



01173
1
**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERIA

DIVISIÓN DE ESTUDIOS DE POSGRADO

SECCIÓN MECÁNICA

**MODELACIÓN Y PRUEBA DE SECUENCIAS DE
MOVIMIENTOS DE UN ROBOT DE DOS GRADOS DE
LIBERTAD PARA UN PROBLEMA DE EVASIÓN DE
OBSTÁCULOS EN EL PLANO**

TESIS

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN INGENIERÍA MECÁNICA

PRESENTA:

ING. OCIEL FLORES DÍAZ

ASESOR: M.I. EUSEBIO JIMÉNEZ LÓPEZ

COASESOR: DR. LUIS REYES ÁVILA

Ciudad Universitaria, agosto del 2003



A





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

CONTENIDO

Resumen

Prólogo

Lista de símbolos

Introducción

i	Importancia del tema.....	1
ii	Definición del problema.....	2
iii	Objetivos general y específicos.....	2
iv	Hipótesis.....	3
v	Justificación del proyecto.....	4

Capítulo 1 Generalidades en la robótica

1.1	Antecedentes y el futuro desarrollo de la robótica.....	5
1.2	Clasificación de los robots.....	7
1.3	Elementos componentes del robot.....	8

Capítulo 2 Modelación cinemática del robot de 2GDL

2.1	Caracterización del multicuerpo R_0	10
2.2	Modelación cinemática de R_0 utilizando la rotación "ROT1".....	12
2.2.1	Análisis en la configuración no deformada.....	13
2.2.1.1	Planteamiento de los problemas cinemático directo e inverso.....	14
2.2.2	Análisis de la configuración deformada.....	16
2.2.2.1	Planteamiento de los problemas cinemático directo e inverso.....	18
2.3	Modelación cinemática de R_0 utilizando la reflexión "REF 1".....	19
2.3.1	Análisis de la configuración reflectada no deformada.....	19
2.3.1.1	Planteamiento de los problemas cinemático directo e inverso.....	21
2.3.2	Análisis de la configuración reflectada deformada.....	22
2.3.2.1	Planteamiento de los problemas cinemático directo e inverso.....	24

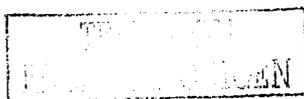


Capítulo 3 Caracterización de los movimientos del robot de 2GDL

3.1 Definición del problema y restricciones.....	25
3.2 Hipótesis y consideraciones importantes.....	26
3.3 Movimientos uno a uno.....	27
3.3.1 Caracterización de los parámetros de rotación.....	28
3.3.2 Criterios de activación.....	29
3.3.3 Tipos de movimientos uno a uno.....	30
3.3.3.1 Movimientos uno a uno reales.....	31
3.3.3.2 Movimientos uno a uno complementarios.....	34
3.3.3.3 Movimientos uno a uno combinados.....	36
3.3.4 Análisis y caracterización de los movimientos uno a uno completos.....	36
3.4 Movimientos simultáneos.....	40
3.4.1 Movimientos simultáneos reales.....	41
3.4.2 Caracterización de movimientos simultáneos reales de la familias 1 y 2	43
3.4.3 Características importantes de los movimientos simultáneos reales.....	44
3.4.4 Movimientos simultáneos complementarios y combinados F_0	44
3.4.4.1 Consideraciones en los movimientos complementarios y combinados...	46
3.4.5 Relaciones generales de los movimientos simultáneos.....	46
3.5 Movimientos híbridos.....	47

Capítulo 4 El problema de evasión de obstáculos en el plano

4.1 Definición del problema y restricciones.....	49
4.2 Hipótesis y consideraciones sobre el problema de evasión de obstáculos.....	51
4.3 La condición de impacto y diferentes tipos de impactos.....	52
4.4 Sensores, regiones de barrido, trayectoria de trabajo y localización de obstáculos.....	55
4.5 Representación teórica del impacto y distancias de seguridad.....	58
4.5.1 Modelación de los impactos de sensado.....	61
4.5.2 Distancias de seguridad.....	63
4.5.3 Impacto por movimientos completos.....	64



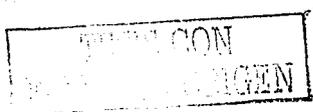
4.6 Jerarquización de movimientos completos por el tiempo total de desplazamiento angular.....	66
4.7 Definición de los elementos participantes en el evento $R_0^{IN} \cap O_H \neq \emptyset$ y los criterios de evasión.....	68
4.7.1 Elementos participantes en el proceso de evasión.....	72
4.7.2 Los criterios de evasión.....	72

Capítulo 5 Construcción, dispositivos electrónicos y programación del prototipo experimental de 2GDL para la evasión de obstáculos en el plano

5.1 El sistema mecánico.....	75
5.1.1 Descripción del robot de 2GDL para la evasión de obstáculos.....	76
5.2 Los motores de pasos y los dispositivos electrónicos del manipulador.....	77
5.2.2. La computadora personal.....	82
5.2.2.1 El puerto paralelo externo.....	83
5.2.3 Los sensores.....	86
5.2.3.1 Sensores utilizados en el proyecto.....	89
5.3. El programa de computación.....	94
5.3.1 Análisis del problema.....	95
5.3.1.1 Definición del problema.....	95
5.3.1.2 Especificaciones de entrada y salida.....	95
5.3.1.3 El modelo matemático.....	96
5.3.2 Diseño del algoritmo.....	96
5.3.2.1 Modularización.....	96
5.3.3 La programación.....	97
5.3.3.1 Diseño y codificación del programa.....	98
5.3.3.2 Ejecución y depuración de EvaSoft 1.0.....	117
5.3.4 Prueba y mantenimiento de EvaSoft 1.0.....	117



Capítulo 6 Resultados obtenidos de la modelación y prueba de secuencias de movimientos del robot de dos grados de libertad para la evasión de obstáculos	
6.1 Tipos de movimientos.....	118
6.2 Instalación.....	119
6.2.1 Requerimientos del programa.....	119
6.2.2 Instalación del hardware.....	119
6.2.3 Instalación de EvaSoft 1.0.....	119
6.2.4 Desinstalación de EvaSoft 1.0.....	119
6.3 El programa EvaSoft 1.0.....	120
6.4 Prueba de funcionamiento de EvaSoft 1.0.....	120
6.4.1 Resultados en el manejo de la forma de movimientos uno a uno de ejes....	122
6.4.2 Resultados en el manejo de la forma de movimientos simultáneos e híbridos de ejes.....	125
6.4.3 Resultados en el manejo de la forma de movimientos simultáneos que siguen una ruta.....	128
6.4.4 Resultados en el manejo de la forma para la evasión de obstáculos en el plano.....	130
Conclusiones y recomendaciones.....	134
Referencias bibliográficas.....	136
Apéndice A Dibujos de los elementos mecánicos y fotografías del prototipo de robot de 2GDL.....	139
Apéndice B Código fuente del programa EvaSoft 1.0.....	160



Resumen

En este trabajo de tesis se presenta el modelado cinemático de posición de un robot de dos grados de libertad (2GDL). Son analizados y caracterizados los movimientos completos del robot, es decir, los movimientos uno a uno, simultáneos e híbridos. Dicho modelo fue construido usando números complejos. Fue generado un marco teórico para analizar y modelar un problema de evasión de obstáculos en el plano caracterizado por el robot antes mencionado y un obstáculo circular.

Fueron utilizados los movimientos del robot para generar criterios de evasión. Los modelos de los movimientos, así como el asociado con el problema de evasión fueron programados y simulados en el lenguaje de programación DELPHI 5. Finalmente, se construyó un prototipo del robot para probar los modelos obtenidos.

Palabras clave: Modelación, movimientos, evasión, programación.



Prólogo

Este trabajo de tesis tiene el propósito de generar conocimiento escrito para las futuras generaciones de la maestría en Ingeniería Mecánica y campos afines, interesados en la modelación y simulación de problemas de evasión de obstáculos.

Generalmente en las industrias se presentan problemas de colisión ya sea entre piezas y robots o robots y operarios. Cada vez que en una línea de producción se presenta una colisión de cualesquier tipo, se generarán pérdidas económicas, pues es posible que se detenga toda la línea de producción o, en su caso, se avería un robot o, en el peor de los casos, una persona se accidente.

Para poder solucionar los problemas de colisión entre un robot y un obstáculo, de tal manera que las líneas de producción sigan operando, es necesario construir simuladores y prototipos con el fin de probar diversas situaciones de evasión. Para generar los simuladores se requieren construir modelos y, estos a su vez, tienen que ser probados en prototipos.

Es por ello que en este trabajo de tesis se analiza, modela y simula un problema de evasión de obstáculos en el plano caracterizado por un robot de dos grados de libertad tipo RR y un obstáculo circular. Los modelos son probados en un prototipo de robot dotado de elementos de sensado.

Este trabajo de tesis está dividido en los apartados de introducción, seis capítulos, conclusiones y recomendaciones, y dos secciones de apéndices.

La introducción consta de los siguientes puntos: importancia del tema, definición del problema, definición del objetivo general y objetivos específicos, planteamiento de la hipótesis y finalmente la justificación del proyecto.



El capítulo 1 presenta algunas generalidades de la robótica, las cuales son los antecedentes y el futuro desarrollo de ésta, la clasificación de los robots de acuerdo a órganos internacionales, los componentes y características de éstos.

En el capítulo 2 se realiza el análisis cinemático del manipulador de 2GDL con números complejos, específicamente el análisis de posición para las configuraciones no deformadas y deformadas.

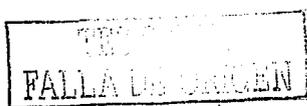
El capítulo 3 trata sobre la caracterización de los movimientos del manipulador de 2GDL. En el cual se realiza el análisis y la caracterización de los movimientos uno a uno, simultáneos e híbridos.

El capítulo 4 presenta el problema de evasión de obstáculos en el plano, sus consideraciones y sus restricciones, formas de representación teórica del robot, del impacto y distancias de seguridad, y finalmente los criterios de evasión.

El capítulo 5, en la sección uno, trata sobre la construcción mecánica del dispositivo manipulador de 2GDL, en la sección dos sobre los dispositivos electromecánicos y electrónicos que se utilizan en dicho prototipo experimental y en la sección tres trata sobre el programa de computación, en los cuales se presenta principalmente los diagramas de flujo.

En el capítulo 6 se presentan los resultados obtenidos de la integración del prototipo experimental y el programa de computación elaborado para describir los movimientos uno a uno, simultáneos, híbridos y el problema de evasión haciendo uso de dichos movimientos.

Finalmente, se presentan las conclusiones obtenidas de este trabajo de tesis. Cabe hacer mención que se presentan dos secciones de apéndices. En el apéndice A se presentan los planos de pieza de los elementos mecánicos y fotografías del prototipo de robot de 2GDL; el apéndice B muestra el código fuente del programa realizado en este trabajo, el cual se le nombró como Evasoft I.



Lista de símbolos

Símbolo	Descripción
R_0	Robot de 2 grados de libertad en estudio.
R_0^{ND}	Configuración no deformada de R_0 .
R_0^D	Configuración deformada de R_0 .
R_0^D	Configuración deformada intermedia de R_0 .
R_0^{DS}	Robot en movimiento equipado con sensores.
R_0^{DS}	Robot con sensores en la configuración no deformada.
R_0^{DS}	Robot con sensores en la configuración deformada.
R_0^{DS}	Robot con sensores con la distancia de seguridad efectiva.
R_0^{DS}	Robot con sensores con la configuración llamada de "impacto en movimiento".
∂R_0	Frontera de R_0 .
$\vec{\partial} R_0^D$	Frontera móvil de R_0^D .
IR_0	Interior de R_0 .
T	Transición de movimiento: de la configuración no deformada a la deformada.
R (pot)	Vector que va del origen al órgano terminal.
R (pot) ^{REF}	Vector que va del origen al órgano terminal reflejado.
E_i	Eslabón i.
φ_1, φ_1^*	Desplazamiento angular real (positivo) y complementaria respectivamente.
F_k, F_l	Formas de activación angular primaria y secundaria respectivamente.
τ	Lugar geométrico de la trayectoria que recorre el robot.
MR^{F, M_j}	Movimiento real de la familia i con activación del actuador M_j en T_j .
MCP^{F, M_j}	Movimiento complementario de la familia i con activación del actuador M_j en T_j .
$MC^{F, M_j}_{M_k}$	Movimiento combinado de la familia i con subcomplementación en M_k y actuador M_j en T_j .
$MCS^{F_0}_{F_0^* Q_0}$	Movimiento combinado simultáneo familia F_0 con complementación en $F_0^*(M_1)$.
$MCS^{F_0}_{F_0^* Q_0^*}$	Movimiento combinado simultáneo familia F_0 con complementación en $Q_0^*(M_2)$.



MRH^{E,M_i}	Movimiento híbrido real de la familia i con activación del actuador M_j en T_j .
$MCPH^{E,M_i}$	Movimiento híbrido complementario de la familia i con activación del actuador M_j en T_j .
MCH^{E,M_i}	Movimiento híbrido combinado de la familia i con activación del actuador M_j en T_j .
Γ_0, Γ_1	Estado inicial y final del actuador 1.
δ_0, δ_1	Estado inicial y final del actuador 2.
C_M	Conjunto de movimientos de R_0 .
A_i	Área de trabajo de R_0 .
O_R	Obstáculo.
$r_{0(O_R)}$	Radio del obstáculo.
∂O_R	Frontera del obstáculo.
S	Conjunto de sensores.
S_i^A	Sensor i activado.
S_i^D	Sensor i desactivado.
C_c	Conjunto de criterios de evasión.
l_{se}	Distancia de seguridad efectiva.
l_s	Distancia que hay de la frontera de los eslabones al sensor.
l_p	Distancia de penetración desde el momento en que el sensor detecta el obstáculo hasta que el robot se para.
T_{TMU}	Tiempo total de los movimientos uno a uno.
T_{TMS}	Tiempo total de los movimientos simultáneos.
T_{TMH}	Tiempo total de los movimientos híbridos.
T_{im_i}	Tiempo total de accionamiento del i -ésimo motor.
z_i	Número de dientes de la polea dentada i .
Dm_i	Ángulo de avance del motor m_i por pulso.
R_i	Relación de transmisión entre poleas dentadas.
P_m	Pulsos enviados al motor i .
R_p	Relación de envío de pulsos de la computadora hacia los motores de pasos.

Introducción

i Importancia del tema

En diversas actividades industriales se presentan problemas de impacto entre robots y obstáculos, e inclusive con operarios. Actualmente se han desarrollado trabajos que están relacionados con la evasión de obstáculos para tratar de evitar impactos. Entre ellos se encuentra el método de los campos potenciales, en el cual el robot tiene un campo potencial artificial atractivo y el obstáculo tiene un campo potencial artificial repulsivo. Y en el momento de entrar en contacto estos campos se genera un lugar geométrico, en el cual el robot está libre de obstáculos.¹³ En otro trabajo llamado "planeación de trayectorias óptimas para un robot del tipo rotacional de n-eslabones ($n = 2,3$) en presencia de obstáculos" se propone un método para la modelación de obstáculos, a través del método se obtienen expresiones de la forma cerrada del obstáculo que se encuentran en términos de las coordenadas generalizadas del manipulador. Dichas expresiones cerradas del obstáculo son obtenidas porque se considera el obstáculo como una composición de objetos primitivos tal como el círculo, la elipse, entre otros. Una vez creadas las expresiones cerradas del obstáculo y referenciadas en términos de las coordenadas generalizadas del manipulador, se genera un área en la cual éste no puede pasar y de esta forma evitar ciertas configuraciones, a partir de esto se procede a obtener el área que está libre de obstáculos y se obtienen las configuraciones que si puede tomar dicho manipulador para realizar su tarea.²²

Estos trabajos muestran metodologías implementadas en robots manipuladores para la modelación de obstáculos y generar lugares geométricos que estén libre de éstos. Sin embargo, todavía existen ciertos vacíos en la implementación de estos métodos para la evasión de obstáculos. Es por ello que este trabajo de tesis pretende llevar a cabo una metodología para la evasión de obstáculos mediante rotaciones y reflexiones, así como la utilización de las secuencias de movimientos; el obstáculo es modelado como un objeto primitivo circular.



Para poder implementar el algoritmo, se construirá un robot prototipo de dos grados de libertad para realizar la evasión de obstáculos mediante las secuencias de movimiento: uno a uno, simultáneas e híbridas.

ii Definición del problema

Analizar, modelar, programar y probar las secuencias de movimientos de un robot y aplicar dichos movimientos a un problema de evasión de obstáculos.

Las restricciones del problema son:

- 1) El robot es de dos grados de libertad (2GDL) del tipo rotacional-rotacional (RR).
- 2) El obstáculo es de geometría circular.
- 3) El proceso de evasión es el plano (x, y).
- 4) Una vez localizado el obstáculo se fija en su posición.
- 5) El problema de evasión es de posicionamiento.
- 6) Las pruebas de los movimientos del robot se realiza en un simulador y un prototipo.
- 7) El proceso de evasión es determinístico.

iii Objetivos

iii.1 Objetivo general

En este trabajo de tesis se pretende analizar, modelar, programar y probar las secuencias de movimiento de un robot de 2GDL . Así como aplicar dichos movimientos a un problema para evadir obstáculos en el plano (x, y).

iii.2 Objetivos específicos

- 1) Encontrar las secuencias de movimientos existentes en el robot de estudio.
- 2) Construir un marco teórico y generar los criterios para la evasión de obstáculos en el plano.
- 3) Graficar los movimientos uno a uno, simultáneos e híbridos de dicho robot utilizando el lenguaje de programación DELPHI 5.
- 4) Programar los algoritmos para la evasión de obstáculos en el lenguaje de programación mencionado en 3).



PAGINACIÓN DISCONTINUA

- 5) Construir el manipulador y armar los circuitos para el control de los motores de pasos para el acondicionamiento de los sensores del manipulador de 2GDL.
- 6) Probar los tipos de movimientos y los algoritmos generados para la evasión de obstáculos en el plano con el prototipo experimental.

iv Hipótesis

Existe un conjunto de movimientos asociados con un robot. Dicho conjunto de movimientos depende del número de GDL del robot. Además, en un proceso de evasión existe un elemento que evade y otro elemento que es evadido. El proceso de evasión del obstáculo se puede realizar con los movimientos asociados del elemento que evade.

Premisas

- 1) El movimiento del robot y el obstáculo satisfacen las leyes de Newton.
- 2) Cada actuador satisface lo siguiente: $\varphi_i + \varphi_i^* = 360$ o $P_0 + P_0^* = 1$.
- 3) $R_0^{MS} \cap O_H = \phi$ o $R_0^{MS} \cap O_H \neq \phi$

Alcances

- 1) De acuerdo con la premisa uno, se debe de partir con las leyes de movimiento de Newton, las cuales relacionan las fuerzas con los movimientos, sin embargo, en este trabajo de tesis sólo se considerará el movimiento, específicamente los desplazamientos angulares del robot.
- 2) La segunda restricción indica que el desplazamiento total en una revolución puede ser particionado en dos desplazamientos $\varphi_i + \varphi_i^* = 360^\circ$. La rotación descrita en la premisa 2 se usa para caracterizar los tipos de movimientos del robot y el proceso de evasión de obstáculos.
- 3) Las relaciones descritas en la premisa tres se usan para representar el impacto entre el robot con sensores y el obstáculo.



v Justificación del proyecto

Durante el desarrollo de la ciencia y la tecnología cada vez se buscan nuevos conocimientos para poder enriquecer a éstas. La robótica está incluida en esta forma de desarrollo. Es por ello que en la robótica actualmente se están desarrollando prototipos que sean capaces de detectar objetos y, posteriormente, tratar de evadirlos de tal manera que realicen la tarea programada (siempre y cuando la tarea lo permita). Aquí existen dos puntos interesantes:

- 1) El obstáculo puede ser el propio operario que se encuentra en el área de trabajo del robot, es decir, lo que se busca es tratar de evitar accidentes de tal manera que no dañe al operario.
- 2) Existen diferentes maneras de detectar objetos, así como diferentes algoritmos propuestos para modelar y evadir obstáculos. Lo que se utilizará en este trabajo, para detectar objetos, son sensores ubicados en los costados de los eslabones, esto es, porque el robot se mueve de manera rotacional en el plano xy, y lo más probable y riesgoso es que se encuentre un obstáculo en los costados de los eslabones. En lo referente al método que se utilizará para realizar la evasión del obstáculo, es a través de la reflexión y las secuencias de movimientos, en el cual el obstáculo es fijo.



Capítulo 1

Generalidades en la robótica

Introducción

En el presente capítulo se realiza una descripción de puntos de interés general relacionados con la robótica. Es decir, sus antecedentes y el futuro desarrollo en esta área, también se describe la clasificación de los robots de acuerdo con ciertas instituciones relevantes en dicha área, así mismo se describen los componentes principales de los robots, y la forma en que éstos interactúan.

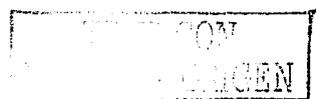
1.1 Antecedentes y el futuro desarrollo de la robótica

La palabra robot fue usada por primera vez en el año de 1921, cuando el escritor checo Karel Kapek (1890-1938) estrena en el teatro nacional de Praga su obra *Rossum's Universal Robot* (R.U.R.). Su origen es la palabra eslava *robot*, que se refiere al trabajo realizado de manera forzada y posteriormente se tradujo al inglés como *robot*.²

Los robots (de la R.U.R.) eran máquinas androides fabricadas a partir de la "fórmula" obtenida por un brillante científico llamado *Rossum*. Estos robots servían a sus jefes humanos desarrollando todos los trabajos físicos, hasta que finalmente se revelan contra sus dueños, destruyendo toda la vida humana, a excepción de uno de sus creadores, con la frustrada esperanza de que les enseñe a reproducirse.

Isaac Asimov (1920-1992) también contribuyó con varias narraciones relativas a robots, a él se le atribuye el acuñamiento del término "robotics" (Robótica). En 1945 publicó en la revista *Galaxy Science Fiction* una historia en la que por primera vez enunció sus *tres leyes de la robótica*:^{1,16}

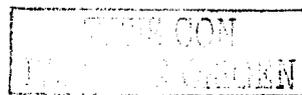
- 1) Un robot no puede perjudicar a un ser humano, ni con su inacción permitir que un ser humano sufra daño.
- 2) Un robot ha de obedecer las órdenes recibidas de un ser humano, excepto si tales órdenes entran en conflicto con la primera.



- 3) Un robot debe de proteger su existencia mientras tal protección no entre en conflicto con la primera o segunda ley.

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

Los futuros desarrollos de la robótica apuntan a aumentar su movilidad, destreza y autonomía de sus acciones. La mayor parte de los robots actuales tienen base estática y se utilizan en aplicaciones industriales tales como ensamblado, soldadura, alimentación de máquinas herramientas, entre otras operaciones. Sin embargo, existen otro tipo de aplicaciones que han hecho evolucionar en gran medida tanto la concepción de los robots como su propia morfología. Entre estos robots dedicados a aplicaciones no industriales destacan los robots espaciales (brazos para lanzamiento y recuperación de satélites, vehículos de exploración lunar, robots para construcción y mantenimiento de hardware en el espacio); robots para aplicaciones submarinas y subterráneas (exploración submarina, instalación y mantenimiento de cables telefónicos submarinos, limpieza e inspección de tuberías y drenajes subterráneos, inspección de sistemas de refrigeración en centrales nucleares), robots militares (desactivación de bombas), robots móviles industriales (robots bibliotecarios, caminantes), aplicaciones médicas (prótesis robotizadas, sistemas de ayuda a discapacitados), entre otras aplicaciones.^{1,2,16}



1.2 Clasificación de los robots

La tabla 1.1 describe una clasificación de robots de acuerdo con la Asociación Japonesa de Robótica Industrial (JIRA):^{2,16}

Clase	Nombre	Características
1	Dispositivo de manejo manual.	Es un dispositivo con múltiples grados de libertad que es manejado por un operador.
2	Robot de secuencia fija.	Es un dispositivo que desarrolla estados sucesivos de acuerdo con una tarea predeterminada. La cual es difícil de cambiar.
3	Robot de secuencia variable.	Igual que la clase 2, pero fácil de modificar dicha tarea.
4	Robot reproductor.	Un operador humano desarrolla la tarea manualmente para manejar el robot, el cual grava las posiciones para posteriormente reproducirlos. El robot repite los mismos movimientos de acuerdo con la información gravada.
5	Robots de control numérico.	El operador suministra al robot con un programa de movimiento, más que enseñarle a éste la tarea de manera manual.
6	Robots inteligentes.	Éstos cuentan con los medios para entender su desarrollo y la habilidad para que sucesivamente completen una tarea a pesar de cambios en las condiciones del medio, bajo la cual éste se esta desarrollando.

Tabla 1.1 Clasificación de robots según la JIRA.

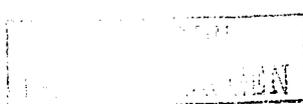
La Federación Internacional de Robótica (AFRI) en su informe técnico realiza la siguiente clasificación de robots:^{2,16}

Tipos	Características
Tipo A	Manipulador con control manual o telemando.
Tipo B	Manipulador automático con ciclos preajustados; regulación mediante fines de carrera o topes; control por PLC; accionamiento neumático o hidráulico.
Tipo C	Robot programable con trayectoria continua o punto a punto. Carece de conocimiento sobre su entorno.
Tipo D	Robot capaz de adquirir datos sobre su entorno, readaptando su tarea en función de éstos.

Tabla 1.2 Clasificación de robots según la AFRI.

Generación	Características
1ª generación	Repite la tarea programada secuencialmente. No toma en cuenta las posibles alteraciones de su entorno.
2ª generación	Adquiere información limitada de su entorno y actúa en consecuencia. Puede localizar, clasificar (visión) y detectar esfuerzos y adaptar sus movimientos en consecuencia.
3ª generación	Su programación se realiza mediante el empleo de un lenguaje natural. Posee capacidad para la planificación automática de tareas.

Tabla 1.3 Clasificación de los robots industriales en generaciones.



1.3 Elementos componentes del robot

Un robot, como un sistema, consiste de los siguientes elementos:^{1,16}

Manipulador. Es el cuerpo principal del robot y consiste de eslabones, juntas y elementos estructurales del robot. Sin los otros elementos, el manipulador solo, no es un robot.

Efector final. Es la parte que es conectada a la última junta (la mano) de un manipulador, el cual generalmente maneja objetos, hace la conexión con otras máquinas, o desarrolla las tareas requeridas.

Actuadores. De manera análoga, los actuadores son los músculos de los manipuladores. Los tipos comunes de actuadores son: los servomotores, motores paso a paso, cilindros neumáticos y cilindros hidráulicos.

Controlador. El controlador es semejante al cerebelo, aunque éste no tiene potencia, controla el cerebro y éste a la vez controla los movimientos. El controlador recibe los datos de la computadora, controla los movimientos de los actuadores y coordina los movimientos con la información enviada de los sensores.

Procesador. El procesador es el cerebro del robot. Éste calcula los movimientos de las juntas del robot, determina cuanto y que tan rápido debe moverse la junta a la localización deseada. El procesador es generalmente una computadora, la cual es dedicada a realizar un proceso. Ésta requiere de un sistema operativo, programas, equipo periférico tal como monitores, teclado, entre otros.

Software. Hay tres tipos de software que son usados en un robot. Uno es el sistema operativo, el cual opera a la computadora. El segundo es el software del robot, el cual calcula los movimientos necesarios de cada junta basado sobre las ecuaciones cinemáticas del robot. Esta información es enviada al controlador. Este software puede ser hecho en diferentes niveles, desde lenguaje de máquina hasta los lenguajes usados por robots



modernos. El tercer grupo es la colección de rutinas y programas de aplicación que son desarrollados para usar dispositivos periféricos de los robots.

En la figura 1.1 se muestra un diagrama de bloques de la forma en que interactúa cada uno de los componentes de un robot como sistema.

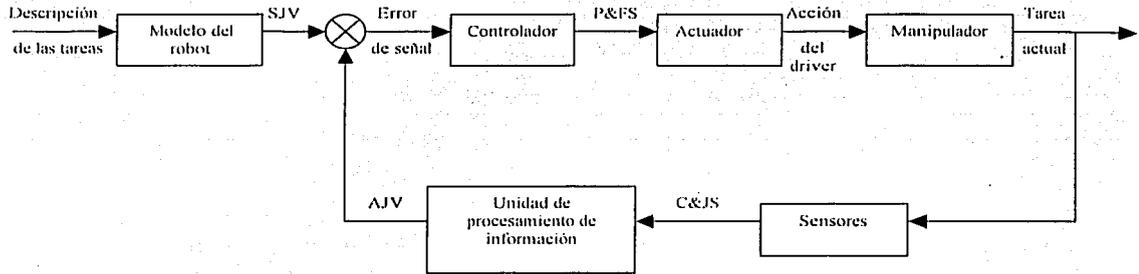


Figura 1.1 Diagrama de bloques de un sistema robótico.¹

Las abreviaturas son explicadas a continuación:

SJV: Variables de juntas sintetizadas (ángulos y torques).

P&FS: Señales de posición y fuerza.

C&JS: Señales cartesianas y angulares.

AJV: Variables angulares actuales (ángulos y torques).

Estos subsistemas se comunican entre si, vía interfaces. Dicha función consiste básicamente en decodificar la información transmitida de un medio a otro. La figura 1.1 muestra un diagrama de bloques de representación típica de un robot como un sistema. Su entrada es una tarea preestablecida, la cual es definida sobre línea o fuera de línea. Las tareas pueden ser descritas por máquinas inteligentes por un software basado sobre alguna técnica artificial. La salida de un sistema es la tarea realizada, la cual es monitoreada por sensores. Los sensores transmiten información en forma de señales de retroalimentación, la cual es comparada con la tarea prescrita y la actual.¹

Capítulo 2

Modelación cinemática del robot de 2GDL

Introducción

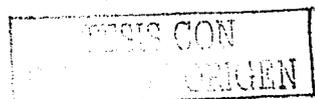
Se presenta en este capítulo el modelado cinemático de un robot de dos grados de libertad articulado por juntas rotacionales, utilizando para este propósito, la rotación usual en \mathbf{R}^2 parametrizada y sistematizada en el espacio vectorial de números complejos. Se generan las ecuaciones que gobiernan los problemas cinemáticos (de posicionamiento) directo e inverso en las configuraciones no deformada y deformada del robot.⁹

2.1 Caracterización del multicuerpo R_0

El multicuerpo por analizar en esta parte, que es llamado R_0 , es una cadena cinemática abierta formada por dos cuerpos rígidos acoplados por medio de juntas rotacionales y su campo de acción es el plano XY. El problema consiste en construir las ecuaciones que gobiernan la cinemática de la cadena mostrada en la figura 2.1. El extremo final del segundo eslabón, está obligado, en este caso, a seguir una trayectoria específica descrita en el plano XY con el fin de colocar una herramienta de trabajo que simule una operación de un proceso determinado.

Considere el multicuerpo (R_0) mostrado en la figura 2.1. Se observa que R_0 puede ser representado por medio de la unión siguiente:

$$R_0 = C_1 \cup C_2 \quad (2.1)$$



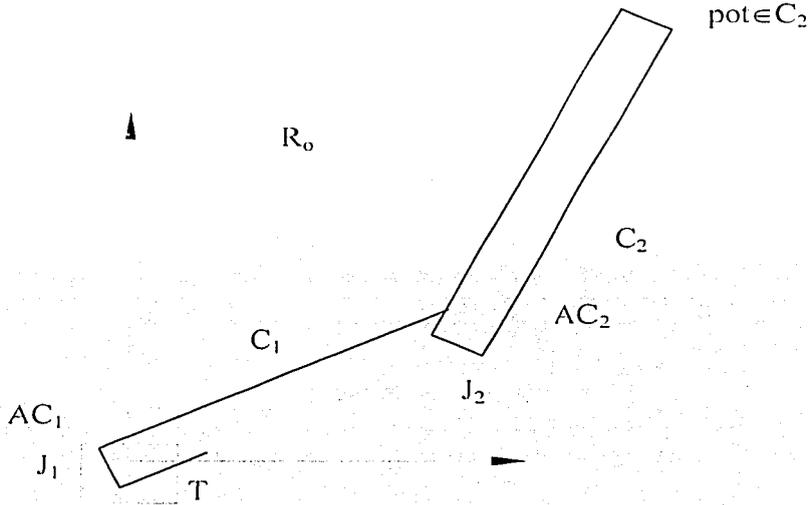


Figura 2.1 Caracterización del robot \$R_0\$.

El multicuerpo está montado en el cuerpo \$T\$ mostrado en la figura 2.1, el cual se llamará la tierra del sistema. Se llama la junta del sistema denotada por \$J_1\$, al conjunto de puntos dados por la intersección siguiente:

$$J_1 = T \cap C_1 \quad (2.2)$$

En la tierra (\$T\$) se monta un actuador, el cual es denominado \$AC_1\$, el cual se acoplará al cuerpo \$C_1\$, a través de la junta \$J_1\$.

La junta que conecta al cuerpo \$C_1\$ con el cuerpo \$C_2\$ se define como:

$$J_2 = C_1 \cap C_2 \quad (2.3)$$

En esta parte del cuerpo se localiza un segundo actuador, el cual se llama \$AC_2\$ y está acoplado con los cuerpos \$C_1\$ y \$C_2\$, a través de la junta \$J_2\$.

En este trabajo el multicuerpo \$R_0\$ se considerará formado de la manera siguiente:

$$R_0 = C_2 \cup J_2 \cup AC_2 \cup C_1 \cup AC_1 \cup J_1 \cap T \quad (2.4)$$



2.2 Modelación cinemática de un robot de 2GDL utilizado la rotación "ROT 1"

En esta sección es aplicada la transformación lineal $\rho(p, \bullet): \mathfrak{R}^2 \rightarrow \mathfrak{R}^2$ para modelar las rotaciones del robot en estudio.²³

Considere ahora que sobre la junta J_1 se define un sistema cartesiano inercial, también sobre el eje de C_1 se define un sistema local y sobre la junta J_2 se define el sistema de referencia del cuerpo C_2 como se muestra en la figura 2.2. Los sistemas de referencia asociados con los cuerpos C_1 y C_2 son llamados en este trabajo, las bases móviles y están fijas en los cuerpos C_1 y C_2 . Las longitudes de los cuerpos C_1 y C_2 son denotadas por L_1 y L_2 según se muestra en la figura 2.2. Por tanto, los vectores de posición que unen el centro de la junta J_1 con la junta J_2 y de la junta J_2 al punto $pot \in C_2$ mostrado en la figura 2.2, al cual se le llama punto final de R_0 , están definidos con respecto a las bases locales ${}^{R11}e'_1, {}^{R11}e''_1 \in \mathfrak{R}^2$, esto es:

$$L_1 = l_1 \cdot {}^{R11}e'_1 \quad (2.5)$$

$$L_2 = l_2 \cdot {}^{R11}e''_1.$$

Por otro lado, los vectores $L_1, L_2 \in \mathfrak{R}^2$, al fijar los puntos $p_1 \in J_1, p_2 \in J_2$ y $pot \in C_2$, pueden ser también expresados mediante la siguiente diferencia de puntos:

$$L_1 = p_2 - p_1, \quad L_2 = pot - p_2 \quad (2.6)$$

También el vector de posición de $pot \in C_2$ de origen en $p_1 \in J_1$ puede ser expresado de la manera siguiente:

$$R(pot) = pot - p_1 \quad (2.7)$$

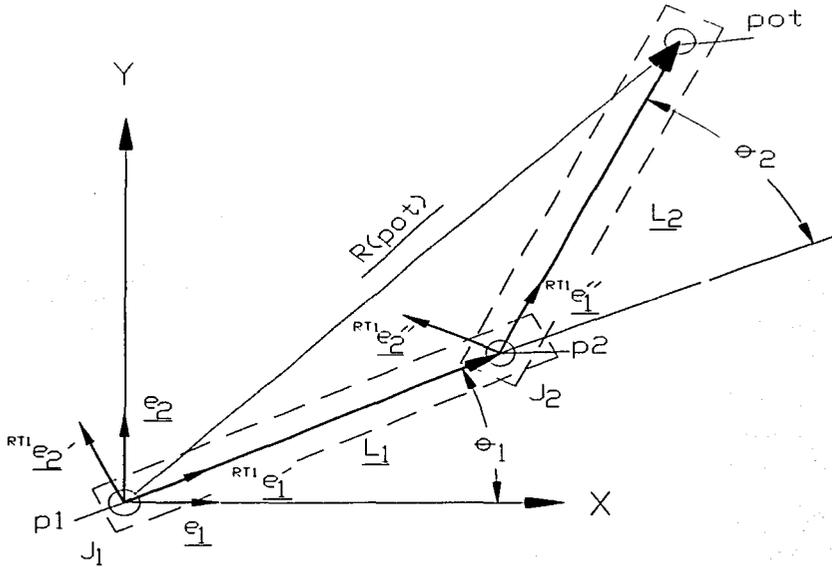


Figura 2.2 Configuración no deformada de R_0 .

El vector $\underline{R}(\text{pot}) \in \mathcal{R}^2$ es representado a través de la siguiente expresión:

$$\underline{R}(\text{pot}) = \underline{L}_1 + \underline{L}_2 \quad (2.8)$$

2.2.1 Análisis en la configuración no deformada

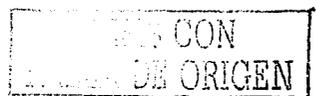
Considere ahora que el multicuerpo R_0 ocupa la posición mostrada en la figura 2.2., la cual es conocida como la configuración "no deformada". Entonces, el vector de posición de $\text{pot} \in C_2$, de acuerdo con las expresiones (2.5), está dado por:

$$\underline{R}(\text{pot}) = l_1 \cdot {}^{RT1} \underline{e}'_1 + l_2 \cdot {}^{RT1} \underline{e}''_1 \quad (2.9)$$

En este trabajo se considera que las bases móviles son rotaciones rígidas de la base inercial fija. Las rotaciones pueden expresarse en términos de complejos unitarios de la manera siguiente:

$${}^{RT1} \underline{e}'_1 = \rho(p, \underline{e}_1) \equiv p * \underline{e}_1 \quad (2.10)$$

$${}^{RT1} \underline{e}''_1 = \rho(q, {}^{RT1} \underline{e}'_1) = \rho(q, \rho(p, \underline{e}_1)) \equiv q * p * \underline{e}_1$$



Aquí, $\rho(p, \bullet): \mathbb{R}^2 \rightarrow \mathbb{R}^2$, $p \in \mathbb{R}^2$, es la transformación lineal asociada con una rotación. De acuerdo con las expresiones (2.10), la ecuación (2.9) puede ser escrita en términos de la base inercial fija de la manera siguiente:

$$R(pot) = l_1 \bullet \{ p * e_1 \} + l_2 \bullet \{ q * p * e_1 \} \quad (2.11)$$

También considerando las normas unitarias de los complejos, esto es:

$$\| p \| = \| q \| = 1 \quad (2.12)$$

se obtienen las ecuaciones que gobiernan la configuración no deformada de R_0 .

Las relaciones geométricas entre los complejos $p, q \in \mathbb{R}^2$ de norma unitaria y las rotaciones son las siguientes:

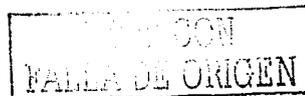
$$\begin{aligned} p \in \mathbb{R}^2 : \quad p &= (p_0, p_1) \quad ; \quad p_0 \in \mathbb{R} : \quad p_0 = \text{Cos } \theta_1 & (2.13) \\ & & p_1 \in \mathbb{R} : \quad p_1 = \pm \text{Sen } \theta_1 \\ q \in \mathbb{R}^2 : \quad q &= (q_0, q_1) \quad ; \quad q_0 \in \mathbb{R} : \quad q_0 = \text{Cos } \theta_2 \\ & & q_1 \in \mathbb{R} : \quad q_1 = \pm \text{Sen } \theta_2 \end{aligned}$$

2.2.1.1 Planteamiento de los problemas cinemáticos directo e inverso

En esta sección se definen dos problemas asociados con el multicuerpo en estudio. El primero de ellos será el problema de la cinemática directa y el segundo corresponde a la cinemática inversa. Tales problemas están asociados con la posición mostrada en la figura 2.2.

a) Problema cinemático directo

“Dados $p=(p_0, p_1)$, $q=(q_0, q_1)$ con $\| p \| = \| q \| = 1$ y $l_1 \in \mathbb{R}^+$, $l_2 \in \mathbb{R}^+$, encuentre: $R(pot) \in \mathbb{R}^2$ tal que la expresión (2.11) sea satisfecha.”



b) Problema cinemático inverso

Se presenta en esta parte el problema inverso de la cinemática de cuerpos rígidos asociado al multicuerpo R_0 motivo de estudio, esto es:

“Dado $R(p_0t) \in \mathcal{R}^3$ y $l_1 \in \mathcal{R}^3$, $l_2 \in \mathcal{R}^3$, encuentre $p=(p_0, p_1)$, $q=(q_0, q_1)$ tal que la expresión (2.11) sea satisfecha y,

$$p_0^2 + p_1^2 = 1 \quad (2.14)$$

$$q_0^2 + q_1^2 = 1$$

El problema de la cinemática inversa genera 4 ecuaciones algebraicas no lineales con 4 incógnitas del tipo polinomial. La figura 2.3 muestra una graficación de los eslabones del multicuerpo R_0 motivo de estudio:

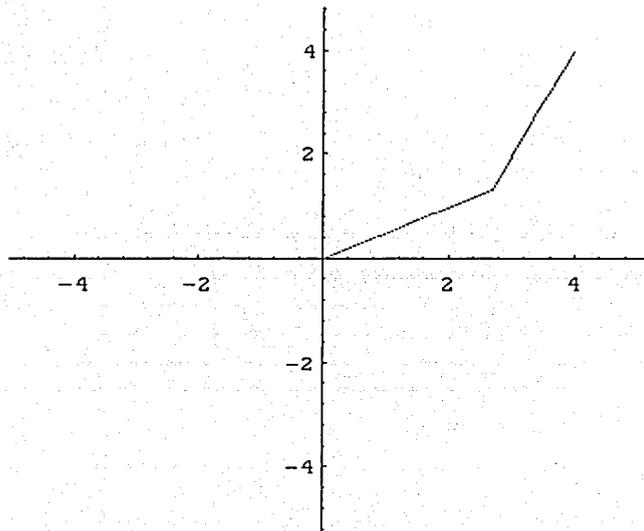


Figura 2.3 Salida gráfica del problema cinemático inverso de la posición no deformada de R_0 .

2.2.2 Análisis de la configuración deformada

En esta parte se supone que el robot (R_0) ha sido movido de su posición inicial (configuración no deformada) a través de accionamientos de los actuadores AC_1 y AC_2 (ver figura 2.4). El problema ahora consiste en determinar el lugar geométrico que ocupa el extremo final del multicuerpo. La configuración así obtenida se denominará configuración deformada.

Se observa en la figura 2.4 que el vector de posición $R(pot) \in \mathbb{R}^2$ correspondiente a la configuración deformada puede ser expresado de la forma siguiente:

$$R(pot)' = L_1' + L_2' \quad (2.15)$$

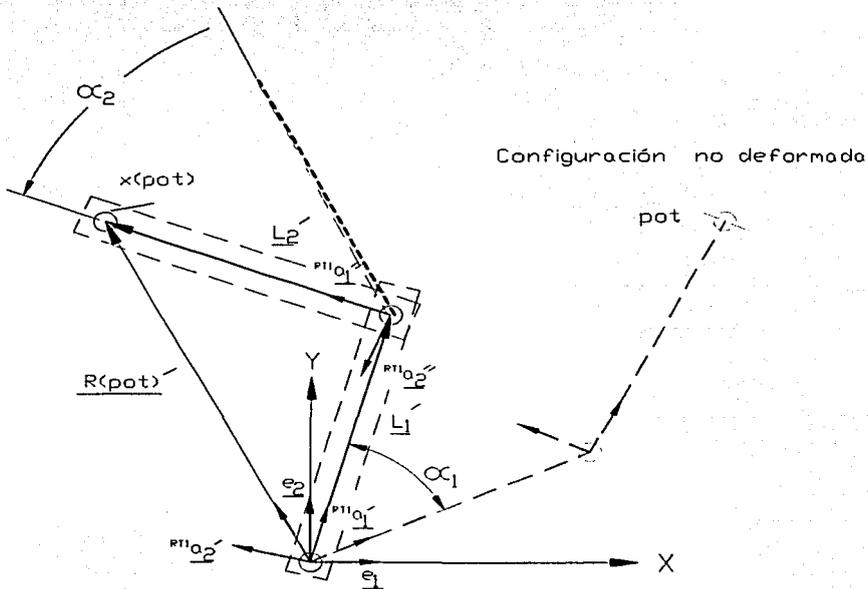


Figura 2.4 Configuración deformada de R_0 .

Los vectores de posición $L_1', L_2' \in \mathfrak{R}^2$ serán llamados los vectores deformados, los cuales también pueden ser escritos en términos de las bases locales, esto es:

$L_1' = l_1 \cdot {}^{RI1}a_1'$, $L_2' = l_2 \cdot {}^{RI1}a_1''$. Las bases ${}^{RI1}a_1'$, ${}^{RI1}a_1'' \in \mathfrak{R}^2$ son rotaciones de las bases ${}^{RI1}e_1'$, ${}^{RI1}e_1'' \in \mathfrak{R}^2$. Dichas rotaciones se pueden representar por complejos de norma unitaria de la manera siguiente:

$${}^{RI1}a_1' = \rho(P, {}^{RI1}e_1') \equiv P * p * e_1 \quad (2.16)$$

$${}^{RI1}a_1'' = \rho(Q, \rho(P, {}^{RI1}e_1'')) \equiv Q * P * q * p * e_1$$

De acuerdo con la expresión (2.16), el vector de posición $R(\text{pot}) \in \mathfrak{R}^2$ puede ser representado en términos de la base canónica, esto es:

$$R(\text{pot}) = l_1 \cdot \{ P * p * e_1 \} + l_2 \cdot \{ Q * P * q * p * e_1 \} \quad (2.17)$$

También considerando las normas unitarias de los complejos $P, Q \in \mathfrak{R}^2$, esto es:

$$\| P \| = \| Q \| = 1 \quad (2.18)$$

se obtienen las ecuaciones que gobiernan la configuración deformada del multicuerpo en estudio.

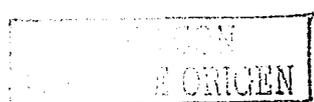
Las relaciones geométricas entre los complejos $P, Q \in \mathfrak{R}^2$ de norma unitaria y los componentes de las rotaciones son las siguientes:

$$P \in \mathfrak{R}^2 : \quad P = (P_0, P_1) \quad ; \quad P_0 \in \mathfrak{R} \quad ; \quad P_0 = \text{Cos } \alpha_1 \quad (2.19)$$

$$P_1 \in \mathfrak{R} \quad ; \quad P_1 = \pm \text{Sen } \alpha_1$$

$$Q \in \mathfrak{R}^2 : \quad Q = (Q_0, Q_1) \quad ; \quad Q_0 \in \mathfrak{R} \quad ; \quad Q_0 = \text{Cos } \alpha_2$$

$$Q_1 \in \mathfrak{R} \quad ; \quad Q_1 = \pm \text{Sen } \alpha_2.$$



2.2.2.1 Planteamiento de los problemas cinemáticos directo e inverso en la configuración deformada

a) Problema cinemático directo

“Dados $P=(P_0, P_1)$, $Q=(Q_0, Q_1)$ con $\|P\| = \|Q\| = 1$, $p=(p_0, p_1)$, $q=(q_0, q_1)$ con $\|p\| = \|q\| = 1$, y $l_1 \in \mathfrak{R}^+$, $l_2 \in \mathfrak{R}^+$, encuentre: $R \text{ (pot)'} \in \mathfrak{R}^2$ tal que la ecuación (2.17) sea satisfecha”.

b) Problema cinemático inverso

“Dado $R \text{ (pot)'} \in \mathfrak{R}^2$ y $l_1 \in \mathfrak{R}^+$, $l_2 \in \mathfrak{R}^+$, $p=(p_0, p_1)$, $q=(q_0, q_1)$ con $\|p\| = \|q\| = 1$, encuentre: $P=(P_0, P_1)$, $Q=(Q_0, Q_1)$ tal que la expresión (2.17) sea satisfecha y

$$\begin{aligned} P_0^2 + P_1^2 &= 1 \\ Q_0^2 + Q_1^2 &= 1 \end{aligned} \tag{2.20}$$

El problema de la cinemática inversa en la posición deformada genera 4 ecuaciones algebraicas no lineales con 4 incógnitas del tipo polinomial. La figura siguiente muestra una gráfica del robot en la configuración deformada:

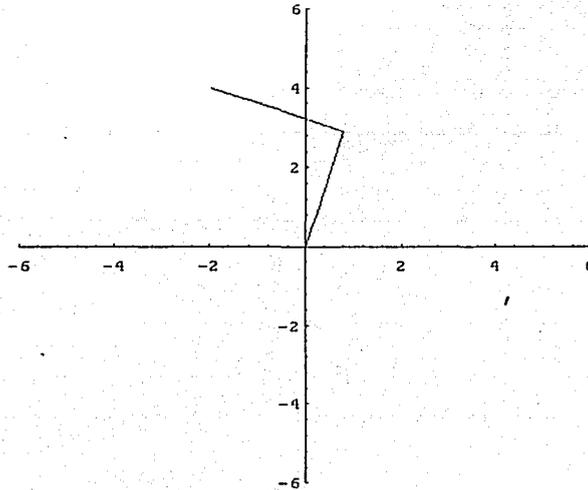
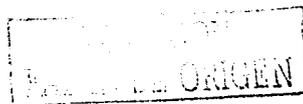


Figura 2.5 Salida gráfica del problema cinemático inverso de la posición deformada de R_0 .



2.3 Modelación cinemática utilizando la reflexión "REF 1"

Se presenta en esta sección el modelado cinemático del multicuerpo R_0 motivo de estudio utilizando la transformación $R(p, \bullet): \mathcal{R}^2 \rightarrow \mathcal{R}^2$, dicha transformación resulta ser una reflexión.²³ Se construyen también los problemas cinemáticos directo e inverso asociados con las configuraciones resultantes al aplicar dicha transformación.

2.3.1 Configuración reflectada no deformada

La configuración real del multicuerpo R_0 es la mostrada en la figura 2.2. Sin embargo, al utilizar la transformación $R(p, \bullet): \mathcal{R}^2 \rightarrow \mathcal{R}^2$ sobre cada uno de los vectores de posición definidos sobre los eslabones de la cadena articulada motivo de nuestro estudio, se obtiene una configuración especial que en este trabajo será llamada "configuración reflectada". En realidad la reflexión tendrá lugar sobre el vector de posición $R(pot) \in \mathcal{R}^2$ puesto que las coordenadas del punto $pot \in C_2$ son las mismas que en el caso del problema asociado con la rotación 1 (ROT 1), según se muestra en la figura 2.6.

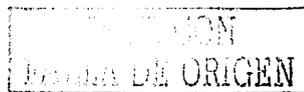
Considere ahora que el multicuerpo R_0 ocupa la posición mostrada en la figura 2.6. Entonces, las coordenadas del punto $pot \in C_2$ son determinadas a través de la siguiente expresión vectorial:

$$R(pot)^{REF1} = L_1 + L_2 \quad (2.21)$$

o, equivalentemente,

$$R(pot)^{REF1} = l_1 \bullet^{REF1} e_1' + l_2 \bullet^{REF1} e_1'' .$$

pues, $L_1 = l_1 \bullet^{REF1} e_1'$ y $L_2 = l_2 \bullet^{REF1} e_1''$.



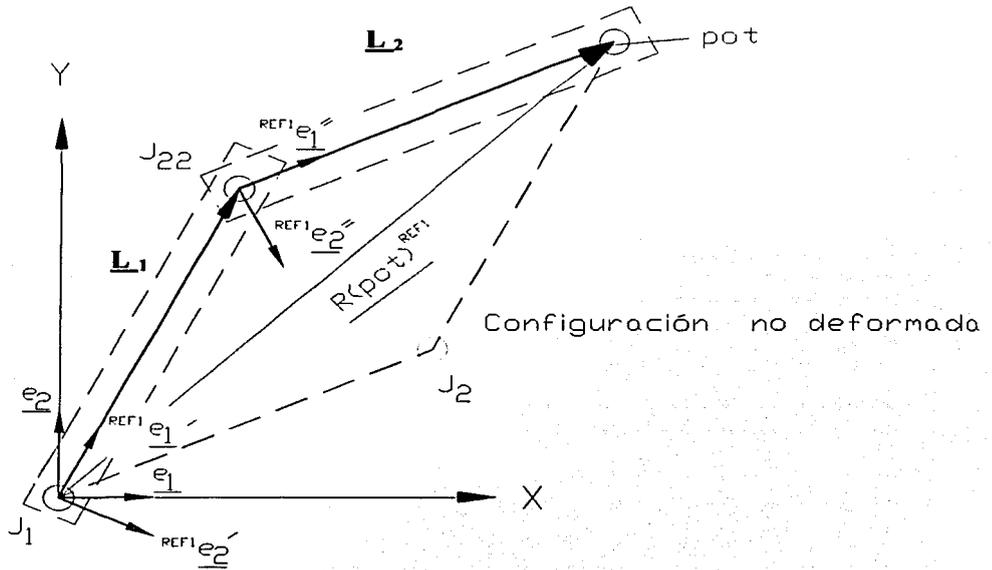


Figura 2.6 Esquematización de la configuración reflectada no deformada.

El vector $\underline{R}(\text{pot})^{\text{REF1}} \in \mathcal{R}^2$ será llamado en este trabajo, el vector reflectado bajo la transformación $\underline{R}(p, \bullet): \mathcal{R}^2 \rightarrow \mathcal{R}^2$. Las bases reflectadas $\underline{e}_1^{\text{REF1}}, \underline{e}_2^{\text{REF1}} \in \mathcal{R}^2$ serán consideradas como reflexiones rígidas de la base inercial fija. Las reflexiones pueden expresarse, en el caso de complejos unitarios, de la manera siguiente:

$$\underline{e}_1^{\text{REF1}} = \underline{R}(p, \underline{e}_1) \equiv \bar{p} * \underline{e}_1 \quad (2.22)$$

$$\underline{e}_2^{\text{REF1}} = \underline{R}(q, \underline{e}_1^{\text{REF1}}) = \underline{R}(q, \underline{R}(p, \underline{e}_1)) \equiv \overline{q * p} * \underline{e}_1$$

De acuerdo con las expresiones (2.22) la ecuación de posición (2.21) puede ser escrita en términos de la base canónica, esto es:

$$\underline{R}(\text{pot})^{\text{REF1}} = l_1 \bullet \{ \bar{p} * \underline{e}_1 \} + l_2 \bullet \{ \overline{q * p} * \underline{e}_1 \} \quad (2.23)$$

FALLA DE ORIGEN

También, considerando las normas unitarias de los complejos, esto es:

$$\| p \| = \| q \| = 1. \quad (2.24)$$

se obtienen las ecuaciones de posición que gobiernan la configuración no deformada reflectada de R_0 .

Las relaciones geométricas entre los complejos $p, q \in \mathbb{R}^2$ de norma unitaria y los componentes de las reflexiones son las siguientes:

$$\begin{aligned} p \in \mathbb{R}^2; \quad p &= (p_0, p_1) \quad ; \quad p_0 \in \mathbb{R} \quad ; \quad p_0 = \text{Sen } \theta_1 & (2.25) \\ & & & & p_1 \in \mathbb{R} \quad ; \quad p_1 = \text{Cos } \theta_1 \\ q \in \mathbb{R}^2; \quad q &= (q_0, q_1) \quad ; \quad q_0 \in \mathbb{R} \quad ; \quad q_0 = \text{Sen } \theta_2 \\ & & & & q_1 \in \mathbb{R} \quad ; \quad q_1 = \text{Cos } \theta_2. \end{aligned}$$

donde,

$$\text{Cos } \theta_1 = p_0(c_1^2 - e_2^2) - 2 p_1 c_1 e_2$$

$$\text{Cos } \theta_2 = q_0(c_1^2 - e_2^2) - 2 q_1 c_1 e_2$$

2.3.1.1 Planteamiento de los problemas cinemático directo e inverso

a) Problema cinemático directo

“Dados $p=(p_0, p_1), q=(q_0, q_1)$ con $\| p \| = \| q \| = 1$ y $l_1 \in \mathbb{R}^+, l_2 \in \mathbb{R}^+$, encuentre:

$R(\text{pot})^{\text{REFI}} \in \mathbb{R}^2$ tal que la expresión (2.23) sea satisfecha.”

b) Problema cinemático inverso

“Dado $R(\text{pot})^{\text{REFI}} \in \mathbb{R}^2$ y $l_1 \in \mathbb{R}^+, l_2 \in \mathbb{R}^+$, encuentre $p=(p_0, p_1), q=(q_0, q_1)$ tal que la expresión (2.23) sea satisfecha y

$$p_0^2 + p_1^2 = 1 \quad (2.26)$$

$$q_0^2 + q_1^2 = 1”$$



La figura siguiente muestra la salida gráfica del multicuerpo en estudio:

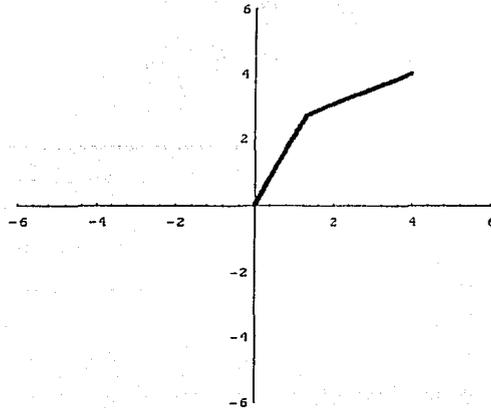


Figura 2.7 Salida gráfica del problema cinemático inverso de la posición no deformada reflectada de R_0 .

2.3.2 Análisis de la configuración reflectada deformada

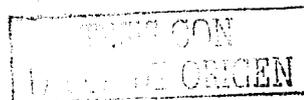
Considere ahora que el multicuerpo R_0 ha sido movido de su configuración reflectada no deformada a través de reflexiones sucesivas a la configuración deformada, como se muestra en la figura 2.8. El lugar geométrico de R (pot)^{REFI/} es:

$$R(\text{pot})^{\text{REFI}/} = L_1' + L_2' \quad (2.27)$$

o equivalentemente,

$$R(\text{pot})^{\text{REFI}/} = l_1 \bullet^{\text{REF1}} a_1' + l_2 \bullet^{\text{REF1}} a_1''$$

pues, $L_1' = l_1 \bullet^{\text{REF1}} a_1'$ y $L_2' = l_2 \bullet^{\text{REF1}} a_1''$.



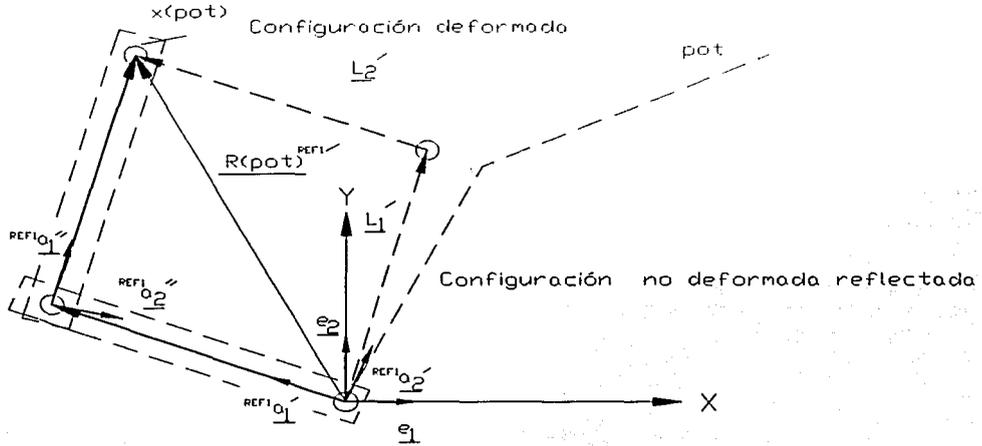


Figura 2.8 Esquematización de la configuración reflectada deformada.

Las bases ${}^{\text{REF1}}\underline{a}'_1, {}^{\text{REF1}}\underline{a}''_1 \in \mathfrak{R}^2$ serán en este caso, las bases móviles deformadas reflectadas. En términos de números complejos las reflexiones de las bases locales definidas en la configuración no deformada reflectada se expresan en la forma siguiente:

$${}^{\text{REF1}}\underline{a}'_1 = \mathbf{R}(P, {}^{\text{RT1}}\underline{e}'_1) \equiv \overline{P * p} * \underline{e}_1 \quad (2.28)$$

$${}^{\text{REF1}}\underline{a}''_1 = \mathbf{R}(Q, \mathbf{R}(P, {}^{\text{RT1}}\underline{e}''_1)) \equiv \overline{Q * P * q * p} * \underline{e}_1$$

De acuerdo con la expresión (2.28), el vector de posición $\underline{\mathbf{R}}(\text{pot})^{\text{REF1/}} \in \mathfrak{R}^2$ puede ser escrito en términos de la base canónica, esto es:

$$\underline{\mathbf{R}}(\text{pot})^{\text{REF1/}} = l_1 * \{ \overline{P * p} * \underline{e}_1 \} + l_2 * \{ \overline{Q * P * q * p} * \underline{e}_1 \} \quad (2.29)$$

Las relaciones geométricas entre los complejos $P, Q \in \mathfrak{R}^2$ de norma unitaria y los componentes de las reflexiones son las siguientes:

$$\begin{aligned} P \in \mathfrak{R}^2 ; \quad P = (P_0, P_1) ; \quad P_0 \in \mathfrak{R} ; \quad P_0 = \text{Sen } \alpha_1 \\ P_1 \in \mathfrak{R} ; \quad P_1 = \text{Cos } \alpha_1 \\ Q \in \mathfrak{R}^2 ; \quad Q = (Q_0, Q_1) ; \quad Q_0 \in \mathfrak{R} ; \quad Q_0 = \text{Sen } \alpha_2 \\ Q_1 \in \mathfrak{R} ; \quad Q_1 = \text{Cos } \alpha_2. \end{aligned} \quad (2.30)$$

donde,

$$\cos \alpha_1 = P_0(c_1^2 - c_2^2) - 2 P_1 c_1 c_2$$

$$\cos \alpha_2 = Q_0(c_1^2 - c_2^2) - 2 Q_1 c_1 c_2$$

2.3.2.1 Planteamiento de los problemas cinemático directo e inverso

a) Problema cinemático directo en la configuración deformada

“Dados $P=(P_0, P_1)$, $Q=(Q_0, Q_1)$ con $\|P\| = \|Q\| = 1$, $p=(p_0, p_1)$, $q=(q_0, q_1)$ con $\|p\| = \|q\| = 1$, $l_1 \in \mathfrak{R}^+$, $l_2 \in \mathfrak{R}^+$, encuentre: $R(\text{pot})^{\text{REFI}} \in \mathfrak{R}^2$: tal que la expresión (2.29) sea satisfecha.”

b) Problema cinemático inverso en la configuración deformada

“Dado $R(\text{pot})^{\text{REFI}} \in \mathfrak{R}^2$ y $l_1 \in \mathfrak{R}^+$, $l_2 \in \mathfrak{R}^+$, $p=(p_0, p_1)$, $q=(q_0, q_1)$ con $\|p\| = \|q\| = 1$, encuentre: $P=(P_0, P_1)$, $Q=(Q_0, Q_1)$, tal que la ecuación (2.29) sea satisfecha y,

$$P_0^2 + P_1^2 = 1$$

$$Q_0^2 + Q_1^2 = 1$$

La figura siguiente muestra la configuración del multicuerpo determinada por la programación de las ecuaciones de posición (ver referencia [9]):

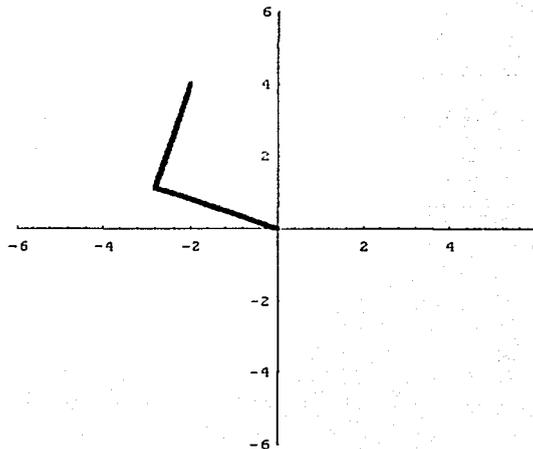


Figura 2.9 Salida gráfica de la configuración reflectada deformada de R_0 .



Capítulo 3

Caracterización de los movimientos del robot de 2GDL

Introducción

En este capítulo se presenta el análisis de los movimientos completos del robot de 2GDL motivo de estudio. Se define el problema de los movimientos, así como sus restricciones fundamentales. Se propone una hipótesis y un sistema de axiomas mediante los cuales se deducirá la solución del problema. Así también, se analizan los movimientos uno a uno, y simultáneos.

3.1 Definición del problema y restricciones

Se requiere encontrar el número de movimientos que un robot de 2GDL tipo RR puede realizar entre dos configuraciones fijas. Asimismo, cada movimiento debe ser analizado, caracterizado y simulado posteriormente en DELPHI 5 de tal manera que dicho robot se mueva en tiempo real, es decir, como se muestra gráficamente en pantalla debe de verse en la realidad dichos movimientos de manera instantánea.

Las restricciones del problema son:

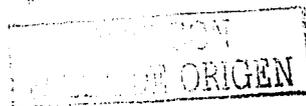
- 1) El robot lo componen dos eslabones rígidos conectados por juntas rotacionales (R_1).
- 2) El movimiento del robot es en el plano (X, Y) (R_2).
- 3) Cada eslabón que compone al robot tiene asociado un actuador (R_3).
- 4) El movimiento de cada eslabón es independiente (R_4).
- 5) Cada eslabón puede ejecutar una rotación de 360° sobre su eje (R_5).
- 6) Si $l_1, l_2 \in \mathbb{R}^+$ son las dimensiones principales de los eslabones que componen al robot (medidos de junta a junta), entonces $l_1 > l_2$ (R_6).

El problema y sus restricciones definidos anteriormente se pueden sintetizar de la manera siguiente:

• Dado: R_0, R_0^{NP}, R_0^{N} con R_0 de 2GDL, encuentre:

- 1) Los desplazamientos angulares entre R_0^{NP} y R_0^{N} .
- 2) El número y tipo de movimientos de R_0^{NP} a R_0^{N} .

tal que:



1) $R_1 = R_1$; 2) $R_2 = \hat{R}_2$; 3) $R_3 = \hat{R}_3$; 4) $R_4 = \hat{R}_4$; 5) $R_5 = \hat{R}_5$; $R_6 = R_6$.
 sean satisfechas. ”

3.2 Hipótesis y consideraciones importantes

Considere la siguiente hipótesis:

“ Existe una relación funcional entre el número de movimientos y los grados de libertad del robot. Además, no existen dos movimientos cinemáticamente equivalentes en el sentido de los lugares geométricos que recorren los eslabones en su conjunto y el órgano terminal. ”

Los axiomas relacionados con el problema por solucionar en este trabajo es el siguiente:

Axioma 3.1. Los movimientos del robot están sujetos a las leyes de Newton.

Axioma 3.2. Cada eslabón del robot se desplaza angularmente, siendo φ_1 el ángulo, de la manera siguiente: 1) φ_1 o 2) φ_1^* si y solo si $\varphi_1 + \varphi_1^* = 2\pi$.

Entre los términos importantes de este trabajo de tesis, es el de “movimiento”, que para éste caso específico de estudio, se define de la manera siguiente:

Definición 3.1. El movimiento del robot se define como la transición T de R_0^{SD} a R_0^D , en donde T está en función de los desplazamientos angulares aplicados por los actuadores de los eslabones. Y los símbolos R_0^{SD} , R_0^D son el estado inicial y final de configuración del robot R_0 respectivamente.

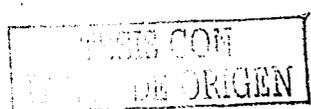
Nótese de la definición descrita anteriormente que, la transición $T: R_0^{SD} \rightarrow R_0^D$ dada por:

$T(R_0^{SD}) = R_0^D$ es quien define el movimiento de R_0 .

Por otro lado, los estados inicial y final de configuración del robot R_0 se definen por medio de la función, $R_t: R_0 \times J \rightarrow \{1, 0\}$, definida por:

$$1) R_t(R_0, t) = 1 \Leftrightarrow R_0^{SD} \text{ si } t = 0; t \in J$$

$$2) R_t(R_0, t) = 0 \Leftrightarrow R_0^D \text{ si } t = 0; t \in J$$



Aquí, $J \subset] 0, +\infty)$ es un intervalo de tiempo y $R_c : R_n \times J \rightarrow \{1, 0\}$ es llamada relación de configuración.

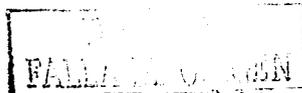
De acuerdo con el problema planteado en la sección 3.1, las configuraciones R_0^{Ni} y R_0^{Nj} deben ser fijas y, por tanto, la variable tiempo no es considerada para el análisis de movimientos.

Por otro lado, según el axioma 3.1, los movimientos del robot están sujetos a las leyes de Newton. Sin embargo, se considerará en este trabajo de tesis que la masa de los eslabones y actuadores no se toma en cuenta. Asimismo, las fuerzas que producen el movimiento tampoco son consideradas por lo que el problema por solucionar es "cinemático". Cabe señalar que la velocidad y la aceleración asociadas con la transición $T : R_0^{Ni} \rightarrow R_0^{Nj}$ no se estudiarán en este trabajo. El análisis únicamente se centra sobre los desplazamientos angulares de los eslabones.

Finalmente, para determinar el estado final de la configuración R_0^{Nj} se usará la configuración R_0^{Ni} resolviendo el problema cinemático inverso y, la transición $T : R_0^{Ni} \rightarrow R_0^{Nj}$ se caracterizará por el problema cinemático directo.

3.3 Movimientos uno a uno

Se analizan las transiciones o movimientos del robot en estudio usando el criterio "secuenciado" del cual se caracterizan los "movimientos uno a uno". Es propuesta una relación funcional entre los grados de libertad del robot y el número de movimientos uno a uno. Finalmente, se propone una nomenclatura para caracterizar dichas transiciones y un código binario de reconocimiento.



Considere las siguientes definiciones:

Definición 3.2. *Un robot tiene eslabones activos si por cada eslabón está asociado un actuador.*

Definición 3.3. *Un robot se dice "completo", si el número de eslabones activos es igual al número de grados de libertad.*

Definición 3.4. *Una transición o movimiento se dice completa si en dicha transición se activan todos los actuadores asociados con un robot completo.*

Definición 3.5. *Una transición se dice incompleta si cuando menos un actuador no fue activado durante dicha transición.*

Cabe señalar que el robot R_0 estudiado en este trabajo está formado por eslabones activos y es completo.

Por otro lado, sean $R_0^{ND} = E_1^{ND} \cup E_2^{ND}$ y $R_0^D = E_1^D \cup E_2^D$ con E_1 y E_2 los eslabones que componen a R_0 . Entonces, $T[R_0^{ND}] = R_0^D \Leftrightarrow T[E_1^{ND}] = E_1^D$ y $T[E_2^{ND}] = E_2^D$

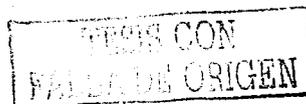
Considere ahora que M es un conjunto de actuadores y R_0 es un robot compuesto de eslabones activos. La función, $R : M \times R_0 \rightarrow \{1, 0\}$, definida por:

- 1) $R_i(M_i, E_i) = 1 \Leftrightarrow T[E_i^{ND}] = E_i^D$; $E_i \in R_0$ y $M_i \in M$
- 2) $R_i(M_i, E_i) = 0 \Leftrightarrow T[E_i^{ND}] = E_i^D \Leftrightarrow T = I$: con I la transición identidad es llamada relación de activación de actuadores.

3.3.1 Caracterización de los parámetros de rotación

Para determinar la configuración deformada R_0^{ND} de R_0 , es necesario solucionar el problema inverso. Supóngase que $P_0 \in \mathfrak{R}$ y $Q_0 \in \mathfrak{R}$ son los parámetros de rotación obtenidos al solucionar la cinemática inversa. Considérese lo siguiente:

"Existen dos parámetros de rotación genéricos asociados con un robot de 2GDL con los cuales se caracterizan todas las transiciones. Dichos parámetros son generados al solucionar el problema inverso."



Sean $P_0 \in \mathfrak{R}$ y $Q_0 \in \mathfrak{R}$ los parámetros de rotación genéricos de R_0 . Entonces, de acuerdo con la propiedad tricotómica de los números reales, las siguientes relaciones se satisfacen:

- 1) $P_0 < Q_0$ o
- 2) $P_0 > Q_0$ o
- 3) $P_0 = Q_0$

Por otro lado, según el axioma 3.2 definido en la sección 3.2 de este trabajo, las siguientes relaciones angulares se satisfacen:

- 1) $P_0 + P_0^* = 1$
- 2) $Q_0 + Q_0^* = 1$

Aquí, $P_0^* \in \mathfrak{R}$ y $Q_0^* \in \mathfrak{R}$ son los complementos angulares de $P_0 \in \mathfrak{R}$ y $Q_0 \in \mathfrak{R}$.

3.3.2 Criterios de activación

Un actuador, de acuerdo con el axioma 3.2, puede ejecutar desplazamientos angulares del tipo $P_0 \in \mathfrak{R}$ y $P_0^* \in \mathfrak{R}$. Por tanto, el siguiente conjunto de parámetros de rotación caracteriza las formas de activación para un robot de 2GDL completo:

$$F_A = \{P_0, P_0^*, Q_0, Q_0^*\}$$

Considere que $F_{A1} = \{P_0, P_0^*\}$ y $F_{A2} = \{Q_0, Q_0^*\}$. El conjunto,

$$F_K = F_{A1} \times F_{A2} = \{(P_0, Q_0), (P_0, Q_0^*), (P_0^*, Q_0), (P_0^*, Q_0^*)\}$$

es llamado el producto cartesiano de las formas de activación angular primaria. Por otro lado, el conjunto, $F_L = F_{A2} \times F_{A1} = \{(Q_0, P_0), (Q_0, P_0^*), (Q_0^*, P_0), (Q_0^*, P_0^*)\}$ es llamado producto cartesiano de las formas de activación secundaria. Considere la siguiente hipótesis:

" Existe un tipo de movimientos completos de un robot de 2GDL que satisfacen:

$$(P_0, Q_0) \in F_K \cup F_L$$

y el número de dichos movimientos está en función de la cardinalidad de $F_K \cup F_L$ y el grado de libertad $m(F_K \cup F_L) = f(n)$, siendo " m " $\in \mathcal{N}$ la cardinalidad de $F_K \cup F_L$ y $n \in \mathcal{N}$ el número de grados de libertad del robot estudiado. "



Para finalizar esta sección considere los siguientes conjuntos:

$$1) M_1 = \{ M_1^{L_1}, M_1^{R_1} \}$$

$$2) M_2 = \{ M_2^{L_2}, M_2^{R_2} \}$$

Aquí, $M_1^{L_1}$ es el actuador M_1 que ejecuta el desplazamiento angular $P_0 \in \mathfrak{R}$, $M_1^{R_1}$ es M_1 el cual ejecuta el complemento angular $P_0^* \in \mathfrak{R}$, $M_2^{L_2}$ y $M_2^{R_2}$ son las formas de activación del actuador M_2 .

3.3.3 Tipos de movimientos uno a uno

En esta sección se analizan y caracterizan los movimientos o transiciones del tipo uno a uno usando la hipótesis definida en la sección 3.3.2. Se determina la relación funcional entre la cardinalidad de $F_k \cup F_l$ y los grados de libertad.

Considere la siguiente definición:

Definición 3.6. Una transición $T: R_0^{ND} \rightarrow R_0^D$ se dice uno a uno si para alcanzar R_0^D de R_0^{ND} existen dos subtransiciones ejecutadas secuencialmente tal que:

$$T_1: R_0^{ND} \rightarrow R_0^D \quad \text{y} \quad T_2: R_0^D \rightarrow R_0^D$$

siendo R_0^D una configuración intermedia de R_0 .

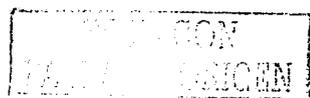
Note que el término "secuencial" se puede representar por medio de la función:

$$T: R_0^{ND} \rightarrow R_0^D$$

definida por:

$$T(R_0^{ND}) = T_2 \circ T_1(R_0^{ND}) = T_2(T_1(R_0^{ND})) = T_2(R_0^D) = R_0^D$$

Por otro lado, una subtransición es en realidad la ejecución de movimiento de un actuador, el cual desplazará angularmente un eslabón o, en su caso, todo el robot.



3.3.3.1 Movimientos uno a uno reales

En esta sección se analizarán los movimientos uno a uno reales caracterizados por las parejas $(P_0, Q_0) \in F_{AK}$ y $(P_0, Q_0) \in F_{AL}$.⁵

Considere los siguientes conceptos:

Definición 3.7. Una subtransición se dice totalmente realizada si un actuador asociado con un eslabón aplica todo el desplazamiento angular en forma continua.

Note que si una subtransición es ejecutada totalmente implica que el actuador se ha activado hasta completar todo el desplazamiento angular.

Definición 3.8. Un movimiento uno a uno se dice ejecutado si para que el actuador asociado con $T_2 : R_0^D \rightarrow R_0^D$ inicie su operación, el actuador relacionado con $T_1 : R_0^{SP} \rightarrow R_0^D$ ha aplicado todo el desplazamiento angular en forma continua.

Los movimientos completos uno a uno serán aquellos que satisfagan la definición 3.6.

Sean $(P_0, Q_0) \in F_{AK}$ y $(Q_0, P_0) \in F_{AL}$. Los movimientos uno a uno son llamados “reales” si cada actuador ejecuta los desplazamientos angulares obtenidos directamente de la cinemática inversa, o sea $P_0 \in \mathfrak{R}$ y $Q_0 \in \mathfrak{R}$.

Note que existen dos movimientos reales; estos son:

- 1) $M_{1,2}$; 2) $M_{2,1}$

o, equivalentemente:

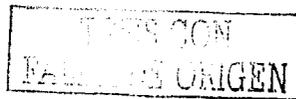
1) $MR^{F_1 A_2}$; movimiento real de la familia F_1 .

2) $MR^{F_2 A_1}$; movimiento real de la familia F_2 .

Aquí, la familia indica qué actuador se activó primero. Considere las siguientes equivalentes:

i) $T_1 : R_0^{SP} \rightarrow R_0^D$ o $T_1 : R_0^D \rightarrow R_0^{SP} \Leftrightarrow 1$

ii) $T_1 : R_0^{SP} \rightarrow R_0^{SP}$ o $T_1 : R_0^D \rightarrow R_0^D \Leftrightarrow 0$



Los movimientos reales uno a uno se pueden representar por medio de los siguientes códigos binarios:

MR^{F_1, M_2}		
T \ M	M_1	M_2
T_1	1	0
T_2	0	1

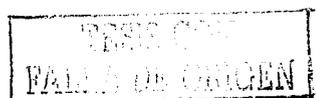
Tabla 3.1 Código binario para el movimiento MR^{F_1, M_2}

MR^{F_2, M_1}		
T \ M	M_1	M_2
T_1	0	1
T_2	1	0

Tabla 3.2 Código binario para el movimiento MR^{F_2, M_1}

Es importante señalar que:

- 1) $(T_1, M_1) = 1 \Leftrightarrow M_1^{F_1}$
- 2) $(T_1, M_2) = 1 \Leftrightarrow M_2^{G_1}$
- 3) $(T_1, M_1) = 0 \Leftrightarrow T_1 = 1$
- 4) $(T_1, M_2) = 0 \Leftrightarrow T = 1$
- 5) $(T_2, M_1) = 1 \Leftrightarrow M_1^{F_2}$
- 6) $(T_2, M_2) = 1 \Leftrightarrow M_2^{G_2}$
- 7) $(T_2, M_1) = 0 \Leftrightarrow T_2 = 1$
- 8) $(T_2, M_2) = 0 \Leftrightarrow T_2 = 1$



a) Análisis del órgano terminal y lugares geométricos

Los movimientos reales uno a uno, al ser secuenciados, implican dos características importantes relacionadas con:

- 1) El lugar geométrico que describe el órgano terminal (punto terminal) es diferente en cada movimiento.
- 2) Las regiones de barrido de los eslabones son diferentes para cada movimiento.

Para cada movimiento existe un punto intermedio sobre el cual el órgano terminal del robot se posiciona una vez efectuada una subtransición. Además, por el hecho de aplicar la cinemática directa para simular los movimientos, y debido con la secuencialidad de las subtransiciones, los lugares geométricos de los movimientos reales uno a uno son semicircunferencias. Por otro lado, el conjunto de regiones de barrido de los eslabones sobre el área de trabajo son diferentes en cada movimiento real.

b) Relaciones importantes de los movimientos uno a uno reales

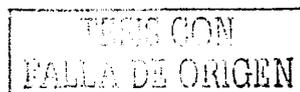
Los movimientos completos del tipo uno a uno reales se relacionan con el grado de libertad del robot R_0 motivo de estudio en este trabajo de la manera siguiente:

- 1) $N(MR) = n$
- 2) $N(F) = n$
- 3) $N(PI) = n - 1$; para cada movimiento

Aquí, $N(MR)$ es el número de movimientos reales, $N(F)$ el número de familias y $N(PI)$ es el número de puntos intermedios o configuraciones intermedias de R_0 .

Las siguientes secuencias de movimiento caracterizan las transiciones completas reales de R_0 :

- 1) $MR^{F_1 M_2}$: movimiento real de la familia 1 con activación del actuador M_2 en T_2 .
- 2) $MR^{F_2 M_1}$: movimiento real de la familia 2 con activación del actuador M_1 en T_2 .



3.3.3.2 Movimientos uno a uno complementarios

Se caracterizan los movimientos uno a uno complementarios relacionados con las parejas $(P_o^*, Q_o^*) \in F_{AK}$ y $(Q_o^*, P_o^*) \in F_{AL}$.

Considere ahora la siguiente definición:

Definición 3.9. *Un movimiento completo uno a uno se llama complementario si los actuadores aplican los complementos angulares de los parámetros de rotación obtenidos de la cinemática inversa para alcanzar R_o^D de R_o^{ND} .*

Los movimientos de este tipo son:

- 1) MCP^{F_1, M_2} ; movimiento complementario de la familia 1 con activación del actuador M2 en T_2 .
- 2) MCP^{F_2, M_1} ; movimiento complementario de la familia 1 con activación del actuador M1 en T_2 .

Los códigos de reconocimiento para esta clase de movimientos uno a uno son los descritos en las tablas 3.3 y 3.4.

MCP^{F_1, M_2}		
T \ M	M ₁	M ₂
T ₁	-1	0
T ₂	0	-1

Tabla 3.3 Código binario para MCP^{F_1, M_2} .

MCP^{F_2, M_1}		
T \ M	M ₁	M ₂
T ₁	0	-1
T ₂	-1	0

Tabla 3.4 Código binario para MCP^{F_2, M_1} .

Note en las tablas 3.3 y 3.4 que se ha asignado el símbolo “-” al código “1” para representar las subtransiciones que utilizan los componentes angulares de los parámetros de rotación. Los puntos intermedios o configuraciones intermedias relacionadas con los movimientos $MCP^{F_1 M_2}$ y $MCP^{F_2 M_1}$ son los mismos que los asociados con los movimientos uno a uno reales.

Las diferencias significativas son:

- 1) Los lugares geométricos que recorre el órgano terminal son diferentes.
- 2) Las relaciones de barrido de los eslabones sobre el área de trabajo son diferentes.

Sin embargo, existe una relación geométrica entre subtransiciones, esto es:

$$1) \tau_{p_1} \cup \tau_{p_2} = \tau_{c_1}$$

$$2) \tau_{c_1} \cup \tau_{c_2} = \tau_{c_3}$$

Aquí, τ_{p_1} y τ_{p_2} son los lugares geométricos que recorren cada punto de un eslabón y τ_{c_1} es un lugar geométrico circunferencial.

Una diferencia importante entre los movimientos uno a uno reales y complementarios es el sentido de aplicación de los desplazamientos angulares. Sean G_+ y G_- dos símbolos asociados con el sentido angular de desplazamiento de los eslabones activos de R_0 .

Un actuador M_1 tiene asociado la pareja (G_+, G_-) para cualesquier movimiento uno a uno.

Los siguientes criterios de giro se relacionan con los movimientos:

$$1) \text{ Si } M_1^{G_+}, \quad \text{entonces } G_+$$

$$2) \text{ Si } M_1^{G_-}, \quad \text{entonces } G_-$$

Cabe señalar que los símbolos G_+ y G_- son solamente criterios y no tienen que ver con los signos “+” o “-” de los valores de $P_1 \in \mathfrak{R}$. Lo único que se quiere poner en claro es que el sentido de una subtransición real es contraria al sentido de operación de una subtransición complementaria.

a) Relaciones importantes de los movimientos uno a uno complementarios

Los movimientos complementarios uno a uno se relacionan con los grados de libertad de R_0 de la manera siguiente:

- 1) $N(MCP) = n$
- 2) $N(F)_{MCP} = n = N(F)_{MR}$
- 3) $N(PI)_{MCP} = n - 1 = N(PI)_{MR}$; para cada movimiento.

Aquí, $N(MCP)$ es el número de movimientos complementarios uno a uno.

3.3.3.3 Movimientos uno a uno combinados

Se analizan los movimientos uno a uno combinados por las parejas $(P_0, Q_0^*), (P_0^*, Q_0) \in F_{AK}$ y $(Q_0, P_0^*), (Q_0^*, P_0) \in F_{AL}$, los cuales serán llamados combinados.

Considere la siguiente definición:

Definición 3.10. Una transición $T: R_0^{ND} \rightarrow R_0^D$ uno a uno se dice "combinada" si una y solo una subtransición es realizada con el complemento angular de un parámetro de rotación.

Los movimientos uno a uno combinados implicados en la definición anterior son los siguientes:

- 1) $MC_{M_1}^{F_1, M_2}$: movimiento combinado de la familia F_1 con subcomplementación en M_1 y actuador M_2 en T_2 .
- 2) $MC_{M_2}^{F_1, M_1}$: movimiento combinado de la familia F_1 con subcomplementación en M_2 y actuador M_1 en T_2 .
- 3) $MC_{M_1}^{F_2, M_1}$: movimiento combinado de la familia F_2 con subcomplementación en M_1 y actuador M_1 en T_2 .
- 4) $MC_{M_2}^{F_2, M_1}$: movimiento combinado de la familia F_2 con subcomplementación en M_2 y actuador M_1 en T_2 .



Los códigos de reconocimiento para los movimientos descritos anteriormente son:

$MC_{M_1}^{F_1 M_2}$			
		M	
T		M ₁	M ₂
	T ₁	-1	0
	T ₂	0	1

Tabla 3.5 Código de $MC_{M_1}^{F_1 M_2}$

$MC_{M_2}^{F_1 M_2}$			
		M	
T		M ₁	M ₂
	T ₁	1	0
	T ₂	0	-1

Tabla 3.6 Código de $MC_{M_2}^{F_1 M_2}$

$MC_{M_1}^{F_2 M_1}$			
		M	
T		M ₁	M ₂
	T ₁	0	1
	T ₂	-1	0

Tabla 3.7 Código de $MC_{M_1}^{F_2 M_1}$

$MC_{M_2}^{F_2 M_1}$			
		M	
T		M ₁	M ₂
	T ₁	0	-1
	T ₂	1	0

Tabla 3.8 Código de $MC_{M_2}^{F_2 M_1}$

TRABAJOS CON
FALLA DE ORIGEN

a) Análisis y consideraciones de los movimientos uno a uno combinados

Los cuatro movimientos combinados tienen en común el mismo punto intermedio. Sin embargo, al igual que los movimientos reales y complementarios, los combinados difieren en:

- 1) El órgano terminal del robot recorre lugares geométricos semicircunferenciales diferentes.
- 2) Las regiones de barrido de los eslabones son diferentes.

Cabe señalar que el hecho de que los movimientos combinados, reales y complementarios dependan solamente de dos familias, implica que existen dos configuraciones intermedias para todos los movimientos completos uno a uno.

Un movimiento combinado es, por tanto, diferente a los reales y complementarios, así como a los de su misma clase en el sentido de los lugares geométricos de recorrido tanto del órgano terminal como los puntos de los eslabones.

b) Relaciones importantes de los movimientos combinados

Para el caso de los movimientos o transiciones analizados en esta sección, el grado de libertad de R_0 está relacionado con dichas transiciones de la manera siguiente:

- 1) $N(MC) = [n!]^2$
- 2) $N(F) = n$
- 3) $N(CI) = n - 1$; para cada familia.

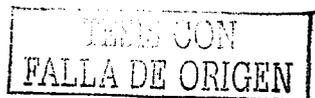
Aquí, $N(MC)$ es el número de movimientos combinados completos uno a uno.

3.3.4 Análisis y caracterización general de lo movimientos uno a uno completos

En esta sección se analizan algunas consideraciones de los movimientos uno a uno completos. De acuerdo con la hipótesis definida en la sección 3.3.3, existe una relación entre la cardinalidad $m(F_k \cup F_r)$ y el número de grados de libertad del robot R_0 motivo de estudio.

Dicha relación es la siguiente: $f(n) = n! + n! + [n!]^2$

o, equivalentemente: $f(n) = N(MR)! + N(MC')! + [N(MC)!]^2$



Nótese que si $n = 2$, entonces $f(2) = 2 + 2 + [2!]^2 = 8$

Es decir, existen 8 movimientos uno a uno completos caracterizados de la manera siguiente:

- 1) 2 reales
- 2) 2 complementarios
- 3) 4 combinados

Cabe señalar que $m(F_k \cup F_p) = 8$, pues

i) $m(F_k) = 4$

ii) $m(F_p) = 4$

iii) $m(F_k \cap F_p) = 0$

y, por tanto, $m(F_k \cup F_p) = m(F_k) + m(F_p) - m(F_k \cap F_p) = 8$

Nótese que se obtiene la relación, $m(F_k \cap F_p) = f(n) = n! + n! + [n!]^2$.

Por otro lado, las relaciones siguientes se satisfacen:

- 1) Por cada familia existen uno y solo una configuración intermedia.
- 2) Cada configuración intermedia tiene asociado un conjunto de 4 movimientos completos uno a uno.

Sean F_1 y CI_1 la familia 1 y su punto intermedio asociado. Sean también F_2 y CI_2 la familia 2 y su configuración intermedia relacionada. Los siguientes conjuntos de movimientos satisfacen a sus familias:

1) $C_{MF1} = \{ MR^{F_1 M_2}, MCP^{F_1 M_2}, MC_{M_1}^{F_1 M_2}, MC_{M_2}^{F_1 M_2} \}$

2) $C_{MF2} = \{ MR^{F_2 M_1}, MCP^{F_2 M_1}, MC_{M_1}^{F_2 M_1}, MC_{M_2}^{F_2 M_1} \}$

Por finalizar esta sección es necesario hacer notar las siguientes consideraciones:

- 1) Cada movimiento completo uno a uno es diferente en el sentido de los lugares geométricos de recorrido del órgano terminal.
- 2) La relación tricotómica $P_0 > Q_0$, $P_0 < Q_0$ y $P_0 = Q_0$ no tiene implicaciones en la característica y número de movimientos completos uno a uno.
- 3) Los lugares geométricos de cada movimiento ésta compuesto por segmentos semicirculares.

FIN CON
FALLA DE ORIGEN

3.4 Movimientos simultáneos

Se analizan las transiciones o movimientos del robot R_0 en estudio usando el criterio de "simultaneidad" de activación de los eslabones. Dicho criterio es usado para caracterizar los "movimientos simultáneos".⁵ Es propuesta, al igual que en los movimientos uno a uno, una relación funcional entre los grados de libertad y el número de transiciones simultáneas. Así como también se define una nomenclatura apropiada para representar dichas movimientos. Y finalmente, se usa el criterio de tricotomía para caracterizar las transiciones simultáneas.

Definición 3.11. *Un movimiento se dice simultáneo si durante la transición $T: R_0^{ND} \rightarrow R_0^D$ los actuadores que componen a R_0 se activan en un mismo estado inicial de operación y finalizan ambas en un mismo estado final de operación.*

La definición 3.11 implica que: "en un movimiento simultáneo existe un número infinito de puntos intermedios que depende de una relación de sincronización discreta o continua definida entre los desplazamientos angulares que ejecutan simultáneamente los actuadores para alcanzar R_0^D de R_0^{ND} ".

Es importante señalar que la proposición definida anteriormente no tiene implicaciones importantes en la caracterización de los movimientos simultáneos. El uso de dicha proposición implica que el punto intermedio no puede ser usado como criterio de diferenciación entre los movimientos simultáneos.

Considere la siguiente hipótesis para esta sección:

"Existe una relación funcional entre el número de movimientos simultáneos y el grado de libertad del robot R_n . Dicha relación depende de uno de los tres casos:

1) $P_n = Q_n$,

2) $P_n > Q_n$,

3) $P_n < Q_n$,

es decir, de una relación tricotómica definida sobre los parámetros de rotación"

3.4.1 Movimientos reales simultáneos

Se analizan los movimientos simultáneos los cuales dependen de la pareja $(P_0, Q_0) \in F_{ik}$ o $(Q_0, P_0) \in F_{iv}$. Considere ahora que para alcanzar R_0'' de $R_0^{N''}$ se han obtenido, al solucionar la cinemática inversa, los parámetros de rotación $P_0 \in \mathfrak{R}$ y $Q_0 \in \mathfrak{R}$ tales que: $P_0 = Q_0$, esto es, los desplazamientos angulares aplicados por los actuadores M_1 y M_2 a los eslabones son iguales.

De acuerdo con la definición de movimiento simultáneo, existen dos estados de activación de actuadores, uno inicial y otro final. Sean Γ_0 y Γ_1 los estados inicial y final del actuador M_1 . Sean también δ_0 y δ_1 los estados relacionados con el actuador M_2 . Entonces, para que R_0 ejecute un movimiento simultáneo, es necesario y suficiente que:

- 1) $\Gamma_0 = \delta_0$
- 2) $\Gamma_1 = \delta_1$

Se calculará ahora la relación en el envío de pulsos entre un motor $m_1(Pm_1)$ y otro motor $m_2(Pm_2)$. Para que se satisfaga la igualdad de los puntos 1 y 2.

Entonces el problema consiste en encontrar la relación de pulsos que se deben enviar al motor 1 (Pm_1) y al motor 2 (Pm_2), de tal manera que el tiempo total del motor 1 (Ttm_1) y el del motor 2 (Ttm_2) sean iguales ($Ttm_1 = Ttm_2$).

De aquí que:

Si $z_i < z_{i+1} \Rightarrow R_{z_i} = \frac{z_{i+1}}{z_i}$ o si $z_i > z_{i+1} \Rightarrow R_{z_i} = \frac{z_i}{z_{i+1}}$. Por lo tanto los pulsos que deben ser

enviados son: $Pm_i = \frac{\alpha_i}{Dm_i} * R_i$

Entonces aplicando la relación de Pm_i e introduciendo los parámetros de rotación (P_0, Q_0) a nuestro problema, se obtiene lo siguiente:

$$Pm_1 = \frac{P_0}{Dm_1} * R_{r1} \text{ y } Pm_2 = \frac{Q_0}{Dm_2} * R_{r2}$$

Considerando el intervalo de tiempo en el evento de envío de pulsos es fijo para ambos actuadores, entonces se puede obtener dicha relación:

$$\text{Si } Pm_1 < Pm_2 \Rightarrow R_r = \frac{Pm_1}{Pm_2} \text{ o } Pm_1 > Pm_2 \Rightarrow R_r = \frac{Pm_2}{Pm_1}$$

Aplicando esta relación de pulsos en cada actuador se garantiza que ambos eslabones inicien y terminen ($\Gamma_0 = \delta_0$ y $\Gamma_1 = \delta_1$) en el mismo tiempo. Y por lo tanto el tiempo total de accionamiento de los actuadores 1 y 2 serán los mismos.

Un movimiento real simultáneo tal que $P_0 = Q_0$ se representa de la manera siguiente:

$$MRS_{P_0=Q_0}^{F_0}$$

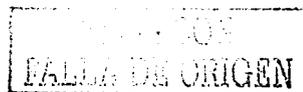
La tabla de reconocimiento de dicho movimiento es la siguiente:

$MRS_{P_0=Q_0}^{F_0}$		
T	M	M ₁
T ₁ , T ₂	1	1

Tabla 3.9 Código del un movimiento simultáneo $MRS_{P_0=Q_0}^{F_0}$.

La nomenclatura $MRS_{P_0=Q_0}^{F_0}$ se interpreta de la manera siguiente: "Movimiento real simultáneo de la familia F_0 "

Aquí, F_0 es la familia asociada con $P_0 = Q_0$. De lo que se obtiene que $W_0 = \frac{P_0}{Q_0} = 1$. Es decir que dicha relación entre los parámetros siempre es fija y es la unidad.



3.4.2 Caracterización de movimientos reales simultáneos de las familias 1 y 2

Considere que las familias 1 y 2 de los movimientos simultáneos reales se asocian con la relación tricotómica de la manera siguiente:

- 1) $P_n > Q_n$; F_1
- 2) $P_n < Q_n$; F_2

Los siguientes movimientos se caracterizan de las relaciones anteriormente descritas;

- 1) $MRS_{P_n > Q_n}^{F_1}$; movimiento real simultáneo de la familia F_1 .
- 2) $MRS_{P_n < Q_n}^{F_2}$; movimiento real simultáneo de la familia F_2 .

Los códigos de reconocimiento de dichos movimientos o transiciones se presentan en las tablas siguientes:

$MRS_{P_n > Q_n}^{F_1}$		
T	M	
T ₁ , T ₂	M ₁	M ₂
	1	1

Tabla 3.10 Código del movimiento simultáneo $MRS_{P_n > Q_n}^{F_1}$.

$MRS_{P_n < Q_n}^{F_2}$		
T	M	
T ₁ , T ₂	M ₁	M ₂
	1	1

Tabla 3.11 Código del movimiento simultáneo $MRS_{P_n < Q_n}^{F_2}$.

3.4.3 Características importantes de los movimientos simultáneos reales

Las transiciones o movimientos reales simultáneos difieren en:

- 1) Los lugares geométricos de recorrido donde va colocado el efector final.
- 2) Las regiones de barrido de los eslabones.

Dichas diferencias son triviales en el sentido de la relación tricotómica. De hecho, cada familia de movimientos requiere del cálculo de la cinemática inversa, independiente de lo que implica. Para cada movimiento existe una configuración deformada de R_0 diferente.

Considere ahora la siguiente proposición:

"Para que la relación tricotómica definida sobre los parámetros angulares se satisfagan, deben existir tres configuraciones deformadas para R_0 , una para cada familia".

Note que bajo la proposición anterior, no es posible encontrar la relación entre los grados de libertad y el número de movimientos reales simultáneos, puesto que las familias F_0, F_1, F_2 son independientes en el sentido de la relación tricotómica.

Considere también la siguiente proposición:

"Sean $R_0^{M^p}$ y $R_0^{D^p}$ dos configuraciones fijas de R_0 . Existe una familia y solo una de movimientos simultáneos que satisface la transición $T: R_0^{M^p} \rightarrow R_0^{D^p}$ ".

Es necesario hacer notar que el lugar geométrico asociado con un movimiento simultáneo se determina por la gráfica generada por el número de puntos intermedios el cual depende de $E \in \mathbb{N}$ fijo; es decir,

$$E = N(PI)$$

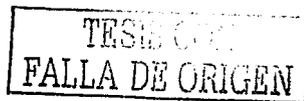
3.4.4 Movimientos complementarios y combinados simultáneos de la familia F_0

En esta sección se hace uso de la hipótesis definida en la sección 3.4.3 para caracterizar los movimientos complementarios y combinados simultáneos. Considere que la familia F_0 está asociada con las configuraciones fijas $R_0^{M^p}$ y $R_0^{D^p}$ de R_0 supóngase además que: $P_0^i = Q_0$

Un movimiento complementario simultáneo relacionado con el robot R_0 es el siguiente:

$$MCPS_{P_0^i=Q_0}^{F_0}$$

El código de dicha transición es el siguiente:



$MCPS_{P_n=Q_n}^{F_n}$		
T	M	M1
M2	T ₁ , T ₂	-1
		-1

Tabla 3.12 Código del movimiento $MCPS_{P_n=Q_n}^{F_n}$.

Observe que el $MCPS_{P_n=Q_n}^{F_n}$ está relacionado con los complementos angulares $P'_0 \in \mathfrak{R}$ y $Q'_0 \in \mathfrak{R}$ de $P_0 \in \mathfrak{R}$ y $Q_0 \in \mathfrak{R}$, respectivamente.

Considere las siguientes relaciones:

- 1) $(P'_0, Q_0) \cong F_0$
- 2) $(P_0, Q'_0) \cong F_0$

Los movimientos simultáneos combinados asociados con la familia F_0 son:

- 1) $MCS_{P'_0, Q_0}^{F_0}$; "movimiento combinado simultáneo familia F_0 con complementación en $P'_0(M_1)$ "
- 2) $MCS_{P_0, Q'_0}^{F_0}$; "movimiento combinado simultáneo familia F_0 con complementación en $Q'_0(M_2)$ "

Los códigos de dichos movimientos son los siguientes:

$MCS_{P'_0, Q_0}^{F_0}$		
T	M	M1
M2	T ₁ , T ₂	-1
		1

Tabla 3.13 Código de $MCS_{P'_0, Q_0}^{F_0}$.



$MCS_{P_0, Q_0}^{F_0}$		
T	M	M1
T ₁ , T ₂		-1

Tabla 3.14 código de $MCS_{P_0, Q_0}^{F_0}$.

3.4.4.1 Consideraciones en los movimientos complementarios y combinados

Los lugares geométricos que recorre el órgano efector terminal del robot R_0 son diferentes para cada movimiento combinado simultáneo y la transición complementaria simultánea.

$$W_1 = \frac{P_0}{Q_0}, \quad 2) \quad W_2 = \frac{P_0}{Q_0} \quad \text{son tales que } W_1 \neq W_2.$$

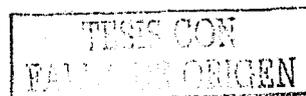
Aquí es importante señalar que aun cuando W_1 sea diferente para cada movimiento necesariamente T_1 y T_2 deben ejecutarse y terminar al mismo tiempo. Y lo mismo se aplica para el caso de W_2 .

3.4.5 Relaciones generales de los movimientos simultáneos

Los movimientos simultáneos, al igual que las transiciones uno a uno, dependen del grado de libertad de R_0 , esto es:

1) $N(MRS) = (n-1)!$; 2) $N(MCPS) = (n-1)!$; 3) $N(MCS) = n!$; 4) $N(F) = n-1!$ si y solo si F es fija. Aquí, $N(MRS)$ es el número de movimientos reales simultáneos, $N(MCPS)$ es el número de movimientos complementarios simultáneos y $N(MCS)$ es el número de movimientos combinados simultáneos, luego existen 4 movimientos simultáneos.

Finalmente, es necesario recordar que, la relación tricotómica definida entre los desplazamientos angulares solo es utilizada para efectos de diferenciación entre las familias de movimientos.



- 1) $F_0 = \{ MRS_{P_0=Q_0}^{F_0}, MCPS_{P_0=Q_0}^{F_0}, MCS_{P_0^*Q_0}^{F_0}, MCS_{P_0Q_0^*}^{F_0} \}$
- 2) $F_1 = \{ MRS_{P_0>Q_0}^{F_1}, MCPS_{P_0>Q_0}^{F_1}, MCS_{P_0^*Q_0}^{F_1}, MCS_{P_0Q_0^*}^{F_1} \}$
- 3) $F_2 = \{ MRS_{P_0<Q_0}^{F_2}, MCPS_{P_0<Q_0}^{F_2}, MCS_{P_0^*Q_0}^{F_2}, MCS_{P_0Q_0^*}^{F_2} \}$

3.5 Movimientos híbridos

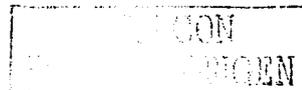
Se analizan y caracterizan los movimientos completos del tipo híbridos relacionados con el robot de estudio. Un movimiento híbrido es una combinación de dos tipos de movimientos, un movimiento uno a uno y un simultáneo. El movimiento híbrido se caracteriza de la manera siguiente: Sean P_0 y Q_0 los parámetros de rotación relacionados con el robot motivo de estudio.

Definición 3.12. *Un movimiento es híbrido si para pasar el robot de una configuración no deformada a una deformada los actuadores Ac_1 y Ac_2 aplican en una primera activación con movimientos simultáneos hasta alcanzar P_0 y posteriormente con movimientos uno a uno, Ac_2 continua su movimiento hasta alcanzar Q_0 o bien de manera inversa, es decir, los actuadores Ac_1 y Ac_2 aplican en una primera activación con movimientos simultáneos hasta alcanzar Q_0 y posteriormente con movimientos uno a uno Ac_1 continua su movimiento hasta alcanzar P_0 ”.*

La tabla 3.15 muestra un movimiento real híbrido:

$MRH^{F_1M_2}$		
T	M1	M2
M		
T_1	1	1
T_2	0	1

Tabla 3.15 Código de movimiento híbrido real.



Note en la tabla 3.15 en T1 se produce el movimiento simultáneo y en T2 el movimiento uno a uno. Así mismo en T1 los actuadores Ac_1 y Ac_2 aplican P_0 y, posteriormente, Ac_2 termina de aplicar Q_0 .

En las tablas 3.16 y 3.17 muestran un movimiento híbrido complementario y un combinado.

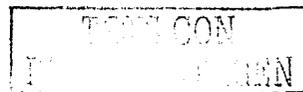
$MCPH^{EM_2}$		
T M	M1	M2
T₁	-1	-1
T₂	0	-1

Tabla 3.16 Código de movimiento híbrido complementario.

MCH^{EM_2}		
T M	M1	M2
T₁	-1	1
T₂	0	-1

Tabla 3.17 Código de movimiento híbrido combinado.

Al realizar la cuantificación de este tipo de movimientos, de acuerdo a la definición 3.12, se obtienen 16 movimientos híbridos. Además, cada movimiento de este tipo es diferente a los demás tanto en términos de las trayectorias del punto terminal como en las regiones de barrido de sus eslabones. Este tipo de movimiento será utilizado en los capítulos posteriores para realizar el proceso de evasión de obstáculos.



Capítulo 4

El problema de evasión de obstáculos

Introducción

En este capítulo se define y analiza el problema de evasión de obstáculos en el plano con sus respectivas restricciones. Se caracterizan las regiones de barrido, la trayectoria de trabajo y la localización de obstáculos. También se realiza una representación teórica del impacto y distancias de seguridad, y finalmente se genera un conjunto de criterios lógicos para la evasión de obstáculos.

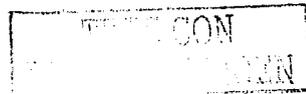
Se considerarán como elementos primitivos del problema de evasión de obstáculos los siguientes:

- 1) El robot de 2GDL estudiado en esta tesis.
- 2) Un obstáculo de geometría circular.
- 3) Un conjunto de movimientos completos del robot.
- 4) Un conjunto de elementos de señalización (sensores).

Cabe señalar que, los procesos de análisis y síntesis del problema de evasión se realizará bajo el método científico y, los resultados obtenidos (criterios), serán usados para controlar un prototipo de robot el cual evadirá obstáculos reales localizados en una región de trabajo. Finalmente, se introducirá un lenguaje simbólico, el cual será de utilidad para los procesos de análisis y síntesis del problema.

4.1 Definición del problema y restricciones

“ Se requiere modelar un problema de evasión de obstáculos en el plano caracterizado por un robot de 2GDL (cadena abierta) y un obstáculo de geometría circular. Se deben utilizar el conjunto de movimientos del robot analizados y representados en el capítulo 3, así como un conjunto de elementos de señalización (sensores) localizados en lugares estratégicos del robot. El modelo generado, debe integrar un conjunto de criterios de evasión. ”



Las restricciones del problema formulado son los siguientes:

- 1) El robot consta de dos elementos rígidos dimensionados conectados por juntas rotacionales. Donde l_1 la dimensión del eslabón 1 (medida de junta a junta) y l_2 la dimensión del eslabón 2 y satisfacen la relación $l_1 > l_2$ (R_1).
- 2) Los eslabones del robot pueden girar una revolución completa (R_2).
- 3) El área de trabajo del robot es finita, conocida y fija (R_3).
- 4) El robot puede realizar 1) movimientos uno a uno, 2) simultáneos, 3) híbridos y finalmente movimientos simultáneos que siguen una ruta (R_4).
- 5) El robot está equipado con cuatro sensores localizados en las caras laterales de los eslabones. Dichos sensores no sensan regiones próximas a las juntas que conectan dichos eslabones (incluyendo las propias juntas) y la región donde se localiza la herramienta de trabajo o punto terminal del robot (R_5).
- 6) El obstáculo tiene geometría circular, está fijo en el área de trabajo y el radio (r_0) satisface la siguiente desigualdad: $0 < r_0 \leq \frac{l_2}{4}$ (R_6).
- 7) Los eslabones de los cuales consta el robot no deben "tocar" el obstáculo (R_7).
- 8) El conjunto de criterios de evasión debe considerar los movimientos completos del robot (R_8).
- 9) El problema de evasión de obstáculos es atemporal (R_9).
- 10) Los obstáculos se localizan en una región restringida del área de trabajo (R_{10}).
- 11) El obstáculo no debe localizarse sobre la trayectoria de trabajo del robot (R_{11}).

El problema definido anteriormente y sus restricciones se sintetiza de la manera siguiente:

•• Dados, R_0, O_R, C_{A_T}, A_T , y S, encuentre:

$$1) R_0 \cap O_R = \emptyset \text{ o } R_0 \cap O_R \neq \emptyset \text{ y}$$

$$2) C_{A_T} = f_{in}(R_0, O_R, A_T, S, C_{A_T})$$

tal que:

1) $R_1 = \hat{R}_1$; 2) $R_2 = \hat{R}_2$; 3) $R_3 = \hat{R}_3$; 4) $R_4 = \hat{R}_4$; 5) $R_5 = \hat{R}_5$; 6) $R_6 = \hat{R}_6$;

7) $R_7 = \hat{R}_7$; 8) $R_8 = \hat{R}_8$; 9) $R_9 = \hat{R}_9$; 10) $R_{10} = \hat{R}_{10}$; 11) $R_{11} = \hat{R}_{11}$.

sean satisfechas."

4.2 Hipótesis y consideraciones sobre el problema de evasión de obstáculos

En esta sección se propone una hipótesis y axiomas básicos, mediante la cual, se podrá solucionar el problema formulado en la sección 4.1. Considere la siguiente proposición (hipótesis):

"Todo proceso de evasión de obstáculos está en función del movimiento del robot o del obstáculo o de ambos. Para que dicho proceso se genere, es necesario y suficiente que el evento: $R_r^{i'n} \cap \partial O_h \neq \emptyset$ se presente en A_r y las formas en que el robot evade o no evade el obstáculo O_h está en función de un conjunto de criterios lógicos relacionados con las dimensiones principales del robot y el obstáculo, los movimientos completos del robot y la localización del obstáculo en A_r . El proceso de evasión es temporal y satisface las leyes del movimiento de Newton."

Los símbolos $R_r^{i'n}$ y ∂O_h representan "el robot dimensionado equipado con sensores" y la frontera del obstáculo, respectivamente.

Los axiomas básicos asociados con el problema de evasión es el siguiente:

Axioma 4.1 En $R_r \subseteq A_r$ se presentan los eventos:

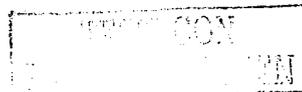
$$1) R_r^{i'n} \cap O_h \neq \emptyset \quad \circ$$

$$2) R_r^{i'n} \cap O_h = \emptyset$$

aquí. R_c es una región de contacto.

Axioma 4.2 El movimiento del robot $R_r^{i'n}$ y del obstáculo O_h satisfacen las leyes del movimiento de Newton.

Axioma 4.3 El robot $R_r^{i'n}$ ejecuta movimientos completos definidos en dos configuraciones: 1) la no deformada y 2) la deformada.



En lo referente a las restricciones que se consideran en esta tesis, es que no se estudiarán los procesos de evasión dinámicos entre el robot y el obstáculo. puesto que el obstáculo localizado en A_r está fijo, antes y después del evento $R_0^{DS} \cap O_n \neq \emptyset$. Además, el robot usará movimientos completos para el proceso de evasión definidos de una configuración no deformada a una deformada. Se supondrá que durante la transición del robot entre dichas configuraciones es donde se presenta la condición de impacto.

Por otro lado, la velocidad y la aceleración asociadas con el proceso de evasión, no serán consideradas en este trabajo.

Tomando en cuenta estas consideraciones, se generarán los criterios lógicos mediante los cuales, el robot R_0^{DS} ejecutará el proceso de evasión

4.3 La condición de impacto y diferentes tipos de impactos

Considere que el robot R_0^{DS} dimensionado y equipado con sensores realiza un movimiento completo en el área de trabajo A_r . Supóngase además que, en una región $R_t \subseteq A_r$ se localiza un obstáculo O_n de geometría circular y fijo en el tiempo.

Se dirá que existe un evento de contacto o impacto entre el robot R_0^{DS} y O_n si y solo si:

$$R_0^{DS} \cap O_n \neq \emptyset \quad \text{en} \quad R_t.$$

Aquí, R_0^{DS} es el robot R_0^{DS} representado en un proceso de ejecución de un movimiento completo, es decir, el impacto se presenta durante el desplazamiento angular que realiza el robot.

Obsérvese que los elementos primitivos de un evento de impacto son:

- 1) El robot R_0^{DS} .
- 2) Un obstáculo fijo O_n .
- 3) Una región de contacto $R_t \subseteq A_r$ tal que $O_n \subseteq R_t$ o $O_n \cap R_t \neq \emptyset$.
- 4) Un movimiento completo.

Sea R_n'' el robot dimensionado y S un conjunto de sensores. entonces: $R_n'^{S} = R_n'' \cup S$

Nótese que siendo $R_n'' = E_1^D \cup E_2^D$ y S el conjunto de sensores, entonces:

$$R_n'^{S} = E_1'^{S} \cup E_2'^{S}$$

Considere ahora que R_n'' es analítico; es decir, $R_n'' = \partial R_n'' \cup IR_n''$

o, equivalentemente: $R_n'' = (\partial E_1^D \cup IE_1^D) \cup (\partial E_2^D \cup IE_2^D)$

Aquí, “ ∂ ” y “I” son el interior y la frontera del robot o de sus eslabones, respectivamente.

Por otro lado, sea O_B el obstáculo tal que: $O_B = IO_B \cup \partial O_B$

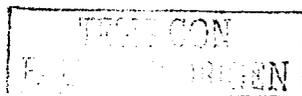
Entonces el evento: $\ddot{\partial} R_n'' \cap \partial O_B \neq \emptyset$ es llamado “impacto real”. Aquí, $\ddot{\partial} R_n''$ es la frontera móvil de R_n'' . Cabe señalar que las condiciones $\ddot{I} R_n'' \cap \partial O_B = \emptyset$ o $\ddot{\partial} R_n'' \cap O_B = \emptyset$ son inadmisibles por el principio de impenetrabilidad.

Sea R_n un robot “reducido” en el cual sus eslabones se representan por dos líneas rectas definidas de los puntos centrales de juntas y punto terminal. Sea también O_n una circunferencia. El evento: $\ddot{R}_n \cap O_n \neq \emptyset$ es llamado “impacto abstracto”; nótese que \ddot{R}_n es en realidad un sistema de dos rectas en movimiento.

Por tanto, los diferentes tipos de impactos se caracterizan de la manera siguiente:

- 1) $\ddot{R}_n \cap O_n \neq \emptyset$ es llamado “impacto abstracto”.
- 2) $\ddot{R}_n'' \cap O_n \neq \emptyset$ es llamado “impacto real”.
- 3) $\ddot{R}_n'^{S} \cap O_n \neq \emptyset$ es llamado “impacto sentido”.

La figura siguiente muestra los tres tipos de impacto caracterizados anteriormente:



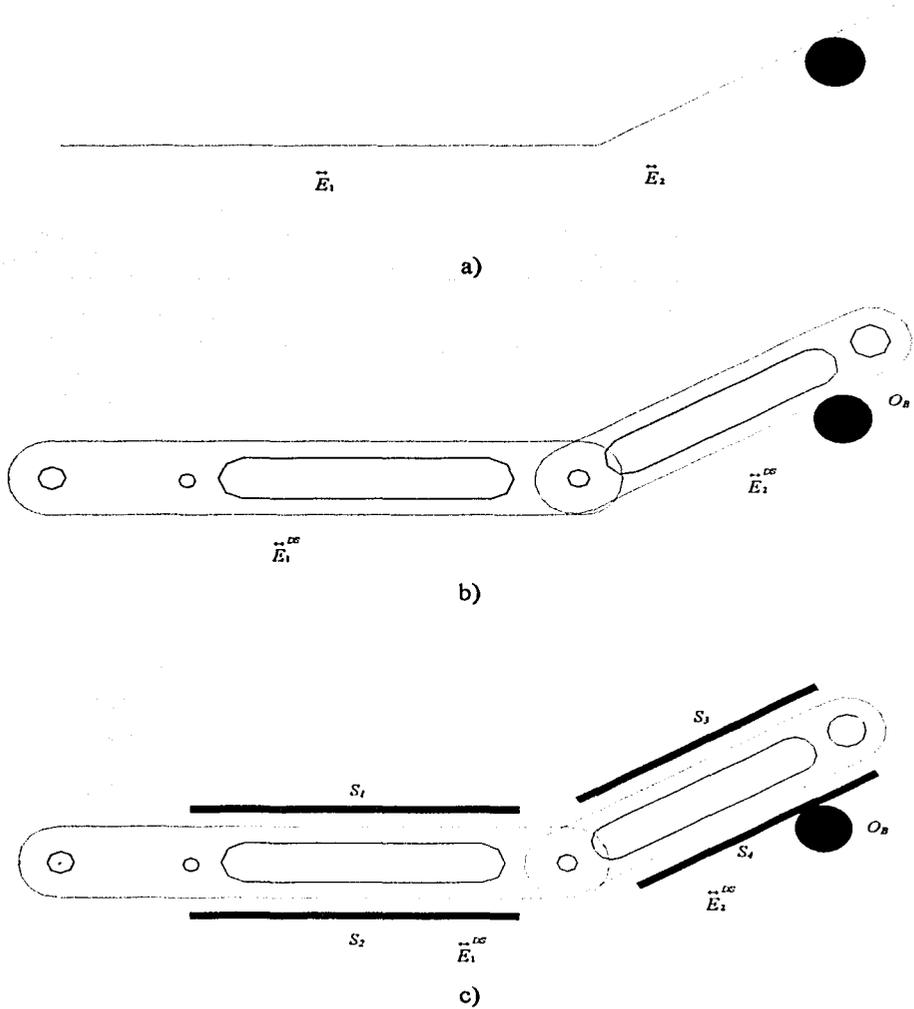


Figura 4.1 Diferentes impactos $\vec{R}_0 \cap O_B \neq \emptyset$: a) abstracto, b) real y c) sentido.

IMPRESION CON
 DE ORIGEN

4.4 Sensores, regiones de barrido, trayectoria de trabajo y localización de obstáculos

Se analizan los elementos primitivos de sensado, las regiones de operación y localización del robot R_0^{DS} y el obstáculo en el área de trabajo. Considere que el robot R_0^{DS} se puede desplazar libremente por el área de trabajo mostrada en la figura 4.2.

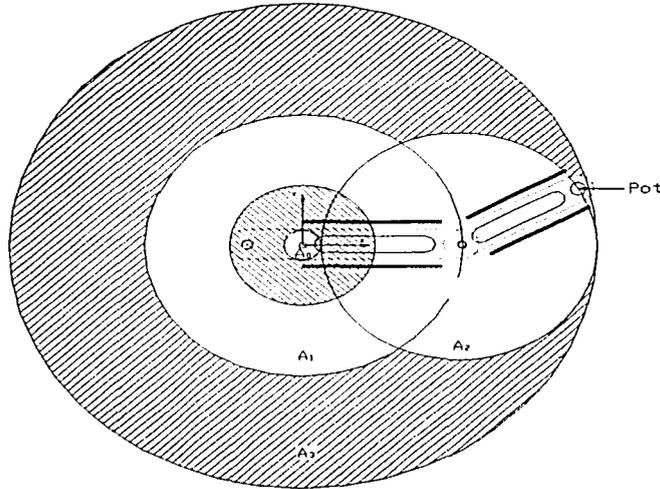


Figura 4.2 Área de trabajo del robot R_0^{DS} .

Nótese en la figura 4.2 que, por la condición $l_1 > l_2$ el área de trabajo real del robot es:

$$A_r = A_3 - A_0$$

Además:

- 1) A_3 es la circunferencia generada por el radio $r_3 = l_1 + l_2$ con centro en $(0, 0)$.
- 2) A_1 es la circunferencia generada por el radio $r_2 = l_1$ con centro en $(0, 0)$.
- 3) A_2 es la circunferencia generada por el radio $r_1 = l_2$ con centro en (h_1, k_1) .

El área de trabajo de un robot es la región en la cual cualesquier punto contenido en dicha región puede ser alcanzado por el órgano terminal o punto terminal "Pot". Como se muestra en la figura 4.2.

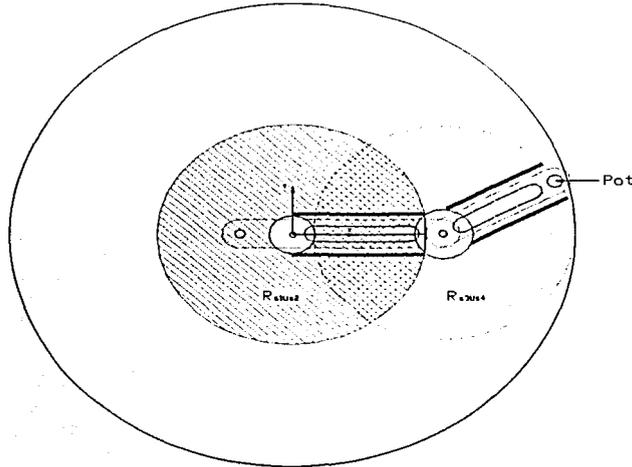


Figura 4.3 Regiones de sensorio en el robot de estudio.

De acuerdo con la figura 4.3, en A_T existen las regiones:

- 1) $R_{S1 \cup S2} \subseteq A_T$; región de sensorio primario.
- 2) $R_{S3 \cup S4} \subseteq A_T$; región de sensorio secundario.

Cabe señalar que las regiones de sensorio primario y secundario son fijas para la posición mostrada en la figura 4.3, sin embargo, dichas regiones son “desplazables” durante la ejecución de los movimientos realizados por el robot R_0^{DS} . Las regiones $R_{S1 \cup S2}$ y $R_{S3 \cup S4}$ son llamadas “regiones de barrido sensorio” durante un movimiento.

Por otro lado, el robot ejecutará una determinada tarea sobre A_T . Dicha tarea se realiza sobre un lugar geométrico τ tal que $\tau \subseteq A_T$. Cabe señalar que dicho lugar geométrico τ está asociado con el órgano terminal del robot en estudio.

El lugar geométrico τ se caracteriza en:

- 1) Descripción del lugar geométrico de ida.
- 2) Descripción del lugar geométrico de vuelta.



La figura siguiente muestra la ruta de trabajo o trayectoria del órgano terminal del robot.

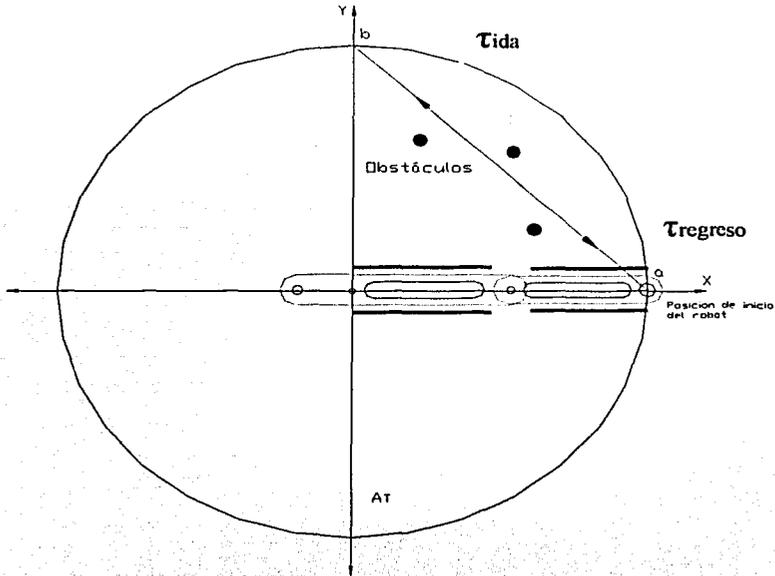


Figura 4.4 Trayectoria de trabajo, región de contacto y posición de inicio del robot.

Note en la figura 4.4 que el ciclo de operación del robot se caracteriza por:

$$\tau = \tau_{ida} \cup \tau_{regreso}$$

o, equivalentemente:

$$\tau_{a-b-a} = \tau_{a-b} \cup \tau_{b-a}$$

Por otro lado, la posición que guarda el robot mostrada en la figura 4.4 es llamada "la configuración no deformada" o posición de inicio, en tanto R_c es la región de contacto; es decir, la región en donde se presentará el evento:

$$R_0^{DS} \cap O_b \neq \emptyset$$

4.5 Representación teórica del impacto y distancias de seguridad

Considere el robot R_0 reducido y el obstáculo O_B localizado en $R_i \subseteq A_i$. Sea $R_0 \cap O_B \neq \emptyset$ fijo; es decir, un evento en el cual, el robot R_0 está fijo.

Supóngase además que, el obstáculo es una circunferencia descrita por la ecuación general siguiente:

$$Ax^2 + By^2 + Cx + Dy + F = 0$$

Sea l_1 la recta asociada con el eslabón E_1 descrita por la ecuación: $y = mx$. Aquí, "m" es la pendiente de la recta.

Considere ahora que la ecuación de la recta es substituida en la expresión de la circunferencia; esto es: $Ax^2 + B(mx)^2 + Cx + D(mx) + F = 0$. Explícitamente:

$$\begin{aligned} Ax^2 + Bm^2x^2 + Cx + Dmx + F &= 0 \quad \text{o} \\ x^2(A + Bm^2) + x(C + Dm) + F &= 0 \end{aligned}$$

Sean G_1 , G_2 y G_3 tales que:

- 1) $G_1 = A + Bm^2$
- 2) $G_2 = C + Dm$
- 3) $G_3 = F$

La solución a la cuadrática es la siguiente:

$$x_{1,2} = \frac{-G_2 \pm \sqrt{G_2^2 - 4G_1G_3}}{2G_1}$$

o, explícitamente:

$$x_{1,2} = \frac{-(C + Dm) \pm \sqrt{(C + Dm)^2 - 4(A + Bm^2)F}}{2(A + Bm^2)}$$

Considere que $D_S = G_2^2 - 4G_1G_3$ es el discriminante, entonces:

- 1) Si $D_S > 0$, entonces $l_1 \cap O_B \neq \emptyset$
- 2) Si $D_S < 0$, entonces $l_1 \cap O_B = \emptyset$

En otras palabras:

- 1) Si $x_1 = x_2$ y son dos raíces reales, entonces existe uno y solo un punto de contacto (intersección) tal que: $l_1 \cap O_B \neq \emptyset$.

- 2) Si $x_1 \neq x_2$ y son raíces reales, entonces existen dos puntos de contacto (intersecciones) tal que: $I_1 \cap O_B \neq \emptyset$.
- 3) Si las raíces son complejas, entonces no existen puntos de intersección y, por tanto, $I_1 \cap O_B = \emptyset$.

La figura siguiente muestra las tres condiciones descritas anteriormente:

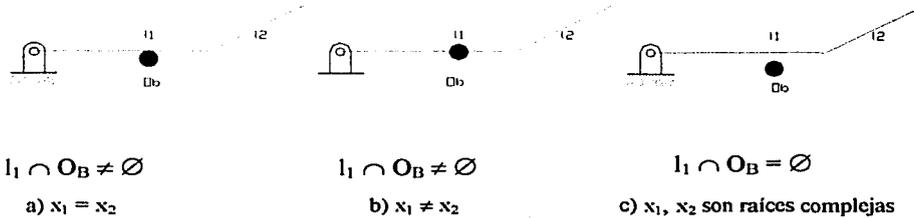


Figura 4.5 Consideraciones geométricas del impacto.

Es importante señalar que las constantes A, B, C, D y F relacionadas con la ecuación general de la circunferencia, son deducidas si se conocen las coordenadas del centro de obstáculo y su radio. Asimismo, las coordenadas del impacto se obtienen al solucionar la cuadrática y, posteriormente, sustituir dichas soluciones en la ecuación de la recta.

Por otro lado, para modelar el impacto entre la recta l_2 y el obstáculo, basta con conocer la ecuación de la recta l_2 y sustituirla en la expresión de la circunferencia.

Considere ahora el robot R_0^p ; es decir, el robot dimensionado el cual es mostrado en la figura 4.6.

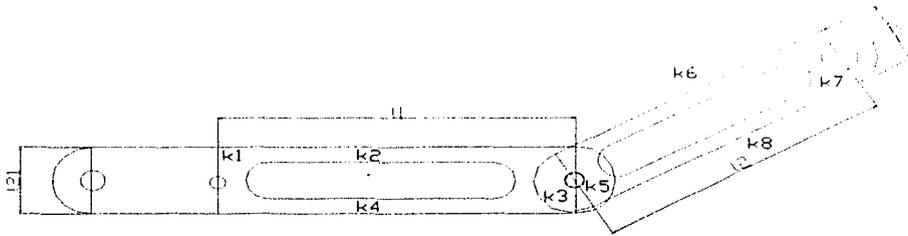


Figura 4.6 El robot dimensionado R_0^D .

De acuerdo con la figura 4.6, el robot R_0^D está formado por "cuerpos dimensionados". Sean k_1, k_2, k_3 y k_4 un conjunto de rectas relacionadas con el eslabón E_1^D y k_5, k_6, k_7 y k_8 el otro conjunto de rectas asociados con el eslabón E_2^D . Dichos sistemas se muestran en la figura 4.6.

Supóngase que el evento $R_0^D \cap O_B \neq \emptyset$ se presenta en $R_C \subseteq A_T$. Entonces:

- 1) Si $k_2 \cap O_B \neq \emptyset \Rightarrow R_0^D \cap O_B \neq \emptyset$
- 2) Si $k_4 \cap O_B \neq \emptyset \Rightarrow R_0^D \cap O_B \neq \emptyset$
- 3) Si $k_6 \cap O_B \neq \emptyset \Rightarrow R_0^D \cap O_B \neq \emptyset$
- 4) Si $k_8 \cap O_B \neq \emptyset \Rightarrow R_0^D \cap O_B \neq \emptyset$

Obsérvese que los eventos: $k_1 \cap O_B = k_3 \cap O_B = k_5 \cap O_B = k_7 \cap O_B \neq \emptyset$

son inadmisibles por el principio de impenetrabilidad. La figura 4.7 muestra los cuatro eventos posibles de impacto real.

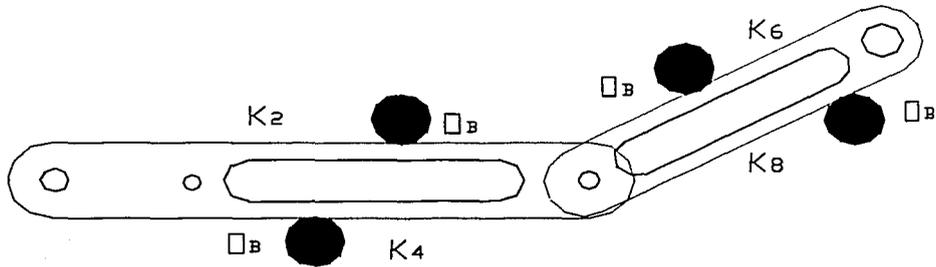


Figura 4.7 Impactos reales entre O_B y R_0^D .

En este trabajo se considerará el robot R_0^D equivalente a: $R_0^D \approx k_0^D$. Es decir, el robot dimensionado es equivalente al robot formado por rectas.

Es importante señalar que, para modelar teóricamente los impactos reales, es necesario conocer:

- 1) El centro y radio de las circunferencias asociadas con los obstáculos y,
- 2) Las ecuaciones de las rectas k_2 , k_4 , k_6 y k_8 .

4.5.1 Modelación de los impactos de sentido

Considere ahora que los eslabones del robot R_0^D tienen asociados elementos de sentido como se muestra en la figura 4.8.

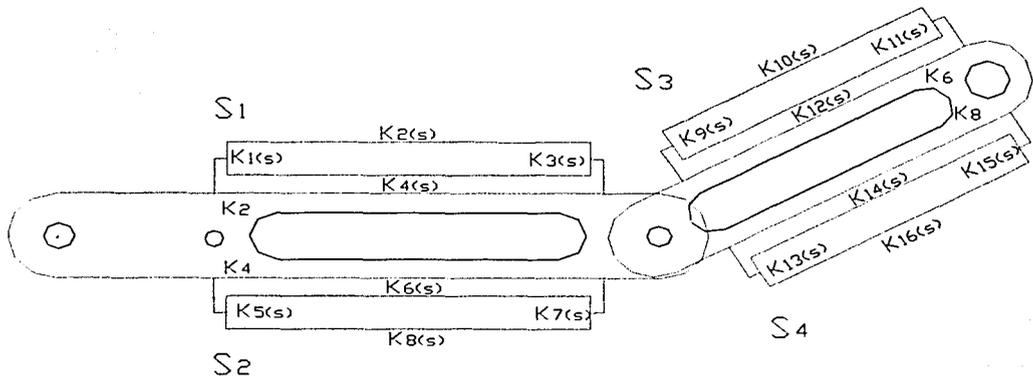


Figura 4.8 El robot dimensionado con elementos de sentido.

De acuerdo con la figura 4.8, cada sensor asociado con los eslabones que componen el robot R_n^{in} genera una región de sensado caracterizado por un sistema de rectas propio.

Considere ahora que un sistema de obstáculos fijos es sensado por el sistema de sensores. Las siguientes condiciones de "impacto sensado" se consideran admisibles:

- 1) Si $S_1 \cap O_H \neq \emptyset \Rightarrow k_{2(S)} \cap O_H \neq \emptyset$
- 2) Si $S_2 \cap O_H \neq \emptyset \Rightarrow k_{8(S)} \cap O_H \neq \emptyset$
- 3) Si $S_3 \cap O_H \neq \emptyset \Rightarrow k_{10(S)} \cap O_H \neq \emptyset$
- 4) Si $S_4 \cap O_H \neq \emptyset \Rightarrow k_{16(S)} \cap O_H \neq \emptyset$

Nótese además que, las siguientes condiciones se deben satisfacer:

- a) $k_{10(S)} \cap k_2 = \emptyset$
- b) $k_{16(S)} \cap k_4 = \emptyset$
- c) $k_{12(S)} \cap k_6 = \emptyset$
- d) $k_{14(S)} \cap k_8 = \emptyset$

Cabe señalar que, si dichas condiciones no se cumplen, los sensores detectarán los propios eslabones lo cual se considera inadmisibile.

Por tanto, las rectas $k_{2(S)}$, $k_{8(S)}$, $k_{10(S)}$ y $k_{16(S)}$ son consideradas límites externos de sensado y, para determinar los impactos de sensado es necesario conocer:

- 1) Los centros y radios de las circunferencias relacionadas con los obstáculos.
- 2) Las ecuaciones de las rectas $k_{2(S)}$, $k_{8(S)}$, $k_{10(S)}$ y $k_{16(S)}$.



La figura 4.9 muestra los cuatro posibles impactos sentidos.

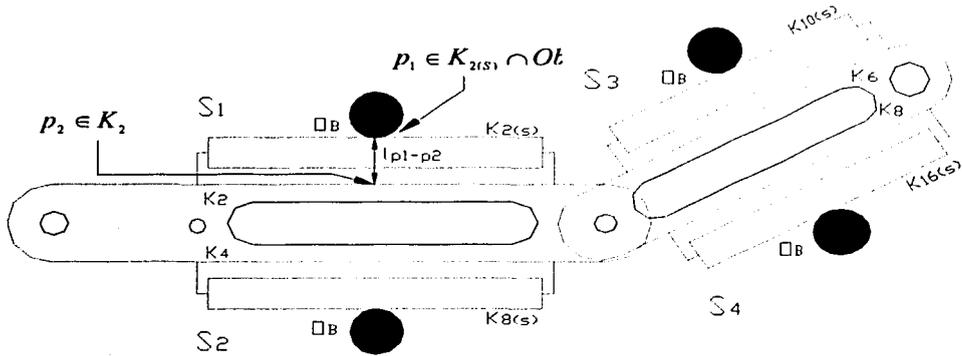


Figura 4.9 Posibles eventos de impactos de sensores.

4.5.2 Distancias de seguridad

Sea $p_1 \in k_{2(s)} \cap O_B \neq \emptyset$ y $p_2 \in k_2$ dos puntos de contacto, uno asociado con el sensor S_1 y otro con la recta k_2 definida sobre el eslabón E_2 la cual es mostrada en la figura 4.9. Supóngase que se define una recta sobre dichos puntos denominada $l_{p_1-p_2}$ (ver figura 4.9) tal que:

- 1) $l_{p_1-p_2} \perp k_{2(s)}$
- 2) $l_{p_1-p_2} \perp k_2$

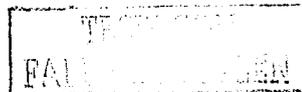
es decir, $l_{p_1-p_2}$, es perpendicular a las rectas paralelas $k_{2(s)}$ y k_2 .

Sea " l_s " una distancia tal que: $l_s > d(p_1 - p_2)$, dicha distancia es llamada de "seguridad primaria".

Es importante señalar que, el evento que no se debe presentar realmente es:

$$R_0^D \cap O_B \neq \emptyset$$

pues dicho evento implicaría que exista un impacto real entre el obstáculo y el robot el cual se debe evitar.



Por otro lado, la distancia " I_s " se llama de "seguridad primaria" por el hecho de que ha sido caracterizada por medio de eventos fijos; es decir, no se considera el movimiento del robot. Además, los puntos de impacto entre la recta $k_{2(s)}$ y O_H y k_2 y O_H no siguen una relación lineal ya que el movimiento del robot o, más bien dicho, el "barrido de los eslabones" es, en general, un movimiento rotacional y no lineal (solo el punto terminal del robot realiza en ocasiones movimientos rectilíneos si así se requiere en una operación específica). Sin embargo la distancia de seguridad primaria " I_s " será usada más adelante para determinar la distancia efectiva de seguridad.

4.5.3 Impacto por movimientos completos

El análisis de impacto entre el robot y el obstáculo realizado en las secciones anteriores, fue realizado considerando fijos los eventos $R_0 \cap O_H \neq \emptyset$, $R_0'' \cap O_H \neq \emptyset$ y $R_0^{DS} \cap O_H \neq \emptyset$.

En esta sección se analizará el evento:

$$R_0^{DS} \cap O_H \neq \emptyset \quad \text{en } R_t \subseteq A_t$$

Sea C_t el conjunto de movimientos completos asociados con el robot R_0^{DS} . Considere que la "posición no deformada" del robot es ${}^{DS}R_0^{ND}$; es decir, la configuración no deformada y ${}^{DS}R_0''$ es la posición o configuración deformada. El evento:

$$T : {}^{DS}R_0^{ND} \rightarrow {}^{DS}R_0''$$

es una transición de movimiento del robot completa del punto "a" al punto "b" sobre la trayectoria de ida τ_{a-b} . Asimismo, el evento:

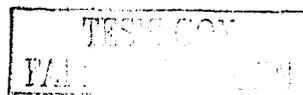
$$T^{-1} : {}^{DS}R_0'' \rightarrow {}^{DS}R_0^{ND}$$

es una transición de movimiento del robot definida sobre la trayectoria de vuelta τ_{b-a} .

Nótese que:

- 1) ${}^{DS}R_0''$ está definida por el punto "b" si τ es de ida.
- 2) ${}^{DS}R_0''$ se define sobre el punto "a" si es de vuelta.

Cabe señalar que:



$$T \circ T^{-1} = T^{-1} \circ T = I$$

Se dirá entonces que el conjunto C_M está caracterizado por las funciones T y T^{-1} , de modo que el robot R_0^{DS} puede utilizar movimientos uno a uno, simultáneos e híbridos durante las transiciones T y T^{-1} , respectivamente.

Considere la siguiente definición:

“Todo evento $R_0^{DS} \cap O_B \neq \emptyset$ o $R_0^{DS} \cap O_B = \emptyset$ se presenta durante las transiciones $T: {}^{DS}R_0^{ND} \rightarrow {}^{DS}R_0^D$ y/o $T^{-1}: {}^{DS}R_0^D \rightarrow {}^{DS}R_0^{ND}$.”

En otras palabras:

“Todo evento de impacto sentido o no, se presenta durante la trayectoria de ida y/o vuelta del robot R_0^{DS} el cual ejecuta movimientos completos.”

Sea $\overset{DS}{R}_0$ el robot en movimiento y supóngase que existe un evento $R_0^{DS} \cap O_B \neq \emptyset$ en $R_C \subseteq A_T$. Entonces, existirá una configuración llamada de “impacto en movimiento” representada por ${}^I R_0^{DS}$ y otra configuración denominada de “seguridad efectiva” caracterizada por el evento $R_0^{DS} \cap O_B = \emptyset$. Dicha configuración es indicada por el símbolo ${}^{SE} R_0^{DS}$ la cual es mostrada en la figura siguiente:

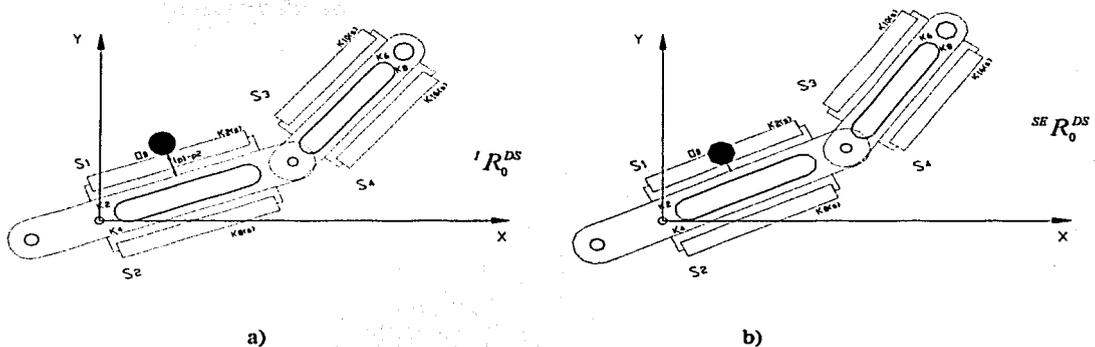


Figura 4.10 Configuraciones de impacto: a) en movimiento, b) de seguridad efectiva.



Observe en la figura anterior que la configuración ${}^{SE}R_0^{DS}$ debe estar fija, en otras palabras, el robot debe detenerse en ${}^{SE}R_0^{DS}$ para que no impacte realmente con el objeto. En la figura 4.11 se muestra la distancia de seguridad efectiva en movimiento definida entre ${}^I R_0^{DS}$ y ${}^{SE}R_0^{DS}$.

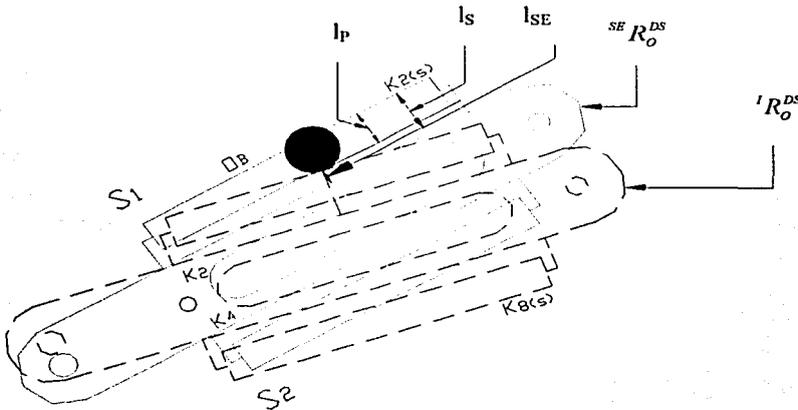


Figura 4.11 Distancia de seguridad efectiva.

Por tanto, la distancia de seguridad efectiva fija l_{SE} se obtiene de la manera siguiente:

$$l_{SE} = l_s - l_p$$

Es importante señalar, que para todos los sensores deben ser fijas las distancias l_{SE} .

4.6 Jerarquerización de movimientos completos debido tiempo total de desplazamiento angular.

En esta sección se analizarán los tiempos totales de recorrido angular asociados con los movimientos completos de C_M . Dichos tiempos serán usados en la siguiente sección para proponer un conjunto de criterios lógicos para el proceso de evasión.



Considere que un actuador asociado con el robot motivo de estudio ejecuta un desplazamiento angular caracterizado por la constante 2π . En otras palabras, dicho actuador ejecuta una revolución. Supóngase que el tiempo usado para tal ejecución es T_k siempre fijo.

Sean $\alpha_1, \alpha_2 \in \mathfrak{R}$ los desplazamientos angulares obtenidos al solucionar la cinemática inversa. Dichos ángulos son usados por los actuadores del robot para ejecutar la transición $T : {}^{WS}R_0^{SD} \rightarrow {}^{WS}R_0^D$.

Las relaciones siguientes son usadas para encontrar los tiempos de ejecución durante el movimiento del robot:

$$1) T_{a1} = \frac{T_k \cdot \alpha_1}{2\pi}$$

$$2) T_{a2} = \frac{T_k \cdot \alpha_2}{2\pi}$$

Obsérvese, que siendo $P_0^* = 1 - P_0$ y $Q_0^* = 1 - Q_0$, entonces:

$$3) T_{a_i} = \frac{T_k(\alpha_1 - 2\pi)}{2\pi}$$

$$4) T_{a_i} = \frac{T_k(\alpha_2 - 2\pi)}{2\pi}$$

Aquí, $\alpha_1^* = \alpha_1 - 2\pi$ y $\alpha_2^* = \alpha_2 - 2\pi$

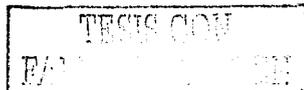
Cabe señalar que el conjunto C_M está caracterizado en términos del tiempo total de ejecución por las cuatro expresiones siguientes:

$$1) T_{ia} = T_{a_1} + T_{a_2} = T_{a_2} + T_{a_1}$$

$$2) T_{ib} = T_{a_i} + T_{a_i} = T_{a_i} + T_{a_i}$$

$$3) T_{ic} = T_{a_1} + T_{a_i} = T_{a_i} + T_{a_1}$$

$$4) T_{id} = T_{a_i} + T_{a_2} = T_{a_2} + T_{a_i}$$



Aquí, T_{iu} es el tiempo asociado con las transiciones completas reales, T_{ib} es el tiempo relacionado con las transiciones complementarias, T_{i_1} y T_{i_2} son los tiempos relacionados con los movimientos o transiciones combinados.

En términos de los tipos de movimientos, los tiempos totales se caracterizan de la manera siguiente:

- 1) Tiempo total de los movimientos uno a uno (T_{IMU}).
- 2) Tiempo total de los movimientos simultáneos (T_{IMS}).
- 3) Tiempo total de los movimientos híbridos (T_{IMH}).

En general, $T_{IMS} < T_{IMH} < T_{IMU}$.

Es decir que el tiempo total de movimiento simultáneo es menor que el híbrido y éste a la vez menor que el uno a uno. Esto es por la instantaneidad o no, en que se ejecutan los movimientos de los eslabones.

4.7 Definición de los elementos participantes en el evento $R_o^{IS} \cap O_H \neq \emptyset$ y los criterios de evasión

En esta sección se proponen criterios lógicos de evasión mediante los cuales, el robot R_o^{IS} evadirá los obstáculos en la región $R_i \subseteq A_f$. Se usarán las operaciones $\bullet : \{1,0\} \times \{1,0\} \rightarrow \{1,0\}$ y $+: \{1,0\} \times \{1,0\} \rightarrow \{1,0\}$ llamadas la conjunción y disyunción, respectivamente, así como una relación de impacto sentido y los movimientos del robot.

Sea R_o^{IS} el robot dimensionado equipado con sensores y O_H un obstáculo.

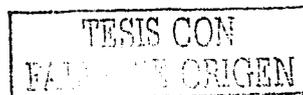
La función,

$$R_1 : R_o^{IS} \times O_H = \{1,0\}$$

definida por:

- 1) $R_{11}(R_o^{IS}, O_H) = 1 \Leftrightarrow R_o^{IS} \cap O_H \neq \emptyset$
- 2) $R_{12}(R_o^{IS}, O_H) = 0 \Leftrightarrow R_o^{IS} \cap O_H = \emptyset$

son relaciones en las cuales puede o no existir un "impacto". Dicha función se puede extender de la manera siguiente:



$$1) R_{11}(E_1^{/s}, O_H) = 1 \quad \text{o} \quad R_{12}(E_1^{/s}, O_H) = 0$$

$$2) R_{11}(E_2^{/s}, O_H) = 1 \quad \text{o} \quad R_{12}(E_2^{/s}, O_H) = 0$$

Nótese que dicha relación también se puede extender para el caso de rectas asociadas con los sensores; esto es: $R_{IR} : 1 \times O_H \rightarrow \{1, 0\}$

Explícitamente:

$$1) (k_2^s, O_H) = 1 \quad \text{o} \quad (k_2^s, O_H) = 0$$

$$2) (k_8^s, O_H) = 1 \quad \text{o} \quad (k_8^s, O_H) = 0$$

$$3) (k_{10}^s, O_H) = 1 \quad \text{o} \quad (k_{10}^s, O_H) = 0$$

$$4) (k_{16}^s, O_H) = 1 \quad \text{o} \quad (k_{16}^s, O_H) = 0$$

Obsérvese que el símbolo "1" es usado para identificar una condición de impacto de sensado y el símbolo "0" se utiliza para identificar un no impacto o no presencia de obstáculo.

Sea $A' = \{1, 0\}$ el conjunto binario. Se definen las operaciones $\bullet : A' \times A' \rightarrow A'$ y $\oplus : A' \times A' \rightarrow A'$ por medio de las tablas siguientes:

\bullet	1	0
1	1	0
0	0	0

a)

\oplus	1	0
1	1	1
0	1	0

b)

Tabla 4.1 Operaciones binarias: a) la conjunción, b) la disyunción.

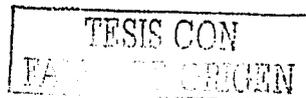
Dichas operaciones serán utilizadas en esta sección para manejar operativamente hablando los símbolos "1" y "0".

Considere que un sensor S_1 se representa por:

$$1) S_1' \Leftrightarrow 1 \Leftrightarrow I_2^s \cap O_H \neq \emptyset$$

$$2) S_1'' \Leftrightarrow 0 \Leftrightarrow I_2^s \cap O_H = \emptyset$$

Es decir, S_1' es un sensor activo y S_1'' es un mismo sensor desactivado.



Sea S_1 y S_2 dos sensores asociados con el eslabón E_1'' . La tabla 4.2 muestra una operación de señales binarias entre éstos:

S_1	S_2	$S_1 \oplus S_2$
1	1	1
0	0	0
0	1	1
1	0	1

Tabla 4.2 Operación de señales en la disyunción.

Así,

- 1) $\oplus (1, 1) = 1 \Leftrightarrow E_1^{DS} \cap O_H \neq \emptyset$
- 2) $\oplus (0, 0) = 0 \Leftrightarrow E_1^{DS} \cap O_H = \emptyset$
- 3) $\oplus (0, 1) = 1 \Leftrightarrow E_1^{DS} \cap O_H \neq \emptyset$
- 4) $\oplus (1, 0) = 1 \Leftrightarrow E_1^{DS} \cap O_H \neq \emptyset$

Por tanto, la operación $\oplus : A' \times A' \rightarrow A'$ se usará como indicadora de eventos de sensado. Cabe mencionar que la operación $\bullet : A' \times A' \rightarrow A'$ no puede usarse para este propósito, pues $\bullet (1, 0) = (0, 1) = 0$ indica sensor encendido pero sin presencia de obstáculo lo cual es inadmisibles. Por otro lado, la tabla 4.3 muestra las relaciones operativas entre los sensores S_3 y S_4 asociados con el eslabón E_2'' .

S_3	S_4	$S_3 \oplus S_4$
1	1	1
0	0	0
0	1	1
1	0	1

Tabla 4.3 Relación de operación de S_3 y S_4

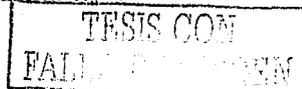
Nótese que:

- 1) $\oplus (1, 1) = 1 \Leftrightarrow E_2^{LN} \cap O_H \neq \emptyset$
- 2) $\oplus (0, 0) = 0 \Leftrightarrow E_2^{LN} \cap O_H = \emptyset$
- 3) $\oplus (0, 1) = 1 \Leftrightarrow E_2^{LN} \cap O_H \neq \emptyset$
- 4) $\oplus (1, 0) = 1 \Leftrightarrow E_2^{LN} \cap O_H \neq \emptyset$

La relación de sentido general asociada con el robot y sus sensores se muestra en la tabla 4.4:

S_1	S_2	S_3	S_4	$S_1 \oplus S_2 \oplus S_3 \oplus S_4$
1	1	1	1	1
1	1	1	0	1
1	1	0	0	1
1	0	0	0	1
0	0	0	0	0
0	0	0	1	1
0	0	1	1	1
0	1	1	1	1
1	0	0	1	1
0	1	1	0	1
1	0	1	0	1
0	1	0	1	1
0	1	0	0	1
0	0	1	0	1
1	1	0	1	1
1	0	1	1	1

Tabla 4.4 Relación de sentido en R^{LN}



Nótese de la tabla 4.4 que las operaciones:

$$1) \oplus (0, 0, 0, 0) = 0 \Leftrightarrow R_0^{DS} \cap O_n = \emptyset$$

$$2) \oplus (1, 0, 0, 0) = 1 \Leftrightarrow R_0^{DS} \cap O_n \neq \emptyset$$

Es decir, cualesquier combinación donde el símbolo "1" esta asociado, entonces la operación $\oplus : A' \times A' \rightarrow A'$ identifica un evento de sensado.

4.7.1 Elementos participantes en el proceso de evasión

Sea $r_{0(O_n)}$ el radio de la circunferencia asociada con el obstáculo O_n . Considere las siguientes definiciones:

1. El robot R_0^{DS} es de dos grados de libertad del tipo rotacional-rotacional (RR), en el cual $l_1 > l_2$. Y los sensores están colocados en los extremos de los costados de los eslabones. Más específicamente a una distancia de $\frac{l_2}{7}$ de los extremos de los eslabones.

2. La tarea a realizar por parte del R_0^{DS} es la de ejecutar una línea recta desde un punto inicial P_i hasta un punto final P_f . Dicha tarea puede ser interrumpida pero siempre se debe iniciar donde fue el último punto donde se quedó el efector final.

3. O_n debe ser del tamaño $0 < r_{0(O_n)} \leq \frac{l_2}{7}$

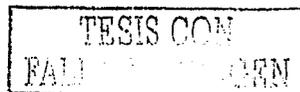
4.7.2 Los criterios de evasión

a) Localización del obstáculo

Criterio 1. Si O_n existe y si $O_n \subseteq R_e$, entonces O_n se considera evadible. Donde R_e es la región de evasión que esta formado por $R_{S_1 \rightarrow S_2}$ para el caso de la trayectoria de ida y por $R_{S_2 \rightarrow S_1}$ para el caso de la trayectoria de regreso.

b) Sensado móvil del obstáculo

Criterio 2. Cumplido el criterio 4, entonces ocurre el evento $R_0^{DS} \cap O_n \neq \emptyset$ durante la transición $T: {}^{DS}R_0^{SD} \rightarrow {}^{DS}R_0^D$ si y sólo si el evento del sensor $S_1 = 1$ o $S_2 = 1$. Y para el caso $T: {}^{DS}R_0^D \rightarrow {}^{DS}R_0^{SD}$ si y sólo si el evento del sensor $S_2 = 1$ o $S_1 = 1$.



Es importante señalar que toda existencia de obstáculo en la R_i debe de satisfacer las siguientes restricciones:

- 1) Si durante el recorrido o en la trayectoria de ida se presenta el evento $R_0^{DS} \cap O_H \neq \phi$, entonces los sensores S_1 y S_3 no deben de detectar obstáculo, puesto que el robot se alejará del obstáculo y una detección por dichos sensores implicará que el robot se detenga.
- 2) Si durante el recorrido de vuelta o trayectoria de vuelta se presenta el evento $R_0^{DS} \cap O_H \neq \phi$, entonces los sensores S_2 y S_4 no deben de detectar obstáculo pues el robot se alejará de dicho obstáculo.

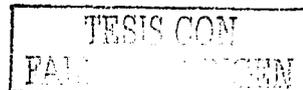
c) El proceso de evasión

Criterio 3. Siempre que se considere el problema de evasión desde el punto de vista de la reflexión, entonces el robot debe ir siguiendo la trayectoria codo abajo.

Criterio 4. Si R_0^{DS} al realizar su movimiento en la trayectoria de ida a través de la línea recta se presenta el evento $R_0^{DS} \cap O_H \neq \phi$, entonces R_0^{DS} debe de pararse y calcular la cinemática inversa en ese punto de la trayectoria τ a través de una reflexión o en términos prácticos codo arriba.

Criterio 5. Una vez calculada la cinemática inversa con el modelo de reflexión en ese punto de τ donde se presentó el evento $R_0^{DS} \cap O_H \neq \phi$, entonces se pueden aplicar los casos siguientes:

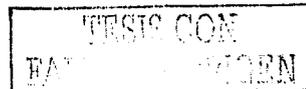
- a) Si es de ida, entonces R_0^{DS} debe regresar a "posición de inicio" $^{DS}R_0^{ND}$.*
- b) Si es de vuelta R_0^{DS} debe de regresar a la configuración $^{DS}R_0^P$.*



Criterio 6. Una vez que R_0^{DS} haya regresado a la posición ${}^{DS}R_0^{SD}$ o ${}^{DS}R_0^D$, entonces debe de calcular los complementos de los parámetros angulares (los parámetros angulares fueron calculados en el criterio 6), y se aplican de tal manera que los eslabones roten en sentido contrario hasta alcanzar el último punto donde se quedó R_0^{DS} en la trayectoria cuando encontró el obstáculo.

Criterio 7. Se puede seleccionar el tipo de movimiento que se desee que el robot aplique, los cuales son: uno a uno, simultáneos e híbridos. O en su defecto por "default" tomará el tipo de movimiento simultáneo.

Criterio 8. Si durante el proceso de evasión se lograra encontrar el mismo obstáculo O_n u otro, entonces R_0^{DS} realiza un paro total y en consecuencia el obstáculo no se pudo evadir.



Capítulo 5

Construcción, dispositivos electrónicos y programación del prototipo experimental de 2GDL para el problema de evasión de obstáculos en el plano

Introducción

El presente capítulo se divide en tres secciones las cuales son los siguientes: la construcción del manipulador de 2GDL experimental, en el cual se describen las medidas geométricas y tipo de material que se utiliza para cada uno de los elementos mecánicos que componen al manipulador. La segunda sección presenta la descripción de cada uno de los dispositivos electrónicos utilizados para el manejo del manipulador. En la tercera sección se da la descripción del algoritmo y la programación implementada en dicho manipulador para la modelación de los movimientos uno a uno, simultáneos e híbridos del robot de 2GDL y finalmente se genera el algoritmo y se realiza la programación de dicho robot para la evasión de obstáculos en el plano.

5.1 Sistema mecánico

Sobre la base de un proceso de selección en cuanto a la posible forma que puede tomar el robot, se llegó a la conclusión que la configuración adecuada del prototipo para evadir obstáculos en el plano es la mostrada en la figura 5.1.

Dicho prototipo experimental consta de: una base (B) la cual sirve para el soporte de la estructura del robot, un eslabón (E_1) y un segundo eslabón (E_2). Los eslabones E_1 y E_2 deben de contar con sensores en los costados llamados: S_1 , S_2 , S_3 y S_4 .

Después de caracterizar una forma determinada del prototipo (R_0), el siguiente paso es determinar los elementos que componen el robot, es decir, la forma de los eslabones, la forma de la base que soporta los eslabones, los actuadores, los elementos de transmisión de potencia entre otros elementos.



5.1.1 Descripción del robot de 2GDL para la evasión de obstáculos

El robot, como se ve en la figura 5.1, está compuesto de una "base fijadora", la cual está a la vez, compuesta por una base de madera (B_m) en la parte inferior, 7 postes y una base de acrílico (B_a) en la parte superior, que en su conjunto sirven para soportar los eslabones. También está formado por un primer eslabón (E_1) y un segundo eslabón (E_2), dos motores unipolares de cuatro fases : M_1 y M_2 , y cuatro ruedas dentadas (rd_1 , rd_2 , rd_3 y rd_4).

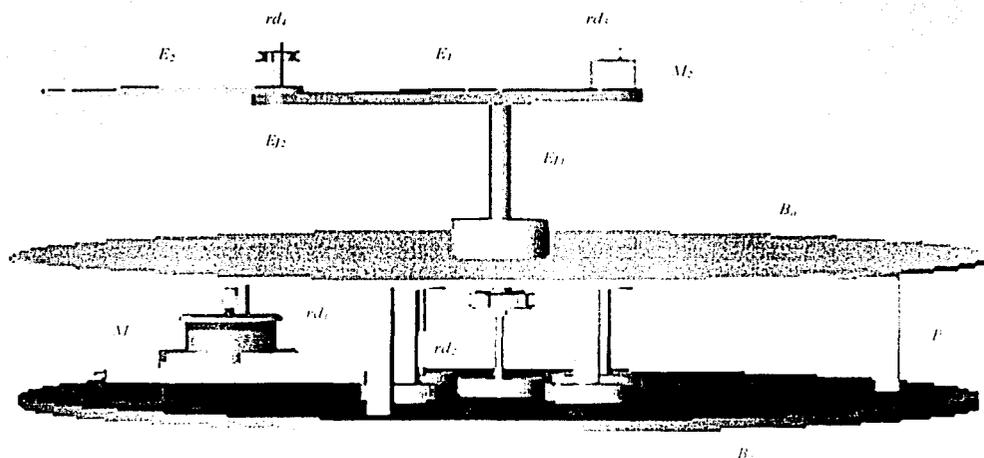
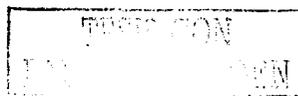


Figura 5.1 El robot de dos grados de libertad para la evasión de obstáculos en el plano.

Los elementos actuadores son dos motores de pasos (unipolares de cuatro fases), los cuales a través de dos ruedas dentadas que se encuentra uno de ellos en el eje del motor (M_1) y la otra en el Ej_1 y que son conectadas a través de una banda dentada que transmite la potencia hacia el eslabón E_1 . Para el caso del eslabón 2, la rd_3 se encuentra en el eje del motor 2. dicho motor se encuentra montado sobre E_1 , la rd_4 se encuentra en el Ej_2 y a través de otra banda dentada que se transmite la potencia y éste a su vez hace rotar al eslabón 2.



En la tabla 5.1 se presenta una lista de los elementos del manipulador y la función que realizan.

Número del elemento	Nombre del elemento	Función
1	Base	Soporte de la estructura del robot
2	Eslabón 1 (E_1)	Elemento de movimiento (rotar)
3	Eslabón 2 (E_2)	Elemento de movimiento (rotar)
6	Ruedas dentadas (rd_1, rd_2, rd_3 y rd_4)	Transmisión de potencia
7	Bandas dentadas (Bd_1 y Bd_2)	Transmisión de potencia
8	Ejes (Ej_1 y Ej_2)	Fijación de los eslabones y ruedas, así como transmisión de potencia
9	Baleros	Ajuste y rotación
10	Motores paso a paso (M_1 y M_2)	Generación de potencia
11	Otros (tornillos, tuercas, roldanas, etc)	Distintas funciones

Tabla 5.1 Principales elementos mecánicos y electromecánicos del robot de 2GDI.

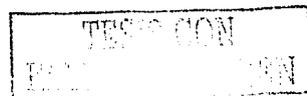
Cabe hacer mención que los dibujos de cada uno de los componentes se presentan en el apéndice A.

5.2 Los motores paso a paso y los dispositivos electrónicos del manipulador

En esta sección se describen los motores de paso a paso (unipolares de cuatro fases), las etapas de potencia para mover este tipo de motores, los optoreflectores utilizados para la posición de home, los sensores de barrera de luz roja con conexión de fibra óptica y los correspondientes circuitos acondicionadores de señal para dichos sensores.

a) Aspectos electromecánicos de los motores paso a paso

El eje del motor gira un ángulo determinado llamado ángulo de paso por cada pulso que recibe, el ángulo de paso es variable y puede ser de 0.9° , 1.8° , entre otros valores, esto es dependiendo del motor. Los pulsos pueden ser generados por un oscilador de frecuencia ajustable o de algún otro tipo.³ Para este caso se generará en el puerto paralelo externo de 8 bits de la computadora.



En la tabla 5.2 se describen algunas características importantes las cuales diferencian los motores de paso a paso y los motores de corriente directa.³

Características	Motores paso a paso	Motores de corriente directa
Mecánicas	Buen torque Baja velocidad Alta precisión Robustos	Bajo torque Alta velocidad Poca precisión Ligeros
Eléctricas	Alta cantidad de corriente de operación	Poca corriente de operación
Control	Complejo	Poco complejo
Costo	Alto	Bajo

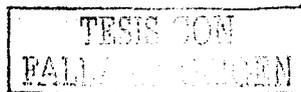
Tabla 5.2 Características mecánicas de los motores de paso y los de corriente directa.

b) Principios de operación de los motores paso a paso

El correcto funcionamiento de los motores paso a paso se logra mediante el cambio sucesivo de la corriente en las bobinas del estator. Para lo cual, se debe de realizar en una secuencia adecuada de tal manera que se obtenga un campo giratorio.^{3,6}

El campo fijo del rotor sigue el campo giratorio aparente según la ley de atracción y repulsión de los campos magnéticos, el sentido de giro del rotor está determinado por la secuencia de conmutación en las bobinas del estator. La velocidad de la secuencia de conmutación de las bobinas del motor determina la velocidad de giro del campo giratorio aparente y éste, a la vez, determina la velocidad de giro del rotor.^{3,6}

Para lograr un movimiento mucho más suave, los motores paso a paso se fabrican aumentando el número de polos del estator y se les debe de aumentar la cantidad de ranuras al estator. El grado de avance por paso es una de las características más importantes en este tipo de motores.



c) Tipos de motores paso a paso

Según su construcción, existen tres tipos de motores paso a paso:^{3,6}

1) De imán permanente

En este tipo de motor, su rotor es un imán permanente que posee una ranura en toda su longitud y el estator esta formado por una serie de bobinas enrolladas alrededor de un núcleo o polo. Su funcionamiento se basa en el principio de atracción y repulsión de polos magnéticos.

2) De reluctancia variable

En estos motores el rotor esta fabricado por un cilindro de hierro dentado y el estator esta formado por bobinas que crean los polos magnéticos. Como este tipo de motor no tiene un imán permanente, su rotor gira libremente cuando las bobinas no tienen corriente, lo que puede ser inconveniente en un momento dado si hay una carga que presione al eje. Este tipo de motor puede trabajar a mayor velocidad que el de imán permanente.

3) Híbridos

Estos motores combinan las características de los dos tipos de motores 1) y 2), y así logran un alto rendimiento a buena velocidad. En cuanto a la forma de conexión y excitación de las bobinas del estator, estos motores se dividen en dos tipos:

Unipolares. Hay dos tipos de bobinas y tienen toma media, es decir, tienen seis terminales.

Bipolares. Estos tienen dos bobinas al igual que los unipolares, con la diferencia de que no poseen toma media, es decir, poseen cuatro terminales.

TESIS CON
EVALUACIÓN

ESTA TESIS HA SIDO
DE LA BIBLIOTECA

d) Modos de operación

Los motores paso a paso tanto unipolares como bipolares, pueden trabajar en dos modos de operación: 1) de paso completo y 2) de paso medio. En el primer caso, con cada secuencia el rotor gira un determinado ángulo que depende de la fabricación del motor. En el momento de medio paso, cada secuencia produce un giro en grados, correspondiente a la mitad de su paso normal. Además, en el sentido de giro, cuya posición queda determinada por la secuencia de los pulsos transmitidos, variando la secuencia de los pulsos aplicados a las bobinas, también puede controlar la velocidad de los motores PAP, dentro de cierto rango.^{3,16}

e) Embobinados de motores paso a paso

Los niveles de tensión, que deben estar presentes en cada terminal, dependen de las secuencias de cada motor. Para encontrar la secuencia es necesario aplicar tensión en las distintas bobinas y deducir dicha secuencia o bien midiendo la resistencia entre las terminales.^{3,16}

Para realizar el control de velocidad del motor, se debe variar la frecuencia de desplazamiento de la secuencia, de forma tal que si aumenta esta frecuencia aumentará la velocidad de giro del motor. Si por el contrario la frecuencia disminuye, la velocidad del motor también lo hace. Para realizar la inversión del giro lo que se debe de hacer es invertir la secuencia de envío de pulsos. Y para realizar el control de posición del motor, se debe de conocer la cantidad de pulsos enviados y los grados que avanza el motor por paso^{3,6}.

f) Etapa de potencia de los motores a pasos

Los motores de paso a paso para su funcionamiento se requiere de una etapa de potencia. Es por ello que en la figura 5.2 se presenta la etapa de potencia utilizada en este trabajo, para realizar dicho accionamiento a los motores de pasos unipolares de cuatro fases.¹⁵

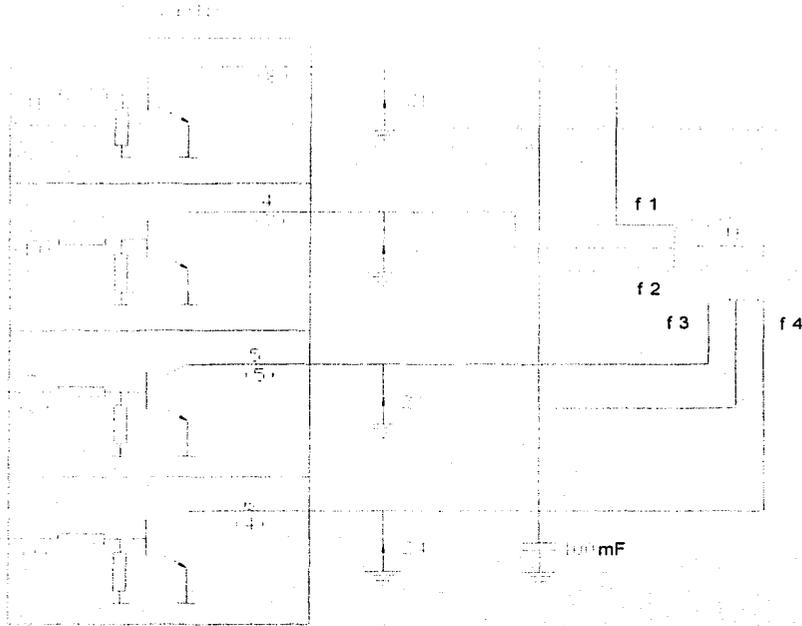


Figura 5.2. Etapa de potencia del motor paso a paso unipolar de cuatro fases.¹⁵

Por otro lado, en la figura 5.3 se muestra en detalle el circuito integrado L702B(N) para su conexión.

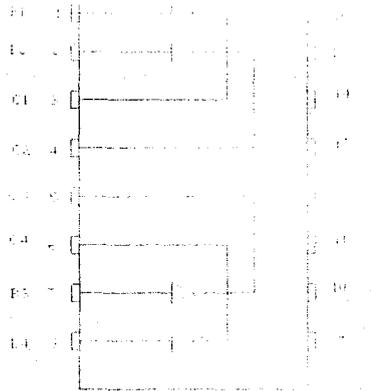


Figura 5.3 Conexión del circuito integrado L702B(N).¹⁵

5.2.2 La computadora personal

Una computadora, como sistema, consiste de tres subsistemas: procesador, memoria y entradas y salidas (figura 5.4).⁴

a) **El procesador** (o unidad central de proceso, CPU) es el corazón de la computadora. Un simple procesador contiene el circuito de control para buscar y ejecutar instrucciones, una unidad lógica aritmética para manipulación de datos y registros para almacenamiento de pequeñas cantidades de datos. Éste también tiene un circuito interface para el control y la comunicación con los subsistemas de la memoria y entradas/salidas.

b) **La memoria** (o memoria principal) de una computadora contiene almacenada instrucciones y datos. La memoria de una computadora tiene algunos números de localizaciones cada una de las cuales almacena una cantidad **b** de **bits**.

c) **El subsistema de entradas y salidas** contiene dispositivos periféricos para comunicarse con el observador y controlando el mundo fuera de la computadora. Los dispositivos

periféricos incluye terminales, impresoras, dispositivos de comunicación, sensores mecánicos y actuadores.

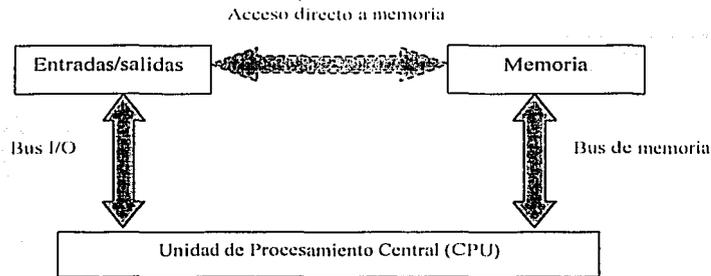


Figura 5.4 Diagrama de bloques de una computadora típica.⁴

5.2.2.1 El puerto paralelo

Una de las partes principales de una computadora (para poder automatizar un sistema) es el puerto. Y en el manejo de las PC's, el manejo de los puertos desde Windows. Los puertos pueden clasificarse de acuerdo con el tipo de acceso físico que permiten, y son: puertos externos y puertos internos. Entre los puertos externos más comunes en las computadoras se encuentra el puerto paralelo, el puerto serial y puerto para juegos. Entre los puertos internos, se pueden mencionar los que corresponden a las ranuras de expansión o *slots* de la tarjeta principal.

Las tarjetas de interface como la de video, la de modem interno y la de sonido, hacen uso permanente de este tipo de puerto ya que ellas deben leer y escribir sobre el bus de datos de la computadora para poder cumplir sus funciones. A estas tarjetas también se les puede leer y enviar información de la misma manera que con los puertos externos, solo que en este caso, se necesita conocer las direcciones que ellas utilizan, ya que en una misma ranura de expansión se tiene acceso a todas las direcciones de puertos, lo que obliga a que la tarjeta de interface sea quien especifique la dirección que este utilizando.¹⁹

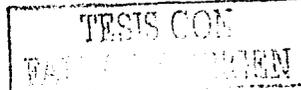
a) Los tres buses del puerto paralelo

Cualquier puerto paralelo está compuesto de tres buses que transportan información de entrada, información de salida o de entrada y salida, todos mediante una serie de pines del conector ubicado generalmente en la parte posterior de las computadoras. Estos buses han sido denominados, de acuerdo con la labor que realizan, como **bus de datos, bus de estado y bus de control**.^{4,19}

a1) Bus de datos. En este bus se transporta la información de datos propiamente dichos y está compuesto por 8 pines externos (1 byte completo), estos son del pin 2 al pin 9. Por ejemplo, si el dispositivo conectado es una impresora, mediante este bus la computadora envía los caracteres que se desean imprimir en el documento. Así mismo, si el documento es un gráfico, este bus transporta la información de pixeles o puntos que se deben marcar sobre la hoja con el fin de formar la imagen deseada.

a2) Bus de estado. Este bus es el encargado de transportar información desde la impresora hasta la computadora. Como su nombre la indica, la información hace referencia a los estados de actividad de la impresora, como son la ausencia de papel, la posición de la cabeza de impresión, puesta en pausa, entre otras tareas. Aunque la computadora utiliza una dirección específica para leer la información, externamente sólo se tiene acceso a los 5 últimos bits (bit 3 a bit 7), los cuales corresponden a los pines del 10 al 13 y el pin 15. La información de estado varía de acuerdo con el tipo de impresora, por lo que el programa controlador que se haya instalado al sistema operativo es quien reconoce y decodifica los bits recibidos.

a3) Bus de control. El bus de control es el encargado del manejo o control del funcionamiento de la impresora. Por medio de este es posible detener la impresión, limpiar o purgar un documento pendiente que ya se encuentre en la memoria de la impresora, etc. Al igual que los demás buses, la computadora utiliza una dirección específica para dirigirse a este bus pero externamente, es decir, que el conector, solamente dispone de cuatro pines (1, 14, 16 y 17) que corresponden a los primeros bits de tal dirección (bit 0 al bit 3).



b) Los datos técnicos del puerto paralelo son:

Tipo de transmisión: Paralelo (Bloques de 8 bits)

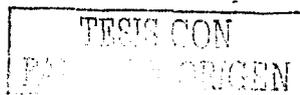
Niveles de señales: Bajo 0.0 a +0.4 V.

Alto +2.4 a +5.0 V.

En la tabla 5.3 se describen cada uno de los pines de los puertos, así como también se esquematiza en la figura 5.5.

No. de pin	Señal	Entrada	Salida	Descripción
1	STROBE		X	Es la señal que valora los datos y sincroniza la toma de señales
2-9	D1-D8		X	Salida de datos D1...D8
10	ACKNLG	X		Esta señal es enviada por la impresora para la confirmación de la recepción de datos y señala al ordenador la disposición de la impresora para la próxima recepción de datos
11	BUSY	X		Esta señal indica si la impresora esta lista para la recepción de datos. Cuando Error = 0, Slct = 0 la impresora no esta en línea
12	PE	X		Es la señal de final de papel
13	SLT	X		Slct = 1, cuando la impresora esta en línea
14	AUTOFD		X	AUTOFD
15	ERROR	X		ERROR
16	INIT		X	INIT
17	SLCT		X	SLCT
18-25	GND		X	Tierra

Tabla 5.3 Descripción de las señales del puerto paralelo.¹⁹



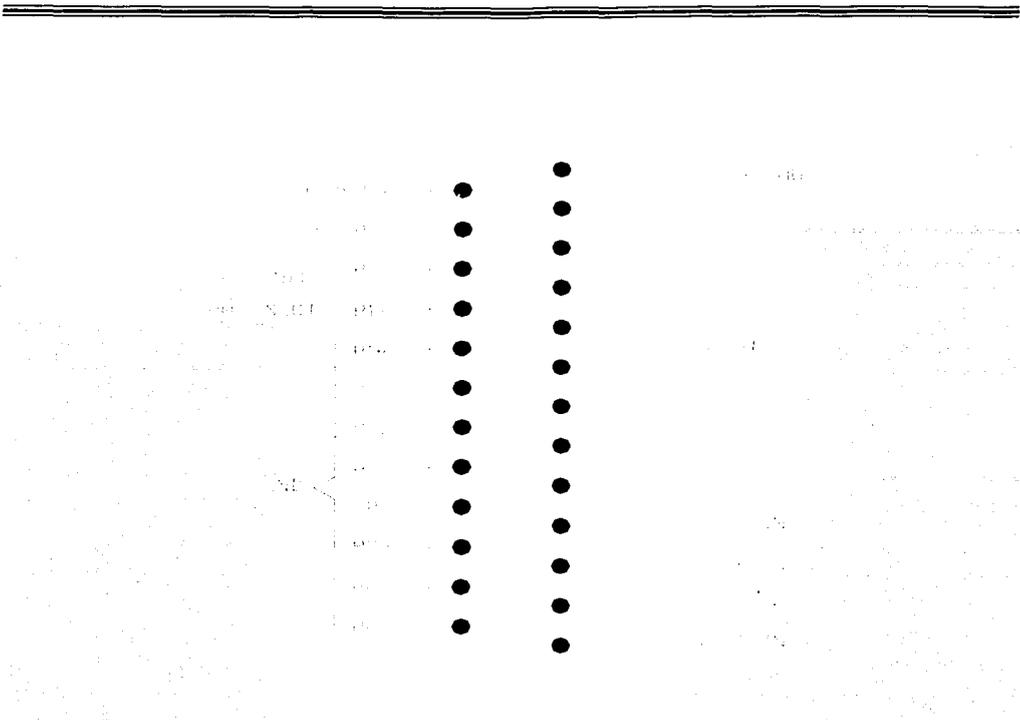


Figura 5.5 Diagrama que muestra la función de cada uno de los pines del puerto paralelo externo.¹⁹

5.2.3 Los sensores

El término *sensor*³ es usado para nombrar a un dispositivo el cual produce una señal relacionada en cuanto a la cantidad que esta midiendo. Así, en el caso de la temperatura es usada una resistencia eléctrica, la cantidad medida es temperatura y el sensor transforma la entrada de temperatura en un cambio en resistencia.

El término *transductor* es a menudo usado en lugar del término sensor. Los transductores son definidos como elementos que están sujetos a un cambio físico. Así, los sensores son transductores. sin embargo, en un sistema medido se puede usar transductores (en adición al sensor) en otras partes del sistema para convertir una señal de una forma a otra forma.

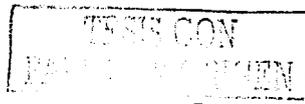
a) Los tipos de sensores

Existen diferentes tipos de sensores, entre los más importantes se tienen: a) sensores de proximidad: inductivos, sensores de efecto "hall", capacitivos, ultrasónicos, de proximidad óptica; b) sensores de tacto: binarios, análogos; c) sensores de fuerza y torque.^{3,6}

a1) Los sensores de proximidad tienen una salida binaria la cual indica la presencia de un objeto en una distancia específica. Estos sensores se subdividen en **inductivos**, que se basan en el cambio de inductancia en la presencia de un objeto metálico. Y los **sensores de efecto "hall"** relacionan el voltaje entre dos puntos en un material conductor o semiconductor en el cual el campo magnético cruza a dicho cuerpo ferromagnético. Este tipo de sensores únicamente pueden detectar objetos magnéticos, se basan en el principio de fuerza de Lorentz el cual actúa sobre el recorrido de una partícula cargada a través de un campo magnético. En lo referente a los **sensores capacitivos** estos son potencialmente capaces (con diferentes grados de sensibilidad) de detectar todo tipo de sólidos y líquidos, están basados en la detección del cambio de capacitancia inducida de la superficie a el elemento sensante. El elemento sensante es un capacitor compuesto de un electrodo sensitivo y un electrodo de referencia, que puede ser por ejemplo de un disco y anillo separados por un material dieléctrico, y la parte restante es el circuito electrónico que sirve para detectar la proximidad basado en el cambio de capacitancia.^{3,6}

En los **sensores ultrasónicos**, el elemento básico de este tipo de sensores es un transductor electroacústico, que es un tipo de material cerámico piezoeléctrico. Los sensores de proximidad ópticos son semejantes a los sensores ultrasónicos en el sentido de que ellos detectan de manera próxima a un objeto por su influencia en viaje de las ondas de propagación del emisor al receptor. El emisor consiste en un diodo emisor de luz de estado sólido, el cual actúa como un transmisor de luz infrarroja y un fotodiodo el cual actúa como un receptor.^{3,6}

a2) Los sensores de tacto son usados en robótica para obtener información asociada con el contacto entre la mano (efector final) del manipulador y el objeto en el espacio de



trabajo. Los cuales pueden ser usados para la localización, reconocimiento y para el control de la fuerza ejercida por el manipulador sobre un objeto dado. Estos sensores se subdividen en binarios y análogos. Los **binarios** son básicamente "interruptores" que responden a la ausencia o presencia de un objeto. Los **sensores análogos** son dispositivos cuya salida es proporcional a una fuerza local.⁶

a3) Los sensores de fuerza y torque. Son usados principalmente para medir las fuerzas de reacción desarrolladas en la interface en el ensamble mecánico. Y se utilizan principalmente en las juntas y en las muñecas. Un sensor de fuerza en las juntas mide principalmente las componentes cartesianas de fuerza y torque que actúan sobre las juntas del robot. Para manejar una junta por un motor de corriente directa., el sensado es hecho midiendo la corriente de armadura.⁶

b) Terminología desarrollada

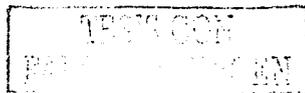
En los sensores se manejan ciertas características importantes para definir el desempeño de éstos. Los cuales son los siguientes:

Rango. El rango de un transductor es el límite que la entrada puede variar. Así, por ejemplo: una celda de carga para la medida de la fuerza puede tener un rango de 0 a 50 kN.

El error es la diferencia entre el resultado de la medida y el valor verdadero de la cantidad que esta siendo medida.

$$\text{Error} = \text{valor medido} - \text{valor verdadero.}$$

La variación es la extensión en la cual el valor indicado por sistema de medida puede ser erróneo. Este es la suma de todos los posibles errores que pueden ocurrir como la variación en la cual los transductores han sido calibrados. Un instrumento de medida de temperatura puede, por ejemplo, especificarse que tiene una variación de $\pm 2^{\circ}\text{C}$. Esto puede significar que la lectura dada por el instrumento puede tener una variación de $+ 0 - 2^{\circ}\text{C}$ de su valor verdadero. La variación es a menudo expresada como un porcentaje de deflexión total escalar que resulta cuando la salida del sistema de medida fue desplegado también exclusivamente en una escala circular o lineal. Un sensor, por ejemplo, puede especificarse

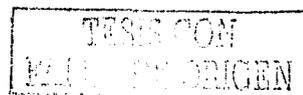


que tiene una variación de $\pm 5\%$ de el rango total de salida. Así, si el rango de el sensor fue de 0 a 200°C . entonces la lectura se puede ser esperada con + o -10°C de el valor de lectura. **La sensibilidad** es la relación que indica la cantidad de la salida por unidad que entrada que se tiene. Por ejemplo, un termómetro de resistencia puede tener una sensibilidad de $0.5 \Omega/^{\circ}\text{C}$. Este término es frecuentemente usado para indicar la sensibilidad para la entrada que esta siendo medida, i.e cambio de desarrollo. De esta manera puede ser la sensibilidad del transductor a cambios de temperatura en desarrollo o quizás de las fluctuaciones en la suministración de voltaje. **En el error de histéresis**, los transductores pueden dar diferentes salidas de algún valor de la cantidad que esta siendo medida de acuerdo con el valor de alcance que tiene éste por un cambio de incremento continuo o un cambio de decremento continuo. Este efecto es llamado "histéresis". El error de histéresis es la máxima diferencia en salida para un incremento y decremento del valor.^{3.6.17}

Repetibilidad. La repetibilidad de un transductor es su capacidad para dar la misma salida para aplicaciones repetidas del mismo valor de entrada. Un transductor para la medida de la velocidad angular típicamente puede dar una repetibilidad de $\pm 0.01\%$ de el rango total en una velocidad angular particular. **Reproducibilidad.** La reproducibilidad de un transductor es su capacidad para dar la misma salida cuando se usa para medir una entrada constante y es medida sobre un número de veces. **Estabilidad.** La estabilidad del transductor es su capacidad para dar la misma salida cuando se usa para medir una entrada constante sobre un periodo de tiempo.^{3.6.17}

5.2.3.1 Sensores utilizados en el proyecto

a) **Sensor de posición.** El sensor de posición utilizado en el robot estudiado en este trabajo de tesis, es un optorelector del tipo HA21A en el cual dos terminales van conectadas a 5 Vcc. La terminal "E" está conectada a una resistencia de 470 Ohms a $\frac{1}{2}$ W y ésta a tierra. La terminal "D" se conecta a una resistencia de 1000 Ohms a $\frac{1}{2}$ W y, posteriormente, a tierra. Entre la resistencia y la terminal "D" sale una conexión la cual va a una terminal del circuito integrado (SN74LS04N). Dicha conexión se muestra en la figura 5.6.



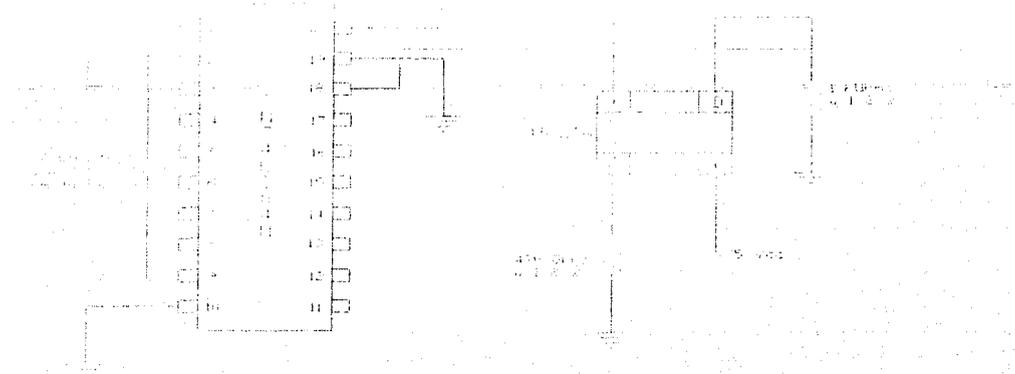


Figura 5.6 Diagrama electrónico de conexión del sensor HA21A

b) Sensores de barrera con luz roja. El robot de 2GDL debe de ser capaz de evadir obstáculos en el plano utilizando secuencias de movimientos de manera determinística, para ello a dicho robot se le equipó con 2 sensores de barrera de luz roja (SOEG-L-Q30-PA-K-2L, 165 326) con conexión de fibra óptica (SOEZ-LLK-SE-2, 0-M4, 165 360), estos se colocaron en los costados de los eslabones E_1 , cuya función es la de detectar objetos que pudiesen colisionar con los eslabones del robot. Para poder recibir la señal en la computadora se construyó un acondicionador de señal por cada sensor anteriormente mencionado.

c) Sensores de contacto binarios. Estos están colocados en los costados del segundo eslabón. el objetivo es también detectar objetos al momento que el obstáculo choque con los dos alambres. El diagrama electrónico de conexión es semejante al presentado en la figura 5.6

TESIS CON
FABRICA DE ORIGEN

En la tabla 5.4 se presentan las características del sensor de barrera de luz roja (SOEG-L-Q30-PA-K-2L, 165 326).⁷

Nombre del dato técnico	Descripción del dato técnico
Tensión de funcionamiento	10...30 Vcc (=Ub)
Frecuencia residual adm.	Máx ± 20% Ub
Corriente sin carga	<25 mA (incluye LED)
Corriente de conmutación adm.	Máximo 200 Ma
Caída de tensión	<2 V
Tiempo de respuesta	0.5 ms
Tiempo de caída	0.5 ms
Anticortocircuitaje	Si
Conexión inconfundible	Si
Capacidad máxima conmutable	250 nF
Temperatura de ambiente	-5 °C...55 °C
Temperatura de almacenamiento	-40 °C...85 °C
Tipo de protección	IP 65
Par de apriete máximo	1.5 Nm
Material del cuerpo	PBTP
Material del cable	Revestimiento exterior

Tabla 5.4 Descripción de los datos técnicos del sensor (SOEG-L-Q30-PA-K-2L, 165 326).⁷

b.1) Fibra óptica de polímero (SOEZ-LLK-SE-2, 0-M4, 165 360)

En lo referente al uso de la fibra óptica, se dan las siguientes recomendaciones:⁸

- a) Determinar la longitud necesaria del conductor.
- b) Cortar únicamente con la herramienta especial tipo SOE-LSK. Usar cada segmento cortado una sola vez (en caso contrario puede haber pérdidas de luz).
- c) Para enchufar el conductor, introducirlo hasta el tope en el conector.
- d) Ajustar el tornillo de fijación.
- e) Evitar fuerzas de tracción o torsión.

TESIS CON
FALLA DE ORIGEN

En la tabla 5.5 se describen los datos técnicos de la fibra óptica de polímero.⁸

Nombre del dato técnico	Descripción del dato técnico
Distancia máxima de detección	40...400 mm
Distancia mínima de detección	20...100 mm
Temperatura ambiente	-40...70 °C
Temperatura de almacenamiento	-40...70 °C
Tipo de protección	IP 65
Par de apriete máximo	1 Nm
Material de la fibra óptica	PMMA
Revestimiento exterior	Polietileno

Tabla 5.5. Descripción de la fibra óptica de polímero (SOEZ-LLK-SE-2, 0-M4, 165 360).⁸

La forma de conexión eléctrica del sensor descrito en la tabla 5.4 se muestra en la figura siguiente:

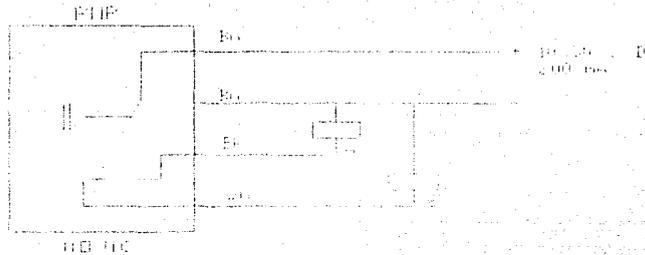


Figura 5.7 Conexión eléctrica del sensor de barrera de luz roja (SOEG-L-Q30-PA-K-2L, 165 326).⁸

Donde: BrownBn = café; Bu =Blue(azul); Bk =Black(negro); Wh =White(blanco).

N.O =Normalmente abierto; N.C =Normalmente cerrado.

La forma geométrica de dicho sensor es la siguiente:

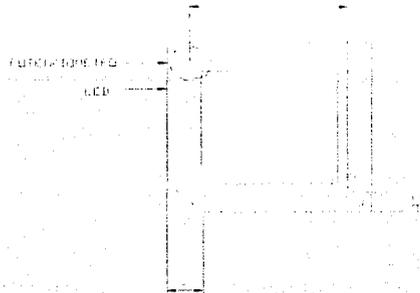


Figura 5.8 Forma geométrica del sensor de barrera de luz roja (SOEG-L-Q30-PA-K-2L, 165 326).⁷

En la tabla 5.4 se observa que la tensión de funcionamiento del sensor tiene un rango de funcionamiento de 10 a 30 V, que en consecuencia las conexiones de salida del sensor es aproximadamente de ese voltaje. Entonces es necesario acondicionar dicha señal para que pueda ser recibida por el puerto paralelo externo de la computadora, para lo cual se utiliza el circuito mostrado en la figura 5.9.

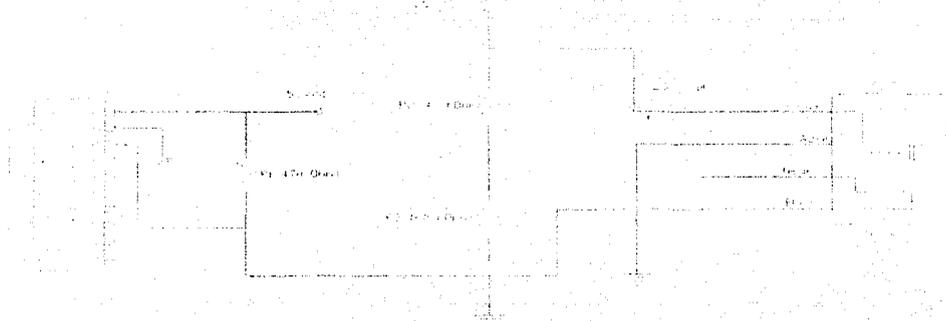


Figura 5.9. Diagrama electrónico del acondicionador de señal del sensor de barrera de luz roja (SOEG-L-Q30-PA-K-2L, 165 326).

5.3 El programa de computación

En el diseño de programas es necesario utilizar metodologías bien elaboradas de tal manera que los resultados obtenidos sean correctos. En este trabajo de tesis se desarrolla el programa de computación de manera metodológica como se muestra esquemáticamente en la figura 5.10.^{10,18}

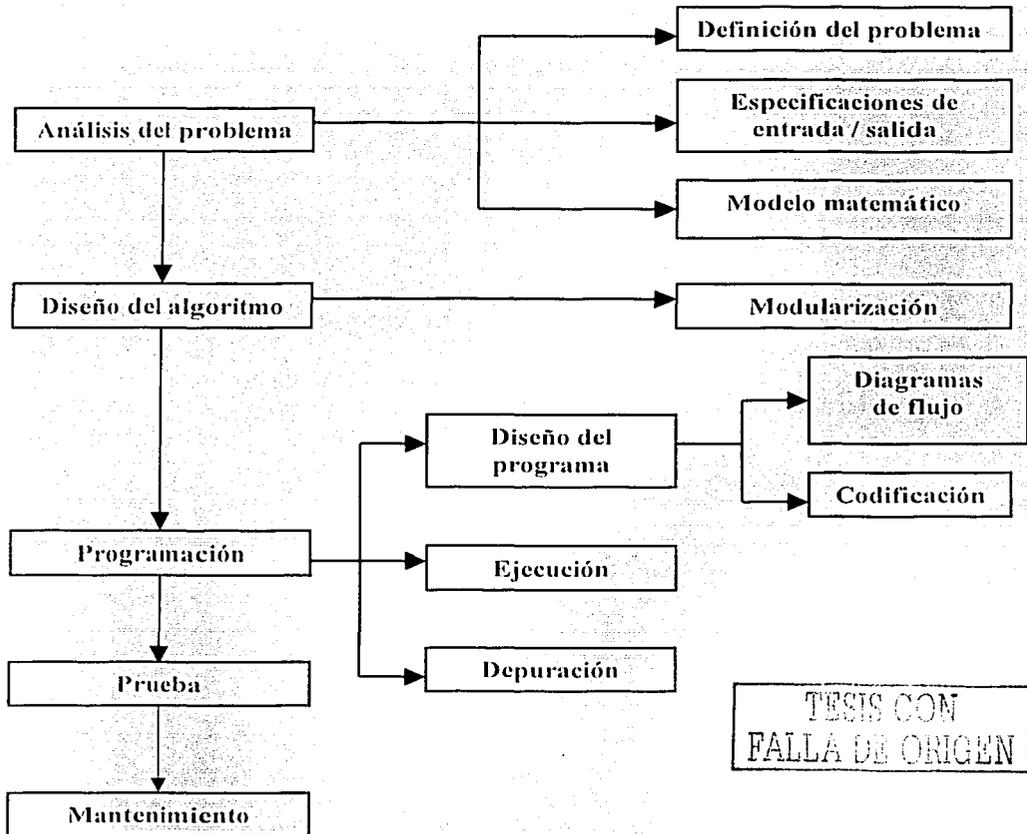


Figura 5.10 Metodología para desarrollar el programa de computación.

En los siguientes puntos se describe cada uno de los pasos de la metodología que se llevó a cabo para elaboración del programa de computación.

5.3.1 Análisis del problema

Esta sección está compuesta de tres partes las cuales son: “definición del problema” desde el punto de vista computacional que es necesario resolver, también se especifican las entradas y salidas que son necesarias manejar. Y finalmente el modelo matemático utilizado.

5.3.1.1 Definición del problema

Este programa es diseñado para cumplir funciones específicas como la de crear un enlace entre la computadora y varios elementos actuadores y sensores que están conectados a ella, para que el robot de 2GDL sea capaz de describir diferentes tipos de movimientos. El problema que se plantea es que el robot debe ser capaz de describir movimientos, que son los movimientos uno a uno, movimientos simultáneos e híbridos y movimientos simultáneos que siguen una ruta (recta), así como la utilización de estos tipos de movimientos para el problema de la evasión de obstáculos en el plano.

5.3.1.2 Especificaciones de entrada/salida

En primer lugar se describen las entradas y salidas por el puerto paralelo dada la importancia que tiene en el desarrollo de este trabajo de tesis y enseguida se describen las entradas y salidas en el cual el usuario interactúa con el programa.

Para saber si el robot se encuentra en posición de inicio, es necesario utilizar dos optorelectores que detecten dicha posición de inicio, el cual muestra que el robot se encuentra listo para describir el movimiento que se seleccione. También se utilizan cuatro sensores para detectar obstáculos en los costados de los eslabones, es decir, en total se necesitan seis entradas hacia la computadora por el puerto paralelo externo. Por otro lado, se requieren cuatro salidas por dicho puerto para los dos motores de pasos.

Las entradas que el programa debe de permitir en pantalla son distintas dependiendo del tipo de subprograma que se seleccione. Esto es para el caso del subprograma que contiene a los movimientos uno a uno se debe de introducir los ángulos α_1 y α_2 que deben de rotar cada uno de los eslabones, el orden en que estos se deben de mover; y las salidas que

TESIS CON
FALLA DE ORIGEN

deben mostrarse en pantalla es el ángulo actual rotado por cada uno de los eslabones y si el robot se encuentra en posición de inicio o final. En los movimientos simultáneos las entradas son el ángulo que deben rotar cada uno de los eslabones y como salidas el ángulo actual rotado por cada uno de los eslabones y el estado en que se encuentra el robot. En los movimientos simultáneos que siguen una ruta deben de permitir los puntos de inicio y final de la recta por describir, el número de particiones de la recta y como salidas el valor de los ángulos y el estado en que se encuentra el robot. Y finalmente para el caso de la evasión de obstáculos deben de mostrarse la secuencia de evasión que se esta utilizando para dicho efecto, el ángulo girado por cada eslabón y finalmente el estado en que se encuentra el robot.

5.3.1.3 Modelo matemático

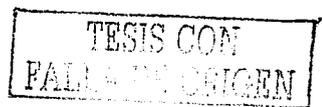
La estructura del programa EvaSoft 1.0, el cual esta constituido principalmente de cuatro subprogramas para describir los tipos de movimientos para el robot de 2GDL y el problema de evasión de obstáculos. En cada uno de estos subprogramas se utilizan ecuaciones de posición para la cinemática directa e inversa para determinar el ángulo de giro de cada uno de los eslabones o bien para hacer rotar los eslabones del robot; para observar los modelos matemáticos en los que se basa este trabajo es necesario remitirse al capítulo 2 .

5.3.2 Diseño del algoritmo

Cuando se hace referencia al diseño del algoritmo prácticamente se refiere a la modularización del problema planteado, es decir, en la división de éste en pequeños programas.

5.3.2.1 Modularización

La modularización es una etapa para la solución de un problema complejo, en donde el problema se divide en elementos de menor tamaño llamados bloques o módulos, que para este caso, se divide en cuatro: movimientos uno a uno, movimientos simultáneos, movimientos híbridos, movimientos simultáneos que siguen una ruta y el problema de evasión de obstáculos. Dicha modularización se muestra en la figura 5.11.



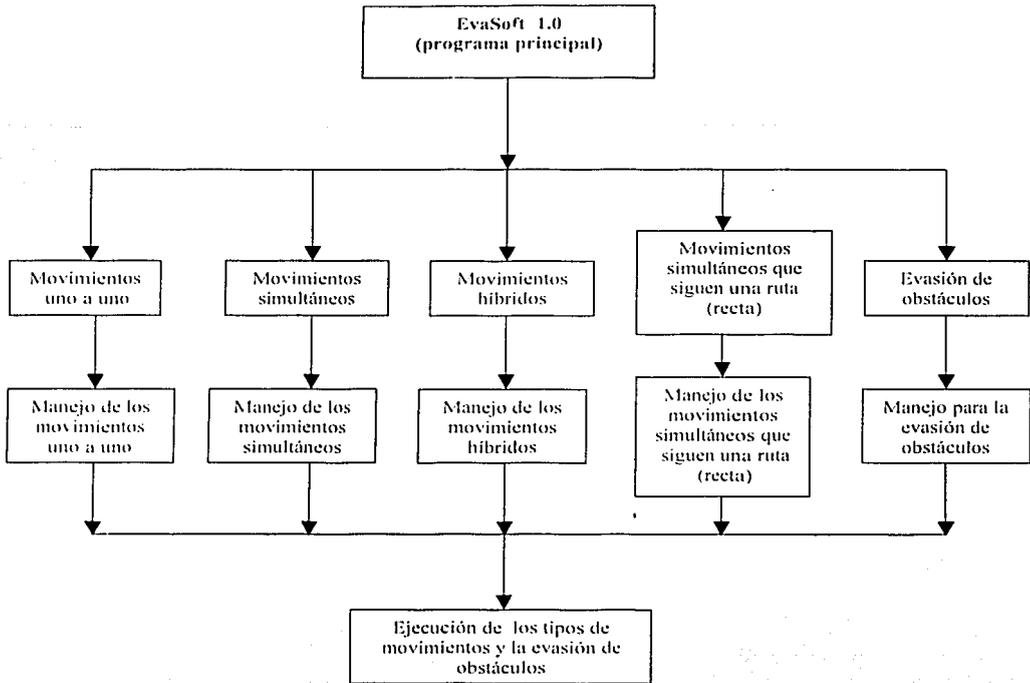


Figura 5.11 Esquema general del programa EvaSoft 1.0.

5.3.3 Programación

La programación está dividida en tres partes las cuales son: el diseño del programa, ejecución y depuración. En los cuales se hace uso de todos los puntos definidos con anterioridad, se considera a ésta, una parte muy importante en este trabajo.

TESIS CON
FALLA DE ORIGEN

5.3.3.1 Diseño y codificación del programa

a) Diagramas de flujo para poner al robot en posición de inicio

El diagrama de flujo de la figura 5.12 muestra el orden en que se mueven los eslabones para ponerse en posición de inicio. Este diagrama de flujo consta en primer lugar del procedimiento llamado “mover_motor_1” el cual hace mover al motor uno y en consecuencia al eslabón 1, el cual es detallado en la figura 5.13. Por cada paso que da el motor se recibe un dato del puerto y se pregunta si es igual a 247 porque es pin al que esta conectado el sensor y corresponde a ese valor su activación. Terminado con el primer eslabón se continua con el segundo de la misma manera y de esta forma termina el proceso de posición de inicio.

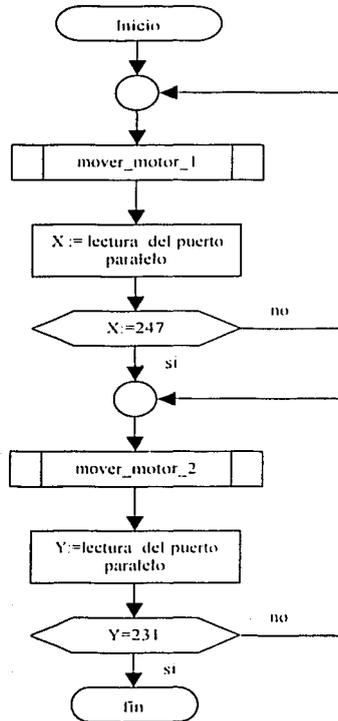


Figura 5.12 Diagrama de flujo para poner al robot en estado de inicio.

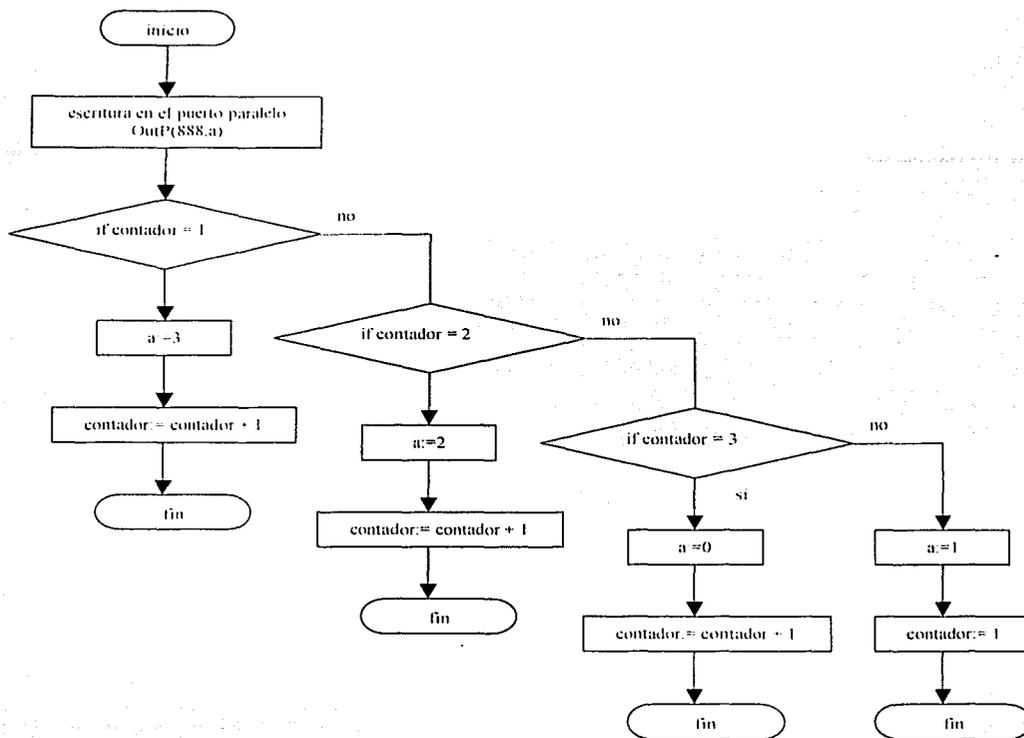


Figura 5.13 Diagrama de flujo del procedimiento Mover_motor_1 utilizado en la figura 5.13.

TESIS CON
FALLA

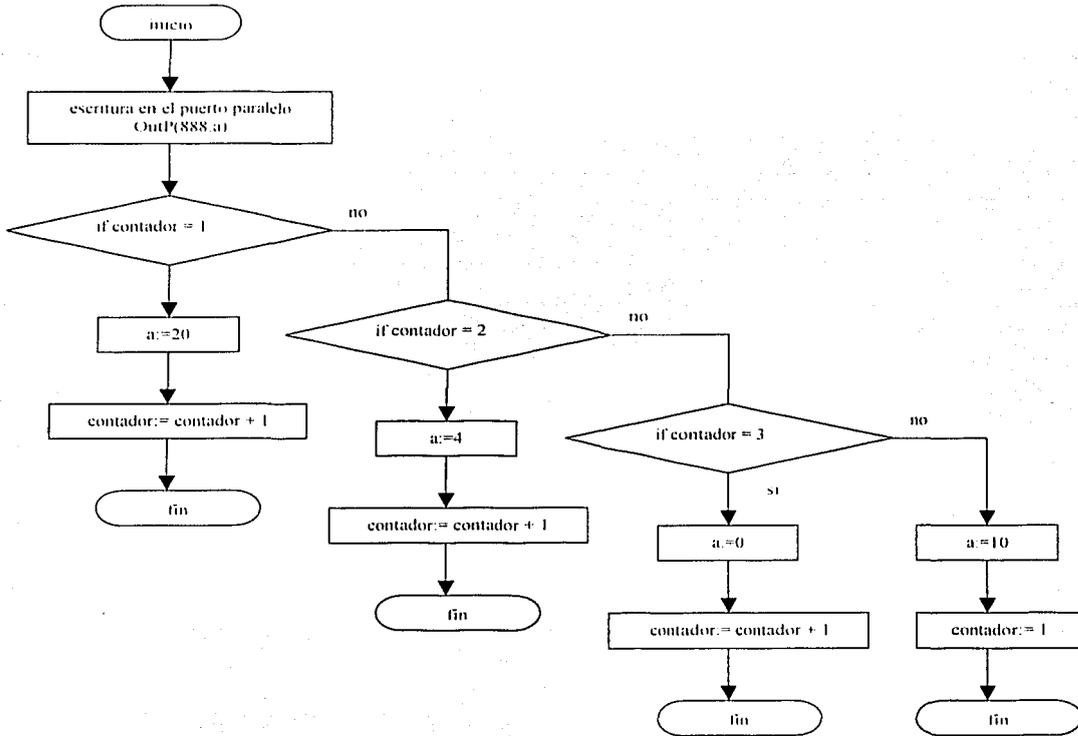


Figura 5.14 Diagrama de flujo del procedimiento mover_motor_2 utilizado en la figura 5.13.

TESIS CON
FALLA EN EL ORIGEN

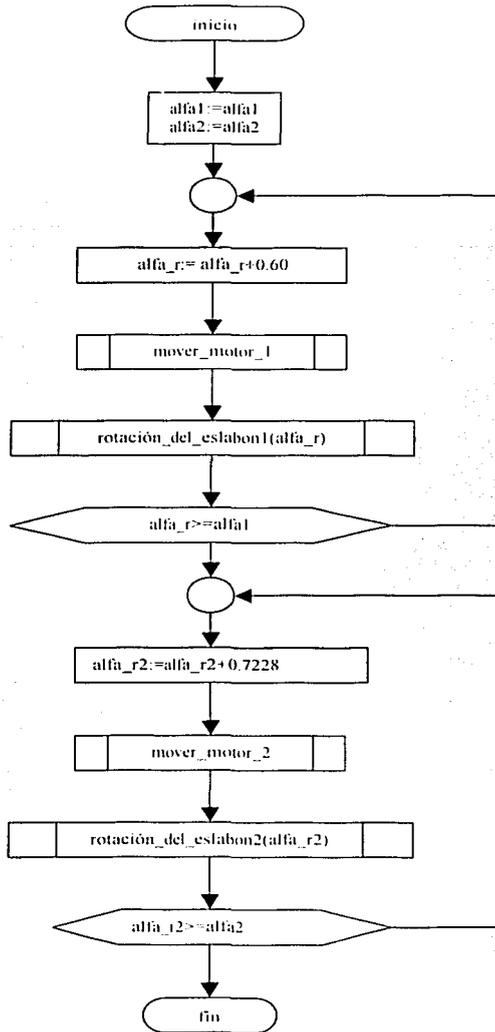


Figura 5.15 Diagrama de flujo para los movimientos uno a uno de ejes.

TESIS CON
FALLA DE TIPO TIEN

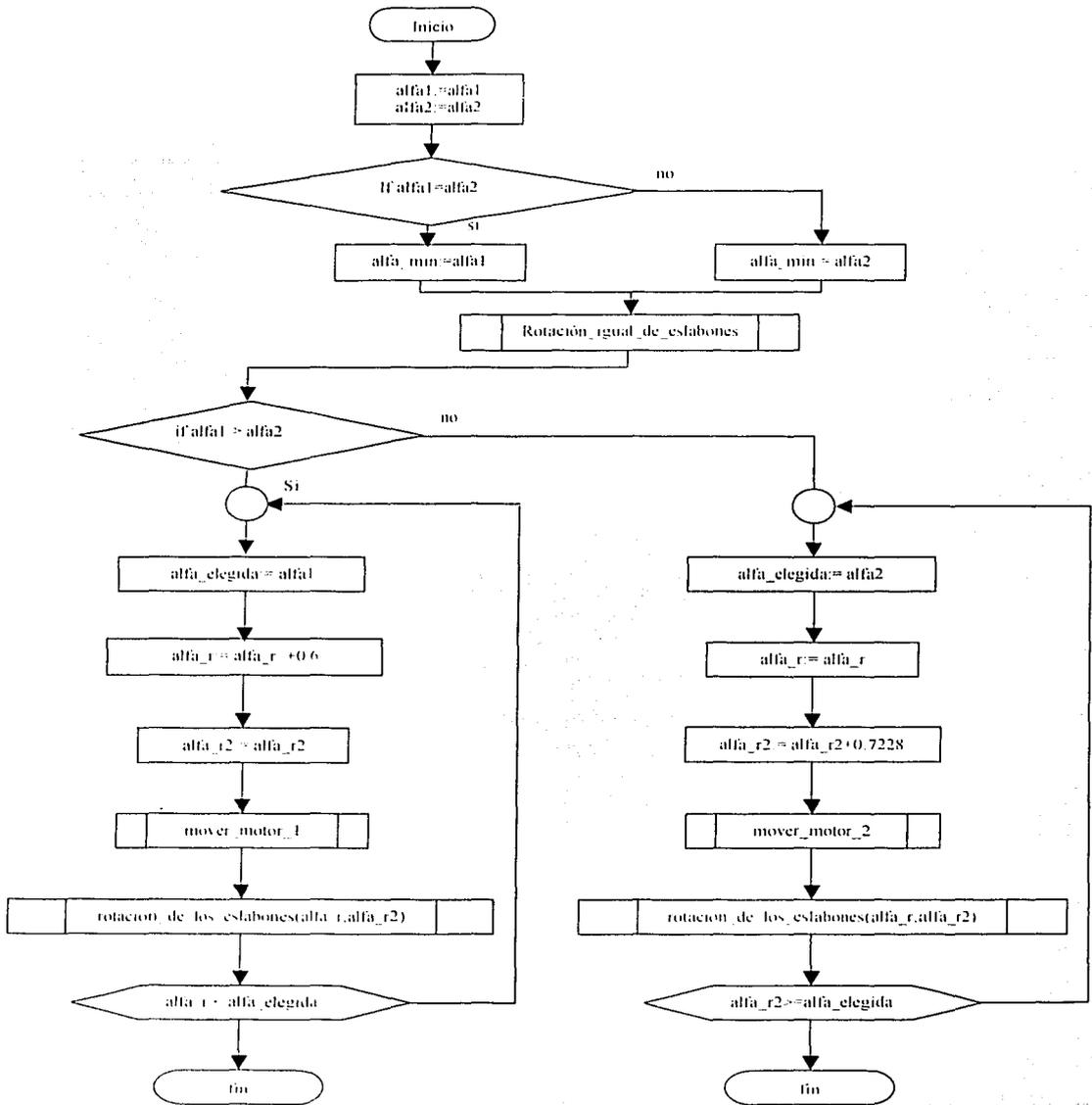
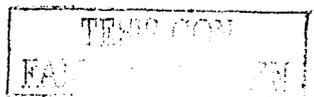


Figura 5.16 Diagrama de flujo para los movimientos híbridos de ejes.



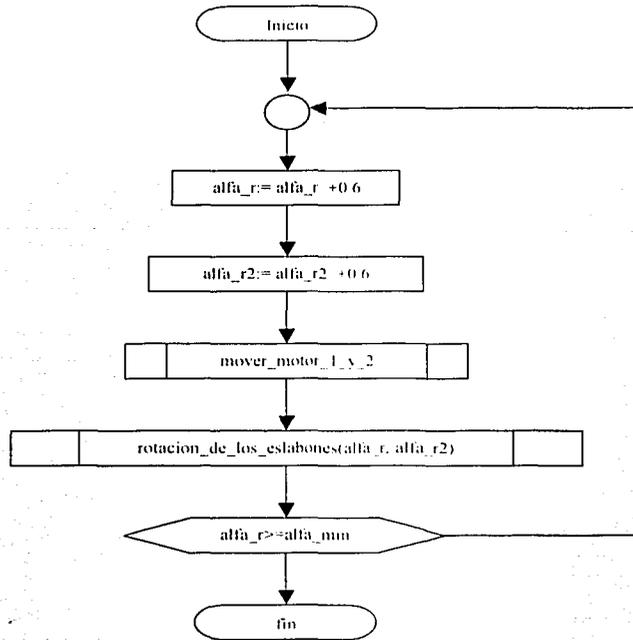


Figura 5.17 Diagrama de flujo del procedimiento "rotación_igual_eslabon_1_y_2" utilizado en la figura 5.16.

TESIS CON
FALLA EN LA ENTREGA

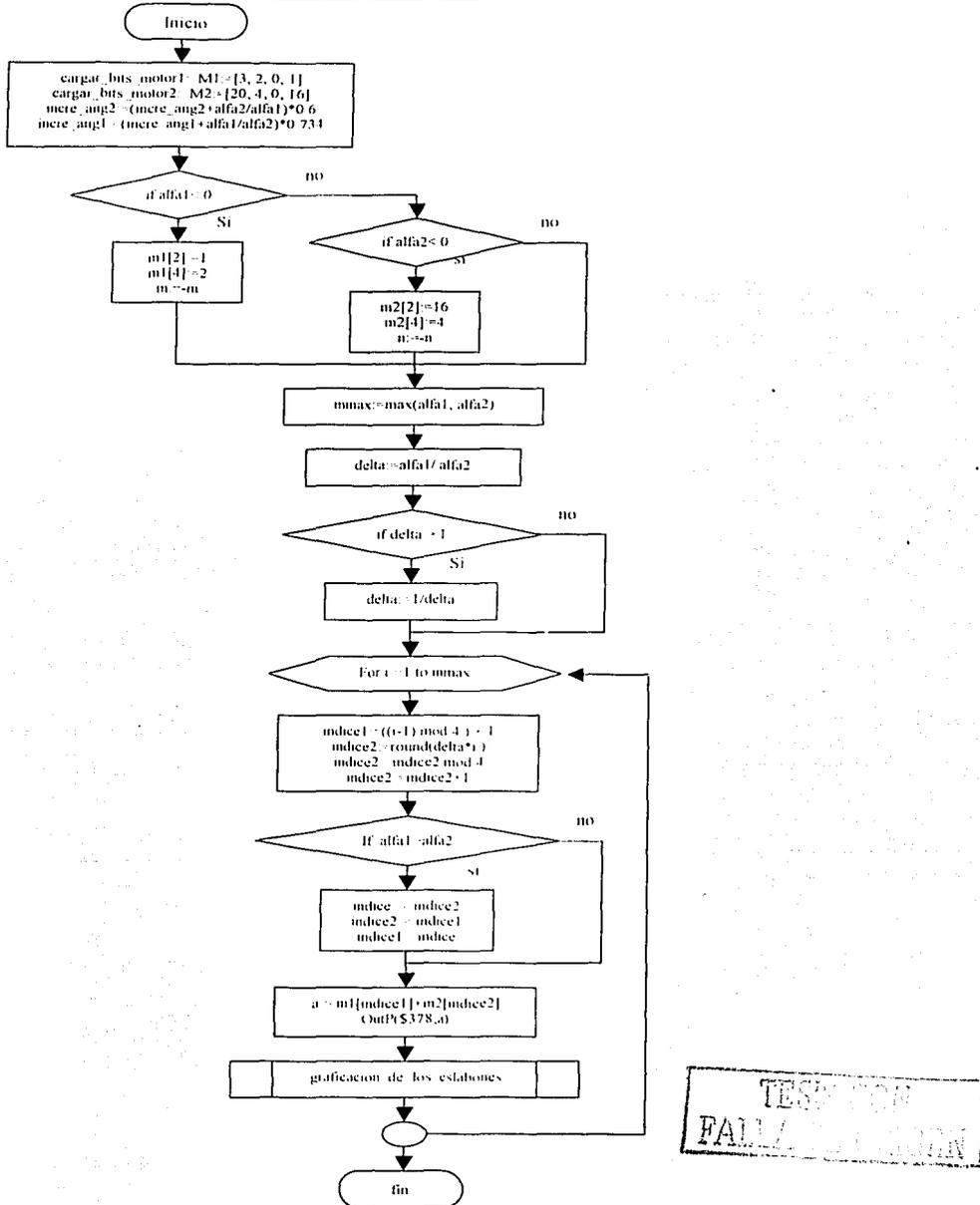


Figura 5.18 Diagrama de flujo para los movimientos simultáneos de ejes.

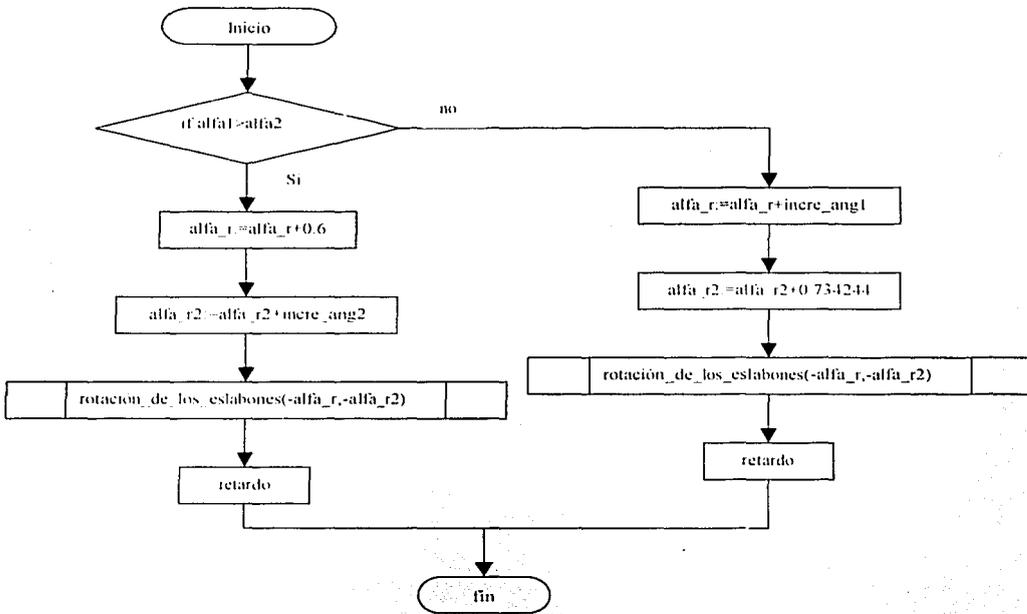


Figura 5.19 Diagrama de flujo del procedimiento "graficación_de_los_eslabones" utilizado en la figura 5.18.

TESIS CON
FALLA DE ORIGEN

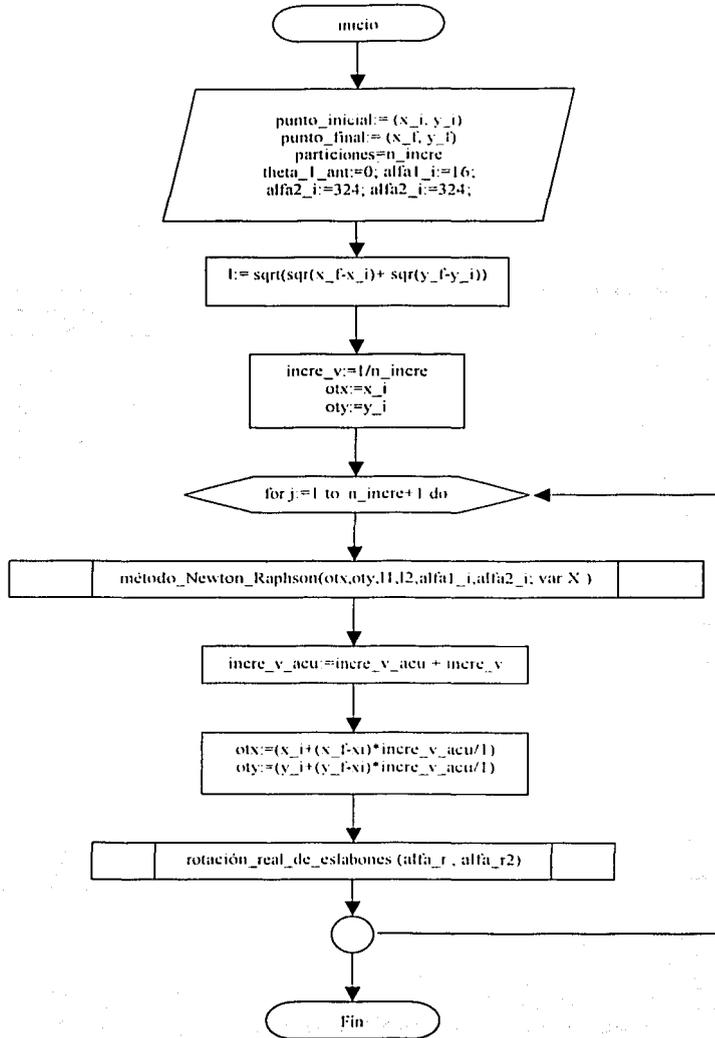
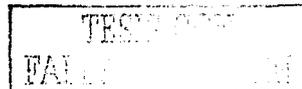


Figura 5.20 Diagrama de flujo utilizado en la descripción de los movimientos simultáneos de ejes que siguen una ruta (recta).



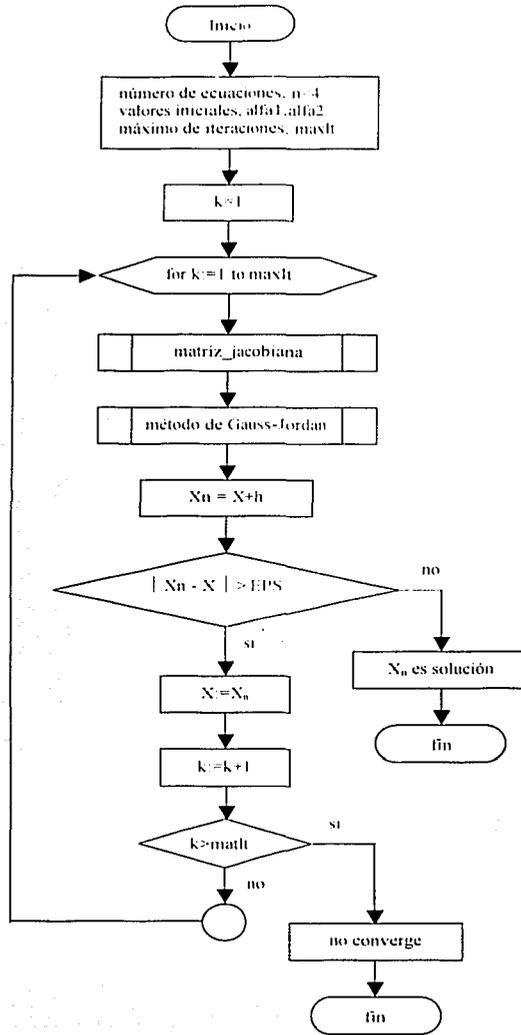


Figura 5.21 Diagrama del método del Newton_Raphson utilizado en la descripción de los movimientos continuos que siguen una ruta (recta).



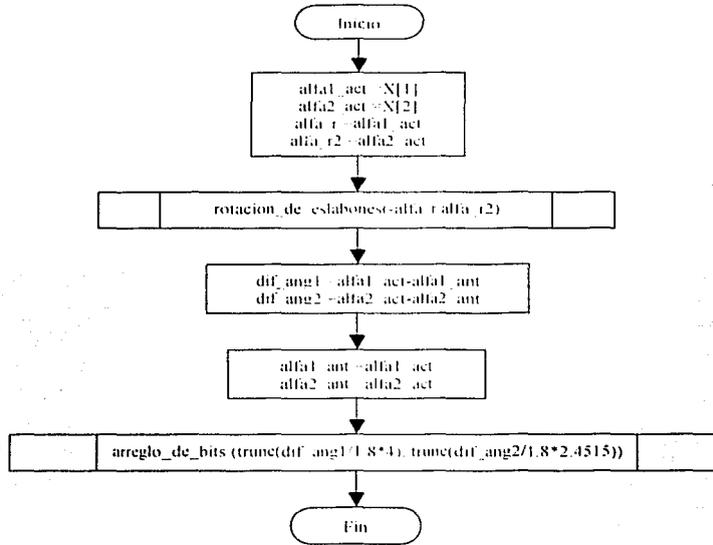


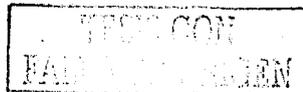
Figura 5.22 Diagrama de flujo del procedimiento "rotación_real_de_eslabones" utilizado en la figura 5.20.

TESIS CON
FACULTAD DE INGENIERIA

Los diagramas de flujo que se implementan para el problema de evasión de obstáculos deben satisfacer el requisito de seguir una trayectoria continua que en este caso es una línea recta.

La evasión se realiza utilizando: a) las secuencias de movimiento uno a uno de ejes, b) las secuencias híbridas de movimientos de ejes y c) las secuencias simultáneas de ejes. Esto dependiendo de lo que seleccione el usuario en el programa.

Cada una de las opciones se divide en tres partes: 1) el diagrama de flujo que se utiliza hasta encontrar el obstáculo (aplicando cinemática inversa y directa para seguir la ruta "recta"), 2) encontrado el obstáculo regresa a posición de inicio y calcula cinemática inversa reflectada para encontrar el ángulo que permita evadir y siga la ruta en el último punto que dejó cuando encontró el obstáculo. 3) evadido el obstáculo y el robot haber llegado en el último punto que dejó cuando encontró el obstáculo, entonces continua con la trayectoria hasta llegar al punto final. Es por ello que al principio se presenta los diagramas de flujo del inciso uno y tres que son similares para los tres tipos de evasión.



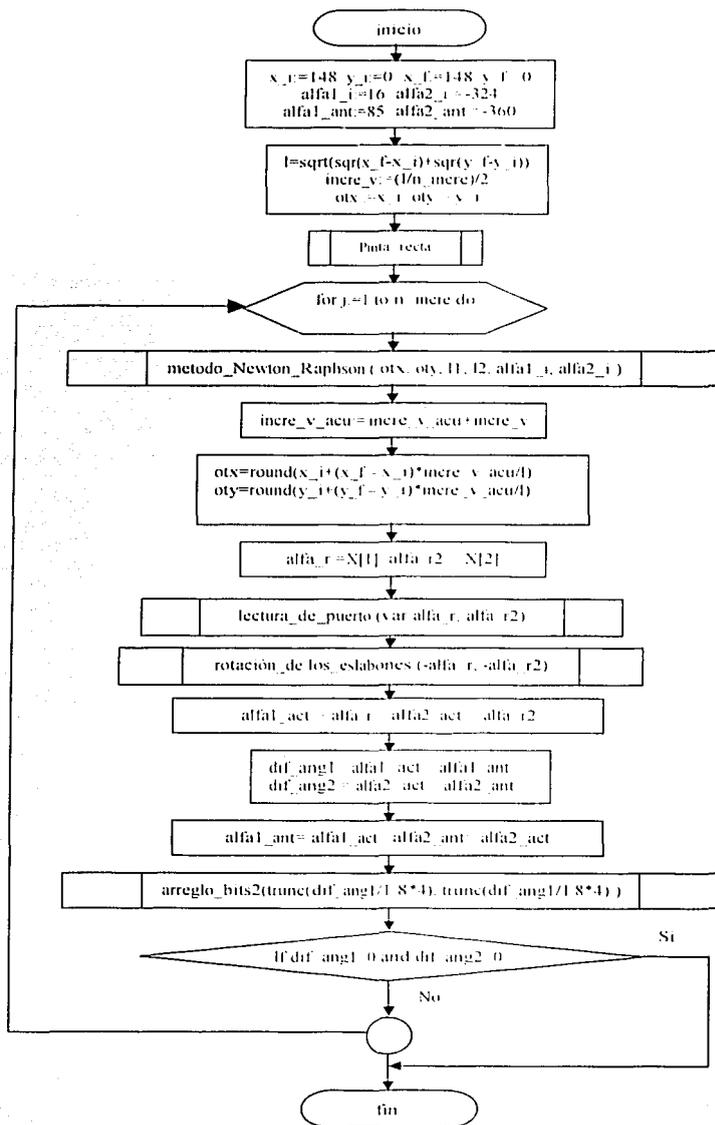
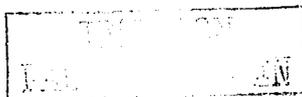


Figura 5.23 Primera parte del diagrama de flujo para la evasión de obstáculos.



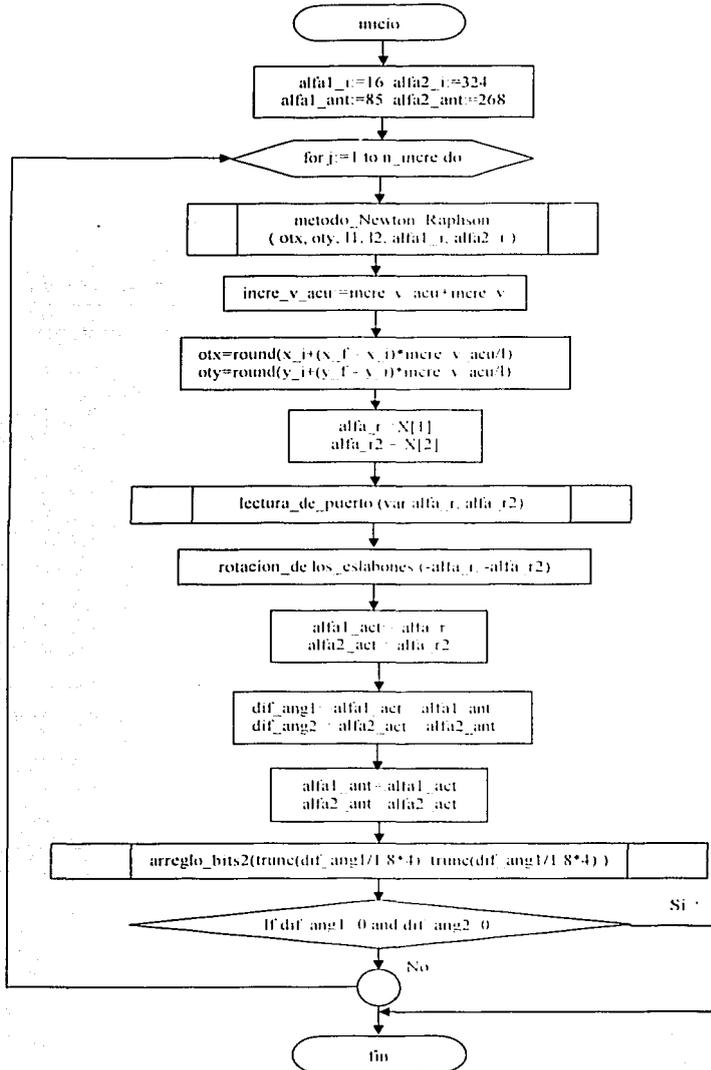
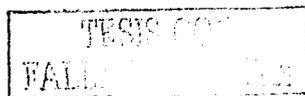
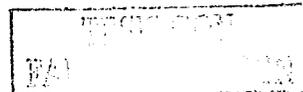
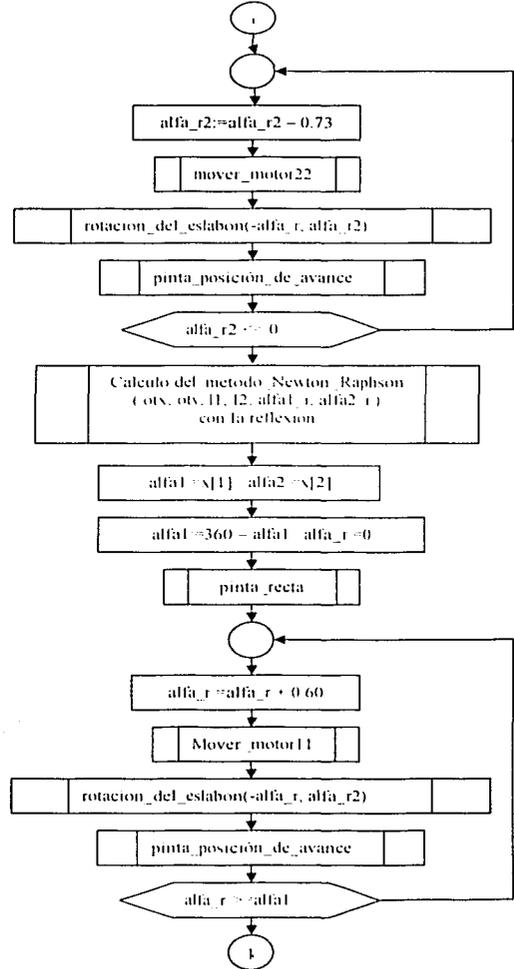
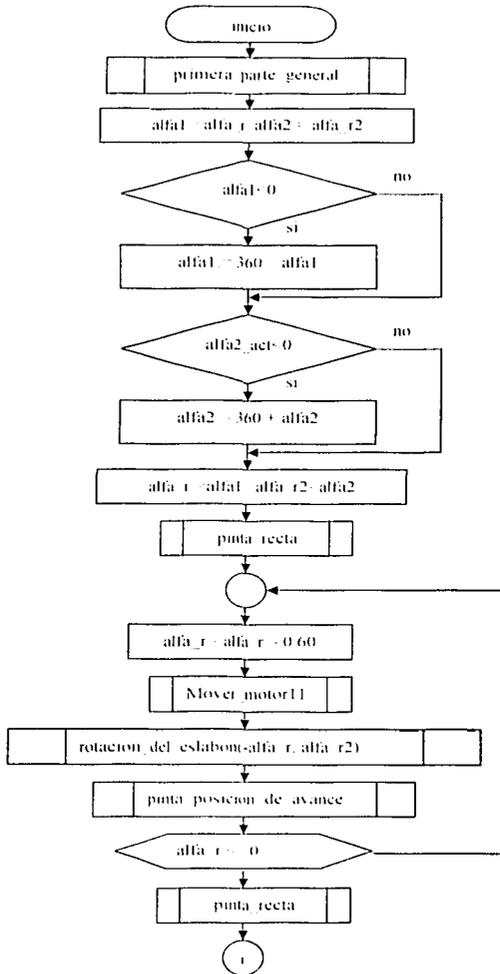


Figura 5.24 Tercera parte del diagrama de flujo para la evasión de obstáculos.





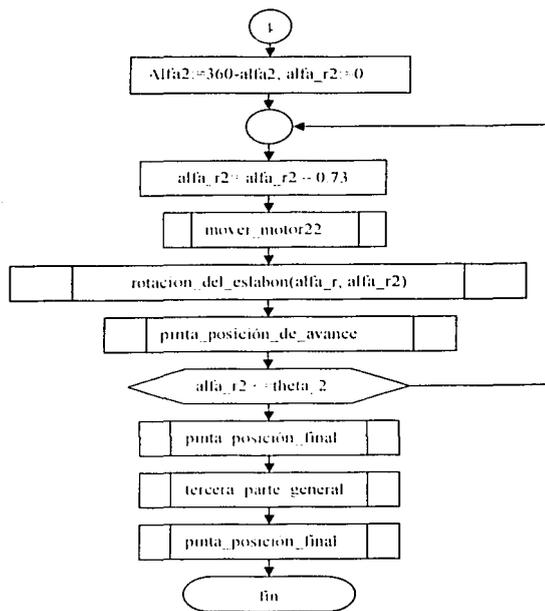
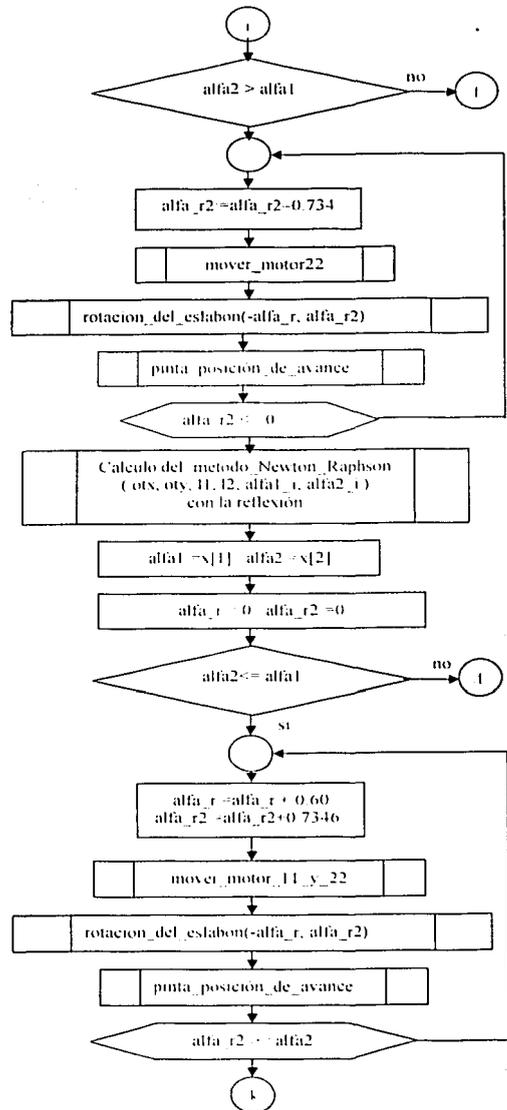
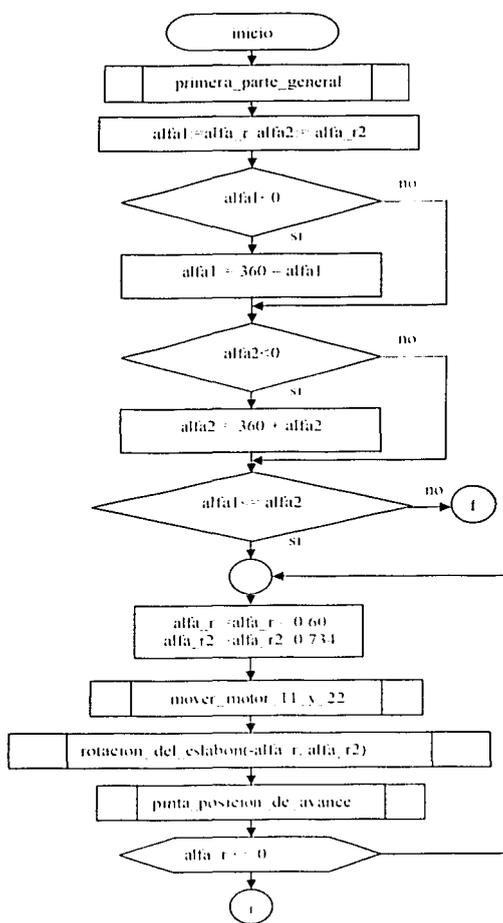


Figura 5.25 Diagrama para evasión con movimientos uno a uno.

TESTS CON
FALLS SCREEN



TESIS COM
 RAM
 BEN

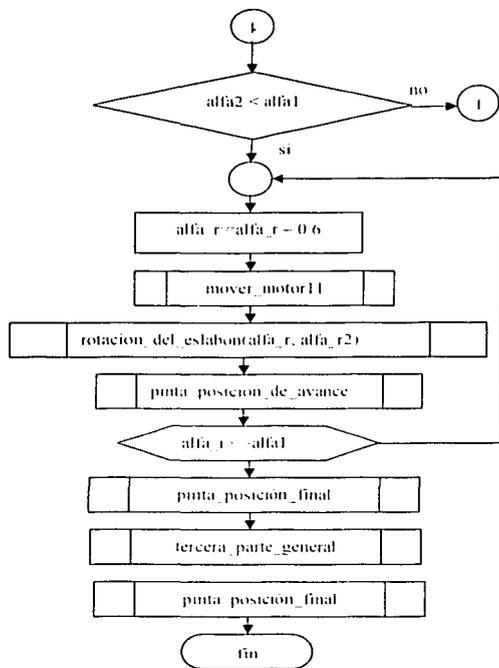


Figura 5.26 Diagrama para evasión con movimientos híbridos.

Nota. La letra "f" simboliza que se va al final de la rutina

TESIS CON
FABRIL ORIGIN

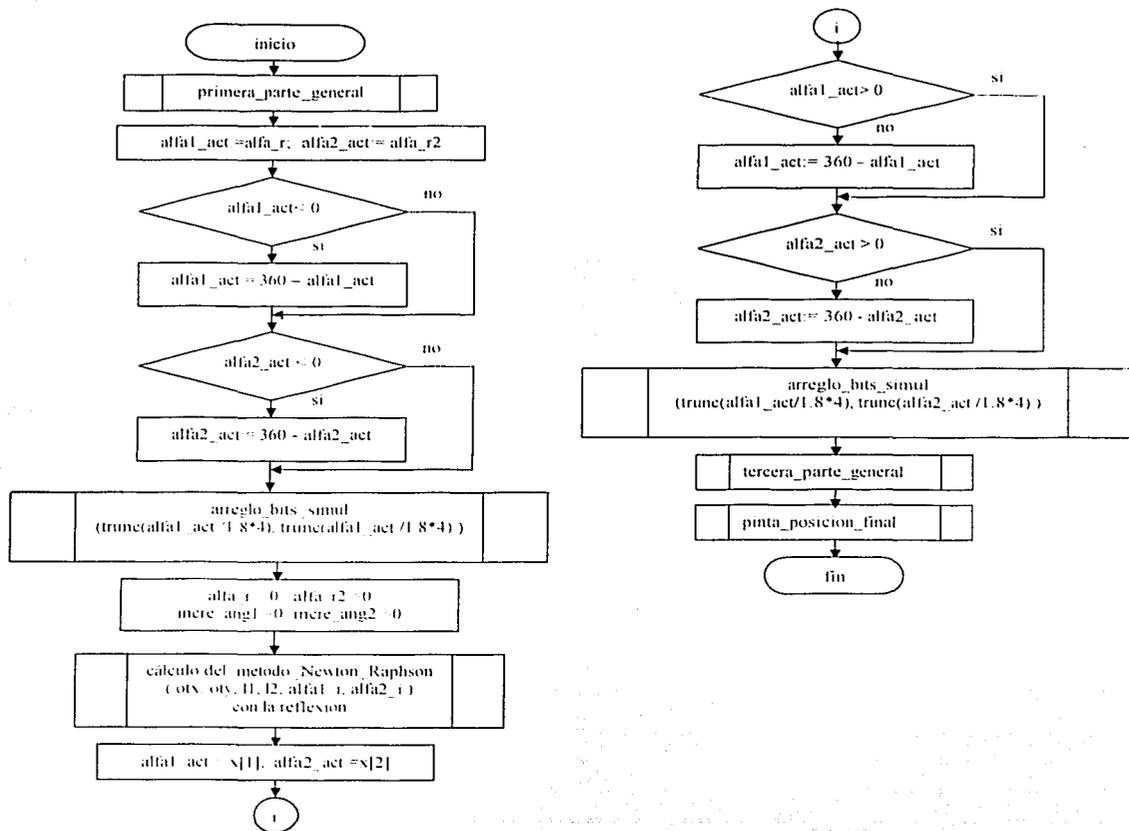


Figura 5.27 Diagrama para evasión con movimientos simultáneos.

COPIA CON
FIRMADO DEL ORIGEN

b) Codificación del programa para el manejo del robot de 2GDL con evasión de obstáculos en el plano

El programa fue codificado de acuerdo con los diagramas de flujo de las figuras 5.11 a la 5.27 en Delphi 5. Dicho programa se muestra en el apéndice A. El programa funciona sobre la plataforma de Windows 95 y 98.

5.3.3.2 Ejecución y depuración de EvaSoft 1.0

Durante la depuración del programa se encontraron tres tipos de errores que son: error de compilación, errores en tiempo de ejecución y los errores lógicos. Los cuales fueron arreglados.

5.3.4 Prueba y mantenimiento de EvaSoft 1.0

La prueba de funcionamiento del programa EvaSoft 1.0 se describe en la sección de resultados y conclusiones.

El mantenimiento de EvaSoft 1.0 implica cambios y mejoras para cumplir un nuevo conjunto de especificaciones. El mantenimiento referente a este programa consistirá en mejorarlo de acuerdo con los avances que se vayan teniendo en el proceso de investigación, es decir, para aumentar su capacidad para ejecutar trayectorias complejas.



Capítulo 6

Resultados obtenidos de la modelación y prueba de secuencias de movimientos del robot de dos grados de libertad para la evasión de obstáculos

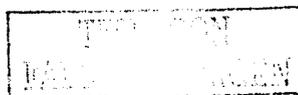
Los resultados obtenidos durante la elaboración de este trabajo de tesis denominado " Modelación y prueba de secuencias de movimientos de un robot de dos grados de libertad para un problema de evasión de obstáculos en el plano" son: la obtención de los modelos cinemáticos de dicho robot, el análisis y la obtención de los tipos de movimientos, la generación de los criterios de evasión, la construcción del manipulador de dos grados de libertad, el armado de los circuitos electrónicos para el funcionamiento de los actuadores y sensores, ya mencionados con mayor detalle en el capítulo 5; así como el programa de cómputo (EvaSoft 1.0) y finalmente la prueba de funcionamiento de dicho programa en el robot de estudio.

En lo referente a EvaSoft 1.0, este es un software con aplicación directa para el manejo de las secuencias de movimiento de dicho robot. Dentro de las formas de manejar el robot incluyen: los movimientos uno a uno, movimientos simultáneos de ejes y movimientos híbridos (que utilizan simultáneos y uno a uno), movimientos simultáneos que siguen una ruta, así como el problema de la evasión mediante las secuencias de movimiento anteriormente mencionadas.

6.1 Tipos de movimientos

Los tipos de movimientos que fueron programados en el robot de 2GDL y ambientados gráficamente en pantalla de la computadora son:

- a) Movimientos uno a uno.
- b) Movimientos simultáneos.
- c) Movimientos simultáneos que siguen una ruta.
- d) Programación de cada uno de las secuencias de movimientos para la evasión de obstáculos.



6.2 Instalación

En lo referente a la instalación, esta se refiere tanto del software como del hardware. Para ello se divide en "los requerimientos del programa", "la instalación del hardware", "la instalación y su desinstalación de EvaSoft 1.0".

6.2.1 Requerimientos del programa

Para el buen funcionamiento del programa se utilizó el equipo que se menciona en el capítulo 5. Y para el caso del microcomputador, también se puede utilizar una IBM PC o compatible con los requisitos mínimos siguientes: microprocesador 486, 32 MB en RAM con plataforma Windows 95.

6.2.2 Instalación del hardware

En lo referente a la instalación del hardware que en este caso se refiere a los dispositivos electrónicos de control del robot, se armaron dándole un especial énfasis a la alimentación de los circuitos para evitar de esta forma cualquier corto circuito. Estos dispositivos son los mencionados en el capítulo 5.2.

6.2.3 Instalación de EvaSoft 1.0

Terminada la instalación de los componentes físicos del sistema, enseguida se procedió con la instalación de EvaSoft 1.0, que se debe ejecutar desde el ambiente Windows 95, y según donde se encuentre el disco de instalación: A:\Principal01.exe o B:\Principal01.exe.

La instalación se realiza en el subdirectorio c:\EvaSoft, en el cual se encuentra el archivo ejecutable **Principal01.exe**, el archivo de desinstalación **Desintalar.exe**.

6.2.4 Desinstalación de EvaSoft 1.0

Si después de usar EvaSoft 1.0, ya no se está interesado en mantenerlo en el disco duro, ni en el administrador de programas, debe hacerse uso del programa de desinstalación, que tiene el nombre de "Desinstalar EvaSoft" el cual pide la confirmación para la desinstalación del programa, y si es afirmativo lo borra completamente.



6.3 El programa EvaSoft 1.0

Este programa esta constituido por un grupo de subprogramas que se interrelacionan entre sí para actuar sobre el robot, y controlar el tipo de trayectoria elegida. En la tabla 6.1 se presenta la relación de los archivos que conforman a éste. cabe hacer mención que la ausencia de alguno de ellos causa problemas en el funcionamiento del software.

Unidad	Descripción
Principal01.dpr	Este es el archivo de proyecto que enlaza todas las unidades
Principal1.pas	En esta unidad se realiza el llamado a otras unidades con el objetivo de utilizar el tipo de trayectoria que el usuario quiera utilizar
Acercade.pas	Esta unidad ofrece información referente al tesista, director y sus asesores
Mov_indepen1.pas	Esta unidad contiene las secuencias de movimientos uno a uno
Tray_Program_.pas	Esta unidad contiene las secuencias de movimientos simultáneos e híbridos
Tray_cont1	Esta unidad contiene las secuencias simultáneas que siguen una ruta (recta)
Tiempo_real_1.pas	Esta unidad contiene las secuencias de movimientos para evadir obstáculos
Prende_motores	Esta unidad contiene los procedimientos para mandar los pulsos a los motores
Port.dll	Esta librería auxilia a la unidad de "Prende_motores", para la entrada y salida de datos por el puerto paralelo externo

Tabla 6.1 Unidades que componen a EvaSoft 1.0.

6.4 Prueba de funcionamiento de EvaSoft 1.0

Una vez construido mecánicamente e instalado los dispositivos electrónicos para el manejo del manipulador se procedió a la prueba de EvaSoft 1.0.

Llamado el programa, aparece el entorno de trabajo de EvaSoft 1.0, el cual es mostrado en la figura 6.1 y en se muestran los menús desplegables del programa, estando disponibles las opciones de: "Instrucciones de uso del robot", "Programas manejadores" y "Acercas del programa".



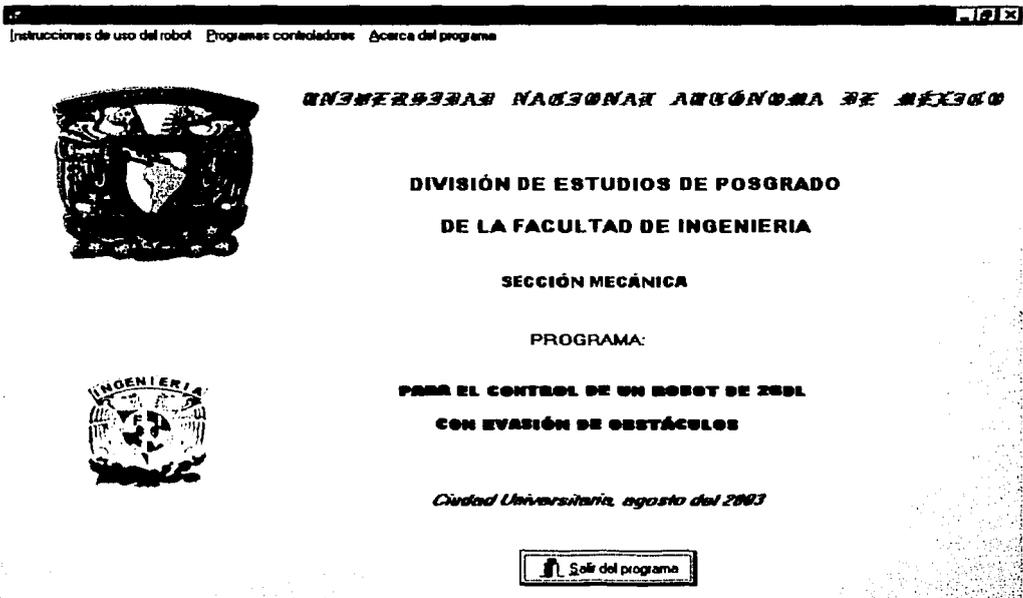


Figura 6.1 Pantalla principal del programa EvaSoft 1.0.

TESIS CON
FALLA EN LA INGEN

6.4.1 Resultados para el manejo de la forma de movimientos uno a uno

Dentro de la forma principal del programa, se selecciona la opción "Programas manejadores", posteriormente "Movimientos uno a uno". En primer lugar se oprime el botón "Posición de inicio" para que el robot busque se posición de inicio a través de sensores (optorelectores). Y lo que resulta de ello, es lo que se muestra en la figura 6.2

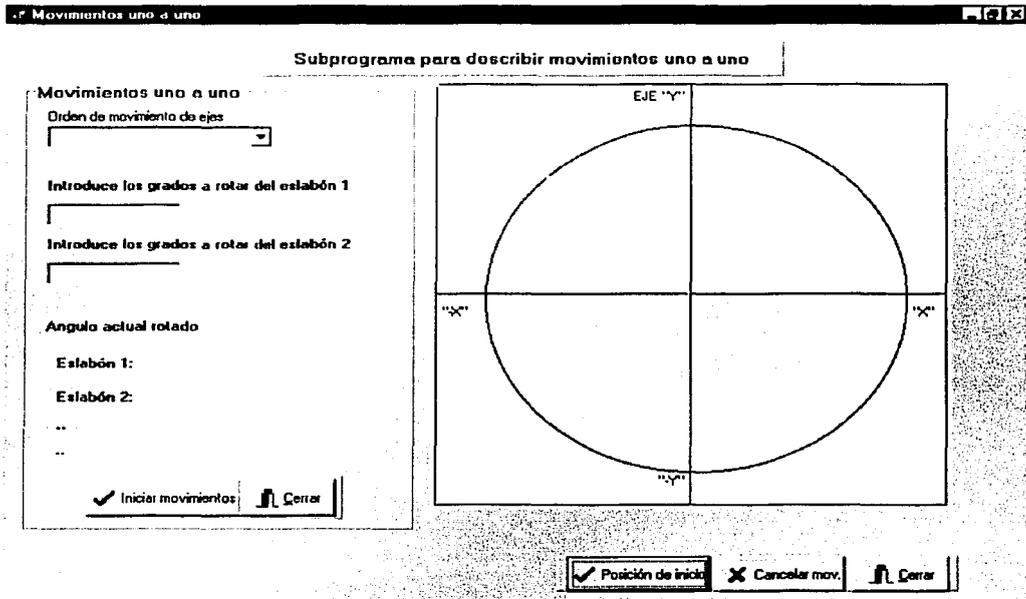


Figura 6.2 Pantalla de la posición de inicio para movimientos uno a uno.

TESIS CON
FALLA TÉCNICA

Posteriormente, se introducen los datos requeridos en la pantalla, tal y como se muestra en la figura 6.3, sin omitir alguno de ellos:

- Seleccionar el movimiento de los eslabones.
- Introducir el ángulo a rotar del eslabón 1.
- Introducir el ángulo a rotar del eslabón 2.

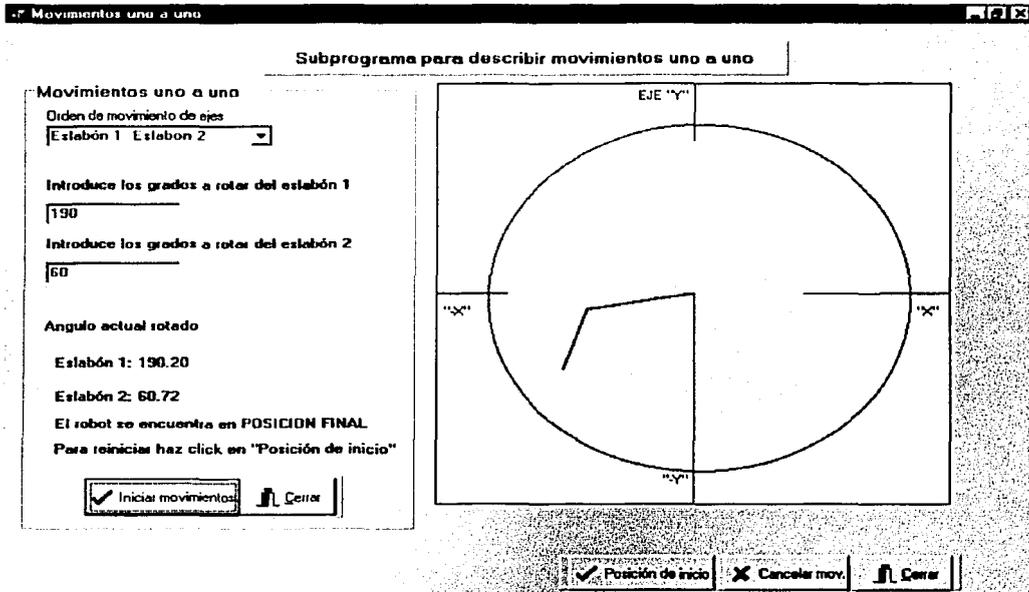


Figura 6.3 Pantalla de los movimientos uno a uno de ejes (primero el eslabón uno y posteriormente el eslabón dos).

TESIS CON
FALLA DE ORIGEN

Subprograma para describir movimientos uno a uno

Movimientos uno a uno

Orden de movimiento de ejes

Introduce los grados a rotar del eslabón 1

Introduce los grados a rotar del eslabón 2

Angulo actual rotado

Eslabón 1: 45.00

Eslabón 2: 110.59

El robot se encuentra en POSICION FINAL

Para reiniciar haz click en "Posición de inicio"

Iniciar movimientos
 Cerrar

Posición de inicio
 Cancelar mov.
 Cerrar

Figura 6.4 Pantalla de los movimientos uno a uno de ejes (primero el eslabón dos y posteriormente el eslabón uno).

Cabe hacer mención que la diferencia de los resultados obtenidos de la figura 6.3 y 6.4 es el orden en que se mueven los eslabones. Para el caso de la figura 6.3, primero se mueve el eslabón 1 y posteriormente el eslabón 2. Y en la figura 6.4 los eslabones se mueven en orden contrario, es decir, primero el eslabón 2 y posteriormente el eslabón 1.

TESIS CON
 FALLA DE ORIGEN

6.4.2 Resultados en el manejo de la forma de movimientos simultáneos e híbridos de ejes

Dentro de la forma principal del programa, se selecciona la opción "Programas manejadores", posteriormente "Movimientos simultáneos e híbridos". En primer lugar se oprime el botón "Posición de inicio" para que el robot busque se posición de inicio a través de sensores (optorelectores). Y lo que resulta de ello, es lo que se muestra en la figura 6.5

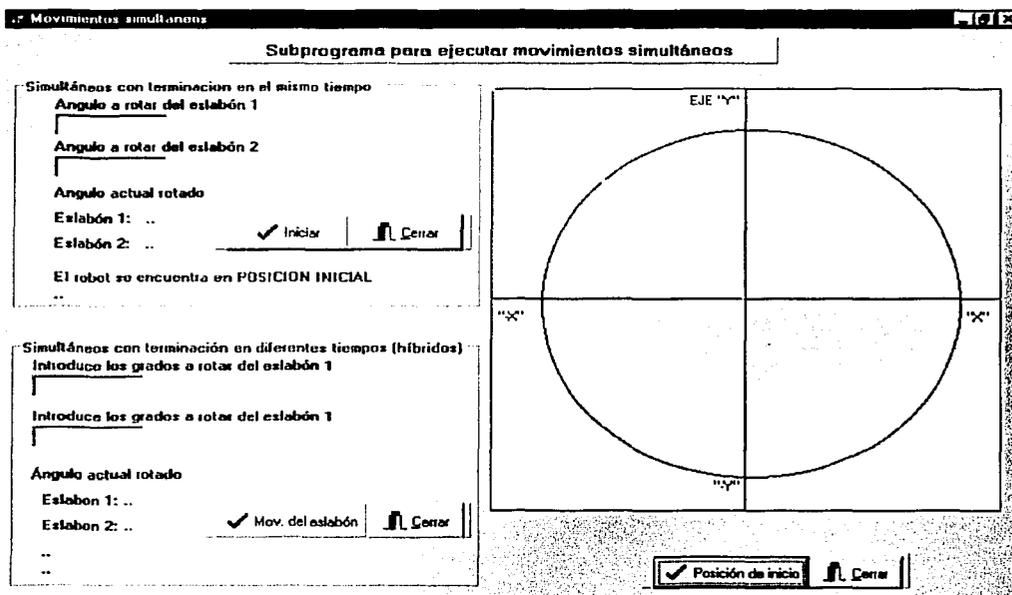


Figura 6.5 Pantalla de la posición de inicio para movimientos simultáneos e híbridos.

TECNOLOGIA
FALLA DE ORIGEN

a) Movimientos simultáneos

Se introducen los datos requeridos en la pantalla, tal y como se muestra en la figura 6.6:

- 1) Introducir el ángulo a rotar del eslabón 1.
- 2) Introducir el ángulo a rotar del eslabón 2.

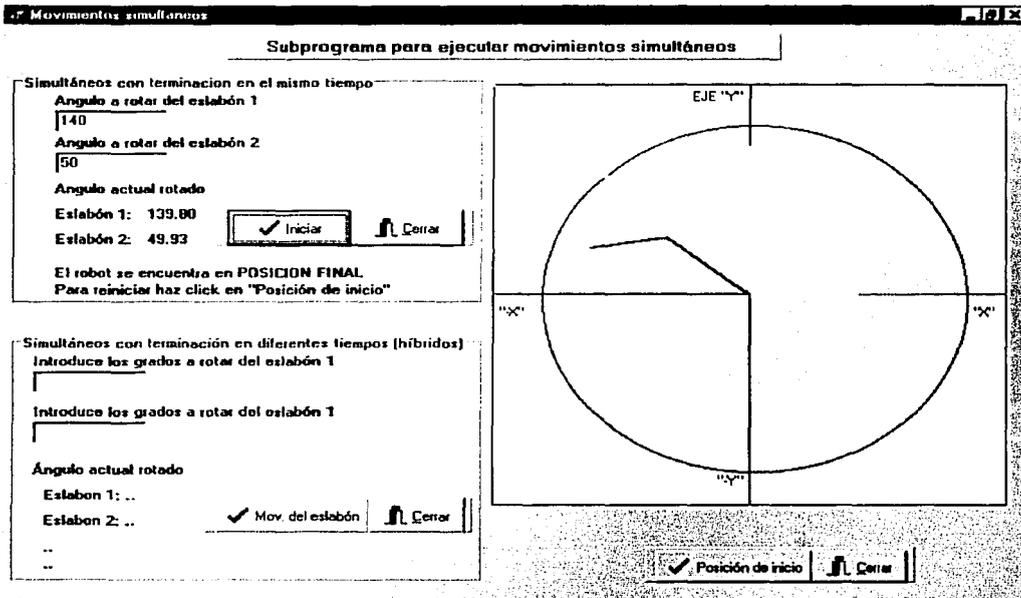
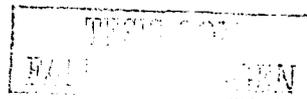


Figura 6.6 Pantalla de los movimientos simultáneo de ejes.

De la misma manera una vez seleccionado e introducido los datos, se oprime el botón **“Posición de inicio”** para que el robot busque su posición de inicio a través de sensores (optorelectores). Y, enseguida, se oprime el botón que tiene la leyenda **“Iniciar”**.

El resultado obtenido es que los dos eslabones inician el movimiento al mismo tiempo y terminan al mismo tiempo, esto es, independientemente del ángulo que tengan que recorrer cada uno de ellos.



b) Movimientos híbridos

Se introducen los datos requeridos en la pantalla, tal y como se muestra en la figura 6.7:

- 1) Introducir el ángulo a rotar del eslabón 1.
- 2) Introducir el ángulo a rotar del eslabón 2.

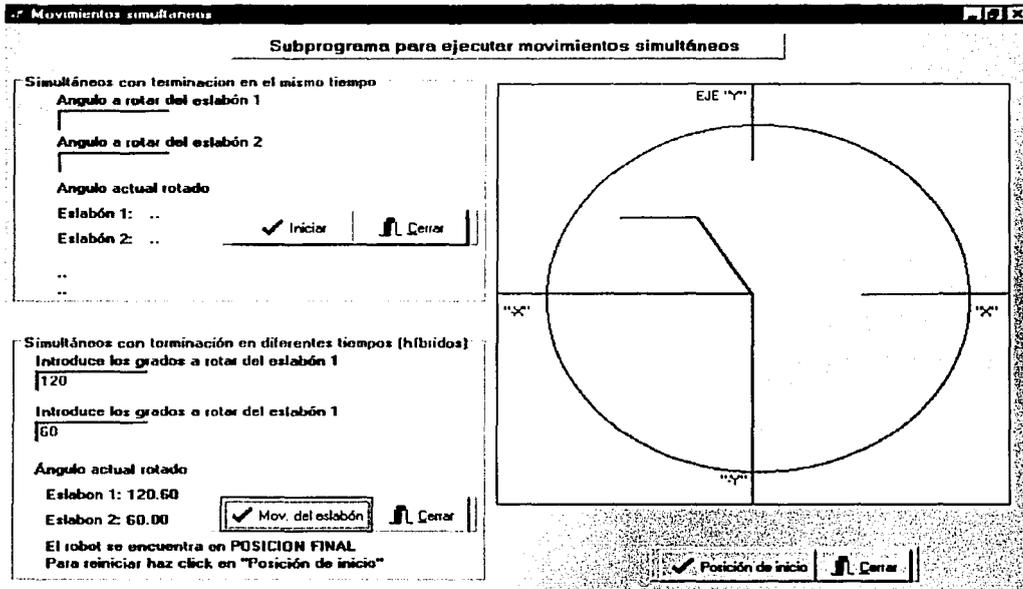


Figura 6.7 Pantalla de los movimientos híbridos.

El resultado obtenido de los movimientos híbridos se puede ver en la figura 6.7. Estos movimientos tienen la característica que inician el movimiento al mismo tiempo y avanzan de igual manera pero su terminación depende de los grados que necesite recorrer cada uno de ellos.

TERMINACIÓN
PARA REINICIAR

6.4.3 Resultados en el manejo de la forma de movimientos simultáneos que siguen una ruta

Dentro de la forma principal del programa, se selecciona la opción "Programas manejadores", posteriormente "Movimientos simultáneos que siguen una ruta" y se oprime el botón "Posición de inicio" para que el robot busque su posición de inicio a través de sensores de posición (optorefectores). Lo cual, se muestra en la figura 6.8

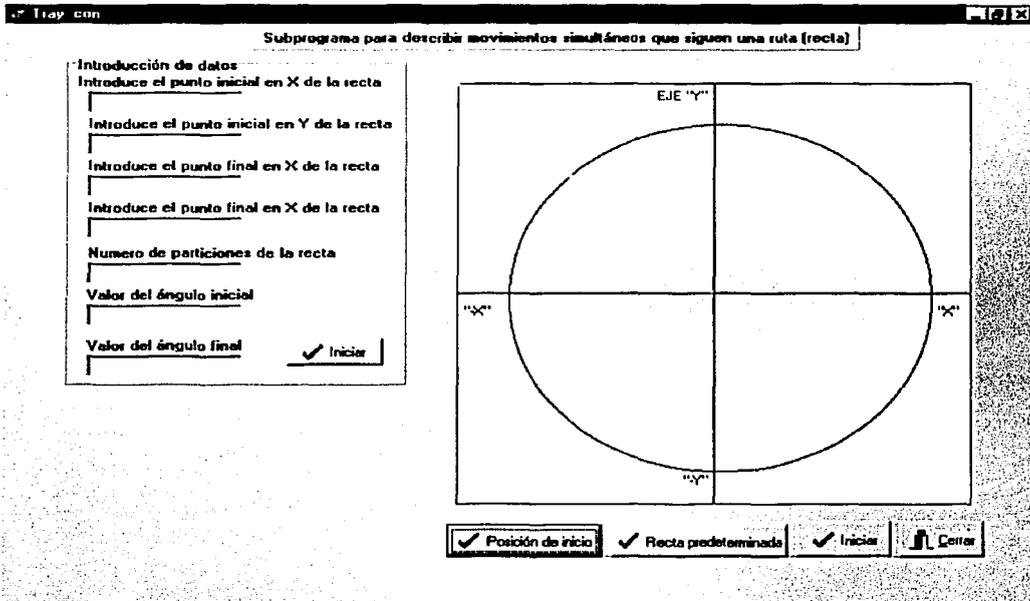


Figura 6.8 Pantalla de la posición de inicio para las trayectorias continuas.

Posteriormente, se introducen los datos requeridos en la pantalla, tal y como se ve en la figura 6.8, sin omitir alguno de ellos:

- a) Introducir el punto inicial en X.
- b) Introducir el punto inicial en Y.
- c) Introducir el punto final en X.
- d) Introducir el punto final en Y.

TESIS CON
FALLAS EN
EL PROCESO

- e) Introducir el número de particiones de la recta.
- f) Valor del ángulo inicial propuesto para alfa1.
- g) Valor del ángulo inicial propuesto para alfa2.

Y a continuación se oprime el botón con la leyenda “Iniciar”. Lo que se obtiene es lo mostrado en la figura 6.9.

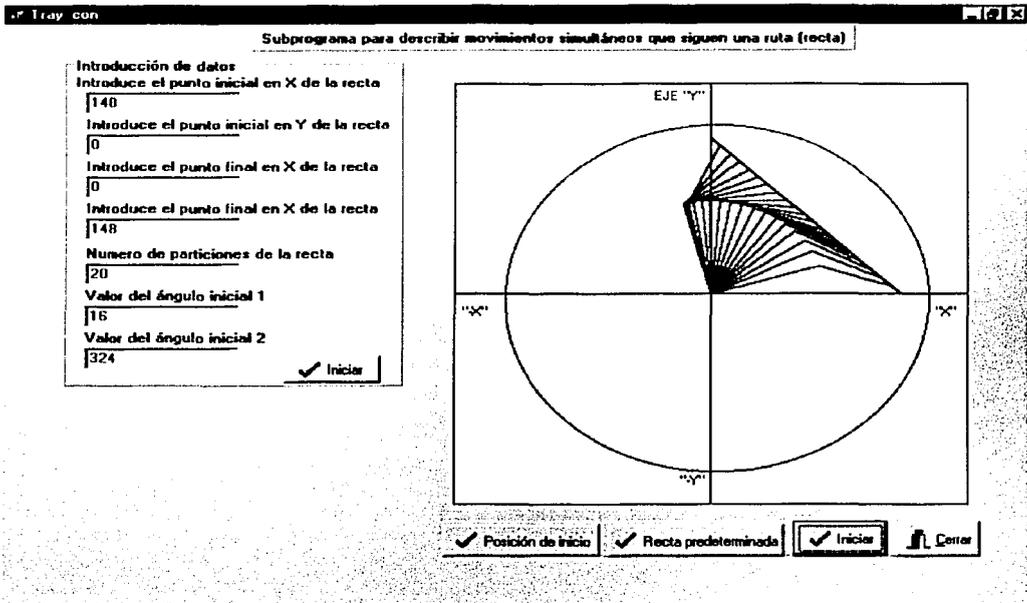


Figura 6.9 Pantalla de los movimientos simultáneos que siguen una ruta.

El resultado obtenido de los movimientos simultáneos que siguen una ruta se presenta en la figura 6.9, en el cual se muestra como el robot va siguiendo una línea recta. Cada una de las posiciones indica el ángulo que debe de recorrer el robot en la realidad. Dicho robot parte de la posición de inicio de la recta hasta llegar al punto final.

TESIS CON
FALTA DE
CEN

6.4.4 Resultados en el manejo de la forma para la evasión de obstáculos en el plano

Dentro de la forma principal del programa, se selecciona la opción "Programas manejadores", posteriormente "La evasión de obstáculos en el plano" y se oprime el botón "Posición de inicio" para que el robot busque su posición de inicio a través de sensores de posición (optorelectores). Lo cual, se muestra en la figura 6.10

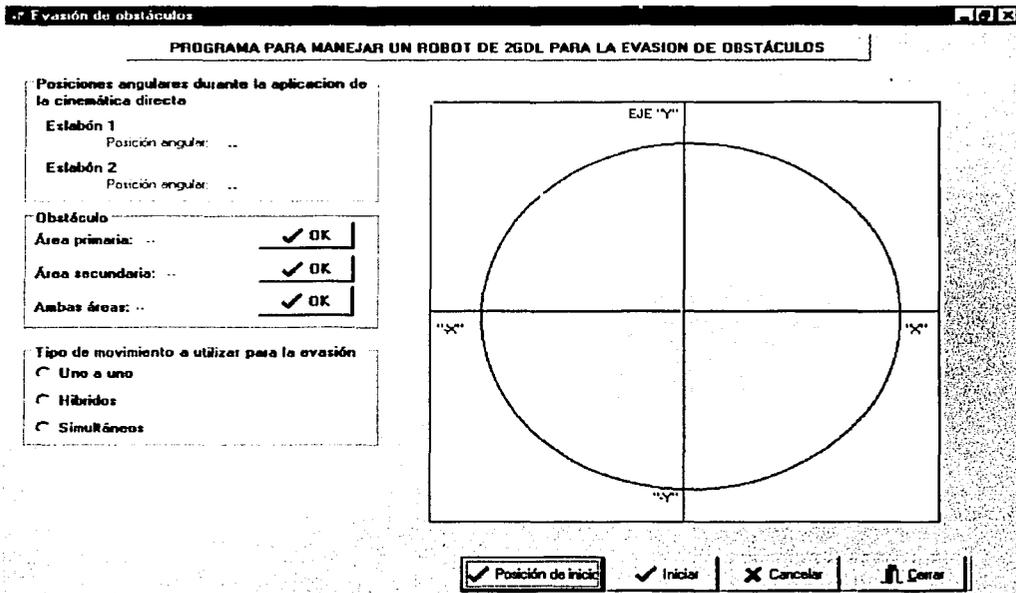


Figura 6.10 Pantalla de la posición de inicio.

TESIS CON
PALABRAS CLAVES

Posteriormente se selecciona el tipo de evasión de obstáculo que se quiera, tal y como se ve en la figura 6.11:

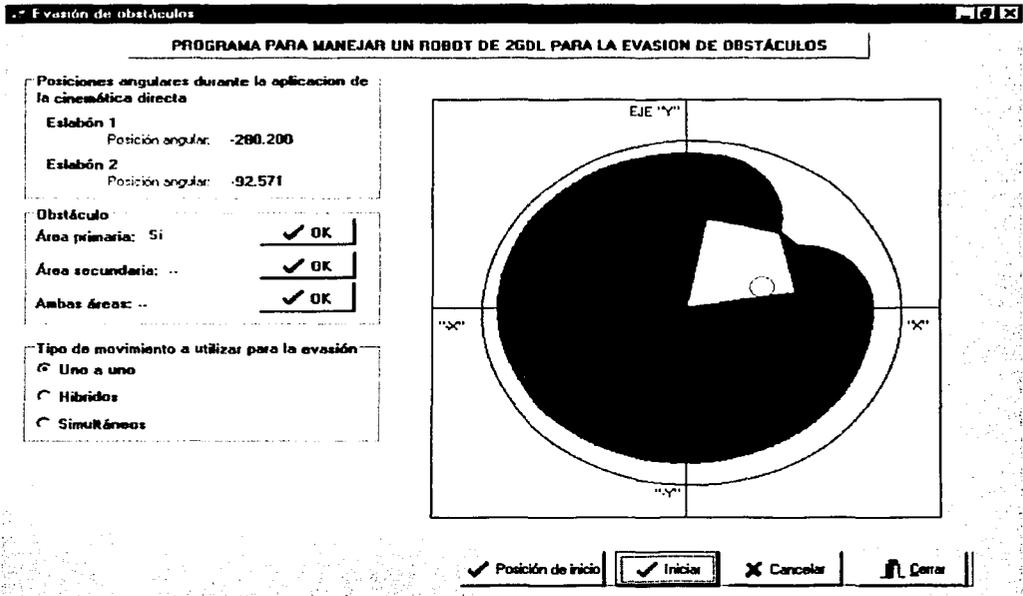


Figura 6.11 Pantalla de la evasión de obstáculos utilizando la secuencia de movimientos uno a uno.

Para este caso, lo que se muestra en la figura 6.11 es la evasión de obstáculos utilizando las secuencias de movimientos uno a uno de ejes del robot de 2GDL. En el cual, lo que se sigue es la línea recta y a mitad del camino encuentra el obstáculo, entonces se regresa a posición de inicio con movimientos uno a uno y posteriormente evade el obstáculo también con movimientos uno a uno de ejes, llegando al último punto que se quedó cuando encontró el obstáculo, y continua el seguimiento de la recta hasta llegar al punto final.

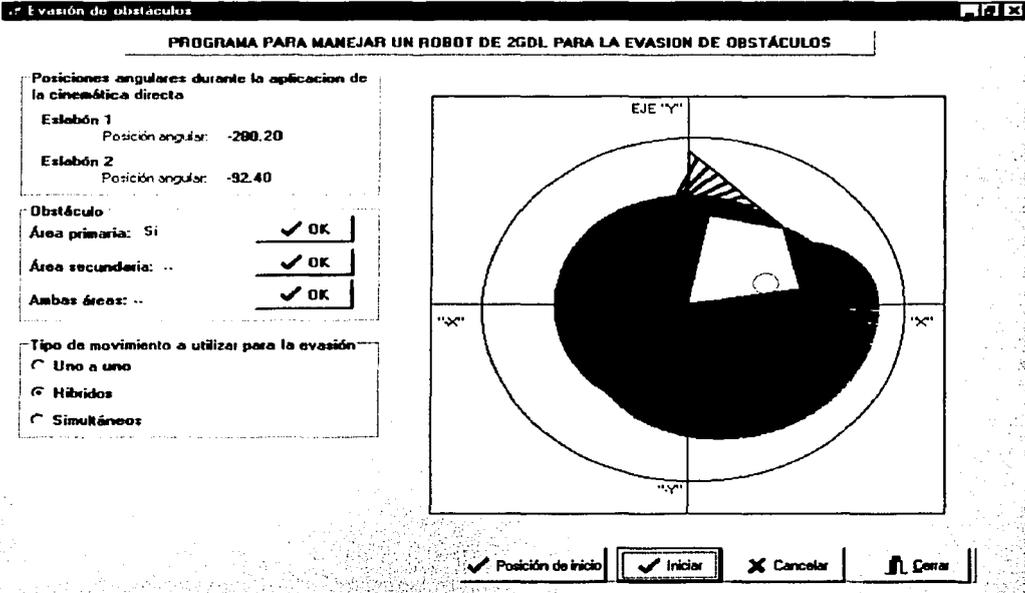


Figura 6.12 Pantalla de la evasión de obstáculos utilizando la secuencia de movimientos híbridos.

Para el caso de la figura 6.12 es la evasión de obstáculos utilizando los movimientos híbridos del robot de 2GDL. Lo que también se sigue es la línea recta y a mitad del camino encuentra el obstáculo, entonces se regresa a posición de inicio utilizando movimientos híbridos y posteriormente evade el obstáculo también con movimientos híbridos, llegando al último punto que se quedó cuando encontró el obstáculo, y continua describiendo la recta hasta llegar al punto final.

TESIS CON
FALSA ORIGEN

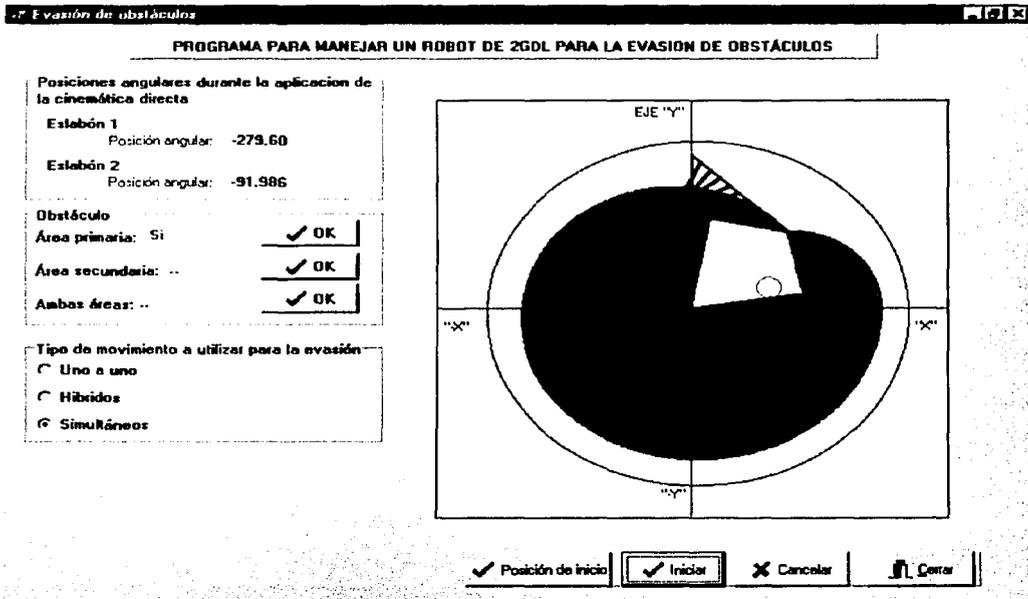


Figura 6.13 Pantalla de la evasión de obstáculos utilizando la secuencia de movimientos simultáneos.

Para el caso de la figura 6.13 lo que muestra, es la evasión de obstáculos utilizando los movimientos simultáneos de dicho robot experimental. Lo que se sigue es la línea recta y a mitad del camino encuentra el obstáculo, entonces se regresa a posición de inicio utilizando movimientos simultáneos y posteriormente evade el obstáculo también con movimientos simultáneos, hasta llegar al último punto que se quedó. Y continua siguiendo la línea recta hasta llegar al punto final de dicha recta.

TRABAJO CON
FARMACIA SINGEN

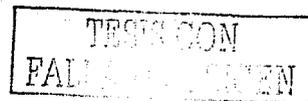
Conclusiones

A través del presente trabajo de tesis fueron cumplidos los objetivos siguientes:

- Se analizó, modeló y programó un problema de evasión de obstáculos en el plano entre un robot de 2GDL y un obstáculo de geometría circular.
- Fueron analizados y caracterizados los movimientos completos del robot y, dichos movimientos, se usaron para generar los criterios de evasión.
- Se construyó un robot prototipo de 2GDL y un software (EvaSoft 1.0) para graficar y probar en tiempo real los tipos de movimientos en el problema de evasión de obstáculos.

Los resultados obtenidos en esta tesis se resumen en los puntos siguientes:

- El modelo cinemático de posicionamiento fue construido con la rotación usual definida en el espacio vectorial de los números complejos. Dicho modelo fue claro y sistemático.
- El número de movimientos completos uno a uno del robot de 2GDL fue de 8: 1) 2 reales, 2) 2 complementarios y 3) 4 combinados.
- El número de movimientos simultáneos fue de 4.
- El número de movimientos híbridos fue de 16.
- El marco teórico desarrollado para analizar, caracterizar y modelar los movimientos del robot fue claro y sistemático.
- Los criterios de evasión fueron usados en forma sistemática, para modelar el problema de evasión. Este proceso sistemático permitió probar con un prototipo experimental, los movimientos del robot y el problema de evasión.



Recomendaciones de trabajos futuros

Es importante señalar que, el proceso de evasión fue analizado bajo el concepto de desplazamiento y, por tanto, será necesario construir un modelo cinemático completo, es decir, desplazamientos, velocidades, aceleraciones y trayectoria (específicamente, el perfil de trayectoria). Dicho modelo tendrá que ser probado tanto computacionalmente como en la realidad.

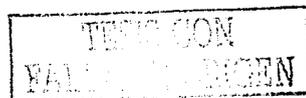
En lo referente al problema de evasión de obstáculos en general, los temas siguientes son importantes:

- Evasión de obstáculos en el espacio.
- Evasión con obstáculos en movimiento.
- Construcción de robots prototipos para el problema de evasión de obstáculos en el espacio.

TESIS CON
FALLAS EN BIEN

Referencias bibliográficas

- [1]. Angeles, J. 1997. Fundamentals of Robotic Mechanical Systems: Theory, Methods and Algorithms. Springer-Verlag, New York, USA.
- [2]. Barrientos, A. *et. al.* 1997. Fundamentos de robótica. McGraw-Hill, Madrid, España. Pags. 169-185.
- [3]. Bolton, W. 1998. Mechatronics: electronic control systems in mechanical ingeniering. Addison Wesley Longman Limited, England.
- [4]. F. Wakerly, J. 1981..Microcomputer Architecture and Programming. Ed. John Wiley. United States of America.
- [5]. Jiménez L., E. 1998. Simulación de un proceso de manufactura con obstáculos en la línea de producción. Tesis de grado de Maestría en Ingeniería Mecánica, DEPFI. UNAM.
- [6]. Fu, K. S.; González, R.C. and Lee, C.S.G. 1987. Robotics: control, sensing, vision and intelligence. McGrawHill, United State of America.
- [7]. Festo. Datos técnicos del manual para el sensor de barrera emisor de luz roja (SOEG-L-Q30-PA-K-2L).
- [8]. Festo. Datos técnicos del manual para la fibra óptica de polimero (SOEZ-LLK-SE-2,O-M4).



- [9]. Jiménez L., E.; Reyes A., L. *et. al.* 2002. Sobre la parametrización de las rotaciones y reflexiones de multicuerpos rígidos en el plano; modelación cinemática de un robot de dos grados de libertad. Informe Interno de Investigación. División de Estudios de Posgrado de la Facultad de Ingeniería, UNAM, México.
- [10]. Joyanes A., L.1993. Turbo Pascal 7.0. McGraw-Hill. Madrid, España.
- [11]. Juárez C., I. Diseño y construcción de un manipulador de 5 GDL.1992. Tesis de grado de Maestría en Ingeniería Mecánica, DEPI, UNAM.
- [12]. Keith, W.; Lorent, H. and Namir, S.1989. Power Graphics using Turbo Pascal. Ed. John Wiley & Sons, Inc. United States of America. Pags 107-133.
- [13]. Latombe, J. 1991. Robot motion planning. Kluwer Academic Publishers, United State of America.
- [14]. Martínez M., R.1997. Modelación y simulación computacional de trayectorias tridimensionales. Tesis de grado de Maestría en Ingeniería Mecánica, DEPI, UNAM.
- [15]. Microelectronics: SGS-THOMSON. Diagrama esquemático y de conexión para motores de pasos unipolares de cuatro fases utilizando el CI L702B(N).
- [16]. Niku B., S. 2001. Introduction to robotics: Analysis, Systems, Applications.Ed. Prentice Hall.USA.
- [17]. Pallas A., A. 2001. Sensores y acondicionadores de señales. Tercera ed. Alfaomega-marcombo. Barcelona, España.



- [18]. Pressman R., S. 2002. Ingeniería del software: un enfoque práctico. McGraw-Hill. 5ª edición. Madrid, España.
- [19]. Ramos R., G. 1998. Adquisición de datos por el puerto paralelo. Revista Electrónica & Computadoras Número 41. Publicaciones CEKIT, S. A. Buenos Aires, Argentina.
- [20]. Reisdorph, K. Delphi 4, version en español. Ed. Pearson-Prentice-Hall. México.
- [21]. Rubió S., A. 2000. Evasión de obstáculos mediante multicuerpos rígidos. Tesis de grado de Maestría en Ingeniería Mecánica, DEPMI, UNAM.
- [22]. Yao-Chon, C. and Mathukumalli, V. 1988. Optimal trajectory planning for planar n-link revolute manipulators in the presence of obstacles. IEEE.
- [23]. Reyes A., L. 1998. Sobre la parametrización de las rotaciones y reflexiones de multicuerpos rígidos en el plano. Universidad Anahúac del Sur. Estudios ocasionales.



Apéndice A

Planos de los elementos mecánicos
y fotografías del prototipo de robot de 2GDL

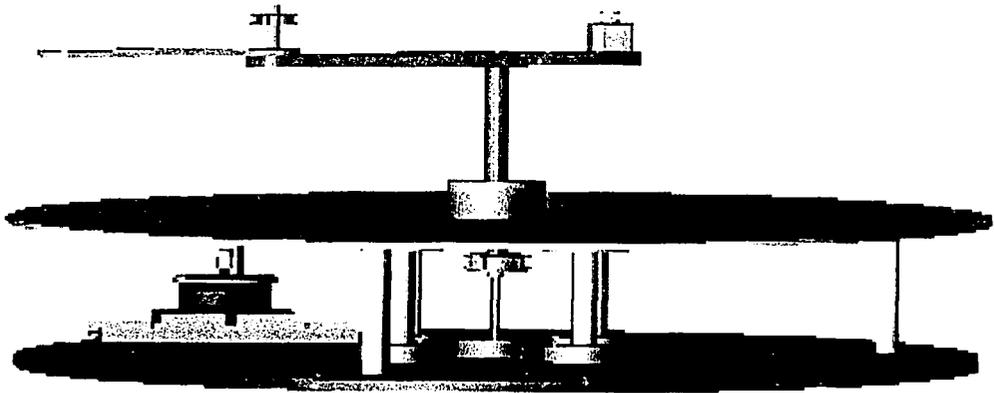
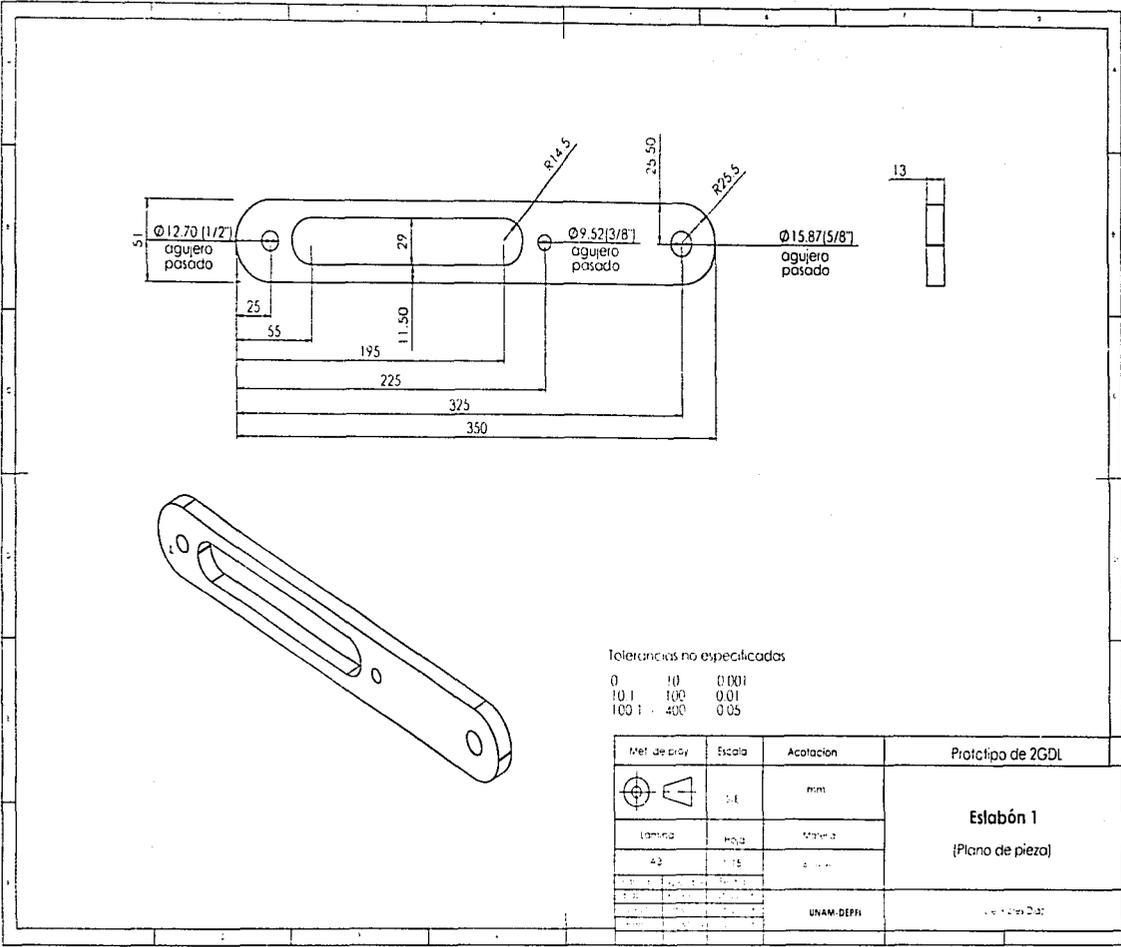


Figura A.1 Dibujo de ensamble en sólido del prototipo de robot de 2GDL.

TRABAJO CON
EQUIPO DE INVESTIGACIÓN

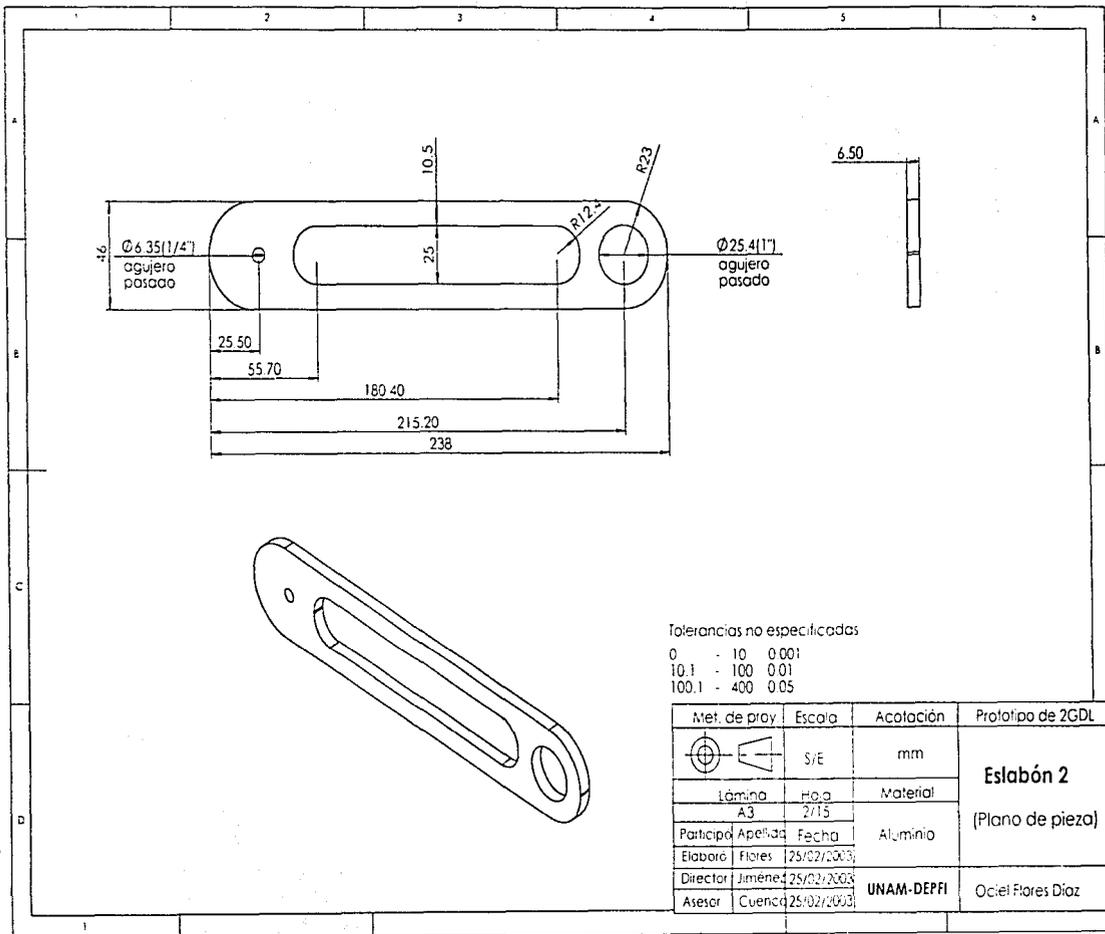
INSTITUTO TECNOLÓGICO DE GUATEMALA
 INSTITUTO TECNOLÓGICO DE GUATEMALA



Tolerancias no especificadas

0	10	0.001
10.1	100	0.01
100.1	400	0.05

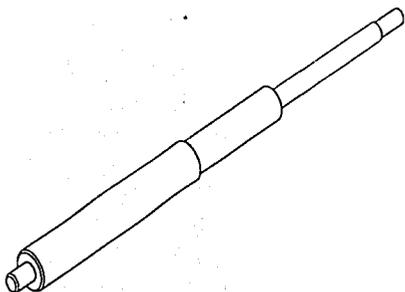
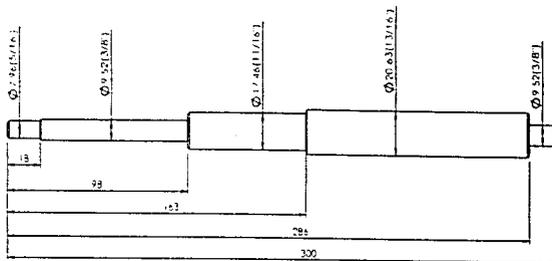
Met. de enq.	Escala	Anotacion	Prototipo de 2GDL
	1:1	mm	Eslabón 1 (Plano de pieza)
Laminado	H202	Módulo 2	
42	1.15	6.1.1.1	
UNAM-DEPH			Ver Anexo D.02



Tolerancias no especificadas

0 - 10 0.001
 10.1 - 100 0.01
 100.1 - 400 0.05

Met. de proy.	Escala	Acotación	Prototipo de 2GDL
	S/E	mm	Eslabón 2 (Plano de pieza)
Lamina A3	Hoj. 2/15	Material	
Participa	Apelada	Fecha	Aluminio
Elabora	Flores	25/02/2003	
Director	Jiménez	25/02/2003	UNAM-DEPFI
Asesor	Cuenca	25/02/2003	



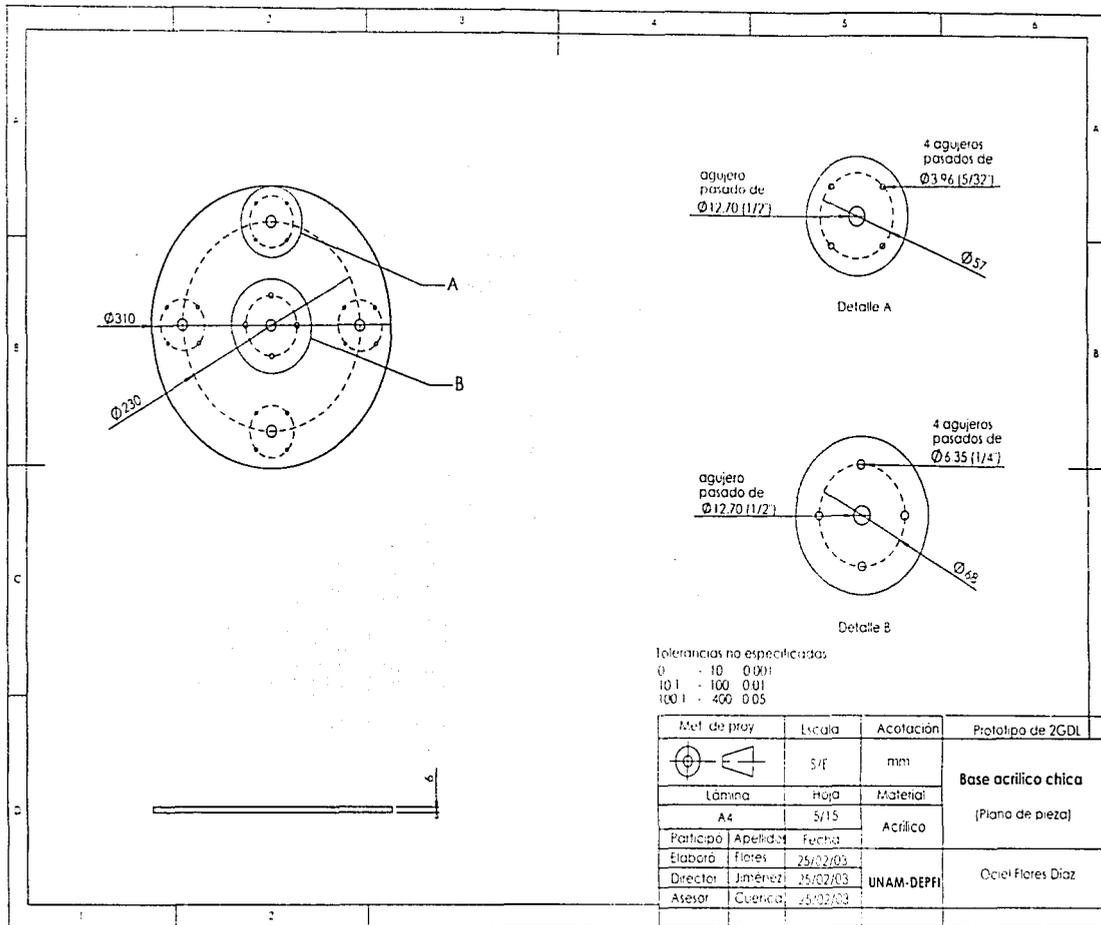
Tolerancias no especificadas:
 G: 0.010 0.001
 H9: 0.060 0.001
 H8: 0.040 0.001

Chatañas
 1 x 45°

Fundición de los diámetros del eje son re-ajustes las tolerancias

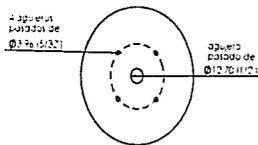
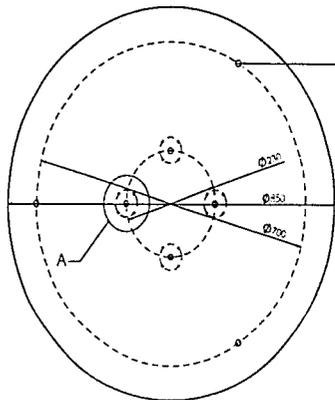
Med. de proy.	Escala	Acotación	Prototipo de 2GDL
	1:1	mm	Eje del eslabón 1 (Plano de pieza)
1:1	3/15	Aleación de aluminio	
1:1	3/15	Acero 42	UNAM-DEPH
1:1	3/15	Acero 42	
1:1	3/15	Acero 42	UNAM-DEPH

EJE DEL ESLOBÓN 1
 UNAM-DEPH



UNAM-DEPFI
 MIPRO-AC

N.º 100
 N.º 100



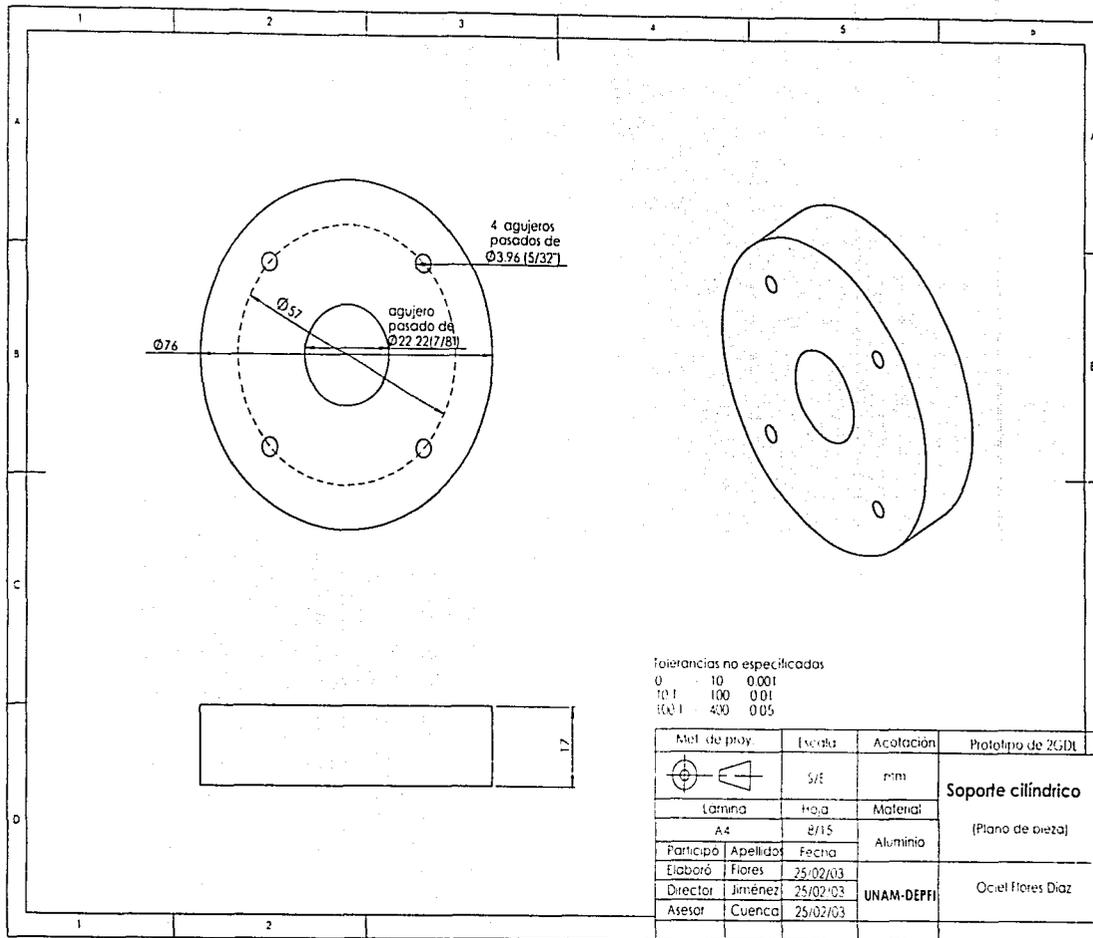
Detalle A

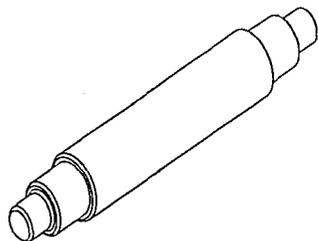
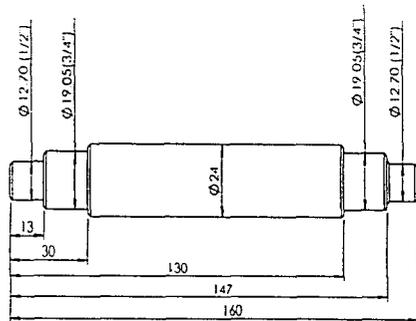
Dimensiones en milímetros

1:100
 1:100
 1:100



Ver. de dibujo	Escala	Proyector	Proyector de DGS
	1:100	1:100	
Camión	M.D.	M.D.	Base de madera
Rev. 1	1:100	1:100	1:100
Rev. 2	1:100	1:100	1:100
Rev. 3	1:100	1:100	1:100
Rev. 4	1:100	1:100	1:100
Rev. 5	1:100	1:100	1:100
Rev. 6	1:100	1:100	1:100
Rev. 7	1:100	1:100	1:100
Rev. 8	1:100	1:100	1:100
Rev. 9	1:100	1:100	1:100
Rev. 10	1:100	1:100	1:100
Rev. 11	1:100	1:100	1:100
Rev. 12	1:100	1:100	1:100
Rev. 13	1:100	1:100	1:100
Rev. 14	1:100	1:100	1:100
Rev. 15	1:100	1:100	1:100
Rev. 16	1:100	1:100	1:100
Rev. 17	1:100	1:100	1:100
Rev. 18	1:100	1:100	1:100
Rev. 19	1:100	1:100	1:100
Rev. 20	1:100	1:100	1:100
Rev. 21	1:100	1:100	1:100
Rev. 22	1:100	1:100	1:100
Rev. 23	1:100	1:100	1:100
Rev. 24	1:100	1:100	1:100
Rev. 25	1:100	1:100	1:100
Rev. 26	1:100	1:100	1:100
Rev. 27	1:100	1:100	1:100
Rev. 28	1:100	1:100	1:100
Rev. 29	1:100	1:100	1:100
Rev. 30	1:100	1:100	1:100
Rev. 31	1:100	1:100	1:100
Rev. 32	1:100	1:100	1:100
Rev. 33	1:100	1:100	1:100
Rev. 34	1:100	1:100	1:100
Rev. 35	1:100	1:100	1:100
Rev. 36	1:100	1:100	1:100
Rev. 37	1:100	1:100	1:100
Rev. 38	1:100	1:100	1:100
Rev. 39	1:100	1:100	1:100
Rev. 40	1:100	1:100	1:100
Rev. 41	1:100	1:100	1:100
Rev. 42	1:100	1:100	1:100
Rev. 43	1:100	1:100	1:100
Rev. 44	1:100	1:100	1:100
Rev. 45	1:100	1:100	1:100
Rev. 46	1:100	1:100	1:100
Rev. 47	1:100	1:100	1:100
Rev. 48	1:100	1:100	1:100
Rev. 49	1:100	1:100	1:100
Rev. 50	1:100	1:100	1:100
Rev. 51	1:100	1:100	1:100
Rev. 52	1:100	1:100	1:100
Rev. 53	1:100	1:100	1:100
Rev. 54	1:100	1:100	1:100
Rev. 55	1:100	1:100	1:100
Rev. 56	1:100	1:100	1:100
Rev. 57	1:100	1:100	1:100
Rev. 58	1:100	1:100	1:100
Rev. 59	1:100	1:100	1:100
Rev. 60	1:100	1:100	1:100
Rev. 61	1:100	1:100	1:100
Rev. 62	1:100	1:100	1:100
Rev. 63	1:100	1:100	1:100
Rev. 64	1:100	1:100	1:100
Rev. 65	1:100	1:100	1:100
Rev. 66	1:100	1:100	1:100
Rev. 67	1:100	1:100	1:100
Rev. 68	1:100	1:100	1:100
Rev. 69	1:100	1:100	1:100
Rev. 70	1:100	1:100	1:100
Rev. 71	1:100	1:100	1:100
Rev. 72	1:100	1:100	1:100
Rev. 73	1:100	1:100	1:100
Rev. 74	1:100	1:100	1:100
Rev. 75	1:100	1:100	1:100
Rev. 76	1:100	1:100	1:100
Rev. 77	1:100	1:100	1:100
Rev. 78	1:100	1:100	1:100
Rev. 79	1:100	1:100	1:100
Rev. 80	1:100	1:100	1:100
Rev. 81	1:100	1:100	1:100
Rev. 82	1:100	1:100	1:100
Rev. 83	1:100	1:100	1:100
Rev. 84	1:100	1:100	1:100
Rev. 85	1:100	1:100	1:100
Rev. 86	1:100	1:100	1:100
Rev. 87	1:100	1:100	1:100
Rev. 88	1:100	1:100	1:100
Rev. 89	1:100	1:100	1:100
Rev. 90	1:100	1:100	1:100
Rev. 91	1:100	1:100	1:100
Rev. 92	1:100	1:100	1:100
Rev. 93	1:100	1:100	1:100
Rev. 94	1:100	1:100	1:100
Rev. 95	1:100	1:100	1:100
Rev. 96	1:100	1:100	1:100
Rev. 97	1:100	1:100	1:100
Rev. 98	1:100	1:100	1:100
Rev. 99	1:100	1:100	1:100
Rev. 100	1:100	1:100	1:100





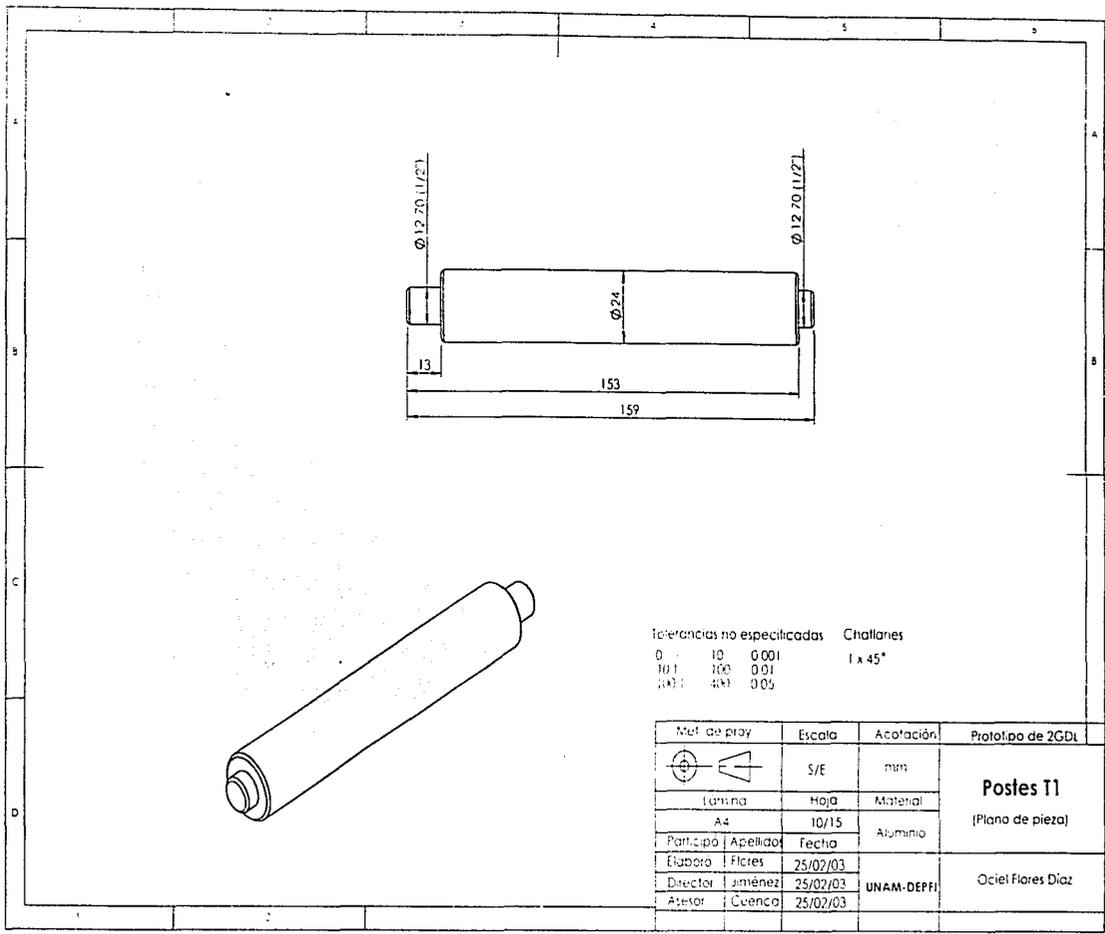
Tolerancias no especificadas

0 - 10 0.001
10.1 - 100 0.01
100.1 - 400 0.05

Chafilados no especificados

1 x 45°

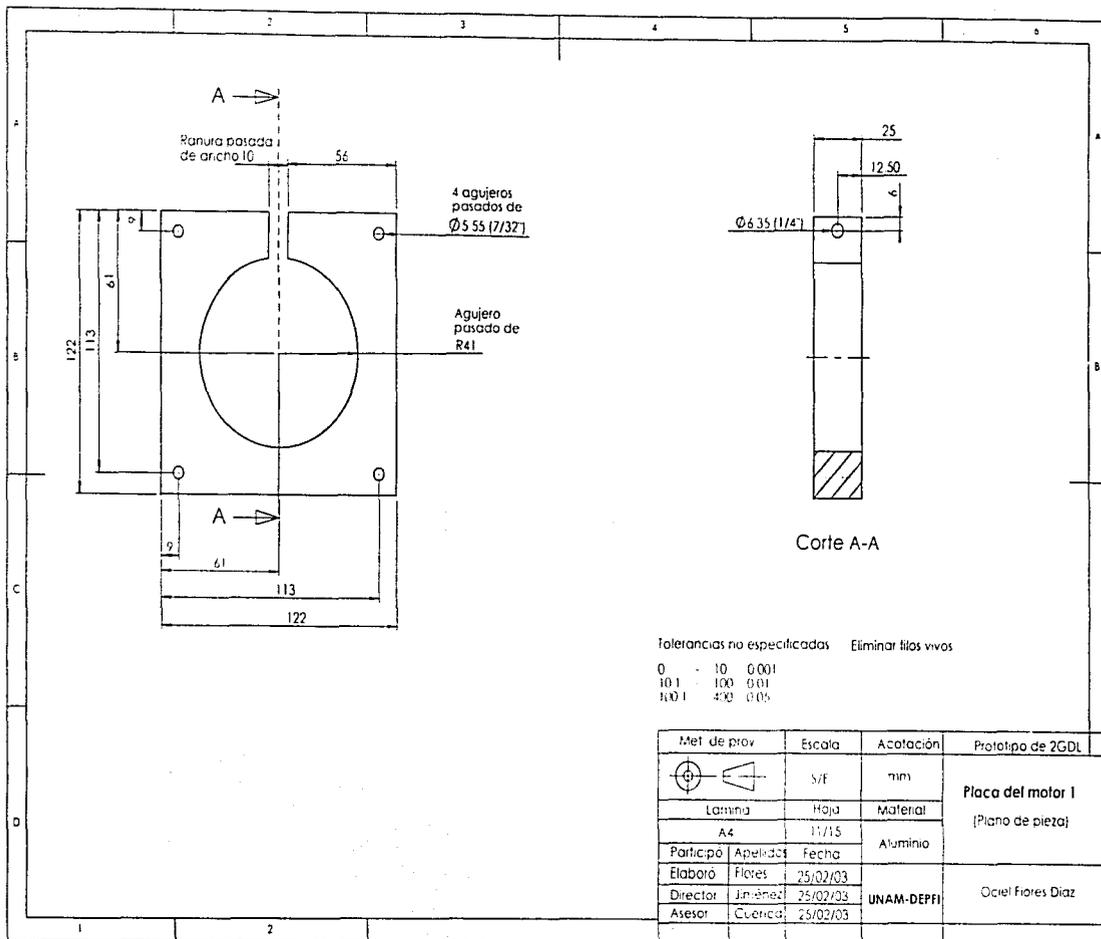
Mat. de proy	Escala	Acatación	Prototipo de 2GDL
	S/E	mm	Postes T2 (Plano de pieza)
Lamina	Hoja	Materia	
A4	9/15	Aluminio	
Participó	Apel. de	Fecha	
Elaboro	Flóres	25/02/03	
Director	J. Pérez	25/02/03	UNAM-DEPFI
Asesor	Cuenca	25/02/03	Ociel Flores Díaz



Tolerancias no especificadas: Challanes
 0 - 13 0.001 1 x 45°
 10 - 100 0.01
 100 - 400 0.05

Método de proy.	Escala	Acotación	Prototipo de 2GDL
	S/E	mm	Postes T1 (Plano de pieza)
Lamina	Hoja	Materia	
A4	10/15	Aluminio	
Participo Apellido	Fecha		
Elaboro Flores	25/02/03		
Director Jiménez	25/02/03	UNAM-DEPFI	Ociel Flores Díaz
Aseor Cuenco	25/02/03		

UNAM-DEPFI
 DEPARTAMENTO DE INGENIERIA EN METALURGIA
 LABORATORIO DE DISEÑO MECANICO
 CARRERA DE INGENIERIA EN METALURGIA
 UNAM-DEPFI



Panura pasada de ancho 10

4 agujeros pasados de $\varnothing 5.55 (7/32)$

Agujero pasado de R41

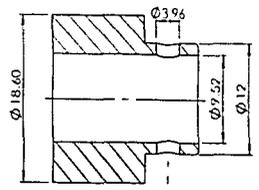
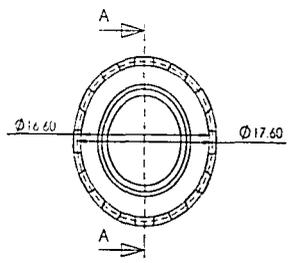
Corte A-A

Tolerancias no especificadas Eliminar filos vivos

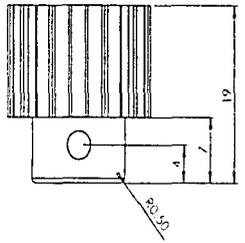
0	10	0.001
10.1	100	0.01
100.1	1000	0.05

Met. de prov.	Escala	Acotación	Prototipo de 2GDL
	S/F	mm	Placa del motor I [Plano de pieza]
Lamina	Hoja	Materia	
A4	11/15	Aluminio	
Participó	Apellidos	Fecha	
Elaboró	Flores	25/02/03	
Director	Jiménez	25/02/03	
Asesor	Cuenca	25/02/03	UNAM-DEPFI Ociel Flores Diaz

TRABAJO CON
 MUCHO CUIDADO



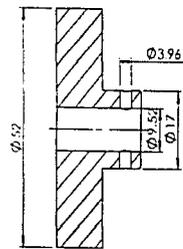
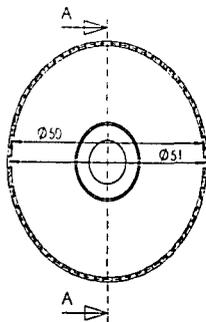
Corte A-A



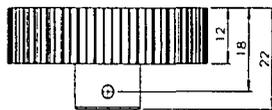
Tolerancias no dimensionadas Número de dientes 12
 0 - 10 - 0.001 Tipo de diente, recto
 10.1 - 100 - 0.01 Módulo 1.46
 100.1 - 400 - 0.02 Altura del diente 1
 Bisel de 2 x 45° Ancho del diente 2.44

Met. de piv.	Escala	Acotación	Prototipo de 2CD
	S/E	n/m	
Lamina	Hoja	Materia	Rueda dentada 1
A4	12/15	Aluminio	(Plano de pieza)
Participa	Apellido	Fecha	
Elaboró	Flores	25/02/03	
Diseñó	Simonez	25/02/03	
Aprobó	Simonez	25/02/03	UNAM-DEPFI
			Ociel Flores Diaz

MEDIDA CON
 NO. 100



Corte A-A



Tolerancias no dimensionadas

0 - 10 0.001
 10.1 - 100 0.01
 100.1 - 400 0.05
 Hojes de 2 x 45*

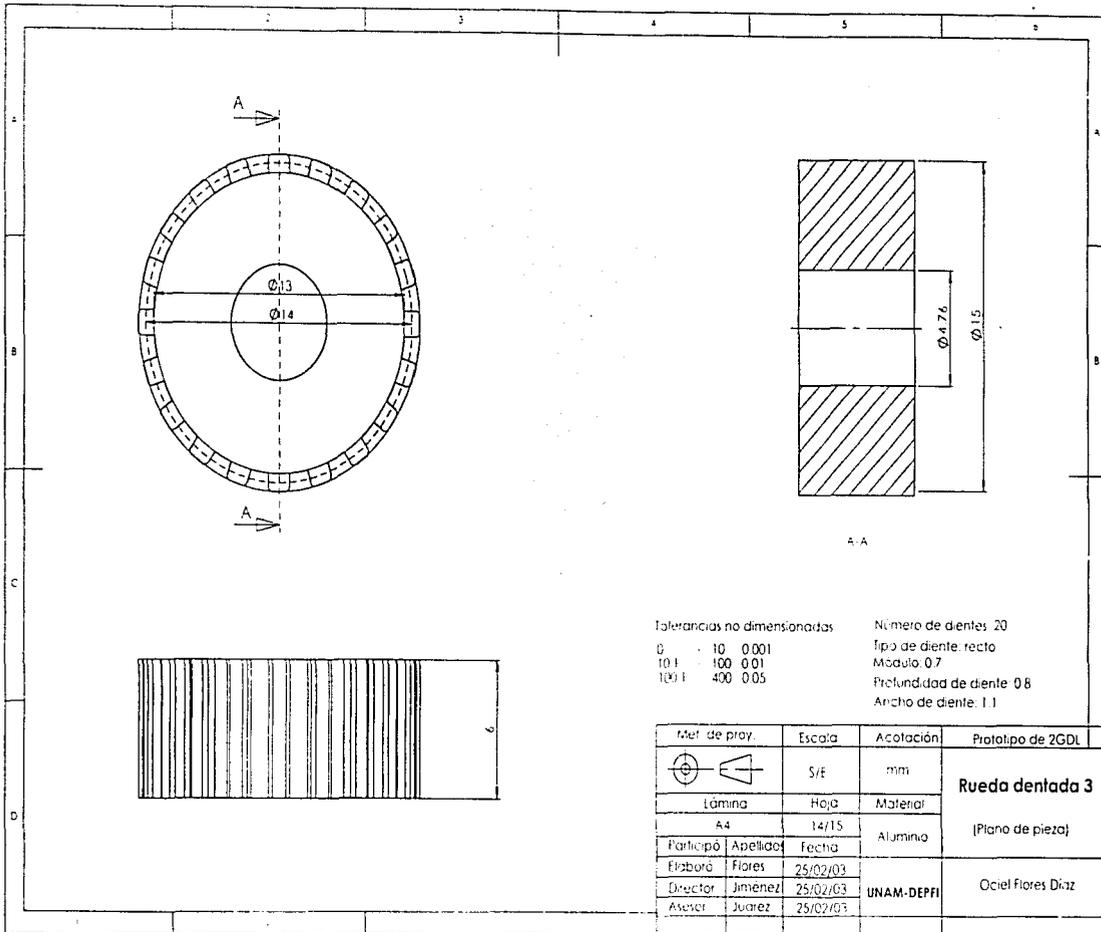
Número de dientes 36

Tipo de diente recto
 Módulo: 1.417
 Profundidad de diente: 1
 Ancho del diente: 2.44

Mét. de proy.	Escala	Acotación	Frotal po de 2GDI
	S/E	mm	Rueda dentada 2 (Plano de pieza)
Lamina	Hoja	Materia	
A 4	13/15	Aluminio	
Participó	Apellido	Fecha	
Elaboró	Flores	25/02/03	
Director	Lamanez	25/02/03	UNAM-DEPFI Oc. el Flores Diaz
Asesor	Juarez	25/02/03	

CON
 NEN

MA 1701
 TERCERA
 1981



Tolerancias no dimensionadas

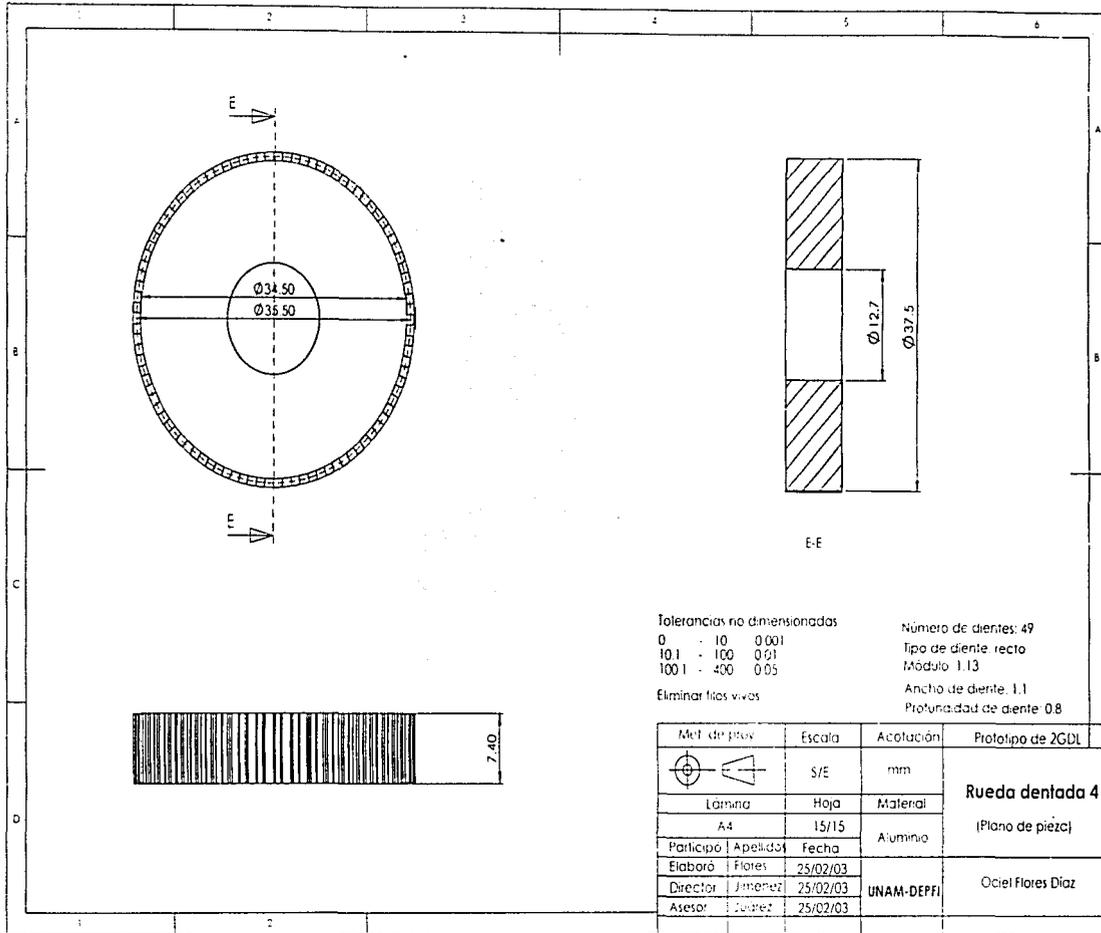
0 - 10 0.001
 10 - 100 0.01
 100 - 400 0.05

Número de dientes 20

Tipo de diente: recto
 Módulo 0.7
 Profundidad de diente 0.8
 Ancho de diente 1.1

Met. de proy.	Escala	Acotación	Prototipo de 2GDL
	S/8	mm	Rueda dentada 3
Lámina	Hoja	Material	
A4	14/15	Aluminio	(Plano de pieza)
Participó	Apellido	Fecha	UNAM-DEPI Ociel Flores Díaz
Elaboró	Flores	25/02/03	
Director	Jiménez	25/02/03	
Aveser	Juarez	25/02/03	

TESIS CON
 ORIGEN



Tolerancias no dimensionadas

0	- 10	0.001
10.1	- 100	0.01
100.1	- 400	0.05

Eliminar filos vivos

Número de dientes: 49

Tipo de diente: recto

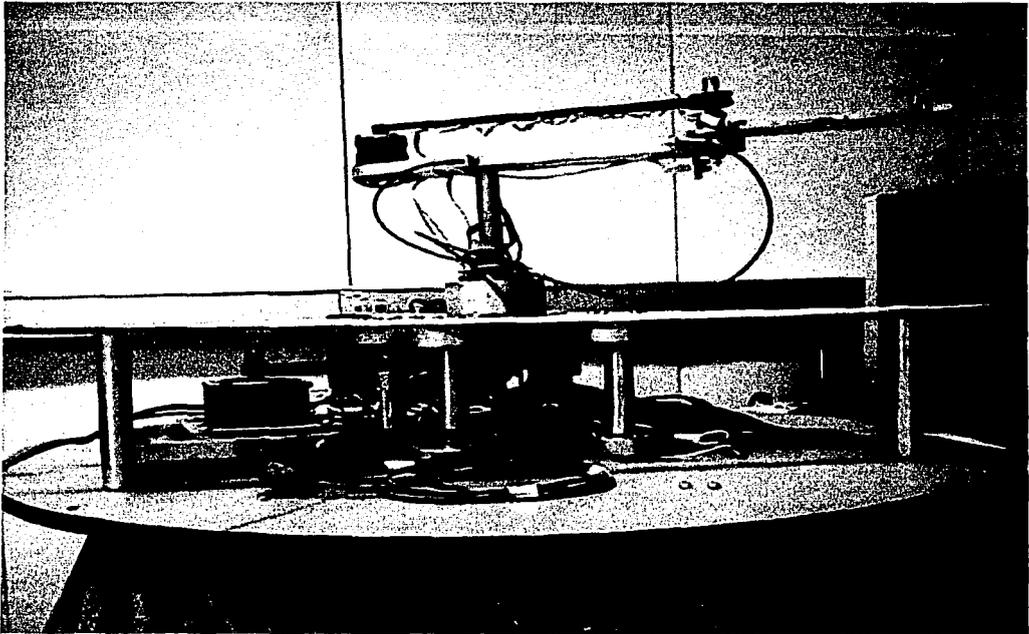
Módulo: 1.13

Ancho de diente: 1.1

Profundidad de diente: 0.8

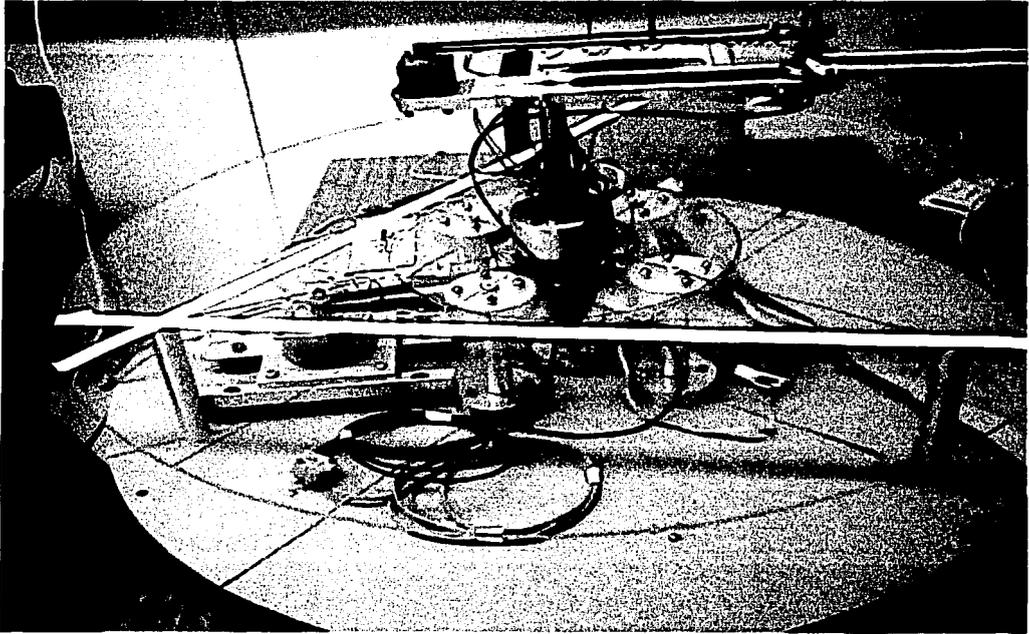
Met. de prov.	Escala	Acotación	Prototipo de 2GDL
	S/E	mm	Rueda dentada 4 (Plano de pieza)
Lámina	Hoja	Material	
A4	15/15	Aluminio	
Participó	Apellido	Fecha	
Elaboró	Flores	25/02/03	
Director	Jiménez	25/02/03	UNAM-DEPFI
Asexor	Jiménez	25/02/03	
			Ociel Flores Díaz

Fotografías del robot prototipo de 2GDL utilizado para la evasión de obstáculos en el plano



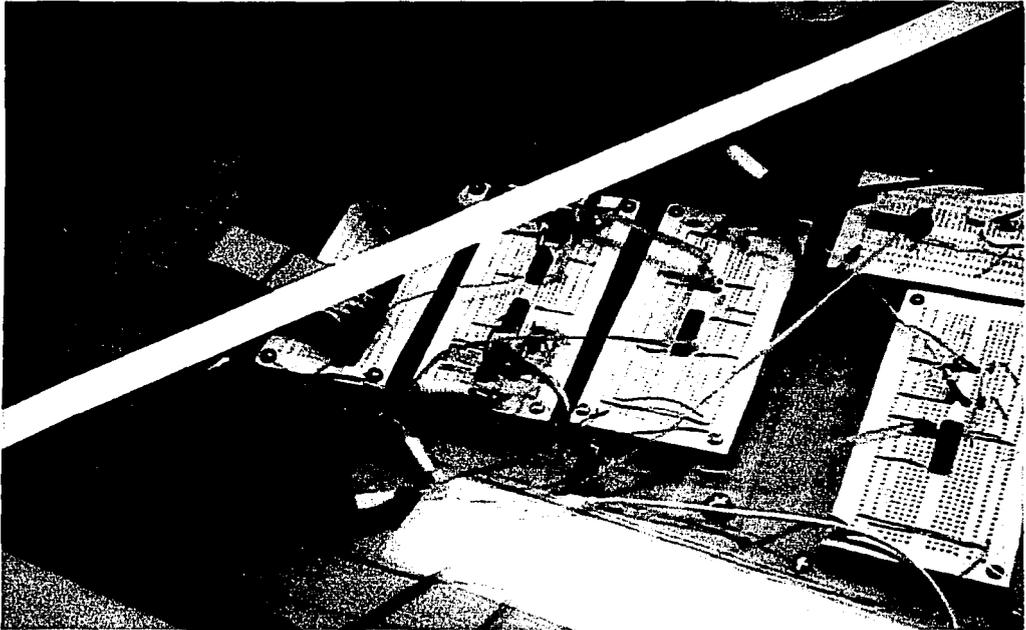
Fotografía A.1 Vista lateral del robot prototipo de 2GDL.

TRIPS CON
FABRICA DE ORIGEN



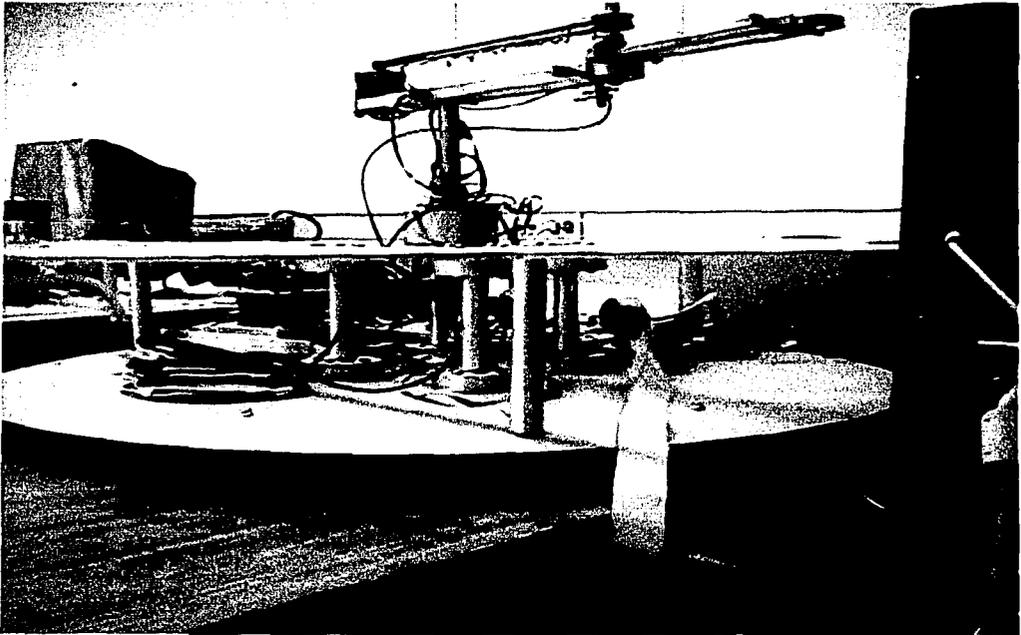
Fotografía A.2 Vista superior del robot prototipo de 2GDL, que muestra el sensor de barrera de luz roja con conexión de fibra óptica.

TESIS CON
FONDO DE ORIGEN



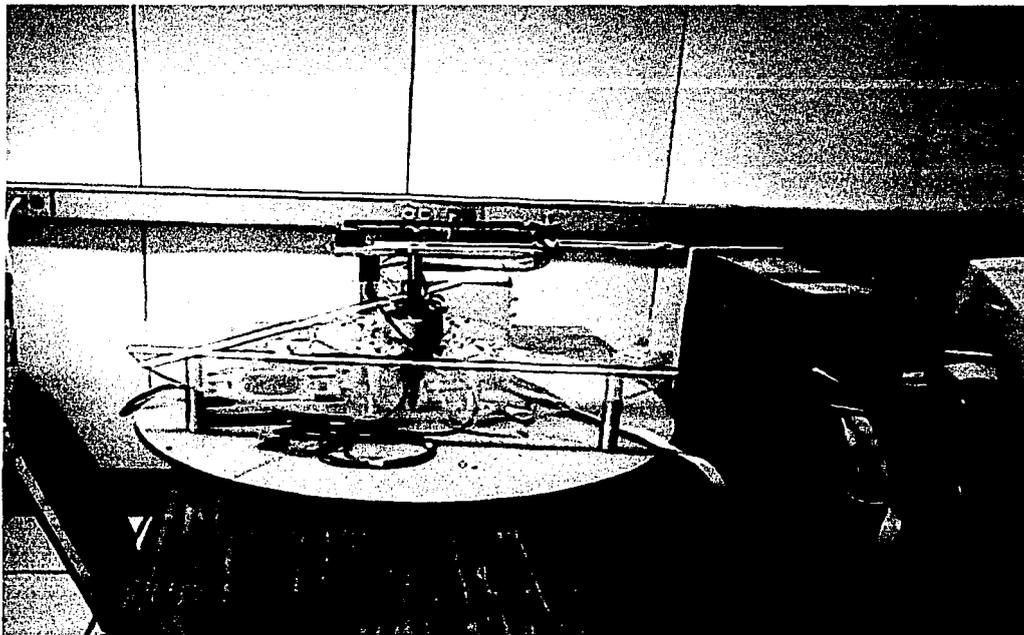
Fotografía A.3 Vista de los circuitos electrónicos utilizados en el robot prototipo de 2GDL.

TRISIS CON
FAN EN ORIGEN



Fotografía A.4 Vista lateral inferior del robot prototipo de 2GDL.

TESIS CON
FALSA DE ORIGEN



Fotografía A.5 Vista lateral superior del robot prototipo de 2GDL, que muestra la conexión al puerto paralelo externo de la computadora.

TESIS CON
FOLIO DE ORIGEN

Apéndice B

Código fuente del programa Evasoft 1.0

TESIS CON
FALLA DE CALIFICACION

CÓDIGO FUENTE DEL PROGRAMA EVASOFT 1.0

UNIDAD PRINCIPAL.

unit Principal_1;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls,
Forms, Dialogs,
Menus, ExtCtrls, StdCtrls, Buttons;

type

```
TPrincipal = class(TForm)
  MainMenu1: TMainMenu;
  Instruccionesdeusodelrobot1: TMenuItem;
  Hardware1: TMenuItem;
  Software1: TMenuItem;
  Programascontroladores1: TMenuItem;
  Controlareslabonesdemaneraindependiente1: TMenuItem;
  Realizacindetrayectoriasprogramadas1: TMenuItem;
  Controldelrobotentemporeal1: TMenuItem;
  Acercaadelprograma1: TMenuItem;
  Autor1: TMenuItem;
  Image1: TImage;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  Label7: TLabel;
  BitBtn1: TBitBtn;
  Label8: TLabel;
  Evasindeobstculos1: TMenuItem;
  Software2: TMenuItem;
  Salir2: TMenuItem;
  procedure FormActivate(Sender: TObject);
  procedure Salir1Click(Sender: TObject);
  procedure PositemHOME1Click(Sender: TObject);
  procedure Controlareslabonesdemaneraindependiente1Click(Sender: TObject);
  procedure Autor1Click(Sender: TObject);
  procedure Realizacindetrayectoriasprogramadas1Click(Sender: TObject);
  procedure Evasindeobstculos1Click(Sender: TObject);
  procedure Controldelrobotentemporeal1Click(Sender: TObject);
  procedure Salir2Click(Sender: TObject);
private
  ; Private declarations ;
public
  ; Public declarations ;
end;
```

var

Principal: TPrincipal;

implementation

uses Tiempo_real_1, Home_1, Mov_indepen_1, Acerca_de,
Tray_program_1, Tray_cont1;

;\$R * DFM;

```
procedure TPrincipal.FormActivate(Sender: TObject);
begin
  WindowState := wsMaximized;
end;
```

```
procedure TPrincipal.Salir1Click(Sender: TObject);
begin
  Application.Terminate;
end;
```

```
procedure TPrincipal.PositemHOME1Click(Sender: TObject);
begin
  Home.Show;
end;
```

```
procedure TPrincipal.Controlareslabonesdemaneraindependiente1Click(
  Sender: TObject);
begin
  Mov_indepen.Show;
end;
```

```
procedure TPrincipal.Autor1Click(Sender: TObject);
begin
  Acerca_de1.Show;
end;
```

```
procedure TPrincipal.Realizacindetrayectoriasprogramadas1Click(
  Sender: TObject);
begin
  Tray_programa.Show;
end;
```

```
procedure TPrincipal.Evasindeobstculos1Click(Sender:
  TObject);
begin
  Tiempo_real.Show;
end;
```

```
procedure TPrincipal.Controldelrobotentemporeal1Click(Sender:
  TObject);
begin
  Tray_con_show;
end;
```

```
procedure TPrincipal.Salir2Click(Sender: TObject);
begin
  Application.Terminate;
end;
```

UNIDAD DE MOVIMIENTOS UNO A UNO

unit Mov_indepen_1;

interface

uses

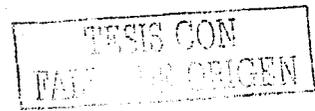
Windows, Messages, SysUtils, Classes, Graphics, Controls,
Forms, Dialogs, StdCtrls, ExtCtrls, Buttons,
Prende_motores, Graf_pantalla_inicial;

const

l1 = 84, l2 = 60;

type

```
TMov_indepen = class(TForm)
  Panel1: TPanel;
  GroupBox1: TGroupBox;
  Label1: TLabel;
  Edit1: TEdit;
  Label3: TLabel;
  Edit3: TEdit;
  Label4: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  Label7: TLabel;
  Label8: TLabel;
  Label9: TLabel;
  ComboBox1: TComboBox;
  Image1: TImage;
  Panel4: TPanel;
  BitBtn7: TBitBtn;
  BitBtn8: TBitBtn;
  BitBtn9: TBitBtn;
  Panel2: TPanel;
  BitBtn1: TBitBtn;
  BitBtn2: TBitBtn;
  Label2: TLabel;
  Label10: TLabel;
  procedure FormActivate(Sender: TObject);
```



```

procedure BitBtn7Click(Sender: TObject);
procedure ComboBox1Change(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
private
  { Private declarations }
Procedure Ejes_de_pantalla;
Procedure Posicion_de_inicio;
Procedure Primera_opcion;
Procedure Segunda_opcion;
Procedure Rotacion_de_los_eslabones(alfa_r,alfa_r2:real);
public
  { Public declarations }
end;
var
Mov_indepen TMov_indepen;
x,x1,y1,x2,y2:integer;
xx1,yy1,xx2,yy2:integer;
theta,alfa1,alfa2,theta_min,theta_elegida:real;
incr_alfa1,incr_alfa2:integer;
r:integer;sa1:real;
alfa_r:real;alfa_r2:real;sa2:real;
entrada,a,veces:integer;
opcion_final,opcion_1:integer;
p0,p1,q0,q1:real;
pp0,pp1,qq0,qq1:real;
M1_MA:Array[1..10,1..10] of Integer;
M1_MB:Array[1..4] of Integer=(3,2,0,1);
Function SLE(S:real) real;far,external'Funciones.dll';
Function COSE(N:real) real;far,external'Funciones.dll';
Function NPC(X:real) integer;far,external'Aux_graficacion.dll';
Function YPC(y:real) integer;far,external'Aux_graficacion.dll';
Procedure OutP(Puerto:word,data:byte);far,external'Port.dll';
Function Inp(Port:word) word;far,external'Port.dll';
implementation

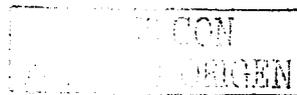
{SR * DFM1;
Procedure TMov_indepen Ejes_de_pantalla;
var ex1,ey1,ex2,ey2:integer;
begin
  Image1.Canvas.Pen.Color:=clgreen;
  Image1.Canvas.Pen.Width:=2;
  Image1.Canvas.Ellipse(40,40,370,370);
  ey1:=Round((Image1.Height)/2);
  ex1:=Image1.Width;
  Image1.Canvas.MoveTo(0,ey1);{eje de las abscisas}
  Image1.Canvas.LineTo(ex1,ey1);
  ex2:=Round((Image1.Width)/2);
  ey2:=Image1.Height;
  Image1.Canvas.MoveTo(ex2,0);
  Image1.Canvas.LineTo(ex2,ey2);
  Image1.Canvas.TextOut(373,210,'"X"');
  Image1.Canvas.TextOut(155,5,'"EJE: "Y"');
  Image1.Canvas.TextOut(5,210,'"X"');
  Image1.Canvas.TextOut(175,370,'"Y"');
end;
procedure TMov_indepen FormActivate(Sender: TObject);
const alfa1=0,alfa2=0;
begin
  WindowState:=wsMaximized;
  with Image1.Canvas do
  begin
    Brush.Color:=clwhite;
    Image1.Width:=400;
    Image1.Height:=400;
    Rectangle(0,0,Image1.Width,Image1.Height);
    Ejes_de_pantalla;
  end;
end;
end;

```

```

Procedure TMov_indepen.Posicion_de_inicio;
begin
  alfa_r:=0;
  alfa_r2:=0;
  {Definición de los parámetros}
  p0:=cosent(alfa_r); p1:=sent(alfa_r);
  q0:=cosent(alfa_r2); q1:=sent(alfa_r2);
  {borra eslabon 1;}
  Image1.Canvas.Pen.Width:=2;
  Image1.Canvas.Pen.Color:=clwhite;
  Image1.Canvas.MoveTo(200,200);
  Image1.Canvas.LineTo(xx1,yy1);
  {borra eslabon 2;}
  Image1.Canvas.Pen.Width:=2;
  Image1.Canvas.Pen.Color:=clwhite;
  Image1.Canvas.LineTo(xx1,yy1);
  Image1.Canvas.LineTo(xx2,yy2);
  xx1:=Round((1*p0+200));
  yy1:=Round((1*p1+200));
  xx2:=Round((1*p0+12*(p0*q0-p1*q1)+200));
  yy2:=Round((1*p1+12*(p0*q1+p1*q0)+200));
  {pinta eslabon 1 en posicion de home}
  Image1.Canvas.Pen.Width:=2;
  Image1.Canvas.Pen.Color:=clgreen;
  Image1.Canvas.MoveTo(200,200);
  Image1.Canvas.LineTo(xx1,yy1);
  Image1.Canvas.LineTo(xx2,yy1);
  {pinta eslabon 2 en posicion de home}
  Image1.Canvas.Pen.Width:=2;
  Image1.Canvas.Pen.Color:=clred;
  Image1.Canvas.LineTo(xx1,yy1);
  Image1.Canvas.LineTo(xx2,yy2);
  Image1.Refresh;
end;
Procedure
TMov_indepen Rotacion_de_los_eslabones(alfa_r,alfa_r2:real);
begin
  {Definición de los parámetros}
  pp0:=cosent(alfa_r); pp1:=sent(alfa_r);
  qq0:=cosent(alfa_r2); qq1:=sent(alfa_r2);
  {Rotacion de los eslabones con los parámetros;}
  xx1:=Round((1*pp0+200));
  yy1:=Round((1*pp1+200));
  xx2:=Round((1*pp0+12*(pp0*qq0-pp1*qq1)+200));
  yy2:=Round((1*pp1+12*(pp0*qq1+pp1*qq0)+200));
  {pinta el eslabon 1 corriendo}
  Image1.Canvas.Pen.Width:=2;
  Image1.Canvas.Pen.Color:=clgreen;
  Image1.Canvas.MoveTo(200,200);
  Image1.Canvas.LineTo(xx1,yy1);
  {pinta el eslabon 2 corriendo }
  Image1.Canvas.Pen.Width:=2;
  Image1.Canvas.Pen.Color:=clred;
  Image1.Canvas.LineTo(xx1,yy1);
  Image1.Canvas.LineTo(xx2,yy2);
  Image1.Refresh;
  sleep(1);
  Image1.Canvas.Pen.Width:=2;
  Image1.Canvas.Pen.Color:=clwhite;
  Image1.Canvas.MoveTo(200,200);
  Image1.Canvas.LineTo(xx1,yy1);
  Image1.Canvas.LineTo(xx1,yy1);
  Image1.Canvas.LineTo(xx2,yy2);
end;
end;

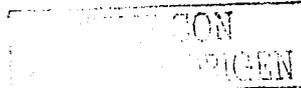
```



```

Procedure TMov_indepen.Primer_a_opcion;
var
  j:integer;
  incre_ang2:real;
begin
  alfa1:=StrToFloat(edit1.text);
  alfa2:=StrToFloat(edit3.text);
  if alfa1>=0 then
  begin
    repeat
      begin
        alfa_r:=alfa_r+0.60;
        Mover_motor1;
        Rotacion_de_los_eslabones(-alfa_r,-alfa_r2);
        Label8.caption:=FormatFloat('0.00',alfa_r);
        Label8.refresh;
      end;
    until alfa_r>=alfa1;
  end
  else
  repeat
  begin
    alfa_r:=alfa_r-0.60;
    Mover_motor1;
    Rotacion_de_los_eslabones(-alfa_r,-alfa_r2);
    Label8.caption:=FormatFloat('0.00',alfa_r);
    Label8.refresh;
  end;
  until alfa_r<=alfa1;
  {pinta posición final del eslabon 1}
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.Pen.Color:=clgreen;
  image1.Canvas.MoveTo(200,200);
  image1.Canvas.LineTo(x1,y1);
  {pinta posición final del eslabon 2}
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.Pen.Color:=clred;
  image1.Canvas.LineTo(x1,y1);
  image1.Canvas.LineTo(x2,y2);
  if alfa2>=0 then
  begin
    repeat
      begin
        alfa_r2:=alfa_r2+0.7228;
        Mover_motor2;
        Rotacion_de_los_eslabones(-alfa_r,-alfa_r2);
        Label9.Caption:=FormatFloat('0.00',alfa_r2);
        Label9.refresh;
      end;
    until alfa_r2>=alfa2;
  end
  else
  repeat
  begin
    alfa_r2:=alfa_r2-0.7228;
    Mover_motor2;
    Rotacion_de_los_eslabones(-alfa_r,-alfa_r2);
    Label9.Caption:=FormatFloat('0.00',alfa_r2);
    Label9.refresh;
  end;
  until alfa_r2<=alfa2;
  {pinta posición final del eslabon 1}
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.Pen.Color:=clgreen;
  image1.Canvas.MoveTo(200,200);
  image1.Canvas.LineTo(x1,y1);
  {pinta posición final del eslabon 2}
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.Pen.Color:=clred;
  image1.Canvas.LineTo(x1,y1);
  image1.Canvas.LineTo(x2,y2);
  if alfa2>=0 then
  begin
    repeat
      begin
        alfa_r:=alfa_r+0.60;
        Mover_motor1;
        Rotacion_de_los_eslabones(-alfa_r,-alfa_r2);
        Label8.caption:=FormatFloat('0.00',alfa_r);
        Label8.refresh;
      end;
    until alfa_r>=alfa1;
  end
  else
  repeat
  begin
    alfa_r:=alfa_r-0.60;
    Mover_motor1;
    Rotacion_de_los_eslabones(-alfa_r,-alfa_r2);
    Label8.caption:=FormatFloat('0.00',alfa_r);
    Label8.refresh;
  end;
  until alfa_r<=alfa1;
  {pinta posición final del eslabon 1}
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.LineTo(x2,y2);
  Label2.Caption:=El robot se encuentra en POSICION FINAL;
  Label10.Font.Color:=clMaroon;
  Label10.Caption:=Para reiniciar haz click en "Posición de inicio";
end;
Procedure TMov_indepen.Segunda_opcion;
var
  j:integer;
  incre_ang2:real;
begin
  alfa1:=StrToFloat(edit1.text);
  alfa2:=StrToFloat(edit3.text);
  if alfa2<=0 then
  begin
    repeat
      begin
        alfa_r2:=alfa_r2+0.7228;
        Mover_motor2;
        Rotacion_de_los_eslabones(-alfa_r,-alfa_r2);
        Label9.Caption:=FormatFloat('0.00',alfa_r2);
        Label9.refresh;
      end;
    until alfa_r2>=alfa2;
  end
  else
  repeat
  begin
    alfa_r2:=alfa_r2-0.7228;
    Mover_motor2;
    Rotacion_de_los_eslabones(-alfa_r,-alfa_r2);
    Label9.Caption:=FormatFloat('0.00',alfa_r2);
    Label9.refresh;
  end;
  until alfa_r2<=alfa2;
  {pinta posición final del eslabon 1}
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.Pen.Color:=clgreen;
  image1.Canvas.MoveTo(200,200);
  image1.Canvas.LineTo(x1,y1);
  {pinta posición final del eslabon 2}
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.Pen.Color:=clred;
  image1.Canvas.LineTo(x1,y1);
  image1.Canvas.LineTo(x2,y2);
  if alfa2>=0 then
  begin
    repeat
      begin
        alfa_r:=alfa_r+0.60;
        Mover_motor1;
        Rotacion_de_los_eslabones(-alfa_r,-alfa_r2);
        Label8.caption:=FormatFloat('0.00',alfa_r);
        Label8.refresh;
      end;
    until alfa_r>=alfa1;
  end
  else
  repeat
  begin
    alfa_r:=alfa_r-0.60;
    Mover_motor1;
    Rotacion_de_los_eslabones(-alfa_r,-alfa_r2);
    Label8.caption:=FormatFloat('0.00',alfa_r);
    Label8.refresh;
  end;
  until alfa_r<=alfa1;
  {pinta posición final del eslabon 1}
  image1.Canvas.Pen.Width:=2;

```



```

image1.Canvas.Pen.Color:=clgreen;
image1.Canvas.MoveTo(200,200);
image1.Canvas.LineTo(xx1,yy1);
{pinta posicion final del eslabon 2}
image1.Canvas.Pen.Width:=2;
image1.Canvas.Pen.Color:=clred;
image1.Canvas.LineTo(xx1,yy1);
image1.Canvas.LineTo(xx2,yy2);
Label2.Caption:='El robot se encuentra en POSICION
FINAL.';
Label10.Font.Color:=clMaroon;
Label10.Caption:='Para reiniciar haz click en "Posición de
micro".';
end;
procedure TForm1.Mov_indepen.BitBtn7Click(Sender: TObject);
begin
Label8.Caption:='';
Label9.Caption:='';
veces:=1;
repeat
begin
Mover_motor1;
Sleep(10);
entrada:=lnp($379);
end;
until entrada=247;
veces:=1;
repeat
begin
Mover_motor2;
Sleep(10);
entrada:=lnp($379);
end;
until entrada=231;
Posicion_de_micro;
end;
procedure TForm1.Mov_indepen.ComboBox1Change(Sender:
TObject);
begin
Case ComboBox1.ItemIndex of
0:opcion_1:=0;
1:opcion_1:=1;
end;
opcion_final:=opcion_1;
end;
procedure TForm1.Mov_indepen.BitBtn1Click(Sender: TObject);
begin
sa1:=0;
sa2:=0;
if opcion_final=0 then
begin
Primera_opcion;
end;
else
begin
Segunda_opcion;
end;
end;
end;
end;

```

UNIDAD DE MOVIMIENTOS SIMULTÁNEOS E HÍBRIDOS.

unit Tray_programa_1;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls,
Forms, Dialogs,

StdCtrls, Buttons, ExtCtrls, Prende_motores, math;

const

TI:=84.12/60;

type

TTray_programa = class(TForm)

Panel1: TPanel, GroupBox1: TGroupBox, Panel2: TPanel;

BitBtn3: TBitBtn, Label1: TLabel, Edit1: TEdit, Label2:

TLabel, Edit2: TEdit;

Image1: TImage, BitBtn4: TBitBtn; Label8: TLabel;

Label10: TLabel, Label3: TLabel;

Label4: TLabel, Label9: TLabel, Label6: TLabel, Label7:

TLabel, GroupBox2: TGroupBox;

Label5: TLabel, Edit3: TEdit, Label11: TLabel, Edit4:

TEdit, Label12: TLabel, Label13: TLabel;

Label14: TLabel, Label15: TLabel, Label16: TLabel, Panel3:

TPanel, BitBtn5: TBitBtn, BitBtn6: TBitBtn;

Panel4: TPanel, BitBtn7: TBitBtn; BitBtn8: TBitBtn;

Label17: TLabel, Label18: TLabel;

procedure FormActivate(Sender: TObject);

procedure BitBtn1Click(Sender: TObject);

procedure BitBtn2Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure BitBtn7Click(Sender: TObject);

procedure BitBtn5Click(Sender: TObject);

private

{ Private declarations }

Procedure Ejes_de_pantalla;

Procedure Posicion_de_micro;

Procedure Pnta_posicion_final;

Procedure Rotacion_de_los_eslabones(alfa_r, alfa_r2: real);

Procedure Graficacion_de_los_eslabones;

Procedure Arreglo_bits(M, N: integer);

public

{ Public declarations }

end;

var

Tray_programa: TTray_programa;

xx1, yy1, xx2, yy2: integer;

xx1, yy1, xx2, yy2: integer;

alfa1, alfa2, theta_min, theta_elegida: real;

mere_alfa1, mere_alfa2: integer;

r: integer;

mere_ang1, mere_ang2: real;

alfa_r: real, alfa_r2: real;

entrada, a, veces: integer;

opcion_final, opcion1: integer;

relacion0, relacion1, relacion2, relacion3, relacion4: real;

p0, p1, q0, q1: real;

pp0, pp1, qq0, qq1: real;

Function SIN(x: real): real; far; external 'Funciones.dll';

Function COSEN(x: real): real; far; external 'Funciones.dll';

Function

Procedure OutP(Puerto: word; dato: byte); far; external 'Port.dll';

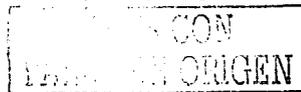
Function Inp(Port: word): word; far; external 'Port.dll';

implementation

{ \$R * DFM }

Procedure TTray_programa.Ejes_de_pantalla;

var ex1, ey1, ex2, ey2: integer;



```

begin
  image1.Canvas.Pen.Color:=clgreen;
  image1.Canvas.Pen.Width:=2;
  Image1.Canvas.Ellipse(40,40,370,370);
  ey1:=Round(image1.height/2);
  ex1:=image1.Width;
  image1.Canvas.MoveTo(0,ey1);{eje de las abscisas}
  image1.Canvas.LineTo(ex1,ey1);
  ex2:=Round(image1.width/2);
  ey2:=image1.height;
  image1.Canvas.MoveTo(ex2,0);
  image1.Canvas.LineTo(ex2,ey2);
  Image1.Canvas.TextOut(373,210,"X");
  Image1.Canvas.TextOut(155,5,4,4,"Y");
  Image1.Canvas.TextOut(5,210,"X");
  Image1.Canvas.TextOut(175,370,"Y");
end;
procedure TTray_programa.FormActivate(Sender: TObject);
const alfa1:=0;alfa2:=0;
begin
  WindowState:=wsMaximized;
  with Image1.Canvas do
    begin
      Brush.Color:=clwhite;
      image1.Width:=400;
      image1.Height:=400;
      Rectangle(0, 0, Image1.Width, Image1.Height);
      Ejes_de_pantalla;
    end;
  end;
Procedure
TTray_programa.Rotacion_de_los_eslabones(alfa_r,alfa_r2:re
al);
begin
  {Definicion de los parametros;
  pp0:=cosen(alfa_r); pp1:=sen(alfa_r);
  qq0:=cosen(alfa_r2); qq1:=sen(alfa_r2);
  {Rotacion de los eslabones con los parámetros}
  xx1:=Round(11*pp0-200);
  yy1:=Round(11*pp1-200);
  xx2:=Round(11*pp0-12*(pp0*qq0-pp1*qq1)+200);
  yy2:=Round(11*pp1-12*(pp0*qq1+pp1*qq0)+200);
  {pinta el eslabon 1;
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.Pen.Color:=clgreen;
  image1.Canvas.MoveTo(200,200);
  image1.Canvas.LineTo(xx1,yy1);
  {pinta el eslabon 2;
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.Pen.Color:=clred;
  image1.Canvas.LineTo(xx1,yy1);
  image1.Canvas.LineTo(xx2,yy2);
  image1.Refresh;
  sleep(1);
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.Pen.Color:=clwhite;
  image1.Canvas.MoveTo(200,200);
  image1.Canvas.LineTo(xx1,yy1);
  image1.Canvas.LineTo(xx1,yy1);
  image1.Canvas.LineTo(xx2,yy2);
  image1.Canvas.LineTo(xx2,yy2);
  end;
Procedure TTray_programa.Arreglo_bits(M,N:integer);
var a:array[0..M-1,0..N-1] of integer;
    indice1, indice2, indice: integer;
    delta real;
begin
  M[1][1]:=3;M[1][2]:=2;M[1][3]:=0;M[1][4]:=1;
  M[2][1]:=20;M[2][2]:=4;M[2][3]:=0;M[2][4]:=16;
  incre_ang2:=(incre_ang2+alfa2/alfa1)*0.6;
  incre_ang1:=(incre_ang1+alfa1/alfa2)*0.734244);
  if (M<0) then
    begin
      M[1][2]:=1;
      M[1][4]:=2;
      M:=-M;
    end;
  if (N<0) then
    begin
      M[2][2]:=16;
      M[2][4]:=4;
      N:=-N;
    end;
  MMax:=max(m,n);
  delta:=m/n;
  if (delta>1) then delta:=1.0/delta;
  for i:=1 to MMax do
    begin
      indice1:=(i-1) mod 4)+1;
      indice2:=round(delta*i);
      indice2:=indice2 mod 4;
      indice2:=indice2-1;
    end;
  if (i=m) then
    begin
      indice:=indice2;
      indice2:=indice1;
      indice1:=indice;
    end;
  a:=M1[indice1]+M2[indice2];
  Outp($378,a);
  Graficacion_de_los_eslabones;
  end;
  Outp($378,0);
  Pinta_posicion_final;
end;
Procedure TTray_programa.pinta_posicion_final;
begin
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.Pen.Color:=clgreen;
  image1.Canvas.MoveTo(200,200);
  image1.Canvas.LineTo(xx1,yy1);
  {pinta posicion final del eslabon 2}
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.Pen.Color:=clred;
  image1.Canvas.LineTo(xx1,yy1);
  image1.Canvas.LineTo(xx2,yy2);
  Label8.Font.Color:=clMaroon;
  Label8.Caption:="El robot se encuentra en POSICION
FINAL";
  Label10.Font.Color:=clMaroon;
  Label10.Caption:="Para reiniciar haz click en "Posición de
inicio";
end;
Procedure TTray_programa.Graficacion_de_los_eslabones;
begin
  if alfa1>alfa2 then begin
    if alfa1>0 then begin
      alfa_r:=alfa_r+0.6;
      alfa_r2:=alfa_r2+incre_ang2;
      Rotacion_de_los_eslabones(-alfa_r,-alfa_r2);
      Label6.Caption:=Format(float('0.00',alfa_r);
      Label6.refresh;
      Label7.Caption:=Format(float('0.00',alfa_r2);
      Label7.refresh;
      sleep(10);
    end
  else begin
    alfa_r:=alfa_r-0.6;
    alfa_r2:=alfa_r2-incre_ang2;
    Rotacion_de_los_eslabones(-alfa_r,-alfa_r2);

```

```

Label6.Caption :=FormatFloat('0.00',alfa_r);
Label6.Refresh;
Label7.Caption :=FormatFloat('0.00',alfa_r2);
Label7.Refresh;
sleep(10);
end;
end;
else begin
if alfa2 > 0 then begin
alfa_r :=alfa_r+merc_ang1;
alfa_r2 :=alfa_r2+0.734244;
Rotacion_de_los_eslabones(-alfa_r,alfa_r2);
Label6.Caption :=FormatFloat('0.00',alfa_r);
Label6.Refresh;
Label7.Caption :=FormatFloat('0.00',alfa_r2);
Label7.Refresh;
sleep(10);
end;
else begin
alfa_r :=alfa_r-merc_ang1;
alfa_r2 :=alfa_r2-0.734244;
Rotacion_de_los_eslabones(alfa_r,alfa_r2);
Label6.Caption :=FormatFloat('0.00',alfa_r);
Label6.Refresh;
Label7.Caption :=FormatFloat('0.00',alfa_r2);
Label7.Refresh;
sleep(10);
end;
end;
end;
end;
end;

```

```

Procedure TTray_programa.Posicion_de_inicio;
begin
alfa_r:=0;
alfa_r2:=0;
{Definición de los parametros}
p0:=cos(alfa_r); p1:=sen(alfa_r);
q0:=cos(alfa_r2); q1:=sen(alfa_r2);
{borra eslabon 1}
image1.Canvas.Pen.Width:=2;
image1.Canvas.Pen.Color:=clwhite;
image1.Canvas.MoveTo(200,200);
image1.Canvas.LineTo(x1,y1);
{borra eslabon 2}
image1.Canvas.Pen.Width:=2;
image1.Canvas.Pen.Color:=clwhite;
image1.Canvas.LineTo(x1,y1);
image1.Canvas.LineTo(x2,y2);
x1:=Round((1+p0)*200);
y1:=Round((1+p1)*200);
x2:=Round((1+p0)*2*(p0*q0-p1*q1)+200);
y2:=Round((1+p1)*2*(p0*q1+p1*q0)+200);
{pinta eslabon 1 en posicion de inicio}
image1.Canvas.Pen.Width:=2;
image1.Canvas.Pen.Color:=clgreen;
image1.Canvas.MoveTo(200,200);
image1.Canvas.LineTo(x1,y1);
image1.Canvas.LineTo(x1,y1);
{pinta eslabon 2 en posicion de inicio}
image1.Canvas.Pen.Width:=2;
image1.Canvas.Pen.Color:=clred;
image1.Canvas.LineTo(x1,y1);
image1.Canvas.LineTo(x2,y2);
image1.Refresh;
end;
end;

```

```

procedure TTray_programa.BitBtn4Click(Sender: TObject);
begin
a:=3;
contador:=1;
repeat
begin
Mover_motor1;
Sleep(10);
entrada:=Inp($379);
end;
until entrada=247;
a:=20;
veces:=1;
repeat
begin
Mover_motor2;
Sleep(10);
entrada:=Inp($379);
end;
until entrada=231;
Posicion_de_inicio;
Label8.Font.Color:=clgreen;
Label8.Caption:='El robot se encuentra en POSICION
INICIAL';
end;
procedure TTray_programa.BitBtn2Click(Sender: TObject);
begin
Posicion_de_inicio;
end;
end;

```

```

procedure TTray_programa.Button1Click(Sender: TObject);
begin
alfa1:=StrToFloat(edit1.text);
alfa2:=StrToFloat(edit2.text);
Arreglo_bits(trunc(alfa1/1.8*3.0),trunc(alfa2/1.8*2.4515));
end;
end;

```

```

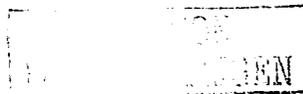
procedure TTray_programa.BitBtn7Click(Sender: TObject);
begin
alfa1:=StrToFloat(edit1.text);
alfa2:=StrToFloat(edit2.text);
Arreglo_bits(trunc(alfa1/1.8*3.0),trunc(alfa2/1.8*2.4515));
end;
end;

```

```

procedure TTray_programa.BitBtn5Click(Sender: TObject);
var
j:integer;
begin
alfa1:=StrToFloat(edit3.text);
alfa2:=StrToFloat(edit4.text);
if alfa1 <= alfa2 then
begin
theta_min:=alfa1;
end;
else
begin
theta_min:=alfa2;
end;
repeat
begin
alfa_r:=alfa_r+0.6;
Mover_motor_1_y_2;
alfa_r2:=alfa_r2+0.6;
Rotacion_de_los_eslabones(-alfa_r,alfa_r2);
Label13.caption:=FormatFloat('0.00',alfa_r);
Label13.Refresh;
Label14.caption:=FormatFloat('0.00',alfa_r2);
Label14.Refresh;
end;
until alfa_r:=theta_min;
end;
end;

```



```

if alfa1 > alfa2 then
begin
repeat
begin
theta_elegida := alfa1;
alfa_r := alfa_r + 0.6;
alfa_r2 := alfa_r2;
Mover_motor1;
Rotacion_de_los_eslabones(-alfa_r,-alfa_r2);
Label13.Caption := FormatFloat('0.00',alfa_r);
Label13.Refresh;
end;
until alfa_r <= theta_elegida;
end;
else
begin
repeat
begin
theta_elegida := alfa2;
alfa_r := alfa_r;
alfa_r2 := alfa_r2 + 0.7228;
Mover_motor2;
Rotacion_de_los_eslabones(alfa_r,alfa_r2);
Label14.Caption := FormatFloat('0.00',alfa_r2);
Label14.Refresh;
end;
until alfa_r2 <= theta_elegida;
end;
{Pinta posicion final del eslabon 1}
Image1.Canvas.Pen.Width := 2;
Image1.Canvas.Pen.Color := clGreen;
Image1.Canvas.MoveTo(200,200);
Image1.Canvas.LineTo(x1,y1);
{Pinta posicion final del eslabon 2}
Image1.Canvas.Pen.Width := 2;
Image1.Canvas.Pen.Color := clRed;
Image1.Canvas.LineTo(x1,y1);
Image1.Canvas.LineTo(x2,y2);
Label17.Caption := 'El robot se encuentra en POSICION FINAL';
Label18.Font.Color := clMaroon;
Label18.Caption := 'Para reiniciar haz click en "Posición de inicio"';
end;

```

end

UNIDAD DE MOVIAIENTOS SIMULTÁNEOS.

```
unit Tray_cont1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls,
Forms, Dialogs,
```

```
StdCtrls, Buttons, ExtCtrls, Prende_motores, Resuelve_NR;
```

```
const
```

```
l11 := 84,122 > 60;
```

```
type
```

```
TTray_con = class(TForm)
```

```
Panel1: TPanel; BitBtn1: TBitBtn; BitBtn3: TBitBtn; Panel2:
TPanel;
```

```
BitBtn4: TBitBtn; GroupBox1: TGroupBox; Label1:
```

```
TLabel; Edit1: TEdit; Label2: TLabel;
```

```
Label3: TLabel; Label4: TLabel; Edit2: TEdit; Edit3:
```

```
TEdit; Edit4: TEdit; Label5: TLabel;
```

```
Edit5: TEdit; Label6: TLabel; Edit6: TEdit; Label7:
```

```
TLabel; Edit7: TEdit;
```

```
Image1: TImage;
```

```
BitBtn5: TBitBtn;
```

```
BitBtn2: TBitBtn;
```

```
Memor1: TMemo;
```

```
procedure TForm1.Activate(Sender: TObject);
```

```
procedure BitBtn5.Click(Sender: TObject);
```

```
procedure BitBtn1.Click(Sender: TObject);
```

```
procedure BitBtn2.Click(Sender: TObject);
```

```
private
```

```
{ Private declarations }
```

```
Procedure Ejes de pantalla;
```

```
Procedure Posicion de inicio;
```

```
Procedure Pinta posicion final;
```

```
Procedure Rotacion_del_eslabon_1(alfa_r,alfa_r2:real);
```

```
Procedure Pinta posicion de avance;
```

```
Procedure Pinta_recta;
```

```
public
```

```
{ Public declarations }
```

```
end;
```

```
var
```

```
Tray_con: TTray_con;
```

```
otx,oty,l1,l2: integer;
```

```
{ Variables de la ecuación de la recta }
```

```
Pil,Pll: real;
```

```
Xc,Xl,Yc,Yl: real; mcre: real;
```

```
Xc,Yc: integer;
```

```
veces: integer;
```

```
{ Variables para dibujar la posicion de inicio }
```

```
xs1,ys1,xs2,ys2: integer;
```

```
theta_1,alfa_1: integer;
```

```
Alfa_1,Alfa_2: real;
```

```
alfa_r,alfa_r2: real;
```

```
alfa2: real;
```

```
{ Para sacar valores }
```

```
a,b: real;
```

```
{ Variables para obtener diferencias de ángulos }
```

```
dif_ang1,dif_ang2,alfa_act,alfa2_act,alfa_ant,alfa2_ant: real;
```

```
Angulos1,Angulos2,coor_rect1,coor_rect2: TextFile;
```

```
t,t_n,mcre,x_1,x_2,x_3,fx_1,fx_2,roty_r: integer;
```

```
l,mcre,y,mcre_y,acutx_int,y_int: real;
```

```
Ang0,Ang1,coor0,coor1: TextFile;
```

```
pp0,pl,q0,q1: real;
```

```
pp0,pp1,qq0,qq1: real;
```

```
Function SEN(x: real): real; far, external 'Funciones.dll'
```

```
Function COSEN(x: real): real; far, external 'Funciones.dll'
```

```
implementation
```

```
{ funciones }
```

```
{ SR * DFM }
```

```
Procedure TForm1.Ejes de pantalla;
```

```
var ex1,ey1,ex2,ey2: integer;
```

```
begin
```

```
Image1.Canvas.Pen.Color := clGreen;
```

```
Image1.Canvas.Pen.Width := 2;
```

```
Image1.Canvas.Ellipse(40,40,370,370);
```

```
ey1 := Round(Image1.Height/2);
```

```
ex1 := Image1.Width;
```

```
Image1.Canvas.MoveTo(0,ey1); { eje de las abscisas }
```

```
Image1.Canvas.LineTo(ex1,ey1);
```

```
ex2 := Round(Image1.Width/2);
```

```
ey2 := Image1.Height;
```

```
Image1.Canvas.MoveTo(ex2,0);
```

```
Image1.Canvas.LineTo(ex2,ey2);
```

```
Image1.Canvas.TextOut(373,210,'X''');
```

```
Image1.Canvas.TextOut(155,5,'EJE: "Y''');
```

```
Image1.Canvas.TextOut(5,210,'"-X''');
```

```
Image1.Canvas.TextOut(175,370,'"-Y''');
```

```
end;
```



```

procedure TTray_con.FormActivate(Sender: TObject);
begin
  WindowState :=wsMaximized;
  with Image1.Canvas do
    begin
      Brush.Color := clwhite;
      image1.Width :=400;
      image1.Height :=400;
      Rectangle(0, 0, Image1.Width, Image1.Height);
      Ejes_de_pantalla;
    end;
end;
procedure TTray_con.Posicion_de_inicio;
begin
  alfa_r:=0;
  alfa_r2:=0;
  {Definicion de los parametros}
  p0:=cosen(alfa_r); p1:=sen(alfa_r);
  q0:=cosen(alfa_r2); q1:=sen(alfa_r2);
  {borra eslabon 1}
  image1.Canvas Pen Width :=2;
  image1.Canvas Pen Color :=clwhite;
  image1.Canvas MoveTo(200,200);
  image1.Canvas LineTo(x1,yy1);
  {borra eslabon 2}
  image1.Canvas Pen Width :=2;
  image1.Canvas Pen Color :=clwhite;
  image1.Canvas LineTo(x1,yy1);
  image1.Canvas LineTo(x2,yy2);
  x1:=Round((11*p0+200));
  yy1:=Round((11*p1+200));
  x2:=Round((11*p0+122*(p0*q0-p1*q1)+200));
  yy2:=Round((11*p1+122*(p0*q1+p1*q0)+200));
  {pinta eslabon 1 en posicion de home}
  image1.Canvas Pen Width :=3;
  image1.Canvas Pen Color :=clgreen;
  image1.Canvas MoveTo(200,200);
  image1.Canvas LineTo(x1,yy1);
  image1.Canvas LineTo(x1,yy1);
  {pinta eslabon 2 en posicion de home}
  image1.Canvas Pen Width :=3;
  image1.Canvas Pen Color :=clred;
  image1.Canvas LineTo(x1,yy1);
  image1.Canvas LineTo(x2,yy2);
  image1.Refresh;
end;

```

```

procedure TTray_con.Pinta_posicion_de_avancee;
begin
  {pinta posicion final del eslabon 1}
  image1.Canvas Pen Width:=2;
  image1.Canvas Pen Color:=clgreen;
  image1.Canvas MoveTo(200,200);
  image1.Canvas LineTo(x1,yy1);
  {pinta posicion final del eslabon 2}
  image1.Canvas Pen Width :=2;
  image1.Canvas Pen Color :=clred;
  image1.Canvas LineTo(x1,yy1);
  image1.Canvas LineTo(x2,yy2);
end;

```

```

procedure TTray_con.Pinta_posicion_final;
begin
  image1.Canvas Pen Width:=2;
  image1.Canvas Pen Color :=clgreen;
  image1.Canvas MoveTo(200,200);
  image1.Canvas LineTo(x1,yy1);
  {pinta posición final del eslabon 2}
  image1.Canvas Pen Width :=2;
  image1.Canvas Pen Color :=clred;

```

```

  image1.Canvas.LineTo(x1,yy1);
  image1.Canvas.LineTo(x2,yy2);
end;

```

```

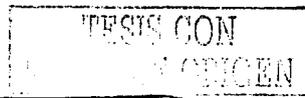
Procedure
TTray_con.Rotacion_del_eslabon_1(alfa_r,alfa_r2:real);
begin
  {Definición de los parámetros}
  pp0:=cosen(alfa_r); pp1:=sen(alfa_r);
  qq0:=cosen(alfa_r2); qq1:=sen(alfa_r2);
  {Rotación de los eslabones con los parámetros}
  xx1:=Round((1*pp0+200);
  yy1:=Round((1*pp1+200);
  xx2:=Round((1*pp0+12*(pp0*qq0-pp1*qq1)+200);
  yy2:=Round((1*pp1+12*(pp0*qq1+pp1*qq0)+200);
  {pinta el eslabón 1 corriendo el prog}
  image1.Canvas Pen Width :=2;
  image1.Canvas Pen Color :=clgreen;
  image1.Canvas MoveTo(200,200);
  image1.Canvas LineTo(x1,yy1);
  {pinta el eslabon 2 corriendo corriendo el prog}
  image1.Canvas Pen Width :=2;
  image1.Canvas Pen Color:=clred;
  image1.Canvas LineTo(x1,yy1);
  image1.Canvas LineTo(x2,yy2);
  image1.Refresh;
  sleep(1);
  image1.Canvas Pen Width :=2;
  image1.Canvas Pen Color :=clwhite;
  image1.Canvas MoveTo(200,200);
  image1.Canvas LineTo(x1,yy1);
  image1.Canvas LineTo(x1,yy1);
  image1.Canvas LineTo(x2,yy2);
  {Pinta trayectoria del eslabon 1}
  image1.Canvas Pen Color :=clgreen;
  image1.Canvas Pen Width :=2;
  image1.Canvas MoveTo(x1,yy1);
  image1.Canvas LineTo(x1,yy1);
  {Pinta trayectoria recta}
  image1.Canvas Pen Color:=clgreen;
  image1.Canvas Pen Width:=2;
  image1.Canvas MoveTo(x2,yy2);
  image1.Canvas LineTo(x2,yy2);
end;

```

```

Procedure TTray_con.pinta_recta;
begin
  x_i:=148;
  y_i:=0;
  x_f:=0;
  y_f:=148;
  image1.Canvas Pen Color:=clblue;
  image1.Canvas Pen Width:=2;
  image1.Canvas.MoveTo(200+x_i,200+y_i);
  image1.Canvas.LineTo(200-x_f,200-y_f);
end;
procedure TTray_con.BitBtn1Chck(Sender: TObject);
var i,j,n_incre,x_i,y_i,x_f,y_f,ots,oty,ots_rot,y_r:integer;
  l_incre,y_incre,x_acu,x_mly,ymt_real;
begin
  ix_i:=StrToInt(Edit1.text);
  y_i:=StrToInt(Edit2.text);
  x_f:=StrToInt(Edit3.text);
  y_f:=StrToInt(Edit4.text);
  n_incre:=StrToInt(Edit5.text);
  Alfa1_i:=StrToFloat(Edit6.text);
  Alfa2_i:=StrToFloat(Edit7.text);
  x_i:=148;
  y_i:=0;
  x_f:=0;

```



```

y_f:=148,
n_incre:=20,
alfa_1:=16,
alfa2_1:=324,
alfa_ant:=0,
alfa2_ant:=360,
l:=Sqrtsqrt(x_f-x_0)*sqrt(y_f-y_0),
Incre_v:=l/n_incre,
l1:=88,l2:=60,
otx:=x_1,
oty:=y_1,
For j:=1 to n_incre+1 do
begin
Metodo_Newton_Rapson(otx,oty,l1,l2,Alfa1_i,Alfa2_i);
incre_v_acu:=incre_v_acu+incre_v;
otx:=Round(x_1+(x_f-x_1)*incre_v_acu/l);
oty:=Round(y_1+(y_f-y_1)*incre_v_acu/l);
For l:=1 To N Do
begin
alfa1_i:=X[1],
alfa2_i:=X[2],
alfa_r:=alfa1_i,
alfa_r2:=alfa2_i,
end,
Tray_con_Memori_Lines Add('alfa' =
'+FormatFloat('0.000',X[1])+' 'alfa2' =
'+FormatFloat('0.000',X[2]));
Rotacion_del_eslabon_1(alfa_r,alfa_r2);
alfa1_act:=X[1],
alfa2_act:=X[2],
dif_ang1:=alfa1_act-alfa1_ant,
dif_ang2:=alfa2_act-alfa2_ant,
alfa1_ant:=alfa1_act,
alfa2_ant:=alfa2_act,
Pinta_posicion_de_avance,
{llama al procedimiento "arreglo_de_bits" para prender
motores};
Arreglo_bits(trunc(dif_ang1/1.8*4),trunc(dif_ang2/1.8*2.4515
)),
end,
Pinta_posicion_final,
end,
procedure TTray_con_BitBtm4Click(Sender: TObject);
begin
Posicion_de_mecio,
a:=3,
veces:=1,
repeat
begin
Mover_motor1;
Mover_motor2;
Sleep(10);
entrada:=lnp($379);
end,
until entrada=247,
a:=20,
veces:=1,
repeat
begin
Mover_motor2;
Sleep(10);
entrada:=lnp($379);
end,
until entrada=231,
end,
procedure TTray_con_BitBtm2Click(Sender: TObject);
begin
pinta_recta;
end,
end,
end,

```

UNIDAD DE EVASIÓN DE OBSTÁCULOS

```
unit Tiempo_real_1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls,
Forms, Dialogs,
```

```
StdCtrls, ExtCtrls,
```

```
Buttons, Resuelve_NR, Prende_motores, math;
```

```
const
```

```
l1:=88,l2:=60,l1:=88,l2:=60;
```

```
type
```

```
TTiempo_real = class(TForm)
```

```
Panel1: TPanel;GroupBox2: TGroupBox;Label4:
```

```
TLabel;Label5: TLabel;
```

```
Label6: TLabel;Label7: TLabel;Label8: TLabel;Label9:
```

```
TLabel;
```

```
Label10: TLabel;Label11: TLabel;Label12: TLabel;Label13:
```

```
TLabel;
```

```
Label14: TLabel;Label15: TLabel;Label16: TLabel;Label17:
```

```
TLabel;
```

```
Label18: TLabel;GroupBox3: TGroupBox;Label19:
```

```
TLabel;Label22: TLabel;
```

```
Label25: TLabel;Label26: TLabel;Panel2: TPanel;BitBtm1:
```

```
TBitBtm;
```

```
BitBtm2: TBitBtm;BitBtm3: TBitBtm;Image1: TImage;BitBtm4:
```

```
TBitBtm;
```

```
Timer1: TTimer;Label31: TLabel;Label3: TLabel;GroupBox1:
```

```
TGroupBox;
```

```
Label27: TLabel;Label28: TLabel;Label29: TLabel;Label30:
```

```
TLabel;Label32: TLabel;
```

```
Label33: TLabel;BitBtm6: TBitBtm; BitBtm7:
```

```
TBitBtm;RadioGroup1: TRadioGroup;
```

```
Label20: TLabel;
```

```
BitBtm8: TBitBtm;
```

```
procedure BitBtm1Click(Sender: TObject);
```

```
procedure FormActivate(Sender: TObject);
```

```
procedure BitBtm4Click(Sender: TObject);
```

```
procedure BitBtm5Click(Sender: TObject);
```

```
procedure Timer1Timer(Sender: TObject);
```

```
procedure BitBtm7Click(Sender: TObject);
```

```
procedure BitBtm6Click(Sender: TObject);
```

```
procedure BitBtm8Click(Sender: TObject);
```

```
procedure RadioGroup1Click(Sender: TObject);
```

```
private
```

```
{ Private declarations }
```

```
Procedure Posicion_de_mecio;
```

```
Procedure Ejes_de_pantalla;
```

```
Procedure Pinta_posicion_de_avance;
```

```
Procedure Pinta_posicion_de_avance2;
```

```
Procedure Pinta_posicion_final;
```

```
Procedure Pinta_posicion_final2;
```

```
Procedure Rotacion_del_eslabon_1(alfa_r,alfa_r2 real);
```

```
Procedure Rotacion_del_eslabon_2(alfa_r,alfa_r2 real);
```

```
Procedure Arreglo_bits(M,N integer);
```

```
Procedure Arreglo_bits_simul(M,N integer);
```

```
Procedure Primera_parte_general;
```

```
Procedure Tercera_parte_general;
```

```
Procedure Primera_opcion_uno_a_uno;
```

```
Procedure Segunda_opcion_hibridas;
```

```
Procedure Tercera_opcion_simultaneas;
```

```
Procedure Pinta_recta;
```

```
Procedure Graficacion_de_los_eslabones;
```

```
Procedure Arreglo_bits_simul_p2(M,N integer);
```

```
Procedure Graficacion_de_los_eslabones2;
```

```
public
```

```

    { Public declarations }
end;

var
Tiempo_real, TTiempo_real,
xx1,yy1,xx2,yy2: integer;
theta_1,alfa_r: integer;
Alfa1_1,Alfa2_1: real;
theta_inc,alfa_r,alfa_r2: real;
alfa2: real;
{Variables de la ecuación de la recta}
Pd,Pt: real;
Xi,Xf,Yi,Yf: real; mere: real;
Xe,Ye: integer;
veces: integer;
otx,oty: integer;
{Para sacar valores}
a,b: real;
{Variables para obtener diferencias de ángulos}
dif_ang1,dif_ang2,alfa1_act,alfa2_act,alfa1_ant,alfa2_ant: real;
{Arreglo de los bits para prender los motores}
M1,M2: Array[1..4] of Integer;
{variables para los obstaculos}
senal,obs: integer;
opcion_final,opcion_1: integer;
I,n_incre_x,Iy,Ix,Iy: Integer; r_oty_r: integer;
I,m_incre_v_incre_v_acu_x,m_t,nt,I_r_incre_vf,incre_f: real;
mere_m: integer;
Ang0: Ang1,coord: coord; TextFile;
theta_min,alfa1,theta_elegida: real;
mere_ang1,mere_ang2: real;
alfa1_a,alfa2_a: real;
p0,p1,q0,q1: real;
pp0,pp1,qq0,qq1: real;
Function SI:(Nx: real) real; far,external'Funciones.dll'
Function COS:(Nx: real) real; far,external'Funciones.dll'
Function
Procedure OutP(Puerto: word; dato: byte); far,external'Port.dll';
Function Inp(Port: word): word; far,external'Port.dll';
implementation

uses Tray, cont1;

{SR * DFM1}
Procedure TTiempo_real Ejes_de_pantalla;
var ex1,ey1,ex2,ey2: integer;
begin
    image1.Canvas.Pen.Color:=clgreen;
    image1.Canvas.Pen.Width:=2;
    Image1.Canvas.Ellipse(40,40,370,370);
    ey1:=Round((image1.height)/2);
    ex1:=image1.Width;
    image1.Canvas.MoveTo(0,ey1); {eje de las abscisas}
    image1.Canvas.LineTo(ex1,ey1);
    ex2:=Round((image1.width)/2);
    ey2:=image1.height;
    image1.Canvas.MoveTo(ex2,0);
    image1.Canvas.LineTo(ex2,ey2);
    Image1.Canvas.TextOut(373,210,"X");
    Image1.Canvas.TextOut(155,5,"Y");
    Image1.Canvas.TextOut(5,210,"X");
    Image1.Canvas.TextOut(175,370,"Y");
end;

procedure TTiempo_real FormActivate(Sender: TObject);
const alfa1:=0,alfa2:=0;
begin
    WindowState:=wsMaximized;
    with Image1.Canvas do begin
        Brush.Color:=clwhite;
        image1.Width:=400;
        image1.Height:=400;
        Rectangle(0, 0, Image1.Width, Image1.Height);
        Ejes_de_pantalla;
        end;
    end;
    Procedure TTiempo_real Posicion_de_inicio;
    Begin
        alfa_r:=0;
        alfa_r2:=0;
        {Definición de los parametros}
        p0:=cosen(alfa_r); p1:=sen(alfa_r);
        q0:=cosen(alfa_r2); q1:=sen(alfa_r2);
        {borra eslabon 1}
        image1.Canvas.Pen.Width:=2;
        image1.Canvas.Pen.Color:=clwhite;
        image1.Canvas.MoveTo(200,200);
        image1.Canvas.LineTo(xx1,yy1);
        {borra eslabon 2}
        image1.Canvas.Pen.Width:=2;
        image1.Canvas.Pen.Color:=clwhite;
        image1.Canvas.LineTo(xx1,yy1);
        image1.Canvas.LineTo(xx2,yy2);
        xx1:=Round((11*p0+200));
        yy1:=Round((11*p1+200));
        xx2:=Round((11*p0+122*(p0*q0-p1*q1)+200));
        yy2:=Round((11*p1+122*(p0*q1+p1*q0)+200));
        {pinta eslabon 1 en posición de home}
        image1.Canvas.Pen.Width:=2;
        image1.Canvas.Pen.Color:=clblue;
        image1.Canvas.MoveTo(200,200);
        image1.Canvas.LineTo(xx1,yy1);
        image1.Canvas.LineTo(xx1,yy1);
        {pinta eslabon 2 en posición de home}
        image1.Canvas.Pen.Width:=2;
        image1.Canvas.Pen.Color:=clblack;
        image1.Canvas.LineTo(xx1,yy1);
        image1.Canvas.LineTo(xx2,yy2);
        image1.Refresh;
    end;

    Procedure TTiempo_real Pinta_posicion_de_avance;
    begin
        {pinta posición final del eslabon 1}
        image1.Canvas.Pen.Width:=2;
        image1.Canvas.Pen.Color:=clgreen;
        image1.Canvas.MoveTo(200,200);
        image1.Canvas.LineTo(xx1,yy1);
        {pinta posición final del eslabon 2}
        image1.Canvas.Pen.Width:=2;
        image1.Canvas.Pen.Color:=clred;
        image1.Canvas.LineTo(xx1,yy1);
        image1.Canvas.LineTo(xx2,yy2);
    end;
    Procedure TTiempo_real Pinta_posicion_de_avance2;
    begin
        {pinta posición final del eslabon 1}
        image1.Canvas.Pen.Width:=2;
        image1.Canvas.Pen.Color:=clblue;
        image1.Canvas.MoveTo(200,200);
        image1.Canvas.LineTo(xx1,yy1);
        {pinta posición final del eslabon 2}
        image1.Canvas.Pen.Width:=2;
        image1.Canvas.Pen.Color:=clblack;
        image1.Canvas.LineTo(xx1,yy1);
        image1.Canvas.LineTo(xx2,yy2);
    end;
end;

```



```

Procedure TTiempo_real Pinta_posicion_final2;
begin
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.Pen.Color:=clblue;
  image1.Canvas.MoveTo(200,200);
  image1.Canvas.LineTo(xx1,yy1);
  {pinta posición final del eslabon 2}
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.Pen.Color:=clblack;
  image1.Canvas.LineTo(xx1,yy1);
  image1.Canvas.LineTo(xx2,yy2);
end;
Procedure
TTiempo_real Rotacion_del_eslabon_1(alfa_r,alfa_r2:real);
begin
  {Definición de los parámetros}
  pp0:=cosen(alfa_r); pp1:=sen(alfa_r);
  qq0:=cosen(alfa_r2); qq1:=sen(alfa_r2);
  {Rotación de los eslabones con los parámetros}
  xx1:=Round(1*pp0+200);
  yy1:=Round(1*pp1+200);
  xx2:=Round(1*pp0+12*(pp0*qq0-pp1*qq1)+200);
  yy2:=Round(1*pp1+12*(pp0*qq1+pp1*qq0)+200);
  {pinta el eslabon 1 corriendo el prog}
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.Pen.Color:=clgreen;
  image1.Canvas.MoveTo(200,200);
  image1.Canvas.LineTo(xx1,yy1);
  {pinta el eslabon 2 corriendo el prog}
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.Pen.Color:=clred;
  image1.Canvas.LineTo(xx1,yy1);
  image1.Canvas.LineTo(xx2,yy2);
  image1.Refresh;
  sleep(1);
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.Pen.Color:=clwhite;
  image1.Canvas.MoveTo(200,200);
  image1.Canvas.LineTo(xx1,yy1);
  image1.Canvas.LineTo(xx2,yy2);
  {Pinta trayectoria del eslabon 1}
  image1.Canvas.Pen.Color:=clgreen;
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.MoveTo(xx1,yy1);
  image1.Canvas.LineTo(xx1,yy1);
  {Pinta trayectoria recta}
  image1.Canvas.Pen.Color:=clgreen;
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.MoveTo(xx2,yy2);
  image1.Canvas.LineTo(xx2,yy2);
end;
Procedure
TTiempo_real Rotacion_del_eslabon_1_c(alfa_r,alfa_r2:real);
begin
  {Definición de los parámetros}
  pp0:=cosen(alfa_r); pp1:=sen(alfa_r);
  qq0:=cosen(alfa_r2); qq1:=sen(alfa_r2);
  {Rotación de los eslabones con los parámetros}
  xx1:=Round(1*pp0+200);
  yy1:=Round(1*pp1+200);
  xx2:=Round(1*pp0+12*(pp0*qq0-pp1*qq1)+200);
  yy2:=Round(1*pp1+12*(pp0*qq1+pp1*qq0)+200);
  {pinta el eslabon 1 corriendo el prog}
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.Pen.Color:=clblue;
  image1.Canvas.MoveTo(200,200);
  image1.Canvas.LineTo(xx1,yy1);
  {Pinta trayectoria del eslabon 1}
  image1.Canvas.Pen.Color:=clgreen;
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.MoveTo(xx2,yy2);
  image1.Canvas.LineTo(xx2,yy2);
end;
Procedure TTiempo_real Pinta_recta;
begin
  image1.Canvas.Pen.Color:=clblue;
  image1.Canvas.Pen.Width:=2;
  image1.Canvas.MoveTo(200+xx1,200+yy1);
  image1.Canvas.LineTo(200+xx1,200+yy1);
end;
Procedure TTiempo_real Primera_parte_general;
begin
  xx1:=148; yy1:=0; xx1_f:=0; yy1_f:=148;
  alfa1_i:=16; alfa2_i:=324; alfa1_ant:=0; alfa2_ant:=360;
  n_mere:=10;
  l:=sqrt(sqrt(x_f-x_i)*sqrt(y_f-y_i));
  mere_x:=(l/n_mere)/2;
  otx:=x_i;oty:=y_i;
  Pinta_recta;
  Rotacion_del_eslabon_1_c(0,0);
  {image1.Canvas.Ellipse(250,170,270,190); }
  For j:=1 to n_mere+1 do
    begin
      Metodo_Newton_Rapsomt(otx,oty,11,12,Alfa1_i,Alfa2_i);
      mere_v_acu:=mere_v_acu+mere_v;
      otx:=Round(x_i+(x_f-x_i)*mere_v_acu/l);
      oty:=Round(y_i+(y_f-y_i)*mere_v_acu/l);
      alfa_r:=X[1];
      alfa_r2:=X[2];
      Rotacion_del_eslabon_1_c(-alfa_r,-alfa_r2);
      Pinta_posicion_de_avance2;
      l_r:=sqrt(sqrt(otx)*sqrt(oty));
      Rotacion_del_eslabon_1_c(-alfa_r,-alfa_r2);
      alfa1_act:=alfa_r;
      alfa2_act:=alfa_r2;
      dif_ang1:=alfa1_act-alfa1_ant;
      dif_ang2:=alfa2_act-alfa2_ant;
      alfa1_ant:=alfa1_act;
      alfa2_ant:=alfa2_act;
    }llama al procedimiento "arreglo_de_bits" para prender
    motores};
    Arreglo_bits2(trunc(dif_ang1/l*8^4),trunc(dif_ang2/l*8^2+151));
    Pinta_posicion_de_avance2;
    end;
  end;
Procedure TTiempo_real Tercera_parte_general;
begin
  alfa1_i:=16; alfa2_i:=324;
  mere_v_acu:=mere_v_acu;
  otx:=otx;oty:=oty;

```



```

if alfa1 < alfa2 then begin
  repeat begin
    alfa_r := alfa_r - 0.6;
    alfa_r2 := alfa_r2 - 0.6;
    Mover_motor_11_y_22;
    Rotacion_del_eslabon_1(-alfa_r, -alfa_r2);
    Pinta_posicion_de_avancee;
    Label3.caption := FormatFloat('0.00', -alfa_r);
    Label3.refresh;
    label31.caption := FormatFloat('0.000', -alfa_r2);
    label31.refresh;
  end;
  until alfa_r = 0;
end;
alfa1 := alfa_r;
alfa2 := alfa_r2;
{Termina de posicionar al robot en la posicion inicial}
if alfa2 < alfa1 then begin
  repeat begin
    theta_elegida := alfa2;
    alfa_r := alfa_r;
    alfa_r2 := alfa_r2 - 0.6;
    Mover_motor22;
    Rotacion_del_eslabon_1(-alfa_r, -alfa_r2);
    Pinta_posicion_de_avancee;
    Label31.caption := FormatFloat('0.00', -alfa_r2);
    Label31.refresh;
  end;
  until alfa_r2 = 0;
end;
Pinta_posicion_final;
Pinta_recta;
{Calcula los angulos a rotar}
alfa1 := 16; alfa2 := 324;
otx := Round(x1*(x1 - x2)/d)*(incv_v_actu - incv_v/d);
oty := Round(y1*(y1 - y2)/d)*(incv_v_actu - incv_v/d);
Metodo_Newton_Rapsont(otx, oty, H1, H2, Alfa1_r, Alfa2_r);
alfa := N[1];
alfa2 := N[2];
alfa1 := 360 - alfa1;
alfa2 := 360 - alfa2;
alfa_r := 0;
alfa_r2 := 0;
if alfa2 < alfa1 then begin
  repeat begin
    alfa_r := alfa_r + 0.6;
    alfa_r2 := alfa_r2 + 0.6;
    Mover_motor_11_y_22;
    Rotacion_del_eslabon_1(alfa_r, alfa_r2);
    Pinta_posicion_de_avancee;
    Label3.caption := FormatFloat('0.00', alfa_r);
    Label3.refresh;
    label31.caption := FormatFloat('0.000', alfa_r2);
    label31.refresh;
  end;
  until alfa_r2 = alfa2;
end;
if alfa2 > alfa1 then begin
  repeat begin
    theta_elegida := alfa2;
    alfa_r := alfa_r + 0.6;
    alfa_r2 := alfa_r2;
    Mover_motor11;
    Rotacion_del_eslabon_1(alfa_r, alfa_r2);
    Pinta_posicion_de_avancee;
    Label31.caption := FormatFloat('0.00', alfa_r);
    Label31.refresh;
  end;
  until alfa_r = alfa1;
end;

Pinta_posicion_final;
Pinta_recta;
Tercera_parte_general;
Pinta_recta;
end;

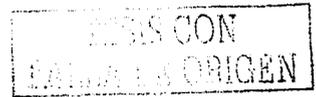
Procedure TTiempo_real Graficacion_de_los_eslabones;
begin
  alfa_r := alfa_r - incre_ang1;
  alfa_r2 := alfa_r2 - 0.734244;
  Rotacion_del_eslabon_1(-alfa_r, -alfa_r2);
  Pinta_posicion_de_avancee;
  Label3.Caption := FormatFloat('0.00', alfa_r);
  Label3.refresh;
  Label31.Caption := FormatFloat('0.00', alfa_r2);
  Label31.refresh;
end;

Procedure TTiempo_real Graficacion_de_los_eslabones2;
begin
  alfa_r := alfa_r - 0.6;
  alfa_r2 := alfa_r2 - incre_ang2;
  Rotacion_del_eslabon_1(-alfa_r, -alfa_r2);
  Pinta_posicion_de_avancee;
  Label3.Caption := FormatFloat('0.00', alfa_r);
  Label3.refresh;
  Label31.Caption := FormatFloat('0.00', alfa_r2);
  Label31.refresh;
end;

Procedure TTiempo_real Pinta_posicion_final;
begin
  image1.Canvas.Pen.Width := 2;
  image1.Canvas.Pen.Color := clblue;
  image1.Canvas.MoveTo(200, 200);
  image1.Canvas.LineTo(xx1, yy1);
  {pinta posicion final del eslabon 2}
  image1.Canvas.Pen.Width := 2;
  image1.Canvas.Pen.Color := clblack;
  image1.Canvas.LineTo(xx1, yy1);
  image1.Canvas.LineTo(xx2, yy2);
end;

Procedure TTiempo_real Arreglo_bits_simul(M,N:integer);
var a,t,MMax:integer;
    mdicc1, mdicc2, mdicc: integer;
    delta:real;
begin
  M1[1] := 3; M1[2] := 1; M1[3] := 0; M1[4] := 2;
  M2[1] := 20; M2[2] := 16; M2[3] := 0; M2[4] := 4;
  incre_ang2 := (incre_ang2 + alfa2_act/alfa1_act)*0.6;
  incre_ang1 := (incre_ang1 + alfa1_act/alfa2_act)*0.734244;
  if (M < 0) then
    begin
      M1[2] := 2;
      M1[4] := 1;
      M := -M;
    end;
  if (N < 0) then
    begin
      M2[2] := 4;
      M2[4] := 16;
      N := -N;
    end;
  MMax := max(m, n);
  delta := m/n;
  if (delta > 1) then delta := 1.0/delta;
  for i := 1 to MMax do
    begin
      mdicc1 := ((i-1) mod 4) + 1;
      mdicc2 := round(delta*i);

```



```

    indice2 := indice2 mod 4;
    indice2 := indice2 + 1;
if (n>m) then
begin
    indice := indice2;
    indice2 := indice1;
    indice1 := indice;
end;
a :=M1[indice1] + M2[indice2];
Outp($378,a);
Graficacion_de_los_estabones;
sleep(10);
end;
Outp($378,0);
Pinta_posicion_final;
end;

```

```

Procedure TTiempo_real Arreglo_bits_simul_p2(M,N:integer);
var a,i,MMax:integer;
    indice1,indice2,indice : integer;
    delta:real;
begin
    M1[1] := 3;M1[2] := 1;M1[3] := 0;M1[4] := 2;
    M2[1] := 20;M2[2] := 16;M2[3] := 0;M2[4] := 4;
    incre_ang2 :=(incre_ang2+alfa2_act/alfa1_act)*0.6;
    incre_ang1 := (incre_ang1 + alfa1_act/alfa2_act)*0.734244);
if (N<0) then
begin
    M1[2] := 2;
    M1[4] := 1;
    M := -M;
end;
if (N<0) then
begin
    M2[2] := 4;
    M2[4] := 16;
    N := -N;
end;
MMax:=max(m,n);
delta := m/n;
if (delta>1) then delta := 1.0/delta;
for i:=1 to MMax do
begin
    indice1 := ((i-1) mod 4) + 1;
    indice2 := round(delta*i);
    indice2 := indice2 mod 4;
    indice2 := indice2 + 1;
if (n>m) then
begin
    indice := indice2;
    indice2 := indice1;
    indice1 := indice;
end;
a :=M1[indice1] + M2[indice2];
Outp($378,a);
Graficacion_de_los_estabones2;
sleep(10);
end;
Outp($378,0);
Pinta_posicion_final;
end;

```

```

Procedure TTiempo_real Tercera_opcion_simultaneas;
begin
    Primera_parte_general;
    {Aplicacion de la cinemática directa hasta llegar al punto
requerido}
    alfa1_act := alfa_r;
    alfa2_act := alfa_r2;

```

```

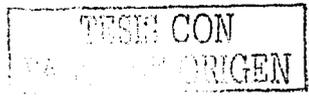
if alfa1_act<0 then begin
    alfa1_act:=360-alfa1_act;
end;
if alfa2_act<0 then begin
    alfa2_act:=360+alfa2_act;
end;
Arreglo_bits_simul(trunc(alfa1_act/1.8*3.0),trunc(alfa2_act/1.8*2.4515));
    alfa_r:=0;
    alfa_r2:=0;
    incre_ang2:=0;
    incre_ang1:=0;
    {Aplicacion de la cinemática inversa para llegar al segundo
punto requerido}
    alfa1_i:=16; alfa2_i:=324;
    incre_v_acu:=incre_v_acu;
    otx:=Round(x_i+(x_f-x_i)*(incre_v_acu-incre_v)/l);
    oty:=Round(y_i+(y_f-y_i)*(incre_v_acu-incre_v)/l);
    Metodo_Newton_Rapson(otx,oty,l,l2,Alfa1_i,Alfa2_i);
    alfa1_act:=X[1];
    alfa2_act:=X[2];
if alfa1_act>0 then begin
    alfa1_act:=360-alfa1_act;
end;
if alfa2_act<0 then begin
    alfa2_act:=360-alfa2_act;
end;
Arreglo_bits_simul_p2(trunc(alfa1_act/1.8*3.0),trunc(alfa2_act/1.8*2.4515));
    Tercera_parte_general;
    Pinta_posicion_final;
end;

```

```

Procedure TTiempo_real Arreglo_bits2(M,N:integer);
var a,i,MMax:integer;
    indice1, indice2,indice : integer;
    delta:real;
    cont,cont1,cont2,cont_act,cont_ant:integer;
begin
    M1[1] := 3;M1[2] := 2;M1[3] := 0;M1[4] := 1;
    M2[1] := 20;M2[2] := 4;M2[3] := 0;M2[4] := 16;
if (N<0) then begin
    M1[2] := 1;
    M1[4] := 2;
    M := -M;
end;
if (N<0) then begin
    M2[2] := 16;
    M2[4] := 4;
    N := -N;
end;
MMax:=max(m,n);
if n=0 then n:=1;
delta := m/n;
if (delta>1) then delta := 1.0/delta;
for i:=1 to MMax do begin
    indice1 := ((i-1) mod 4) + 1;
    cont1:=cont1+1;
    indice2 := round(delta*i);
    indice2 := indice2 mod 4;
    indice2 := indice2 + 1;
    cont_act:=indice2;
    if cont_act=cont_ant then begin
        cont2:=cont2;
    end
    else begin
        cont2:=cont2+1;
    end;
    cont_ant:=cont_act;

```



```

if (n>m) then begin
  indice := indice2;
  indice2 := indice1;
  indice1 := indice;
  cont := cont2;
  cont2 := cont1;
  cont1 := cont;
end;
a := M1[indice1] + M2[indice2];
Outp($378.a);
sleep(100);
end;
Outp($378.0);
end;
procedure TTiempo_real.BitBtn4Click(Sender: TObject);
begin
  a:=3;
  veces:=1;
  repeat begin
    Mover_motor1;
    Sleep(10);
    entrada:=Inp($379);
  end;
  until entrada=247;
  a:=20;
  veces:=1;
  repeat begin
    Mover_motor2;
    Sleep(10);
    entrada:=Inp($379);
  end;
  until entrada=231;
  Posicion_de_inicio;
end;
procedure TTiempo_real.BitBtn5Click(Sender: TObject);
begin
  image1.Canvas.Ellipse(250.170,270.190);
end;
procedure TTiempo_real.Timer1Timer(Sender: TObject);
begin
  Label3.caption:=IntToStr(obs);
  label3.refresh;
end;
procedure TTiempo_real.BitBtn7Click(Sender: TObject);
begin
  image1.Canvas.Pen.Width:=1;
  image1.Canvas.Pen.Color:=clred;
  image1.Canvas.Ellipse(263.150,283.170);
  label32.Caption:='Si';
end;
procedure TTiempo_real.BitBtn6Click(Sender: TObject);
begin
  image1.Canvas.Pen.Width:=1;
  image1.Canvas.Pen.Color:=clred;
  image1.Canvas.Ellipse(250.170,270.190);
  label30.Caption:='Si';
end;

procedure TTiempo_real.BitBtn8Click(Sender: TObject);
begin
  image1.Canvas.Pen.Width:=1;
  image1.Canvas.Pen.Color:=clred;
  image1.Canvas.Ellipse(263.150,283.170);
  image1.Canvas.Ellipse(250.170,270.190);
  label33.Caption:='Si';
end;

procedure TTiempo_real.RadioGroup1Click(Sender: TObject);
begin
  Case RadioGroup1.ItemIndex of
    0:opcion_1:=0;
    1:opcion_1:=1;
    2:opcion_1:=2;
  end;
  opcion_final:=opcion_1;
end;
end.

```

