

“PROPUESTA DE SOLUCIÓN AL PROBLEMA DE GENERACIÓN DE TRAYECTORIAS DE TIEMPO MÍNIMO PARA UN BRAZO MANIPULADOR DE SEIS GRADOS DE LIBERTAD MEDIANTE UN APLICATIVO BASADO EN UNA TRAYECTORIA ESPECÍFICA EN UN AMBIENTE SIMULADO.”

TRABAJO DE GRADO

**Presentado como requisito parcial para la obtención del título de
INGENIERO INDUSTRIAL**

AUTORES:

**FELIPE RIVERA SANÍN
LUISA FERNANDA ROSAS CASTRO**

DIRECTOR:

Lic. MARTHA RUTH MANRIQUE TORRES

CO-DIRECTOR:

ING. GABRIEL MAURICIO ZAMBRANO REY



**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
BOGOTÁ D. C., 23 de OCTUBRE DE 2009**

Agradecemos a nuestros padres por la motivación y el apoyo incondicional que nos han brindado, así como a nuestros compañeros y amigos, Lina, Nathalia, Carlos y Germán, sin los cuales esta etapa no habría sido igual.

Damos gracias a los directores de este proyecto, Martha Manrique y Gabriel Zambrano, quienes confiaron en nosotros, nos orientaron y nos apoyaron en cada paso.

Finalmente, agradecemos a Wilson, Rai y Deivy, funcionarios del CTAI, que nos colaboraron en todo momento.

TABLA DE CONTENIDO

Introducción	1
Componentes de la celda de trabajo	1
Funciones del robot Mitsubishi Melfa IV en la celda de trabajo.....	3
1. Formulación del Problema	4
2. Planteamiento del problema	4
3. Justificación.....	4
4. Objetivo General	6
4.1. Objetivos específicos	6
5. Marco Conceptual.....	7
5.1. Robótica	7
5.2. Robot manipulador	9
5.3. Aplicaciones.....	12
5.4. Robot Mitsubishi Melfa IV.....	14
5.5. Tipo de trayectorias que sigue un robot manipulador.....	15
5.6. Procedimiento para la programación de un robot industrial.....	26
5.7. Métodos de optimización matemática	28
5.8. Definición del espacio de trabajo del robot	28
6. Selección del algoritmo de optimización	32
7. Análisis de la trayectoria de carga y descarga del torno CNC	36
8. Herramientas	39
8.1. Herramientas de software.....	39
8.2. Equipo (Hardware)	40
9. Implementación	41
9.1. Definición del espacio de trabajo del robot MELFA RV-2A	42
9.2. Búsqueda de la solución inicial.....	48
9.3. Funcionamiento del algoritmo para el problema de generación de trayectorias de tiempo mínimo en la celda de trabajo del CTAL.....	50
9.4. Diseño experimental	56
9.5. Imágenes de la interfaz y diagramas de flujo para la programación del algoritmo	59
9.6. Procedimiento de identificación de puntos vía.....	62

9.7.	Resultados de la ejecución del algoritmo ACS	65
9.8.	Procedimiento de refinamiento de la solución	67
9.9.	Transformación de la solución a escala del robot	70
9.10.	Programación de la solución en lenguaje Melfa IV	71
10.	Parámetros de comparación de los resultados	72
11.	Resultados y análisis de la implementación de las soluciones obtenidas	73
12.	Impacto	78
13.	Conclusiones	81
14.	Recomendaciones.....	83
Bibliografía		I
Anexos.....		V
A.	Antecedentes para la comparación del algoritmo.....	V
B.	Antecedentes para la comparación de los métodos de representación del espacio de trabajo.....	IX
C.	Información adicional sobre la metaheurística ACO	XI
D.	Protocolo para la definición del espacio de trabajo del Robot Mitsubishi Melfa IV en las trayectoria de carga y descarga del torno CNC.....	XIII
E.	Problemas encontrados en la ejecución del procedimiento	XVII
F.	Imágenes del proceso de definición del espacio de trabajo del robot.....	XIX
G.	Jerarquía de VI's para la definición del espacio de trabajo y búsqueda de la solución inicial.....	XXIII
H.	Protocolo para la implementación y calificación de las trayectorias generadas por medio del algoritmo ACS.....	XXIV
I.	Descripción de las pruebas experimentales para el ajuste de los parámetros del algoritmo ACS XXVIII	
J.	Programación del robot Melfa RV- 2A en lenguaje Melfa IV (COSIMIR® Professional).....	XXXII

ÍNDICE DE FIGURAS

Figura 1. Celda de Trabajo del CTAI. Fuente: Los Autores	1
Figura 2. Robots de Servicio	7
Figura 3. Sistema de un robot manipulador [2]	9
Figura 4. Morfología del robot [2].....	10
Figura 5. Tipos de articulaciones comúnmente utilizadas en la construcción de un robot industrial [2].....	11
Figura 6. Robot de seis grados de libertad [2].....	11
Figura 7. Configuraciones básicas del sistema del cuerpo de un manipulador industrial.....	12
Figura 8. Aplicaciones de robots industriales en la Cadena de Abastecimiento. Fuente: Los Autores.	13
Figura 9. Desestibado de botes [45].....	13
Figura 10. Robot realizando proceso de inyección de piezas [44]	13
Figura 12. Robot Mitsubishi Melfa Iv [7]	14
Figura 11. Robot paletizado sacos [46]	14
Figura 13. ejecución de movimientos del robot [6]	15
Figura 14. Movimiento herramienta relativo a la trama estación. Fuente: [5] <i>Modificado por los autores</i>	17
Figura 15. Segmentos lineales con enlaces parabólicos y múltiples puntos de paso [38]	19
Figura 16. Interpolaciones lineales con enlaces parabólicos. [38].....	19
Figura 17. Uso de <i>pseudopuntos</i> vía para crear un “punto de paso”.....	20
Figura 18. Vector de posición [5]	21
Figura 19. Representación ángulo-eje equivalente [5]	22
Figura 20. Representación posición y orientación de la herramienta.....	22
Figura 21. Descripción de los comandos de programación. Fuente: Los Autores.....	25
Figura 22. Obstáculo “agrandado” [7, p. 89].....	29
Figura 23. <i>road maps or skeleton methods</i>	29
Figura 24. Descomposición de celda. <i>quadtrees</i> [9].....	30
Figura 25. Campos portenciales artificiales (<i>APF Potential Fields</i>) [7]	30
Figura 26. Ejemplo de un espacio “2 y medio”, se toman puntos a diferentes alturas y se unen formando un poliedro.....	42
Figura 27. Orientaciones para las cuales se midieron los espacios de trabajo	43
Figura 28. representación del espacio de trabajo para la posición <i>pcyl4</i>	44

Figura 29. Vi “Transformación imagen de trabajo a matriz de 0 y 255”	45
Figura 30. VI “Definición del espacio de trabajo”	46
Figura 31. Centro de las celdas.....	46
Figura 32. Solución inicial del hallada para ambas direcciones de giro	48
Figura 33. Movimientos posibles para una hormiga en la posición (i,j). Fuente: Los Autores.....	50
Figura 34. Identificación puntos cambios de dirección y corrección irregularidades del Camino	62
Figura 35. Ejemplo Identificación de irregularidades y cambios de dirección. Fuente: Los Autores	63
Figura 36. Distancias entre puntos vía. Fuente: Los Autores	67
Figura 37. Valores límite de x y y para cada orientación. Fuente: Los Autores.....	71
Figura 38. Búsqueda del camino más corto	XI
Figura 39. Ubicación del TCP	XVII
Figura 40. sistemas de referencia	XVIII
Figura 41. Interfaz prueba para la definición de número de colonias	XXIX
Figura 42. Gráfica Distancia vs. resolución. Fuente: Los Autores.....	XXXI

ÍNDICE DE DIAGRAMAS

Diagrama 1. Diagrama de flujo para la configuración de un proyecto de robótica. Fuente: Los Autores.....	27
Diagrama 2. Diagrama de flujo procesos de carga y descarga del torno. Fuente: Los Autores	37
Diagrama 3. Diagrama de flujo metodología para la implementación Fuente: Los Autores	41
Diagrama 4. Diagrama de flujo proceso de captura de la imagen y definición del espacio de trabajo. Fuente: Los Autores.....	47
Diagrama 5. Diagrama de flujo proceso de búsqueda de solución inicial. Fuente: Los Autores.....	49
Diagrama 6. Diagrama de flujo Algoritmo Sistema Colonia de Hormigas (ACS) para el problema de generación de trayectorias. Fuente: Los Autores	54
Diagrama 7. Diagrama de flujo procedimiento “Acciones del demonio”. Fuente: Los Autores	55
Diagrama 8. Procedimeinto para encontrar puntos vía y corrección de irregularidades. Fuente: Los Autores	64
Diagrama 9. Procedimiento de refinamiento de la solución. Fuente: Los Autores.....	68
Diagrama 10. Metodología proceso medición espacio de trabajo. Fuente: Los Autores	XV
Diagrama 11. Metodología proceso implementación de las soluciones en el Robot Melfa RV 2A. Fuente: Los Autores	XXVI
Diagrama 12. Proceso Definición movimiento entre puntos vía. Fuente: Los Autores	XXVII

ÍNDICE DE TABLAS

Tabla 1 - ELEMENTOS DE LA CELDA DE TRABAJO.....	2
Tabla 2 – especificaciones equipo de computo	40
Tabla 3. Ajuste de Parámetros para la ejecución del algoritmo	58
Tabla 4. resultados de la ejecución del algoritmo para los espacios definidos.....	65
Tabla 5. resultados de la ejecución del algoritmo para los espacios con obstáculo	66
Tabla 6. Resultados del proceso de refinamiento de trayectorias.....	69
Tabla 7- PARámetros para la comparación de las soluciones generadas	72
Tabla 8. Distancias de las trayectorias obtenidas con el algoritmo ACS	73
Tabla 9. Tiempos de ejecución de las trayectorias implementadas en el robot Melfa rv2a (en segundos)	73
Tabla 10. Resultados parámetros de comparación de trayectorias.....	74
Tabla 11. Matriz de ponderación para la calificación de trayectorias.....	75
Tabla 12. Resultados de la implementación de las trayectorias en los espacios modificados.....	76
Tabla 13. Resultados definición número colonias.....	XXVIII
Tabla 14. Resultados muestreo para el ajuste de parámetros	XXX

RESUMEN EJECUTIVO

El presente trabajo de grado expone una metodología para la solución al problema de generación de trayectorias de manipuladores industriales, basada en la metaheurística Optimización por Colonia de Hormigas (*Ant Colony Optimization, ACO*).

El problema de generación de trayectorias consiste básicamente en encontrar la mejor configuración de puntos vía, que permitan al robot ir de un punto inicial a un punto final en términos de tiempo, distancia y facilidad para ejecutar el movimiento. La determinación de esta configuración cobra importancia cuando se analiza que esta influye directamente en la eficiencia de la ejecución de las tareas, asociada a los tiempos de ciclo y naturalmente, a los costos del proceso.

Teniendo en cuenta lo anterior, se considero el impacto del proyecto en la cadena logística, al identificar las aplicaciones que presentan los robots manipuladores en operaciones que se encuentran en los diferentes eslabones.

Inicialmente, y con el fin de contextualizar al lector, se presenta en el Marco Teórico una revisión de conceptos de robótica industrial, métodos de optimización matemática y métodos para definición de espacios de trabajo.

Más adelante, se describen las soluciones planteadas por diferentes autores para la solución de este problema, para finalmente describir la metaheurística seleccionada que se adapta al problema. Como punto de partida, se realiza un análisis de los proceso de carga y descarga del torno en la celda de manufactura del CTAI¹, y se exponen las herramientas que se utilizaron para la realización del proyecto.

En el Capítulo 9, se encuentra la exposición del desarrollo del trabajo, desde la medición del espacio, hasta la implementación de las soluciones obtenidas con e.l algoritmo en el robot.

Finalmente, en los últimos capítulos se presentan los resultados obtenidos, su posterior análisis, conclusiones y recomendaciones para futuros trabajos.

El principal aporte del presente trabajo es por tanto, la realización de una propuesta metodológica clara para conseguir la generación de trayectorias que, aún cuando no garantizan ser las óptimas, presentan valores bajos en términos de tiempo y distancia, y garantizan la evasión de obstáculos.

¹ Centro Tecnológico de Automatización Industrial

INTRODUCCIÓN

Una celda de manufactura simulada es un espacio orientado a la aplicación y generación de conocimiento donde se busca encontrar soluciones a diferentes problemáticas, a nivel académico o dentro del sector productivo, promoviendo la investigación en campos como el análisis de operaciones, innovación y desarrollo y la automatización de procesos.

La sala CNC del Centro Tecnológico de Automatización Industrial (CTAI) se considera un sistema flexible de manufactura a escala, dado que se incorporan elementos de robótica industrial, logística de almacenamiento y distribución, integrados mediante sistemas de información, con el fin de programar y controlar procesos realizados en condiciones industriales simuladas y en ocasiones reales. Allí se encuentran varias estaciones de trabajo, que incluyen un almacén de materia prima y producto terminado, una banda transportadora para el flujo de material, un robot manipulador, una estación de control de calidad por visión artificial y una estación de mecanizado CNC.²

Dentro de la celda de trabajo, se considera la importancia del robot manipulador en el proceso debido a la naturaleza de sus funciones; el robot es responsable del manejo y alimentación de piezas a las máquinas CNC, y, aunque cuando estas no agregan valor al producto, generan un impacto en los costos asociados a la operación.

COMPONENTES DE LA CELDA DE TRABAJO

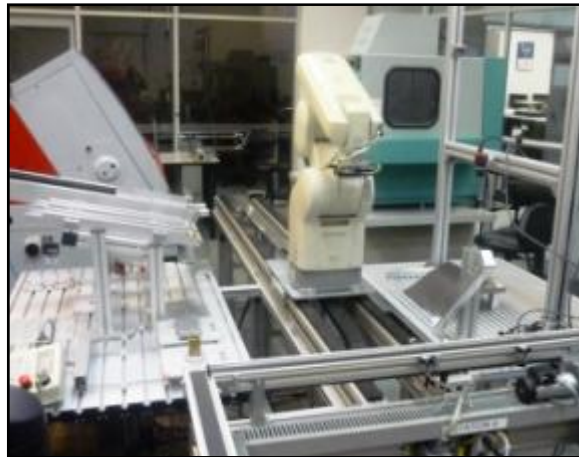


FIGURA 1. Celda de Trabajo del CTAI. Fuente: Los Autores

² Pontificia Universidad Javeriana. Centro Tecnológico de Automatización Industrial. Recuperado el 27 de Marzo de 2009, de http://ingenierias.javeriana.edu.co/portal/page/portal/facultad_ingenieria/espanol/industrial/laboratorios/C_T_A_1?tab=inicio


Cada una de las estaciones descritas anteriormente cuenta con diferentes elementos necesarios para el desarrollo de los procesos que se llevan a cabo en la celda; para aquellos procesos en los que se involucra el robot Melfa IV³ se tiene⁴:

TABLA 1 - ELEMENTOS DE LA CELDA DE TRABAJO

ELEMENTO	FIGURA	FUNCIÓN
Torno EMCO CNC		Realiza procesos de mecanizado de piezas cilíndricas de aluminio y bronce, entre otros materiales.
Fresadora BENCHMAN		Realiza procesos de mecanizado de piezas de materiales blandos.
Estaciones soporte estiba 1 y 2		Son las estaciones sobre las que se ubican las estibas provenientes del almacén, con el fin de recoger o llevar material.
Alimentadores piezas de aluminio y bronce		Son los alimentadores de piezas cilíndricas para procesos de mecanizado en torno.
Alimentador de piezas (base plate)		Es el alimentador de piezas de aluminio para procesos de mecanizado en fresa.

³ Una descripción detallada del robot se encuentra en el Capítulo 5.4 "Robot Mitsubishi Melfa IV".

⁴ Todas las imágenes de este cuadro pertenecen a: Festo Didactic GmbH & Co.KG. (2003). iCIM CNC Coupling A004.

<p align="center">Estación 4 de Banda Transportadora</p>		<p>Es la estación 4 de la banda transportadora, desde la cual se recoge la estiba que viene del almacén para ser ubicado en los soportes 1 o 2.</p>
---	---	---

Fuente: Los Autores.

FUNCIONES DEL ROBOT MITSUBISHI MELFA IV EN LA CELDA DE TRABAJO

El robot Mitsubishi Melfa IV desempeña tres funciones básicas en la celda de trabajo:

- 1) Colocar y retirar estibas de la banda transportadora: *MP – Move pallet*
El brazo manipulador toma la estiba y lo mueve entre la banda y las estaciones 1 o 2.
- 2) Alimentación de las maquinas CNC
 - Carga del torno: *LTURN – Load Turn*
 - Descarga torno: *ULTURN – Unload Turn*
 - Carga fresadora: *LMILL – Load Mill*
 - Descarga fresadora: *ULMILL – Unload Mill*
- 3) Manejo de las piezas en el proceso de estibado
El brazo toma cubos de un alimentador de piezas para ubicarlos sobre una estiba con un patrón tipo matriz colocada sobre la estación 1 o 2.

Como se puede observar de la definición de cada una de las funciones, existen diferentes trayectorias para cada una de ellas, dependiendo del punto de origen y destino final de las piezas en una tarea particular

1. FORMULACIÓN DEL PROBLEMA

¿Es posible lograr para el robot Melfa IV en la celda de trabajo del CTAI⁵ la generación de trayectorias libres de colisiones que cumplan las condiciones de tiempo mínimo, utilizando una aplicación basada en un algoritmo particular?

2. PLANTEAMIENTO DEL PROBLEMA

Los movimientos que realizan los manipuladores industriales para la realización de una tarea en una celda de manufactura están determinados por una secuencia de puntos que generan un trayecto continuo (trayectoria), en el cual se controla la posición y orientación de la herramienta [1], de acuerdo con los requerimientos la misma y de la conformación del espacio de trabajo.

Dicha configuración de puntos vía (también conocidos como puntos críticos), es definida por el operador utilizando el control de mando manual (*teach box*), por medio del cual se ubica el manipulador en la posición deseada y se almacena en el controlador con el fin de ser utilizada en la ejecución de un programa. Sin embargo, esta configuración es producto del conocimiento y la experiencia del operador, quien no emplea procedimientos que permitan confirmar que dicha posición es adecuada para el desempeño del proceso.

En consecuencia, la determinación de estos puntos se considera un proceso crítico, dado que su ubicación incorrecta puede influir negativamente en la eficiencia de la ejecución de las tareas [2], lo cual se refleja en los tiempos de ciclo, impactando los costos asociados al proceso.

3. JUSTIFICACIÓN

Desde el objetivo fundamental de la ingeniería de métodos, el cual busca facilitar la realización del trabajo por medio de mejoras que permitan que este se haga en el menor tiempo posible y con una menor inversión por unidad producida [3], se hace necesario estudiar la posibilidad de implementar una metodología que permita la generación de trayectorias de tiempo y distancia mínimos en la celda de manufactura.

Con este fin, el manual de funciones y operaciones del robot Melfa IV [4] ilustra tres métodos diferentes que posibilitan la reducción de los tiempos de ciclo:

- Garantizar que la velocidad del robot sea apropiada en todo momento por medio de los comandos de programación. En otras palabras, que cada vez que el robot deba disminuir la velocidad para realizar tareas específicas, retorne a su velocidad máxima una vez terminada la tarea.
- Construir la trayectoria de forma tal que sea lo más “suave” posible, evitando cambios de dirección repentinos; para esto, puede obligarse al robot a realizar un movimiento continuo, con lo cual la velocidad no se reducirá en los puntos vía, mejorando la velocidad total.
- Aumentar o reducir el número de puntos vía, de forma tal que la posición final sea alcanzada en el menor número de operaciones posibles.

⁵ CTAI (Centro Tecnológico de Automatización Industrial de la Facultad de Ingeniería de la Pontificia Universidad Javeriana)

El primer método mencionado, hace referencia a la forma en la que se estructura el programa con el cual se lleva a cabo la tarea, mientras que los restantes, apuntan a la ubicación de puntos vía como solución a la optimización de los tiempos de ciclo, lo cual confirma la importancia de este paso, dentro del proceso de programación de tareas para el manipulador.

Por otra parte, se tiene que un ahorro en un segundo de tiempo de ciclo puede representar un aumento de la capacidad instalada de las operaciones, y teniendo en cuenta las múltiples aplicaciones que pueden tener los brazos manipuladores en la industria, el impacto de las mejoras obtenidas se refleja en diferentes eslabones de la cadena de abastecimiento, alcanzando uno de los objetivos fundamentales de la gestión logística.

Considerando que el mercado de robots industriales se encuentra en crecimiento⁵, esta propuesta se presenta como un anticipo a la necesidad de resolver el problema de minimización de tiempos de ciclo en entornos robotizados. La intención del proyecto es generar un desarrollo cuyos resultados trasciendan, al demostrar el impacto que genera el análisis y mejora de los sistemas automatizados en la industria, impulsando así el desarrollo de nuevos proyectos de investigación de ingeniería industrial en esta área y su aplicación en ambientes reales donde los resultados sean tangibles en términos económicos.

Finalmente, se considera que es posible una futura generalización del modelo partiendo del estudio de una trayectoria particular; en un primer momento, para las diferentes trayectorias que se manejan en la celda robótica del CTAI, y más adelante, en aplicaciones reales que impliquen el movimiento de un brazo manipulador entre dos puntos para la ejecución de una tarea. Esto sería posible, teniendo en cuenta que este proceso sólo requeriría una parametrización ajustada a las características propias de la tarea y su entorno.

⁵ Tomado de la publicación del IFR Statistical Department .“World robotics 2008”.

4. OBJETIVO GENERAL

Generar una aplicación⁷ que permita la planeación de una familia de trayectorias buscando optimizar las variables tiempo y distancia para la carga del torno desde las diferentes estaciones de la celda de trabajo en el CTAI.

4.1. OBJETIVOS ESPECÍFICOS

- Reconocer y comparar las diferentes soluciones que se han planteado al problema de la optimización de trayectorias para manipuladores en robótica industrial.
- Identificar las variables críticas en el proceso de carga del torno de la celda de trabajo del laboratorio de automatización del CTAI.
- Determinar un algoritmo o heurística que se ajuste a las características del robot manipulador MELFA IV y la tarea objeto de estudio.
- Desarrollar una aplicación que permita posicionar los puntos críticos para la generación de la mejor trayectoria en términos de las variables definidas.
- Comparar los resultados obtenidos con la situación actual para determinar el impacto de los desarrollos.

⁷ Entiéndase por aplicación un programa elaborado en un lenguaje de programación basado en el funcionamiento de un algoritmo específico.

5. MARCO CONCEPTUAL

5.1. ROBÓTICA

“Un robot, es un dispositivo mecánico versátil - puede ser un brazo manipulador, una mano con dedos y juntas múltiples, un vehículo rodante, una plataforma móvil, o la combinación de estos - equipado con actuadores y sensores bajo control de un sistema computarizado. Opera en el entorno del mundo real rodeado de objetos físicos, supeditado a las leyes de la naturaleza, realizando tareas mediante su movimiento en el espacio” [4].

La palabra **robot** es de origen eslavo, en ruso “robota” significa trabajo. Sin embargo, esto no limita la aplicación de los robots al campo industrial, y por tanto, se distinguen dos clases de robots: Los robots de servicio, y los robots industriales.

Los robots de servicio, son aquellos que prestan servicios para los seres humanos o equipos excluyendo de esta definición las actividades de manufactura. Éstos se clasifican en tres grupos: Los robots de servicio al ser humano (seguridad personal, entretenimiento), robots de servicio a equipos (mantenimiento, reparación, limpieza), y los robots con funciones autónomas (transporte, recopilación de datos, supervisión), y todos aquellos que no clasifiquen en los dos grupos anteriores. Algunos ejemplos de este tipo de robots se observan a continuación.



(a)

(b)

(c)

- a) *“MANUS-arm”, Silla de ruedas aprovisionada con un brazo robótico.*
- b) *“Skywash”, Robot equipado para realizar la limpieza de aviones.*
- c) *“Solar Mower”, Podadora de césped con fotoceldas.*

FIGURA 2. Robots de Servicio⁸

Los robots industriales, grupo dentro del que se clasifica el robot objeto de estudio, “forman parte del desarrollo de la automatización industrial, favorecido notablemente por el avance de las técnicas de control

⁸ IFR. International Federation of Robotics. Recuperado el 27 de Marzo de 2009, de <http://www.ifr.org/modules.php?name=News&file=article&sid=14>

por computadora, y contribuyen de manera decisiva a la automatización en los procesos de fabricación de series de mediana y pequeña escala” [5].

Actualmente, la utilización de robots industriales se ha generalizado a múltiples aplicaciones en las que se requiere realizar movimientos repetitivos que demandan fuerza y precisión; aunque no abarquen todas las posibilidades de aplicación de los robots en la industria, algunas de las definiciones más conocidas para un robot industrial son:

“Un robot industrial, es un manipulador multifuncional reprogramable capaz de mover cargas, piezas, herramientas o dispositivos especiales, según trayectorias varias: programado para realizar trabajos diversos”⁹,

“Un mecanismo diseñado para manipular y transportar piezas, herramientas o útiles especiales, por medio de movimientos variables, programados para la ejecución de tareas específicas de manufactura”.¹⁰

“Un manipulador multipropósito de tres o más ejes, controlado automáticamente, reprogramable”.¹¹

“Una máquina programable, de propósito general, que posee ciertas características antropomórficas, es decir, con características basadas en la figura humana”.¹²

De acuerdo a la definición presentada, se tiene que un robot industrial está programado para la realización de una tarea específica, sin embargo, antes de decidir si se debe implementar un robot para que la ejecute, es necesario evaluar algunos aspectos de la actividad tales como [2]:

1. Ambiente de trabajo (seguridad, comodidad)
2. Ciclos de trabajo repetitivos.
3. Dificultad en el manejo de materiales (peso, forma de las piezas)
4. Turnos de trabajo
5. Re-acomodación frecuente del puesto de trabajo
6. Condiciones particulares de la tarea

Una vez analizadas estas condiciones, se puede determinar si la implementación de un robot para la ejecución de la tarea es adecuada.

⁹ RIA (Robot Industry Association).

¹⁰ BRA (British Robotic Association).

¹¹ ISO (Norma 8373)

¹² Mikel Groover (Automation, Production Systems and Computer Integrated Manufacturing).

5.2. ROBOT MANIPULADOR

Un robot manipulador es un robot articulado, comúnmente utilizado en la industria cuya característica principal es la flexibilidad, lo cual lo diferencia de la *automatización fija*, en la medida en que no sólo sus movimientos son programables, sino que también, a través del uso de sensores y la comunicación con otros dispositivos, se puede adaptar a las variaciones de la tarea [5].

5.2.1. SISTEMA BÁSICO DE UN ROBOT MANIPULADOR

Los requerimientos mínimos para la correcta operación de un robot manipulador son:

- El **brazo manipulador (robot)**, consiste en un conjunto de articulaciones mecánicas (eslabones, engranajes, transmisión por cadena o correa), actuadores (motores eléctricos o hidráulicos) y sensores de posición usados en el sistema de control de bucle cerrado.
- El **controlador**, es el dispositivo que recibe las señales de los sensores de posición usados en el sistema de control de bucle cerrado.
- La **unidad de conversión** de potencia, que alimenta los motores que actúan las articulaciones.

Puede contar también con sensores externos como cámaras de video, con los que se controla el movimiento del brazo. [1]

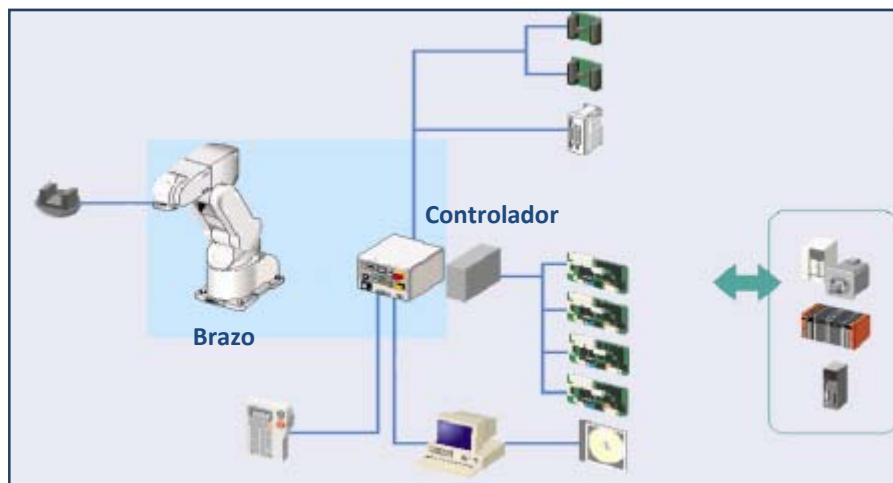


FIGURA 3. Sistema de un robot manipulador [2]

5.2.2. CARACTERÍSTICAS DE UN ROBOT MANIPULADOR

5.2.2.1. MORFOLOGÍA

La morfología del robot, depende de los tamaños y tipos de estos, así como de otros aspectos referentes a las características físicas del mismo. El brazo manipulador de un robot puede dividirse en dos partes: el cuerpo, el cual posiciona el elemento terminal, y la muñeca que lo orienta con más precisión. Generalmente, se asocian tres grados de libertad al sistema del cuerpo, y dos o tres a la muñeca.

La morfología es un concepto importante porque la forma y la estructura de los robots condicionan su funcionamiento, así como su campo de aplicación; para el caso de los brazos manipuladores, esta viene inspirada en el cuerpo humano [2] como se observa en la figura 4:

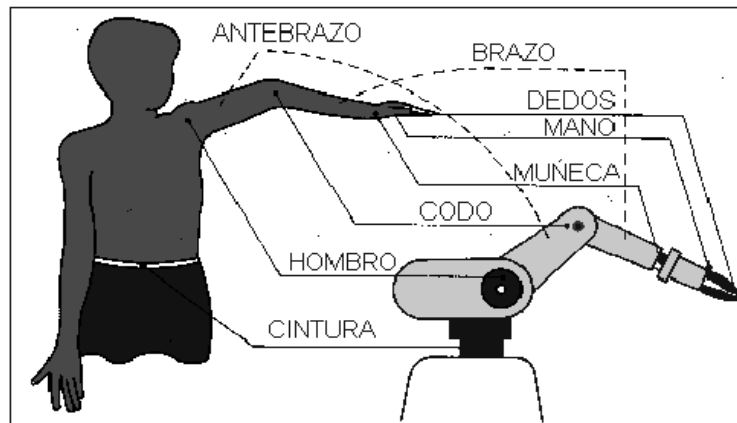


FIGURA 4. Morfología del robot [2]

Eslabones y Articulaciones

Los eslabones son los componentes rígidos del brazo manipulador, los cuales se conectan por medio de articulaciones que permiten el movimiento relativo entre ellos.

Las articulaciones se clasifican según el tipo de movimiento que permiten realizar entre los eslabones del robot, existen dos tipos de articulación que permiten movimientos traslacionales y tres, movimientos rotacionales. (Ver Figura 5)

Generalmente los robots industriales son clasificados de acuerdo a la cantidad de articulaciones (también llamadas juntas) que poseen y a la forma en la cual estas configuran el cuerpo; cada articulación da al robot un **“grado de libertad”** de movimiento, siempre y cuando estas no se hayan dispuesto en posición análoga (una tras otra) caso en el cual únicamente se mejora el alcance de los movimientos sin añadir un grado de libertad adicional. [2]


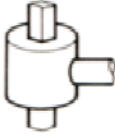
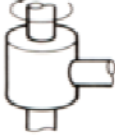


ESQUEMA	ARTICULACION	GRADOS LIBERTAD
	ROTACION	1
	PRISMATICA	1
	CILINDRICA	2
	PLANAR	2
	ESFERICA (ROTULA)	3

FIGURA 5. Tipos de articulaciones comúnmente utilizadas en la construcción de un robot industrial [2]

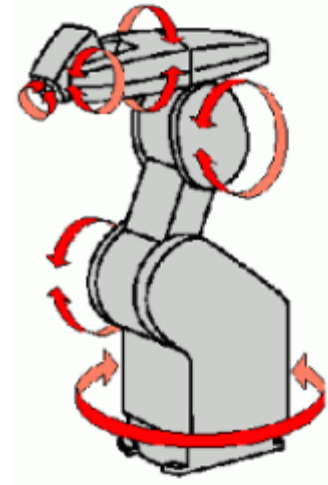


FIGURA 6. Robot de seis grados de libertad [2]

Gracias a los cinco tipos de configuración de articulaciones presentadas anteriormente (Figura 5) existen $5 \times 5 \times 5 = 125$ combinaciones diferentes de articulaciones que pueden ser usadas para el diseño del sistema del cuerpo de un manipulador de tres grados de libertad [6], sin embargo, existen cinco configuraciones básicas a nivel comercial (Figura 7). Cada una de estas, define un área de alcance del robot diferente, la cual puede ser cartesiana, cilíndrica o esférica, que puede ser grande o pequeña según el tamaño de los eslabones.


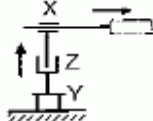


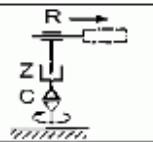


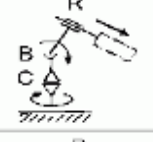

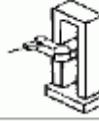
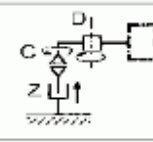


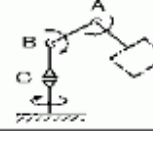

	Configuración	Modelo Cinemático	Configuración área de trabajo
Robot Cartesiano			
Robot Coordenadas Cilíndricas			
Robot Coordenadas Esfericas			
Brazo Robótico Tipo Horizontal			
Brazo Robótico Tipo Vertical			

FIGURA 7. Configuraciones básicas del sistema del cuerpo de un manipulador industrial.¹³

5.2.2.2. ELEMENTOS TERMINALES

Un elemento terminal se define como el órgano de aprehensión sujetado a la muñeca del robot, diseñado y fabricado para efectuar la manipulación o el soporte de la herramienta necesaria para la realización del trabajo asignado.

Los elementos terminales se dividen en dos categorías: elementos de sujeción (pinzas, ventosas, ganchos o palas) y herramientas (pistola para soldadura, soplete, herramientas de corte, láser, entre otros). [2]

5.3. APLICACIONES

Recordando los aspectos que deben ser evaluados para la implementación de un robot (Capítulo 5.3), se tiene que las principales aplicaciones de los robots manipuladores se encuentran en los procesos de la cadena de abastecimiento de empresas manufactureras, donde las tareas son normalmente repetitivas y se manejan cargas pesadas para un operario; además, la implementación de un mecanismo autómatas genera un aumento de la productividad que justifica la inversión.

¹³ Festo. (2000). *Cosimir Robotics Assistant: Workspace*. Traducida por los autores.

Las aplicaciones más frecuentes se encuentran clasificadas en tres categorías (manejo de materiales, operaciones propias del proceso productivo, y ensamble), las cuales se enmarcan claramente en los diferentes eslabones de la cadena de suministro:

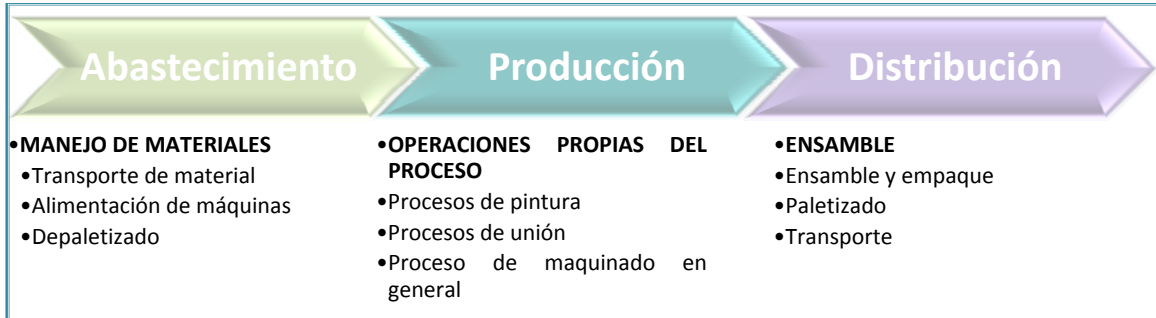


FIGURA 8. Aplicaciones de robots industriales en la Cadena de Abastecimiento. Fuente: Los Autores.

5.3.1. ABASTECIMIENTO: MANEJO DE MATERIALES



FIGURA 9. Desestibado de botes [47]

Son todas aquellas aplicaciones en las que el robot se encarga de mover material de un punto a otro.

- *Carga y descarga de máquinas:* El robot introduce y/o retira las piezas de las máquinas de producción. Típicamente, esta aplicación se da en procesos de fundición a presión, moldeo de plásticos, mecanizado de metales, forja, tratamientos térmicos.
- *Trasporte de material:* El objetivo principal del robot es recoger piezas en una locación y llevarlas a otra. Generalmente, se debe orientar la pieza de diferentes formas durante el proceso.
- *Desestibado:* El robot separa la mercancía que viene paletizada (agrupada), y la lleva al lugar que se requiera según el proceso para el que se necesite.

5.3.2. PRODUCCIÓN: APLICACIONES DONDE SE REALIZAN OPERACIONES DEL PROCESO



FIGURA 10. Robot realizando proceso de inyección de piezas [46]

Son todas aquellas operaciones en las que el robot ejecuta alguna operación sobre la pieza o producto. Las tareas más comunes son soldadura de punto, de arco y pintura con spray, debido a que requieren de un alto nivel de precisión. También se utilizan robots en los procesos de fresado, pulido, cepillado, corte en frío, corte con láser y mecanizado en general.

5.3.3. DISTRIBUCIÓN: ENSAMBLE



FIGURA 11. Robot paletizado sacos [48]

El ejemplo más común de esta aplicación, es el proceso de estibado, que consiste en disponer mercancía sobre una estiba para su almacenaje y transporte, para lo cual el robot debe recalcular las posiciones a las que se dirige cada vez que cumple un ciclo.

En esta área la aplicación más frecuente de robots se presenta cuando un grupo de productos o modelos similares se producen en la misma celda o línea de ensamble, esto aplica para motores eléctricos, pequeños aparatos eléctricos y otros productos de ese tipo.

Las operaciones de inspección para robots incluyen asegurar que un proceso ha sido completado, garantizar que las partes a ensamblar se situaron correctamente e identificar imperfectos en los productos terminados.

5.4. ROBOT MITSUBISHI MELFA IV

El robot Mitsubishi Melfa IV de la celda de trabajo del CTAI cuenta con seis grados de libertad; es el responsable del manejo de las piezas en el proceso de estibado, y la alimentación de las máquinas CNC.

Características del robot:



FIGURA 12. Robot Mitsubishi Melfa Iv [7]

- *Alcance Máximo* 418 mm
- *Carga Máxima:* 1 Kg
- *Precisión* +/- 0.02mm
- *Rapidez Max.* 2m/s
- *Medida de encoder* Absoluta
- *Drive* AC servo motors with harmonic drive
- *Sistema Operativo multitarea* Procesador 64 Bit RISC
- *Número máx. de tareas* 32
- *Número máx. de líneas de programa* 5000
- *Número de programas* 88
- *I/O Digitales* 16/240

Como elemento terminal, el robot cuenta con una pinza (*gripper*) de funcionamiento electro neumático; en la cual a su vez se encuentra dos sensores inductivos que controlan si la pinza está abierta o cerrada.

5.5. TIPO DE TRAYECTORIAS QUE SIGUE UN ROBOT MANIPULADOR

5.5.1. EJECUCIÓN DE MOVIMIENTOS DEL ROBOT MELFA-IV

Hay tres posibilidades para realizar movimientos libres con el robot: por medio del movimiento independiente de cada una de las juntas, por movimiento sobre los 3 ejes de coordenadas X, Y, Z y por el movimiento relacionado con el terminal de sujeción del robot (pinza). [6]

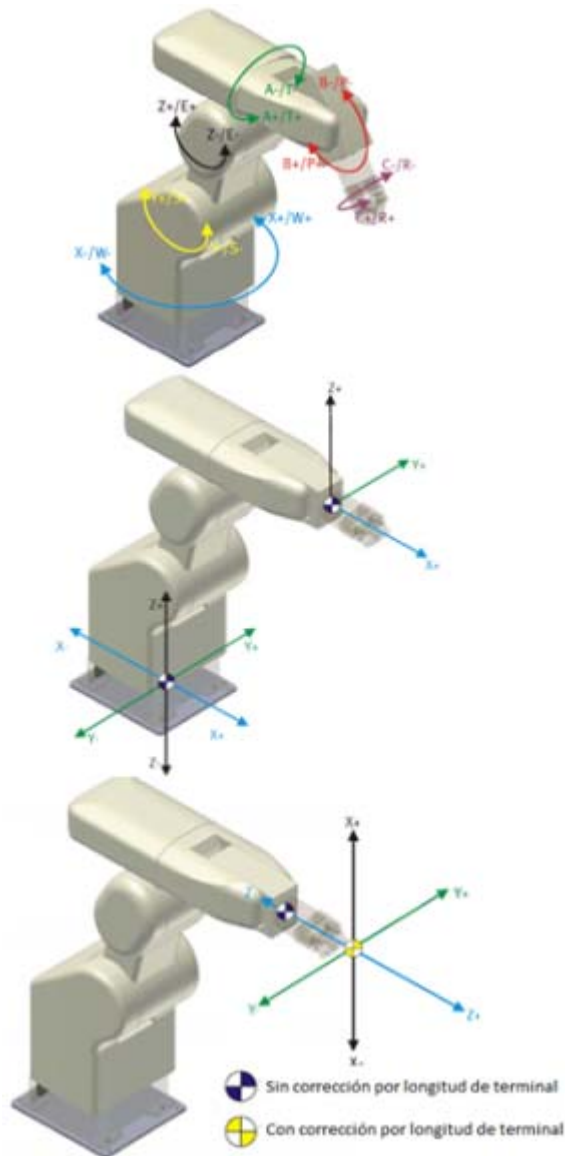


FIGURA 13. EJECUCIÓN DE MOVIMIENTOS DEL ROBOT [6]

JOINT Jog

Mueve el robot controlando cada articulación individualmente.

XYZ Jog

Mueve todos los ejes del robot siguiendo un desplazamiento lineal a través de los ejes cartesianos (*world coordinate axes*).

TOOL Jog

Mueve el robot a través del sistema de coordenadas básico del robot, cuyo origen está definido por el TCP (Tool Centre Point), siendo este generalmente la pinza.

5.5.2. POSICIONES

La posición del robot se define como una configuración de los ángulos de cada una de las juntas y articulaciones del mismo, que determina la ubicación de la punta del efector final en un punto dado en el espacio con una orientación específica.

Con el fin de programar las tareas a realizar es preciso almacenar las posiciones por las cuales el robot se desplazará, para lo cual se utiliza una matriz en la que se registran los diferentes ángulos en grados de las juntas en forma consecutiva: (J1, J2, J3, J4, J5, J6). Otra forma de almacenar las posiciones consiste en utilizar una matriz donde se registre la posición y orientación de la punta del efector final. La posición se refiere a la ubicación en el espacio cartesiano XYZ con respecto a un punto de origen previamente configurado al instalar el robot, cuyas unidades están dadas en mm. Por otra parte, la orientación hace referencia a la rotación de la herramienta respecto al origen y se expresa por medio de los parámetros A, B, C medidos en grados. De esta manera, se obtiene el arreglo (X, Y, Z, A, B, C).

5.5.3. GENERACIÓN DE TRAYECTORIAS QUE SIGUE EL ROBOT

Al hablar del proceso de generación de trayectorias¹⁴ del robot se hace referencia a los procedimientos internos del sistema que definirán de forma estricta los movimientos del brazo manipulador en función del movimiento de las articulaciones en el espacio de trabajo, de acuerdo a los parámetros que el programador considere necesarios para la ejecución de la tarea específica (puntos vía, formas del movimiento, velocidades y aceleraciones) [1].

Una manera de incluir detalle en una descripción de ruta es proporcionar una secuencia de puntos vía (puntos intermedios entre las posiciones inicial y final); que se refieren a tramas que proporcionan orientación y posición para la herramienta con relación a la estación. De acuerdo con esto, los movimientos de un manipulador se consideran como los movimientos de la trama herramienta {T} relativos a la trama estación {S} (Figura 14). El problema básico en la generación de trayectorias, es entonces el movimiento del manipulador de la trama {T_{inicial}} a {T_{final}}. La anterior definición permite especificar y planear los movimientos en relación con una estación móvil haciendo que en tiempo de ejecución la definición de {S} cambie con el tiempo.

En operaciones de “coger-y-colocar” solo se hace importante conocer qué trayectoria sigue la herramienta de un punto a otro si se desea optimizar el tiempo y evitar colisiones con posibles objetos en el espacio de trabajo. En otras situaciones, como la soldadura o el corte de materiales, la trayectoria está dada por las especificaciones de la tarea.

¹⁴ Trayectoria se define como la relación temporal de posición, velocidad y aceleración para cada uno de los grados de libertad del robot. [1].

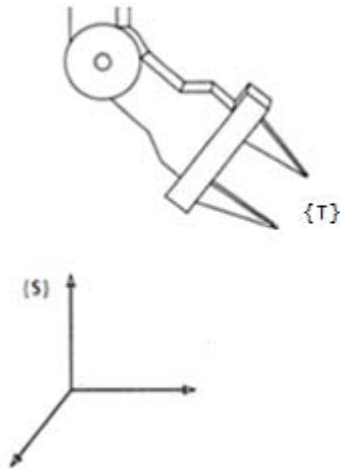


FIGURA 14. Movimiento herramienta relativo a la trama estación. Fuente: [5] Modificado por los autores

5.5.3.1. FUNCIÓN DE TIEMPO QUE ESPECIFICA EL MOVIMIENTO

Dado que los movimientos bruscos tienden a producir un mayor desgaste en el mecanismo y producen vibraciones al excitar las resonancias en el manipulador, es conveniente que el movimiento de este sea uniforme¹⁵, para lo cual es posible usar cualquier función continua de tiempo que pase a través de los puntos vía¹⁶ para especificar la forma exacta de la ruta. A continuación se explican brevemente los métodos más simples para este fin:

Esquemas en el espacio de articulación

Las formas de las rutas en espacio y tiempo se describen en términos de funciones de ángulos de articulación, por tanto, cada uno de los puntos vía se “convierte” en un conjunto de ángulos de articulación deseados mediante la aplicación de la cinemática inversa¹⁷. Después de esto, se encuentra una función uniforme para cada una de las n articulaciones que pasan a través de los puntos vía y que terminan en el punto de destino¹⁸, conformando la trayectoria final.

Se dice que la función de ángulo de articulación deseada es independiente para cada una de las articulaciones dado que “el tiempo requerido por segmento es el mismo para cada articulación, de manera que todas las articulaciones llegarán al punto vía al mismo tiempo, produciendo por consecuencia la posición cartesiana deseada de $\{T\}$ en cada punto vía”.¹⁹

¹⁵ Craig, J. J. (2006). *Robótica* (3ra. ed.). Prentice Hall. Pág. 202.

¹⁶ Recordar que un punto vía (o punto crítico) se refiere a la configuración que toman las articulaciones del robot en las que la herramienta se encuentra en una posición y orientación específica en el espacio en un momento dado. (Ver Planteamiento del problema)

¹⁷ La cinemática inversa hace referencia a la transformación del espacio cartesiano al espacio de articulación.

¹⁸ CRAIG, Op. Cit., Pág. 203.

¹⁹ Ibid.

Existen muchas funciones uniformes $\theta(t)$ que pueden usarse para interpolar el valor de las articulaciones, teniendo en cuenta que en t_0 se tiene la posición inicial de cada articulación y en t_f la posición deseada²⁰. Se presentan a continuación algunas de estas funciones:

Polinomios Cúbicos

Para crear un solo movimiento uniforme son evidentes cuando menos cuatro restricciones sobre $\theta(t)$, las cuales corresponden a las posiciones y las velocidades finales e iniciales:

- $\theta(0) = \theta_0$
- $\theta(t_f) = \theta_f$
- $\dot{\theta}(0) = 0$
- $\dot{\theta}(t_f) = 0$

Estas restricciones pueden satisfacerse mediante un polinomio de por lo menos tercer grado (gracias a que este tiene cuatro coeficientes), determinando de manera única una ecuación cúbica específica, de la forma:

$$\theta(t) = a_0 + a_1t + a_2t^2 + a_3t^3,$$

De forma tal que la velocidad a lo largo de la ruta es

$$\dot{\theta}(t) = a_1 + 2a_2t + 3a_3t^2$$

Reemplazando las restricciones anteriormente descritas y despejando el valor de los coeficientes a_i , se obtienen los polinomios cúbicos que conectan cualquier posición de ángulo de articulación inicial con cualquier posición final deseada.

Las anteriores restricciones aplican para el caso en que la articulación inicia y termina con velocidad cero, lo cual es válido para la ruta si se desea que el manipulador se detenga en cada punto vía; comúnmente esto no es lo que se busca, por tanto debe considerarse el problema de calcular ecuaciones cúbicas que conecten entre sí los valores de los puntos vía para cada articulación de una manera uniforme. Esto se logra reescribiendo las ecuaciones anteriores teniendo en cuenta que las velocidades en cada extremo son diferentes de cero.

Existen varias formas en las que podría especificarse la velocidad deseada en los puntos vía:

- ◆ El usuario especifica la velocidad deseada en términos de velocidades cartesianas lineal y angular de la trama herramienta en ese instante.
- ◆ El sistema selecciona automáticamente las velocidades mediante la aplicación de una heurística adecuada, o de forma tal que la aceleración en los puntos vía sea continua.

La primera opción requeriría de un trabajo dispendioso para el usuario, por lo cual un sistema conveniente debería incluir las opciones automáticas.

²⁰ Ibid.

- *Polinomios de mayor orden*

Si se desea especificar por ejemplo posición, velocidad y aceleración al inicio y final de cada segmento de ruta, se hará necesario un polinomio de quinto grado,

$$\theta(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5,$$

Siendo las restricciones un conjunto de seis ecuaciones con seis variables desconocidas.

- *Segmentos lineales con transición parabólica*

Debido a que la función de posición debe ser suave, es decir, no tener picos o cambios bruscos en la pendiente, se hace necesario transformar los segmentos de la trayectoria en los que se presente esta situación, ya que de lo contrario dichos picos significarían que el manipulador experimente una aceleración infinita durante un tiempo cero, lo cual es imposible. El método de transición parabólica consiste en interpolar linealmente para realizar un movimiento desde la posición de articulación actual hasta la posición final, reemplazando las regiones inicial y final del movimiento por segmentos basados en una función cuadrática cuya segunda derivada será una constante y por tanto, no se presentará el caso de una aceleración infinita (Ver Figura 16).

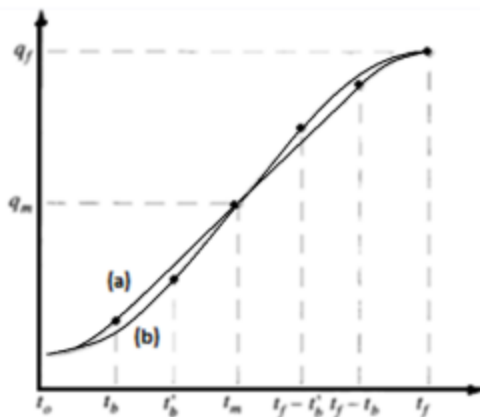


FIGURA 16. Interpolaciones lineales con enlaces parabólicos. [40]

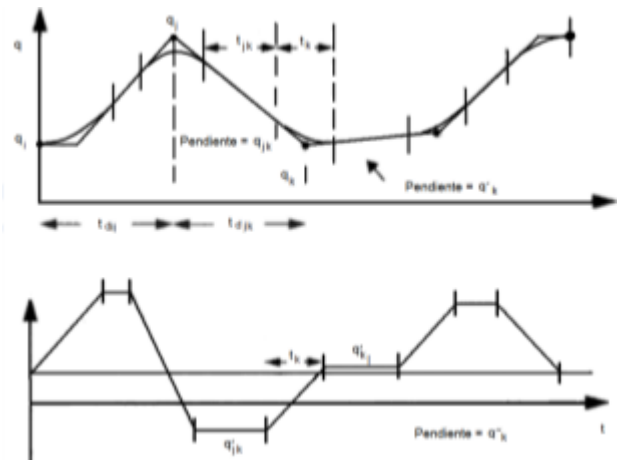


FIGURA 15. Segmentos lineales con enlaces parabólicos y múltiples puntos de paso [40]

Cabe anotar que aunque el movimiento de cada articulación en este esquema es lineal, el efector final no se mueve en general en una línea recta en el espacio.

Para la aplicación de este método deben cumplirse las siguientes condiciones:

1. La curva debe ser simétrica con respecto al punto medio en el tiempo.
2. La velocidad del segmento lineal debe tener el mismo signo que $\theta_f - \theta_0$.
3. En los puntos inicial y final de la trayectoria, la velocidad y aceleración deben ser cero

A continuación se muestran las ecuaciones en forma general para los segmentos parabólicos:

$$\text{Posición} \quad \theta(t) = \alpha + \beta t + \gamma t^2$$

$$\text{Velocidad} \quad \dot{\theta}(t) = \beta + 2\gamma t$$

$$\text{Aceleración} \quad \ddot{\theta}(t) = 2\gamma = a$$

Donde la aceleración "a" debe ser menor que la máxima especificada por el fabricante.

Para $t_0=0$

$$\theta(0) = \alpha = \theta_0$$

$$\dot{\theta} = \beta = 0$$

Para $t = t_b$ (punto de unión del segmento parabólico con el segmento lineal)

$$\dot{\theta} = 2\gamma t_b = V$$

$$\gamma = V/(2t_b)$$

Es decir,

Se tiene entonces como resultado la ecuación para el segmento parabólico:

$$\theta(t) = \theta_0 + (V/(2t_b))t^2$$

$$\theta(t) = \theta_0 + \frac{1}{2}at^2$$

Al hacer uso de los segmentos lineales con mezclas parabólicas, hay que tener en cuenta que en realidad no se llega a los puntos vía, a menos que el manipulador se detenga. "Si se deseará que le manipulador pase *exactamente* a través del punto vía sin detenerse, puede acomodarse esta especificación con la misma formulación de antes, adicionando una condición bajo la cual el sistema reemplace automáticamente el punto vía a través del cual se desea pasar, por dos *seudopuntos vía*, uno en cada lado del original"²¹; el punto vía se encontrará ahora en la región lineal de la ruta que pasa por los dos *seudopuntos vía* como se observa en la Figura 17. Otra forma de lograr esto, es usa el término **punto de paso** para especificar un punto de la ruta por el cual se fuerza al manipulador para que pase a través suyo.

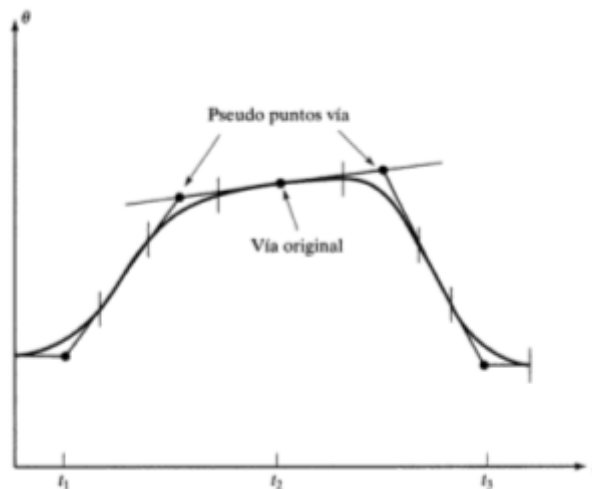


FIGURA 17. Uso de *seudopuntos vía* para crear un "punto de paso". [5, P. 217]

²¹ Ibid. Pág. 216.

Esquemas en el espacio cartesiano

Al generar una trayectoria haciendo uso de los esquemas del espacio de articulación se asegura que la herramienta se mueva desde un punto de inicio a un punto de destino siguiendo una función en el tiempo que no tiene en cuenta previamente la forma del movimiento de la herramienta misma en el espacio. En ocasiones esto no es adecuado debido a que para el desarrollo de muchas tareas o movimientos es necesario especificar la forma espacial de la ruta tomada por la herramienta entre los puntos vía, para lo cual, se utiliza entonces el esquema en el espacio cartesiano, haciendo posible definir diferentes tipos de desplazamiento para la herramienta como movimiento circular o sinusoidal, sin embargo, el más común es el movimiento rectilíneo.

Para que el extremo de la herramienta se mueva linealmente entre dos puntos, se puede aplicar el método de segmentos lineales con transición parabólica a las coordenadas x , y , z , de modo que las componentes de posición cambien linealmente (por interpolación lineal) y por lo tanto la herramienta se mueva a lo largo de una recta en el espacio. Es conveniente entonces, que cada punto vía sea especificado en forma relativa a la estación, para lo cual se define un sistema coordenado fijo $\{S\}$ en la estación del manipulador. Entonces, un punto vía A se define como la posición del efector final en dicho sistema coordenado, indicada por el vector ${}^S P_{A-ORG}$ (Ver Figura 18).

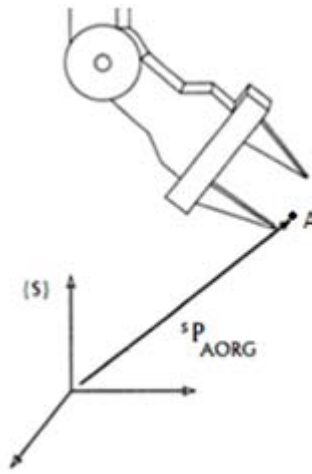


FIGURA 18. VECTOR DE POSICIÓN [5]

Es preciso recordar que cada punto vía es definido como una trama, es decir que además de la descripción de la posición de la herramienta, también considera la orientación de la misma. Desafortunadamente, el método de interpolación lineal no puede aplicarse directamente a la orientación de la herramienta [5] como se hace con las componentes de posición, debido a que dicho procedimiento no necesariamente resulta en una matriz de orientación válida. Para solucionar este problema, se hace uso de la representación “ángulo-eje-equivalente”, donde la orientación se describe a partir de un vector ${}^A K$ en una dirección general (es decir no necesariamente en dirección unitaria \hat{x} , \hat{y} o \hat{z}).

Para describir la orientación, se define inicialmente que los sistemas coordenados de la herramienta {A} y de la estación {S} coinciden, después, el sistema A es girado θ_A alrededor del vector unitario ${}^A\hat{K}$, obteniendo así la orientación relativa buscada de {A} con respecto a {S}.²²

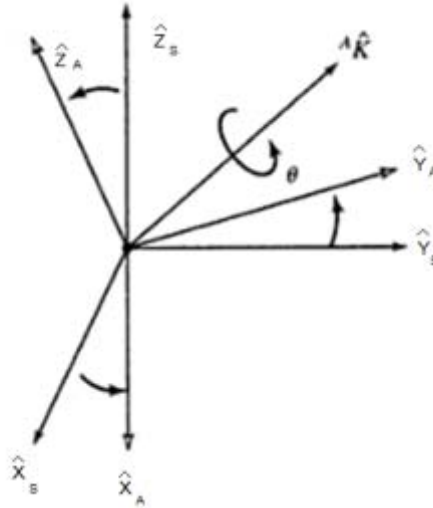
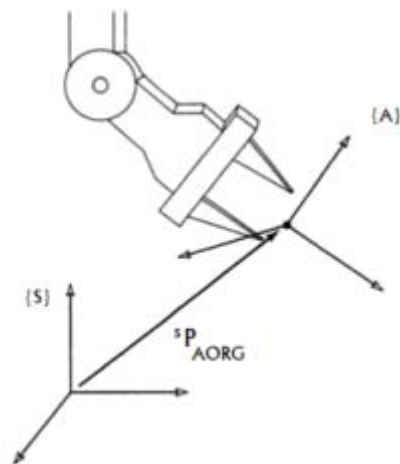


FIGURA 19. Representación ángulo-eje equivalente [5]

Al multiplicar el vector ${}^A\hat{K}$ por su ángulo de rotación, se obtiene el vector compacto de 3×1 AK (sin gorro). Si combinamos esta representación de la orientación con la representación de la posición cartesiana de 3×1 mencionada, tendremos una representación de 6×1 de la posición y orientación cartesianas. Se utiliza el símbolo χ para representar este nuevo vector.



$${}^S\chi_A = \begin{bmatrix} {}^S P_{AORG} \\ {}^S K_A \end{bmatrix}$$

FIGURA 20. REPRESENTACIÓN POSICIÓN Y ORIENTACIÓN DE LA HERRAMIENTA [5]

²² Según el Teorema de rotación Euler, existe ${}^A\hat{K}$ y θ_A para cualquier orientación de A relativa a S [5, p. 46]

Existen diferentes problemas geométricos que limitan la aplicación de los métodos en espacio cartesiano. Cabe mencionar:

- Puntos intermedios inalcanzables: es muy posible que no todos los puntos distribuidos en una línea recta que conecten los puntos inicial y final, se encuentren en el espacio de trabajo del manipulador.
- Velocidades articulares altas cerca de las singularidades: para mantener la velocidad y forma del movimiento de la herramienta en áreas cercanas a puntos singulares, las articulaciones pueden requerir moverse a velocidades fuera de las capacidades del manipulador, haciendo imposible realizar dicho movimiento.
- Puntos alcanzables solo con diferentes orientaciones: Existen algunos puntos a los cuales solo es posible acceder con determinadas configuraciones, si se especifica una orientación distinta, el controlador evitará ejecutar el movimiento.

Para manejar estos problemas, se deben usar rutas en espacio cartesiano únicamente cuando sea realmente necesario para la aplicación.

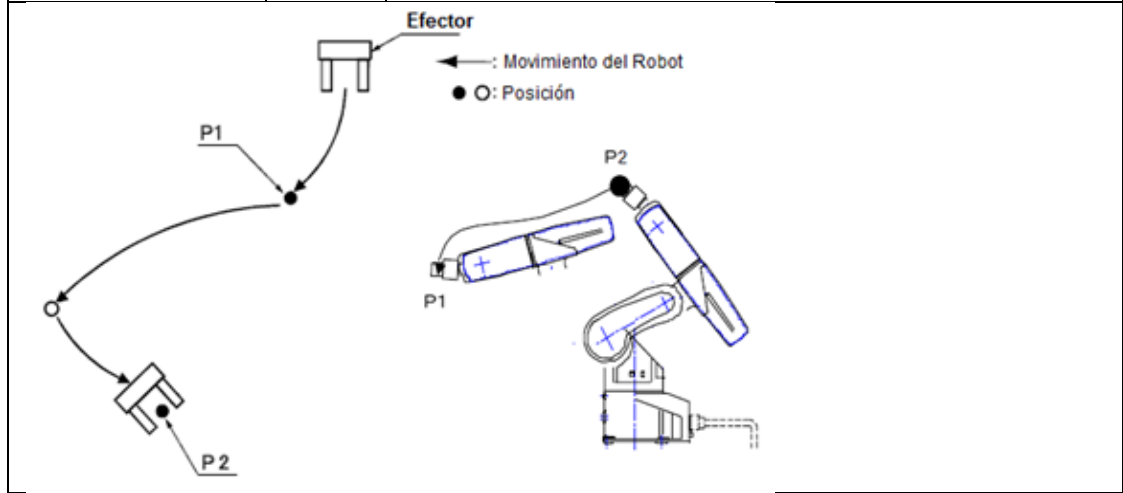
5.5.4. PLATAFORMA DE PROGRAMACIÓN

El robot Mitsubishi MELFA RV-2A se programa por medio del software COSIMIR® PROFESSIONAL, en el lenguaje de programación MELFA BASIC IV. Este lenguaje, desarrollado por Mitsubishi, está basado en BASIC y por lo tanto integra sentencias conocidas como los comandos de ciclos, condicionales y en general aquellos que determinan el flujo del proceso (*IF, GOTO, RETURN, ELSE, entre otras*).

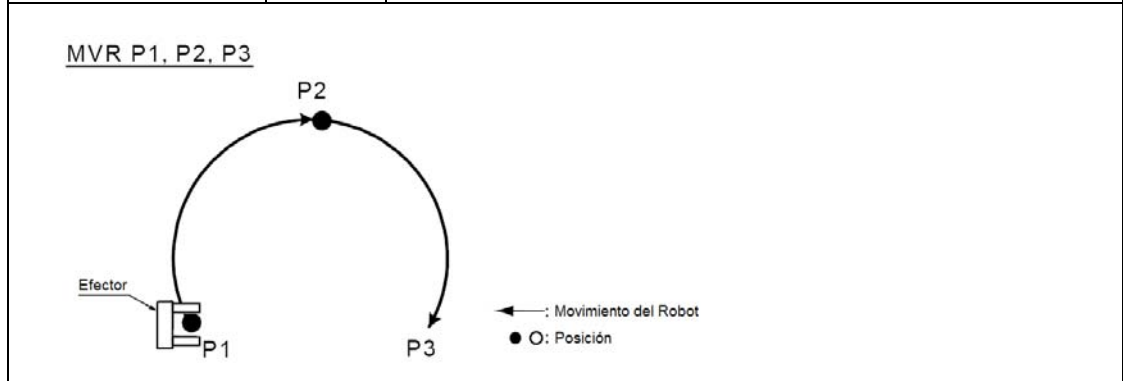
Para la correcta ejecución el programa en MELFA BASIC IV, debe generarse un archivo plano de texto cuya extensión es *.MB4*, y una lista de posiciones (archivo de extensión *.POS*) donde se encuentren registradas las posiciones (o puntos vía) a las que se dirigirá el robot.

5.5.5. DESCRIPCIÓN DE LOS COMANDOS

<p>Move</p>	<p>MOV</p>	<p>El robot se mueve a la posición designada con base en las juntas formando trayectorias curvas. La trayectoria que siga la herramienta entre el punto inicial y el de destino es calculada por el controlador por medio de la interpolación de los ángulos inicial y final de cada articulación buscando que el movimiento sea lo más “suave” posible. [3]</p>
--------------------	-------------------	--



<p>Move Round</p>	<p>MVR MVR2 MVR3</p>	<p>El robot se mueve a lo largo de un arco formado por la interpolación circular entre diferentes puntos de tránsito. Para construir el círculo, deben especificarse 2 puntos con los que se forme la figura y por medio de un punto de referencia se señala la dirección del arco. También puede especificarse 2 puntos, inicial y final, y una posición adicional que represente el centro de la circunferencia.</p>
--------------------------	-------------------------------------	--



Move Straight	MVS	El manipulador realiza el movimiento a la posición designada por medio de la interpolación lineal formando trayectorias rectas en el espacio.

Continuous	CNT	Designa un movimiento continuo para la interpolación. Es posible reducir el tiempo de operación haciendo uso de éste comando.

FIGURA 21. Descripción de los comandos de programación. Fuente: Los Autores

5.6. PROCEDIMIENTO PARA LA PROGRAMACIÓN DE UN ROBOT INDUSTRIAL

El procedimiento que se debe seguir para la programación de un robot dada una tarea específica es el siguiente:

1) Definir la tarea a realizar

Una vez establecido el objetivo del robot, debe determinarse una secuencia a seguir, con el fin de generar una estructura para el programa, esto es lo que se conoce como “*sequence plan*”, el cual puede realizarse en un diagrama de flujo.

2) Definición de Puntos Vía

Para un robot industrial que realiza tareas de manipulación de piezas, los puntos vía son aquellos en donde el robot comienza y termina una trayectoria, además de todos aquellos puntos que le permitan al robot crear trayectorias libres de obstáculos. La definición de estos puntos es crítica para el proceso ya que pueden influir en la eficiencia de la ejecución de las tareas.

El robot generalmente puede ser llevado a las posiciones de puntos vía de dos formas diferentes, por medio de la caja de enseñanza o *Teach Box*, en la cual se podrán almacenar las posiciones y programas; o por medio de un software como COSIMIR® PROFESSIONAL, en el cual se almacenan las posiciones y se crean los programas a ejecutar.

3) Escribir el programa

El programa del robot se puede describir como la traducción del plan de secuencia diseñado anteriormente al lenguaje de programación del mismo.

A continuación se ilustra el proceso en un diagrama de flujo.

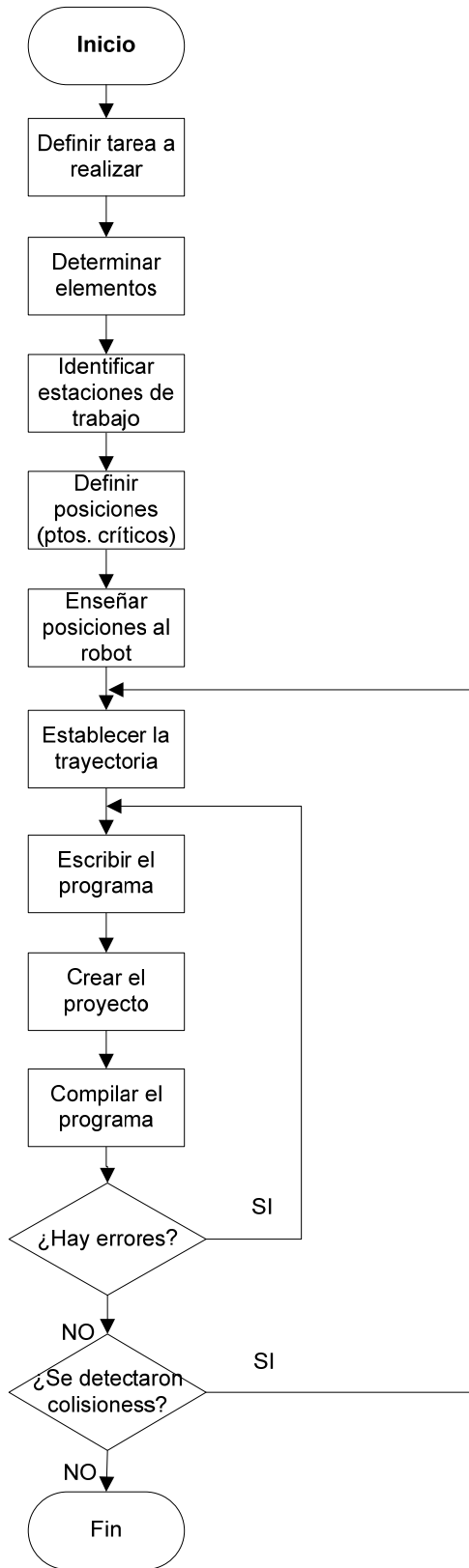


DIAGRAMA 1. DIAGRAMA DE FLUJO PARA LA CONFIGURACIÓN DE UN PROYECTO DE ROBÓTICA. Fuente: Los Autores

5.7. MÉTODOS DE OPTIMIZACIÓN MATEMÁTICA

El problema de generación de trayectorias consiste básicamente en encontrar la mejor configuración de puntos vía, que permitan al robot ir de un punto inicial a un punto final en términos de tiempo, distancia y facilidad para ejecutar el movimiento. Desde esta perspectiva, es entonces posible plantear una analogía de este problema con los problemas de ruteo de vehículos en logística, donde se busca encontrar la mejor configuración de clientes a visitar bajo las mismas condiciones, con restricciones adicionales de capacidad y demanda de acuerdo con el problema que se desee manejar.

Dada la naturaleza de estos problemas, considerados como computacionalmente difíciles (NP-Hard)²³, suele hacerse uso de meta-heurísticas²⁴ para encontrar una solución que, si bien puede no ser la solución óptima del problema, se aproxima lo suficiente a esta en un tiempo razonable.

Se realizó una revisión de problemas tratados en diferentes campos tales como robótica móvil, robótica industrial y problemas de ruteo logístico (ANEXO A), este último considerando que el problema de generación de trayectorias representa un problema de minimización de ruta en sí mismo, como se expuso anteriormente.

5.8. DEFINICIÓN DEL ESPACIO DE TRABAJO DEL ROBOT

Con el fin de hacer posible el proceso de generación de trayectorias es necesario construir una representación del espacio de trabajo del manipulador que relacione el sistema del robot con el entorno físico en el que se mueve. Como se mencionó en el capítulo 5.2 al tratar la morfología del robot, el espacio de trabajo está definido como aquel que el robot puede alcanzar por medio de todas las posibles configuraciones de sus articulaciones (Figura 7).

Es posible representar matemáticamente este espacio físico de forma directa y construir, basándose en dicha representación, modelos que permitan la planeación del movimiento del manipulador, sin embargo, debido a la complejidad de estos modelos es común implementar como alternativa para la solución de este problema el concepto de *Configuration space (c-space)* o espacio de configuración.

Latombe [7] define que la construcción del espacio de configuración consiste básicamente en representar el robot como un punto, modificando el espacio adecuadamente al establecer que los obstáculos ‘crecen’ incluyendo el espacio ocupado por todas las configuraciones del robot que generen una colisión de cualquier parte del cuerpo físico del mismo con algún obstáculo (Figura 22). Por medio de esta representación se simplifica el problema de generación de trayectorias al hacer explícitas las restricciones de movimiento del manipulador.

²³ Problemas cuyo número de decisiones factibles y esfuerzo computacional crece exponencialmente a medida que aumentan las variables.

²⁴ Las heurísticas clásicas son algoritmos exactos o aproximados que realizan una exploración limitada del espacio de búsqueda y típicamente producen soluciones buenas en poco tiempo. Las meta-heurísticas son un conjunto de conceptos, aplicables a un gran número de problemas distintos, diseñados para guiar a otras heurísticas subyacentes específicas para el problema hacia regiones prometedoras del espacio de búsqueda. Realizan así una exploración más profunda en busca de soluciones de mayor calidad pero a expensas de un mayor tiempo de procesamiento [42]

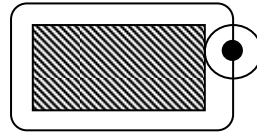


FIGURA 22. OBSTÁCULO "AGRANDADO" [7, P. 89]

De acuerdo con Latombe [7] los métodos de representación del espacio de trabajo se dividen en tres grandes categorías:

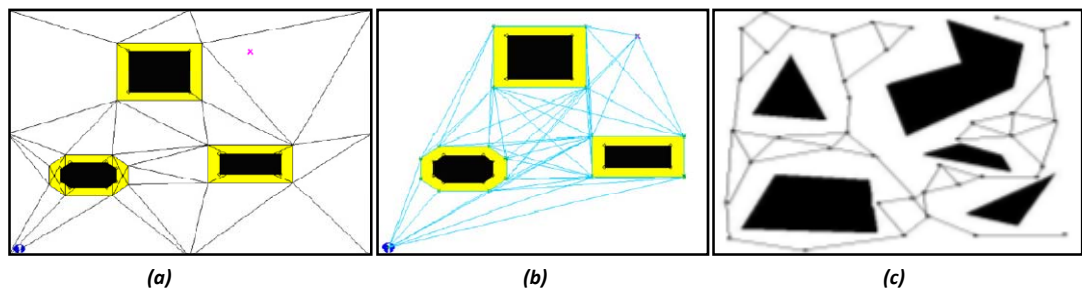
- **Road Map or Skeleton methods.** [8]

Estos métodos consisten en reducir el espacio de configuración libre a una red de líneas por las cuales el algoritmo de planeación de trayectorias debe llevar al robot desde un punto inicial hasta uno final buscando puntos medios entre estos. El espacio libre entre obstáculos puede ser representado de diferentes formas como gráficos de visibilidad, diagramas de Voronoi o mapas probabilísticos.

- *Gráficos de visibilidad:* Consiste en una gráfica de conexiones no dirigidas entre los nodos que puedan ser unidos por líneas rectas sin intersectar los obstáculos. Los nodos que permiten formar la red son los vértices de los obstáculos, el punto inicial y el punto final. (Figura 23.a)

- *Diagramas de Voronoi:* Son usados principalmente para generación de trayectorias en 2D y se limitan a espacios de configuración simples. Se define una función continua en el espacio de configuración. Los nodos se determinan cuando existen dos puntos cuya distancia mínima con el obstáculo es la misma. Se completan los posibles caminos para alcanzar el punto final y así como en los grafos de visibilidad se busca un camino entre estos. (Figura 23.b)

- *Probabilistic Roadmaps:* Es uno de los métodos más promisorios actualmente. El PRM busca en el espacio por puntos libres, tras lo cual un planeador local conecta estos puntos en una gráfica de puntos válidos. Este método ha sido aplicado exitosamente en diferentes casos, sin embargo, debido a su naturaleza probabilística es difícil de analizar y no se conoce información sobre su convergencia. La complejidad del PRM depende en gran medida de la complejidad y dimensión del espacio. El método aún tiene problemas en cuanto a la suavidad de las trayectorias y a movimientos redundantes dentro de las mismas. (Figura 23.c)



- a) *Gráficos de visibilidad. Los nodos que forman la red son los límites de los obstáculos.* [9]
- b) *Diagramas de Voronoi. Se realiza una triangulación entre vértices.* [9]
- c) *Probabilistic Roadmaps. Conecta puntos válidos.* [10]

FIGURA 23. ROAD MAPS OR SKELETON METHODS

- **Descomposición de la celda:**

Es uno de los métodos más extensamente estudiados, este utiliza la teoría gráfica para encontrar una trayectoria valida u óptima, el espacio libre se descompone en un conjunto de celdas simples y la adyacencia entre ellas se determina formando una gráfica no dirigida a partir de las celdas libres (donde no se encuentran obstáculos). A continuación, se determina la trayectoria haciendo uso de algoritmos de búsqueda.

El método se puede aplicar de forma exacta o aproximada, en la descomposición exacta se abarca todo el espacio amoldando la forma y tamaño de las celdas de acuerdo a los obstáculos, mientras que en el método aproximado se utilizan celdas de forma y tamaño determinadas. Un ejemplo de esto es la descomposición por *quadtrees*, donde cada celda se subdivide progresivamente en 4 celdas de igual tamaño, empezando por el espacio total. Las celdas contengan en su totalidad obstáculos o se encuentren vacías, se definirán como ocupadas o libres respectivamente y no se subdividirán posteriormente. El procedimiento se lleva a cabo hasta que las celdas estén completamente definidas o se alcance una resolución (tamaño de celda) máxima predefinida.

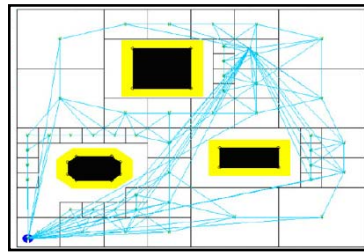


FIGURA 24. DESCOMPOSICIÓN DE CELDA. *QUADTREES* [9]

- **Campos Artificiales (APF Artificial Potential Fields)**

Se trata de construir un campo potencial artificial en el que el punto final posee un coeficiente de atracción (un potencial mínimo global) y los obstáculos son caracterizados por coeficientes de repulsión (máximos locales). Los métodos que utilizan APF son veloces, sin embargo, de acuerdo con Latombe [7], este tipo de algoritmos requieren de poderosas heurísticas que apoyen la búsqueda de posibles caminos debido a que es usual que encuentre soluciones en mínimos locales en lugar de llegar al óptimo, razón por la cual requieren de gran capacidad computacional. Los algoritmos basados en APF pueden ser usados cuando el robot es representado como un punto y el espacio es relativamente despejado.

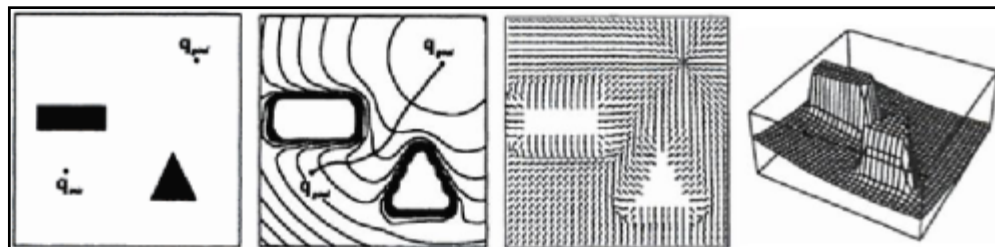


FIGURA 25. CAMPOS PORTENCIALES ARTIFICIALES (*APF POTENTIAL FIELDS*) [7]

Tan solo algunos de los métodos mencionados anteriormente han sido aplicados en robots manipuladores con resultados exitosos. Dado que la formulación del *c-space*²⁵ es compleja y requiere un alto grado de memoria computacional, el número de grados de libertad de los robots manipuladores incrementa el tiempo computacional exponencialmente.

En el cuadro anexo se presenta un resumen de algunas investigaciones anteriores basadas en estos métodos (ver Anexo B).

El primer paso para generar el modelo del espacio de configuración (*c-space*), es definir el espacio físico de trabajo (*workspace*), para lo cual pueden existir varias alternativas. Una de ellas es construir un modelo CAD, sin embargo, esto supone algunas desventajas:

- El proceso es largo debido a que implica la medición de la geometría y ubicación relativa de cada obstáculo.
- Son necesarios procedimientos que relacionen el espacio cartesiano con las configuraciones del robot, es decir, habría que resolver la cinemática inversa²⁶ para todos los puntos.
- El proceso de identificar las colisiones del robot manipulador con obstáculos para las articulaciones diferentes al efector final aumenta el grado de complejidad.

Una alternativa más sencilla, consiste en guiar el manipulador manualmente por el área libre del espacio de trabajo, registrando las configuraciones que permitan definir geoméricamente los puntos donde pueden generarse colisiones. De esta forma, se tienen en cuenta todas las articulaciones y al mismo tiempo se evitan los procedimientos de cinemática inversa. La confiabilidad de esta alternativa depende de la rigurosidad con la que se recorra el espacio de trabajo.

²⁵ La definición fue tratada en el capítulo 5.8

²⁶ Recuerde que la cinemática inversa hace referencia a la transformación del espacio cartesiano al espacio de articulación.

6. SELECCIÓN DEL ALGORITMO DE OPTIMIZACIÓN

De acuerdo con Kyung min Han [11] es posible considerar como bueno un algoritmo para este tipo de problemas (ruteo), cuando este es capaz de identificar la posición instantánea del robot, evadir obstáculos, y determinar un camino hacia el objeto. Por lo tanto, las características más importantes a considerar en la selección de un algoritmo para el problema de generación de trayectorias son [11]:

1. *Eficiencia*

Hace referencia al tiempo de ejecución del algoritmo, se busca que encuentre soluciones en un tiempo razonable.

2. *Eficacia en la búsqueda de soluciones*

Se busca que el algoritmo encuentre las soluciones cercanas al óptimo, evitando que el robot realice movimientos innecesarios o se detenga en mínimos locales.

3. *Seguridad*

Es un factor crítico del algoritmo, la trayectoria resultante debe garantizar la evasión de todos los obstáculos conocidos en el área.

Para este caso particular, se desea que el algoritmo este en la capacidad de procesar fácilmente diferentes variables, dado que no se conoce la cantidad de puntos vía a trabajar y se manejan vectores de posición.

A partir de la revisión bibliográfica (ver anexo A) y teniendo en cuenta las características mencionadas anteriormente, se evaluaron los resultados que se presentan en diferentes artículos para la aplicación del algoritmo en problemas típicos de minimización de ruta como el TSP²⁷.

En [12] se aplican cinco metaheurísticas diferentes (*Algoritmos Genéticos, Simulated Annealing, Colonia de Hormigas, Búsqueda Tabú y GRASP*) para la solución de este problema, para finalmente concluir que la metaheurística "*Optimización por Colonia de Hormigas*" (*Ant Colony Optimization, ACO*) presenta buenos resultados, es fácil de implementar y converge con el menor número de iteraciones. Igualmente, en [13] el autor realiza una comparación entre los resultados de algoritmos evolutivos y ACO para la solución de problemas dinámicos, destacando de este último la versatilidad para adaptarse a problemas de distinta naturaleza.

Por otra parte, en el momento en que se realizó la revisión bibliográfica, se encontraron mayor cantidad de artículos donde se hace uso de esta metaheurística (ACO) para el problema de generación de trayectorias de manipuladores específicamente.

De la misma manera se hizo evidente en las múltiples aplicaciones de la metaheurística, la flexibilidad que presenta para ser modificado de acuerdo a la situación, que para este caso era necesaria teniendo en cuenta el método que se utilizó para la definición el espacio de trabajo (Ver capítulo 9.1.)

²⁷ El TSP (*Traveling Salesman Problem*) es un problema NP-Hard típico, que consiste en visitar un número determinado de ciudades al menor coste posible.

6.1. DESCRIPCIÓN DEL ALGORITMO

El término metaheurística “Optimización por Colonia de Hormigas” (*Ant Colony Optimization, ACO*), señala el modo genérico de operación de un amplio rango de algoritmos de formas variadas, derivadas de la relación entre algunos procesos del algoritmo general como la generación de hormigas, la planeación de las actividades, la actualización de la feromona y las acciones del demonio.

Los algoritmos de optimización por colonia de hormigas son ejemplos de algoritmos de enjambre, donde un gran número de agentes cooperan con el fin de alcanzar un objetivo sin necesidad de un control central que los lleve a ello.

El concepto principal de los algoritmos de *ACO*, es la forma en la que las hormigas encuentran el camino más corto entre el hormiguero y la fuente de comida [14]. Al caminar, las hormigas depositan en el suelo una sustancia llamada feromona, esta sustancia –que puede perdurar desde días hasta meses-, sirve de guía para las hormigas del resto de la colonia, las cuales siguen la ruta con la mayor concentración de la misma. En caso de no encontrar rastros de feromona en algún punto del camino, las hormigas se mueven aleatoriamente. El hecho de que la feromona se evapore con el tiempo, y las distancias más cortas sean recorridas mayor número de veces por las hormigas, garantiza que al final el recorrido de la colonia converja en la ruta más corta de todas las posibles.

Se han desarrollado diferentes métodos para la actualización de la feromona [15]:

- Paso a paso (o en línea): Cada hormiga deposita una cantidad constante de feromona en cada paso.
- A posteriori: Las hormigas depositan la feromona una vez han construido una solución. La cantidad puede depender de la calidad de la solución o parámetros externos definidos por el programador.

Los algoritmos *ACO* se consideran *algoritmos constructivos*, es decir, construyen la solución desde cero añadiendo componentes en cada iteración. Cada agente (hormiga) genera una solución individual y se mueve aplicando una regla de decisión estocástica basada en dos tipos de información:

- *Información heurística*. Mide la preferencia heurística de la hormiga para moverse de un nodo (i,j) a un nodo $(i+1,j+1)$. Esta información la proporciona el programador o experto en el problema que se está solucionando, dando una función general que cada agente calcula para cada nodo al cual puede pasar, y no se modifica durante la ejecución del algoritmo. Generalmente depende de la distancia entre los nodos (visibilidad), sin embargo, puede ser modificada de acuerdo a las características del problema. [16]
- *Información de los rastros de feromona artificial*. Corresponde a la cantidad de feromona en un nodo, esta información se modifica durante la ejecución del algoritmo; cada hormiga deposita cierta cantidad de feromona dependiendo de su solución y esta se evapora transcurrido cierto número de iteraciones.

En la literatura se han propuesto diversos algoritmos que siguen la metaheurística *ACO*. El algoritmo ‘Sistema de Hormigas’, *AS* por sus siglas en inglés (*Ant System*), presenta principalmente un interés histórico ya que fue el primer algoritmo desarrollado con esta metaheurística²⁸. A pesar de que se desarrollaron diferentes versiones del mismo²⁹, lo que hoy en día se conoce como tal, es aquel que lleva a cabo la actualización de la feromona una vez que la solución este completa (a posteriori) [14].

El proceso es el siguiente: inicialmente, los rastros de feromona se reducen en un factor constante (evaporación), a continuación cada hormiga deposita una cantidad de feromona que es función de la calidad de su solución. Puede hacerse uso de una solución inicial generada con otro algoritmo con el fin de agilizar el proceso del seguimiento de feromonas en las primeras hormigas de la colonia, así como también pueden implementarse algoritmos de búsqueda local que refinan las soluciones generadas.

Una variante de este algoritmo, es el **Sistema de Colonia de Hormigas (*Ant Colony System, ACS*)** el cual incluye tres modificaciones importantes [14]:

- Un procedimiento denominado “regla proporcional pseudoaleatoria”, que garantiza que las hormigas tengan la posibilidad de explorar nuevas soluciones.
- La existencia de un procedimiento externo global (se conoce como: “acciones del demonio”) que actualiza la feromona fuera de línea de los rastros, para lo cual se toma únicamente la hormiga que genera la mejor solución. Además de esto, “el demonio” puede aplicar un algoritmo de búsqueda local para mejorar las soluciones.
- Las hormigas aplican una actualización en línea paso a paso de los rastros de feromona que incluye tanto la deposición, como la evaporación de la misma, siendo esta última mayor, lo que disminuye la concentración de feromona en los nodos recorridos, favoreciendo la generación de soluciones diferentes a las ya encontradas.

6.2. APLICACIONES DEL ALGORITMO

Los algoritmos de *ACO* han sido aplicados fundamentalmente en dos tipos de problemas diferentes; los problemas dinámicos de caminos mínimos, donde el algoritmo se adapta a los cambios del problema como el enrutamiento de redes en comunicaciones; y los problemas de optimización combinatorio NP-Hard, para los cuales las técnicas clásicas ofrecen a menudo un comportamiento muy pobre. [14]

El primer problema para el cual se adecuó un algoritmo de *ACO*, fue el problema del agente viajero (*Traveling Salesman Problem, TSP*)³⁰, para el cual es evidente la adaptación del comportamiento real de las hormigas dado que se trata de un problema de camino mínimo. Los resultados obtenidos fueron satisfactorios y convirtieron a este problema en el patrón de comparación para los siguientes modelos propuestos. Los siguientes problemas para los que se implementaron algoritmos *ACO*, fueron el problema

²⁸ Se presentó inicialmente en la tesis de Dorigo y Gambardella en el año 1992.

²⁹ AS- Densidad, AS- Cantidad, AS- Ciclo.

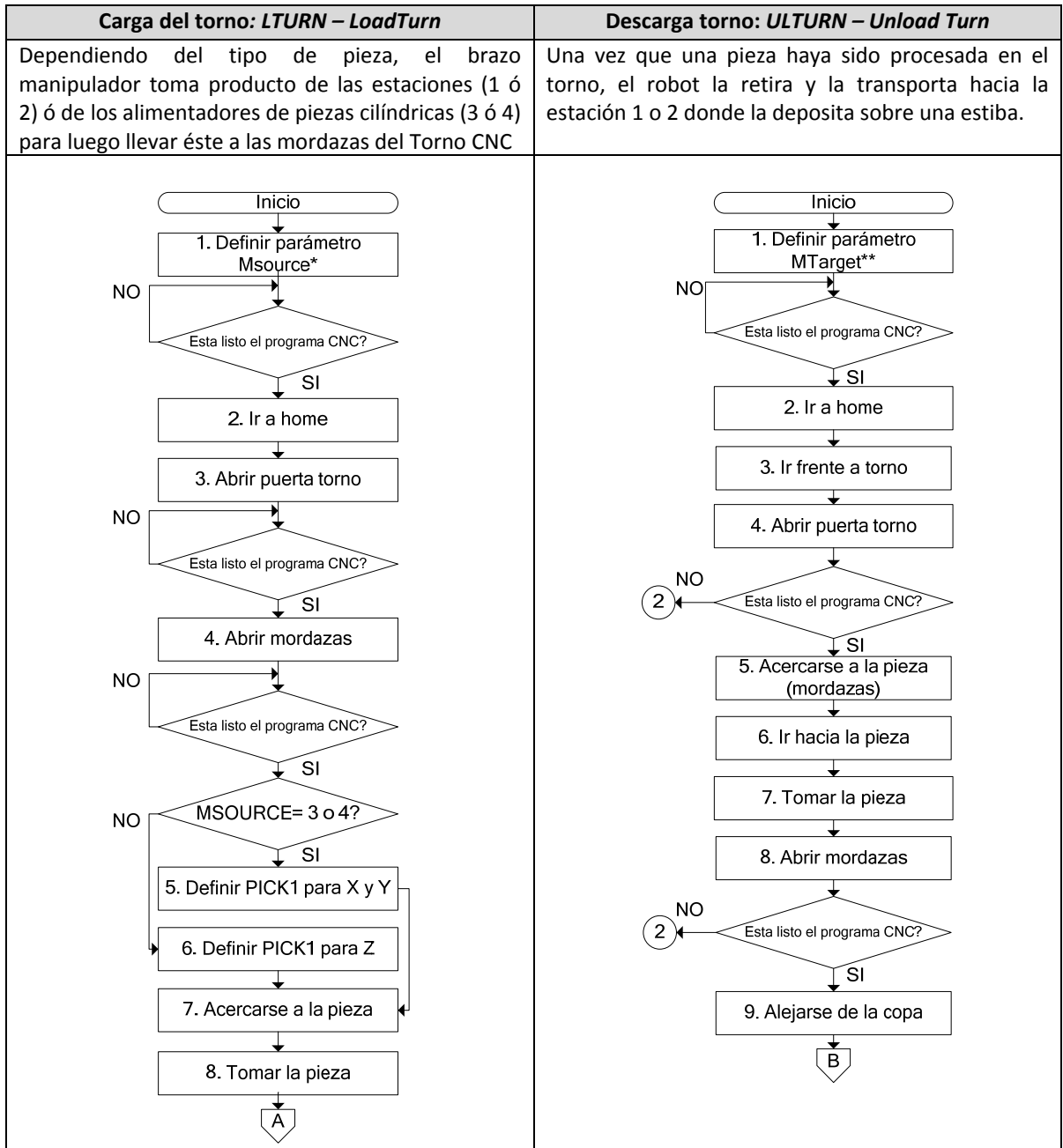
de asignación cuadrática (*Quadratic Assignment Problem, QAP*), y el problema de secuenciación de trabajos (*Jobshop Scheduling Problem, JSP*), probando las ventajas de la metaheurística en campos como locación y programación de tareas [12].

A partir de esto, muchos autores distintos han implementado *ACO* para la solución de problemas de optimización combinatoria, como los problemas de enrutamiento de vehículos, la secuencia común más corta, el problema de la mochila, entre otros. *ACO* ha demostrado su capacidad para obtener los mejores resultados para varios de los problemas a los que ha sido aplicada³¹, contando además con un buen desempeño computacional, lo cual hace de esta metaheurística una buena elección para el problema de generación de trayectorias.

³¹ Para mayor información sobre el algoritmo ver anexo C, y [14] [21] [16].

7. ANÁLISIS DE LA TRAYECTORIA DE CARGA Y DESCARGA DEL TORNO CNC

Los procesos de carga y descarga del torno son una aplicación del manipulador en el CIM³² de tipo alimentación de máquinas.



³² Manufactura Integrada por Computador (CIM, Computer Integrated Manufacturing)

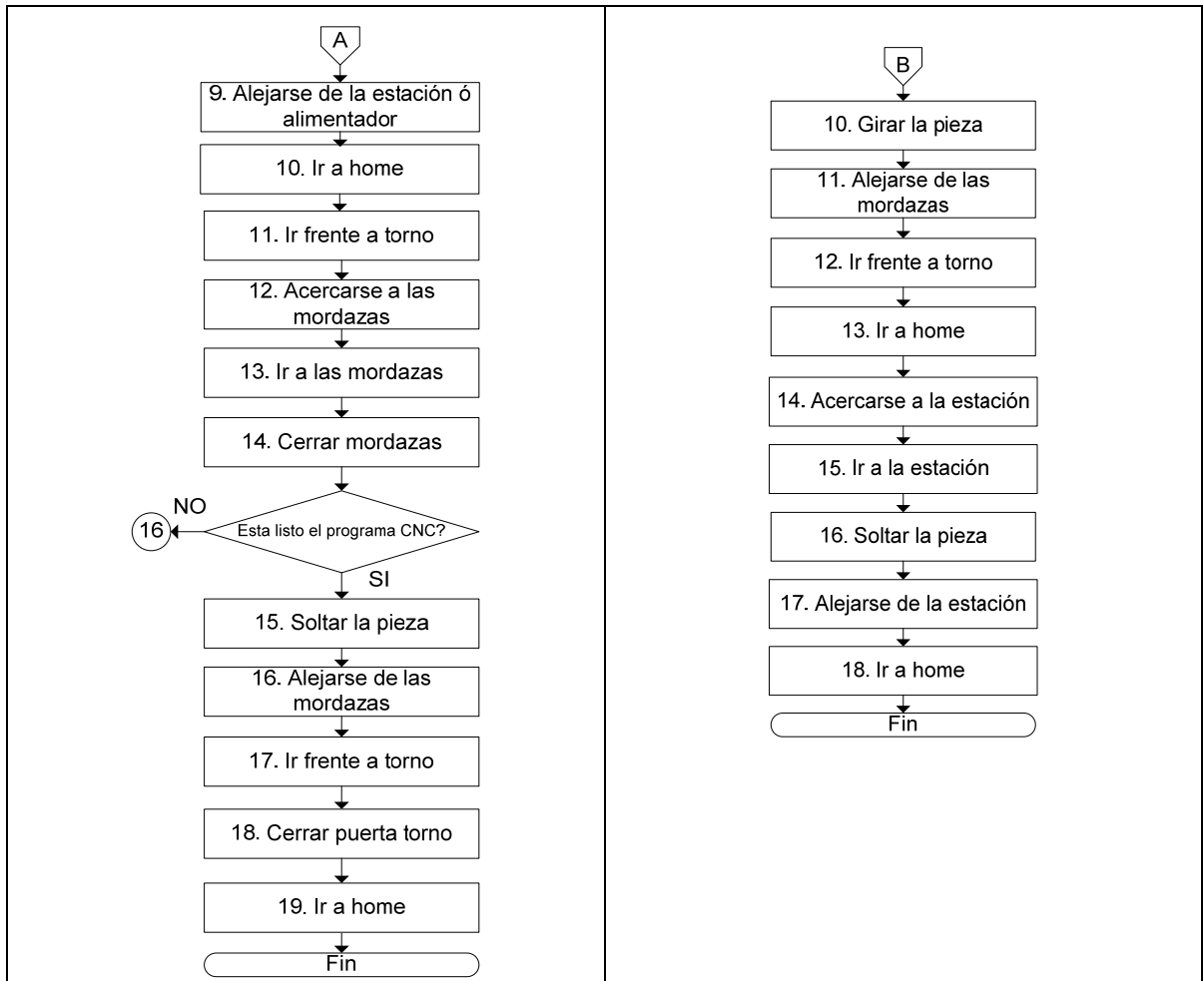


DIAGRAMA 2. DIAGRAMA DE FLUJO PROCESOS DE CARGA Y DESCARGA DEL TORNO. Fuente: Los Autores

***MSource:** Es la variable que el usuario ingresa para definir la estación o el alimentador del que se toma la pieza

****MTarget:** Es la estación a la que se debe llevar la pieza

Se observa que la trayectoria que sigue el robot es igual para los dos procesos, teniendo en cuenta que los puntos vía que se definen para cada caso son los mismos:

- PCYL(1 y 2): Son las posiciones de las estaciones en la cuales el robot recoge o descarga las piezas que vienen (o van) a la banda transportadora.
- PCYL(3 y 4): Son las posiciones de los alimentadores donde se encuentran las piezas, bien sea de cobre o aluminio.
- PGRUND: Es el *home* del robot.
- PWAIT: Es la posición frente al torno.
- PCNC: Es la posición dentro del torno, en las mordazas.
- PICK1 ó PLACE: Es el punto que se redefine en cada proceso con el fin de que cuando el robot vaya hacia alguna posición, primero llegue a una posición segura y luego se acerque con menor velocidad a su destino.

Teniendo en cuenta que lo que se pretende es generar una aplicación que permita realizar una correcta ubicación de los puntos vía para el movimiento del robot desde los alimentadores hasta el torno, se considera que la trayectoria actual para estos movimientos es eficiente en términos de número de puntos vía, pero ineficiente si se analiza que pueden existir trayectorias más cercanas a los puntos final e inicial que eviten los obstáculos sin necesidad de regresar a la posición inicial (PHOME).

En consecuencia, es necesario en este punto mencionar, que no se modificarán las posiciones relacionadas con el momento en que se lleva a cabo la carga del torno como tal, como es el caso de PCNC, sino que con el algoritmo se construirán nuevos puntos vía que delinee una trayectoria garantizando que esta sea libre de colisiones.

8. HERRAMIENTAS

8.1. HERRAMIENTAS DE SOFTWARE

A continuación se describen las herramientas relevantes utilizadas durante el desarrollo del trabajo:

COSIMIR® PROFESSIONAL

COSIMIR® (*Cell Oriented Simulation of Industrial Robots*) es un sistema de simulación en 3D que sirve para programar celdas de trabajo (*workcells*) basadas en robots. COSIMIR® es capaz de determinar el alcance de un robot en todas sus posiciones, desarrollar programas para robots y controladores, y optimizar la distribución de la celda de trabajo. En este pueden simularse todos los movimientos y operaciones del robot, con el fin de evitar colisiones y optimizar tiempos de ciclo de trabajo. [17]

COSIMIR® PROFESSIONAL combina tres herramientas en una: Simulación, Modelación y Programación:

El módulo de modelación y simulación permite diseñar celdas a partir de elementos disponibles en librerías, o a través del diseño de cajas cuyas características pueden ser definidas a conveniencia del usuario.

El módulo de programación del robot en COSIMIR® Professional permite que está sea realizada en el lenguaje propio del robot (Mitsubishi), u otros robots como Fanuc o KUKA, que sean compatibles con el lenguaje IRL (DIN 66312) [18]. Para el caso particular del robot MELFA RV-2A, el robot se programa con el lenguaje MELFA BASIC IV, del cual se hizo mención en el Capítulo 5.5.

LabVIEW (Versión 8.5)

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) es un entorno de programación gráfico para el diseño de sistemas de adquisición de datos, instrumentación y control. LabVIEW puede ser integrado a dispositivos de hardware y cuenta con gran diversidad de herramientas para análisis avanzado y visualización de datos [19].

LabVIEW se diferencia de los programas convencionales para el desarrollo de aplicaciones principalmente en un aspecto: la programación se realiza por medio de programación gráfica (lenguaje G), donde íconos y cables representan un diagrama de flujo.

Cada programa que se desarrolla se denomina VI (*Virtual Instrument*), debido a que su apariencia y operación imita instrumentos físicos tales como osciloscopios, y multímetros. Para construir el código de programación, inicialmente se construye la interfaz con el usuario (panel frontal), lo cual permite una mayor comprensión de lo que se busca en el momento de programar para el diseño del diagrama de bloques.

La programación en LabVIEW presenta varias ventajas [20], entre otras:

- Dado que se trata de un lenguaje gráfico, cuenta con numerosas herramientas de presentación en gráficas, botones, indicadores y controles, los cuales son muy esquemáticos y versátiles; generando

que la curva de aprendizaje sea más corta, lo cual reduce a su vez el tiempo de desarrollo de las aplicaciones.

- Presenta una interfaz amigable para la visualización de la ejecución del código.
- LabVIEW cuenta con múltiples herramientas y módulos para aplicaciones que requieran procesamiento de señales, imágenes, control, comunicación, entre otros, y es compatible con herramientas de desarrollo similares y puede trabajar a la vez con programas de otras áreas de aplicación, como Matlab o Excel.
- Permite crear soluciones complejas y se pueden usar otros bloques creados anteriormente (SubVI), para hacer aún más comprensible la estructura del código de programación. Igualmente da la posibilidad de incorporar aplicaciones escritas en otros lenguajes.
- El sistema está dotado de un compilador gráfico para lograr la máxima velocidad de ejecución posible.

Teniendo en cuenta las ventajas de LabVIEW, y el hecho de que el Departamento de Ingeniería Industrial de la Universidad cuenta con la licencia del programa, se decidió hacer uso de este, considerando que uno de los propósitos del proyecto es extender su aplicación en las diferentes trayectorias que se manejan en la celda de manufactura del CTAI, así como impulsar la investigación en este campo.

8.2. EQUIPO (HARDWARE)

Para poder establecer comparaciones en cuanto al tiempo de procesamiento y para evitar variaciones en los resultados por variables no previstas, se llevó a cabo la ejecución de los algoritmos en un único equipo. A continuación se muestran las especificaciones del mismo:

TABLA 2 – ESPECIFICACIONES EQUIPO DE COMPUTO

COMPONENTE	ESPECIFICACIÓN
Sistema Operativo (SO)	Microsoft Windows Vista
Versión	Home Premium Service Pack 1
Desarrollador SO	Microsoft Corporation
Tipo de sistema	32-Bit Operating System
Fabricante	TOSHIBA
Modelo	Satellite L305D
Procesador	AMD Turion X2 Dual-Core Mobile RM 70 2.00GHz
Memoria (RAM)	4.00 GB

9. IMPLEMENTACIÓN

El siguiente diagrama de flujo explica la secuencia que se siguió para la implementación del algoritmo definido para la generación de trayectorias del robot Melfa IV en la celda de trabajo del CTAI, desde la medición del espacio de trabajo, hasta el análisis de la solución obtenida en el escenario real.

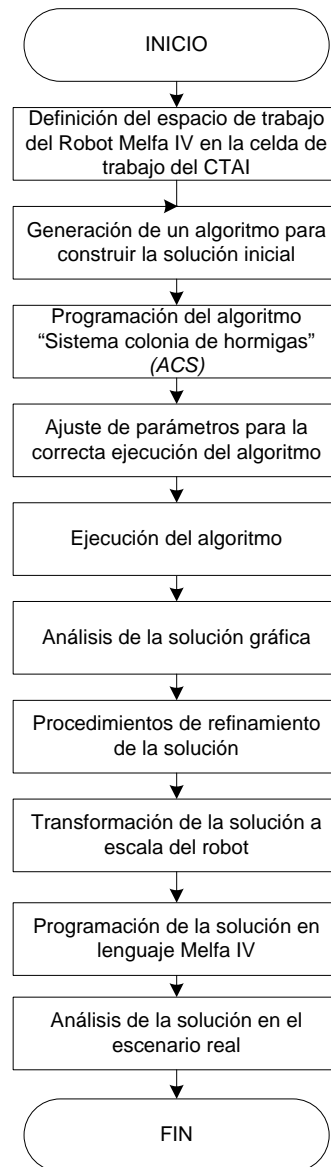


DIAGRAMA 3. Diagrama de flujo metodología para la implementación *Fuente:* Los Autores

9.1. DEFINICIÓN DEL ESPACIO DE TRABAJO DEL ROBOT MELFA RV-2A

9.1.1. MEDICIÓN DEL ESPACIO DE TRABAJO

Teniendo en cuenta las restricciones que se presentan para el desarrollo del proyecto, y la complejidad que supone la definición del espacio físico del robot, se decidió guiar al manipulador haciendo uso del 'Teach box'.

Con el fin de simplificar el espacio de trabajo, en un primer momento se determinaron las alturas críticas en las que existe posibilidad de colisión de alguna de las articulaciones del robot en la celda, y se decidió trabajar en un espacio "2 ½" (Figura 26), en donde al caracterizar cada uno de los planos XY de las alturas mencionadas y unirlos en un poliedro, se obtendría el espacio de configuración del robot manipulador. Para este fin se elaboró un protocolo en el que se consigna la metodología para la definición del espacio de trabajo (ver anexo D).

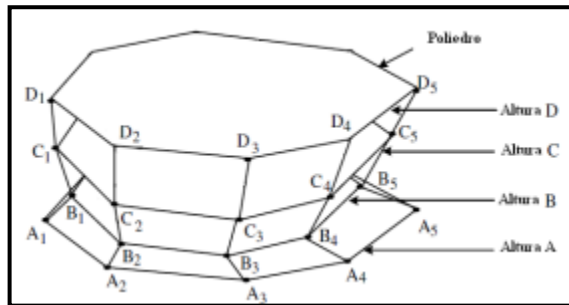


FIGURA 26. Ejemplo de un espacio "2 y medio", se toman puntos a diferentes alturas y se unen formando un poliedro.³³

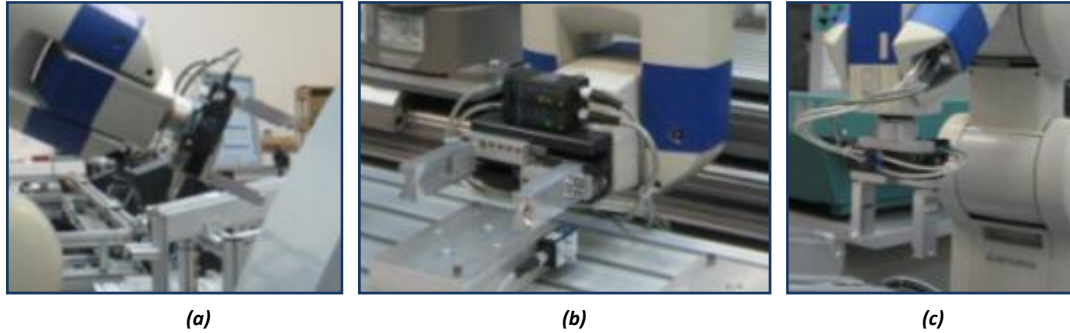
Sin embargo, durante el desarrollo del método se presentaron inconvenientes debido a algunos aspectos como la orientación de la herramienta, los puntos singulares³⁴, y la dificultad de plantear una solución para que el algoritmo evaluara los diferentes niveles de Z y generara una trayectoria entre ellos garantizando la evasión total de obstáculos (ver anexo E).

En consecuencia, y a partir de la revisión realizada sobre investigaciones anteriores, se concluyó que la representación del espacio de trabajo en 3D (aún trabajado en "2 ½") implica un alto grado de complejidad que no se encuentra dentro del alcance del presente trabajo de grado. Por esta razón, se decidió partir de una altura Z específica y definir diferentes orientaciones de la herramienta, generando así variados espacios de configuración en 2D, en los cuales se trabajó el algoritmo para la generación de trayectorias para finalmente hacer comparaciones entre ellas y la configuración actual. El resultado de estos estudios preliminares se encuentra consignado en el anexo F.

³³ Workspace generation and planning singularity-free path for parallel manipulators Anjan Kumar Dash a, I-Ming Chen a,*, Song Huat Yeo a, Guilin Yang b. Modificado por los autores

³⁴ Son aquellos puntos distribuidos en una línea recta que aunque conecten los puntos inicial y final, no se encuentran en el espacio de trabajo del manipulador.

Esta situación, sin embargo, no afectó el desarrollo de la metodología planteada en el anexo D, tan sólo modificó la definición de las alturas Z. A continuación se ilustran las diferentes orientaciones para las cuales se tomaron los puntos y se definieron los espacios de trabajo:



- a) **Orientación 1. Posición PWAIT**
- b) **Orientación 2. Posición PCYL**
- c) **Orientación 3. Posición Pinza Vertical**

FIGURA 27. ORIENTACIONES PARA LAS CUALES SE MIDIERON LOS ESPACIOS DE TRABAJO³⁵. FUENTE: LOS AUTORES

Es importante resaltar en este punto, que el uso de esta metodología para configurar el espacio de trabajo permite delinear el perímetro de los obstáculos ‘agrandados’³⁶, mientras se trata el robot como un punto en el espacio. Es decir, el hecho de guiar manualmente el robot hacia los puntos donde podrían generarse colisiones garantiza la definición del espacio de configuración (*c-space*).

9.1.2. REPRESENTACIÓN DE LOS ESPACIOS DE TRABAJO DEFINIDOS

Con el fin de representar gráficamente el espacio de trabajo permitiendo su manipulación, se hizo uso de un procesador de imágenes, delineando el espacio de trabajo encontrado en blanco y negro, a partir de la unión de los puntos hallados por medio del movimiento del robot en diferentes orientaciones, definiendo cada píxel de la imagen en el mapa de bits como un milímetro (1 píxel = 1 mm).

Para poder graficar efectivamente cada punto en el mapa de bits debe tenerse en cuenta que cada píxel se define como su posición dentro de la cuadrilla en el mapa de bits (*i, j*) siendo el punto (0,0) la esquina superior izquierda de la imagen, lo que significa que no existen valores negativos y por lo tanto es necesario realizar un proceso de translación de coordenadas. De igual manera, los valores deben ser aproximados a la unidad de forma que se haga posible su ubicación en la cuadrilla. Al realizar los anteriores procedimientos se está convirtiendo las coordenadas x,y a coordenadas i,j, a continuación se describen los pasos a seguir y se muestra un ejemplo:

³⁵ Las orientaciones tomadas corresponden a la orientación que se le da a la pinza en posiciones definidas para los programas *LTURN* y *ULTURN* (carga y descarga del robot)

³⁶ Recordar definición del espacio de configuración Capítulo 5.8

1. Multiplicar los valores de Y por (-1)
2. Hallar los valores mínimos de X y Y
3. Sumar el valor mínimo de X y Y a cada valor respectivamente
4. Aproximar valores a la unidad más cercana

Ejemplo: conversión de coordenadas x,y a i,j
 Se tienen los siguientes puntos

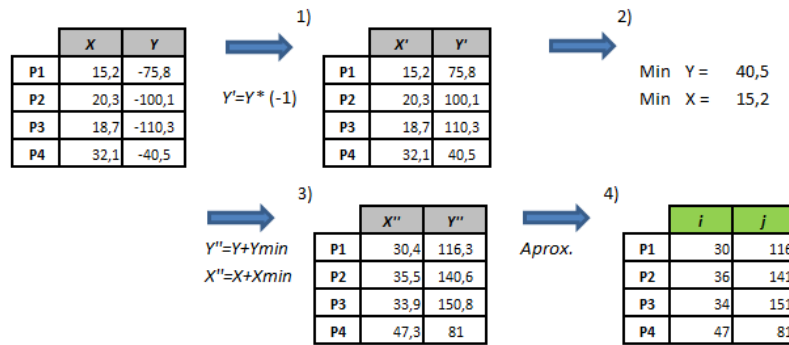


FIGURA 28. REPRESENTACIÓN DEL ESPACIO DE TRABAJO PARA LA POSICIÓN PCYL4³⁷

Una vez el espacio de trabajo ha sido graficado, es posible el procesamiento y manipulación de la imagen en LabVIEW. Se generó un primer VI³⁸ (programa) que la toma y transforma el espacio de trabajo en una matriz compuesta por los números 0 y 255, que representan los colores negro y blanco, para los espacios libres y los obstáculos respectivamente. En la Figura 29. se presenta la interfaz (el panel frontal) de este VI.³⁹

NOTA: La relación de VI's (jerarquía) para el proceso de definición del espacio, se encuentra consignado en el anexo G.

³⁷ Entiéndase como representación del espacio de trabajo la gráfica que muestra el contorno de los obstáculos definidos para una orientación y altura determinada.

³⁸ Virtual Instrument: El nombre que se le da a los programas de LabVIEW.

³⁹ Cabe notar que el tamaño de la matriz es tal (590 x 3030) que no es posible visualizar su tamaño en esta pantalla.

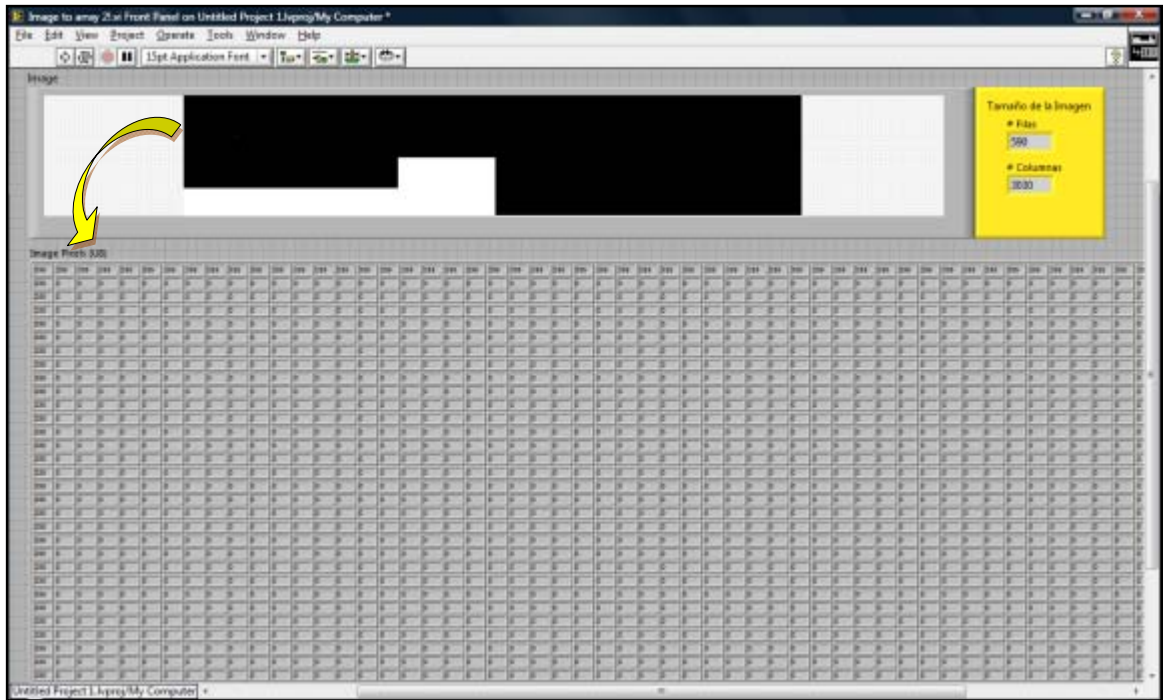


FIGURA 29. VI “TRANSFORMACIÓN IMAGEN DE TRABAJO A MATRIZ DE 0 Y 255”

Al obtener la matriz que identifica el espacio de trabajo a examinar, se determinó el método a utilizar para la definición del espacio de configuración (Capítulo 5.8). Se seleccionó la descomposición de la celda en forma aproximada, debido a las características del problema y la facilidad que esta ofrece en la implementación. Se dice que este método es aproximado debido a que no se generan celdas de diferentes tamaños ni se realizan distinciones entre celdas mixtas (no es jerárquico).

En este método, se divide el espacio total (la matriz) en pequeñas celdas, cuyo tamaño (resolución) es posible especificar. Al finalizar este proceso, se construye un algoritmo que identifique cada celda como libre (0) en el caso en que todos los elementos (píxeles) de la celda sean 0; u ocupada por obstáculo (255) cuando cualquiera de estos sea 255. (Figura 30)

La determinación de la ubicación de cada una de las celdas libres dentro de la matriz, se hace por medio de sus ‘centros de gravedad’, es decir, el píxel que se encuentra en el punto central de la misma, razón por la cual se subdividió cada imagen en resoluciones impares, donde dicho píxel existe (

Figura 31)

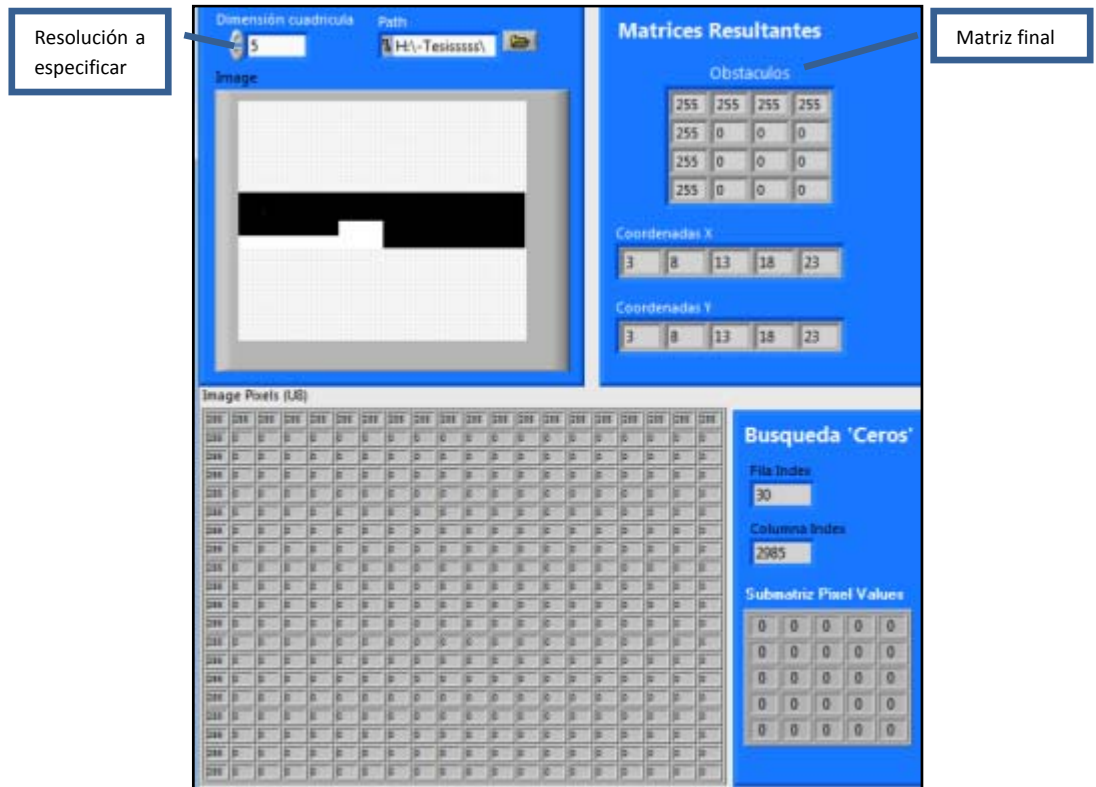
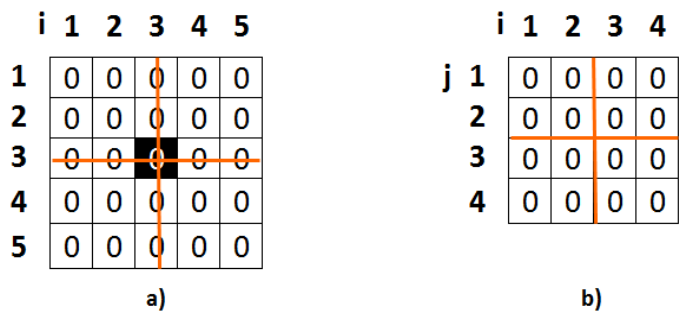


FIGURA 30. VI "DEFINICIÓN DEL ESPACIO DE TRABAJO"



- a) Celda impar, el elemento de la mitad (3,3) representa exactamente el centro de la matriz
- b) Celda par, no existe un elemento medio. Fuente: Los Autores

FIGURA 31. CENTRO DE LAS CELDAS

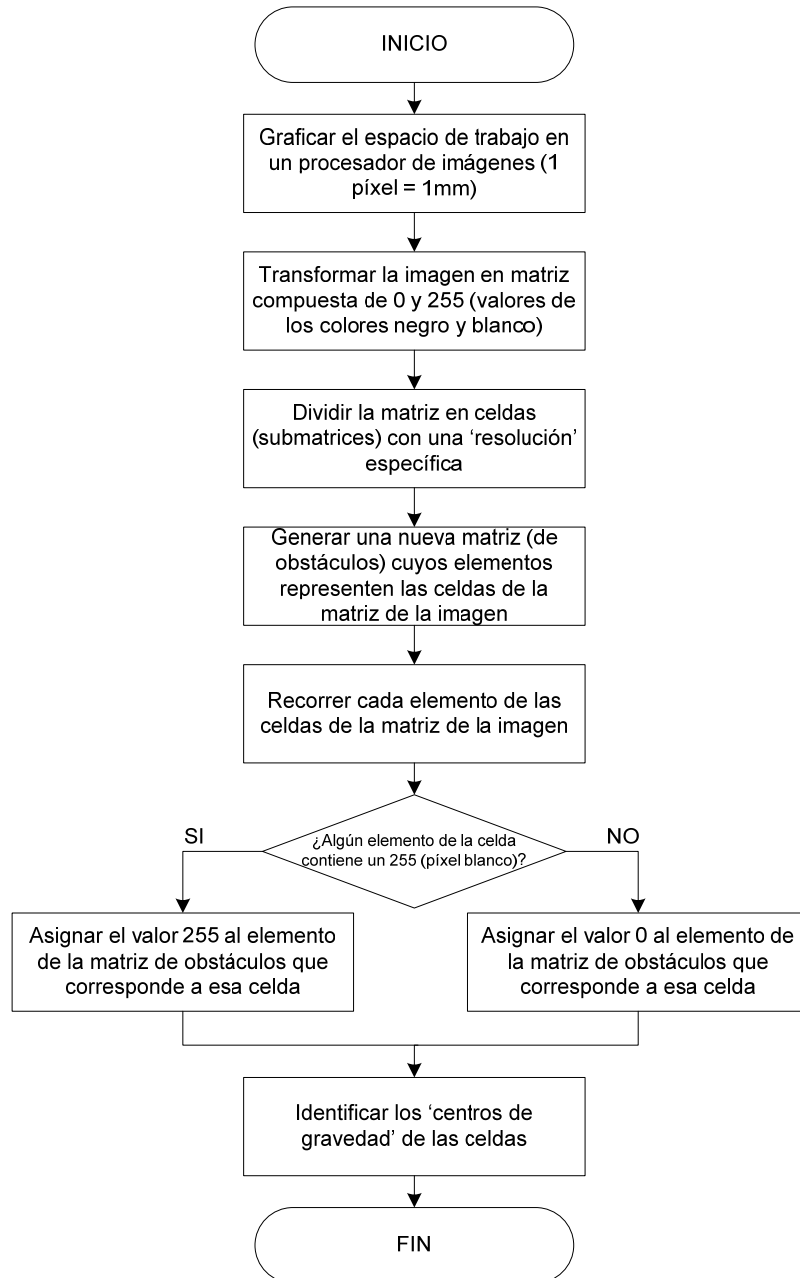
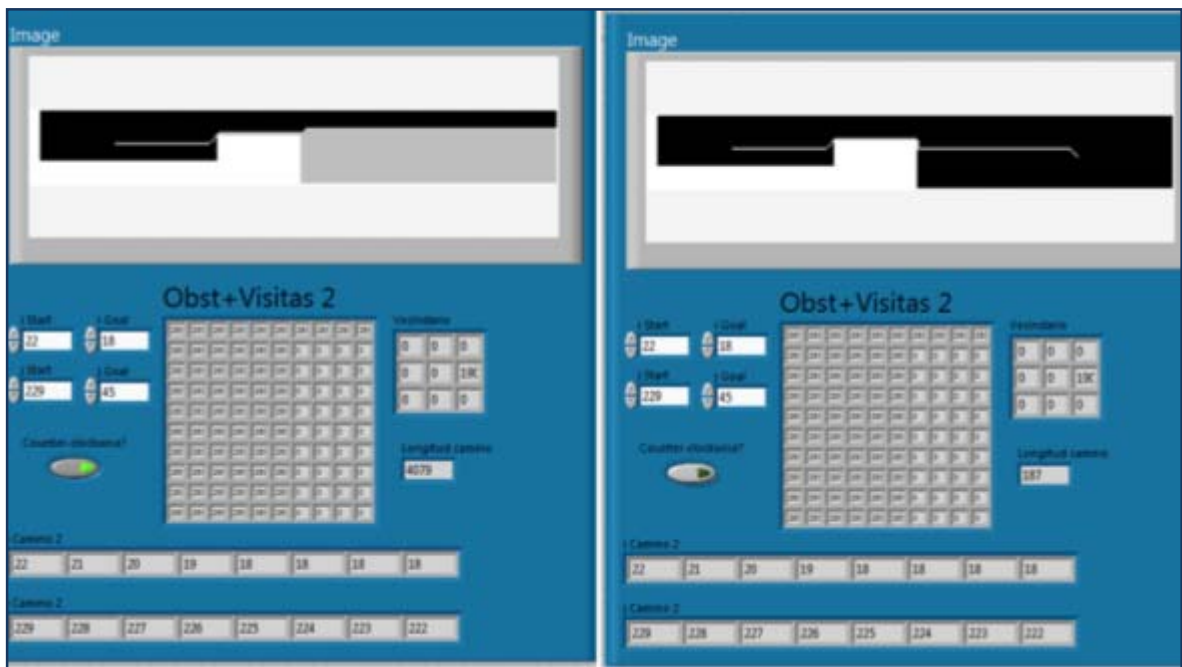


DIAGRAMA 4. Diagrama de flujo proceso de captura de la imagen y definición del espacio de trabajo. Fuente: Los Autores

9.2. BÚSQUEDA DE LA SOLUCIÓN INICIAL

La construcción de la solución inicial para la generación de trayectorias del robot se realiza a partir de la matriz de obstáculos, obtenida del procedimiento de definición del espacio (descomposición de celda). El proceso se lleva a cabo de la siguiente forma: Ubicado en un punto inicial definido, el algoritmo traza una línea entre este y un punto meta, identificando la dirección hacia la que debe ir, evalúa entonces el elemento correspondiente a esa dirección y se mueve a este si esta libre, en caso contrario, continúa evaluando las celdas adyacentes de acuerdo a la dirección determinada por el usuario (a favor o en contra de las manecillas del reloj). Como restricción se crea una lista de posiciones ya visitadas, de tal forma que no se generen ciclos.



(a)

(b)

(a) En contra de las manecillas del reloj

(b) A favor de las manecillas del reloj

FIGURA 32. SOLUCIÓN INICIAL DEL HALLADA PARA AMBAS DIRECCIONES DE GIRO

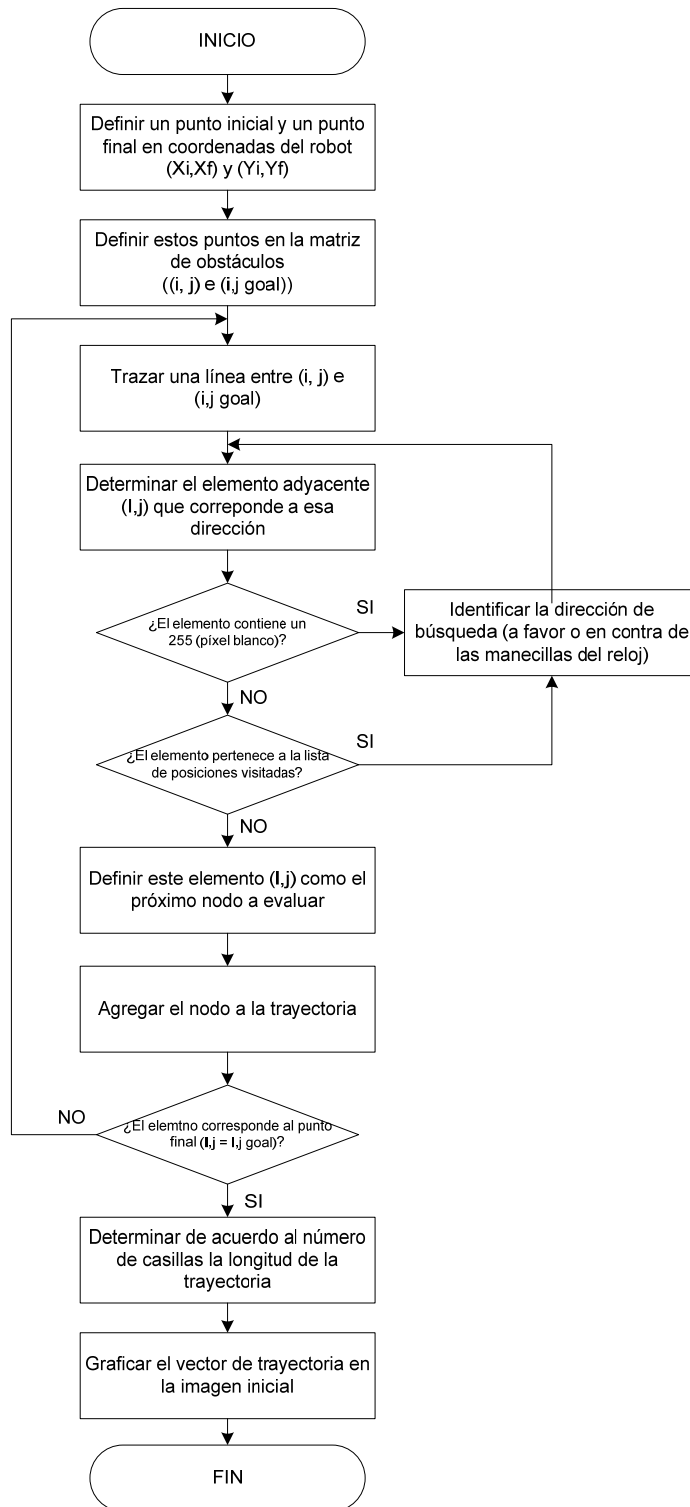


DIAGRAMA 5. Diagrama de flujo proceso de búsqueda de solución inicial. Fuente: Los Autores

9.3. FUNCIONAMIENTO DEL ALGORITMO PARA EL PROBLEMA DE GENERACIÓN DE TRAYECTORIAS DE TIEMPO MÍNIMO EN LA CELDA DE TRABAJO DEL CTAI

Teniendo en cuenta la naturaleza gráfica de la programación que se utilizó para el presente trabajo de grado, la construcción del algoritmo se realizó con base en diagramas de flujo de los procedimientos que para este se requieren. En el capítulo 9.5 se presenta la interfaz (*Front Panel*) de los procedimientos más importantes, con su respectivo diagrama de programación (*block diagram*)

En la primera etapa, se realizó la programación para la trayectoria de una hormiga, avanzando luego al enlace entre hormigas hasta culminar el recorrido de una colonia, para finalmente conectar los resultados entre varias colonias por medio de un procedimiento externo mejor conocido como “acciones del demonio”.

Proceso de generación de trayectorias en la colonia

Las hormigas, son en realidad agentes virtuales que toman la decisión de moverse dentro de una red de nodos definidos; para problemas como el TSP, las hormigas depositan la feromona en los arcos por los que transitan, y el peso de cada arco influye en la información heurística. Sin embargo, para resolver problemas de generación de trayectorias, la deposición se realiza directamente en los nodos dado que estas hormigas deben circular entre las regiones libres de una gráfica, evaluando en cada punto la posibilidad de moverse a los puntos adyacentes en las ocho⁴⁰ direcciones⁴¹, tal como se observa en la Figura 33.

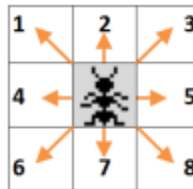


FIGURA 33. Movimientos posibles para una hormiga en la posición (i,j). Fuente: Los Autores

Para cada una de estas posiciones (que llamaremos nodos), se calcula un valor de probabilidad (P) a partir de la información heurística (η) y los rastros de feromona (τ), con el cual se decidirá el siguiente nodo de la trayectoria ($inext$, $jnext$). La forma en que se calcula esta información para la generación de trayectorias, representa otra diferencia con la aplicación tradicional del algoritmo, teniendo en cuenta que la distancia entre nodos adyacentes es un valor constante (igual a 1 para verticales y horizontales y $\sqrt{2}$ para diagonales), la heurística de preferencia de movimiento se calcula de acuerdo a la distancia euclidiana⁴² de cada nodo (i,j) al nodo final (igoal, jgoal).

⁴⁰ Para este trabajo, se evalúan las 8 direcciones adyacentes, lo que se conoce como conectividad 8, sin embargo, es posible trabajar únicamente con las 4 direcciones en X y Y, sin tener en cuenta las diagonales, lo que se denomina conectividad 4.

⁴¹ El conjunto de las ocho direcciones adyacentes es lo que se denomina en el presente trabajo como “vecindario”.

⁴² $D_{(i,j) \rightarrow (igoal, jgoal)} = \sqrt{(igoal - i)^2 + (jgoal - j)^2}$

Para cada nodo la **información heurística** se calcula como [14]:

$$\eta_{(i,j)} = \begin{cases} \frac{(\text{máx}D_{(i,j) \rightarrow (igoal, jgoal)} - D_{(i,j) \rightarrow (igoal, jgoal)})}{\sum (\text{máx}D_{(i,j) \rightarrow (igoal, jgoal)} - D_{(i,j) \rightarrow (igoal, jgoal)})}, & \text{Si } (i,j) \in \text{'vecindario'}, \\ 0 & \text{en otro caso} \end{cases}$$

Donde $\text{máx}D_{(i,j) \rightarrow (igoal, jgoal)}$ es la distancia más larga calculada dentro del conjunto de celdas adyacentes (vecindario). Se considera el valor cero (0) para $\eta_{(i,j)}$ en caso de que el nodo este definido como un obstáculo, o pertenezca a la "lista tabú de posiciones recientemente visitadas", la cual se definirá más adelante.

Recordando las características especiales del algoritmo descritas en el capítulo 6.1, los **rastros de feromona se actualizan** en un primer momento en línea paso a paso, lo que significa que para cada uno de los nodos visitados por la hormiga, esta evapora mayor cantidad de feromona de la que deposita, haciendo:

$$\tau_{(i,j)} = (1 - \rho)\tau_{(i,j)} + \rho\tau_0, \text{ donde}$$

$\rho \in (0,1]$ es un primer parámetro⁴³ de decremento de feromona, y

τ_0 es el valor inicial del rastro de feromona.

Una vez finalizados los procedimientos de cálculo η y los rastros de feromona τ , para cada una de las celdas adyacentes, se procede a calcular el valor P:

$$P_{(i,j)} = \eta^\beta \tau, \text{ donde}$$

β es el parámetro que pondera la importancia relativa de la información heurística; su importancia radica en que, para el caso de $\beta=0$, por ejemplo, sólo se tienen en cuenta los rastros de feromona para guiar el proceso de construcción, lo cual podría generar que las hormigas lleguen siempre a las mismas soluciones, dado que los rastros asociados a estas serán siempre los mismos (estancamiento en óptimos locales).

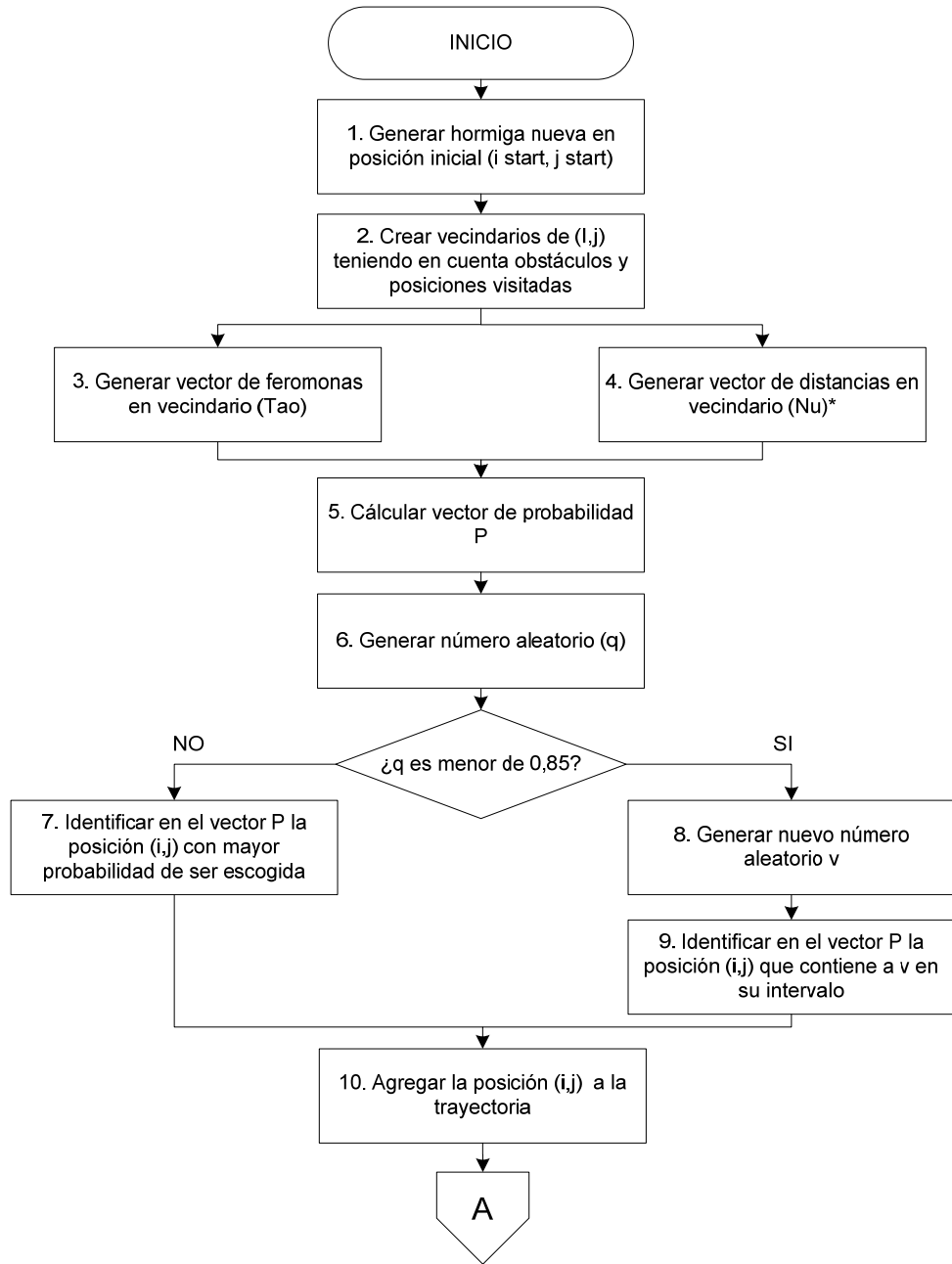
La "**regla proporcional pseudoaleatoria**", que garantiza que las hormigas tengan la posibilidad de explorar nuevas soluciones, consiste en establecer un valor "q" y generar un primer número aleatorio, que, de ser menor, escoge visitar el nodo del vecindario con el valor P más alto, y en caso contrario, genera un segundo número aleatorio "v", con el cual se selecciona probabilísticamente un nodo del conjunto de ocho.

⁴³ El ajuste de cada uno de los parámetros se explicara con más detalle en el capítulo 9.3.

Habiendo seleccionado el nodo siguiente, se procede a actualizar la “lista tabú de posiciones visitadas”, la cual contiene los últimos cinco nodos seleccionados por la hormiga, con el fin de evitar que se generen ciclos en el recorrido. Si la distancia encontrada por la hormiga k es la distancia más corta encontrada hasta el momento entre las hormigas de m colonia, este valor será almacenado junto con la lista de posiciones de los nodos visitados en la trayectoria.

En el momento en que la última hormiga de la colonia termine su recorrido, se grafica el vector de la mejor trayectoria en la imagen inicial, y se inicia el procedimiento de “acciones del demonio”.

A continuación se presenta el diagrama de flujo para el proceso señalado hasta el momento:



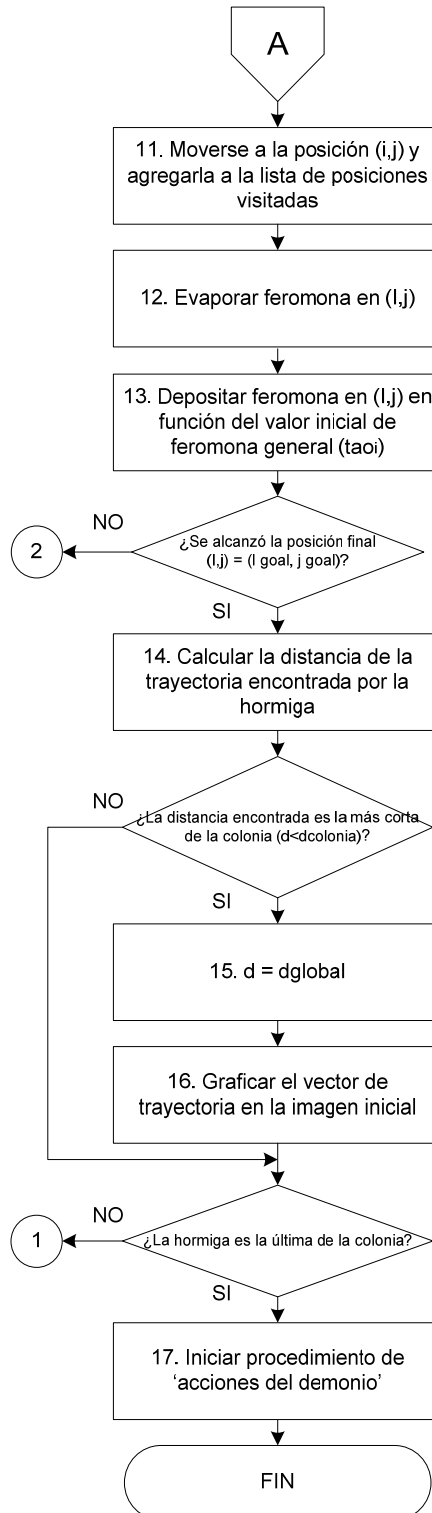


DIAGRAMA 6. Diagrama de flujo Algoritmo Sistema Colonia de Hormigas (ACS) para el problema de generación de trayectorias.

Fuente: Los Autores

Procedimiento “Acciones del demonio”

El procedimiento “acciones del demonio”, como se explicó anteriormente, es un enlace para transmitir información entre las colonias; este procedimiento se realiza bajo la idea de que quien lo realiza es un ente externo a las colonias de hormigas, como un ajuste del comportamiento real de las mismas para la obtención de mejores soluciones.

Cada vez que una colonia finaliza su recorrido y arroja una solución, el ‘demonio’ evalúa el resultado obtenido, comparándolo con la última ‘mejor distancia’ encontrada, las ‘acciones del demonio’ se llevan a cabo únicamente en el caso en que la distancia de la mejor trayectoria seguida en la colonia sea la mejor encontrada hasta el momento entre las m colonias.

Cuando la solución entre colonias mejora, el demonio toma la nueva ‘mejor solución global’, y realiza una actualización de feromona únicamente en los nodos que la componen, de la siguiente forma:

$$\tau_{(i,j)} = (1 - \varphi_1)\tau_{(i,j)} + [\varphi_1(1/d) + \varphi_2(1/T)], \text{ donde}$$

D es la distancia de la ‘mejor solución global’,

T es el número de cambios de dirección o ‘turnings’ que presenta la trayectoria,

$\varphi_1 \in (0,1]$ es un segundo parámetro de decremento de feromona y,

$\varphi_2 \in (0,1]$ es un parámetro adicional para la deposición de feromona de acuerdo con T .

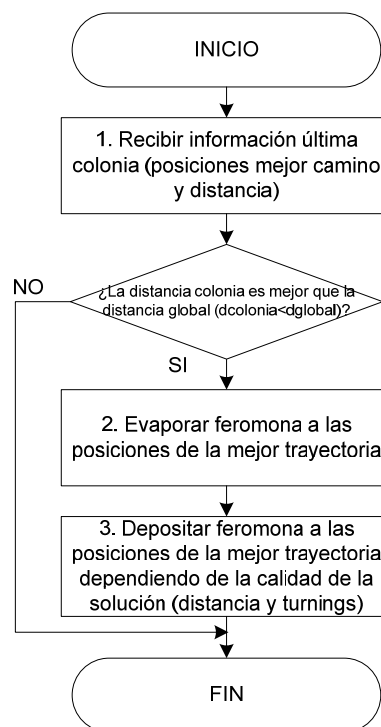


DIAGRAMA 7. Diagrama de flujo procedimiento “Acciones del demonio”. Fuente: Los Autores

Cabe resaltar la diferencia entre la actualización en línea paso a paso y las acciones del demonio en términos de cantidad de feromona depositada, dado que esta última se realiza en función de variables cuyo valor mejora con el tiempo y no una constante como en el primer caso.

9.4. DISEÑO EXPERIMENTAL

9.4.1. SELECCIÓN DE LOS VALORES DE LOS PARÁMETROS DEL ALGORITMO

La definición de los parámetros para cada uno de los procedimientos del algoritmo se realizó de acuerdo a dos criterios; en primer lugar se definieron algunos valores específicos por referencia bibliográfica, y por otra parte, se llevó a cabo una serie de pruebas experimentales que permitieran definir los valores más adecuados para los parámetros restantes.

Los parámetros definidos por referencia bibliográfica están basados principalmente en desarrollos y pruebas llevadas a cabo por Marco Dorigo y Luca Gambardella, quienes fueron los primeros en diseñar e implementar algoritmos basados en la heurística de optimización por colonia de hormigas [21]. Por su parte, Guan-Zheng Tan, *et al* [16] realizaron diferentes experimentos aplicando la metaheurística para el problema de generación de trayectorias, ajustando algunos parámetros por medio de la experimentación y haciendo uso de aquellos establecidos por Dorigo y Gambardella, reforzando los hallazgos de estos últimos. De la misma manera, autores como Alonso, *et al.* [14] realizaron extensas pruebas estadísticas para la solución del problema del agente viajero haciendo uso de diferentes algoritmos basados en ACO, utilizando diferentes valores de los ya establecidos para su parametrización. Los parámetros definidos con base en estos antecedentes fueron n , q , ρ , φ_1 , φ_2 y τ_0 . Por otra parte, los parámetros m , β y r se definieron de acuerdo con pruebas experimentales.

La descripción y valor de cada parámetro se consigna a continuación en la

Tabla 3, y la descripción de las pruebas experimentales se encuentra en el Anexo I.

TABLA 3. AJUSTE DE PARÁMETROS PARA LA EJECUCIÓN DEL ALGORITMO

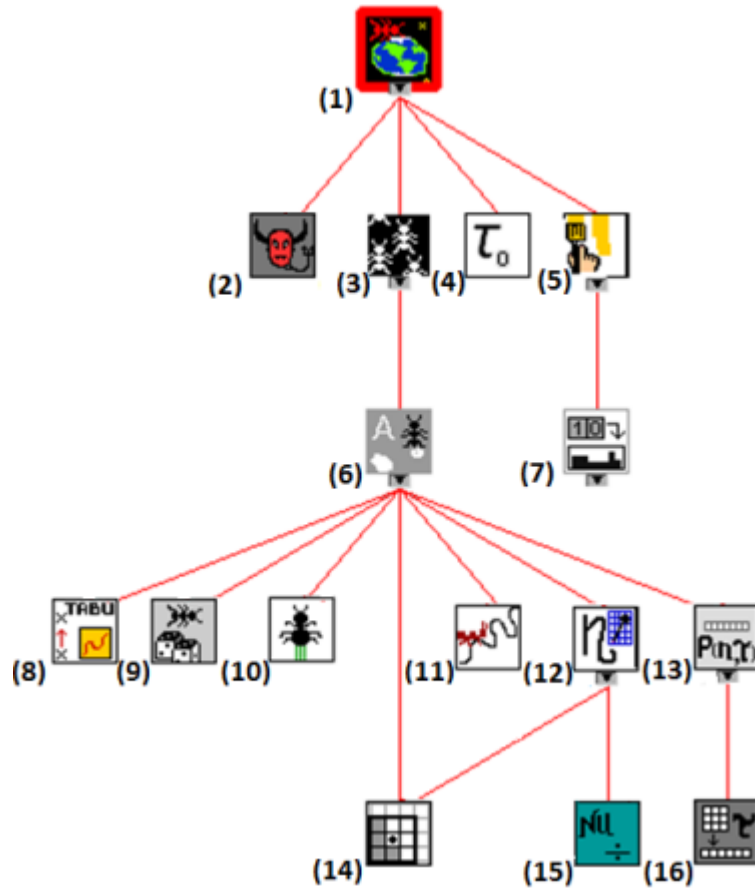
PARÁMETRO	DESCRIPCIÓN	VALOR	COMENTARIO
n	Número de hormigas de cada colonia	$\frac{8.7}{q} \approx 10$	Este valor corresponde a la formulación ⁴⁴ [21] $m = \frac{\log(\varphi_1 - 1) - \log(\varphi_2 - 1)}{\tau_0 \cdot \log(1 - \rho)}$
q	Valor límite para la ‘regla proporcional pseudoaleatoria’	0.85	A diferencia de Dorigo [21], este valor resulto mejor en la aplicación específica en [16]
ρ	Parámetro de actualización local de feromona	0.1	Recomendado por experimentación en [21]
φ_1	Parámetro de actualización global de feromona	0.1	Recomendado por experimentación en [21]
φ_2	Parámetro de actualización global de feromona para los turnings	0.01	Se considero este parámetro teniendo en cuenta la naturaleza del problema, de acuerdo con [11]
τ_0	Valor inicial de feromona	$\tau_0 = (m \times \bar{L}_{sub})^{-1}$	De acuerdo con [16], , donde m es el número de hormigas y L_{sub} el valor de la solución inicial generada.
m	Número de colonias para cada iteración	80	Experimental
β	Parámetro que pondera la importancia relativa de la información heurística	2	Experimental
r	Resolución de la imagen a utilizar	11	Experimental

Fuente: Los Autores

⁴⁴ Dorigo *et al.* Definen esta función para m teniendo en cuenta la relación de la cantidad de feromona en un nodo antes y después del proceso de actualización global, la cual determinan como: $(\varphi_1 - 1)/(\varphi_2 - 1) \approx 0.4$.

9.5. IMÁGENES DE LA INTERFAZ Y DIAGRAMAS DE FLUJO PARA LA PROGRAMACIÓN DEL ALGORITMO

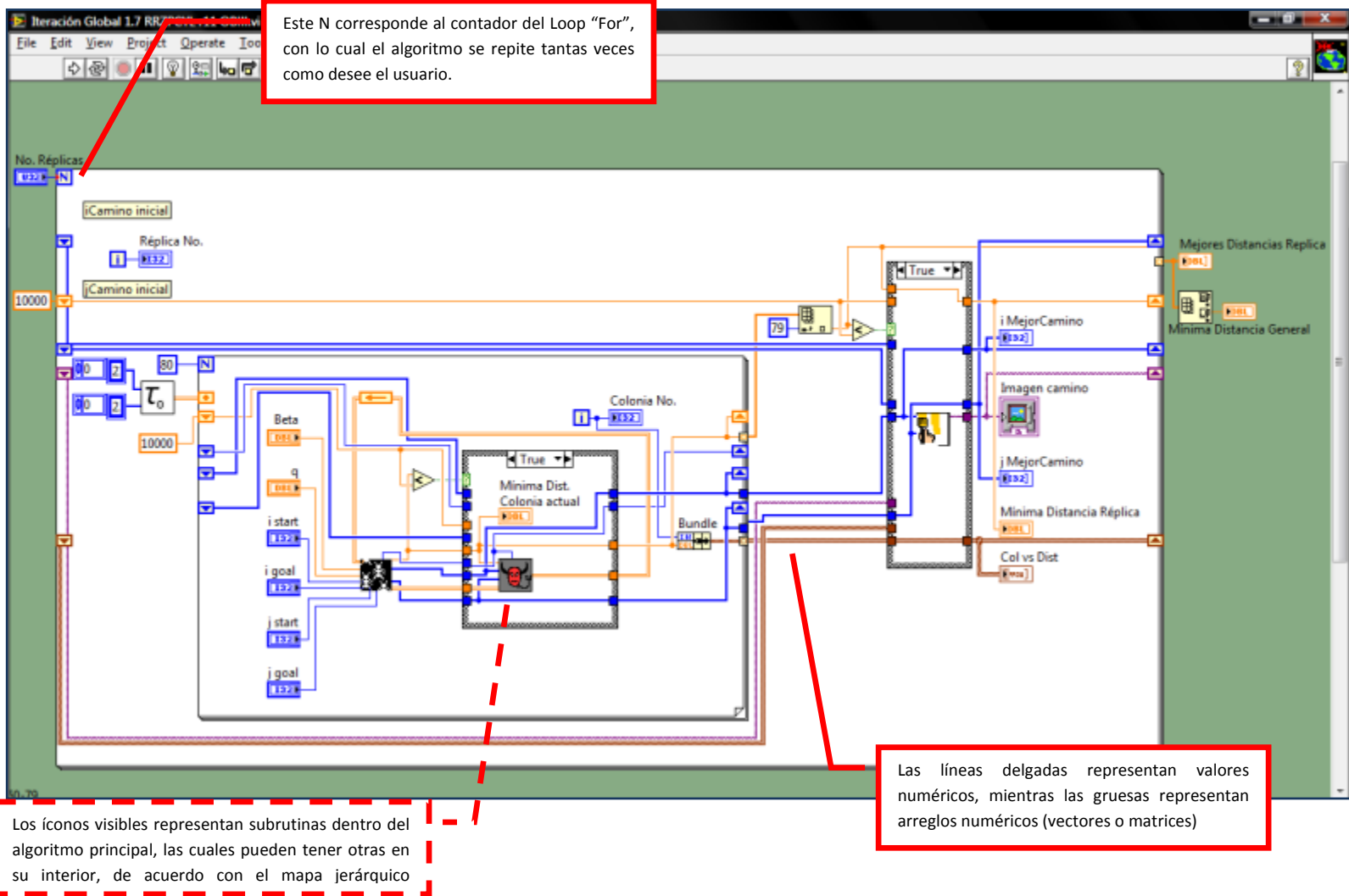
El siguiente diagrama ilustra la forma en que se construyó el algoritmo ACS en LabVIEW⁴⁵, donde el proceso número (1) corresponde al programa (VI) principal, cuya interfaz (*Front Panel*) contiene los parámetros principales a evaluar, y por lo tanto, es el único que se manipula para la ejecución de todo el algoritmo.



- | | |
|-----------------------------|--|
| (1) Iteración Global | (9) Regla proporcional pseudoaleatoria |
| (2) Acciones del demonio | (10) Actualización en línea |
| (3) Colonia de Hormigas | (11) Almacenar camino y <i>No. Turnings</i> |
| (4) Iniciación de feromonas | (12) Generar vector N_u |
| (5) Dibujar mejor camino | (13) Calcular vector P |
| (6) Recorrido Hormiga | (14) Creación del 'vecindario' |
| (7) Vector a Imagen | (15) Calculo Un |
| (8) Lista Visitas Tabú | (16) Convertir vecindario T_{ao} en vector |

⁴⁵ Recuerde la descripción del entorno de programación en el capítulo 8.1

- **Código de programación (Block Diagram)**

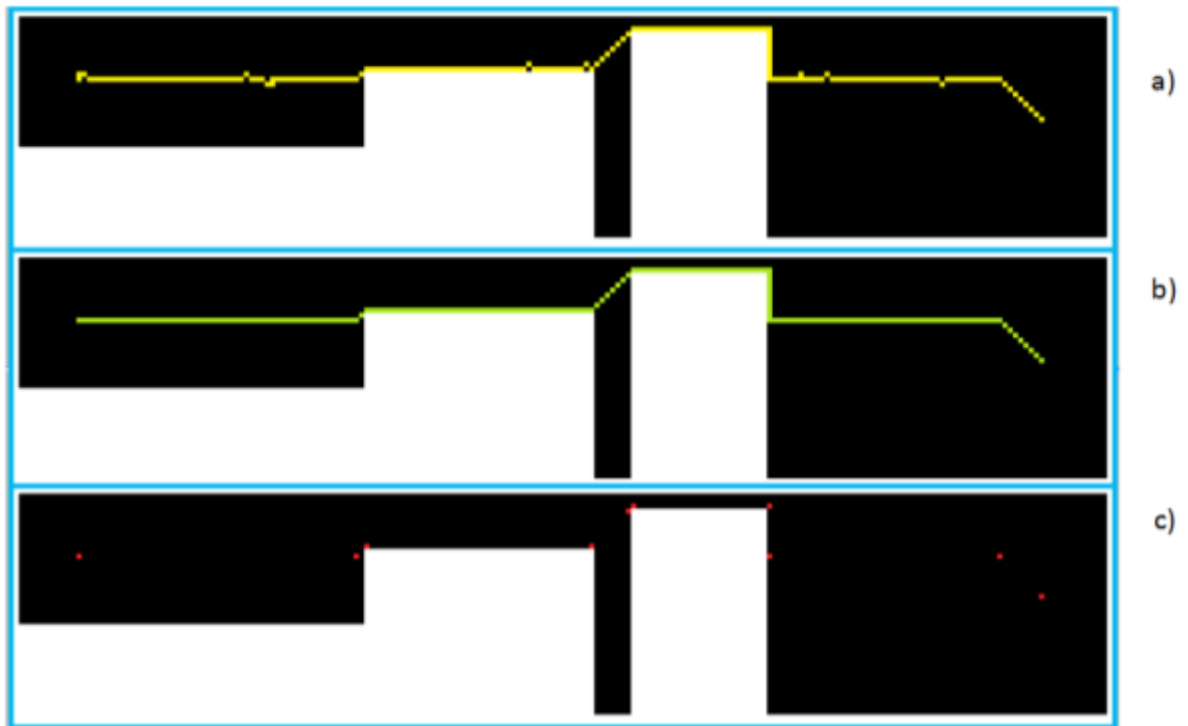


9.6. PROCEDIMIENTO DE IDENTIFICACIÓN DE PUNTOS VÍA

9.6.1. IDENTIFICACIÓN DE IRREGULARIDADES Y CAMBIOS DE DIRECCIÓN

Una vez las hormigas han encontrado el camino final, se procede a identificar los puntos en que la trayectoria cambia de dirección, para definir los mismos como los **Puntos Vía** que permitirán programar la trayectoria en el robot MELFA IV.

Durante el proceso de identificación, se realiza igualmente la búsqueda y corrección de irregularidades o 'saltos' en el camino (ver Figura 34), debido a que estos son considerados como 'ruido' aumentando la distancia total recorrida y el número de puntos vía necesarios para la definición de la trayectoria.



- a) Solución encontrada por ACS
- b) Solución después de los procedimientos de Corrección irregularidades
- c) Puntos vía de la solución

FIGURA 34. IDENTIFICACIÓN PUNTOS CAMBIOS DE DIRECCIÓN Y CORRECCIÓN IRREGULARIDADES DEL CAMINO

Para poder identificar las irregularidades, el programa analiza el camino paso a paso comparando en cada posición, la dirección en que se realizará el siguiente movimiento con la dirección que lo llevó desde la casilla anterior a la actual. Al hacer esto se comprueba si la posición actual se encuentra en línea recta con respecto a las anteriores.

Se sabe que cuando la dirección del movimiento siguiente coincide con la del movimiento anterior, se tiene un movimiento en línea recta, de manera que no es necesario realizar correcciones. Por otra parte, cuando dichas direcciones no coinciden, existen dos opciones: 1) el camino presenta una irregularidad ó 2) se está tomando una nueva dirección de movimiento. Para conocer cuál de estos casos se cumple, es necesario revisar la dirección del movimiento entre la casilla actual y la casilla posterior a la siguiente (Ej.: Dirección de movimiento de la posición $Pos_{(k-1)}$ hacia la posición $Pos_{(k+2)}$). (Ver Figura 35), si esta dirección coincide con la del movimiento anterior entonces se tiene una irregularidad en el camino, por lo cual no se considera realmente un cambio de dirección en este punto y se ignora en la revisión la siguiente casilla; de lo contrario (si las direcciones no coinciden) se debe almacenar la casilla actual como un punto vía, debido a que se está cambiando efectivamente la dirección del movimiento.

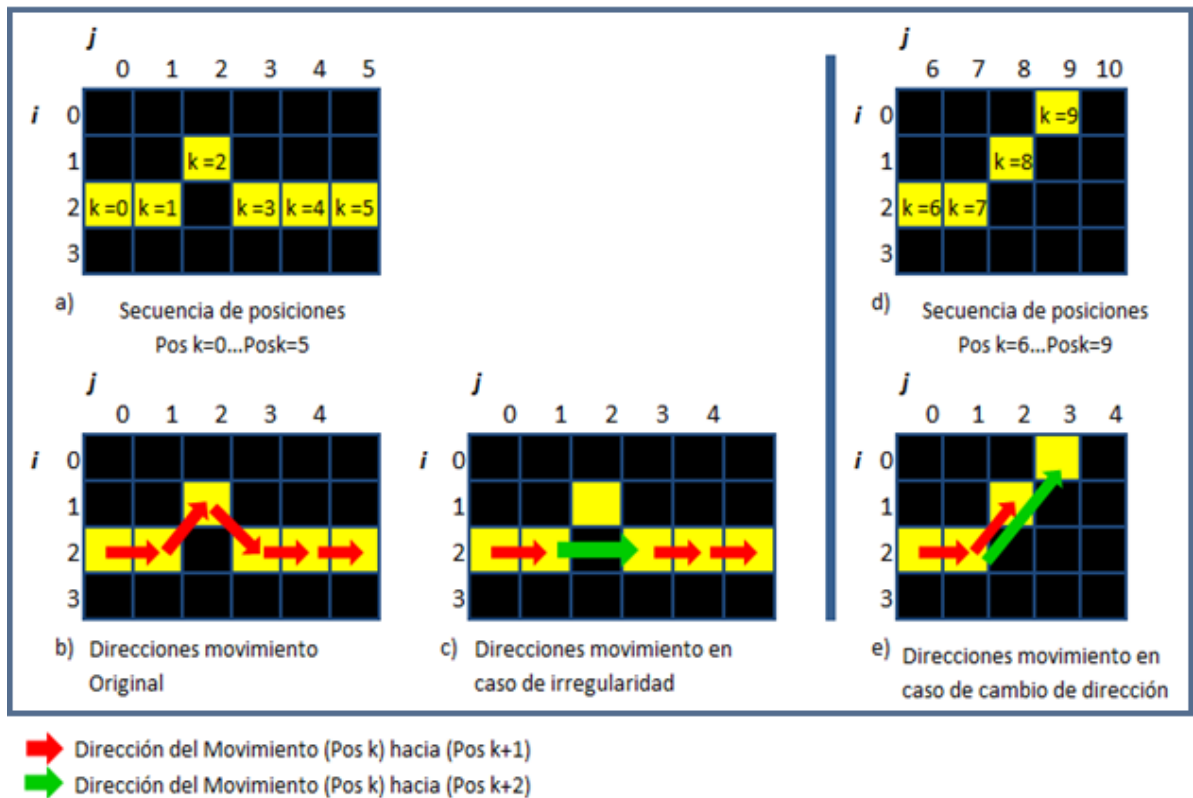


FIGURA 35. EJEMPLO IDENTIFICACIÓN DE IRREGULARIDADES Y CAMBIOS DE DIRECCIÓN. Fuente: Los Autores

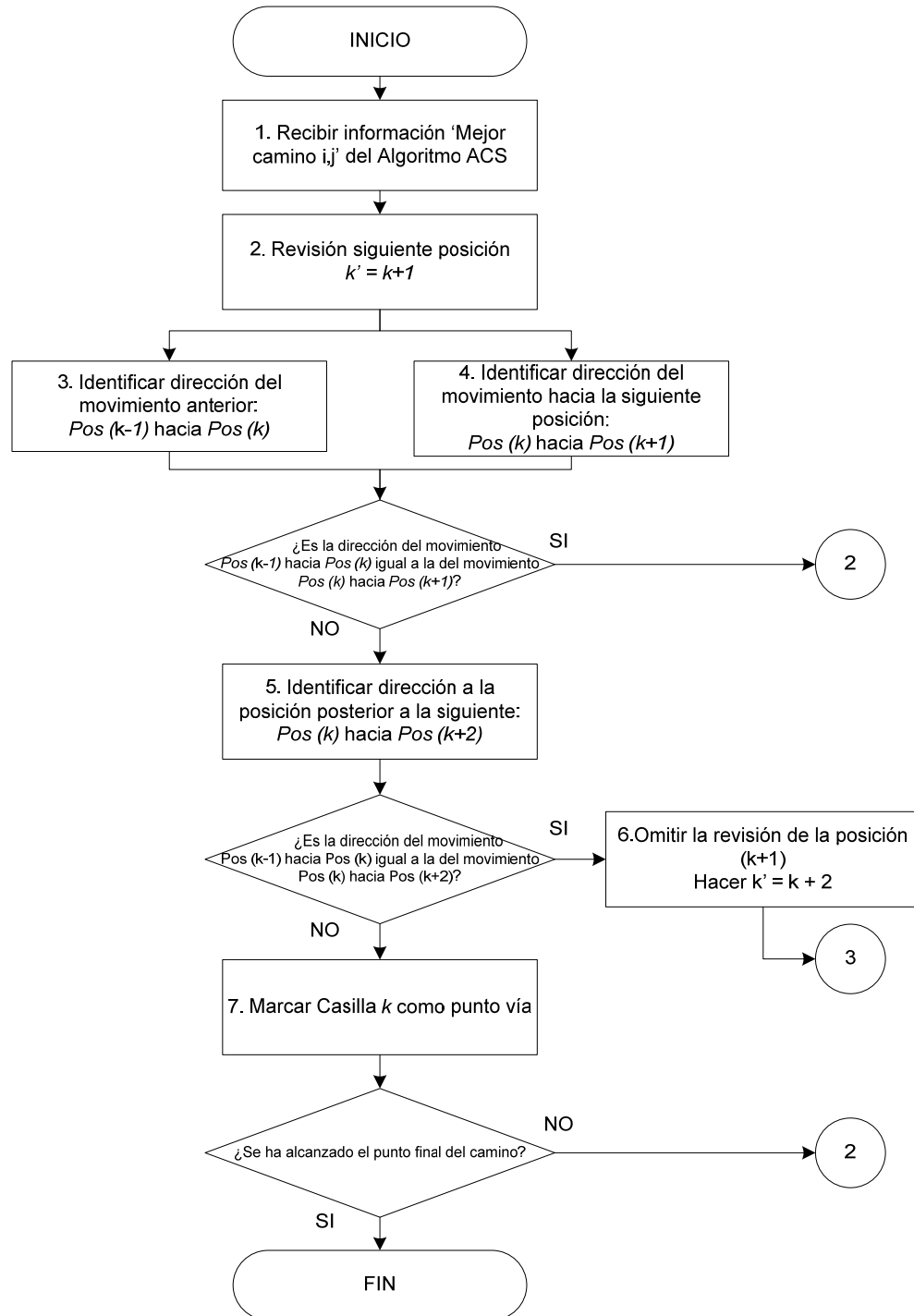
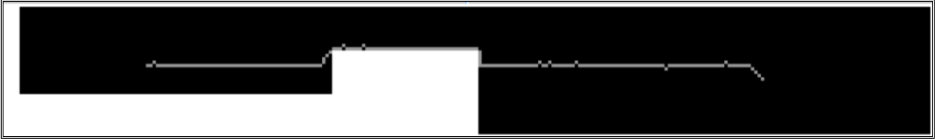
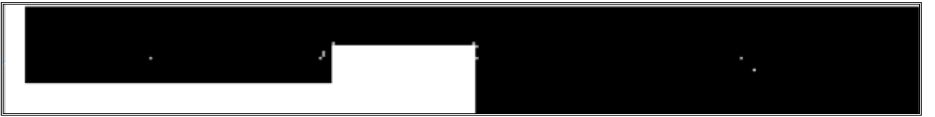
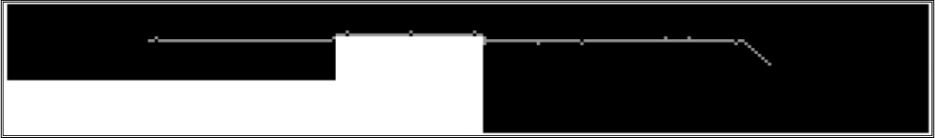





DIAGRAMA 8. PROCEDIMIENTO PARA ENCONTRAR PUNTOS VÍA Y CORRECCIÓN DE IRREGULARIDADES. Fuente: Los Autores

9.7. RESULTADOS DE LA EJECUCIÓN DEL ALGORITMO ACS

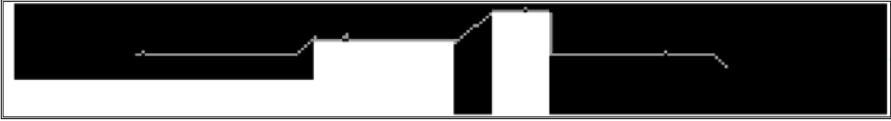

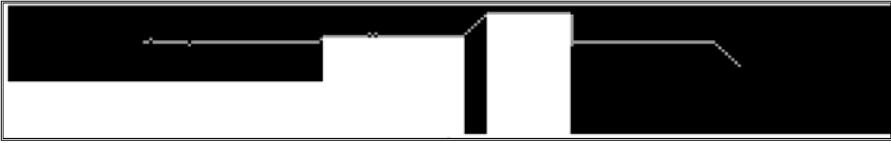



A continuación se presentan las imágenes de la trayectoria y los puntos vía obtenidos con el algoritmo para cada uno de los espacios definidos (Ver. Figura 27)

TABLA 4. RESULTADOS DE LA EJECUCIÓN DEL ALGORITMO PARA LOS ESPACIOS DEFINIDOS

PWAIT	Tray.	
	PV	
PPCYL	Tray.	
	PV	
PVERTICAL	Tray.	
	PV	

Con el fin de analizar los resultados del algoritmo en un espacio de trabajo diferente se creó un obstáculo en la celda de trabajo y se realizó la medición para cada una de las orientaciones.

TABLA 5. RESULTADOS DE LA EJECUCIÓN DEL ALGORITMO PARA LOS ESPACIOS CON OBSTÁCULO

OPWAIT	Tray.	
	PV	
OPPCYL	Tray.	
	PV	
OPVERTICAL	Tray.	
	PV	

A partir de las imágenes se puede analizar que el camino que siguen las hormigas efectivamente evita los obstáculos, sin embargo, cuenta con dos grandes restricciones impuestas por la definición del espacio. En primer lugar, el hecho de que solo se permita el desplazamiento de la hormiga al vecino más cercano (conectividad 8 (Figura 33)), imposibilita que se realicen los recorridos entre puntos vía con ángulos diferentes a 45 grados. En segundo lugar, la información heurística hace que la hormiga delinee los obstáculos cuando el punto final (i,j Goal) se encuentra justo detrás de estos.

Teniendo en cuenta lo anterior, se desarrolló un procedimiento para el refinamiento de las soluciones encontradas, el cual se expone en el siguiente capítulo. Más adelante, después del proceso de implementación, se ampliará el análisis de resultados incluyendo las soluciones en términos de tiempo y distancia. (Capítulo 11)

9.8. PROCEDIMIENTO DE REFINAMIENTO DE LA SOLUCIÓN

El procedimiento de refinamiento de la solución se realizó con el fin de mejorar las soluciones obtenidas, considerando las restricciones mencionadas anteriormente. El funcionamiento del algoritmo consiste en tomar cada uno de los puntos vía y calcular la distancias euclidianas hacia los puntos vía siguientes a este, en el momento en que se detecte un obstáculo entre un par de puntos del conjunto, se selecciona la distancia mayor entre las anteriormente calculadas y se almacena el par de puntos como **puntos vía**. Esto debido a que mientras mayor sea la distancia en línea recta, menor será el recorrido entre los dos puntos.

En la Figura 36 se observa que la distancia 3 (P1-P4) es la distancia más larga, sin embargo, es evidente que el camino original y las demás trayectorias entre estos dos puntos implican recorridos mayores y por lo tanto, una distancia total mayor.

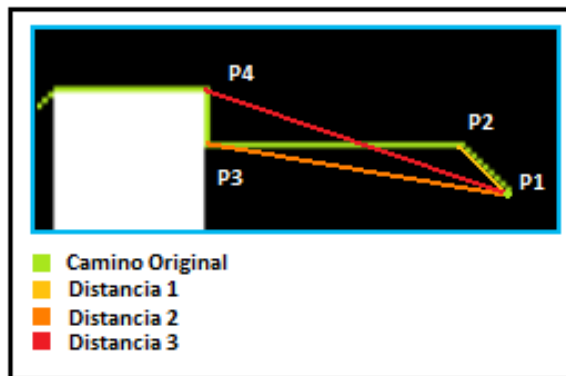


FIGURA 36. DISTANCIAS ENTRE PUNTOS VÍA. Fuente: Los Autores

A continuación se ilustra en un diagrama de flujo el procedimiento anteriormente descrito.

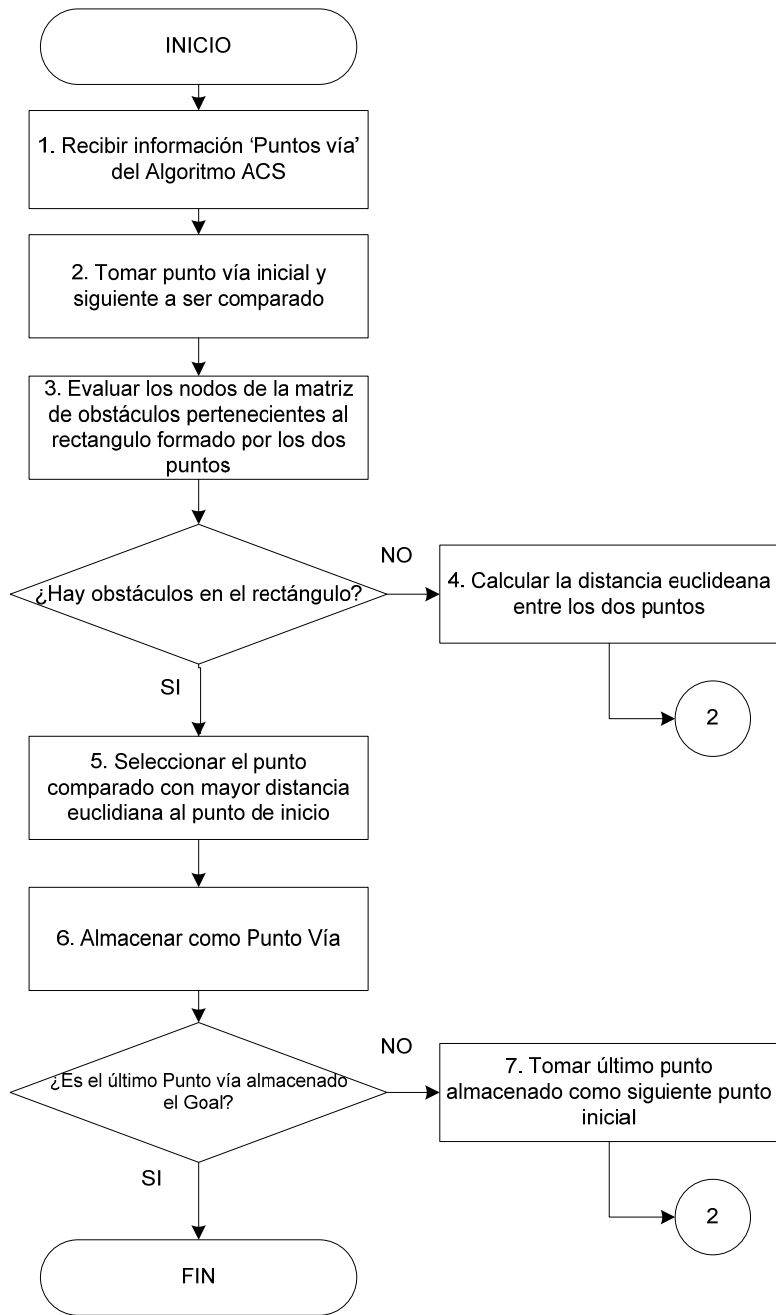

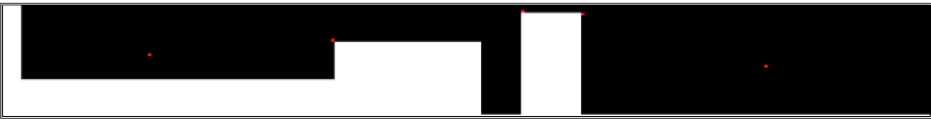
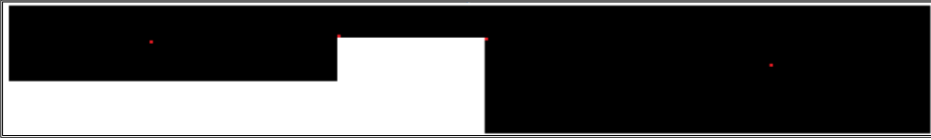


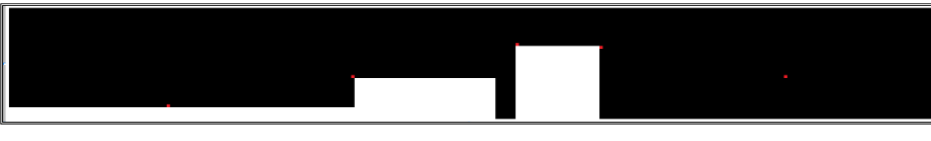


DIAGRAMA 9. PROCEDIMIENTO DE REFINAMIENTO DE LA SOLUCIÓN. Fuente: Los Autores

9.8.1. RESULTADOS DEL PROCEDIMIENTO DE REFINAMIENTO DE LAS SOLUCIONES

En las Tabla 6 se muestran las trayectorias finales luego del proceso de refinamiento de las soluciones.

TABLA 6. RESULTADOS DEL PROCESO DE REFINAMIENTO DE TRAYECTORIAS

PWAIT	Libre	
	Obstáculo	
OPPCYL	Libre	
	Obstáculo	
OPVERTICAL	Libre	
	Obstáculo	

9.9. TRANSFORMACIÓN DE LA SOLUCIÓN A ESCALA DEL ROBOT

9.9.1. PROCEDIMIENTO “CONVERSIÓN PUNTOS VÍA (I,J) A COORDENADAS REALES”

Después de la identificación de los puntos vía, es preciso convertirlos a unidades reales de distancia, de forma que estos puedan ser programados en el Robot Melfa IV. Para esto, se debe tener en cuenta la traslación específica del sistema coordinado de cada orientación del robot para la representación gráfica del espacio de configuración (Capítulo 9.1) y la resolución en la que se ha ejecutado el algoritmo de ‘definición del espacio’.

Dado que las coordenadas obtenidas se refieren a los valores de $[(x + L), y]$ y que los valores de $[z, A, B, C]$ ⁴⁶ son conocidos y constantes para cada espacio definido, es preciso especificar valores de x ó L para cada posición, de manera que se tenga valores para cada uno de los parámetros de posicionamiento del robot. Para la especificación de estos valores, es necesario identificar posiciones cómodas⁴⁷ que permitan al robot alcanzar los puntos y desplazarse entre estos en línea recta. Esto último se considera importante, teniendo en cuenta que al haber definido el espacio, o más específicamente los obstáculos, utilizando una orientación y altura constante, solo es posible afirmar que no existen colisiones en el camino si al programar el robot se asegura que estas variables mantengan dichos valores (orientación y altura) constantes a lo largo de toda la trayectoria. Por esta razón, las pruebas iniciales deben llevarse a cabo haciendo uso del comando de movimiento MVS (movimiento lineal).⁴⁸

De acuerdo con lo anterior, para cada configuración de altura y orientación se encontraron los puntos límite de movimiento y una extensión cómoda del manipulador, de forma que sea posible establecer valores de x ó L constantes (de acuerdo al caso). Debe tenerse en cuenta que para la posición en que recoge la pieza, el manipulador se encuentra extendido en dirección a $X+$, mientras que en la posición de carga del torno se encuentra extendido en dirección contraria ($X-$). Por esta razón, es preciso definir la forma en que el robot llevará la herramienta desde $X+$ hacia $X-$, teniendo en cuenta que esta operación está condicionada por la configuración particular del robot, es decir que existen valores de Y para los cuales el robot no está en capacidad de realizar dicho movimiento, al igual que existen valores de $(X + L)$ a los que este no llega debido al alcance total del manipulador, teniendo en cuenta el alcance del riel.

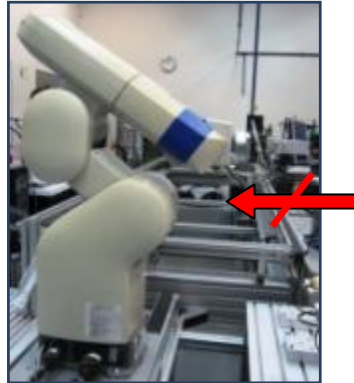
A continuación, se describen los puntos límite de movimiento anteriormente mencionados:

- **Límite valor Y:** Se tomo el valor de Y para el cual $x=0$, dado que el manipulador no puede transitar desde $(X+)$ hacia $(X-)$ si el valor de Y se encuentra por encima de este límite.
- **Límite valor X:** Se identificó el valor de X para el cual el robot es capaz de sobrepasar el límite de movimiento en Y previamente definido. Este valor debe medirse extendiendo el robot hacia $(X-)$ debido a que el movimiento se encuentra más restringido en esta dirección.

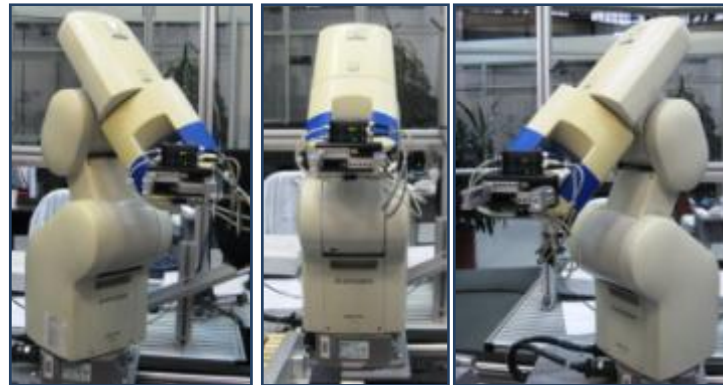
⁴⁶ Recordar que una posición es almacenada de acuerdo con la ubicación de la herramienta del manipulador y su orientación en el espacio, para lo cual se utilizan los parámetros $[X,Y,Z,A,B,C]$. Para más información ver capítulo 5.2.

⁴⁷ Al hablar de posiciones cómodas se hace referencia al valor de los ángulos de rotación de cada articulación. Ver capítulo 9.10.

⁴⁸ Recuerde los tipos de movimiento y comandos con los cuales es posible programar el robot descritos en el capítulo 5.5.



(a)



(b)

(c)

(d)

(a) y (c). Límite valor Y. No es posible conseguir una posición hacia Y+.

(b) y (d). Límites en X+ y X-. A partir de estos valores de X es posible conseguir posiciones más allá del Límite en Y.

FIGURA 37. VALORES LÍMITE DE X Y Y PARA CADA ORIENTACIÓN. FUENTE: Los Autores

Dado el caso de no encontrar configuraciones de X y L en que el robot sea capaz de transitar entre dos puntos en MVS, se utilizará entonces el comando MOV para dicho movimiento particular, haciendo la debida consideración para la calificación de la trayectoria.

9.10. PROGRAMACIÓN DE LA SOLUCIÓN EN LENGUAJE MELFA IV

Para el proceso de implementación y programación de las soluciones obtenidas con el algoritmo ACS en el robot Melfa RV- 2A, se realizó un protocolo (Anexo J), con el cual se estableció la metodología a seguir para este fin. Se recomienda al lector familiarizarse con el procedimiento allí expuesto.

10. PARÁMETROS DE COMPARACIÓN DE LOS RESULTADOS

A continuación se presentan las variables que caracterizan el desempeño de una trayectoria, a partir de las cuales se compararán los resultados obtenidos una vez se implementan las trayectorias en el robot:

TABLA 7- PARÁMETROS PARA LA COMPARACIÓN DE LAS SOLUCIONES GENERADAS

Parámetro	Descripción	Indicador
Distancia	La distancia total que recorre el manipulador es una variable independiente que actualmente genera tiempos ineficientes en la operación. Depende de la ubicación de los puntos vías en el espacio de trabajo.	Distancia total recorrida calculada por el algoritmo.
Tiempo	El tiempo total de recorrido del manipulador es la variable principal a optimizar, sin embargo, esta es una variable que no puede ser afectada directamente, dado que depende de la distancia del recorrido y la velocidad fijada para los movimientos.	Tiempo transcurrido desde el momento que el manipulador inicia el movimiento y el momento en que alcanza el punto final.
Suavidad de la trayectoria	La suavidad de la trayectoria hace referencia a la forma en la que el manipulador deberá moverse al ejecutar el movimiento. Se trata de encontrar una función continua de tiempo que pase a través de los puntos vía evitando los movimientos bruscos que desgastan el mecanismo e inducen error en el movimiento del manipulador. [22]	<i>Vibración en el movimiento del robot</i> Es una variable cualitativa, la cual se mide en porcentaje de acuerdo con la percepción del observador, donde 0% representa un movimiento sin vibración.
Torque	El torque es la tendencia de una fuerza a causar rotación sobre un eje fijo; para este caso, hace referencia a la fuerza que ejerce toda la cadena de eslabones para sostener la pieza en cada articulación. Se desea que los ángulos de rotación de cada articulación se encuentren dentro de un rango establecido bajo el cual se presentan menor cantidad de movimientos abruptos, generando un menor desgaste del mecanismo. [22]	Es una variable cualitativa, que se mide en porcentaje (% Comodidad) de acuerdo con la percepción del observador, dependiendo de los ángulos de rotación de cada articulación y el tiempo que el manipulador permanezca en posiciones fuera del rango establecido para estos. De igual forma, se califica la ejecución de movimientos abruptos.

11. RESULTADOS Y ANÁLISIS DE LA IMPLEMENTACIÓN DE LAS SOLUCIONES OBTENIDAS

Para cada una de las trayectorias evaluadas se realizaron videos a partir de los cuales se determinaron los tiempos de ejecución de las mismas. Así mismo, se registraron los valores de los parámetros definidos para la comparación.

- **Análisis de resultados para cada etapa del proceso de implementación**

Las **¡Error! La autoreferencia al marcador no es válida.** y Tabla 9 muestran las distancias y tiempos de las trayectorias obtenidos para los espacios definidos inicialmente. Los resultados se encuentran divididos en tres grupos, correspondientes a las modificaciones que se realizaron durante el proceso de implementación de las trayectorias.

TABLA 8. DISTANCIAS DE LAS TRAYECTORIAS OBTENIDAS CON EL ALGORITMO ACS

Nombre Trayectoria	Distancias (mm)		Mejora %
	Originales	Refinamiento	
PZWAIT	2166.35	2034.10	6.10%
PCYL	2186.47	2057.47	5.90%
PVERT	2211.91	2041.71	7.69%
LTURN*	2587.80	-	-

TABLA 9. TIEMPOS DE EJECUCIÓN DE LAS TRAYECTORIAS IMPLEMENTADAS EN EL ROBOT MELFA RV2A (EN SEGUNDOS)

Nombre Trayectoria	Originales			CNT		Mejora (MOV vs. MOV CNT)	Refinamiento	
	MVS	MOV	Mejora (MVS vs. MOV)	MVS	MOV		MOV	Mejora (MOV CNT vs. Refina.)
PZWAIT	16.07	9.40	71%	15.33	6.50	31%	5.43	20%
PCYL	20.50	8.70	136%	19.90	6.30	28%	5.97	6%
PVERT	27.83	8.93	212%	22.03	8.60	4%	6.53	32%
LTURN *	-	5.25	-	-	5.07	3%	-	-

*Trayectoria original para la carga del torno

En un primer momento y de acuerdo con el protocolo de pruebas (Anexo H) cada una de las trayectorias se ejecutó haciendo uso del comando MVS y posteriormente del comando MOV. De esta primera etapa se pueda analizar que las trayectorias programadas con MVS restringen los movimientos del robot, dado que solo permite desplazamiento en línea recta entre los puntos vía, lo que genera que el manipulador se demore más del doble con respecto a las trayectorias ejecutadas con el comando MOV. Adicionalmente, los

tiempos de ejecución de las trayectorias no cambian proporcionalmente debido a la configuración de las posiciones en cada una de ellas, dado que existen puntos que restringen en mayor o menor medida el movimiento.

Además de esto, se encontraron puntos en los cuales no fue posible realizar el movimiento en línea recta (MVS) en el momento de la ejecución de las trayectorias, esto debido a los problemas geométricos que implica el uso de este comando (ver capítulo 5.5.3.1).

La primera modificación que se realizó a la programación de las trayectorias consistió en adicionar el comando para el movimiento continuo (CNT), considerando el hecho de que el manipulador se detenía en cada uno de los puntos vía disminuyendo su velocidad. El comando CNT evita este inconveniente al interpolar a cada lado del punto vía, generando dos pseudo-puntos vía (capítulo 5.5.3.1), aumentando así la suavidad de la trayectoria. Es importante aclarar que la distancia de interpolación debe ser seleccionada cuidadosamente con el fin de evitar colisiones; para este caso las pruebas fueron realizadas con una distancia de interpolación igual a 11 mm, considerando la resolución utilizada en la construcción de la gráfica para la definición del espacio, lo cual asegura que el manipulador permanezca dentro del espacio libre.

Para este caso, se observa igualmente que los tiempos no se modifican proporcionalmente, lo que se evidencia en la trayectoria PZWAIT, la cual mejora en un 31%, mientras que la trayectoria PVERT mejora únicamente en un 4%.

Finalmente, como se expuso en el capítulo 9.8, se realizó un procedimiento de refinamiento, el cual es fundamental para la obtención de una trayectoria con menor cantidad de puntos vía, disminuyendo con esto las distancias de los recorridos, y por tanto, los tiempos de ejecución, que al igual que en los casos anteriores no cambiaron de forma proporcional.

- **Comparación de las trayectorias obtenidas**

En la Tabla 10 se presenta para cada trayectoria el resultado de las variables que caracterizan su desempeño:

TABLA 10. RESULTADOS PARÁMETROS DE COMPARACIÓN DE TRAYECTORIAS

Nombre Trayectoria	Distancias (mm)	Tiempo (seg)	% Comodidad	% Vibración
PZWAIT	2034.10	5.43	40%	35%
PCYL	2057.47	5.97	85%	25%
PVERT	2041.71	6.53	85%	40%
LTURN *	2587.80	5.07	95%	15%

Los valores de los parámetros ‘comodidad’ y ‘vibración’, se establecieron por observación según se explicó en el capítulo 10. La trayectoria PZWAIT, por ejemplo, presenta un porcentaje de comodidad muy bajo por dos razones; en primera instancia, la configuración del manipulador obliga a la articulación J4 a permanecer en una posición en la cual se encuentra cerca del límite, a lo que se suma el efecto de la gravedad que por el peso de la pieza en el efector produce un torque⁴⁹. En segundo lugar, durante el trayecto entre las posiciones P3 y P4, se encontró un punto singular, por lo que el manipulador se ve obligado a realizar un movimiento abrupto para cambiar de dirección, dicho movimiento se considera peligroso tanto para el manipulador como para la pieza misma.

Por su parte, la vibración se evaluó ejecutando las trayectorias a baja velocidad, observando que esta se presenta para todos los casos así sea muy ligera. Se encontró que esto ocurre principalmente para los acercamientos a la primera posición (sobre la estiba), donde el efector final del manipulador se encuentra más alejado de su base, este es el caso de la trayectoria PVERT, en la que aún cuando el porcentaje de comodidad es alto, presenta 40% de vibración.

Con el propósito de seleccionar la mejor trayectoria de acuerdo con la importancia relativa de los parámetros planteados, se diseñó una matriz de ponderación, y se calificó cada trayectoria siguiendo una escala de 1 a 10, donde 10 es el valor mejor calificado:

TABLA 11. MATRIZ DE PONDERACIÓN PARA LA CALIFICACIÓN DE TRAYECTORIAS

Comodidad	20%
Tiempo	50%
Comodidad	20%
Vibración	10%

Nombre Trayectoria	Distancias	Tiempo	Comodidad*	Vibración**
PZWAIT	10	8	4	6.5
PCYL	8	7	8.5	7.5
PVERT	9	4	8.5	6
LTURN	4	10	9.5	8.5

*El valor comodidad corresponde a los porcentajes asignados en la Tabla 10.

**El valor vibración corresponde a 10 menos el valor asignado en la Tabla 10, con el fin de mantener la escala definida.

Nombre Trayectoria	Distancias	Tiempo	Comodidad	Vibración	Suma
PZWAIT	2	4	0.8	0.65	7.45
PCYL	1.6	3.5	1.7	0.75	7.55
PVERT	1.8	2	1.7	0.6	6.1
LTURN	0.8	5	1.9	0.85	8.55

⁴⁹ Esta afirmación se fundamenta en el hecho que durante la medición del espacio de trabajo para esta orientación se encontraron puntos en los cuales el manipulador no era capaz de mantener la posición.

La suma de los puntajes obtenidos para cada trayectoria, indica que de las trayectorias generadas por medio del algoritmo ACS, la más atractiva de acuerdo con la ponderación de parámetros definida, es la trayectoria PCYL, la cual presenta buenos valores en general aún cuando no es la más corta en tiempo de ejecución o distancia recorrida.

La trayectoria actual (LTURN), presenta los mejores valores en tiempo, comodidad y vibración, sin embargo la distancia recorrida es mucho mayor. Esto es consecuencia de que se encuentra construida en un espacio en tres dimensiones, y por lo tanto tiene mayor libertad de movimiento. Sin embargo, se concluye por esta misma razón que la comparación no se está realizando en condiciones iguales, lo cual invita al desarrollo de un modelo de generación de trayectorias robusto que incluya la configuración de un espacio de trabajo en tres dimensiones.

- **Resultados obtenidos para los espacios de trabajo modificados**


Con el fin de comprobar el correcto funcionamiento del algoritmo de generación de trayectorias, y obtener resultados que permitan análisis adicionales, se incluyó un obstáculo en la celda de trabajo y se midió dentro del espacio de cada una de las configuraciones definidas (ver capítulo 9.7).

Teniendo en cuenta que el obstáculo diseñado restringe considerablemente el movimiento del manipulador en el eje Y (Figura 37), los puntos vía encontrados por el algoritmo obligan al manipulador a tomar posiciones que se encuentran cerca de los límites de movimiento de las articulaciones. Esto puede generar que no sea posible el movimiento lineal (comando MVS) entre estos, por lo tanto, es necesario definir con mayor precisión los valores para X y L, para evitar la realización de movimientos abruptos.

TABLA 12. RESULTADOS DE LA IMPLEMENTACIÓN DE LAS TRAYECTORIAS EN LOS ESPACIOS MODIFICADOS

Nombre Trayectoria	Distancias (mm)		Mejora %
	Originales	Refinamiento	
OPZWAIT	2226.69	2078.95	6.63%
OPCYL	2158.02	2040.74	5.43%
OPVERT	2224.48	2072.26	6.84%

Nombre Trayectoria	Originales			CNT		Mejora (MOV vs. MOV CNT)	Refinamiento MOV	Mejora (MOV vs. MOV CNT)
	MVS	MOV	Mejora (MVS vs. MOV)	MVS	MOV			
OPZWAIT	20.45	9.47	116%	19.7	8.63	9%	6.90	25%
OPCYL	11.36	8.83	29%	8.07	8.13	8%	7.00	16%
OPVERT	24.53	9.77	151%	23.70	8.90	9%	6.80	31%

 Estas trayectorias incluyen al menos un movimiento de tipo MOV

Los tiempos y distancias de ejecución para las trayectorias en los espacios modificados, tomaron valores más altos como se esperaba. De la misma forma, los porcentajes de mejora relacionados con las modificaciones en la programación no fueron proporcionales entre las trayectorias, al igual que en las originales.

Una vez refinadas las soluciones, se observa que el tiempo más corto de ejecución de este grupo de trayectorias es OPVERT (6,80 seg), lo cual se esperaba, teniendo en cuenta que desde el momento en que se graficaron los espacios de trabajo el manipulador contaba con mayor libertad de movimiento (Tabla 5. resultados de la ejecución del algoritmo para los espacios con obstáculo). Esto es producto de la orientación de la herramienta cuyo rango de movimiento es más amplio.

12. IMPACTO

A lo largo del desarrollo del proyecto se realizó una extensa revisión bibliográfica, en la cual se encontró que existen diversos métodos y propuestas de solución al problema de generación de trayectorias de tiempo mínimo para robots industriales. La mayoría de estas investigaciones se concentran en la solución del problema manejando espacios de trabajo en 3D, factor que eleva la complejidad en la construcción y análisis de modelos de generación y optimización de trayectorias. Como parte del análisis de los antecedentes, se evidenció que gran parte de los autores de este tipo de modelos propuestos son doctores o candidatos a doctor en universidades alrededor del mundo.

El desarrollo de este proyecto se considera valioso al presentarse como una alternativa de solución a un problema con la complejidad mencionada, teniendo en cuenta que aún cuando las soluciones están sujetas a restricciones, se consiguen buenos resultados que pueden ser utilizados de manera eficaz en una celda de manufactura real. De igual manera, la metodología desarrollada tiene gran potencial, dado que hace posible que su implementación la realicen personas no necesariamente expertas en generación de trayectorias para robots. De igual manera, tanto la metodología como la aplicación se conforman como una herramienta pedagógica que está en capacidad de aclarar conceptos e impulsar futuros desarrollos relacionados con el tema.

Por otra parte, debe notarse que el modelo propuesto puede adaptarse fácilmente para la solución de otro tipo de problemas, bien sea en robótica, en problemas de ‘transporte’ de materiales, o en otras áreas en problemas como el desplazamiento de personal en planta, todo esto gracias a que la aplicación desarrollada es capaz de procesar cualquier imagen en 2D que pueda ser interpretada como un espacio de trabajo.⁵⁰

A continuación se ilustra por medio de un ejemplo sencillo una aplicación alternativa del módulo de generación de trayectorias de distancia mínima haciendo uso del algoritmo ACS:

Suponga que desea reducir los tiempos de transporte de material al interior de un almacén. Actualmente el transporte es realizado por medio de un vehículo auto guiado⁵¹, sin embargo no se conoce cuál es la trayectoria más adecuada para el vehículo.

Siguiendo los pasos de la metodología:

- 1. Analizar la operación para identificar restricciones y objetivos de la mejora.*
- 2. Medir el área de trabajo definiendo los lugares de la superficie por donde no puede transitar el vehículo.*
- 3. Definir el espacio de configuración por medio de la representación gráfica de la superficie.*

A continuación se ilustra un ejemplo de la definición de la superficie de trabajo

⁵⁰ Imagen de mapa de bits en blanco y negro donde los pixeles blancos son considerados como el espacio ocupado por los obstáculos.

⁵¹ Vehículo robótico no tripulado también conocido como AGV por sus siglas en inglés (Auto-guided vehicle).

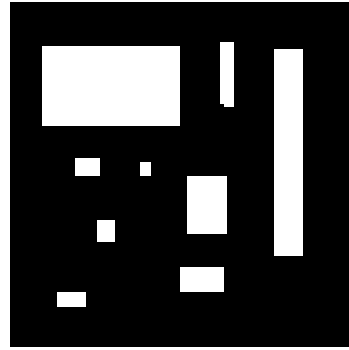


FIGURA 38. EJEMPLO DEFINICIÓN DEL ESPACIO DE LA SUPERFICIE DE UN ALMACEN PARA UN VEHÍCULO AUTOGUIADO

4. Ejecutar el Algoritmo ACS

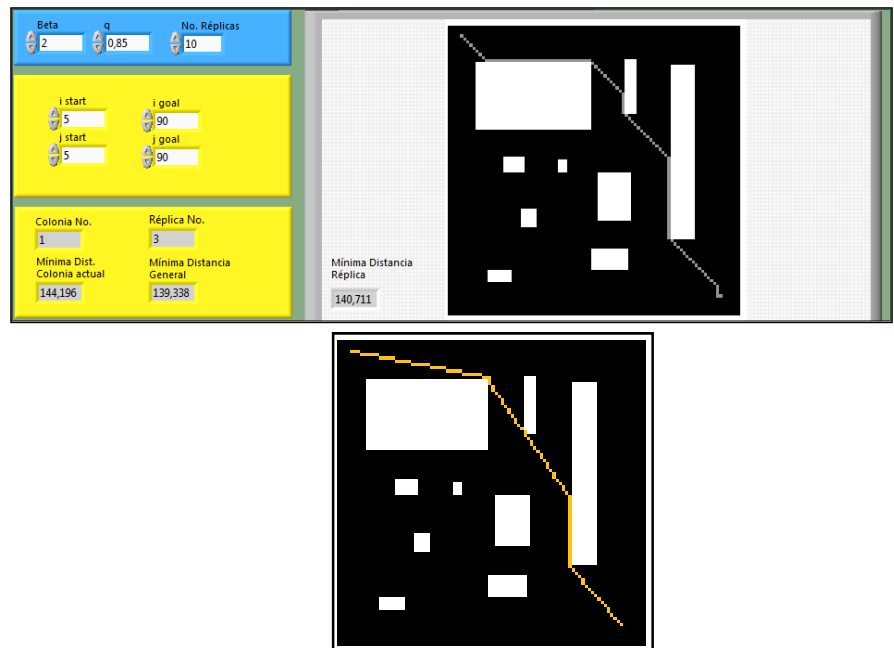


FIGURA 39. EJEMPLO EJECUCIÓN DEL ALGORITMO ACS PARA LA DEFINICIÓN DE LA SUPERFICIE DE UN ALMACEN

5. Traducir la trayectoria encontrada por el algoritmo a términos de programación del vehículo auto guiado.

El ejemplo pretende demostrar que el algoritmo puede brindar buenos resultados para un problema de esta naturaleza, siempre y cuando se analicen correctamente las características del entorno y de acuerdo con esto se interpreten adecuadamente los resultados.

En la figura se observa como el algoritmo ACS es capaz de evadir los obstáculos entre los puntos inicial y final, recorriendo la menor distancia posible. Sin embargo, y como se mencionará más adelante, las soluciones podrían ser aún mejores haciendo uso de procedimientos de búsqueda local.

NOTA: Los espacios de trabajo deberán representarse siempre como espacios de configuración (c-space) de forma que los obstáculos allí representados están “agrandados”, según se describió en el capítulo 5.8.

13. CONCLUSIONES

El objetivo principal de este trabajo se centró en el desarrollo de una aplicación como propuesta de solución al problema de generación de trayectorias para manipuladores industriales. Para esto fue necesario realizar una revisión bibliográfica, donde se recopiló información sobre diferentes metodologías que se han utilizado para resolver el problema; desde modelos cinemáticos, hasta diferentes metaheurísticas como algoritmos genéticos, recocido simulado y colonia de hormigas, las cuales han demostrado encontrar buenas soluciones en tiempos cortos de ejecución, por lo que fueron seleccionadas como metodología a estudiar para el desarrollo de la aplicación.

Durante el desarrollo del trabajo, fue necesario definir el espacio de trabajo del manipulador, lo que se presentó como un nuevo problema a resolver, y para lo cual fue necesaria una revisión de antecedentes con el fin de establecer el método más adecuado que se adaptara al problema específico en la celda del CTAI. Se encontraron entonces autores como Jean Latombe, que explican y definen con claridad las metodologías utilizadas para este fin, y se concluyó que se trabajaría el espacio de configuración a una altura Z determinada y en dos dimensiones con el fin de simplificar el problema.

La selección de la metodología para la definición del espacio de trabajo y el algoritmo a implementar para la búsqueda de soluciones, se realizaron en conjunto teniendo en cuenta que la primera define la forma en que se realiza la búsqueda de trayectorias. La descomposición por celdas se presentó como la mejor opción para la representación del espacio, con la ventaja que toma toda la información de la gráfica, en contraste con los grafos de visibilidad que se limitan a los vértices de los obstáculos. En consecuencia, se decidió que las ventajas que ofrece el algoritmo ACS podían ser aprovechadas en esta configuración del espacio, y se procedió a desarrollar la aplicación en LabVIEW, software que se encuentra disponible en el laboratorio para futuros trabajos, es fácil de utilizar y ofrece versatilidad.

Se concluye de esta primera etapa del proyecto que es necesario el acercamiento a los antecedentes de solución de los problemas para conocer el estado del arte y determinar las características de cada metodología que permitan la selección de la más adecuada.

La siguiente etapa que se llevo a cabo correspondió al desarrollo de la aplicación, para lo cual fue necesario establecer los valores de los nueve parámetros de ejecución del algoritmo. Teniendo en cuenta que el ajuste de dichos parámetros para cada espacio de configuración, requiere un número elevado de experimentos y análisis, se consideró conveniente asignar algunos valores preestablecidos por autores como Dorigo y Gambardella, quienes fueron los desarrolladores del algoritmo, y por tanto, recomiendan valores confiables para la ejecución del mismo.

Por otra parte, para los parámetros restantes se ejecutaron pruebas, y se realizó un análisis de las medidas de tendencia y dispersión de los resultados, teniendo en cuenta la naturaleza probabilística del modelo. Se observó que para $\beta = 2$ y resolución 21, se obtuvo el valor más bajo, sin embargo, la combinación de estimadores indicó que en promedio la combinación de parámetro $\beta=2$ y resolución 11,

presentaba mejores resultados, siendo además una resolución que aprovecha más información de la gráfica, en un tiempo de ejecución razonable. Se concluye además que la resolución es un parámetro importante para la ejecución del algoritmo, ya que los resultados en cada una de ellas varían considerablemente, obteniendo para la menor (5) los valores de distancia más altos en tiempos de ejecución elevados.

Una vez ajustados los parámetros se ejecutó el algoritmo 10 veces para cada espacio de configuración. Aún cuando los resultados encontrados se consideran buenos tanto en distancia total recorrida por las hormigas, como en tiempos de ejecución, se observó que era posible mejorar las soluciones considerando el hecho de que la búsqueda en gradilla limita el movimiento de las hormigas, restringiendo los caminos en ángulos diferentes a 45° y 90°, y que la información heurística impide que estas busquen caminos que se encuentran en direcciones alejadas de la meta, cuando esto en ocasiones es necesario. Como una solución a esta situación, se desarrolló un procedimiento de refinamiento, con el cual no solo se disminuyó la distancia total para cada caso, sino que facilitó la programación e implementación de las soluciones al reducir la cantidad de puntos vía en cada espacio definido.

Durante el proceso de implementación de las soluciones se concluyó que es necesario tener en cuenta aspectos de la programación del robot que aportan a la generación de trayectorias más suaves y permiten obtener mejores tiempos de ejecución de las mismas. En el análisis de resultados debe por tanto considerarse no solo las variables tiempo y distancia para la optimización de trayectorias, sino también aspectos como la suavidad del movimiento (continuidad), las vibraciones del mecanismo, y la “comodidad” del manipulador al ejecutar los movimientos entre puntos vía, los cuales en conjunto determinan las características ideales para una trayectoria.

Al analizar los resultados obtenidos frente a la trayectoria actual, se llegó a la conclusión que las restricciones impuestas para la construcción del modelo son tales, que no permiten una comparación directa de algunos parámetros como por ejemplo la comodidad del manipulador, puesto que para este trabajo se fijan una altura y orientación determinada que impiden el libre movimiento del robot. No obstante se considera que las soluciones encontradas podrían calificarse aún mejor que la actual gracias a que los tiempos de ejecución no son muy lejanos y estas últimas se construyen con fundamentación matemática.

Llevando los resultados a un contexto real, es importante mencionar que el uso de una metodología para la generación de trayectorias en la cual se ha definido completamente el espacio de configuración permite la generación de caminos libres de colisiones fuera de línea, lo que significa que las operaciones del manipulador no se detendrán al no ser necesario el uso del mando manual (*teach box*), evitando con esto retrasar procesos productivos o logísticos relacionados al robot.

Para finalizar, la aplicación desarrollada se presenta como una solución que impacta en diferentes eslabones de la cadena de valor de la industria manufacturera, teniendo en cuenta que el valor agregado de la automatización no es la tecnología por sí misma, sino la forma en que esta sea aprovechada a partir del análisis de las condiciones de los procesos, lo cual conduce a resultados que impactan la eficiencia y por tanto rentabilidad de los mismos.

14. RECOMENDACIONES

- Durante el desarrollo de la metodología planteada se seleccionaron diferentes métodos y heurísticas de acuerdo con los criterios expuestos para la construcción del modelo de generación de trayectorias. Sin embargo, debido a imposibilidad de analizar y comparar todos los posibles métodos y procedimientos, no hay garantía de que los seleccionados sean los más adecuados para solución del presente problema. Se recomienda para futuros trabajos que se desarrolle un algoritmo diferente y se comparen los resultados con ACS.
- Con el fin de obtener soluciones que permitan hacer eficiente del movimiento del robot, se recomienda el desarrollo del modelo de solución para el espacio de configuración en 3D y con libertad en la orientación del manipulador. De la misma manera, sería interesante incorporar variables de optimización cuantitativas como el torque una vez se halla sido desarrollado este modelo.
- Durante la fase de pruebas de los procedimientos de definición del espacio de configuración se identificó que estos pueden ser optimizados por medio del uso de modelos jerárquicos como *quadtrees*, evitando la pérdida de información, dado que estos no definen cuadrillas libres como obstáculo para ningún caso.
- En el desarrollo del algoritmo se definieron diferentes parámetros por medio de experimentación y referencia bibliográfica, sin embargo se considera que el modelo podría tener un mejor desempeño si se llevaran a cabo experimentos para calibrar dichos parámetros en cada situación (cada espacio definido teniendo en cuenta diferentes resoluciones).
- Se recomienda extender el estudio a las diferentes trayectorias del manipulador en el CTAI, incluyendo las estaciones de fresado, visión artificial y banda transportadora.

BIBLIOGRAFÍA

- [1] Iñigo Madrigal, R., & Vidal Idiarte, E. (2004). *Robots Industriales Manipuladores* (1ra ed.). Alfa Omega.
- [2] Zambrano Rey, G., & Rosas Castro, L. F. (2008). *Manual de funcionamiento del robot Mitsubishi MELFA IV y el software Cosimir Professional, integrados al sistema de manufactura integrada por computador CIM del Laboratorio de Automatización Industrial*. Bogotá.
- [3] Niebel, B. W. (1996). *Ingeniería Industrial: Métodos, Tiempos y Movimientos*. México: Alfa Omega 9 ed.
- [4] MITSUBISHI ELECTRIC CORPORATION. (2002). *MELFA Industrial Robot Instruction Manual (Functions & Operations)*. Alemania.
- [5] Craig, J. J. (2006). *Robótica* (3ra. ed.). Prentice Hall.
- [6] Festo Didactic GmbH & Co.KG. (2003). *iCIM CNC Coupling A004*.
- [7] Latombe, J. C. (1991). *Robot motion planning*. Norwell: Kluwer academic publishers.
- [8] ScienceDirect: Philip, A., & Aspragathos, N. (2005, Mayo 31). *Obstacle representation by Bump-surfaces for optimal motion planning*. Retrieved Agosto 2, 2009, from Robotics and Autonomous Systems, Volume 51, Issues 2-3, Pages 129-150.
- [9] ScienceDirect: Guzmán, J., Berenguel, M., & Rodríguez, F. (2007, Octubre 7). *An interactive tool for mobile robot motion planning*. Retrieved Agosto 2009
- [10] Kravaki, L. E. (n.d.). *Robotic Path Planning and Protein Molding*. Retrieved Septiembre 2009, from <http://cnx.org/content/m11457/latest/>
- [11] Han, K. m. (2007, Agosto). *Collision free path planning algorithms for robot navigation problem*. Retrieved Marzo 24, 2009, from <http://edt.missouri.edu/Summer2007/Thesis/HanK-062008-T8352/research.pdf>
- [12] Hincapie, R., Ríos, C., & Gallego, R. (2003). *TÉCNICAS HEURÍSTICAS APLICADAS AL PROBLEMA DEL CARTERO VIAJANTE (TSP)*. Recuperado el Julio de 2009, de www.utp.edu.co/~planeamiento/.../Tecnicas_heuristicas_TSP4.pdf
- [13] de San Pedro, M., Pandolfi, D., Lasso, M., & Villagra, A. (2005). *Metaheurística ACO aplicada a problemas de planificación en entornos dinámicos*. Retrieved Julio 2009, from <http://www.ing.unp.edu.ar/wicc2007/trabajos/ASI/025.pdf>
- [14] Alonso, S., Córdón, O., & Fernández de Viana, I. (2003). *La Metaheurística de Optimización Basada en Colonias de Hormigas: Modelos y Nuevos Enfoques*. Retrieved Agosto 2009, from

<http://sci2s.ugr.es/publications/ficheros/OCH%20Modelos%20y%20Nuevos%20Enfoques%20%28Chapter%29.pdf>

[15] Vázquez Hurtado, C. (2005, Septiembre). Planificación de Trayectorias Basada en optimización mediante colonia de hormigas (OCH).

[16] ScienceDirect: TAN, G.-Z., HE, H., & SLOMAN, A. (2007, Marzo). *Ant Colony System Algorithm for Real-Time Globally Optimal Path Planning of Mobile Robots*. Retrieved Agosto 2009

[17] (s.f.). Obtenido de COSIMIR® Short introduction into 3D Simulation and Offline Programming of robot-based workcells with COSIMIR®: <http://www.koningshartman.com/NR/rdonlyres/DD76487B-95F3-4F78-9458-68986040EE4D/0/COSIMIRGettingStartedE.pdf>

[18] Technology, C. T. (s.f.). Obtenido de <http://www.motionshop.com/pr/festocosimir.shtml>

[19] INSTRUMENTS, N. (n.d.). *Qué es LabVIEW?* Retrieved from <http://www.ni.com/labview/whatis/esa/>

[20] *Tutorial de LabVIEW.* (s.f.). Obtenido de http://www.gte.us.es/ASIGN/IE_4T/Tutorial%20de%20Labview.pdf

[21] IEEE Xplore: Dorigo, M., & Gambardella, L. M. (Abril de 1997). *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*. Obtenido de IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 1, NO. 1.

[22] Gasparetto, A., & Zanotto, V. (2005, Abril). *A new method for smooth trajectory planning of robot manipulators*. Retrieved Octubre 22, 2009

[23] Seshadri, C., & Ghosh, A. (Abril de 1993). Optimum Path Planning for Robot Manipulators Amid Static and Dynamic Obstacles. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS VOL. 23, NO. 2* .

[24] Lee, s., & Kardaras, G. (1997). Collision-Free Path Planning with Neural Networks. *Proceedings of the 1997 IEEE International Conference on Robotics and Automation* .

[25] Heim, A., & Von Stryk, O. (1999). Trajectory Optimization of industrial Robots with Applications to CAR and Robot Controllers. *Optimization* .

[26] Dudek, G., & Simhon, S. (13 de Septiembre de 2003). Path Planning Using Learned Constraints and Preferences. *Proceedings of the 2003 IEEE International Conference on Robotics and Automation* .

[27] Kiiqiik, S., & Bingul, Z. (2004). The Inverse Kinematics Solutions of Industrial Robot Manipulators. *IEEE* .

[28] Chasparis, G. C., & Shamma, J. S. (2005). Linear-Programming-Based Multi-Vehicle Path Planning with Adversaries. *American Control Conference* .

[29] Coy Calixto, C. A. (2005). Implementación en Hidroinformática de un Método de Optimización Matemática Basado en la Colonia de Hormigas. Pontificia Universidad Javeriana.

- [30] Mohamad, M. M., Taylor, N. K., & Dunnigan, M. W. (Septiembre de 2006). Articulated Robot Motion Planning Using Ant Colony Optimisation. *3rd International IEEE Conference Intelligent Systems* .
- [31] Mei, H., Yantao, T., & Zu, L. (2006). *A Hybrid Ant Colony Optimization Algorithm for Path Planning of Robot in Dynamic Environment*. Retrieved Marzo 24, 2009, from Journal of Information Technology, Vol.12, No.3: http://www.icis.ntu.edu.sg/scs-ijit/1203/1203_8.pdf
- [32] ScienceDirect: Donati, A., Montemanni, R., Casagrande, N., Rizzoli, A., & Gambardella, L. (2006, Junio 1). *Time dependent vehicle routing problem*. Retrieved Mayo 2009, from European Journal of Operational Research, Volume 185, Issue 3.
- [33] Zeqing, Y., Libing, L., & Weidong, Y. (2008). Flexible Inspection Path Plannig based on Adaptive Genetic Algorithm. *Chinese Control and Decision Conference* .
- [34] ScienceDirect: Chen, M., & Zalzal, A. (1997, Enero). *Dynamic modeling and genetic-based trajectory Generation for non-holonomic mobile manipulators*. Retrieved Agosto 2, 2009, from Control Engineering Practice Volume 5, Issue 1.
- [35] IEEE Xplore: Bellier, C., Mazer, E., Troccaz, J., & Augier, C. (1991, Noviembre). *Planning/executing six d.o.f. robot motions in complex environments*. Retrieved Agosto 8, 2009, from IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS.
- [36] IEEE Xplore: Zhu, D., & Latombe, J.-C. (1991, Febrero). *New Heuristic Algorithms for Efficient Hierarchical Path Planning*. Retrieved Agosto 4, 2009, from IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 7, NO. 1.
- [37] ScienceDirect: Ting, Y., Lei, W., & Jar, H. (2002, Abril). *A path planning algorithm for industrial robots*. Retrieved Agosto 2, 2009, from Computers & Industrial Engineering, Volume 42, Issues 2-4, 11 April 2002, Pages 299-308.
- [38] ScienceDirect: Dash, A., Chen, I.-M., & Yeo, S. H. (2007, Julio). *Workspace generation and planning singularity-free path for parallel manipulators*. Retrieved Agosto 2, 2009, from Mechanism and Machine Theory Volume 40, Issue 7, Pages 776-805.
- [39] Jiménez B., D., & Peñuela, D. F. (n.d.). Escenarios para el aprendizaje de tendencias bioinspiradas.
- [40] Ollero Bauterone, A. En *ROBOTICA MANIPULADORES Y ROBOTS MOVILES* (págs. 200-225). Madrid: Marcombo.
- [41] Marco, I. D. (n.d.). *Organización Argentina de Producción, Logística y Operaciones*. Retrieved Marzo 6, 2009, from <http://www.oaplo.com.ar/Articulos/Tecnologia-1005.pdf>
- [42] Balseiro, S. (n.d.). Algoritmos para problemas de ruteo de vehículos con restricciones de capacidad y ventanas de tiempo. *TESIS DE INGENIERÍA INDUSTRIAL Logística y Distribución* . Buenos Aires, Argentina: Facultad de ingeniería.

- [43] *Probabilistic Roadmap - PRM Basic*. (s.f.). Recuperado el 23 de Febrero de 2009, de ai.stanford.edu/~latombe/cs26n/2007/prm-basic.ppt
- [44] MITSUBISHI ELECTRIC CORPORATION. (2002). *CR1/CR2/CR2A Controller Instruction Manual - Explanations of MOVEMASTER COMMANDS*. Japón.
- [45] MITSUBISHI ELECTRIC CORPORATION. (2003). *CR1 Controller Instruction Manual*. Japón.
- [46] Inser-robótica. (s.f.). Recuperado el 13 de Agosto de 2009, de <http://www.inser-robotica.com/Imagenes/Fundicion2y3Mod.jpg>
- [47] Federation, I. I. (s.f.). Recuperado el 13 de Agosto de 2009, de <http://www.inser-robotica.com/Imagenes/paletizadoBotes02.jpg>
- [48] Kawasaki Robotics GmbH. (s.f.). Recuperado el 13 de Agosto de 2009, de <http://www.inser-robotica.com/PDFS/CatalogoGralKawa.pdf>
- [49] DONG-HOON, Y., & SUK-KYO, H. (2005, Octubre). *A roadmap construction algorithm for mobile robot path planning using skeleton maps*. Retrieved Agosto 2009, from *Advanced Robotics*, Vol. 21, No. 1–2, pp. 51–63 (2007).
- [50] ScienceDirect: Chen, M., & Salzala, A. (1997). *DYNAMIC MODELLING AND GENETIC-BASED TRAJECTORY GENERATION FOR NON-HOLONOMIC MOBILE MANIPULATORS*. Retrieved Septiembre 2009
- [51] IEEE Xplore: Faverjon, B. (1984). *OBSTACLE AVOIDANCE USING AN OCTREE IN THE CONFIGURATION SPACE OF A MANIPULATOR*. Retrieved Agosto 2009
- [52] IEEE Xplore: Edward Red, W., & Hung-Viet, T.-C. (1983). *THE CONFIGURATION SPACE APPROACH TO ROBOT PATH PLANNING*. Retrieved Julio 2009

ANEXOS

A. ANTECEDENTES PARA LA COMPARACIÓN DEL ALGORITMO

FECHA	AUTORES	DESCRIPCIÓN	MÉTODOS UTILIZADOS	REFERENCIA
1993	Commuri Seshadri, Arindam Ghosh	Se plantea la solución a los problemas de planeación de trayectorias de tiempo mínimo y trayectorias de energía mínima, por medio de métodos de variación local.	Method of local variations (MLV) .	“Optimum Path Planning for Robot Manipulators Amid Static and Dynamic Obstacles” <i>IEEE transactions on systems, man, and cybernetics, vol. 23, no. 2, March-April 1993</i> [23]
1997	Sukhan LeeGeorge Kardaras	Se plantea un enfoque en el que se representa la trayectoria por medio de una serie de puntos vía conectados entre sí por ligamentos o hilos elásticos de forma que están sujetos a ser desplazados debido a las posibles colisiones con obstáculos en el área de trabajo.	Artificial Potential Field (APF) , Simulated Annealing (SA) , y Neural Networks (NN) .	“Collision-free path planning with neural networks” <i>Proceedings of the 1997 IEEE International Conference on Robotics and Automation Albuquerque, New Mexico - April 1997</i> [24]
1999	Alexander Heim, Oskar Von Stryk	Se describe el método para solucionar el problema de optimización de trayectorias restringidas usando modelos dinámicos de robótica, incorporando al modelo dinámico variables como la inercia, las aceleraciones centrífugas, gravitacionales, de coriolis y las fuerzas de fricción.	Cinemática Inversa (IK) en modelos dinámicos, Deterministic Optimal Control.	“Trajectory optimization of industrial robots with application to Computer-Aided Robotics and Robot Controllers” <i>Optimization 1999</i> [25]
2003	Gregory Dudek and Saul Simhon	Se propone un método para la planeación de trayectorias basado en aprendizaje de patrones de movimiento utilizando leyes de probabilidad.	Hidden Markov Model (HMM) – Modelo de Markov Oculto	“Path Planning Using Learned Constraints and Preferences” <i>Proceedings of the 2003 IEEE international Conference of Robotics & Automation. Taipei, Taiwan, September 14-19. 2003</i> [26]

2004	Serdar Kucuk Zafer Bingül	Se halla la solución de la cinemática inversa para 16 robots manipuladores industriales de diferentes clasificaciones.	Cinemática Inversa (IK)	“The Inverse Kinematics Solutions of Industrial Robot Manipulators” <i>Proceedings of the IEEE International Conference on Mechatronics, 2004 Volume, Issue , 3-5 June 2004 Page(s): 274 – 279 [27]</i>
2005	Chasparis, G.C.; Shamma, J.S.	Se expone un algoritmo de planeación de trayectorias basado en programación lineal, que permite obtener trayectorias óptimas para un grupo de vehículos autónomos en un lugar de trabajo, se plantean los diferentes vehículos como adversarios y se soluciona el sistema como un problema de asignación de recursos.	Programación lineal (LP) , Receding Horizon for multiple vehicles.	“Linear-Programming-Based Multi-Vehicle Path Planning with Adversaries” <i>“Proceedings of the 2005 American Control Conference.”</i> <i>Volume , Issue , 8-10 June 2005 Page(s): 1072 - 1077 vol. 2 [28]</i>
2005	Carlos Alfredo Coy Calixto	El artículo presenta en forma clara los pasos a seguir en la programación del algoritmo ACO aplicado a un problema particular en el campo de la hidroinformática.	Ant Colony Optimization (ACO)	“Implementación en Hidroinformática de un método de optimización matemática basada en la colonia de hormigas.” <i>Trabajo de grado, Ingeniería Civil PUJ [29]</i>
2005	Carlos Vázquez Hurtado, Jan Rosell Gratacós	Expone un desarrollo teórico de las soluciones a los problemas de generación de trayectorias. Hace uso de la metaheurística ACO para encontrar la trayectoria óptima para un robot que va de un punto inicial a un punto final, genera un programa en C++ y compara el desempeño del algoritmo con otros similares.	Colonia de Hormigas OCH (ACO)	Planificación de Trayectorias Basada en Optimización Mediante Colonias de Hormigas [15]

2006	Mohd Murtadha Mohamad, Nicholas K. Taylor, y Matthew W. Dunnigan	Describe la aplicación de la inteligencia artificial basada en el comportamiento de colonias de hormigas, a la solución del problema de planeación del movimiento de un brazo robótico, apoyándose en el uso de técnicas de mapeo probabilístico.	Ant Colony Optimization (ACO) , Probabilistic roadmap planner (PRM) .	“Articulated Robot Motion Planning Using Ant Colony Optimization.” <i>3rd International IEEE Conference Intelligent Systems, September 2006 [30]</i>
2006	Hao Mei, Yantao Tian, Linan Zu	Propone una metodología en la cual se usa ACO para planear la ruta global basado en información estática, mientras que a través de APF se encuentra la ruta local. Se encuentra también una comparación de ACO con algoritmo genético.	Artificial Potential Field (APF) , Ant Colony Optimization (ACO) .	“A Hybrid Ant Colony Optimization Algorithm for Path Planning of Robot in Dynamic Environment”. <i>International Journal of Information Technology, Vol.12, No.3, 2006 [31]</i>
2007	Kyung min Han	Expone el problema de generación de trayectorias. Haciendo uso de un gráfico en 2D, se generan obstáculos aleatorios (círculos) y se evalúa el desempeño de dos algoritmos diferentes en función de la distancia y la evasión de obstáculos.	Ant Colony Optimization (ACO) . Genetic Algorithm (GA) .	“Collision free path planning algorithms for robot navigation problem”. <i>Faculty of the Graduate School University of Missouri-Columbia. Presentada como Trabajo de grado para la obtención del título de Maestría en Ciencias. [11]</i>
2006	Alberto Donati, Roberto Montemanni, Norman Casagrande, Andrea Rizzoli, Luca Gambardella.	Presenta una solución al problema de ruteo de vehículos con ventanas de tiempo, teniendo en cuenta la variación de la velocidad de tránsito durante el día. En el modelo, se hace uso de dos colonias de hormigas, en la que una colonia minimiza el tiempo total de tránsito, mientras la segunda toma las mejores soluciones de la primera y busca minimizar el número de ‘tours’ que deben programarse. Se utilizan y explican condiciones de búsqueda que permiten mejorar la calidad de las soluciones halladas.	Multi Ant Colony System (MACS) .	“Time dependent vehicle routing problem with a multi ant colony system” <i>European Journal of Operational Research Volume 185, Issue 3. [32]</i>

2008	Yang Zeqing Liu Libing Yang Weidong	Propone una metodología rápida para hallar soluciones cercanas al óptimo para problemas grandes de <i>“flexible inspection path planning”</i> , haciendo uso del algoritmo AGA, el cual se basa en el teorema darwiniano de la selección natural.	Adaptive Genetic Algorithm (AGA)	“Flexible inspection path planning based on Adaptive Genetic Algorithm.” <i>Control and Decision Conference, China 2008.</i> [33]
1996	M.W. Chen and A.M.S. Zalzal	Expone un modelo para la generación de trayectorias en manipuladores instalados sobre bases móviles. El modelo incluye restricciones no-holónicas, ya que se tienen en cuenta que las posibles trayectorias del efector final dependen de la posición determinada de la base móvil a lo largo del tiempo. El artículo hace énfasis en la solución de los sistemas de fuerza y torque en el robot, tratando de forma breve los aspectos de definición del área de trabajo, evasión de obstáculos y generación de caminos cercanos al óptimo.	Genetic Algorithm (GA) .	“Dynamic modeling and genetic-based trajectory Generation for non-holonomic mobile manipulators” <i>Control Engineering Practice Volume 5, Issue 1, January 1997.</i> [34]

B. ANTECEDENTES PARA LA COMPARACIÓN DE LOS MÉTODOS DE REPRESENTACIÓN DEL ESPACIO DE TRABAJO

FECHA	AUTORES	DESCRIPCIÓN	MÉTODOS UTILIZADOS	REFERENCIA
1991	C. Bellier, C. Augier, Mazer, Troccaz, E. J.	Por medio de un modelo CAD del espacio de trabajo, se construye el espacio de configuración <i>C-space</i> . El método consiste inicialmente en calcular recursivamente los rangos de parámetros que generan colisión con un obstáculo en el espacio de trabajo, evaluando planos generados por cada obstáculo y cada articulación, obteniendo distancias d^* (rangos máximos). Después de esto, se construyen octrees donde se estructura el espacio libre de obstáculos, representando el espacio por nodos adyacentes, relacionados por una gráfica de conectividad.	<ul style="list-style-type: none"> - Modelación del espacio por medio de modelos jerárquicos - Computación de distancia entre eslabones y objetos - Algoritmo de computación de configuraciones válidas - Modelo aproximado del espacio de configuración combinando diferentes 2^n-trees 	<p>“Planning/executing six d.o.f. robot motions in complex environments.” <i>IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS '91. Nov. 3-5. 1991</i> [35]</p>
1991	David Zhu, Jean-Claude Latombe	<p>Descompone el <i>c-space</i> del objeto en movimiento (robot) en células rectangulares. Las celdas se clasifican en VACIAS, LLENAS o MIXTAS dependiendo de si se encuentran completa o parcialmente fuera o dentro de los obstáculos.</p> <p>El algoritmo busca la gráfica que conecte una cadena de celdas VACIAS entre las configuraciones inicial y final deseadas; si no lo consigue, descompone la celdas MIXTAS en celdas más pequeñas, las clasifica nuevamente y realiza una nueva búsqueda. El proceso termina cuando se encuentra una solución, cuando se comprueba que no existe tal, ó, cuando las celdas MIXTAS son más pequeñas de cierto tamaño</p>	<ul style="list-style-type: none"> - Descomposición en celdas de forma jerárquica - Hierarchical approximate cell decomposition 	<p>“New Heuristic Algorithms for Efficient Hierarchical Path Planning” <i>IEEE transactions on robotics and automation, vol. 7, No. 1.</i> [36]</p>

		predeterminado. Este algoritmo da como resultado un canal más que un simple camino.		
2002	Y. Ting, W. I. Lei	El método utiliza en general una estructura ramificada basada en <i>quadtrees</i> para segmentar el espacio en numerosas áreas representadas por nodos que se categorizan como libres u obstáculos, denotando las áreas transitables y las áreas prohibidas. Se utilizan mapas de distancia para transformar el espacio continuo a tres mapas 2D con cuadrículas y nodos equidistantes (distancia unitaria). Wave expansión se aplica marcando cada nodo libre con la distancia al nodo-obstáculo más cercano. La búsqueda de posibles caminos se apoya en la estructura jerárquica que une los nodos adyacentes.	<ul style="list-style-type: none"> - Distance maps method - Wave expansión based don hierachical tree structures 	<p>“A planning algorithm for industrial robots” <i>Computers & Industrial Engineering Volume 42, Issues 2-4, 2002. [37]</i></p>
2005	Anjan Kumar Dash, I-Ming Chen, Song Huat Yeo, Guilin Yang	Se presenta una técnica matemática para determinar trayectorias de un robot con 3 manipuladores paralelos. Cada uno se modela como una esfera y se divide el espacio en capas verticales. El volumen del espacio de trabajo se determina por la unión de las capas. Se generan “clusters” de singularidades en cada capa para encontrar los obstáculos dentro del espacio de trabajo.	Partición del espacio de trabajo en capas y discretización de sectores.	<p>“Workspace generation and planning singularity-free path for parallel manipulators” <i>Mechanism and Machine Theory Volume 40, Issue 7, July 2005, Pages 776-805. [38]</i></p>

C. INFORMACIÓN ADICIONAL SOBRE LA METAHEURÍSTICA ACO

Con el fin de profundizar un poco sobre el concepto de la metaheurística ACO, se presenta la siguiente figura, donde se demuestra la capacidad de las mismas para encontrar el camino más corto:

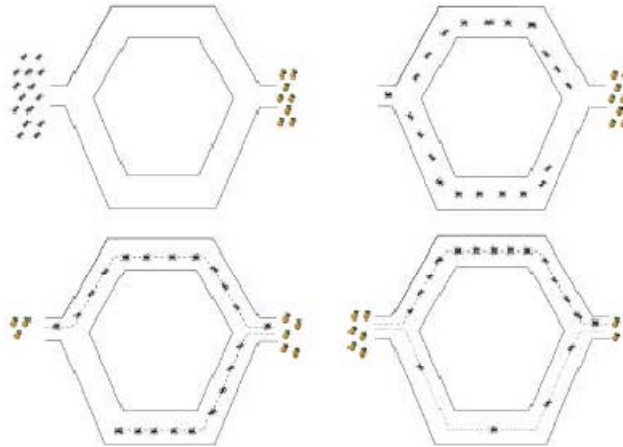


FIGURA 40. BÚSQUEDA DEL CAMINO MÁS CORTO. [21]

En la segunda figura, se observa que igual número de hormigas toman los caminos largo y corto, en la tercera, es evidente que las hormigas que tomaron el camino más corto, regresan más rápido al nido, mientras que las que tomaron el camino largo demoran. Finalmente, debido a que los rastros de feromona se evaporan, el camino largo no tendrá una cantidad de feromona representativa, en contraste con el camino corto, por donde seguirán las hormigas restantes de la colonia.

Este comportamiento de las hormigas, en el que utilizan una sustancia llamada feromona para pasar la información de un individuo a otro en el proceso de generación de trayectorias cortas, es lo que se conoce como estigmergia.

El modelo matemático que realizaron Dorigo y Gambardella en 1992, representa los conceptos de este comportamiento de la siguiente forma [39] :

BIOLOGÍA	MODELO MATEMÁTICO
Hormigas	<i>Agente Artificial</i>
Feromona	<i>Dato de feromona artificial</i>
Comida	<i>i start</i>
Hormiguero	<i>j goal</i>
Obstáculo	<i>Restricción</i>
Vía	<i>Aristas del grafo</i>
Ecosistema	<i>Matriz de imagen</i>
Colonia	<i>Grupo de agentes, información después de x iteraciones</i>

Similitudes y Diferencias Entre las Hormigas Naturales y Artificiales [14]

SIMILITUDES	DIFERENCIAS
Tanto las hormigas naturales como las artificiales modifican su “entorno” a través de una comunicación estímulo-respuesta basada en la feromona.	En el caso de las hormigas artificiales, los rastros de feromona artificiales son valores numéricos que están disponibles únicamente de manera local.
comparten una tarea común: la búsqueda del camino más corto (construcción iterativa de una solución de costo mínimo) desde un origen, el hormiguero (decisión inicial), hasta un estado final, la comida (última decisión).	Los algoritmos ACO pueden enriquecerse con habilidades adicionales como la capacidad de mirar más allá de la siguiente transición (“lookahead”), la optimización local y “backtracking”, que persiguen mejorar la eficacia.
Las hormigas artificiales construyen las soluciones iterativamente aplicando una estrategia de transición local estocástica para moverse entre estados adyacentes, tal como hacen las hormigas naturales.	Las hormigas artificiales viven en un mundo discreto y tienen algunas capacidades adicionales.
Uso de una colonia de individuos que interactúan y colaboran para solucionar una tarea dada.	Las hormigas artificiales pueden hacer uso de la información heurística (no solo de los rastros locales de feromona) en la política estocástica de transición que aplican.

Algoritmos de Optimización Basada en ACO

A partir del primer algoritmo desarrollado con base en la metaheurística, se han propuesto diversos algoritmos para problemas NP-*Hard*⁵² que imitan el comportamiento de las hormigas con algunas modificaciones. Entre estos se encuentran:

- *Sistema de Hormigas Max-Min (SHMM, Max-Min Ant System)*
- *SH con ordenación (Rank-Based Ant System)*
- *Sistema de la Mejor-Peor Hormiga (SMPH o Best-Worst Ant System)*

⁵² Problemas cuyo número de decisiones factibles y esfuerzo computacional crece exponencialmente a medida que aumentan las variables

D. PROTOCOLO PARA LA DEFINICIÓN DEL ESPACIO DE TRABAJO DEL ROBOT MITSUBISHI MELFA IV EN LAS TRAYECTORIA DE CARGA Y DESCARGA DEL TORNO CNC

Con el fin de realizar una medición precisa del espacio de trabajo del robot, se desarrollarán diferentes pruebas que permitirán conocer su configuración. No se trabajará un espacio volumétrico, sino que se trabajará en un espacio 'dos y medio', en el cual se manejan planos de dos dimensiones (X,Y) a diferentes alturas determinadas en el eje Z que facilitan el desarrollo de la tarea.

OBJETIVO

Determinar las trayectorias en el plano (X,Y), y los valores de Z que hacen factible el movimiento del robot desde la estación hasta el torno CNC.

NÚMERO DE TRAYECTORIAS

Dado que la similitud de las trayectorias de carga y descarga del torno, se manejará en el experimento únicamente una de ellas.

VARIABLES A DETERMINAR

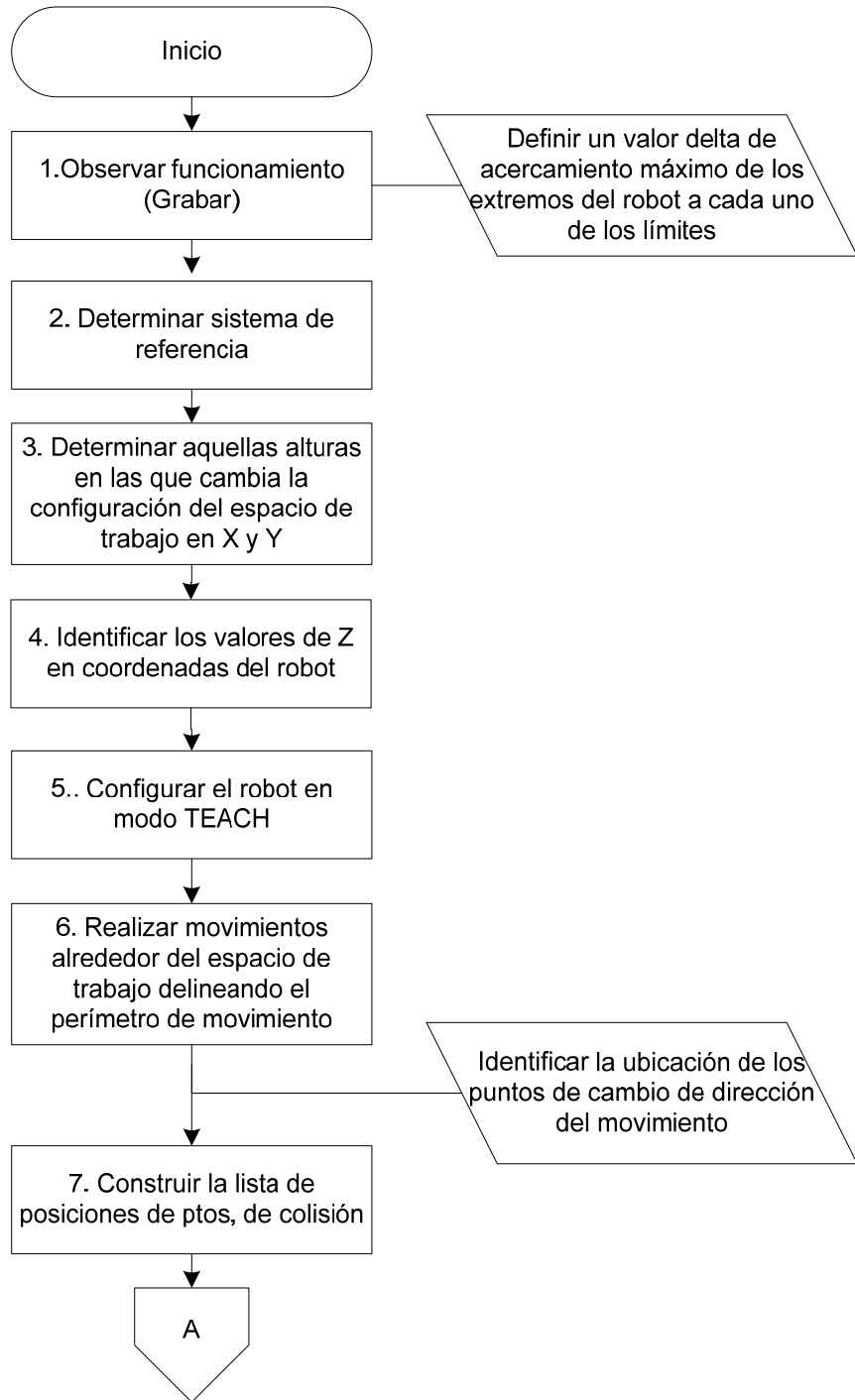
1. Número de planos en Z a realizar
2. Puntos de colisión para el robot a lo largo de la trayectoria para las diferentes alturas.
3. Zonas (áreas en x-y) en las cuales es conveniente aumentar o reducir la velocidad del brazo.
4. Velocidad de Recorrido y Velocidad de Acercamiento
 - Factores de seguridad
 - Espacio 3d
 - Obstáculos
 - Velocidades (mínimo 2: de trabajo y recorrido)

MATERIALES

- Cronómetro
- Cinta métrica
- Calibrador
- Lápiz y borrador
- Libreta
- Cámara de video⁵³

⁵³ Se realizan videos con el fin de posibilitar análisis posteriores.

METODOLOGÍA



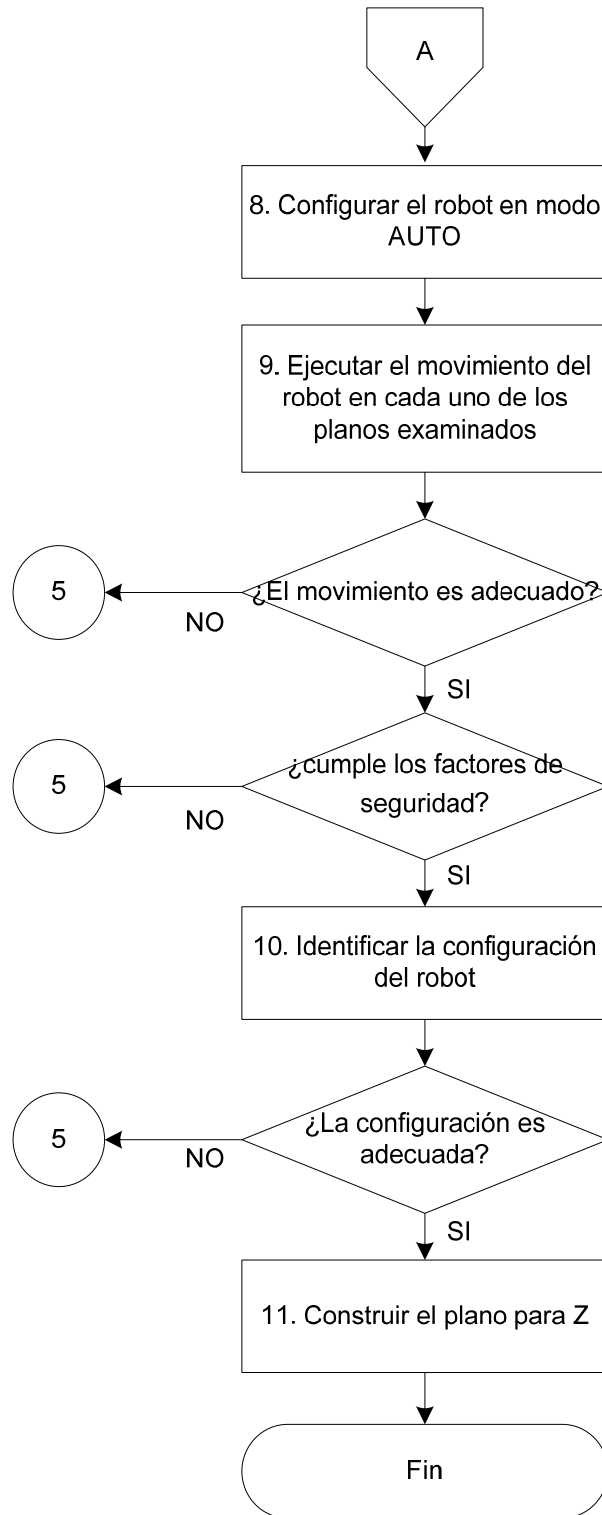


DIAGRAMA 10. Metodología proceso medición espacio de trabajo. Fuente: Los Autores

HOJA DE RUTA

Para cada punto en cada Z

1. Llevar el robot a la posición deseada recogido
2. Grabar la posición en el Teach Box
3. Grabar la posición en Cosimir
4. Tomar Foto
5. Anotar en plano
6. Anotar descripción
7. Alejar el robot en J7
8. Con Y y Z constante. Estirar el robot moviéndolo en X
9. Calcular el L (J7) final y llevarlo a esa posición
10. Grabar la posición en el Teach Box
11. Grabar la posición en Cosimir
12. Tomar Foto
13. Anotar en plano
14. Anotar descripción
15. Repetir de 7 a 14 para el alcance de X en sentido contrario.

ANOTACIONES

1. Observar el funcionamiento actual del robot:
 - 1.1. Hacer una grabación del movimiento del robot en las trayectorias de carga y descarga en diferentes vistas (Top, Front, Right).
2. Determinar si el movimiento es adecuado y cumple con los factores de seguridad establecidos.
 - 2.1. Identificar cómo está ecualizado (configurado) el robot en cada movimiento

E. PROBLEMAS ENCONTRADOS EN LA EJECUCIÓN DEL PROCEDIMIENTO

A continuación se presentan diferentes situaciones e inconvenientes encontrados durante la fase de toma de datos, así como las soluciones propuestas para la solución de los mismos:

- Al iniciar la toma de datos, el primer paso fue determinar niveles tentativos de Z. En una segunda instancia se identificó que de la orientación de la pinza depende en gran medida la forma en que se miden los puntos y las alturas, debido a que el 'TCP' (el punto donde el sistema identifica las coordenadas cartesianas del efector), no se encuentra ubicado en la punta de la pinza, sino en el punto central donde se conecta la pinza con el manipulador (ver Figura 41). Para solucionar este inconveniente, se decidió en un primer momento calcular las posiciones bajo las mismas condiciones de orientación de la pinza.

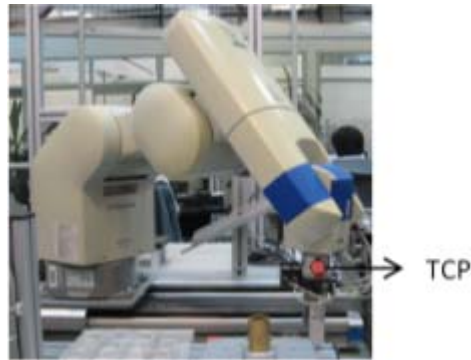


FIGURA 41. UBICACIÓN DEL TCP

Se escogió la orientación de la pinza en sentido vertical, como se aprecia en la Figura 41, teniendo en cuenta que el manipulador tiene la capacidad de mover únicamente la herramienta manteniendo una misma configuración para el resto de las articulaciones, lo cual genera que la altura en la que se encuentra la punta de la pinza cambie.

Por otra parte, durante las pruebas se determinó que también es necesario realizar la medición de los obstáculos manteniendo una orientación horizontal de la herramienta, ya que en este caso, aún cuando la altura del TCP no sea diferente, los parámetros X y Y son diferentes para cada obstáculo medido anteriormente con la orientación vertical.

- Debido a que el manipulador sobre el que se está realizando el estudio se encuentra instalado sobre una base móvil, se sabe que la ubicación espacial de la herramienta con respecto a un origen fijo puede ser alcanzada por una serie de configuraciones diferentes del manipulador. La base móvil es una banda transportadora o riel, sobre la que se desplaza el robot en el sentido del eje X, por lo cual se establece

un sistema de coordenadas global, localizado en un extremo fijo de la banda y otro sistema de coordenadas dependiente localizado en la base del robot (Figura 42).

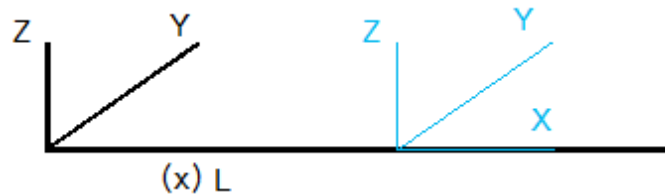


FIGURA 42. SISTEMAS DE REFERENCIA. FUENTE: LOS AUTORES

***Nota:** En adelante, el eje coordenado X del sistema local será denominado L, de forma que no se confunda con el eje X del sistema localizado en la base del manipulador.

Para hallar dicha serie de configuraciones, se realizó el siguiente procedimiento:

1. Tomar la posición del obstáculo desde la configuración más 'recogida' del manipulador, esto es, buscando que los parámetros X y Y sean lo más próximos a cero.
 2. Transformar las coordenadas del efector final al sistema de coordenadas global.
 3. Buscar la extensión máxima del manipulador modificando únicamente la coordenada X.
 4. Teniendo en cuenta la longitud hallada, calcular la localización de la base del robot, de forma que el efector final regrese a la ubicación inicial sin modificar la configuración alcanzada en el paso 3.
 5. Repetir los pasos 3 y 4, buscando la extensión máxima en el sentido opuesto de X.
- Durante la medición de los puntos de colisión, es de vital importancia considerar el manipulador en su totalidad, no únicamente el efector final. Cualquiera de las articulaciones, la base, e incluso los cables y mangueras tienen posibilidad de colisionar con obstáculos, afectando posiblemente el manipulador u otros componentes de la celda de trabajo. En el caso de los cables y mangueras conectados al efector final, se recomienda variar la orientación de la herramienta en el sentido paralelo al eje Z, ya que la extensión horizontal de los cables hacia fuera del manipulador varía.
 - Los niveles de Z medidos, fueron escogidos de acuerdo a los siguientes criterios:
 1. El manipulador debe comenzar o finalizar la trayectoria con una configuración y orientación determinada, por lo cual a la altura inicial y final se debe identificar los obstáculos del espacio de trabajo.
 2. Las alturas en las que se presentan cambios en los obstáculos que posiblemente se encuentren en las zonas por donde se presume un camino óptimo (para esto, se considera en principio una línea recta que conecta el punto de carga y descarga independientemente de su viabilidad).
 3. Alturas intermedias entre los niveles considerados de acuerdo a los criterios 1 y 2.

F. IMÁGENES DEL PROCESO DE DEFINICIÓN DEL ESPACIO DE TRABAJO DEL ROBOT

A continuación se presenta el recuento de las imágenes y coordenadas de los puntos tomados en la definición preliminar de los obstáculos para una de alturas del robot con orientación vertical. Para un mejor entendimiento, es preciso recordar la forma en que se tomaron los puntos (ver. Anexo D).

Las imágenes e información de las demás alturas y orientaciones del estudio preliminar en espacio de 2 ½ se encuentran en medio digital, no son incluidas en el documento debido a que como se menciona en el capítulo 9.1 el estudio preliminar en 2 ½ fue descartado, concluyendo en la definición de una altura única con orientaciones definidas.

Las imágenes a continuación muestran un ejemplo de los puntos en que el robot está próximo a colisionar con un obstáculo, para una altura Z determinada:

Nivel Z1 Altura Z= 196.8

Orientación: *vertical*

A = 180

B= 0

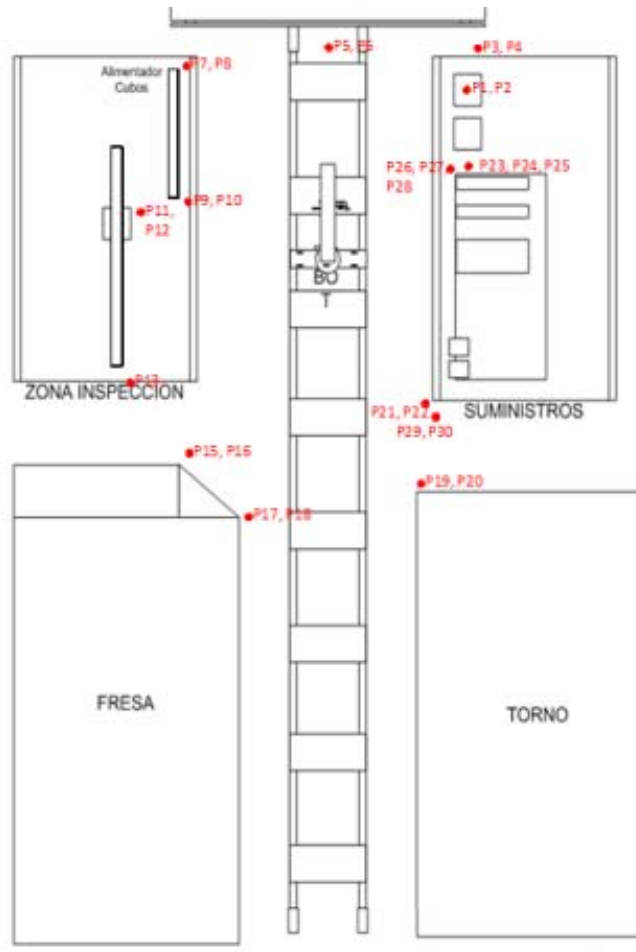
C = (Valor variable de acuerdo al obstáculo)





Se observa en las imágenes cómo se tomaron puntos en diferentes orientaciones de la pinza para cada obstáculo.

Distribución de los puntos definidos en la celda de manufactura:



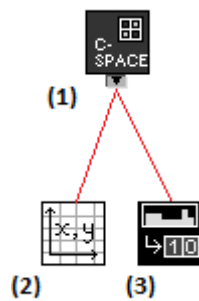
Especificación de los puntos definidos:

	X	Y	Z	A	B	C	L
P1	307.9	-531.2	197.0	-178	0	-1	1695.7
P2	4.7	-531.2	197.0	-178	0	-1	1997.9
P3	311.1	-529.1	197.1	-178	0	-1	1817.3
P4	33.8	-529.0	196.9	-178	0	-1	2094.6
P5	386.3	-12.3	196.8	-180	-2	179	1726.8
P6	613.2	-12.4	196.9	-180	-2	179	1499.8
P7	361.1	137.5	196.8	-180	-2	-158	1726.8
P8	597.9	137.5	196.7	178	-1	-94	1489.9
P9	362.0	137.4	197.1	178	-1	-91	1414.5
P10	597.6	137.4	196.9	178	-1	-91	1178.7
P11	-21.6	367.8	196.8	180	-1	1	1697.7
P12	494.6	367.8	196.7	-179	-0	-180	1181.7
P13	41.3	467.3	197.1	-179	0	-178	1102.9
P14	401.9	467.3	196.9	-179	0	-180	742.2
P15	-227.7	455.1	197.6	180	-0	0	882.9
P16	-416.6	455.0	198.0	180	-0	0	1071.4
P17	-268.7	313.4	197.1	180	-0	-0	694.7
P18	-531.2	313.4	197.0	180	-0	-0	957.3
P19	-38.2	-402.6	197.0	-180	-0	90	666.2
P20	-467.5	-402.6	196.9	-180	-0	90	1095.6
P21	266.6	-303.8	196.9	-180	-0	138	700.1
P22	536.7	-303.9	196.9	-180	-0	138	429.9
P23	4.7	-531.1	196.8	-178	0	-1	1791.9
P24	-264.0	-531.2	196.8	-178	0	-1	2060.6
P25	307.9	-531.1	196.8	-178	0	-1	1488.7
P26	-158.4	-362.1	196.8	-179	-1	0	1825.4
P27	-393.7	-362.2	196.7	-179	-1	0	2060.6
P28	498.7	-362.0	197.1	179	0	89	1168.2
P29	-266.5	-303.7	196.3	180	-0	65	1233.3
P30	-536.6	-303.8	196.2	180	-0	65	1503.3

G. JERARQUÍA DE VI'S PARA LA DEFINICIÓN DEL ESPACIO DE TRABAJO Y BÚSQUEDA DE LA SOLUCIÓN INICIAL

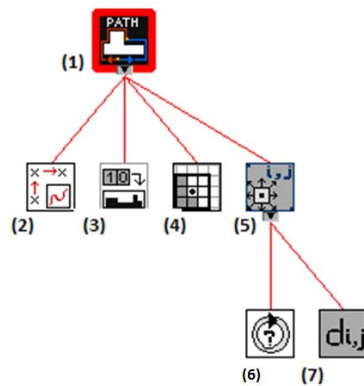
A continuación se presentan los diagramas de jerarquía que ilustran la forma en que están contruidos el algoritmo para la definición del espacio de configuración y el algoritmo de búsqueda de la solución inicial en *LabVIEW*⁵⁴, donde el proceso número (1) corresponde al programa (VI) principal. Igualmente se muestra la interfaz (*Front Panel*) diseñada. El funcionamiento de los algoritmos se explica en el los capítulos 9.1.2 y 9.2.

- Definición del espacio de configuración:



- (1) Definición del espacio (c-space)
- (2) Búsqueda de coordenadas centro
- (3) *Image to array*

- Búsqueda de la solución inicial:



- (1) Búsqueda solución inicial
- (2) Marcar casillas visitadas
- (3) *Image to array*
- (4) Creación del 'Vecindario'
- (5) Navegación
- (6) Selector dirección de navegación
- (7) Regla de navegación

⁵⁴ Recuerde la descripción del entorno de programación en el capítulo 8.1

H. PROTOCOLO PARA LA IMPLEMENTACIÓN Y CALIFICACIÓN DE LAS TRAYECTORIAS GENERADAS POR MEDIO DEL ALGORITMO ACS

El presente protocolo se realiza con el fin de establecer una metodología clara para implementación y evaluación de las distintas trayectorias en los diferentes espacios de configuración establecidos. En el capítulo 9.10 se definieron los parámetros de comparación y a continuación se ilustrará el procedimiento a seguir para este fin.

OBJETIVOS

- Programar en el robot Melfa IV las trayectorias encontradas
- Analizar los resultados de la implementación

PARÁMETROS DE COMPARACIÓN

A continuación se recuerdan las variables que caracterizan el desempeño de una trayectoria:

- Tiempo
- Distancia
- Suavidad de la trayectoria
- Torque

CONSIDERACIONES PARA EL PROCESO DE IMPLEMENTACIÓN DEL ALGORITMO

- Dado que se trabajo para espacios con altura constante, es necesario que se garantice por medio de la programación del robot que este no modifique la altura durante el recorrido, para lo cual se utiliza el comando MVS.
- Al hacer uso del comando MOV debe tenerse especial cuidado, teniendo en cuenta que es posible que durante la trayectoria entre puntos vía la altura u orientación de la herramienta se modifique en algún momento. (Ver tipos de movimiento del robot en el Capítulo 5.5.1 y descripción de comandos en el capítulo 5.5.5)
- La definición de la velocidad de cada una de las articulaciones del robot depende de la precisión que se requiera para el movimiento y el riesgo de colisión, para hacer posible la comparación entre las trayectorias, es preciso definir un valor constante para la velocidad del robot.
Teniendo en cuenta que se requiere un valor bajo de velocidad para la observación del comportamiento del robot en cada trayectoria, y un valor alto para conseguir tiempos eficientes y obtener una mejor evaluación de los parámetros, se definen los siguientes valores:
 - $V_1 = 30\%$, $V_2 = 100\%$

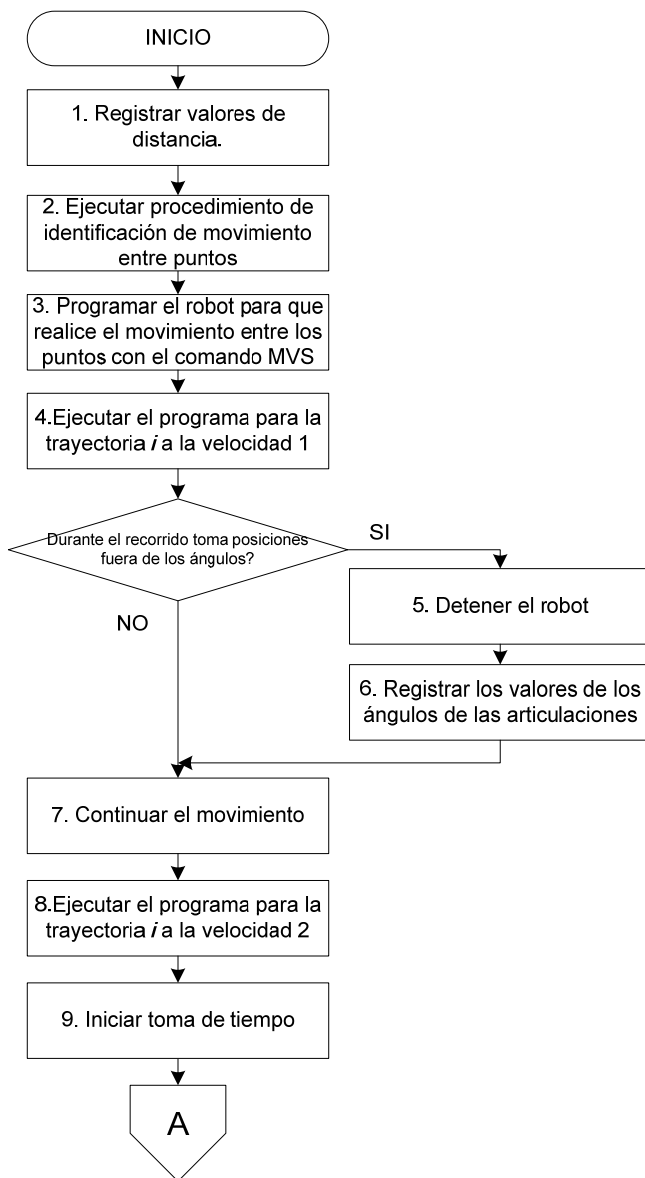
MATERIALES

- Cronómetro
- Computador (Formato de Evaluación)
- Cámara de video⁵⁵

⁵⁵ Se realizan videos con el fin de posibilitar análisis posteriores.

METODOLOGÍA

A continuación se presenta la metodología utilizada para la implementación:



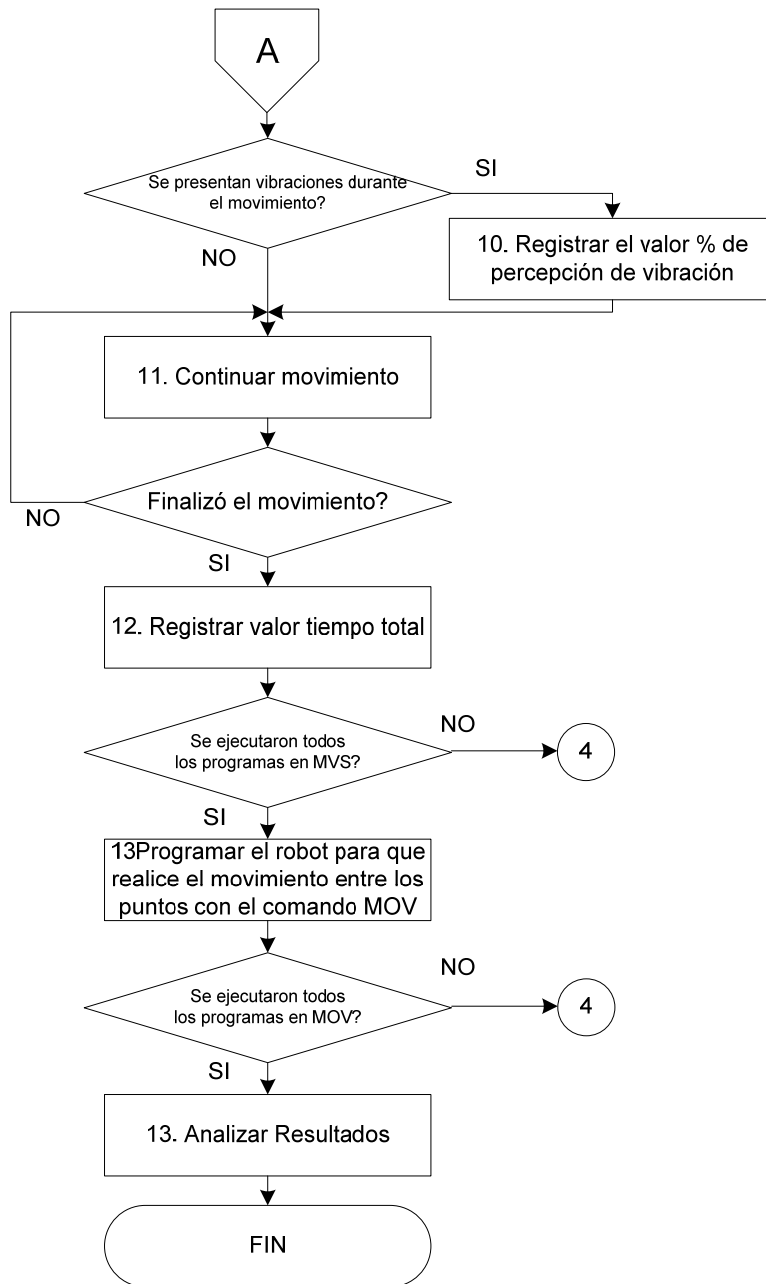


DIAGRAMA 11. Metodología proceso implementación de las soluciones en el Robot Melfa RV 2A. Fuente: Los Autores

El proceso de definición del tipo movimiento entre puntos es fundamental para la programación del robot:

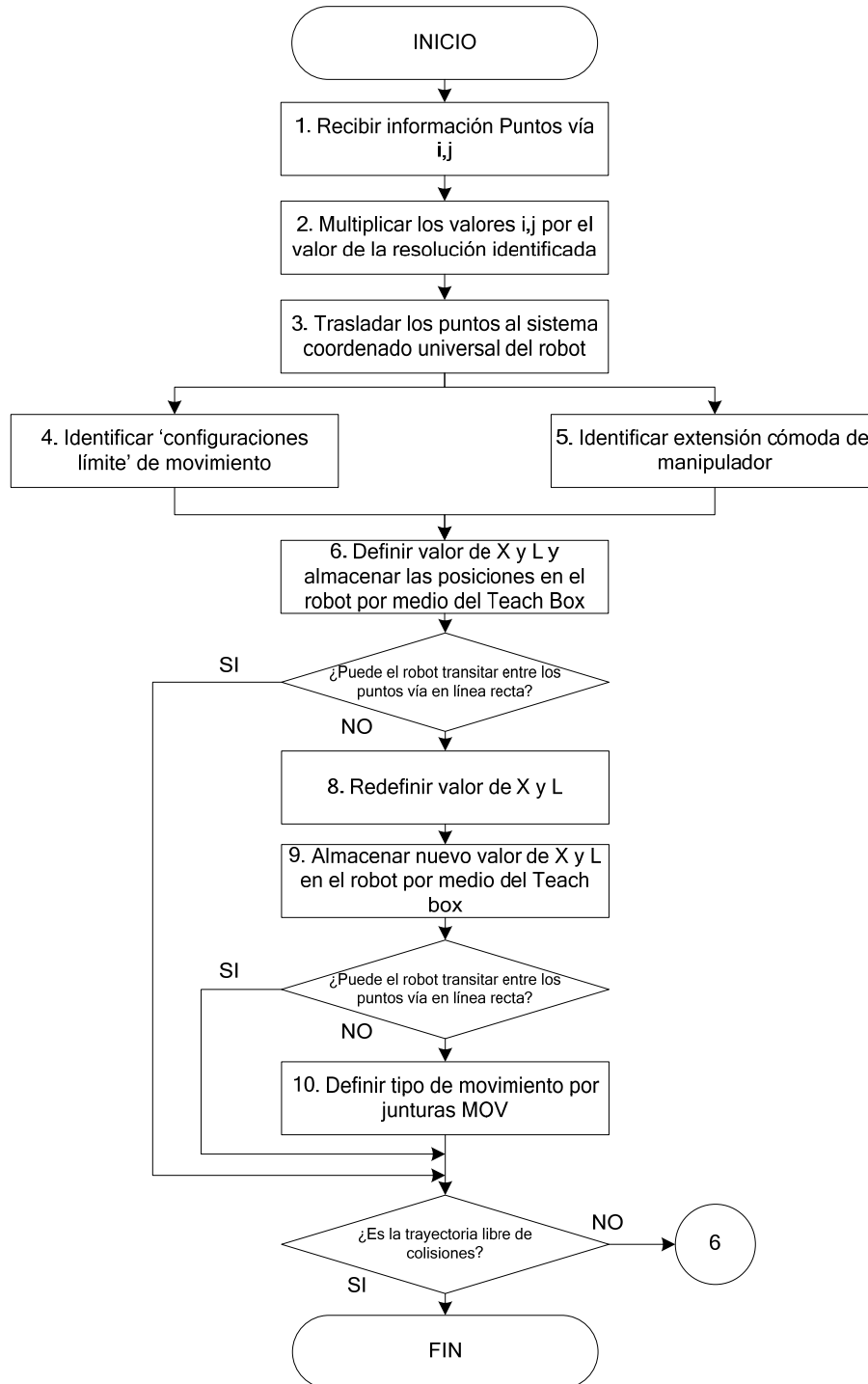


DIAGRAMA 12. PROCESO DEFINICIÓN MOVIMIENTO ENTRE PUNTOS VÍA. Fuente: Los Autores

I. DESCRIPCIÓN DE LAS PRUEBAS EXPERIMENTALES PARA EL AJUSTE DE LOS PARÁMETROS DEL ALGORITMO ACS

La definición de los parámetros se llevó a cabo en dos fases; la primera se realizó con el fin de establecer el número de colonias (m) adecuado que debían utilizarse, y la segunda, para ajustar el parámetro (β) y la resolución (r) de la imagen para la ejecución del algoritmo en los espacios representados.

1ra Fase: Ejecución del algoritmo para la definición del número de colonias por réplica:

Se ejecutó el algoritmo ajustando los parámetros con los valores recomendados por Dorigo [21], (capítulo 9.3):

Número de Hormigas	10
β	2
q	0.85
Número de réplicas	30

Se utilizaron 160 colonias con el fin observar el comportamiento del algoritmo y se determinó el número de veces que la solución mejoraba en cada rango para las diferentes réplicas, así como el 'delta' en el que mejoraba la solución para cada caso (Ver Tabla 13)

TABLA 13. RESULTADOS DEFINICIÓN NÚMERO COLONIAS

Intervalos de colonias	No. Veces Que mejora la solución	No. Veces Promedio	Promedio Delta variación	Varianza	Desviación Estándar
0-19	99	3.30			
20-39	28	0.93	1.826	3.166	1.779
40-59	9	0.30	0.404	0.793	0.891
60-79	9	0.30	0.415	0.789	0.888
80-99	3	0.10	0.098	0.121	0.348
100-119	4	0.13	0.246	0.587	0.766
120-139	3	0.10	0.193	0.772	0.879
140-159	5	0.17			

En la tabla anterior, se observa que las colonias mejoran las soluciones considerablemente antes de la colonia 80, después de este número de colonias, el promedio de variación se reduce drásticamente. De la misma forma, el delta de variación (el promedio de la mejora entre soluciones), disminuye significativamente para el mismo rango, lo cual lleva a la conclusión que no vale la pena ejecutar el algoritmo con más de **80 colonias**, puesto que no se mejoran las soluciones con más frecuencia, ni en mayor medida.

A continuación se presenta la imagen de la interfaz del algoritmo modificado para la realización de las pruebas:

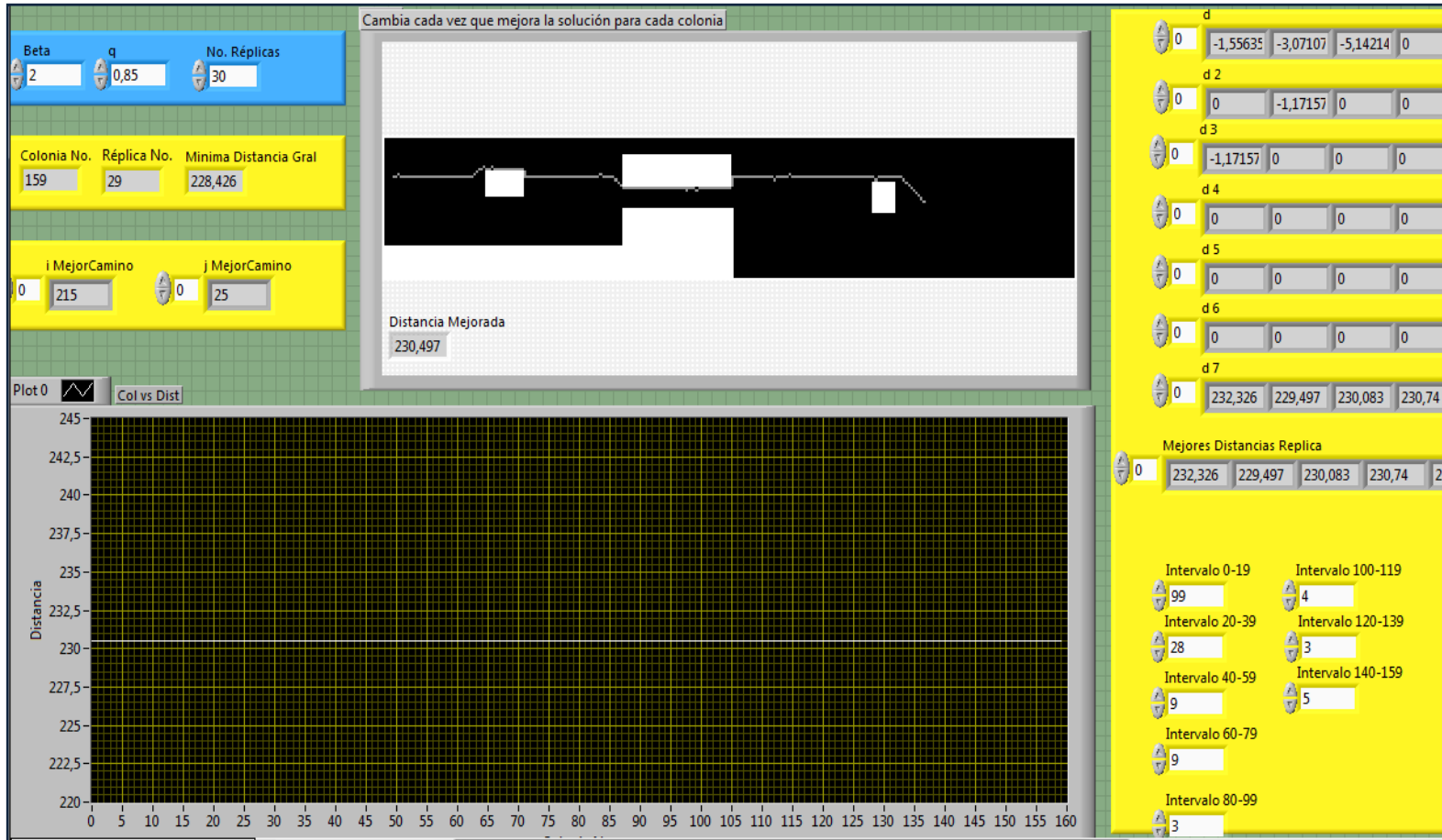


FIGURA 43. INTERFAZ PRUEBA PARA LA DEFINICIÓN DE NÚMERO DE COLONIAS

2ª Fase: Premuestreo para el ajuste de los parámetros β y resolución:

Una vez seleccionado el número de colonias a utilizar para la ejecución del algoritmo, se procedió a realizar un pre-muestreo con el fin de determinar el valor de β que mejores resultados permitiera, así como la mejor resolución de la imagen a trabajar.

El experimento se llevó a cabo con los siguientes parámetros:

n	10
m	80
q	0,85
Número de réplicas	30

r	5
	11
	21

β	0.5
	1
	2

De acuerdo con lo señalado anteriormente, los parámetros restantes se ajustaron por referencia bibliográfica (ver Tabla 3).

En total se ejecutó el algoritmo $(9 \times 30) = 270$ veces. Los resultados se presentan resumidos en la siguiente tabla:

TABLA 14. RESULTADOS PREMUESTREO PARA EL AJUSTE DE PARÁMETROS

Combinación de parámetros (r,B)	Mínimo	Promedio	Intervalo de confianza		varianza	Desv. Estándar	Rango	Diferencia con el mínimo	Tiempo Ejecución (min)	Seg/Colonia
			Límite inf	Límite sup						
5,0.5	2562.905	2605.591	2572.88	2638.30	278.449	16.687	70.415	54.329	475	11.875
5,1	2554.830	2587.094	2556.76	2617.43	239.546	15.477	63.845	46.254	481	12.025
5,2	2546.040	2566.612	2545.38	2587.85	117.380	10.834	37.785	37.464	476	11.900
11,0.5	2552.913	2585.879	2551.52	2620.24	307.262	17.529	70.235	44.337	35	0.875
11,1	2517.251	2570.083	2528.38	2611.79	452.717	21.277	87.670	8.675	30	0.750
11,2	2527.140	2551.761	2524.94	2578.58	187.198	13.682	58.454	18.564	32	0.800
21,0.5	2520.882	2585.676	2538.35	2633.00	583.040	24.146	116.676	12.306	15	0.375
21,1	2538.270	2573.869	2539.29	2608.45	311.266	17.643	68.103	29.694	16	0.400
21,2	2508.576	2555.616	2519.12	2592.11	346.766	18.622	80.388	0.000	15	0.375
MIN	2508.576	2551.761			117.380	10.834	37.785		15	0.375

Nota: Los colores del cuadro representan una escala para cada columna, donde el color más claro es el valor más bajo y el color más oscuro, el valor más alto.

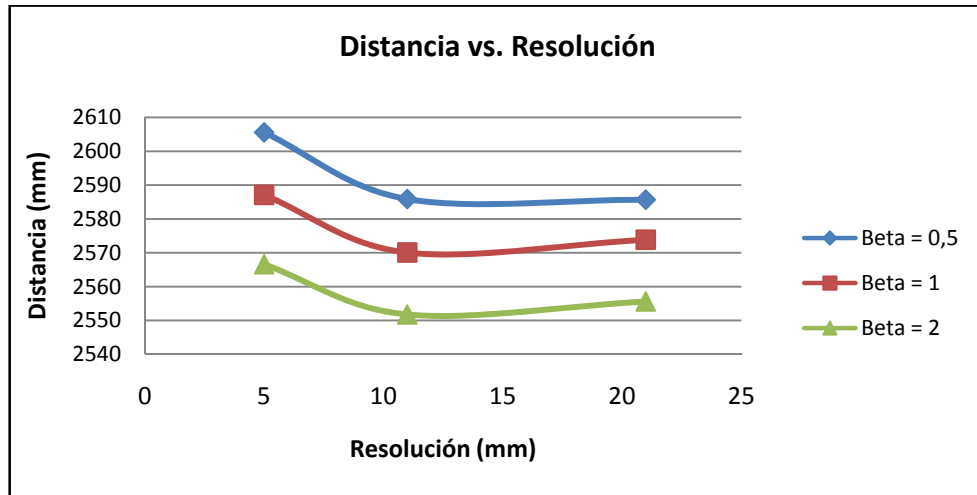


FIGURA 44. GRÁFICA DISTANCIA VS. RESOLUCIÓN. Fuente: Los Autores

En los datos presentados se observa que la combinación de parámetros (21,2) presenta el resultado más bajo en términos de distancia, por lo cual se podría concluir que es la mejor combinación. Sin embargo, al analizar los demás indicadores, se aprecia que presenta una desviación estándar alta, con lo cual los resultados se encuentran en un intervalo amplio, considerando además que con esta resolución (21) se pierde mayor cantidad de información en la gráfica que con las demás.

Por otra parte, la combinación de parámetros (11,2) presenta buenos resultados en términos de varianza y promedio, aún cuando el valor mínimo encontrado en el pre-muestreo es aproximadamente nueve (9) unidades superior al mejor. Por esta razón, y teniendo en cuenta que los tiempos de ejecución son razonablemente bajos, se decidió que esta es la mejor combinación de parámetros para trabajar este problema específico.

NOTA: Los experimentos se realizaron en un espacio de trabajo diferente al real con el fin de evitar el sesgo de los resultados hacia alguno de los espacios seleccionados. Se recomienda para futuros trabajos que se evalúen y seleccionen parámetros para cada caso.

J. PROGRAMACIÓN DEL ROBOT MELFA RV- 2A EN LENGUAJE MELFA IV (COSIMIR® PROFESSIONAL)

Recordando el capítulo 5.5, en el cual se exponen los tipos de trayectorias que puede seguir un manipulador, así como los comandos que usa el Lenguaje Melfa IV.

Inicialmente se realizaron las pruebas programando los movimientos entre puntos con los comandos MVS y MOV, más adelante se incorporó el comando CNT (Movimiento continuo). Se presenta a continuación el código modelo que se utilizó para cada una de las trayectorias con las que se trabajó:

- **Iniciación motores y definición de sensores:**

```
10 SERVO ON
20 DEF IO P1AV = BIT, 8
20 DEF IO GROPEN = BIT, 900
30 DEF IO GRCLOSE = BIT, 901
```

- **Definición de velocidades:**

Aquí se definen las velocidades para el movimiento entre juntas, en X,Y,Z y modo 'Joint'.

```
40 ACCEL 70,70
50 JOVRD 30
60 SPD 30
```

- **Definición del movimiento entre puntos**

En esta parte del pseudocódigo se mueve el robot entre los puntos vía definidos por el algoritmo.

```
70 MOV PHOME
80 HOPEN 1
50 WAIT GROPEN = 1
55 CNT 1, 11, 11
60   MOV P1
70   MVS P2
80   MVS P3
90   MVS P4
100  MVS P5
110  MVS P6
120  MVS P7
145 CNT 0
150  MOV PWAIT
160 END
```

NOTA: Este pseudocódigo mueve el robot en los ejes cartesianos desde la posición P1 hasta PWAIT, para la segunda parte de la implementación se reemplazan todos los comandos MVS por MOV.