot(P(:,3),P(:,7),'o');
axis square;
axis ([-0 3000 -0 1000]);
title('Ganancias Kp3 y Kv3');
xlabel('Kp3');
ylabel('Kv3');
drawnow;
hold off;

subplot(2,2,4);
plot(P(:,4),P(:,8),'o');

```



```

axis square;
axis ([-0 3000 -0 1000]);
title('Ganancias Kp4 y Kv4');
xlabel('Kp4');
ylabel('Kv4');
drawnow;
hold off;

pause(5.0);

%ahora evaluamos el fitness de cada particula

for j=1:par %
    Kp1=P(j,1);
    Kp2=P(j,2);
    Kp3=P(j,3);
    Kp4=P(j,4);
    Kv1=P(j,5);
    Kv2=P(j,6);
    Kv3=P(j,7);
    Kv4=P(j,8);
    Ka4=P(j,9);
    simout=sim('untitled');%aquí va el archivo de simulacion
    e=max(fitness_ss);%% aquí va el error
    Y(j,:)=[Kp1 Kp2 Kp3 Kp4 Kv1 Kv2 Kv3 Kv4 Ka4 e];
end

%ordenamos la matriz de menor fitness a mayor fitness y sacamos el Gbest de
%esta evaluacion
Yt=Y';
Ytant=Yt;
[Xf Yf]=sort(Yt(10,:));
Yord=Yt(:,Yf);
Gbest=[Yord(1,1) Yord(2,1) Yord(3,1) Yord(4,1) Yord(5,1) Yord(6,1) Yord(7,1) Yord(8,1) Yord(9,1)];
%%%evaluacion de las particulas

%primera iteracion Pbest es igual a la misma particula a evaluar
if flag==0
    Pbest=P;
else
    for i=1:par
        if Yt(10,i)<Ytant(10,i)
            Pbest(i,:)=[Yt(1,i) Yt(2,i) Yt(3,i) Yt(4,i) Yt(5,i) Yt(6,i) Yt(7,i) Yt(8,i) Yt(9,i)];
        else Pbest(i,:)=[Ytant(1,i) Ytant(2,i) Ytant(3,i) Ytant(4,i) Ytant(5,i) Ytant(6,i) Ytant(7,i)
            Ytant(8,i) Ytant(9,i)] ; end
    end
end

%evaluacion de las particulas de acuerdo a la ecuacion que define al PSO
for i=1:par
    V=W*[Vid(i,1) Vid(i,2) Vid(i,3) Vid(i,4) Vid(i,5) Vid(i,6) Vid(i,7) Vid(i,8) Vid(i,9)] +
        C1*([Pbest(i,1) Pbest(i,2) Pbest(i,3) Pbest(i,4) Pbest(i,5) Pbest(i,6) Pbest(i,7) Pbest(i,8)
            Pbest(i,9)]-[P(i,1) P(i,2) P(i,3) P(i,4) P(i,5) P(i,6) P(i,7) P(i,8) P(i,9)]) + C2*(Gbest-[P(i,1)
            P(i,2) P(i,3) P(i,4) P(i,5) P(i,6) P(i,7) P(i,8) P(i,9)]);
    P(i,:)=P(i,:)+V; %se actualizo la matriz P sumandole a cada punto el V
    Vid(i,:)=V;%se actualizo la matriz Vid con V

    for i=1:par
        for j=1:9
            if P(i,j)<0
                P(i,j)=100*rand(1);
            end
        end
    end
end

end
flag=1;
clc
P
end

```

Tabla D. 1: Código en MATLAB con algoritmo PSO de sintonización.

Tras la ejecución del código, se puede observar en la siguiente gráfica la población inicial de partículas, que comprende para cada uno de los grados de libertad.

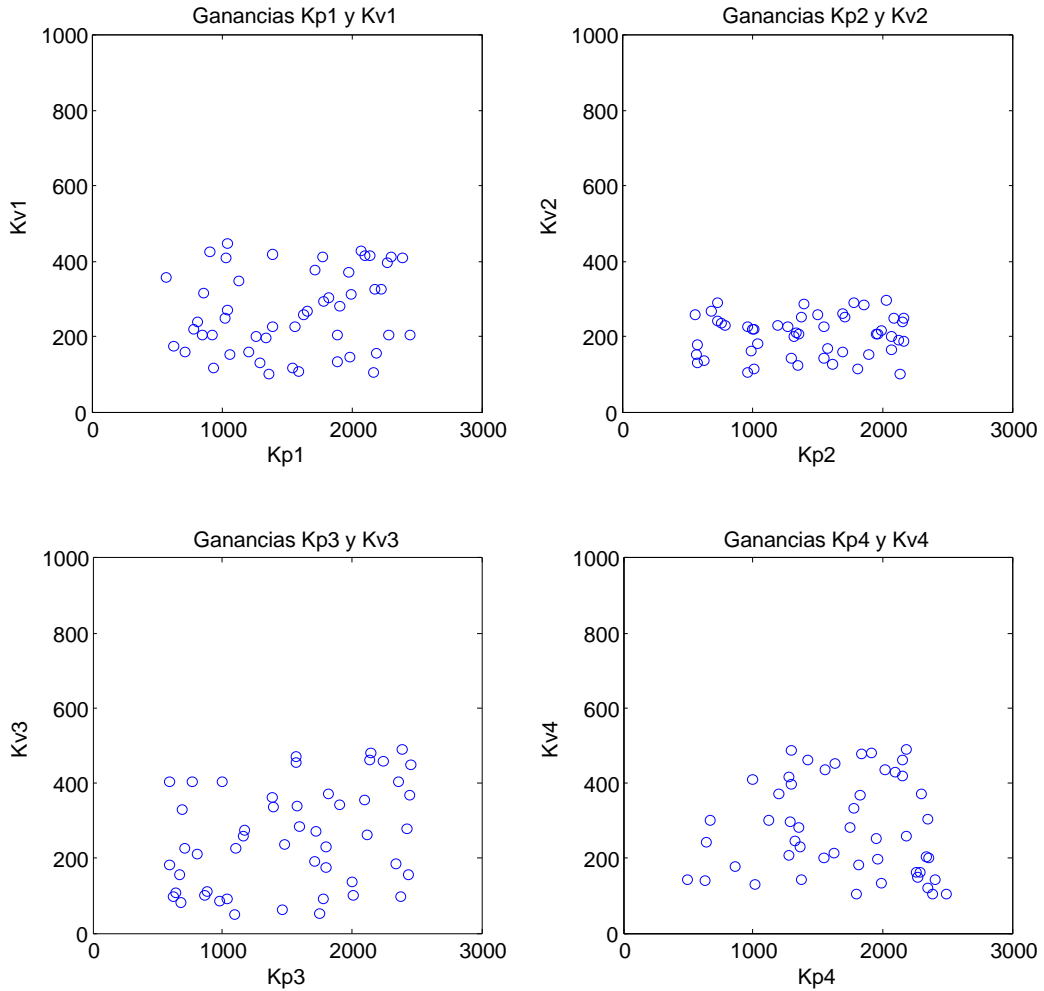


Fig. D. 5: Población inicial de partículas en PSO.

Al término del algoritmo, se puede observar la convergencia de las partículas, las cuales son las correspondientes a la Tabla 4.2.

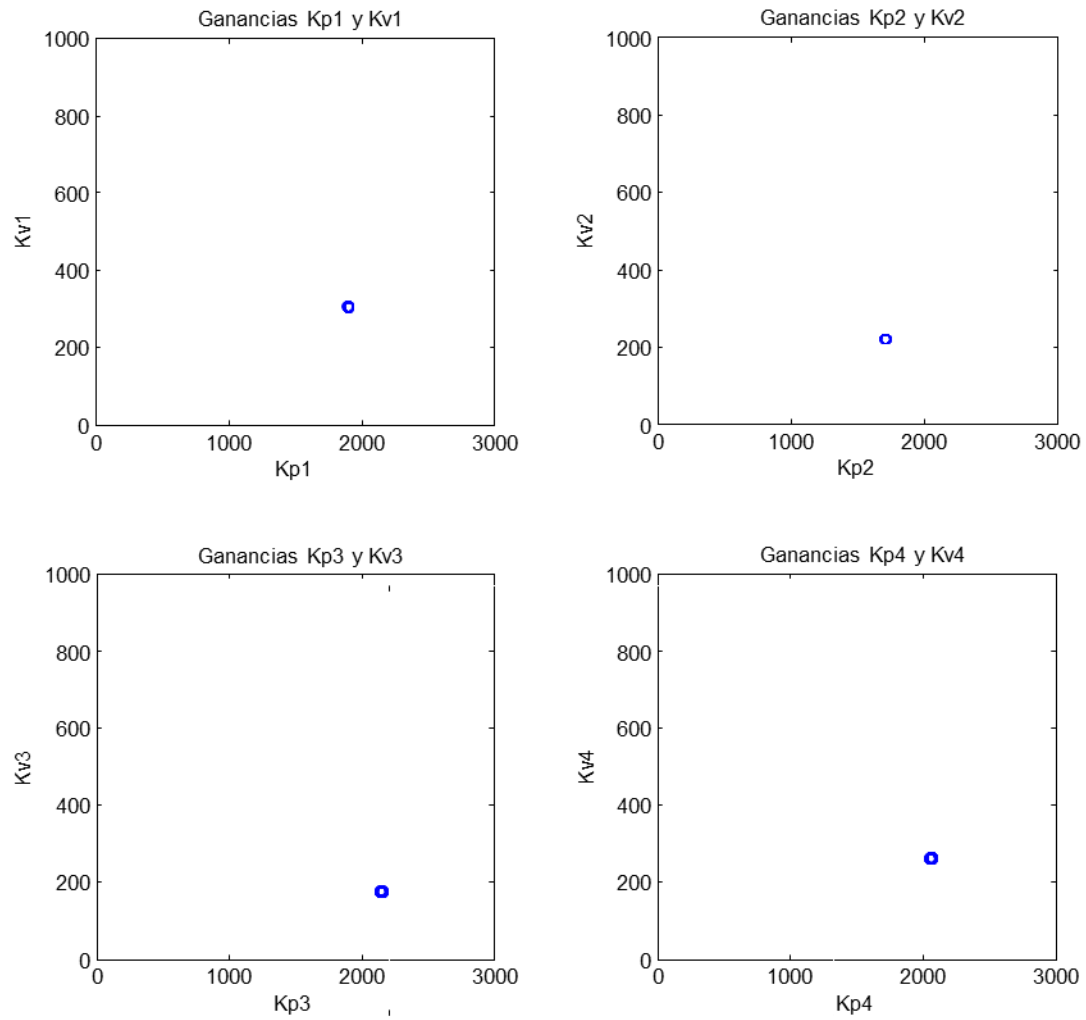


Fig. D. 6: Convergencia de los valores al finalizar el algoritmo PSO.