



UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ

FACULTAD DE INGENIERIA

CENTRO DE INVESTIGACIÓN Y ESTUDIOS DE POSGRADO

**"Control de posición de un robot industrial
sobre superficies arbitrarias utilizando
manipulación de espacio de cámara"**

Tesis
que para obtener el grado de
Maestro en Ingeniería Eléctrica
opción
Control Automático

presenta
RUBÉN DARÍO GONZÁLEZ LIZCANO

Asesor:
Dr. Emilio J. González Galván

SAN LUIS POTOSI, S.L.P.

OCTUBRE, 2000





Julio 03, 2000.

**AL ING. RUBÉN DARÍO GONZÁLEZ LIZCANO
P R E S E N T E . -**

En atención a su solicitud de Tema y Temario, presentada por el Dr. Emilio Jorge González Galván Asesor de la Tesis que desarrollará Usted, con el objeto de obtener el Grado de Maestría en *Ingeniería Eléctrica*. Me es grato comunicarle que en la Sesión de Consejo Técnico Consultivo celebrada el día 29 de junio del presente año, fue aprobado el Tema y Temario propuesto:

TEMARIO:

**"CONTROL DE POSICIÓN DE UN ROBOT INDUSTRIAL SOBRE
SUPERFICIES ARBITRARIAS UTILIZANDO MANIPULACIÓN DE ESPACIO
DE CÁMARA."**

- I.- INTRODUCCIÓN.
 - II.- DESARROLLO DE LA METODOLOGÍA DE CONTROL.
 - III.- RESULTADOS EXPERIMENTALES.
 - IV.- CONCLUSIONES Y TRABAJO FUTURO.
- APÉNDICES
BIBLIOGRAFÍA

"MODOS ET CUNCTARUM RERUM MENSURAS AC DE BO"



UNIVERSIDAD AUTÓNOMA
DE SAN LUIS POTOSÍ
FACULTAD DE INGENIERÍA
DIRECCIÓN

**ING. JOSÉ ÁRNOLDO GONZÁLEZ ORTÍZ
DIRECTOR**

**SMH



Universidad Autónoma de San Luis Potosí

Facultad de Ingeniería

Centro de Investigación y Estudios de Posgrado

Maestría en Ingeniería Eléctrica
Opción Control Automático.

**Control de posición de un robot industrial
sobre superficies arbitrarias utilizando
manipulación de espacio de cámara**

Presenta:

Ing. Rubén Darío González Lizcano

Sinodales:

Dr. Emilio J. González Galván
(Asesor de tesis)

Dr. Alejandro R. Femat Flores

Dr. Felipe Bazos Flores

*Para Darío y Elsa,
de quienes tuve la fortuna
de haber nacido hijo.*

Agradecimientos

Desco expresar un profundo agradecimiento a mis padres, por la libertad que me han dado de ser, y a mis hermanos Karlo y Elsi: gracias por formar entre todos una verdadera familia.

Al Dr. Emilio González, por su asesoría y sobre todo por la confianza depositada en mí y en mi capacidad de llevar a buen fin este proyecto.

A mis compañeros de laboratorio y de jornada: Liz, Oscar y Aída, y por supuesto a mis colegas y paisanos huastecos, a todos ellos gracias por ser también mis amigos.

Debo agradecer también al CONACyT por el apoyo económico brindado durante los últimos dos años y a los profesores del CIEP de los cuales en algún momento requerí su ayuda.

A Dios lo menciono al final y no por ello es menos importante. Él fue quien cruzó todas estas personas en mi camino y me dio la determinación para alcanzar todos mis objetivos. Es por Él que ahora puedo estar tan agradecido.

Índice General

Agradecimientos	i
Lista de símbolos	v
Resumen	ix
1 Introducción	1
1.1 El laboratorio de robótica	1
1.2 Objetivo y resultados esperados	2
1.3 Robótica y visión artificial	3
1.3.1 Visión artificial	3
1.3.2 Sistemas de visión	4
1.3.3 Análisis y procesamiento de una imagen	5
1.3.4 Aplicaciones de robótica	11
1.4 Manipulación de espacio de cámara	12
1.4.1 Justificación	12
1.4.2 Descripción general del método	13
1.4.3 Modelo de cámara	15
1.4.4 Modelo ortográfico de cámara	16
1.4.5 Estimación de los parámetros de visión	18
1.4.6 Flattening	21
1.4.7 Estimación de la configuración interna	26
1.5 Modelo cinemático del manipulador	27
1.5.1 Emulación de un paralelogramo	29
1.5.2 Matriz de transformación homogénea E	32
2 Desarrollo de la metodología de control	35
2.1 Análisis de la imagen	35

2.2	Estimación de los parámetros de visión	36
2.2.1	Determinación de la posición de las marcas luminosas	38
2.2.2	Todas las marcas luminosas encendidas	40
2.3	Estimación de la configuración interna	45
2.3.1	Interpretación geométrica de \mathbf{M}_T	51
2.4	Relación entre los puntos objetivo	52
2.4.1	Identificación de los puntos objetivo	55
2.5	Integración del sistema	56
3	Resultados experimentales	61
3.1	Parámetros de visión y trayectoria preplaneada	61
3.2	Primera aproximación	63
3.2.1	Segunda aproximación	65
3.2.2	Error en el posicionamiento de la herramienta	66
3.2.3	Mejora de la maniobra	67
4	Conclusiones y trabajo futuro	69
4.1	Aplicaciones	71
4.2	Trabajo futuro	72
A	Programación Orientada a Objetos	75
A.1	Breve introducción a la teoría de objetos	75
A.1.1	Interfaces	77
A.1.2	La implementación oculta	78
A.1.3	Reutilización de la implementación	79
A.1.4	Herencia: Reutilización de la interface	79
A.2	Aplicación de la POO en el proyecto	81
A.2.1	Objetivo general del sistema	81
A.2.2	Diseño de clases	82
A.2.3	La clase CFanuc	83
A.2.4	La clase CCamara	88
A.2.5	La clase CConfig	90
A.3	Aplicación: La interfaz gráfica	92
B	Esquemas y diagramas	95
C	Fotografías	99
	Índice de Figuras	103
	Índice de Tablas	105
	Bibliografía	107

Lista de símbolos

(x_{ci}, y_{ci})	Coordenadas en el plano de la imagen obtenida de la i -ésima cámara
e_1, \dots, e_4	Números de Euler o cuaterniones
f	Distancia focal
$\mathbf{C}_i \in \mathbf{R}^6$	Vector de parámetros de visión para la i -ésima cámara
\mathbf{C}_c	Vector de parámetros de visión actuales
$\Theta \in \mathbf{R}^6$	Vector de ángulos del manipulador
$J(\mathbf{C})$	Función escalar de minimización del vector \mathbf{C}
m	Número de muestras en un proceso de minimización
W_i	Ponderación o peso relativo asignado a la i -ésima muestra
$\Delta Y_p, \Delta Y_c \in \mathbf{R}^m$	Residuales actuales (c) y residuales que se predicen (p) después de una corrección $\Delta \mathbf{C}$
ΔY_f	Residuales generados durante el procedimiento de <i>flattening</i>
$\Delta \mathbf{C} \in \mathbf{R}^6$	Correcciones diferenciales al vector \mathbf{C}
$\Delta \mathbf{C}_f$	Correcciones diferenciales al vector \mathbf{C} usando <i>flattening</i>
$\mathbf{b} \in \mathbf{R}^8$	Parámetros de visión representados en forma de coeficientes
$\mathbf{W} \in \mathbf{R}^{m \times m}$	Matriz de ponderaciones
$\mathbf{A} \in \mathbf{R}^{2m \times 6}$	Matriz de derivadas parciales en los parámetros de visión
Z_0	Distancia a lo largo del eje óptico de la cámara hasta la base del robot. En la deducción de los parámetros de visión se considera $Z_0 \rightarrow \infty$
(x_r, y_r, z_r)	Coordenadas de un punto de referencia durante el procedimiento de <i>flattening</i>
Z_r	Distancia a lo largo del eje óptico de la cámara hasta un punto de referencia

$\gamma(\Theta)$	Función escalar de minimización del vector Θ
(x_{f_i}, y_{f_i})	Coordenadas en el plano de la imagen de la i -ésima cámara modificadas mediante <i>flattening</i>
n_c	Número de cámaras
$n_i(i)$	Número de puntos objetivo en la i -ésima cámara
$(x_{t_j}^i, y_{t_j}^i)$	Coordenadas del j -ésimo punto objetivo en el plano de la imagen de la i -ésima cámara
θ_i	Ángulo asociado a la i -ésima junta del manipulador
a_i, d_i, α_i	Parámetros de Denavit-Hartenberg para el i -ésimo eslabón
$Trans_{a,l}$	Traslación de l unidades de longitud a lo largo del eje a
$Rot_{a,r}$	Rotación de r unidades alrededor del eje a
$\mathbf{A}_i \in \mathbb{R}^{4 \times 4}$	Transformación homogénea asociada al i -ésimo eslabón del manipulador
$\mathbf{E} \in \mathbb{R}^{4 \times 4}$	Transformación homogénea asociada a la herramienta instalada en el manipulador
$\mathbf{T}_K(\Theta) \in \mathbb{R}^{4 \times 4}$	Modelo cinemático del manipulador
$\mathbf{F} \in \mathbb{R}^4$	Coordenada homogénea de una marca luminosa en la herramienta
$\mathbf{V}_k \in \mathbb{R}^4, k = 1, \dots, 6$	Vector unitario asociado a la k -ésima cara de la herramienta
$\mathbf{V}'_k \in \mathbb{R}^4$	Vector \mathbf{V}_k con respecto al sistema coordenado en la base del robot
$\vec{\mathbf{EO}} \in \mathbb{R}^4$	Vector asociado al eje óptico de una cámara
$\mathbf{D}^i \in \mathbb{R}^{4 \times 4}$	Matriz de transformación homogénea asociado a la pieza no manipulable y relacionado con la i -ésima cámara
$\Gamma^i \in \mathbb{R}^{4 \times 4}$	Matriz de transformación homogénea que permite definir posiciones intermedias con respecto al sistema coordenado de la herramienta.
$\mathbf{T}^i \in \mathbb{R}^{4 \times 4}$	$\mathbf{T}_K(\Theta)\Gamma_i$
$\mathbf{M}_T^i \in \mathbb{R}^{4 \times 4}$	Matriz de momentos asociada a la i -ésima cámara
$\mathbf{L}^i \in \mathbb{R}^{4 \times 4}$	Matriz de diagonalización de \mathbf{M}_T^i
$\mathbf{M}_T^{d,i} \in \mathbb{R}^{4 \times 4}$	Matriz de momentos diagonalizada
$\mathbf{L}_1^i, \mathbf{L}_2^i \in \mathbb{R}^{4 \times 4}$	Matrices que intervienen en la obtención de \mathbf{L}^i
$\mathbf{M}_{cm} \in \mathbb{R}^{3 \times 3}$	Matriz de momentos centrales
$v_1, v_2, v_3 \in \mathbb{R}^3$	Eigenvectores de la matriz \mathbf{M}_{cm}
$\tilde{l}_x, \tilde{l}_y, \tilde{l}_z$	Centro de gravedad del conjunto de n_t puntos objetivo
$\lambda_1^i, \dots, \lambda_4^i$	Elementos de la diagonal principal de $\mathbf{M}_T^{d,i}$
$\mathbf{M}_T^{d,i} \in \mathbb{R}^{4 \times 4}$	Raíz cuadrada de $\mathbf{M}_T^{d,i}$
$\mathbf{R}_r^i, \mathbf{R}_p^i, \Delta \mathbf{Y}_M \in \mathbb{R}^8$	Vectores utilizados en la minimización de la configuración interna

$\mathbf{A}_\theta \in \mathbb{R}^{(n_c \times 6)}$	Matriz de derivadas parciales utilizada en la minimización de la configuración interna
$\Delta \Theta \in \mathbb{R}^6$	Correcciones diferenciales al vector Θ
a, \dots, f	Parámetros que definen el modelo lineal que relaciona las superficies observadas por dos cámaras

Resumen

Se presenta una nueva implementación del método de *manipulación de espacio de cámara* para la estimación de la configuración interna de un robot de seis grados de libertad. Esta configuración corresponde directamente al vector de ángulos relacionados con la posición, sobre una superficie plana, de una herramienta instalada en el extremo del manipulador. Esta superficie, también llamada objeto no manipulable, permanece estacionaria durante el transcurso de la maniobra y puede tener una posición y orientación arbitraria dentro del espacio tridimensional del manipulador. Debido a limitaciones en el modelo que caracteriza la superficie de trabajo, el posicionamiento del robot solo puede realizarse sobre superficies planas.

La posición que se desea alcanzar sobre la superficie, es seleccionada previamente desde una computadora personal (PC) a través de la imagen obtenida desde una de las cámaras de video que forman parte del sistema de visión instalado dentro del laboratorio de robótica.

Para caracterizar la superficie de trabajo, se ha hecho uso de un conjunto de puntos luminosos producidos por un haz de luz láser dividido para formar una matriz de aproximadamente 7×7 haces de luz. Esto evita la necesidad de colocar marcas artificiales que dejen una huella permanente, haciendo el método más versátil.

Se incluyen además los resultados experimentales que demuestran la precisión y margen de error de los algoritmos de estimación de la configuración interna del manipulador en el desempeño de una tarea de posicionamiento simple de un robot Fanuc® modelo ArcMate 100i R-J2® equipado con un sistema de soldadura por arco (MIG/TIG). La comunicación con el robot se establece a través de una interfaz Ethernet, y es utilizada para obtener el vector actual de los ángulos que conforman la configuración del manipulador y enviarle órdenes de movimiento correspondientes a las posiciones estimadas.

Todos los algoritmos involucrados han sido desarrollados bajo el modelo de programación orientada a objetos (específicamente C++). Se presenta además una interfaz gráfica desarrollada bajo el lenguaje de programación Visual C++ que facilita la aplicación del método de posicionamiento a nivel experimental.

Introducción

En este capítulo se presentan algunas de las bases teóricas en las que se soportan los principales argumentos de esta tesis, el objetivo y los resultados esperados de la misma. Se describen también las características del laboratorio de robótica dentro del cual fue desarrollado el proyecto.

Un breve resumen de las aplicaciones de la robótica en la industria muestra la necesidad de la utilización de la visión artificial como una herramienta en la integración y automatización de sistemas y procesos industriales.

Se describe además el desarrollo del método de manipulación de espacio de cámara (CSM)¹ como una aplicación directa de la visión artificial a la robótica, así como algunos antecedentes históricos del método.

Por último, una descripción general del robot Fanuc[®] ArcMate 100i R-J2[®] utilizado, así como el desarrollo de un modelo cinemático del mismo.

1.1 El laboratorio de robótica

El Centro de Investigación y Estudios de Posgrado (CIEP) de la Facultad de Ingeniería, en la Universidad Autónoma de San Luis Potosí, cuenta dentro de sus instalaciones con un Laboratorio de Robótica. Inaugurado en el mes de octubre de 1999 con el fin de impulsar la investigación y desarrollo de tecnologías tales como visión artificial para control de robots industriales. Entre otras cosas, el laboratorio cuenta con:

- Un robot industrial Fanuc[®], modelo ArcMate 100i RJ-2, equipado con un sistema de soldadura por arco (MIG-TIG). Entre otras características, se trata de un manipulador articulado con seis grados de libertad y que cuenta además con un puerto de comunicación Ethernet-Protocolo TCP/IP para el desarrollo de interfaces de monitoreo y control con una computadora personal.

¹Por sus siglas en inglés: *Camera Space Manipulation*

El equipo de soldadura consiste de un sistema Lincoln[®] Electric POWER WAVE, totalmente digital y programable para un gran número de clases de soldadura por arco.

- Una tarjeta de adquisición de imágenes de video (Data Translation con 640×480 pixeles de resolución) instalada en una computadora personal (Intel Pentium III, 64MB RAM, 400 MHz) y bajo el sistema operativo Windows NT.
- Dos cámaras de video blanco y negro (SONY Modelo SPT-M304/Zoom 25-75 mm).

1.2 Objetivo y resultados esperados

Se pretende realizar el control de posición del extremo de la herramienta instalada en el robot arriba mencionado sobre superficies planas, cuya posición y orientación dentro del espacio de trabajo del manipulador puede ser arbitraria.

Mediante el uso de una variante del método de CSM que se ha venido utilizando a la fecha, se ha considerado alcanzar los siguientes objetivos:

1. Implementar un algoritmo de posicionamiento que permita, de manera confiable, y precisa, colocar una herramienta fija en el extremo del manipulador sobre un punto seleccionado en una superficie arbitraria.
2. Incrementar la versatilidad del método de CSM al utilizar una matriz de puntos láser sobre la pieza no manipulable a manera de puntos objetivo, con lo cual se pretende evitar la necesidad de colocar marcas visuales artificiales sobre ésta, y agilizar el procedimiento de selección de la posición a la cual se desea llevar el manipulador. Este enfoque proporciona algunas ventajas al algoritmo que estima los ángulos de la configuración interna del manipulador.
3. Desarrollar una interfaz gráfica que permita, a cualquier usuario de PC, controlar la posición del manipulador sobre una superficie plana mediante la selección visual a través de la imagen captada por una cámara de video orientada hacia el espacio de trabajo del manipulador.

En este último inciso, se describe la aplicación directa del método, conocida también como *teleoperación semiautónoma*. Esta modalidad presenta ventajas con respecto a las técnicas convencionales de programación de procesos no repetitivos en robots industriales. A diferencia de los procesos altamente repetitivos, en los que el programa que controla los movimientos del manipulador solo se realiza al ser instalado en el proceso, en una tarea no repetitiva este procedimiento requiere de una gran cantidad de tiempo muerto, en términos de detener la producción, para realizar cada nuevo programa. Además de requerir la contratación de personal especializado para realizar la programación de cada uno de ellos. Ante estos obstáculos, una empresa, cuyos procesos de producción posean estas características, generalmente no está en disposición de adquirir un robot para integrarlo en su planta productiva.

La utilización de un sistema de *teleoperación semiautónoma guiada visualmente* facilita en gran medida la programación de un robot para realizar cualquier tarea, ya que mediante una interfaz gráfica implementada en una computadora personal, la tarea de llevar el extremo del manipulador a una posición deseada se reduce a la selección de un punto en una imagen proyectada en el monitor.

1.3 Robótica y visión artificial

Ante la necesidad de elevar la productividad y la aparición de normas de calidad cada vez más estrictas, los procesos de producción se están viendo orientados cada vez más hacia la automatización.

Hasta hace tiempo, la mayoría de las tareas de fabricación automatizadas se realizaban mediante máquinas diseñadas para realizar funciones específicas dentro de un proceso. La inflexibilidad y alto costo de estas máquinas dedicadas (también llamadas *sistemas de automatización duros*) han favorecido el éxito de los robots en la industria, ya que éstos son capaces de efectuar una variedad de funciones de fabricación en un entorno de trabajo más flexible y a un menor costo de producción.

En la actualidad están disponibles comercialmente muchos tipos de robots industriales. Destacan por su popularidad los dedicados a tareas de fabricación y ensamblaje, además de los relacionados con el manejo de materiales, soldadura por arco y por punto, montaje de piezas, pintura, carga y descarga de máquinas de control numérico, exploraciones espaciales y submarinas, manejo de materiales peligrosos y prótesis. Cada una de estas situaciones conlleva alguna estrategia de control relacionada intrínsecamente con el proceso, como pueden ser el seguimiento y la regulación.

La mayoría de los robots industriales, aunque están controlados por mini y microcomputadoras, son básicamente simples máquinas posicionales. Ejecutan una tarea dada mediante la grabación de secuencias prerregistradas o preprogramadas de movimientos previamente guiados por el usuario con un control de mando portátil (*teach-pendant*). Además, el contacto con su entorno de trabajo a través de sensores es muy bajo dada la complejidad en la implementación de los mismos. Es por esto que los robots se utilizan en tareas repetitivas simples. Al respecto, el presente trabajo tiene por finalidad facilitar el uso y la programación de robots en áreas de la industria que cuentan con procesos no repetitivos, y que conllevan costo de integración mucho mayor al utilizar la técnicas tradicionales.

1.3.1 Visión artificial

El empleo de mecanismos de detección externos permite al robot interactuar con su entorno de una manera flexible. Esto contrasta con el funcionamiento preprogramado en el que un robot se le *enseña* a realizar tareas repetitivas a través de una serie de funciones establecidas. Sin embargo, un robot que pueda *ver* y *sentir* es más fácil de entrenar en tareas complejas mientras que, al mismo tiempo, exige mecanismos de control más estrictos que las máquinas preprogramadas.

Un área muy importante dentro del conjunto de sensores externos es la *visión artificial* (también llamada *visión por computadora*; *visión de máquina*). Al igual que sucede en el ser humano, la capacidad de visión provee al robot, a través de un sistema de cómputo, con un sofisticado mecanismo de percepción que permite a la máquina responder a su entorno de una forma inteligente y flexible.

Los sistemas de visión se emplean para realizar tareas entre las que se incluyen la selección de objetos que se orientan aleatoriamente desde un recipiente o un transportador, la identificación de objetos y la inspección limitada. Estas capacidades se utilizan, de forma selectiva, en aplicaciones tradicionales para reducir el costo de los montajes de sujeción de la pieza o de la herramienta y para adaptarse a variaciones limitadas dentro de su entorno.

La visión artificial se refiere a la detección de datos de visión y su interpretación por una computadora. El sistema de visión típico está constituido por la cámara y equipos de digitalización, una computadora digital y los elementos de hardware y software necesarios para su interconexión. El funcionamiento de un sistema de visión está constituido por tres bloques [10]:

1. Detección y digitalización de datos de imagen.
2. Análisis y procesamiento de imagen.
3. Aplicación.

Las funciones de *detección y digitalización* implican la entrada de datos de visión por medio de una cámara enfocada en la escena de interés conectada a una tarjeta de adquisición de imágenes de video dentro de una PC. Técnicas de iluminación especiales suelen ser empleadas para obtener una imagen de suficiente contraste para su posterior procesamiento.

La matriz de la imagen digitalizada de cada cuadro se almacena para su posterior análisis mediante las funciones de *procesamiento y análisis de imágenes*. En este tipo de procedimientos, la cantidad de datos es reducida y se hace una interpretación de la imagen mediante la evaluación de sus características.

Existen varias técnicas para extraer los valores de características que pueden programarse en una computadora para obtener descriptores de la imagen. Estos descriptores van desde características de forma y tamaño, que pueden calcularse de forma relativamente sencilla a partir de la matriz de imagen con umbrales, hasta la detección de la posición de puntos luminosos sobre la imagen.

La tercera función de un sistema de visión de máquina es la función de las *aplicaciones*. Las aplicaciones actuales de la visión de máquina en la robótica incluyen la inspección, identificación de piezas, localización y orientación. Además existen aplicaciones avanzadas para el empleo de sistemas de visión en tareas complejas de inspección, guiado y navegación.

1.3.2 Clasificación de los sistemas de visión artificial

Los sistemas de visión pueden ser clasificados de varias formas. Una clasificación obvia se basa en el caso de que el sistema trate un modelo bidimensional o tridimensional. Algunas aplicaciones de visión solo requieren un análisis bidimensional. Ejemplos de problemas de visión bidimensionales incluyen la comprobación de las dimensiones de una pieza o la verificación de la presencia de componentes en una línea de montaje.

Muchos sistemas de visión bidimensionales pueden operar sobre una imagen binaria, que es una técnica de fijación de umbrales simple. Se basa en el supuesto de un alto contraste entre los objetos y el fondo. El contraste deseado puede obtenerse, con frecuencia, utilizando un sistema de iluminación controlada.

Sistemas de visión tridimensionales pueden exigir técnicas de iluminación especiales y algoritmos de procesamiento de imágenes sofisticados. Algunos sistemas requieren al menos dos cámaras para conseguir una visión estereoscópica de la escena, mientras que otros sistemas tridimensionales se basan en el empleo de técnicas de triangulación óptica y luz estructurada con una sola cámara.

Otra manera de clasificar los sistemas de visión es de acuerdo con el número de niveles de gris (niveles de intensidad de luz) utilizados para caracterizar la imagen. En una imagen binaria los

valores de los niveles de gris están divididos en dos categorías: negros o blancos. Otros sistemas permiten la clasificación del nivel de grises de cada elemento de imagen en varios niveles, cuyo rango se denomina escala de grises².

1.3.3 Análisis y procesamiento de una imagen

Para el uso de una imagen almacenada en una computadora, se debe programar a ésta para que opere sobre la imagen. Ésta es una tarea importante considerando la gran cantidad de datos que se deben analizar. Una imagen común obtenida de una tarjeta digitalizadora de video consiste en una matriz de datos (*píxeles*) de 480 renglones por 640 columnas. Asumiendo que cada píxel ocupa 1 byte de memoria, se tiene un conjunto 307.2 kbytes de datos obtenidos cada vez que se desea saber que es lo que se encuentra a la vista de una cámara. Esto es una gran cantidad de datos para ser procesada en un corto tiempo; por ello se debe recurrir a varias técnicas para reducir la magnitud del problema del procesamiento de una imagen. Dentro de estas técnicas se incluyen [7]:

1. Captación y preprocesamiento.
2. Reducción de datos de la imagen.
3. Segmentación.
4. Extracción de características (descripción).
5. Reconocimiento del objeto e interpretación.

La *captación* es el primer paso que se realiza antes de comenzar el análisis de una imagen. En esta etapa, es utilizada una cámara de video para obtener una imagen, aunque la mayoría de las aplicaciones de visión de máquina utilizan dos cámaras para propósitos de sensibilidad a la profundidad y la perspectiva. Esta imagen es almacenada en una computadora a través del uso de una tarjeta de digitalización de video.

El *preprocesamiento* incluye técnicas tales como la reducción de ruido y realce de detalles. Estos filtros permiten la obtención de imágenes libres de imperfecciones producidas por factores externos tales como la iluminación, interferencia, etc.

A continuación se analizarán con más detalles algunas de estas técnicas. A excepción de la captación y el preprocesamiento, estas definiciones no requieren seguir el orden señalado.

²Para el caso específico que plantea esta tesis, se trata de un sistema de visión basado en un modelo tridimensional, por lo cual se utilizan como mínimo dos cámaras para la adquisición de las imágenes. Tales cámaras proporcionan una matriz de píxeles cuyo valor corresponde a una escala de grises que va desde el negro (0) hasta el blanco (255). Dentro de los algoritmos de extracción de características de la imagen se maneja también un umbral con un valor de aproximadamente 250. Cabe aclarar que no se trata de un modelo de visión estereoscópica, ya que para esto se requiere conocer de manera muy precisa la relación geométrica entre una cámara y otra.

Reducción de los datos de la imagen

El objetivo de la reducción de los datos de la imagen es, como su nombre lo indica, la reducción del volumen de la información contenida dentro de cada imagen. A modo de paso preliminar en el análisis de datos, los dos métodos que han encontrado un mayor empleo para la reducción de datos son:

- Conversión digital.
- Uso de ventanas.

La *conversión digital* reduce el número de niveles de grises utilizados por el sistema de visión de máquina. Por ejemplo, un registro de 8 bits utilizado para cada uno de los píxeles, tiene $2^8 = 256$ niveles de gris. Dependiendo de los requerimientos de la aplicación, la conversión digital puede utilizarse para reducir el número de niveles de gris mediante el uso de menos bits para representar la intensidad de luz del píxel. Así, cuatro bits, reducen el número de niveles de gris a 16. Este tipo de conversión reducirá, de modo significativo, la magnitud del problema de procesamiento de la imagen.

El *uso de ventanas* implica la utilización de solo una parte de la imagen total almacenada. Esta parte recibe el nombre de ventana. Una de las razones para la utilización de las ventanas es que el reconocimiento adecuado de un objeto requiere solo unas partes determinadas de la escena total.

Segmentación

La *segmentación* es un método general que se aplica a varios métodos de la reducción de datos. El objetivo es agrupar áreas de una imagen que tengan características similares dentro de entidades distintas, que representan partes de la imagen. Los bordes (límites) o las regiones (áreas) representan dos segmentos naturales de una imagen. Existen muchos métodos de segmentar una imagen, entre los cuales se encuentran [10], [7]:

- Fijación de umbrales.
- Crecimiento de región.
- Detección de borde.

En su forma más simple, la *fijación de umbrales* es una técnica de conversión binaria, en la que cada píxel es convertido en un valor binario, blanco o negro. De acuerdo a varias características del sistema (como iluminación, tipo de cámaras de video, etc.) se establece que intensidad o nivel de gris será el límite entre el blanco y el negro. Puesto que tratamos de distinguir el objeto y el segundo plano, el procedimiento consiste en establecer un umbral y en asignar bits diferentes al objeto y el segundo plano o fondo. Para mejorar la capacidad de distinción se deben aplicar técnicas especiales de iluminación para generar un elevado contraste.

La fijación de umbrales es la técnica utilizada con más frecuencia para la segmentación en aplicaciones de visión industriales. Las razones son la rapidez y facilidad de realización, además, la iluminación es una variable factible de controlar en un entorno industrial.

Una vez que se establece la fijación de umbrales para una imagen particular, el próximo paso es identificar las áreas asociadas con los objetos dentro de la imagen. Dichas regiones suelen tener propiedades de píxeles uniformes sobre el área como color e intensidad de luz. Para la gran mayoría de los casos se manejan solo los atributos relacionados con la intensidad de luz (niveles de gris).

El *crecimiento de región* es un conjunto de técnicas de segmentación en las que los píxeles se agrupan en regiones llamadas elementos de cuadrícula basadas en similitudes de atributos. Las regiones definidas pueden ser examinadas en cuanto a si son independientes o se pueden fusionar a otras regiones por medio de un análisis de la diferencia en sus propiedades y su conectividad espacial. Por ejemplo, considerando la imagen que se muestra en la Figura 1.1a, para distinguir entre los objetos y el segundo plano, se asigna 1 a cualquiera de los elementos de la cuadrícula ocupados por un objeto y 0 a los elementos del segundo plano. Una práctica usual es utilizar una retícula de muestreo cuadrada con píxeles igualmente espaciados a lo largo de cada uno de los lados de la cuadrícula. Para la imagen bidimensional de una llave, como la que se muestra en la Figura 1.1a, resulta en el modelo indicado en la 1.1b. Esta técnica se utiliza con frecuencia como un análisis de primer paso para la división de la imagen en segmentos identificables.

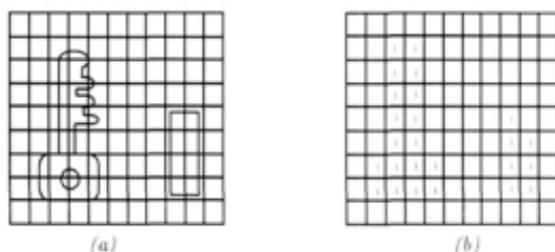


Figura 1.1: Segmentación de imagen, (a) modelo de imagen con cuadrícula y (b) imagen segmentada.

Para una imagen simple como una mancha clara sobre un fondo oscuro, la técnica anterior puede proporcionar una información utilizable, sin embargo, para imágenes complicadas, esta técnica puede no proporcionar una división adecuada de una imagen en un conjunto de regiones significativas. Dichas regiones pueden tener píxeles que estén conectados entre sí y que tengan atributos similares, por ejemplo, el nivel de gris. Una técnica típica de *crecimiento de región* para imágenes complejas debe tener el procedimiento siguiente, el cual se ilustra en la Figura 1.2:

1. Seleccionar un píxel que cumpla con un criterio para la inclusión en una región. El caso más simple es el de seleccionar un píxel blanco (o con una intensidad superior a un umbral establecido) y asignar un valor 1.
2. Comparar el píxel seleccionado con todos los píxeles adyacentes. Asignar a los píxeles adyacentes un valor equivalente si se produce una concordancia de los atributos.
3. Ir a un píxel adyacente equivalente y repetir el proceso hasta que no se pueda añadir ningún píxel equivalente más a la región.

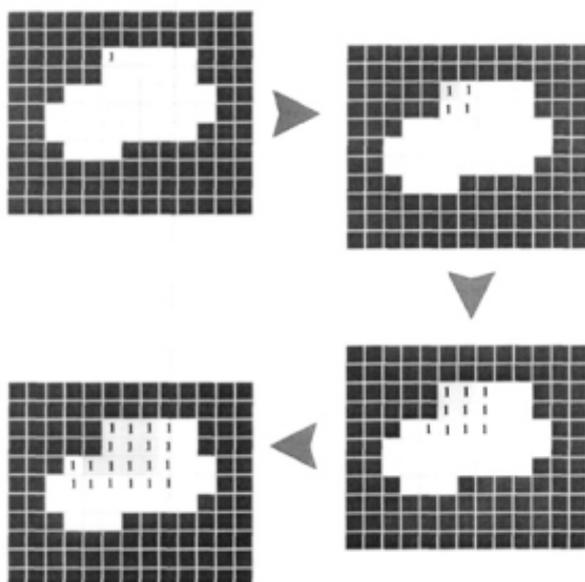


Figura 1.2: Procedimiento de crecimiento de región para el análisis de una imagen.

Este procedimiento simple de “crecer” regiones al rededor de un pixel se repite hasta que ninguna región nueva pueda añadirse a la imagen.

La *detección de bordes* considera el cambio de intensidad que se produce en los pixeles en el contorno o bordes de un objeto. Dado que se ha encontrado una región con atributos similares, pero se desconoce la forma del contorno, éste último se puede determinar mediante un procedimiento simple de seguimiento de borde. Este método se ilustra mediante el esquema de una imagen binaria, tal como se muestra en la Figura 1.3, el procedimiento es el de explorar la imagen hasta que se encuentra un pixel dentro de la región. Una vez encontrado un pixel en el interior de una región, torcer a la izquierda y avanzar o, de lo contrario torcer a la derecha y avanzar un paso. El procedimiento se detiene cuando se atraviesa el contorno y el camino vuelve al pixel inicial. El procedimiento de seguimiento del contorno descrito se puede extender a imágenes de nivel de gris.

Extracción de características

En las aplicaciones de visión de máquina, es necesario con frecuencia distinguir un objeto de otro. Esto suele realizarse mediante las características que describen unívocamente a un objeto, como por ejemplo el área, el diámetro o el perímetro. Una característica, en el contexto de los sistemas de visión, es un parámetro único que permite la facilidad de comparación e identificación. A conti-

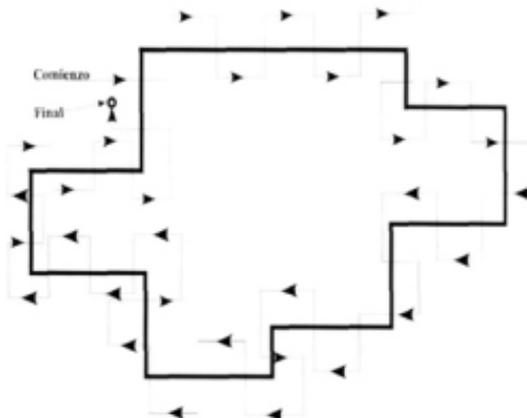


Figura 1.3: Procedimiento de seguimiento de bordes para detectar el límite de una imagen binaria.

nuación se describen algunas de las características utilizadas con más frecuencia en las aplicaciones de visión [10]:

- **Nivel de gris** (máximo, medio, mínimo).
- **Área**.
- **Longitud del perímetro**.
- **Diámetro**.
- **Centro de gravedad**. Para todos los píxeles (n) en una región donde cada pixel está especificado por las coordenadas (x, y) ; las coordenadas x e y del centro de gravedad se definen como:

$$C_x = \frac{1}{n} \sum_x x$$

$$C_y = \frac{1}{n} \sum_y y$$

- **Relación de aspecto**. La relación longitud-anchura de un rectángulo de contorno que rodea al objeto. Un objetivo es encontrar el rectángulo que da la relación de aspecto mínima.

Las técnicas disponibles para la extracción de valores de características en los casos bidimensionales se pueden clasificar en forma general como las que se refieren a las características de contorno y las que tratan las características de área. Las diversas características se pueden utilizar para

identificar el objeto completo o solo una parte del mismo, y para determinar su localización y/u orientación.

Los procedimientos de crecimiento de región, descritos en la sección anterior, pueden utilizarse para determinar el área de una imagen de un objeto. El perímetro o el contorno que encierra un área específica se puede determinar indicando la diferencia en la intensidad del píxel en el contorno y contando simplemente los píxeles de la región segmentada que son adyacentes a los que no pertenecen a la región, es decir, que no pertenecen al contorno. Un objetivo importante en la selección de estas características es que no dependan de la posición ni de la orientación. El sistema de visión no debe depender de que el objeto se presente ante la cámara con una relación conocida y fija.

Reconocimiento del objeto

El próximo paso en el procesamiento de los datos de la imagen es identificar el objeto que la imagen representa. Este problema de identificación se resuelve utilizando la información sobre las características extraídas que se describe en la subsección anterior. El algoritmo de reconocimiento debe ser lo suficientemente potente para identificar unívocamente al objeto. Las técnicas de reconocimiento de objeto utilizadas se pueden clasificar dentro de dos categorías principales [10]:

1. Técnicas de coincidencia de plantilla.
2. Técnicas estructurales.

Las *técnicas de coincidencia de plantilla* son un subconjunto de las técnicas estadísticas más generales de reconocimiento de modelos que sirven para clasificar los objetos en una imagen dentro de las categorías predeterminadas. El problema básico de la coincidencia de plantilla es comparar el elemento con un conjunto de características del modelo almacenado, definido como modelo de plantilla. Éste último modelo se obtiene durante el procedimiento de adiestramiento en el que el sistema de visión se programa para reconocer los elementos prototipos conocidos. Estas técnicas son aplicables si no existen requerimientos para un gran número de plantillas de modelos. El procedimiento está basado en el uso de un número suficiente de características para minimizar la frecuencia de errores en el proceso de clasificación. Las características del elemento en la imagen (por ejemplo, su área, diámetro, relación de aspecto, etc.) se comparan con los valores correspondientes almacenados que constituyen la plantilla almacenada. Cuando se encuentra una coincidencia, considerando algunas variaciones estadísticas en el proceso de comparación, el elemento ha sido clasificado.

Las *técnicas estructurales* de reconocimiento de modelos consideran las relaciones entre las características o bordes de un objeto. Por ejemplo, si la imagen de un objeto se puede subdividir en cuatro líneas (las líneas reciben el nombre de primitivas) conectadas en sus puntos extremos y las líneas conectadas forman ángulos rectos, entonces el objeto es un rectángulo. Esta clase de técnica, conocida como reconocimiento de modelo sintáctico, es la técnica estructural utilizada con más frecuencia. Las técnicas estructurales difieren de las técnicas de coincidencia de plantilla, en que estas últimas tratan a un modelo sobre una base cuantitativa y en que se ignoran la mayoría de las interrelaciones de partes entre las primitivas del objeto.

Puede considerarse excesivo el tiempo necesario para el reconocimiento completo de un modelo. En consecuencia, suele ser más adecuado la búsqueda de regiones más simples bordes dentro de una imagen. Estas regiones más simples pueden utilizarse para extraer las características requeridas.

1.3.4 Aplicaciones de robótica

Algunas de las aplicaciones más usuales de la visión de máquina son tareas de inspección que no implican el empleo de un robot industrial. Una aplicación típica es aquella en donde el sistema de visión de máquina se instala en una línea de producción de alta velocidad para aceptar o rechazar las piezas fabricadas en la línea. Las piezas no aceptables se sacan de la línea mediante un dispositivo mecánico comunicado con el sistema de visión.

Las aplicaciones de robótica de la visión artificial están dentro de una de estas tres amplias categorías indicadas a continuación [7]:

1. Inspección.
2. Identificación.
3. Servopresentación visual y navegación.

En la primera categoría, la función primaria es el proceso de *inspección*. Este proceso es llevado a cabo por el sistema de visión y el robot se utiliza en un papel secundario para apoyar la aplicación. Dentro de la inspección se incluye la comprobación de los defectos en bruto de superficies, detección de defectos en etiquetados, verificación de la presencia de componentes en el montaje, medición de la exactitud dimensional y comprobación para detectar la presencia de agujeros y otras características en una pieza. Cuando estas clases de operaciones de inspección se realizan manualmente, existe una tendencia al error. Además, el tiempo necesario para la mayoría de las operaciones manuales de inspección requiere que los procedimientos se realicen sobre una base de muestreo. Con la visión de máquina, estos procedimientos se llevan a cabo de forma automática, realizando un porcentaje de inspección del 100% y, por lo general, en un tiempo mucho menor.

En la segunda categoría, *identificación*, se hace referencia a las aplicaciones en las que el propósito del sistema de visión es reconocer y clasificar un objeto en vez de inspeccionarlo. La inspección implica, además, la aceptación o rechazo de la pieza. La identificación lleva consigo un proceso de reconocimiento en el que la pieza, su posición y/u orientación es determinada. Esto suele ir seguido por una decisión posterior y una acción tomada por el robot. Entre las aplicaciones de identificación de visión de máquina se incluyen la clasificación de las piezas y captación de objetos que se orientan de forma aleatoria desde un transportador o receptáculo.

En la tercera categoría de aplicación, *servopresentación visual y control navegacional*, el objetivo del sistema de visión es dirigir las acciones del robot (y demás dispositivos periféricos del robot) basándose en su entrada visual. El ejemplo genérico de esta categoría es el uso del sistema de visión de máquina para regular la posición del efector final del robot hacia un objeto dentro del espacio de trabajo (seguimiento de trayectorias). Entre los ejemplos industriales de esta aplicación se incluyen el posicionamiento de una pieza, recuperación y reorientación de piezas en movimiento a lo largo de un transportador, montaje, recolección desde recipientes y seguimiento del cordón en la soldadura por arco continua.

Un ejemplo de control de navegación es la planificación automática de las trayectorias de robots y en evitar colisiones con el empleo de datos visuales. Es evidente que los datos visuales solamente son una entrada importante en este tipo de tarea y se requiere el desarrollo de los algoritmos de poderosos algoritmos en el controlador para usar estos datos en navegación y evitar colisiones.

Por otra parte, la *interpretación* es el proceso que dota a un sistema de visión de un nivel superior de conocimiento sobre su entorno al que pueden ofrecer cualquiera de los procesos vistos hasta ahora. Según este enfoque, la interpretación incluye todos aquellos métodos que tengan que ver con la comprensión de una escena visual.

La potencia de un sistema de visión artificial se determina por su capacidad de extraer información con significado de una escena, dentro de un amplio margen de condiciones de visión y usando un mínimo de conocimiento sobre los objetos que se están observando. Hay gran cantidad de factores que hacen de este tipo de procesos una tarea difícil, como pueden ser las variaciones de la iluminación, los cuerpos ocultos o la geometría de la visión.

1.4 Manipulación de espacio de cámara

En esta sección se describen, en términos generales, los fundamentos del método de CSM además se incluye una pequeña introducción al modelo de cámara en el cual se basa dicha técnica. En el capítulo siguiente se presenta el desarrollo completo del método y la implementación en este trabajo.

1.4.1 Justificación

El uso de visión permite a un robot industrial realizar tareas sobre una pieza de trabajo cuya posición y orientación en el espacio físico es desconocido. En este tipo de operaciones la información obtenida desde los sensores de imagen es usada para guiar el extremo de interés del robot hacia la posición y orientación deseada. Para la interpretación correcta de esta información se han seguido diferentes enfoques:

Calibración de cámaras antes de realizar la maniobra mediante la observación de un conjunto de puntos cuya posición es conocida con precisión (también llamados *puntos de calibración*) y estimación de los parámetros en un modelo que relaciona el espacio cartesiano tridimensional de estos puntos con sus coordenadas correspondientes dentro del plano de la imagen.

Los parámetros evaluados en el proceso de calibración algunas veces son usados (junto con el modelo cinemático del manipulador) para evaluar directamente la configuración de ángulos necesaria para alcanzar una posición y orientación particular del elemento final del manipulador ó de la pieza sujeta por el mismo.

Puede haber algunas dificultades inherentes asociadas con los métodos de control basados en calibración [3]:

1. Estos métodos están limitados directamente por la exactitud del modelo cinemático, por ello es necesaria la elaboración de un procedimiento específico de calibración cinemática. El exceso de calor y las deformaciones por movimiento, son algunas de las causas que pueden introducir un error dentro del modelo cinemático de un robot.

2. La transformación de coordenadas entre el marco de referencia asignado a la base del robot y el marco de referencia asignado a la cámara debe ser establecido con precisión. Éste es un esquema difícil de evaluar. Además, el uso de un número mayor de cámaras aumenta la posibilidad de introducir error en el modelo.
3. Si la posición-orientación de la cámara o sus parámetros internos cambian después de la etapa de calibración y antes de ser realizada la maniobra (o incluso durante la maniobra), la calibración inicial no puede seguir siendo aplicada.
4. Si la imagen del objeto sujetado por el manipulador es obstruida por otro objeto en alguna etapa de la maniobra, será necesario el uso de una estimación de su localización dentro del plano de imagen.

Un enfoque alternativo para el control de posicionamiento basado en visión consiste en estimar y redefinir la relación no lineal local entre la configuración interna del manipulador y la localización dentro del plano de la imagen de algunos puntos, marcas colocadas sobre la herramienta en el extremo del manipulador o sobre la pieza de trabajo.

1.4.2 Descripción general del método

La técnica de CSM surgió a mediados de los ochenta (Skaar y col. 1987, 1990) como una forma de alcanzar tanto robustez como precisión en la manipulación guiada visualmente sin la necesidad de asegurar y mantener una calibración precisa de las cámaras y la cinemática del manipulador, como sucede en los métodos basados en calibración.

CSM es una técnica libre de la fragilidad que conllevan los métodos basados en calibración. Su robustez se deriva del uso de los datos actuales obtenidos durante la aproximación al punto que se desea alcanzar, ya que requiere del muestreo de la configuración interna del manipulador y su posición en el plano de la imagen de cada cámara. Estas muestras son utilizadas para modificar los parámetros fundamentales del método de acuerdo a las perturbaciones del sistema. De esta forma, el modelo se adapta a las modificaciones de su entorno, tales como cambios en la posición y orientación de las cámaras, cambios de sus parámetros internos (distancia focal, apertura, etc.), deformación mecánica de los eslabones del robot, etc. Mediante el ajuste de la velocidad del manipulador, y por consecuencia, de la densidad de las muestras de video y configuración interna, la precisión final puede ser establecida como se requiera.

Uno de los conceptos en los que la manipulación de espacio de cámara se sustenta es que la identificación de los parámetros requeridos para describir el dispositivo de obtención de imágenes se realiza directamente en términos del marco de referencia de cada cámara. Esto reduce notoriamente la sensibilidad a las perturbaciones causadas por una calibración relativa a un marco de referencia fijo.

Otro enfoque para la manipulación guiada directamente que trabaja de la misma manera con el plano de la imagen son las técnicas conocidas en forma genérica como *visual servoing*. En este tipo de técnicas, los incrementos en las juntas del robot, necesarios para producir los incrementos deseados en el plano de la cámara, son establecidos en primer lugar, a menudo usando la cinemática del manipulador y algún tipo de calibración de cámara. Entonces se produce una acción de control

para reducir la diferencia, dentro del espacio de cámara, entre la posición actual y la deseada. Este enfoque se basa principalmente en el cálculo del *Jacobiano*, y produce solo correcciones locales (primer orden). En cambio, CSM usa una descripción algebraica no lineal de la relación local entre las configuración interna del manipulador y las coordenadas en el espacio de cámara de puntos localizados en el extremo del mismo.

CSM requiere un mínimo de dos cámaras estacionarias adecuadamente separadas. A cada una de estas cámaras le es asignado un espacio bidimensional, dentro del cual es perseguido independientemente un conjunto de puntos objetivo, utilizando estimaciones actualizadas frecuentemente de la relación entre la configuración interna del manipulador y la localización de la zona de interés del mismo dentro del área de visión de cada cámara participante.

Basado en estimación³, CSM no solo hace uso de la información más reciente de la imagen, sino que incorpora toda la información relacionada con la maniobra actual. Por lo tanto, con CSM esta información debe, de alguna forma, ser almacenada durante el desarrollo de la maniobra actual. No es necesario que los datos incluyan la representación instantánea de la posición actual del robot; los datos disponibles pueden terminar incluso mucho antes de que la maniobra esté concluida. La estimación de la posición de las juntas del manipulador es realizada en “lazo abierto” (*open-loop*) mientras la maniobra termina, de tal manera que en promedio, el error de espacio de cámara final, en todas las cámaras participantes es minimizado. Estas posiciones calculadas en lazo abierto pueden ser ajustadas mediante la adquisición de nuevos datos visuales y mejorar las estimaciones dentro de las áreas de interés.

La particular forma del algoritmo de estimación no lineal considerada en el modelo permite el uso de varias matrices de “momentos” [9] para compactar los datos en una forma que no crece con la adquisición de más información. De la misma manera, permite compactar datos relacionados con el punto objetivo relacionado con el espacio de cámara, a los cuales se les puede atribuir un peso individualmente. Esta matriz de momentos es usado junto con las mejores estimaciones del modelo que describe el espacio de cámara para determinar la configuración interna del manipulador para las posiciones intermedias y final de la maniobra.

En términos generales, el método de manipulación de espacio de cámara funciona de la manera siguiente:

- La posición y orientación física deseada del extremo del manipulador es alcanzada cuando el sistema coordinado asociado a éste ocupa una posición específica en el sistema coordinado que está asignado a la pieza de trabajo.
- La tarea es realizada después de estimar la configuración interna del robot con base en el modelo de cámara no lineal y el conocimiento de la posición en el espacio de cámara de las marcas colocadas en la herramienta del manipulador.
- Los parámetros involucrados en el modelo que describe a las cámaras son refinados para realizar una nueva estimación de la configuración interna del manipulador. Los parámetros

³**Estimación de parámetros.** La teoría de estimación se deriva del principio de mínimos cuadrados de Gauss. Consiste principalmente en la determinación de los parámetros desconocidos que conforman la hipótesis matemática que describe a un fenómeno, basándose únicamente en observaciones o mediciones del mismo.

requeridos para esta nueva evaluación de parámetros son obtenidos durante la ejecución de la tarea.

CSM permite precisión dentro de una región nominal específica del espacio, utilizando un modelo de cámara simple junto con el modelo cinemático del manipulador. Esto, en parte, porque pueden ser asignados grandes “pesos” a los datos obtenidos cerca del punto final de una maniobra.

El método de CSM provee de un procedimiento robusto para realizar una maniobra de un cuerpo rígido guiada visualmente y dentro de un espacio tridimensional, sin la necesidad de cualquier tipo de calibración de las cámaras o del modelo cinemático del manipulador.

1.4.3 Modelo de cámara

La relación entre el espacio físico tridimensional y el espacio bidimensional de la imagen es descrito aproximadamente por una proyección en perspectiva de dicho espacio de tres dimensiones sobre el plano de la imagen. Esta proyección está definida por el *modelo de orificio de cámara*, el cual relaciona puntos de coordenada (X_i, Y_i, Z_i) referidos a un sistema coordenado unido a la cámara XYZ con sus coordenadas correspondientes dentro del plano de la imagen (x_{ci}, y_{ci}) de acuerdo a:

$$x_{ci} = f \frac{X_i}{Z_i} \quad (1.1a)$$

$$y_{ci} = f \frac{Y_i}{Z_i} \quad (1.1b)$$

donde f es la longitud focal efectiva y (X, Y) son paralelos a (x_c, y_c) respectivamente, además, el eje Z está en la dirección del eje óptico de la cámara, como se muestra en la Figura 1.4.

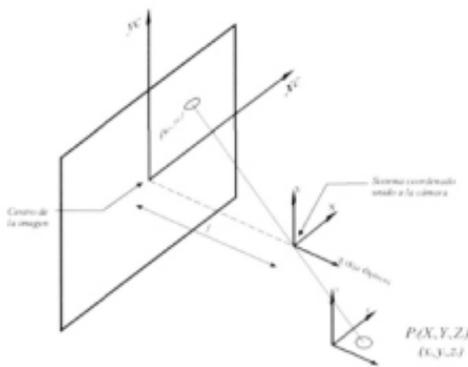


Figura 1.4: Sistema de referencia unido a la cámara.

Los puntos referidos al marco de referencia unido a la cámara pueden especificarse en relación a un segundo sistema coordenado xyz usando la relación:

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = \begin{bmatrix} e_1^2 + e_2^2 - e_3^2 - e_4^2 & 2(e_2e_3 + e_1e_4) & 2(e_2e_4 - e_1e_3) & X_0 \\ 2(e_2e_3 - e_1e_4) & e_1^2 - e_2^2 + e_3^2 - e_4^2 & 2(e_3e_4 + e_1e_2) & Y_0 \\ 2(e_2e_4 + e_1e_3) & 2(e_3e_4 - e_1e_2) & e_1^2 - e_2^2 - e_3^2 + e_4^2 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (1.2)$$

donde los parámetros e_1, e_2, e_3 y e_4 son los parámetros de Euler que satisfacen la condición $e_1^2 + e_2^2 + e_3^2 + e_4^2 = 1$ y (X_0, Y_0, Z_0) localiza el origen del sistema coordenado xyz relativo al sistema coordenado en la cámara XYZ . De esta forma, puede verse claramente que el modelo de orificio de cámara contiene siete parámetros independientes: X_0, Y_0, Z_0, f y tres de los cuatro parámetros de Euler.

1.4.4 Modelo ortográfico⁴

El modelo de orificio de cámara puede ser simplificado al considerar el límite asintótico de las ecuaciones 1.1, obtenido cuando Z_0 es mucho mayor que las cantidades x_i, y_i, z_i, X_0 y Y_0 ($\lim_{Z_0 \rightarrow \infty}$). Las coordenadas en el plano de la imagen (x_{ci}, y_{ci}) obtenidas de estas ecuaciones pueden reescribirse como sigue:

$$x_{ci} = f \frac{X_i}{Z_i} = \frac{N_x}{D + \frac{1}{Z_0}} = \Psi_x(\epsilon) \quad (1.3a)$$

$$y_{ci} = f \frac{Y_i}{Z_i} = \frac{N_y}{D + \frac{1}{Z_0}} = \Psi_y(\epsilon) \quad (1.3b)$$

donde,

$$N_x = f[(e_1^2 + e_2^2 - e_3^2 - e_4^2)x_i + 2(e_2e_3 + e_1e_4)y_i + 2(e_2e_4 - e_1e_3)z_i + X_0]$$

$$N_y = f[2(e_2e_3 - e_1e_4)x_i + (e_1^2 - e_2^2 + e_3^2 - e_4^2)y_i + 2(e_3e_4 + e_1e_2)z_i + Y_0]$$

$$D = [2(e_2e_4 + e_1e_3)x_i + 2(e_3e_4 - e_1e_2)y_i + (e_1^2 - e_2^2 - e_3^2 + e_4^2)z_i]$$

$$\epsilon = \frac{1}{Z_0}$$

Expandiendo Ψ_x a través de una serie de Taylor alrededor de $\epsilon = 0$ se obtiene la siguiente expresión:

⁴Debido a sus características, este modelo realiza una proyección ortogonal sobre un plano de las imágenes obtenidas, eliminando el elemento de perspectiva al hacer parecer del mismo tamaño dos objetos situados a diferente distancia. Éste es el sentido del término *ortográfico*.

$$\begin{aligned}\Psi_x(\epsilon) &= \Psi_x \Big|_{\epsilon=0} + \frac{\partial \Psi_x}{\partial \epsilon} \Big|_{\epsilon=0} \epsilon + \frac{\partial^2 \Psi_x}{\partial \epsilon^2} \Big|_{\epsilon=0} \frac{\epsilon^2}{2} + \dots \\ &= 0 + N_x \epsilon - M_x D \epsilon^2 + \dots\end{aligned}$$

El mismo procedimiento se repite para y_c . Si se considera únicamente la aproximación de primer orden ($\Psi_x(\epsilon) = N_x \epsilon$), entonces $\frac{f}{z_0}$ puede combinarse con el resto de los parámetros para formar un nuevo vector de parámetros \mathbf{C}^5 . La aproximación resultante del modelo de cámara de orificio es:

$$\begin{aligned}x_{ci} &= (C_1^2 + C_2^2 - C_3^2 - C_4^2)x_i + 2(C_2C_3 + C_1C_4)y_i + 2(C_2C_4 + C_1C_3)z_i + C_5 \\ &\equiv f_x(x, y, z; \mathbf{C})\end{aligned}\quad (1.4a)$$

$$\begin{aligned}y_{ci} &= 2(C_2C_3 - C_1C_4)x_i + (C_1^2 - C_2^2 + C_3^2 - C_4^2)y_i + 2(C_3C_4 + C_1C_2)z_i + C_6 \\ &\equiv f_y(x, y, z; \mathbf{C})\end{aligned}\quad (1.4b)$$

Las ecuaciones (1.4) representan el límite asintótico de (1.1) para un valor de Z_0 muy grande y además definen el *modelo ortográfico de cámara*. Si asumimos una proyección en perspectiva ideal, las muestras obtenidas del plano de la imagen deben concordar exactamente con el modelo de cámara de orificio, mientras que el modelo ortográfico tiene un error. Sin embargo, al confinar las muestras a una pequeña región de la superficie de trabajo, en relación con Z_0 , este error es minimizado y tales muestras son concordantes con el modelo.

Las constantes C_1 a C_6 , representan los *parámetros de visión*, los cuales definen la relación entre la localización física de puntos tridimensionales y su correspondiente localización en el plano de imagen. Los primeros cuatro parámetros están relacionados con los parámetros de Euler, utilizados para caracterizar la orientación relativa entre el sistema coordenado fijo en la cámara y el sistema coordenado asociado al robot. Los últimos dos parámetros de visión definen la localización, en el espacio de cámara, del origen del sistema coordenado en la base del robot.

En el modelo ortográfico, los parámetros de visión implican las siguientes relaciones:

$$x_c \propto X_i$$

$$y_c \propto Y_i$$

donde el factor de proporcionalidad incluye no solo el factor de escala, sino también una conversión entre las unidades usadas para la medición de la posición física de los puntos en un espacio tridimensional (mm, in, m, etc.) a las usadas para medir la posición de los puntos proyectados en el plano de la imagen (píxeles).

⁵Por razones de espacio, la estricta terminología de los símbolos utilizados será omitida. Para una completa referencia, una lista de símbolos es presentada al inicio de esta tesis.

1.4.5 Estimación de los parámetros de visión

En CSM, las ecuaciones (1.4) son combinadas con el modelo cinemático del manipulador para formar la relación local y no lineal entre el valor actual de sus ángulos y la posición, dentro del espacio de cámara, de las marcas colocadas sobre el objeto manipulado⁶ fijo en el extremo del robot. El modelo cinemático es usado para obtener la relación entre la posición de las articulaciones del manipulador y la posición de dichas marcas dentro del espacio cartesiano tridimensional asociado a la base del manipulador.

El modelo representado por las ecuaciones (1.4) es usado como base para establecer localmente la relación no lineal entre el vector de ángulos de la configuración interna del robot Θ y la apariencia, dentro del espacio de cámara, de la i -ésima de n marcas visuales colocadas sobre el objeto manipulable.

Definiendo (x_i, y_i, z_i) como las coordenadas físicas de una marca luminosa, se puede establecer la siguiente relación con base en el modelo cinemático del manipulador:

$$\begin{aligned}x_i &= r_{x_i}(\Theta) \\y_i &= r_{y_i}(\Theta) \\z_i &= r_{z_i}(\Theta)\end{aligned}\tag{1.5}$$

De acuerdo a las ecuaciones (1.4), el vector de parámetros de visión \mathbf{C}_j , donde j indica la cámara asociada, relaciona la apariencia de la i -ésima marca en el plano de imagen (x_{c_i}, y_{c_i}) con Θ como sigue:

$$\begin{aligned}x_{c_i} &= f_x[r_{x_i}(\Theta), r_{y_i}(\Theta), r_{z_i}(\Theta); \mathbf{C}_j] \\y_{c_i} &= f_y[r_{x_i}(\Theta), r_{y_i}(\Theta), r_{z_i}(\Theta); \mathbf{C}_j]\end{aligned}\tag{1.6}$$

El procedimiento para obtener el vector de seis parámetros de visión $\mathbf{C} = [C_1, \dots, C_6]^T$ de cada cámara consiste en minimizar la ecuación (1.7), de tal manera que satisfaga el modelo dado en las ecuaciones (1.4), tomando como referencia un conjunto de m muestras x_{c_i} , y_{c_i} , x_i , y_i y z_i , siguiendo el criterio de mínimos cuadrados de Gauss.

$$J(\mathbf{C}) = \sum_{i=1}^m \left\{ [x_{c_i} - f_x(r_{x_i}(\Theta), r_{y_i}(\Theta), r_{z_i}(\Theta); \mathbf{C})]^2 + [y_{c_i} - f_y(r_{x_i}(\Theta), r_{y_i}(\Theta), r_{z_i}(\Theta); \mathbf{C})]^2 \right\} W_i \tag{1.7}$$

⁶Para el caso particular de esta tesis, las marcas colocadas sobre el manipulador consisten en fuentes luminosas. La Sección 2.2 presenta detalles al respecto.

donde m indica el número de mediciones de la localización en el plano de la imagen de las señales visuales (x_{ci}, y_{ci}) , correspondientes a la localización física (x_i, y_i, z_i) , la cual, de acuerdo a (1.5), se obtienen a partir del modelo cinemático del manipulador. W_i representa un peso relativo positivo dado a cada medición [11] y es evaluado de forma que favorezca las mediciones más recientes (que deben de estar relacionadas con la tarea de aproximación actual)⁷.

Las relaciones $f_x(x_i, y_i, z_i; \mathbf{C})$ y $f_y(x_i, y_i, z_i; \mathbf{C})$, definidas mediante la suposición del modelo ortográfico, pueden reescribirse de la forma indicada en las ecuaciones siguientes:

$$f_x(x_i, y_i, z_i; C) \equiv b_1(\mathbf{C})x_i + b_2(\mathbf{C})y_i + b_3(\mathbf{C})z_i + b_4(\mathbf{C}) \quad (1.8a)$$

$$f_y(x_i, y_i, z_i; C) \equiv b_5(\mathbf{C})x_i + b_6(\mathbf{C})y_i + b_7(\mathbf{C})z_i + b_8(\mathbf{C}) \quad (1.8b)$$

donde b_1, \dots, b_8 agrupa el conjunto de parámetros de visión dentro de una expresión no lineal, como se muestra a continuación:

$$\begin{aligned} b_1(\mathbf{C}) &= C_1^2 + C_2^2 - C_3^2 - C_4^2 & b_5 &= 2(C_2C_3 - C_1C_4) \\ b_2(\mathbf{C}) &= 2(C_2C_3 + C_1C_4) & b_6 &= C_1^2 - C_2^2 + C_3^2 - C_4^2 \\ b_3(\mathbf{C}) &= 2(C_2C_4 - C_1C_3) & b_7 &= 2(C_3C_4 + C_1C_2) \\ b_4(\mathbf{C}) &= C_5 & b_8 &= C_6 \end{aligned} \quad (1.9)$$

Una definición equivalente de $J(\mathbf{C})$ puede obtenerse de forma matricial como sigue:

$$J(\mathbf{C}) = \Delta \mathbf{Y}^T \mathbf{W} \Delta \mathbf{Y} \quad (1.10)$$

donde el vector $\Delta \mathbf{Y}$ y la matriz cuadrada diagonal \mathbf{W} se expresan como:

$$\Delta \mathbf{Y} = \begin{bmatrix} x_{c1} - f_x(x_1, y_1, z_1; \mathbf{C}) \\ y_{c1} - f_y(x_1, y_1, z_1; \mathbf{C}) \\ \vdots \\ x_{cm} - f_x(x_m, y_m, z_m; \mathbf{C}) \\ y_{cm} - f_y(x_m, y_m, z_m; \mathbf{C}) \end{bmatrix} \quad (1.11)$$

⁷Dentro del procedimiento de mínimos cuadrados, el valor W_i^j representa el énfasis o la ponderación que se le da a cada muestra tomada de acuerdo a la precisión asociada a la medición. Empíricamente, el peso W asignado a cada medición puede ser definido proporcional a la precisión estimada de las mediciones. Siendo más estrictos, W se puede representar como el recíproco de la variancia de la medición.

$$\mathbf{W} = \begin{bmatrix} W_1 & \dots & 0 \\ & W_2 & \\ \vdots & \ddots & \vdots \\ 0 & \dots & W_{m-1} & W_m \end{bmatrix} \quad (1.12)$$

Como ya ha sido mencionado, debido a que los parámetros de visión incluidos en f_x y f_y dentro de las ecuaciones (1.8a) y (1.8b) respectivamente, aparecen en forma no lineal, el proceso de estimación usado es iterativo y puede derivarse a partir de un proceso de estimación no lineal de *corrección de mínimos cuadrados* [11]. Este proceso debe tomar un valor inicial para los parámetros de visión, \mathbf{C}_c ; a partir de este valor, se calcula un incremento $\Delta\mathbf{C} = [\Delta C_1, \dots, \Delta C_6]^T$, que se evalúa al considerar una linealización del vector $\Delta\mathbf{Y}$ alrededor de \mathbf{C}_c usando una expansión en series de Taylor:

$$\Delta\mathbf{Y}_p \approx \Delta\mathbf{Y}_c - \mathbf{A}\Delta\mathbf{C} \quad (1.13)$$

donde $\Delta\mathbf{Y}_p$ puede considerarse como el vector de predicción de los residuales obtenidos cuando se aplica la corrección $\Delta\mathbf{C}$. El vector $\Delta\mathbf{Y}_c$ representa los residuos actuales obtenidos al sustituir $\Delta\mathbf{C}$ en la ec. (1.11). La matriz \mathbf{A} contiene las derivadas parciales del modelo, evaluadas con las estimaciones de \mathbf{C} actuales, \mathbf{C}_c (ec. 1.15). Con lo anterior, es posible obtener la expresión para resolver la corrección $\Delta\mathbf{C}$ [11]:

$$\Delta\mathbf{C} = [\mathbf{A}^T \mathbf{W} \mathbf{A}]^{-1} [\mathbf{A}^T \mathbf{W} \Delta\mathbf{Y}_c] \quad (1.14)$$

donde,

$$\mathbf{A} = \begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 \\ & & & \vdots & & & & \\ x_m & y_m & z_m & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_m & y_m & z_m & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial b_x}{\partial C_1} & \dots & \frac{\partial b_x}{\partial C_6} \\ \vdots & \ddots & \vdots \\ \frac{\partial b_y}{\partial C_1} & \dots & \frac{\partial b_y}{\partial C_6} \end{bmatrix} \quad (1.15)$$

En [9], los autores presentan un enfoque para realizar el producto anterior basado en matrices de momentos, mediante el cual los datos obtenidos son “comprimidos” dentro de cada una de estas matrices, de tal manera que la cantidad de memoria requerida para este proceso no es afectada por el número de mediciones, reduciendo así la carga computacional que conlleva.

Un conjunto actualizado de parámetros de visión se obtiene al sumar al conjunto actual \mathbf{C}_c el valor de $\Delta\mathbf{C}$ obtenido mediante la ecuación (1.14). El proceso iterativo continúa con el conjunto actualizado hasta que se obtiene la convergencia.

El vector de parámetros de visión \mathbf{C} es estimado inicialmente utilizando un conjunto de posiciones predeterminadas que constituyen el muestreo de posiciones dentro del espacio de cada una de las cámaras y la configuración interna del robot correspondiente. La secuencia de posiciones

en la *trayectoria preplanada* no está relacionada con la tarea que se pretende realizar mediante el manipulador. En cada posición, son tomadas muestras del vector de ángulos del robot junto con las imágenes de cada cámara donde se observan las marcas luminosas colocadas en la herramienta. Estos son los datos utilizados para inicializar los parámetros de visión. Más adelante se verá como, con el fin de actualizar los parámetros C , más muestras son obtenidas durante el desempeño de una tarea.

Cabe señalar que durante la primera estimación de los parámetros C , y debido a que no se tiene ninguna información previa, solo una de las marcas luminosas debe de estar encendida, ya que al haber solo una *marca de referencia* se puede tener la certeza de conocer su posición en el espacio tridimensional. Una vez que se tiene una primera aproximación de los parámetros, es posible considerar una imagen con más de una marca. En el siguiente capítulo se presenta la técnica utilizada para identificar la posición espacial de varias marcas dentro de una imagen.

1.4.6 Estimación de parámetros C mediante el procedimiento de Flattening⁸

El modelo ortográfico de cámara, al no ser mas que una simplificación del modelo de orificio de cámara, implica algunas inconsistencias que llegan a inducir errores de magnitud considerable en el desempeño de las tareas de posicionamiento. La principal fuente de este error es el hecho de considerar, durante la simplificación del primer modelo, el límite asintótico obtenido cuando Z_0 es mucho mayor que el resto de las cantidades x_i, y_i, z_i, X_0 y Y_0 , es decir, cuando $Z_0 \rightarrow \infty$. Una de las consecuencias de este concepto es la deformación de los elementos de una imagen descritos por el modelo. Como puede apreciarse en la Figura 1.5, al aumentar la distancia a lo largo del eje Z_0 , el elemento de perspectiva se reduce drásticamente. En el modelo de cámara de orificio, al colocar dos objetos de la misma altura, de tal manera que uno de ellos se encuentre más cerca de la cámara, es fácil imaginar por qué, en la imagen, el objeto que se encuentra más cerca, se apreciará de mayor tamaño. Por otro lado, el modelo ortográfico de cámara anula completamente este efecto, haciéndolos parecer, dentro de su representación, del mismo tamaño. Es por tanto necesario implementar una técnica para adaptar las muestras que ve la cámara al modelo que las representa.



Figura 1.5: Error en el modelo ortográfico inducido por $Z_0 \rightarrow \infty$.

El concepto que respalda el procedimiento de *flattening* es modificar las muestras tomadas desde el espacio de cámara, de tal manera que sean consistentes con una proyección ortográfica. Esto

⁸Hasta ahora no se ha encontrado una traducción al español que describa satisfactoriamente este procedimiento. La palabra inglesa *flatten* equivale al concepto aplastar, aplanar o alisar, así que *flattening* puede interpretarse como "aplanado o alisado de imágenes", en el sentido de "aplanar la perspectiva". Por comodidad se seguirá usando el término en inglés a lo largo de esta tesis.

se consigue introduciendo una corrección a la información que describe las posición de las marcas luminosas en la herramienta del manipulador, ya que esta información es obtenida de cámaras cuyo modelo es descrito por el modelo de cámara de orificio.

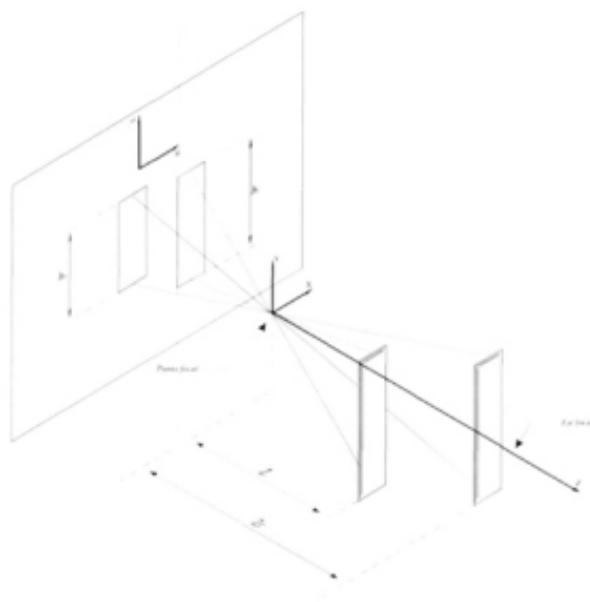


Figura 1.6: Proyección de acuerdo al modelo de cámara de orificio.

Considerando la altura de los dos objetos mostrados en la Figura 1.6 tal y como es proyectada sobre la imagen bidimensional de la cámara, donde h_i es la altura de la imagen correspondiente al objeto más cercano a la cámara y h_r la altura para el segundo objeto; además, Z_i es la distancia medida a lo largo del eje óptico de la cámara, desde el punto focal hasta el primero de los objetos, y Z_r es la distancia desde el punto focal hasta el objeto colocado más lejos de la cámara. A partir de esta representación se pueden utilizar conceptos básicos de trigonometría para determinar que con solo multiplicar la altura h_i por el factor Z_i/Z_r , será obtenido el valor de la altura h_r , lo cual está en concordancia con el modelo ortográfico de cámara. Éste es el concepto básico de *flattening*. De la misma manera, las muestras modificadas (x_{f_i}, y_{f_i}) se obtienen a partir de las muestras de la imagen la siguiente manera:

$$x_{f_i} = x_{ci} \frac{Z_i}{Z_r} \quad (1.16a)$$

$$y_{f_i} = y_{ci} \frac{Z_i}{Z_r} \quad (1.16b)$$

donde x_c , y_c son las coordenadas de un punto en la imagen, Z_i es la coordenada z en el sistema tridimensional unido a la cámara y Z_r es la coordenada z de un punto de referencia cercano al punto en el que se desea aplicar flattening. Durante la ejecución de la maniobra de posicionamiento, Z_r será determinada cuando se tenga una buena estimación de los parámetros de visión para cada cámara, cerca del final de la maniobra.

El procedimiento de *flattening* requiere además de un conocimiento aproximado de Z_0 , el cual, de acuerdo a la Figura 1.4, representa la distancia desde el origen del sistema coordenado unido a la cámara al origen del sistema coordenado desde el que se mide la posición física (x_i, y_i, z_i) , medida a lo largo del eje óptico de la cámara.

Las ecs. (1.16) definen la localización proporcional a la proyección ortográfica correspondiente de una marca luminosa sobre un plano XY .

Una vez que el vector de parámetros \mathbf{C} es estimado mediante la minimización de (1.7), se puede hacer una aproximación de los parámetros de Euler por medio de la siguiente normalización:

$$e_i \approx \frac{C_i}{(C_1^2 + C_2^2 + C_3^2 + C_4^2)^{\frac{1}{2}}} \quad i = 1, \dots, 4 \quad (1.17)$$

Los parámetros de Euler son usados para establecer la matriz que define la orientación del marco de referencia asociado a la cámara (X, Y, Z) relativo al sistema coordenado asociado al manipulador (x, y, z) (ver eq. (1.2)). El origen del sistema coordenado (X, Y, Z) se ubica en el punto focal de la cámara, con X y Y paralelos a los ejes del plano de imagen x_c y y_c , y Z está dirigido a lo largo del eje focal. De esta manera, Z_i en las ecuaciones (1.16) representa la coordenada en el eje Z de la i -ésima marca visual detectada, mientras que Z_r representa la coordenada a lo largo de Z de una marca visual cercana a la posición física donde finalizará la maniobra.

Una relación entre las coordenadas físicas (x_i, y_i, z_i) de la i -ésima marca y sus coordenadas correspondientes (X_i, Y_i, Z_i) relacionadas al sistema coordenado fijo en la cámara se puede obtener de acuerdo a (1.2):

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 & \beta_4 \\ \beta_5 & \beta_6 & \beta_7 & \beta_8 \\ \beta_9 & \beta_{10} & \beta_{11} & \beta_{12} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (1.18)$$

donde cada valor de β_i depende los primeros cuatro parámetros de Euler, así como del valor de X_0 , Y_0 y Z_0 . Sustituyendo la definición de Z_i obtenida de (1.18) en (1.16), las muestras modificadas pueden obtenerse a partir de las siguientes ecuaciones:

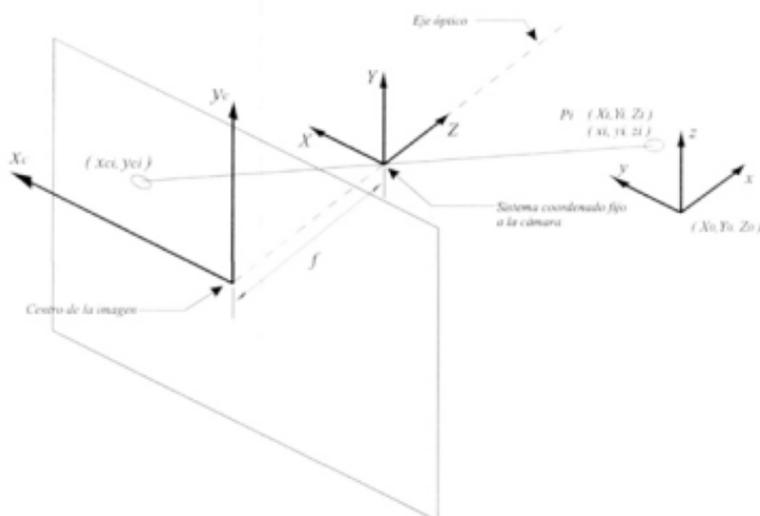


Figura 1.7: Sistemas de referencia relacionados con *flattening*.

$$x_{f_i} = x_{c_i} \left(\frac{\beta_9}{Z_r} x_i + \frac{\beta_{10}}{Z_r} y_i + \frac{\beta_{11}}{Z_r} z_i + \frac{\beta_{12}}{Z_r} \right) \quad (1.19a)$$

$$y_{f_i} = y_{c_i} \left(\frac{\beta_9}{Z_r} x_i + \frac{\beta_{10}}{Z_r} y_i + \frac{\beta_{11}}{Z_r} z_i + \frac{\beta_{12}}{Z_r} \right) \quad (1.19b)$$

Como puede apreciarse, se requiere también del conocimiento aproximado de Z_0 , el cual representa la distancia desde el origen del sistema coordenado unido a la cámara hasta el origen del sistema coordenado desde el que se mide la posición física (en nuestro caso, el sistema coordenado unido a la base del manipulador). Esta distancia es medida a lo largo del eje óptico de la cámara.

Suponiendo conocidos los parámetros de Euler, el procedimiento para la estimación de los parámetros de visión usando *flattening* es similar al procedimiento iterativo usado para la estimación inicial de estos parámetros, pero considerando que las muestras obtenidas desde las cámaras están dadas por las ecuaciones (1.19). De igual forma se puede llegar a la siguiente ecuación que describe la corrección a una suposición inicial de parámetros C:

$$\Delta C_f = [A^T W A]^{-1} [A^T W \Delta Y_f] \quad (1.20)$$

Estimación de Z_r

Una vez que se tiene una buena estimación de los parámetros de visión, es posible evaluar Z_r cerca del final de la maniobra. Esto con el objetivo de determinar una adecuada configuración del manipulador en el punto donde se termina la maniobra. Un punto específico se selecciona, y su localización física (x_r, y_r, z_r) , referida a un sistema coordenado unido al manipulador, es utilizada para definir Z_r a través de la siguiente relación:

$$Z_r = \beta_0 x_r + \beta_{10} y_r + \beta_{11} z_r + \beta_{12} \quad (1.21)$$

Este punto debe ser ubicado en dos o más cámaras para poder realizar el siguiente procedimiento.

Las coordenadas físicas (x_r, y_r, z_r) del punto de referencia cerca del final de la maniobra puede estimarse al minimizar la siguiente función:

$$\phi = \sum_{i=1}^{n_c} [x_{cr}^i - (b_1^i x_r + b_2^i y_r + b_3^i z_r + b_4^i)]^2 + [y_{cr}^i - (b_5^i x_r + b_6^i y_r + b_7^i z_r + b_8^i)]^2 \quad (1.22)$$

en la que los parámetros b_1^i, \dots, b_8^i están definidos independientemente de acuerdo a (1.9) para cada i -ésima cámara. La sumatoria se realiza hasta n_c , que es el número de cámaras apuntando al mismo punto de referencia en el cuerpo no manipulado.

La condición necesaria para la minimización de (1.9) es:

$$\frac{\partial \phi}{\partial x_r} = 0, \quad \frac{\partial \phi}{\partial y_r} = 0, \quad \frac{\partial \phi}{\partial z_r} = 0 \quad (1.23)$$

Las ecuaciones (1.23) se pueden representar como la solución para (x_r, y_r, z_r) :

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = \mathbf{A}_r^{-1} \mathbf{V}_r \quad (1.24)$$

donde la matriz no singular \mathbf{A}_r es simétrica⁹ ($\mathbf{A}_r = \mathbf{A}_r^T$) y su definición se presenta a continuación junto con la de \mathbf{V}_r :

$$\mathbf{A}_r = \begin{bmatrix} \Sigma(b_1^2 + b_5^2) & \Sigma(b_1^i b_2^i + b_5^i b_6^i) & \Sigma(b_1^i b_3^i + b_5^i b_7^i) \\ & \Sigma(b_2^2 + b_6^2) & \Sigma(b_2^i b_3^i + b_6^i b_7^i) \\ & & \Sigma(b_3^2 + b_7^2) \end{bmatrix} \quad (1.25a)$$

$$\mathbf{V}_r = \begin{bmatrix} \Sigma(b_1^i (x_{cr}^i - b_4^i) + b_5^i (y_{cr}^i - b_8^i)) \\ \Sigma(b_2^i (x_{cr}^i - b_4^i) + b_6^i (y_{cr}^i - b_8^i)) \\ \Sigma(b_3^i (x_{cr}^i - b_4^i) + b_7^i (y_{cr}^i - b_8^i)) \end{bmatrix} \quad (1.25b)$$

⁹En adelante, a menos que se especifique lo contrario, las matrices representadas de la misma forma que en la ecuación (1.25a) son diagonales. Cualquier definición estricta será omitida.

Cada una de las sumatorias que conforman las matrices (1.25) son evaluadas desde 1 hasta n_c , donde n_c representa el número de cámaras incluidas en el sistema de visión. Una vez obtenido el valor de $\{x_r, y_r, z_r\}$, el valor de Z_r se evalúa a través de la ecuación (1.21).

1.4.7 Estimación de la configuración interna del manipulador

Una vez elegida la posición sobre la superficie estacionaria que se desea alcanzar con el manipulador, es necesario determinar el valor de cada grado de libertad necesario para conseguirla.

El extremo del manipulador tiene asignado un sistema coordenado por medio del cual es posible definir las características del objeto manipulable instalado y su posición con respecto al sistema coordenado fijo en la base del robot. Por otra parte, la superficie en la cual se ha especificado el punto al que se desea llevar el manipulador posee su propio sistema de referencia, el cual es utilizado para definir la posición del conjunto de puntos objetivo proyectados sobre esta superficie¹⁰.

El objetivo final de una tarea de posicionamiento consiste en hacer coincidir el sistema coordenado en el extremo del manipulador con el sistema de referencia en la superficie estacionaria. Es posible hacer uso del enfoque de *mínimos cuadrados* con el fin de determinar el vector de ángulos que posicione el sistema coordenado en el extremo del robot de tal manera que los puntos objetivos definidos dentro de la superficie de trabajo estén definidos idénticamente dentro del sistema en el manipulador.

La determinación de los ángulos del robot se realiza mediante la minimización de la ecuación (1.26) sobre el vector de ángulos Θ , a partir de un conjunto de n_t muestras provenientes de las n_c cámaras que integran el sistema de visión:

$$\gamma(\Theta) = \sum_{i=1}^{n_c} \sum_{j=1}^{n_t(i)} \left\{ \left[x_{t_j}^i - f_x(r_{x_j}^i(\Theta), r_{y_j}^i(\Theta), r_{z_j}^i(\Theta); \mathbf{C}^i) \right]^2 + \left[y_{t_j}^i - f_y(r_{x_j}^i(\Theta), r_{y_j}^i(\Theta), r_{z_j}^i(\Theta); \mathbf{C}^i) \right]^2 \right\} W_j^i \quad (1.26)$$

para n_c cámaras y $n_t(i)$ puntos objetivo en la i -ésima de las n_c cámaras. Cada j -ésimo punto objetivo de la i -ésima cámara está definido como $(x_{t_j}^i, y_{t_j}^i)$, y su correspondiente posición física está representada por $(r_{x_j}^i, r_{y_j}^i, r_{z_j}^i)$, que depende de la configuración angular del manipulador incluida en el vector Θ , de acuerdo al modelo cinemático del manipulador. Las funciones $f_x(\dots)$ y $f_y(\dots)$ corresponden a la utilización del modelo ortográfico descrito en las ecuaciones (1.8).

En el pasado, se venía utilizando un conjunto de parámetros de visión $\hat{\mathbf{C}}^i = [\hat{C}_1^i, \dots, \hat{C}_6^i]$, relacionados con el objeto no manipulable y obtenidos a través de unas marcas visuales colocadas sobre el mismo [9]. La utilidad de este conjunto de parámetros consiste en poder definir la localización, en el espacio de cámara, de puntos objetivo intermedios en la trayectoria de aproximación hacia el cuerpo no manipulable. Este enfoque se ve seriamente limitado por la necesidad de colocar las marcas visuales sobre un objeto específico y del cual es necesario conocer además su geometría.

¹⁰La matriz de puntos láser mencionada en la introducción. Evita la necesidad de colocar marcas físicas permanentes sobre la superficie de trabajo, haciendo el procedimiento más versátil.

En el siguiente capítulo se presenta en detalle el desarrollo del sistema de posicionamiento basado en CSM, mediante la utilización de un solo conjunto de parámetros de visión. Este nuevo enfoque permite posicionar el robot sobre superficies *arbitrarias* sin la necesidad de colocar ninguna clase de marcas visuales *fijas* ni de conocer más información acerca de ella que la condición de que sea plana, o al menos ligeramente curva.

1.5 Modelo cinemático del manipulador

El desarrollo de un sistema de posicionamiento efectivo de un manipulador implica el conocimiento de un modelo que describa el comportamiento cinemático del mismo. Es necesario poder predecir de manera confiable tanto la posición como la orientación del efector final en términos del valor de los pares cinemáticos (que pueden ser de *revoluta* o *prismáticos*) relacionados con cada uno de los grados de libertad del manipulador.

La cinemática del brazo del robot trata con el estudio analítico de la geometría de su movimiento con respecto a un sistema de coordenadas de referencia fijo, como una función del tiempo, sin considerar las fuerzas/momentos que originan dicho movimiento. Es decir, trata con la descripción del desplazamiento espacial del robot como función del tiempo, en particular las relaciones entre las variables espaciales de tipo articulación y la posición-orientación del efector final.

El robot Fanuc® ArcMate 100i es un manipulador de tipo antropomórfico, que cuenta con seis grados de libertad, todos ellos de tipo revoluta¹¹ accionados por servomotores. Como más adelante se verá, a excepción de las articulaciones 2 y 3, la obtención de las matrices de transformación homogéneas correspondientes se realiza de acuerdo a la representación de Denavit y Hartenberg [7]. Este método es una convención usada para la selección sistemática de los sistemas coordenados asignados a cada eslabón.

En esta convención, cada transformación homogénea \mathbf{A}_i está representada por el producto de cuatro transformaciones básicas:

$$\mathbf{A}_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,\alpha_i} Rot_{x,\alpha_i} \quad (1.27a)$$

$$= \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.27b)$$

Como lo muestra la Figura 1.8, los parámetros a_i , d_i , θ_i y α_i están relacionados con el eslabón i . Al parámetro a se le conoce como *longitud*, α representa un *giro*, d representa la *distancia* o *desplazamiento* y θ representa el *ángulo* de rotación.

¹¹La variable relacionada con cada grado de libertad es una rotación alrededor de un eje del sistema coordenado asociado a la junta.

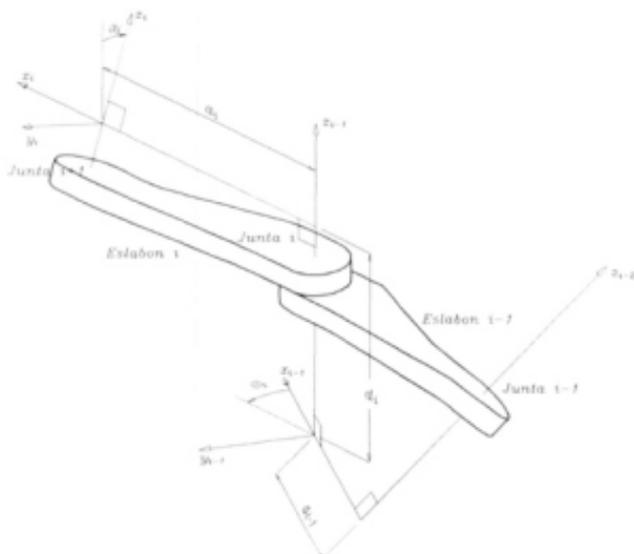


Figura 1.8: Esquema de los parámetros involucrados en el procedimiento de Denavit-Hartenberg.

La matriz \mathbf{A}_i es función de una sola variable y tres de los cuatro parámetros son constantes para un eslabón determinado. En el caso de un grado de libertad de tipo revoluta, θ_i es la variable, mientras que d_i constituye la variable prismática. Siguiendo el procedimiento descrito en [8] se obtiene la Tabla 1.1, que contiene los parámetros correspondientes a cada eslabón del manipulador.

i	$a_i(\text{mm})$	$d_i(\text{mm})$	$\alpha_i(^{\circ})$	θ_i
1	215	0	-90	θ_1
2	*	*	*	*
3	*	*	*	*
4	0	0	90	θ_4
5	0	0	-90	θ_5
6	0	-100	180	θ_6

Tabla 1.1: Parámetros de Denavit-Hartenberg para el modelo cinemático del robot Fanuc[®] ArcMate 100i.

1.5.1 Emulación de la configuración paralelogramo en el robot Fanuc

Los manipuladores denominados articulados se pueden distinguir actualmente en tres subclases [14]:

1. Esféricos.
2. Paralelogramo.
3. Cilíndricos.

Los del subtipo *esférico* son los más comunes entre las configuraciones articuladas. Todas las barras que conforman el manipulador pueden describir un movimiento de rotación respecto a la precedente.

La principal ventaja de este diseño es que el robot puede alcanzar puntos situados en la cercanía de la base, evitando obstáculos que pudieran estar situados dentro del campo de trabajo del mismo. Como se aprecia en la figura la parte superior del brazo está conectada a la parte inferior (antebrazo), el punto de pivote entre ambos elementos se denomina codo y permite el giro del antebrazo. La parte superior del brazo está conectada a la base del robot en un punto denominado hombro, en el cual se produce un giro en un plano paralelo a la base. El espacio de trabajo de estos robots es aproximadamente esférico.

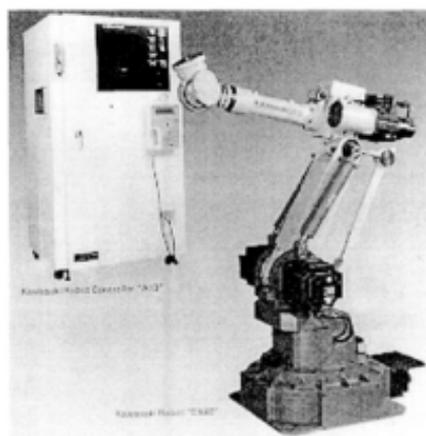


Figura 1.9: Ejemplo de robot articulado configurado en paralelogramo.

En los robots clasificados como *paralelogramo* (Figura 1.9) se reemplaza la parte superior del brazo, constituida en los articulados esféricos por un único sólido rígido, por una cadena cinemática cerrada con forma de paralelogramo. La mayor ventaja de esta configuración es que permite situar los actuadores cerca o incluso en la misma base del robot, lo cual repercute favorablemente en los

pesos e inercias soportados por el manipulador. Así mismo, otra ventaja de esta configuración es que da lugar a robots considerablemente rígidos desde el punto de vista mecánico. Su principal desventaja radica en un espacio de trabajo limitado comparado con el de los robots que responden a la configuración anterior.

La Figura 1.10 muestra un esquema del robot Fanuc utilizado. Como puede apreciarse, la clasificación de la configuración en este robot, de acuerdo a lo mencionado anteriormente es de tipo *esférico*. Esto es cierto desde el punto de vista mecánico, sin embargo, el controlador que se encarga de accionar cada una de las juntas, lleva a cabo los movimientos siguiendo un patrón semejante al de una configuración paralelogramo en los ángulos θ_2 y θ_3 . Es decir, el control del movimiento de estos ángulos emula, desde el punto de vista de software, el comportamiento de un paralelogramo.

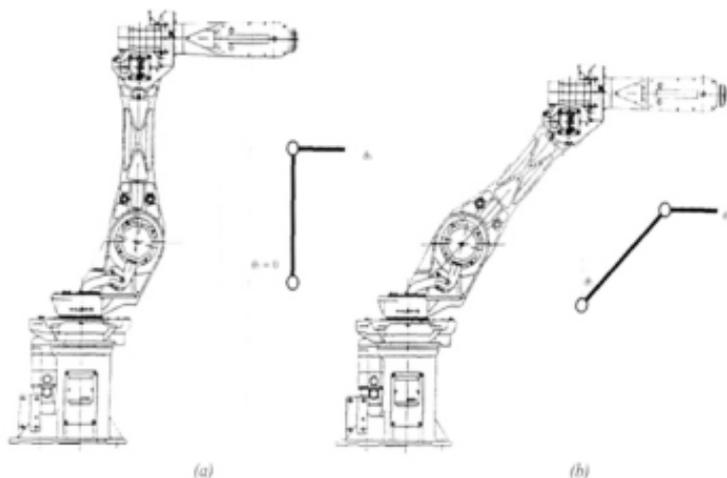


Figura 1.10: Emulación de la configuración paralelogramo en el robot Fanuc®.

En la Figura 1.10 se muestra un esquema simplificado del funcionamiento de las juntas J_2 y J_3 en el Fanuc. La Figura 1.10a muestra una posición inicial para estos ángulos. Al efectuarse una rotación en el ángulo θ_2 (Figura 1.10a), el lazo mecánico que conforma el paralelogramo *imaginario* transmite una rotación de la misma magnitud pero en sentido contrario hacia la junta J_3 . Ya que el extremo del eslabón realiza un movimiento en línea recta proporcional al ángulo de rotación de J_2 , la matriz homogénea \mathbf{A}_2 puede ser modelada como una traslación:

$$\mathbf{A}_2 = \text{Trans}_{x,R \text{ sen } \theta_2} \text{Trans}_{y,R \text{ cos } \theta_2} \text{Rot}_{y,180^\circ}$$

$$= \begin{bmatrix} -1 & 0 & 0 & R \text{ sen } \theta_2 \\ 0 & 1 & 0 & -R \text{ cos } \theta_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

donde R representa la longitud del eslabón en milímetros con un valor de 550 y la rotación alrededor del eje y , al ser de 180° , implica solo un cambio de dirección para los ejes x y z , de acuerdo con el esquema de sistemas coordenados de la Figura 1.11.

Debido a que la orientación del sistema coordenado (X_3, Y_3, Z_3) no está de acuerdo con el modelo Denavit-Hartenberg, la obtención de la matriz \mathbf{A}_3 se realiza directamente mediante el siguiente grupo de rotaciones y traslaciones:

$$\mathbf{A}_3 = \text{Rot}_{z,\theta_3} \text{Trans}_{x,-d} \text{Trans}_{y,a} \text{Rot}_{z,-90^\circ} \text{Rot}_{x,-90^\circ}$$

$$= \begin{bmatrix} \text{sen } \theta_3 & 0 & \text{cos } \theta_3 & a \text{ sen } \theta_3 - d \text{ cos } \theta_3 \\ -\text{cos } \theta_3 & 0 & \text{sen } \theta_3 & -a \text{ cos } \theta_3 - d \text{ sen } \theta_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

donde $a = 100$ y $d = 550$ representan magnitudes en milímetros relacionadas con estructura geométrica del manipulador.

A continuación se muestra el conjunto de matrices de transformación homogéneas obtenidas al aplicar los datos de la Tabla 1.1 en la ecuación (1.27b), además de las matrices \mathbf{A}_2 y \mathbf{A}_3 mencionadas anteriormente. Estas ecuaciones constituyen el modelo cinemático para el robot Fanuc[®] ArcMate 100i:

$$\mathbf{A}_1 = \begin{bmatrix} \text{cos } \theta_1 & 0 & -\text{sen } \theta_1 & 210 \text{ cos } \theta_1 \\ \text{sen } \theta_1 & 0 & \text{cos } \theta_1 & 210 \text{ sen } \theta_1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.28a)$$

$$\mathbf{A}_2 = \begin{bmatrix} -1 & 0 & 0 & 600 \text{ sen } \theta_2 \\ 0 & 1 & 0 & -600 \text{ cos } \theta_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.28b)$$

$$\mathbf{A}_3 = \begin{bmatrix} \text{sen } \theta_3 & 0 & \text{cos } \theta_3 & 100 \text{ sen } \theta_3 - 550 \text{ cos } \theta_3 \\ -\text{cos } \theta_3 & 0 & \text{sen } \theta_3 & -100 \text{ cos } \theta_3 - 550 \text{ sen } \theta_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.28c)$$

$$\mathbf{A}_4 = \begin{bmatrix} \cos \theta_4 & 0 & \text{sen } \theta_4 & 0 \\ \text{sen } \theta_4 & 0 & -\cos \theta_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.28d)$$

$$\mathbf{A}_5 = \begin{bmatrix} \cos \theta_5 & 0 & -\text{sen } \theta_4 & 0 \\ \text{sen } \theta_5 & 0 & \cos \theta_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.28e)$$

$$\mathbf{A}_6 = \begin{bmatrix} \cos \theta_6 & \text{sen } \theta_6 & 0 & 0 \\ \text{sen } \theta_6 & -\cos \theta_6 & 0 & 0 \\ 0 & 0 & -1 & -100 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.28f)$$

En la Figura 1.11 se muestra un esquema del modelo cinemático, en donde los cilindros indican una articulación de tipo revoluta. Los sistemas coordenados $(X_0, Y_0, Z_0), \dots, (X_6, Y_6, Z_6)$ asociados a cada junta están distribuidos a lo largo de la estructura. Note que estos sistemas coordenadas no necesariamente deben de estar colocados junto a la articulación relacionada.

1.5.2 Matriz de transformación homogénea E

Una de las principales características de los robots es su capacidad de ser programados para realizar tareas completamente diferentes con solo cambiar la herramienta instalada en su extremo libre (*face plate*). Esta versatilidad conlleva la necesidad de conocer la posición del extremo de dicha herramienta dentro del espacio de trabajo del manipulador.

Las matrices $\mathbf{A}_1, \dots, \mathbf{A}_6$ describen la posición-orientación del *face plate* referido al sistema coordenado en la base del manipulador (X_0, Y_0, Z_0) . Una forma de conocer la posición del extremo efectivo de la herramienta es mediante la determinación de los parámetros de orientación y posición de su extremo con respecto al último sistema coordenado del manipulador. La matriz que define esta transformación se denomina \mathbf{E} y se puede definir como:

$$\begin{aligned} \mathbf{E} &= \text{Trans}_{z,X} \text{Trans}_{y,Y} \text{Trans}_{z,Z} \text{Rot}_{z,R} \text{Rot}_{y,P} \text{Rot}_{x,W} \\ &= \begin{bmatrix} \cos R \cos P & -\text{sen } R \cos W + \cos R \text{sen } P \text{sen } W & \text{sen } R \text{sen } W + \cos R \text{sen } P \cos W & X \\ \text{sen } R \cos P & \cos R \cos W + \text{sen } R \text{sen } P \text{sen } W & -\cos R \text{sen } W + \text{sen } R \text{sen } P \cos W & Y \\ -\text{sen } P & \cos P \text{sen } W & \cos P \cos W & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (1.29)$$

donde X, Y y Z definen el desplazamiento en cada eje coordenado desde el *face plate* hasta el extremo de la herramienta, mientras que W, P y R corresponden a los ángulos alrededor de cada eje coordenado para alcanzar la orientación del sistema coordenado asignado a dicho efector final¹².

¹²Aunque estos parámetros pueden obtenerse conociendo únicamente las dimensiones mecánicas de la herramienta,

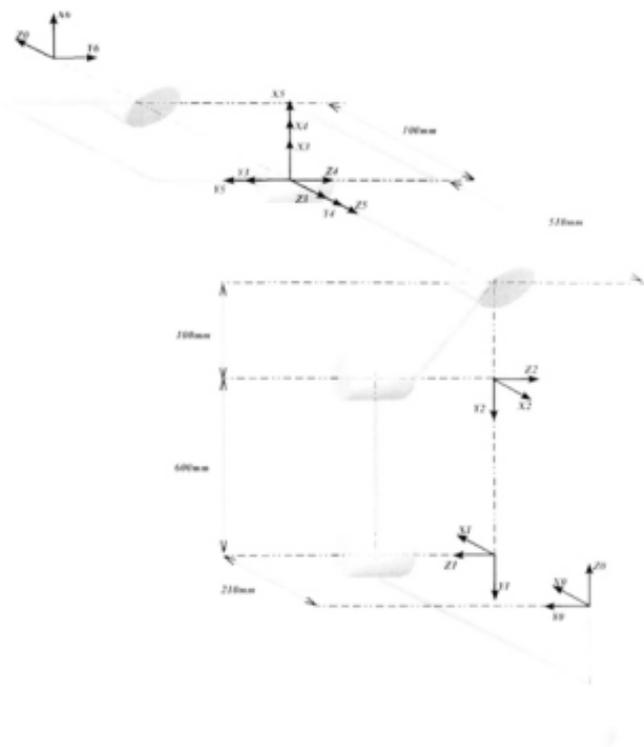


Figura 1.11: Modelo Cinemático del robot Fanuc[®] ArcMate 100i.

De esta forma, la matriz homogénea que transforma las coordenadas de un punto desde un marco de referencia asociado a la herramienta (y definido por \mathbf{E}) al marco de referencia colocado en la base del robot (X_0, Y_0, Z_0) se puede definir como sigue:

$$\mathbf{T}_K(\Theta) = \mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5\mathbf{A}_6\mathbf{E} \quad (1.30)$$

también pueden ser obtenidos fácilmente desde la terminal de enseñanza (*teach-pendant*) del robot, ya que son parte de la configuración general de su sistema interno.

Desarrollo de la metodología de control

Este capítulo presenta las bases para la implementación del método de CSM en un sistema de control¹ de posición de un robot sobre superficies.

En primer lugar se presenta la descripción del dispositivo que provee al manipulador de las marcas visuales necesarias para la estimación de los parámetros de visión junto con el análisis que permite su implementación. El proceso de estimación de la configuración interna del manipulador es desarrollado utilizando las consideraciones presentadas en el capítulo anterior (sección 1.2). A continuación se analiza la utilización de un modelo que relaciona los puntos observados en cada una de las cámaras como un medio para detectar los puntos objetivo proyectados sobre la superficie de trabajo. Estos puntos, proyectados por un haz de luz láser dividido para formar una matriz, son utilizados en el proceso de estimación para determinar la configuración angular del robot asociada a un punto específico sobre la superficie.

Por último, se presenta un panorama general de la implementación del método considerando cada una de las partes contempladas en este capítulo.

2.1 Análisis de la imagen: Detección de centroides

La tarjeta de digitalización de video instalada en la PC provee al sistema de las imágenes provenientes de las cámaras blanco y negro que apuntan hacia el espacio de trabajo del manipulador. Estas imágenes contienen la información necesaria para ejecutar los algoritmos que conforman el método de manipulación de espacio de cámara. Las imágenes obtenidas están formadas por un arreglo de 640×480 números de tipo entero de 8 bits, que representan un valor en la escala de grises de 0 a 255, donde un 0 representa el color negro y 255 el blanco.

El análisis de imágenes requerido por el sistema debe ser capaz de detectar el centroide de las marcas luminosas que están colocadas tanto en la herramienta como en la pieza de trabajo

¹A lo largo de esta tesis, el término *control* será utilizado en el sentido de *control cinemático* [1].

que aparecen dentro del espacio de cámara. Estos requerimientos son satisfechos por la técnica presentada en la sección 1.3.3 y mostrada en la Figura 1.3. El procedimiento de detección de bordes presentado utiliza un umbral de 250^2 , esto significa que las imágenes son *filtradas* para no tomar en cuenta cualquier píxel cuyo valor, en la escala de grises, sea menor a este umbral. La posición (x_c, y_c) de estas marcas es utilizada, junto con su posición en el espacio de trabajo del robot, para realizar la estimación de los parámetros de visión \mathbf{C}_i que describen el modelo ortográfico de cámara.

El procedimiento de detección de manchas luminosas inicia haciendo un barrido de la imagen. Al detectar un píxel con un valor superior o igual al umbral, la secuencia de detección de bordes es ejecutada, al término de la cual, se obtiene un conjunto de coordenadas (x_c, y_c) que corresponden a cada uno de los píxeles que conforman el borde de la mancha. El centro de la mancha es obtenido al promediar cada uno de los valores de x_c y y_c obtenidos.

La Figura 2.1 muestra un ejemplo de la detección de el centroide de algunas marcas luminosas dentro de una imagen. El centroide está representado por un píxel de color negro dentro de la mancha clara.

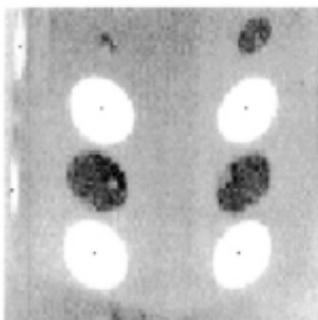


Figura 2.1: Detección de centroides en una imagen.

2.2 Estimación de los parámetros de visión: Diseño de la herramienta

Durante el proceso de estimación de los parámetros de visión es necesario obtener muestras de la posición de un punto en el espacio tridimensional del robot así como de su correspondencia dentro del espacio bidimensional de cada cámara.

El uso de marcas luminosas colocadas sobre el extremo del manipulador facilita las obtención de las muestras requeridas para la determinación del modelo ortográfico de cámara (secciones 1.4.4 y

²Existen varios procedimientos ampliamente utilizados para determinar el valor correspondiente al umbral [10], [7], que está relacionado intrínsecamente con las condiciones de iluminación ambiental. Sin embargo, dada la simplicidad del análisis de imágenes requerido en este trabajo y el color casi completamente blanco de las marcas luminosas, el umbral se ha asignado empíricamente a 250.

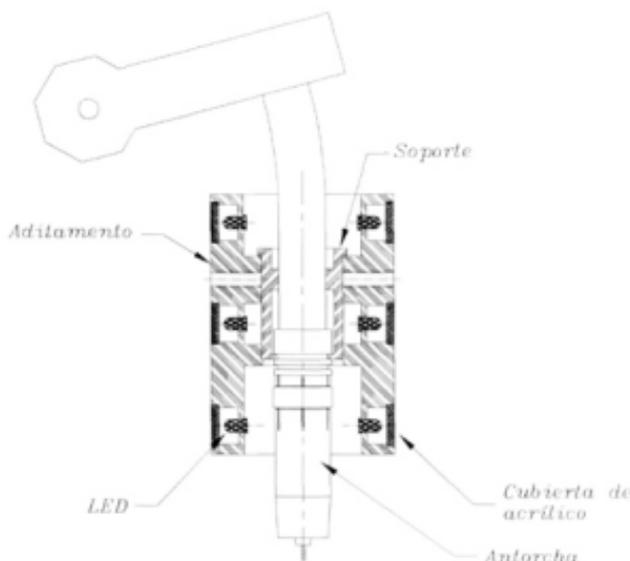


Figura 2.2: Esquema del aditamento utilizado para agregar el conjunto de marcas luminosas a la herramienta.

1.4.5). La detección de las coordenadas dentro del espacio de cámara donde aparecen estas marcas junto con la configuración actual del robot (Θ) y el modelo cinemático que lo describe son utilizados para realizar este proceso.

Con este fin, se ha diseñado un aditamento especial para ser colocado alrededor del tubo de la antorcha de soldadura con que cuenta el robot. Este dispositivo tiene por objeto el proveer al extremo del manipulador de un conjunto de señales luminosas, siendo posible controlar su encendido y apagado desde el puerto serial de una PC.

Esta herramienta consta de 18 leds de alta luminosidad colocados alrededor de un prisma hexagonal fabricado en aluminio, de tal forma que en cada una de las seis caras aparezcan tres de los leds colocados verticalmente. Cada uno de los leds está colocado detrás de una pantalla de acrílico que difunde uniformemente la luz emitida por éstos, facilitando así la detección de su centroide dentro de una imagen. La Figura 2.2 muestra un diagrama de la pieza.

2.2.1 Determinación de la posición de las marcas luminosas

Hasta ahora, el modelo cinemático dado por la ecuación (1.30), proporciona la posición tridimensional de algún punto referido al sistema coordenado asociado a la herramienta. Esta posición depende de la configuración interna actual del manipulador. Con la utilización del dispositivo de la Figura 2.2, es necesario conocer la posición, en el espacio de trabajo del manipulador, de las marcas luminosas que aparecen en éste. La ubicación de los puntos sobre la herramienta está descrita por un vector de posición homogéneo tridimensional referido al sistema (X_E, Y_E, Z_E) cuyo origen coincide con el extremo de la antorcha de la Figura 2.2:

$$\mathbf{F} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ 1 \end{bmatrix} \quad (2.1)$$

donde F_x , F_y y F_z representan las coordenadas tridimensionales de cada uno de las marcas visuales. Estos parámetros son conocidos a partir del diseño de la herramienta, teniendo muy en cuenta la dirección de los ejes coordenados de su sistema de referencia.

En la Figura 2.3 se muestra un esquema de la herramienta junto con su sistema coordenado. Cada led es referido por una coordenada (renglón-columna), de acuerdo a una estructura matricial descrita por los números que aparecen rodeando la estructura. En la Tabla 2.1 se presentan los valores utilizados por cada led dentro del vector \mathbf{F} obtenidos directamente de las dimensiones físicas de la herramienta.

El proceso iterativo para la primera estimación de los parámetros de visión comienza con un valor inicial arbitrario, y por comodidad se asigna $\mathbf{C} = [1, 1, 1, 1, 1, 1]^T$. Durante este proceso el robot ejecuta una trayectoria que cubre completamente el campo de visión de una de las cámaras (*trayectoria preplaneada*) con el fin de obtener las muestras necesarias para la estimación inicial de los parámetros \mathbf{C} .

En esta primera etapa, el conjunto de parámetros inicial, al ser arbitrario, no es capaz de proveer la información necesaria para discernir entre varias marcas luminosas encendidas. El uso de una sola marca luminosa (conocida como *led de referencia*) permite conocer con certeza la correlación entre la configuración angular actual del robot y la posición dentro la imagen de alguna marca que la identifique dentro del espacio de cámara.

El led de referencia es designado por los valores de $LED(1,1)$ (Tabla 2.1), y su posición está descrita por:

$$\mathbf{F}(\Theta)_{1,1} = \mathbf{A}_1 \mathbf{A}_2 \mathbf{A}_3 \mathbf{A}_4 \mathbf{A}_5 \mathbf{A}_6 \mathbf{E} \underbrace{\begin{bmatrix} 36 \\ 0 \\ 124 \\ 1 \end{bmatrix}}_{\mathbf{F}} \quad (2.2)$$

Con esta transformación, son obtenidas las muestras de la posición en el espacio del led encendido para realizar el procedimiento de estimación de parámetros de visión descrito en la sección 1.4.5. El procedimiento es repetido para cada una de las cámaras.

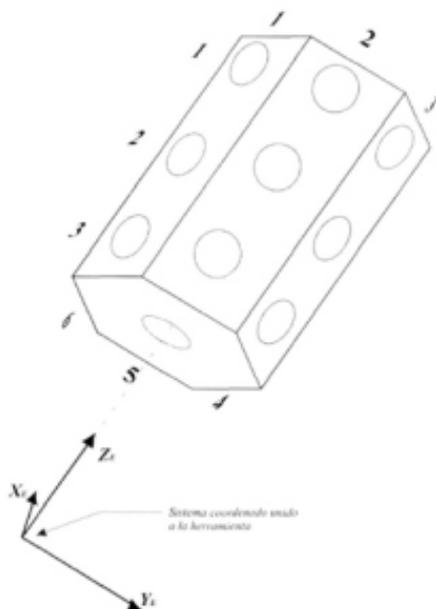


Figura 2.3: Marcas visuales colocadas sobre la herramienta del manipulador.

Una vez realizada la estimación preliminar de parámetros de visión, es posible encender todas las marcas luminosas de la herramienta y realizar por segunda vez la trayectoria preplanificada con el fin de obtener un conjunto denso de muestras. Con la ayuda del modelo descrito por estos parámetros y la transformación F asociada a cada led, se realiza un procedimiento de búsqueda en la imagen que permite asociar la coordenada tridimensional de una marca con su correspondiente posición en la imagen. El procedimiento es descrito por los siguientes pasos:

1. Al estar encendidos todos los leds, en una imagen aparecerán como máximo 9 marcas luminosas visibles. Mediante el vector F , se cuenta con la posición tridimensional de los 18 leds existentes en la herramienta.
2. Cada una de estas posiciones es sometida a la transformación dada por las ecuaciones (1.4) para obtener su posición en el espacio de cada cámara.
3. Esta posición es comparada con cada una de las coordenadas encontradas en la imagen y es seleccionada aquella que tenga el menor error promedio.

LED(r,c)	F_x (mm)	F_y (mm)	F_z (mm)
(1,1)	36.0	0.0	124.0
(1,2)	17.5	30.0	124.0
(1,3)	-17.5	30.0	124.0
(1,4)	-36.0	0.0	124.0
(1,5)	-17.5	-30.0	124.0
(1,6)	17.5	-30.0	124.0
(2,1)	36.0	0.0	85.0
(2,2)	17.5	30.0	85.0
(2,3)	-17.5	30.0	85.0
(2,4)	-36.0	0.0	85.0
(2,5)	-17.5	-30.0	85.0
(2,6)	17.5	30.0	85.0
(3,1)	36.0	0.0	46.0
(3,2)	17.5	30.0	46.0
(3,3)	-17.5	30.0	46.0
(3,4)	-36.0	0.0	46.0
(3,5)	-17.5	-30.0	46.0
(3,6)	17.5	-30.0	46.0

Tabla 2.1: Traslaciones en el sistema coordinado de la herramienta correspondientes a cada LED.

2.2.2 Todas las marcas luminosas encendidas

El procedimiento anterior posee una inconsistencia. Hasta ahora no se ha tomado en cuenta el hecho que en la imagen aparecen solo una parte del total de marcas luminosas (9 como máximo), ya que el resto está oculto en la cara posterior de la herramienta. Como se puede apreciar en la Figura 2.4, al utilizar las 18 posibles coordenadas (previa transformación por medio de la ec. 1.4) para compararlas con las que aparecen en la imagen, existe la posibilidad de considerar marcas que en realidad no son visibles en la imagen bidimensional.

Es necesario poder determinar, en todo momento, cuales son los leds visibles dentro del espacio de cada cámara, para así poder reducir las posibles concordancias con las cuales hacer las comparaciones de las marcas que aparecen en cada imagen.

La asignación de un vector unitario normal \mathbf{V}_k (Figura 2.5) al punto donde se encuentra cada uno de los leds permite la determinación del criterio *está o no está a la vista* de la cámara i . Si se considera el ángulo que forma el eje óptico que sale de la cámara (colineal al eje Z del sistema coordinado unido a la cámara) con cada uno de los vectores unitarios como referencia, la condición necesaria para que la marca luminosa sea visible es que este ángulo sea mayor que 90° .

Debido a que para el caso específico de esta herramienta, tres vectores colocados en una columna (a lo largo del eje Z) son paralelos, y por lo tanto forman el mismo ángulo con el eje óptico de la cámara, no es necesario definir más que seis vectores unitarios, es decir, uno por cara. La definición de estos vectores, realizada utilizando la estructura geométrica de la herramienta, se presenta a

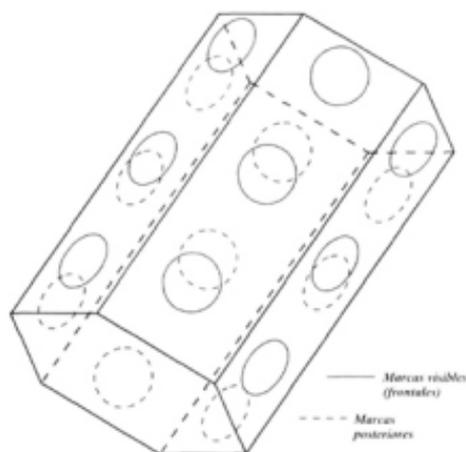


Figura 2.4: Superposición de las marcas luminosas dentro de la imagen bidimensional.

continuación:

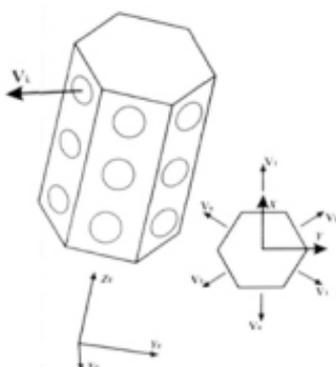
$$\mathbf{V}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{V}_2 = \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{V}_3 = \begin{bmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{V}_4 = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{V}_5 = \begin{bmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{V}_6 = \begin{bmatrix} -\frac{1}{2} \\ -\frac{\sqrt{3}}{2} \\ 0 \\ 0 \end{bmatrix} \quad (2.3)$$

Estos vectores están escritos en términos del sistema coordenado asociado a la herramienta. La siguiente transformación se realiza con el fin de designar cada uno de los vectores con respecto al sistema coordenado asociado a la base del manipulador:

$$\mathbf{V}'_k = \mathbf{T}_K(\Theta) \mathbf{V}_k \quad (2.4)$$

donde \mathbf{V}'_k representa cada uno de los vectores referidos al sistema coordenado en la base del robot y $\mathbf{T}_K(\Theta)$ es el modelo cinemático del manipulador.

Es preciso describir ahora matemáticamente el vector relacionado con el eje óptico de la cámara. Considerando la representación con parámetros de Euler para la transformación desde el sistema

Figura 2.5. Vector V_k asociado a cada uno de los leds

coordenado unido a la base del robot hasta el de la cámara, podemos obtener lo siguiente:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} e_1^2 + e_2^2 - e_3^2 - e_4^2 & 2(e_2e_3 + e_1e_4) & 2(e_2e_4 - e_1e_3) & X_0 \\ 2(e_2e_3 - e_1e_4) & e_1^2 - e_2^2 + e_3^2 - e_4^2 & 2(e_3e_4 + e_1e_2) & Y_0 \\ 2(e_2e_4 + e_1e_3) & 2(e_3e_4 - e_1e_2) & e_1^2 - e_2^2 - e_3^2 + e_4^2 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{R}} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (2.5)$$

donde (x_i, y_i, z_i) son coordenadas referidas a la base del robot y (X, Y, Z) al sistema coordenado en la cámara. Luego, se puede describir una posición en el sistema coordenado XYZ en términos de x_i, y_i, z_i por medio de la siguiente relación:

$$\begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \mathbf{R}^{-1} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.6)$$

A partir de la ecuación (2.6) se puede escribir la definición para el vector unitario asociado al eje óptico Z , el cual es representado por $\vec{\mathbf{EO}}$:

$$\vec{\mathbf{EO}} = \mathbf{R}^{-1} \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} \quad (2.7)$$

El elemento -1 en la ec. (2.7) se puede ilustrar por medio de la Figura 2.6. El eje Z del sistema coordenado en la cámara, debido a la orientación de los ejes (x_c, y_c) en la imagen (y por consecuencia la orientación de los ejes (X, Y)), debe de estar *entrando* hacia la imagen para formar

un sistema coordenado de mano derecha. Por esta razón, la dirección de un vector que sale de la imagen es en el sentido negativo de Z .

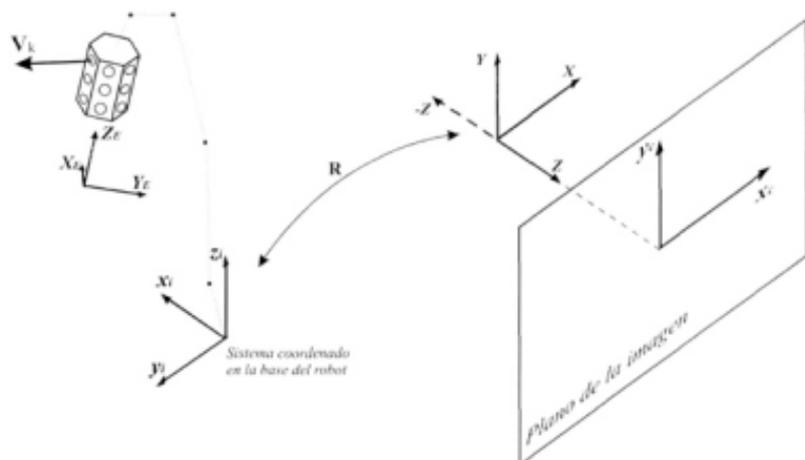


Figura 2.6: Sistemas coordenados (X, Y, Z) , (x_i, y_i, z_i) y (X_E, Y_E, Z_E)

A partir de la ec. (2.7) se deriva la definición del eje óptico de una cámara como:

$$\vec{\mathbf{EO}} = \begin{bmatrix} -2(e_2e_4 + e_1e_3) \\ -2(e_3e_4 + e_1e_2) \\ -e_1^2 + e_2^2 + e_3^2 - e_4^2 \\ 0 \end{bmatrix} \quad (2.8)$$

Una marca visual k , es visible si su vector asociado \mathbf{V}'_k forma un ángulo mayor que 90° con el vector que representa el eje óptico de la cámara, $\vec{\mathbf{EO}}$. El producto punto de dos vectores es negativo si el ángulo entre éstos es mayor que 90° , por lo tanto, una forma de saber si una marca luminosa se encuentra en una posición visible para una de las cámaras es obtener el producto punto entre el vector \mathbf{V}'_k relacionado con la k -ésima marca luminosa y el eje óptico $\vec{\mathbf{EO}}$ asociado a la cámara. Es decir, la condición necesaria para que una marca luminosa esté a la vista de una cámara es:

$$\vec{\mathbf{EO}} \cdot \mathbf{V}'_k < 0 \quad (2.9)$$

Tomando en cuenta esta condición, el paso 2 del procedimiento descrito en la Sección 2.2.1 debe ser renunciado como:

2. Cada uno de los vectores \mathbf{V}_k es sometido al criterio dado por la ecuación (2.9). Si la marca es visible, se procede a evaluar su posición dentro del espacio de cámara (ec. 1.4), de lo contrario se descarta y se continúa directamente con el siguiente led.

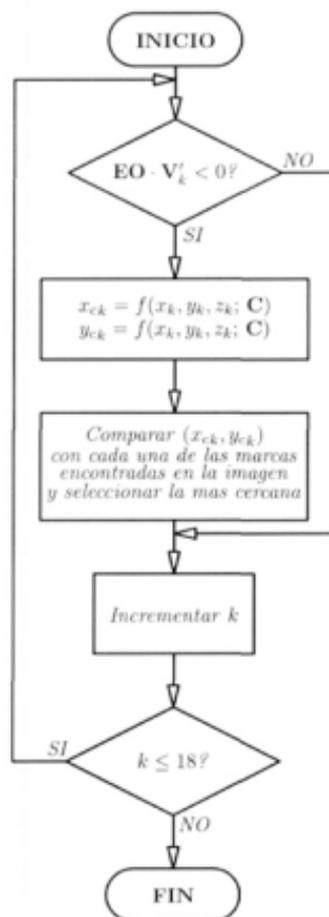


Figura 2.7: Diagrama de flujo para la identificación de múltiples marcas luminosas en la herramienta

Finalmente, el procedimiento para determinar la posición tridimensional de las marcas luminosas que aparecen en una imagen puede resumirse en el diagrama de flujo de la Figura 2.7. Este algoritmo es aplicado a cada una de las 18 posibles marcas luminosas colocadas sobre la herramienta del manipulador, es decir, $k = 1, \dots, 18$.

El encendido de todas las marcas luminosas de la herramienta es utilizado cuando ya se cuenta con una estimación preliminar y completamente válida de los parámetros de visión, con el fin de refinarlos (p. ej. utilizando el método del *flattening* presentado en la Sección 1.4.6) y así obtener un conjunto de parámetros que definan más precisamente el modelo ortográfico de cámara.

2.3 Estimación de la configuración interna del manipulador

Para la minimización de la función (1.26) es necesario conocer la estimación más reciente de parámetros de visión incluidos en el vector \mathbf{C}_i correspondientes a la i -ésima cámara, y el peso relativo asociado a cada punto objetivo es especificado por W_j^i , el cual, para propósitos prácticos, puede ser asignado con un valor de 1.0. Además, un conjunto de puntos objetivo son utilizados como datos de entrada para la obtención de la configuración interna del manipulador requerida para alcanzar una posición deseada en el espacio.

La función $\gamma(\Theta)$ se muestra una vez más a continuación:

$$\gamma(\Theta) = \sum_{i=1}^{n_c} \sum_{j=1}^{n_o(i)} \left\{ \left[x_{r_j}^i - f_x(r_{x_j}^i(\Theta), r_{y_j}^i(\Theta), r_{z_j}^i(\Theta); \mathbf{C}^i) \right]^2 + \left[y_{r_j}^i - f_y(r_{x_j}^i(\Theta), r_{y_j}^i(\Theta), r_{z_j}^i(\Theta); \mathbf{C}^i) \right]^2 \right\} W_j^i \quad (2.10)$$

El criterio de mínimos cuadrados es aplicado en la ecuación (2.10) sobre el vector de ángulos Θ , a partir de un conjunto de n_t muestras provenientes de las n_c cámaras que integran el sistema de visión. Los puntos objetivo $(x_{r_j}^i, y_{r_j}^i)$ que aparecen dentro del espacio de cámara están definidos a través del haz de luz láser dividido para proyectar una matriz de puntos sobre la superficie no manipulable. Estos puntos están referidos aun sistema coordenado unido a tal superficie.

El algoritmo de estimación de la configuración interna del manipulador implica la utilización de un sistema coordenado colocado sobre la superficie de trabajo, y respecto al cual estén definidos los puntos objetivo que intervienen en el actual proceso de estimación. Conceptualmente, el objetivo final de una tarea de posicionamiento, es hacer coincidir un sistema coordenado fijo en la herramienta, descrito con respecto a ésta por una matriz de transformación homogénea \mathbf{T} , con un sistema coordenado sobre la superficie de trabajo (Figura 2.8). La matriz \mathbf{T} aplica una traslación/rotación al sistema coordenado unido a la herramienta. Mediante la modificación de los parámetros de ésta matriz, es posible definir posiciones intermedias que permiten diseñar la trayectoria de aproximación hacia la pieza no manipulable.

Como fue señalado en el capítulo anterior, en trabajos anteriores, además del conjunto de parámetros de visión que relacionan el espacio tridimensional del robot con el espacio bidimensional de cada cámara, se ha utilizado además un conjunto de parámetros \mathbf{C}^i tomado a partir de las marcas visuales fijas colocadas sobre la pieza no manipulable [9]. Debido a las desventajas relacionadas con el uso de este conjunto extra de parámetros de visión, se ha decidido eliminar \mathbf{C}^i , utilizando únicamente los relacionados con la base del robot.

Una de las primeras consecuencias de este enfoque está relacionado con la obtención de la posición tridimensional de los puntos objetivo involucrados en el algoritmo de estimación. Ya que la obtención de estas posiciones es realizada por medio de un algoritmo de estimación basado en el conjunto de parámetros de visión que describe la relación entre el espacio de trabajo del manipulador y el espacio de cámara, dicha posición es obtenida pero con respecto a un sistema coordenado en la base del robot.

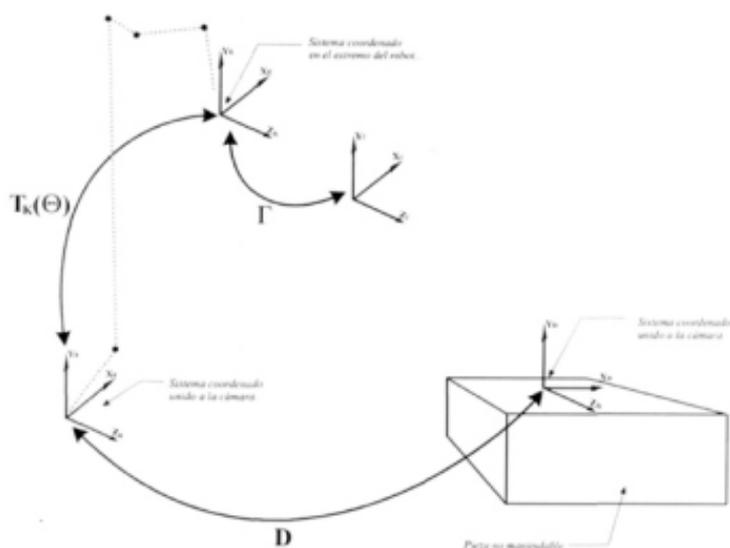


Figura 2.8: Sistemas coordinados descritos por las matrices D y Γ .

Con afán de ser prácticos, el sistema coordinado asociado a la superficie de trabajo, es definido de tal manera que su origen se encuentre exactamente sobre el punto hacia el cual se desea llevar el extremo de la herramienta. La matriz D^i es definida entonces como una traslación simple a partir del sistema coordinado en la base del robot (aunque se puede especificar cualquier rotación que se requiera para inducir una inclinación específica a la herramienta) de la siguiente manera:

$$D^i = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

donde el punto (t_x, t_y, t_z) define las coordenadas a las cuales se desea llevar el manipulador, y a su vez, define también el origen del sistema coordinado asociado a la superficie de trabajo.

En trabajos anteriores, la matriz de transformación D^i , [9] es utilizada para definir la trayectoria de aproximación al introducir una rotación/traslación con respecto al sistema coordinado unido a la pieza no manipulable. Esta matriz se ha *redefinido* con el fin de especificar el sistema coordinado sobre la superficie de trabajo que requiere el algoritmo. De manera semejante, este sistema está definido como una rotación/traslación desde el sistema coordinado unido a la base del robot.

Por otra parte, los puntos objetivo deben estar definidos con respecto al sistema coordinado

dado por \mathbf{D} , y son obtenidos de la siguiente manera:

$$\begin{bmatrix} t_x^j \\ t_y^j \\ t_z^j \\ 1 \end{bmatrix} = \mathbf{D}^{i-1} \begin{bmatrix} t_x^{j,j} \\ t_y^{j,j} \\ t_z^{j,j} \\ 1 \end{bmatrix} \quad (2.12)$$

donde los puntos $(t_x^{j,j}, t_y^{j,j}, t_z^{j,j})$ están definidos con respecto a la base del robot, y (t_x^j, t_y^j, t_z^j) al sistema en la superficie de trabajo.

Con el fin de incluir el sistema coordenado \mathbf{D}^i en el algoritmo de estimación, se puede definir:

$$\begin{bmatrix} t_x^{j,j} \\ t_y^{j,j} \\ t_z^{j,j} \\ 1 \end{bmatrix} = \mathbf{D}^i \begin{bmatrix} t_x^j \\ t_y^j \\ t_z^j \\ 1 \end{bmatrix} \quad (2.13)$$

Las posiciones en el espacio de cámara de cada punto objetivo son evaluadas usando el modelo definido por (1.8):

$$x_t^j = b_1^i t_{x_j}^i + b_2^i t_{y_j}^i + b_3^i t_{z_j}^i + b_4^i \quad (2.14a)$$

$$y_t^j = b_5^i t_{x_j}^i + b_6^i t_{y_j}^i + b_7^i t_{z_j}^i + b_8^i \quad (2.14b)$$

donde b_1^i, \dots, b_8^i representan los coeficientes dados en las ecuaciones (1.8) que dependen de los parámetros incluidos en \mathbf{C}^i para la i -ésima cámara.

Con todo lo anterior, es posible definir los objetivos dentro del espacio de cámara como función de (t_x^j, t_y^j, t_z^j) en forma matricial:

$$\begin{bmatrix} x_{t_j}^j \\ y_{t_j}^j \end{bmatrix} = \begin{bmatrix} t_{x_j}^j & t_{y_j}^j & t_{z_j}^j & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & t_{x_j}^j & t_{y_j}^j & t_{z_j}^j & 1 \end{bmatrix} \begin{bmatrix} \mathbf{D}^{i^T} & \vdots & 0 \\ \dots & \cdot & \dots \\ 0 & \vdots & \mathbf{D}^{i^T} \end{bmatrix} \begin{bmatrix} b_1^i \\ \vdots \\ b_8^i \end{bmatrix} \quad (2.15)$$

Por otra parte, las funciones $f_x(\dots)$ y $f_y(\dots)$, dentro del escalar $\gamma(\Theta)$ (ec. 2.10), que dependen tanto de la información obtenida mediante el modelo cinemático del manipulador $(r_{x_j}^i(\Theta), r_{y_j}^i(\Theta), r_{z_j}^i(\Theta))$, y la versión actualizada de parámetros de visión incluidos en \mathbf{C}^i correspondientes a la i -ésima cámara, pueden ser reescritas como:

$$f_x(r_{x_j}^i(\Theta), r_{y_j}^i(\Theta), r_{z_j}^i(\Theta); \mathbf{C}^i) = b_1^i r_{x_j}^i(\Theta) + b_2^i r_{y_j}^i(\Theta) + b_3^i r_{z_j}^i(\Theta) + b_4^i \quad (2.16a)$$

$$f_y(r_{x_j}^i(\Theta), r_{y_j}^i(\Theta), r_{z_j}^i(\Theta); \mathbf{C}^i) = b_5^i r_{x_j}^i(\Theta) + b_6^i r_{y_j}^i(\Theta) + b_7^i r_{z_j}^i(\Theta) + b_8^i \quad (2.16b)$$

donde $b_1^i, b_2^i, \dots, b_8^i$, están constituidos por los mismos parámetros de visión incluidos en (2.14) para la i -ésima cámara. Las coordenadas definidas por el modelo cinemático utilizadas dentro de

la ecuación (2.16) pueden ser expresadas en términos de los puntos objetivo deseados $(t_{x_j}^i, t_{y_j}^i, t_{z_j}^i)$, usando la expresión dada en (1.30). Así, es posible obtener la siguiente relación para el j -ésimo objetivo en la i -ésima cámara:

$$\begin{aligned} \begin{bmatrix} r_{x_j}^i(\Theta) \\ r_{y_j}^i(\Theta) \\ r_{z_j}^i(\Theta) \\ 1 \end{bmatrix} &= \mathbf{T}_K(\Theta) \mathbf{T}^i \begin{bmatrix} t_{x_j}^i \\ t_{y_j}^i \\ t_{z_j}^i \\ 1 \end{bmatrix} \\ &= \mathbf{T}^i(\Theta) \begin{bmatrix} t_{x_j}^i \\ t_{y_j}^i \\ t_{z_j}^i \\ 1 \end{bmatrix} \end{aligned} \quad (2.17)$$

Sustituyendo la transformación (2.17) en las ecuaciones (2.16) se obtiene la siguiente expresión en forma matricial:

$$\begin{bmatrix} f_x(\dots) \\ f_y(\dots) \end{bmatrix} = \begin{bmatrix} t_{x_j}^i & t_{y_j}^i & t_{z_j}^i & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & t_{x_j}^i & t_{y_j}^i & t_{z_j}^i & 1 \end{bmatrix} \begin{bmatrix} \mathbf{T}^{iT} & \vdots & 0 \\ \dots & \dots & \dots \\ 0 & \vdots & \mathbf{T}^{iT} \end{bmatrix} \begin{bmatrix} b_1^i \\ \vdots \\ b_8^i \end{bmatrix} \quad (2.18)$$

A partir de la ecuación anterior y la expresión para $(x_j^i, y_j^i)^T$ definida en la ec. (2.16) se puede escribir $\gamma(\Theta)$ como sigue:

$$\gamma(\Theta) = \sum_{i=1}^{n_c} \Delta \mathbf{V}^{iT} \begin{bmatrix} \mathbf{M}_T^i & \vdots & 0 \\ \dots & \dots & \dots \\ 0 & \vdots & \mathbf{M}_T^i \end{bmatrix} \Delta \mathbf{V}^i \quad (2.19)$$

para n_c cámaras, donde *matriz de momentos* [9] \mathbf{M}_T^i simétrica y el término $\Delta \mathbf{V}^i$ son definidos a continuación:

$$\mathbf{M}_T^i = \begin{bmatrix} \sum t_{x_j}^{i2} W_j^i & \sum t_{x_j}^i t_{y_j}^i W_j^i & \sum t_{x_j}^i t_{z_j}^i W_j^i & \sum t_{x_j}^i W_j^i \\ & \sum t_{y_j}^{i2} W_j^i & \sum t_{y_j}^i t_{z_j}^i W_j^i & \sum t_{y_j}^i W_j^i \\ & & \sum t_{z_j}^{i2} W_j^i & \sum t_{z_j}^i W_j^i \\ & & & \sum W_j^i \end{bmatrix} \quad (2.20)$$

$$\Delta \mathbf{V}^i = \begin{bmatrix} \mathbf{D}^{iT} & \vdots & 0 \\ \dots & \dots & \dots \\ 0 & \vdots & \mathbf{D}^{iT} \end{bmatrix} \begin{bmatrix} b_1^i \\ \vdots \\ b_8^i \end{bmatrix} - \begin{bmatrix} \mathbf{T}^{iT} & \vdots & 0 \\ \dots & \dots & \dots \\ 0 & \vdots & \mathbf{T}^{iT} \end{bmatrix} \begin{bmatrix} b_1^i \\ \vdots \\ b_8^i \end{bmatrix} \quad (2.21)$$

Las sumatorias incluidas en \mathbf{M}_T son especificadas por el subíndice j , desde 1 hasta el total de puntos objetivo ubicados dentro de cada cámara i (de un total de n_c cámaras).

\mathbf{M}_T^i contiene toda la información requerida para determinar la configuración interna del manipulador. El haz de luz láser proyectado sobre la superficie de trabajo, al estar dividido en una matriz de puntos, constituye un conjunto de puntos objetivo, cuyas coordenadas en el espacio del robot son utilizadas para evaluar \mathbf{M}_T .

Debido a que la matriz de momentos \mathbf{M}_T^i es simétrica, se puede demostrar una simplificación en la cual aparezca la matriz diagonal $\mathbf{M}_T^{d,i}$, como se muestra a continuación:

$$\mathbf{M}_T^i = \mathbf{L}^i \mathbf{M}_T^{d,i} \mathbf{L}^{i^T} \quad (2.22)$$

donde,

$$\mathbf{M}_T^{d,i} = \begin{bmatrix} \lambda_1^i & & 0 \\ & \ddots & \\ 0 & & \lambda_n^i \end{bmatrix} \quad (2.23)$$

Para obtener la matriz \mathbf{L} es posible aplicar el procedimiento de *Jacobi* con la matriz simétrica \mathbf{M}_T , sin embargo, existe un esquema para la determinación de \mathbf{L} que se relaciona más intrínsecamente con la estructura geométrica de \mathbf{M}_T , el cual ha sido utilizado en la implementación práctica del presente desarrollo.

Al peso relativo W_j^i asociado con cada punto objetivo (t_x, t_y, t_z) se le asigna un valor de 1.0. Intuitivamente, se ha considerado que este valor puede ser utilizado para manipular el resultado final de la estimación de la configuración interna. Bajo este enfoque, el peso relativo o ponderación asociada al punto sobre la superficie en el cual se desea posicionar la herramienta se le puede asignar un valor sumamente grande³ (idealmente infinito). Experimentalmente, este procedimiento indujo resultados inesperados, tales como el mal condicionamiento⁴ de las matrices fundamentales en la determinación de la configuración interna, y aunque este resultado no produjo errores en la estimación final, se ha tomado la decisión de eliminar este concepto. El análisis del comportamiento del algoritmo de estimación en función de la modificación del parámetro W excede los alcances y objetivos de esta tesis; su desarrollo será dejado para trabajos posteriores.

La expresión (2.22) puede ser substituida en la función $\gamma(\Theta)$ (ec. 2.10) para escribir una versión simplificada:

$$\gamma(\Theta) = \sum_{i=1}^{n_c} \Delta \mathbf{Y}_M^i{}^T \mathbf{Y}_M^i \quad (2.24)$$

donde,

$$\Delta \mathbf{Y}_M^i = \mathbf{R}_t^i - \mathbf{R}_\theta^i \quad (2.25)$$

³La Sección 2.3.1 puede justificar este enfoque.

⁴El número de condición de una matriz está definido como la razón entre el más grande y el más pequeño de sus valores propios (en magnitud). Una matriz es singular si su número de condición es infinito, y está mal condicionada si su número de condición es muy grande en términos computacionales, esto es, si su recíproco se acerca al límite dado por la precisión de punto flotante de la máquina (10^{-6} para precisión simple y 10^{-12} para precisión doble).

aquí, \mathbf{R}_r^i y \mathbf{R}_θ^i están definidos por:

$$\mathbf{R}_r^i = \begin{bmatrix} \mathbf{M}_T^{d,i\frac{1}{2}} & \vdots & 0 \\ \dots & \dots & \dots \\ 0 & \vdots & \mathbf{M}_T^{d,i\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{D}^{*i\mathcal{T}} & \vdots & 0 \\ \dots & \dots & \dots \\ 0 & \vdots & \mathbf{D}^{*i\mathcal{T}} \end{bmatrix} \begin{bmatrix} b_1^i \\ \vdots \\ b_6^i \end{bmatrix} \quad (2.26)$$

$$\mathbf{R}_\theta^i = \begin{bmatrix} \mathbf{M}_T^{d,i\frac{1}{2}} & \vdots & 0 \\ \dots & \dots & \dots \\ 0 & \vdots & \mathbf{M}_T^{d,i\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{T}^{*i\mathcal{T}} & \vdots & 0 \\ \dots & \dots & \dots \\ 0 & \vdots & \mathbf{T}^{*i\mathcal{T}} \end{bmatrix} \begin{bmatrix} b_1^i \\ \vdots \\ b_6^i \end{bmatrix} \quad (2.27)$$

donde el asterisco representa

$$\mathbf{D}^{*i} = \mathbf{D}^i \mathbf{L}^i, \quad \mathbf{T}^{*i} = \mathbf{T}^i \mathbf{L}^i \quad (2.28)$$

$\mathbf{M}_T^{d,i\frac{1}{2}}$ es una matriz diagonal que contiene la raíz cuadrada de los elementos de la diagonal principal de $\mathbf{M}_T^{d,i}$, es decir:

$$\mathbf{M}_T^{d,i\frac{1}{2}} = \begin{bmatrix} \sqrt{\lambda_1^i} & & 0 \\ & \ddots & \\ 0 & & \sqrt{\lambda_6^i} \end{bmatrix} \quad (2.29)$$

Cabe destacar la utilidad de las matrices \mathbf{D} y $\mathbf{\Gamma}$. Es posible diseñar la trayectoria de aproximación del manipulador hacia un punto destino mediante la modificación de los parámetros de $\mathbf{\Gamma}$, y una vez alcanzada la posición deseada, con sólo desplazar el origen del sistema coordenado descrito por \mathbf{D} , es posible mover el manipulador a lo largo de la superficie de trabajo. Como se mencionaba anteriormente, el objetivo implícito de una tarea de posicionamiento, es hacer coincidir los sistemas coordenados especificados dentro del sistema por \mathbf{D} y $\mathbf{\Gamma}$. Hasta ahora se ha definido el primero sin elementos de rotación (una traslación simple), pero queda claro como es posible asignar una inclinación a la herramienta del manipulador con respecto a la superficie a través de la modificación de esta matriz.

La estimación de la configuración interna del manipulador se realiza mediante la minimización de la función γ (ec. 2.24). Debido a la naturaleza no lineal del modelo cinemático del manipulador, el proceso de minimización es iterativo, comenzando con un valor inicial asignado al vector Θ , es calculado un vector incremento $\Delta\Theta = [\Delta\theta_1, \Delta\theta_2, \dots, \Delta\theta_6]$ consecutivamente, hasta alcanzar la convergencia [11]. $\Delta\Theta$ es obtenido al utilizar la linealización usando series de Taylor del vector de residuos $\Delta\mathbf{Y}_M^i$:

$$\begin{bmatrix} \Delta\mathbf{Y}_M^i \\ \vdots \\ \Delta\mathbf{Y}_M^{n_c} \end{bmatrix}^P \approx \begin{bmatrix} \Delta\mathbf{Y}_M^i \\ \vdots \\ \Delta\mathbf{Y}_M^{n_c} \end{bmatrix}^C - \mathbf{A}_\theta \Delta\Theta \quad (2.30)$$

donde se consideran las n_c cámaras participantes. Los superíndices p y c representan los residuos predichos cuando la corrección $\Delta\Theta$ es aplicada y los residuos actuales, respectivamente. La matriz \mathbf{A}_θ tiene la forma:

$$\mathbf{A}_\theta = \begin{bmatrix} \frac{\partial \mathbf{R}_p^1}{\partial \theta_0} & \cdots & \frac{\partial \mathbf{R}_p^1}{\partial \theta_6} \\ \vdots & \vdots & \vdots \\ \frac{\partial \mathbf{R}_p^{n_c}}{\partial \theta_0} & \cdots & \frac{\partial \mathbf{R}_p^{n_c}}{\partial \theta_6} \end{bmatrix} \quad (2.31)$$

A partir de la condición necesaria⁵ para la minimización de $\gamma(\Theta)$, se puede escribir la solución:

$$\Delta\Theta = [\mathbf{A}_\theta^T \mathbf{A}_\theta]^{-1} \mathbf{A}_\theta^T \begin{bmatrix} \Delta \mathbf{Y}_M^1 \\ \vdots \\ \Delta \mathbf{Y}_M^{n_c} \end{bmatrix}^c \quad (2.32)$$

donde se supone que $[\mathbf{A}_\theta^T \mathbf{A}_\theta]$ no es singular.

El procedimiento para calcular $\Delta\Theta$ es realizado mediante la implementación de la ecuación (2.32) en un algoritmo de cómputo, este algoritmo desarrolla las matrices incluidas dentro de (2.32) en forma de sumatorias [8], reduciendo considerablemente la carga computacional que conlleva su cálculo.

2.3.1 Interpretación geométrica de la matriz \mathbf{M}_T

Es posible definir un significado geométrico relacionado con la diagonalización de la matriz \mathbf{M}_T . La matriz de transformación \mathbf{L} presentada en (2.22) puede ser escrita como [9]:

$$\mathbf{L} = [\mathbf{L}_2^T \mathbf{L}_1^T]^{-1} \quad (2.33)$$

donde

$$\mathbf{L}_1 = \begin{bmatrix} 1 & 0 & 0 & -\tilde{t}_x \\ 0 & 1 & 0 & -\tilde{t}_y \\ 0 & 0 & 1 & -\tilde{t}_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.34a)$$

$$\mathbf{L}_2 = \begin{bmatrix} \left\{ \begin{matrix} v_1 \\ v_2 \\ v_3 \end{matrix} \right\} & \left\{ \begin{matrix} v_2 \\ v_3 \\ v_1 \end{matrix} \right\} & \left\{ \begin{matrix} v_3 \\ v_1 \\ v_2 \end{matrix} \right\} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.34b)$$

⁵Condición necesaria para la minimización: $\nabla_{\Delta\theta} \gamma_p = 0$, donde $\gamma_p = \Delta \mathbf{Y}_M^{pT} W \Delta \mathbf{Y}_M^p$ [11]

aquí, el centro de gravedad del conjunto de n_t puntos objetivo es representado, en la ecuación (2.34a), por $(\bar{t}_x, \bar{t}_y, \bar{t}_z)$ y están definidos como:

$$\bar{t}_x = \frac{\sum_{j=1}^{n_t} t_{xj} W_j}{\sum_{j=1}^{n_t} W_j}, \quad \bar{t}_y = \frac{\sum_{j=1}^{n_t} t_{yj} W_j}{\sum_{j=1}^{n_t} W_j}, \quad \bar{t}_z = \frac{\sum_{j=1}^{n_t} t_{zj} W_j}{\sum_{j=1}^{n_t} W_j} \quad (2.35)$$

y los vectores $\{v_1\}$, $\{v_2\}$ y $\{v_3\}$ de la ecuación (2.34b) son los eigenvectores de una matriz simétrica \mathbf{M}_{cm} . Esta matriz representa los momentos centrales relacionados con el conjunto de puntos objetivo:

$$\mathbf{M}_{cm} = \begin{bmatrix} \sum (t_{xj} - \bar{t}_x)^2 W_j & \sum (t_{xj} - \bar{t}_x)(t_{yj} - \bar{t}_y) W_j & \sum (t_{xj} - \bar{t}_x)(t_{zj} - \bar{t}_z) W_j \\ \sum (t_{yj} - \bar{t}_y)^2 W_j & \sum (t_{yj} - \bar{t}_y)(t_{zj} - \bar{t}_z) W_j & \\ \sum (t_{zj} - \bar{t}_z)^2 W_j & & \end{bmatrix} \quad (2.36)$$

Este procedimiento se puede asociar con la definición de los momentos y ejes principales de un elipsoide de inercia, dentro del contexto de la dinámica de cuerpo rígido. Con base en este concepto y de acuerdo a las ecuaciones (2.35) se planteó la hipótesis (aunque no fue corroborada) de que es posible utilizar el peso W_j asociado al punto objetivo t_j para desviar el centro de masa del elipsoide de inercia hacia este punto. De esta manera, el resultado final de la estimación de la configuración interna puede ser controlado al asignar un peso relativo mayor que el resto de los puntos objetivo al punto sobre la pieza de trabajo en el cual se desea posicionar el manipulador.

Estos datos describen de manera natural la superficie de la pieza no manipulable incluida dentro del proceso de posicionamiento del manipulador.

2.4 Relación entre los puntos objetivo observados por cada cámara

Un sistema de control basado en CSM contempla la utilización de n_c cámaras, de las cuales, una de ellas puede de estar definida como *cámara de observación*. Esta cámara puede tener las mismas funciones que el resto de las que integran el sistema y es utilizada como una interfaz entre el usuario y los algoritmos implementados en una PC. La *cámara de observación* provee al usuario del sistema de un medio de selección del punto sobre la superficie de trabajo en el cual desea posicionar la herramienta del manipulador, para tal efecto, puede estar colocada en una posición más cercana a la superficie de trabajo, con el fin de apreciar claramente los detalles.

Para poder conocer la posición en el espacio tridimensional del manipulador del punto seleccionado, se puede aplicar un algoritmo de estimación semejante al utilizado para encontrar Z_r , presentado en el capítulo anterior (ec. 2.20). Un procedimiento de este tipo requiere conocer la localización del punto en la imagen en al menos dos cámaras, por lo tanto, la selección de un punto en la imagen de la cámara de observación ocasiona la necesidad de conocer la localización de éste dentro la imagen proporcionada por el resto de las cámaras. Como lo muestra la Figura 2.9, los números colocados sobre una superficie de trabajo plana aparecen, en las imagen de cada una de las cámaras, en diferente posición, dependiendo de la orientación de las cámaras.

Es posible establecer un *mapeo* que relacione cualquier coordenada obtenida de la cámara de observación (y proyectada sobre una superficie plana) con su correspondiente posición en el resto

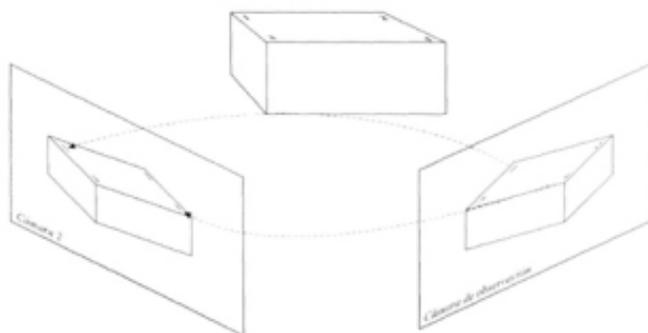


Figura 2.9: Mapeo de los puntos observados por la cámara dos hacia la cámara de observación

de las cámaras, con el fin de poder establecer el número suficiente de muestras para realizar la estimación de la posición tridimensional de este punto.

Si se asume una transformación lineal que relacione que relacione una superficie plana proyectada en dos espacios de cámara diferentes, podemos escribir:

$$X_i = a x_i + b y_i + c \quad (2.37a)$$

$$Y_i = d x_i + e y_i + f \quad (2.37b)$$

donde (X_i, Y_i) representan las coordenadas del primer espacio y (x_i, y_i) las coordenadas del segundo.

Un procedimiento normal de estimación lineal [11] por mínimos cuadrados es seguido al plantear la siguiente relación:

$$\phi = \sum_{i=1}^n [x_i - (a x_i + b y_i + c)]^2 + [y_i - (d x_i + e y_i + f)]^2 \quad (2.38)$$

Aplicando la condición necesaria para minimización se obtiene el siguiente conjunto de ecuaciones:

$$\frac{\partial \phi}{\partial a} = -2 \sum_{i=1}^n (X_i x_i - a x_i^2 - b x_i y_i - c x_i) = 0$$

$$\frac{\partial \phi}{\partial b} = -2 \sum_{i=1}^n (X_i y_i - a x_i y_i - b y_i^2 - c y_i) = 0$$

$$\frac{\partial \phi}{\partial c} = -2 \sum_{i=1}^n (X_i - a x_i - b y_i - c y_i) = 0$$

$$\frac{\partial \phi}{\partial d} = -2 \sum_{i=1}^n (Y_i x_i - d x_i^2 - e x_i y_i - f x_i) = 0$$

$$\frac{\partial \phi}{\partial e} = -2 \sum_{i=1}^n (Y_i y_i - d x_i y_i - e y_i^2 - f y_i) = 0$$

$$\frac{\partial \phi}{\partial f} = -2 \sum_{i=1}^n (Y_i - d x_i - e y_i - f y_i) = 0$$

de donde se puede desprender el siguiente sistema matricial:

$$\begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i & \sum y_i & \sum 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum X_i x_i \\ \sum X_i y_i \\ \sum X_i \end{bmatrix} \quad (2.39a)$$

$$\begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i & \sum y_i & \sum 1 \end{bmatrix} \begin{bmatrix} d \\ e \\ f \end{bmatrix} = \begin{bmatrix} \sum Y_i x_i \\ \sum Y_i y_i \\ \sum Y_i \end{bmatrix} \quad (2.39b)$$

donde cada sumatoria está descrita en $i = 1, \dots, n$ para n muestras.

El número n de muestras puede ser elegido arbitrariamente, pero se puede demostrar que al utilizar $n = 3$, el problema de estimación se reduce a la solución del siguiente sistema lineal:

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \quad (2.40a)$$

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} d \\ e \\ f \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} \quad (2.40b)$$

aquí, son utilizadas únicamente las tres coordenadas conocidas en cada cámara (x_1, y_1) , (x_2, y_2) , (x_3, y_3) y (X_1, Y_1) , (X_2, Y_2) , (X_3, Y_3) .

El modelo planteado por (2.37) permite estimar la posición de un punto, seleccionado en la cámara de observación, en un segundo espacio de cámara, con un conjunto mínimo de muestras obtenidas individualmente para cada cámara.

2.4.1 Identificación de los puntos objetivo sobre la superficie de trabajo

La utilización de una matriz de puntos láser proyectada sobre la superficie de trabajo permite la obtención de un denso número de puntos objetivo, con cuyas coordenadas tridimensionales (t_x, t_y, t_z) es posible completar la matriz de momentos dada por (2.20). En la Figura 2.10 se muestra la proyección del arreglo de 7×7 puntos láser sobre el plano en el cual se desea posicionar el manipulador.

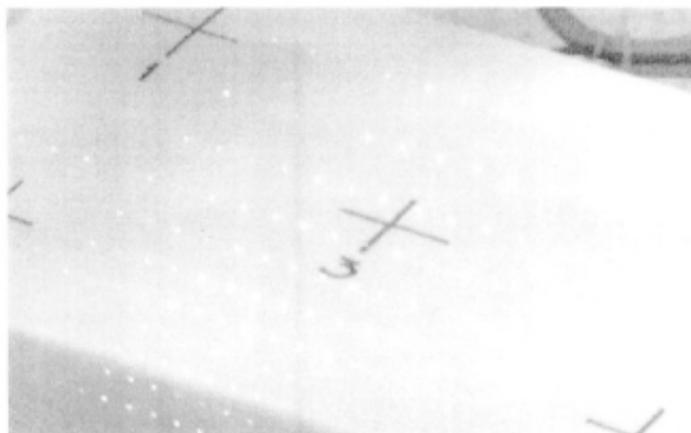


Figura 2.10: Arreglo de 7×7 puntos láser proyectados sobre la superficie de trabajo

En este proceso, la cámara de observación juega un papel importante. Como se mencionaba en el apartado anterior, para conocer la posición tridimensional de un punto (x_c, y_c) observado en una cámara, es necesario conocer su posición en el espacio de al menos otra cámara de las que integran el sistema. Los parámetros de *mapeo de superficies* entre cada una de las cámaras y la cámara de observación, proporcionan la información necesaria para determinar la posición de un punto láser dentro de todas las imágenes de las cámaras participantes, para así poder aplicar el algoritmo descrito por (1.24).

Los puntos láser son detectados dentro de la imagen obtenida de la cámara de observación, y con cada uno de ellos (su coordenada (x_{ct}, y_{ct})) es aplicado el siguiente procedimiento:

1. Se aplica el mapeo descrito por la ecuación (2.37) para obtener la posición estimada del punto en cada cámara.
2. Ya que se conoce la posición de todos los puntos láser *reales* encontrados en cada una de las cámaras y debido a que el procedimiento de mapeo proporciona solamente una aproximación

de éstos, se utiliza un esquema de mínimo error promedio entre el punto estimado y tales puntos reales para conocer su posición verdadera en todas las cámaras. Es decir, el punto láser real más cercano al estimado es seleccionado.

Es posible que no se encuentre una correspondencia lo suficientemente cerca⁶, en este caso, el punto objetivo es descartado para esta cámara, aunque si es encontrada alguna concordancia en otra de las cámaras, será posible establecer la posición tridimensional del punto, ya que como se ha mencionado antes, es necesario localizar el punto en al menos dos imágenes.

3. Con esta información y el conocimiento de los parámetros de visión de cada cámara, es aplicada la ecuación (1.24) para obtener la coordenada (t_x, t_y, t_z) del punto objetivo en cuestión.

El diagrama de flujo de la Figura 2.11 describe el procedimiento para localizar cada punto objetivo encontrado en la cámara de observación.

La selección del punto destino al cual se desea llevar el manipulador es un proceso muy similar al proceso descrito por el diagrama de flujo de la Figura 2.11, a excepción de que la posición seleccionada en la cámara de observación no está relacionada con ningún punto láser, lo cual implica el tener que confiar plenamente en el mapeo de superficies para determinar la localización del punto dentro de la segunda cámara. Una vez establecido el punto destino en las dos cámaras (o en las n_c cámaras que integren el sistema), el procedimiento para establecer su localización dentro del espacio tridimensional se sigue normalmente, para después aplicar el procedimiento descrito brevemente en la Sección 2.3.

2.5 Integración del sistema

Hasta este punto, se cuenta con tres módulos principales, con los cuales es posible construir el sistema propuesto en la introducción de la tesis:

1. **Estimación de los parámetros de visión.** En esta primera etapa se determina el vector de parámetros de visión \mathbf{C} (ecs. 1.4) asociado a cada cámara, el cual satisface el modelo que relaciona el espacio tridimensional del manipulador con el de dos dimensiones de una imagen. Para tal efecto es necesario conocer:
 - *El modelo cinemático del manipulador*, que describe la posición en el espacio tridimensional del extremo de la herramienta.
 - *La posición del led de referencia* con respecto al sistema coordinado unido a la herramienta (sistema coordinado \mathbf{E}), la cual es descrita por el vector \mathbf{L} . Éstas son las muestras, que junto con la posición en el espacio de cámara de la marca luminosa proyectada por el led de referencia, son utilizadas para realizar la minimización de (1.7).
 - El algoritmo de *detección de centroides* utilizado para conocer la posición de las marcas luminosas dentro de una imagen, el cual está basado en la técnica de detección de bordes presentada en la Sección 2.1.

⁶Experimentalmente se ha manejado una distancia mínima de 10 pixels

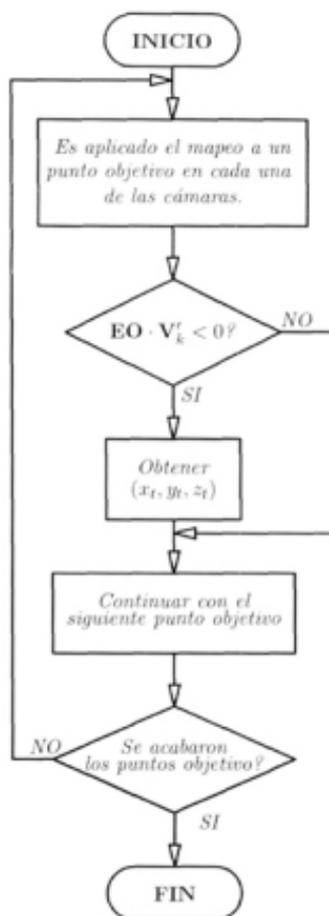


Figura 2.11: Procedimiento para encontrar las coordenadas tridimensionales de los puntos objetivo encontrados en la cámara de observación.

El procedimiento de *flattening* (Sección 1.4.6), aunque es también utilizado para determinar un nuevo conjunto de parámetros de visión, es en realidad aplicado solo con el fin de obtener una mejora en el modelo ortográfico de cámara. Las muestras utilizadas para la aplicación de este modelo son obtenidas durante la trayectoria de aproximación hacia el punto seleccionado, una vez hecha la primera estimación de la configuración interna del manipulador.

2. **Mapeo de superficies** entre espacios de cámara, el cual permite conocer la relación entre los puntos seleccionados en la cámara de observación y el resto de las cámaras. Permite la ubicación de un punto en más de una cámara, con el fin de poder estimar su posición en el espacio. El conjunto de puntos objetivo es obtenido con base en este modelo para realizar la estimación de la configuración interna del manipulador.

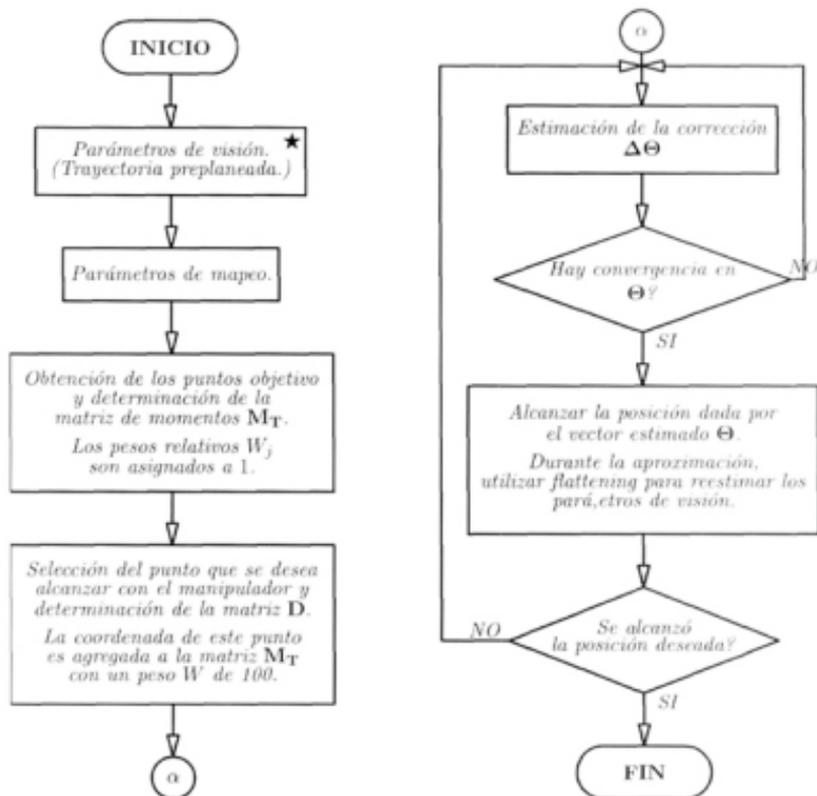
Los parámetros que describen este modelo (ecs. 2.37) son estimados cada vez que se inicia la tarea debido al cambio de la posición de la superficie de trabajo.

3. **Estimación de la configuración interna del manipulador.** La determinación del vector de ángulos del robot constituye la parte nodal del proyecto. Implica el conocimiento de la estimación más reciente de los parámetros de visión, la aplicación de *flattening* (lo cual incluye la descripción de la herramienta presentada en la Sección 2.2.2 al utilizar todas las marcas luminosas encendidas), el modelo cinemático del manipulador y sus derivadas parciales con respecto a cada uno de los ángulos y el mapeo de cámaras utilizado para la determinación de los puntos objetivo.

El procedimiento para la determinación de la configuración interna es descrito en el diagrama de flujo de la Figura 2.12.

Para la realización del sistema de posicionamiento del robot ha sido utilizada una PC como interfaz de control. Los algoritmos aquí descritos, han sido desarrollados en el lenguaje de programación C++ (concretamente Visual C++), bajo el modelo de Programación Orientado a Objetos (POO).

El apéndice A presenta una breve introducción a los conceptos básicos de la POO, así como el desarrollo y la justificación de las funciones que se han utilizado en el proyecto.



★ Este procedimiento se realiza una sola vez al configurar el sistema.

Figura 2.12: Descripción del procedimiento general para la determinación de la configuración interna del manipulador.

Resultados experimentales

Se presentan ahora los resultados obtenidos al implementar experimentalmente las técnicas desarrolladas en el capítulo anterior. La primera estimación de los parámetros de visión es obtenida a partir del diseño de la trayectoria preplaneada del manipulador a lo largo de todo el espacio bidimensional abarcado por cada cámara así como la comprobación experimental de la precisión de los parámetros de visión obtenidos.

Se incluye además el desempeño de una tarea de posicionamiento sobre una superficie plana arbitraria y el error encontrado en el resultado final. Por último, se describe un análisis relacionado con la convergencia del proceso de estimación de la configuración interna del manipulador.

Los experimentos aquí presentados fueron realizados usando el robot Fanuc ArcMate 100i mencionado en la sección 1.5 y la interfaz gráfica que implementa los algoritmos de control es utilizada para agilizar su desempeño.

3.1 Parámetros de visión y trayectoria preplaneada

La estimación preliminar de los parámetros de visión asociados a cada cámara es obtenida a través de la ejecución de una trayectoria preplaneada. Esta trayectoria consiste de una serie de posiciones a las que se mueve el manipulador con el fin de colocar la herramienta dentro de campo de visión de cada cámara. Durante la ejecución de la misma, una (y solo una) de las marcas luminosas colocadas sobre la herramienta es encendida, lo cual proporciona al sistema de un conjunto de muestras $(x_c, y_c; \Theta)$ obtenidas mediante el algoritmo de detección de centroides y el vector de ángulos de la configuración interna proporcionados por el controlador del robot. El conjunto de muestras obtenidas es utilizado para aplicar las solución de la ecuación (1.7) descrita en la sección 1.4.5.

Es importante que la trayectoria preplaneada recorra completamente todo el espacio de cámara, de tal forma que cubra de la mejor manera posible el espacio de trabajo del manipulador. Esto

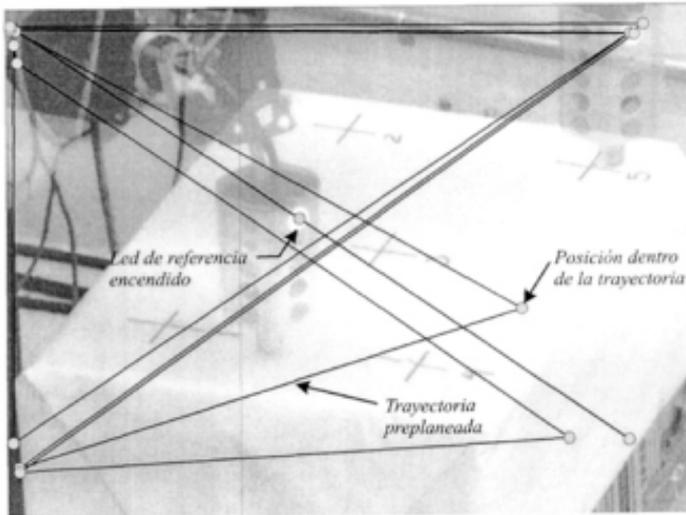


Figura 3.1: Trayectoria preplaneada para la cámara uno.

está relacionado con la validez de la estimación de los parámetros de visión al tomar muestras en una región amplia tanto del espacio de cámara como del espacio de juntas del manipulador.

La Figura 3.1 muestra la trayectoria utilizada para la obtención de los parámetros C de la cámara uno. En ella se aprecia la herramienta utilizando el led de referencia encendido en tres posiciones a lo largo de la trayectoria. Durante este procedimiento se han obtenido un promedio de 140 muestras, las cuales satisfacen ampliamente los requerimientos del algoritmo de estimación de los parámetros de visión. Como fue mencionado en el desarrollo de este procedimiento, debido a la naturaleza no lineal del modelo ortográfico de cámara, los parámetros C son obtenidos iterativamente. Experimentalmente, la estimación de los parámetros de visión se ha realizado en alrededor de 14 iteraciones.

La Tabla 3.1 muestra la primera estimación de los parámetros de visión, obtenida después de aplicar una trayectoria preplaneada a cada una de las cámaras de video utilizadas dentro del sistema de visión.

Este conjunto de parámetros define satisfactoriamente el modelo ortográfico de cámara y aunque es una estimación preliminar, son suficientes para realizar una primera estimación de la configuración interna del manipulador.

C	Cam 1	Cam 2
C_1	0.977967	0.488037
C_2	0.575753	0.311791
C_3	0.129628	0.446596
C_4	0.198820	0.842691
C_5	-1493.09	683.842
C_6	237.234	628.234

Tabla 3.1: Parámetros de visión experimentales obtenidos para las cámaras uno y dos.

Como se mencionó en capítulos anteriores, el modelo ortográfico de cámara, al ser una aproximación al modelo de cámara de orificio, no describe exactamente la relación entre el espacio tridimensional del manipulador y el espacio bidimensional de la cámara. Este error se puede mostrar fácilmente al aplicar a las ecuaciones (1.4) algunas de las posiciones tridimensionales que conforman la trayectoria preplaneada. El resultado obtenido describe la posición estimada dentro del espacio de cámara asociado a dicha coordenada tridimensional. La diferencia entre la posición real y la posición estimada por los parámetros de visión (error en pixels) dentro de el espacio de la imagen muestra el error introducido por el modelo. Experimentalmente, este error ha tenido un valor promedio de 15 píxeles.

El error en la estimación de la configuración interna del manipulador debido a esta inconsistencia en el modelo ortográfico de cámara es minimizado en gran medida por medio de el uso del procedimiento de corrección mediante *flattening* (sección 1.4.6) aplicado durante el desarrollo de una tarea de posicionamiento. Es deseable además, el uso de un gran número de muestras tomadas durante la estimación de los parámetros de visión asociados a cada una de las cámaras.

3.2 Primera aproximación

La estimación de la configuración interna del manipulador es realizada a través de una interfaz gráfica¹ que permite la selección del punto sobre la superficie de trabajo al cual se desea llevar el extremo del manipulador. La coordenada (x_c, y_c) , seleccionada dentro de la cámara de observación, es utilizada junto con el mapeo y el vector de parámetros de visión de cada cámara para obtener su posición en el espacio de trabajo del manipulador. Posteriormente se aplica el procedimiento del capítulo anterior con el fin de obtener el vector de ángulos Θ (configuración interna del manipulador) que hagan coincidir el extremo del manipulador sobre el punto inicialmente seleccionado.

Una vez estimado el vector Θ , el manipulador es dirigido hacia esta posición. Durante la aproximación de la herramienta, son tomadas nuevas muestras $(x_c, y_c; \Theta)$, con el fin de refinar los parámetros de visión y ajustarlos al punto seleccionado de la imagen. Esta nueva estimación de parámetros C es utilizada para realizar una segunda aproximación de la configuración interna del manipulador, la cual estará aún mas cerca de la coordenada seleccionada sobre la superficie.

¹En el apéndice A se describe brevemente el desarrollo, características y ventajas de esta interfaz, así como la implementación de los algoritmos de estimación.

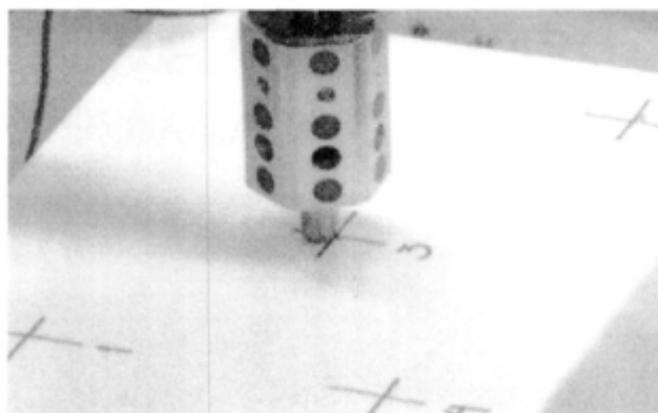


Figura 3.2: Primera aproximación dentro del proceso de estimación de la configuración interna.

Experimentalmente se observó que la posición seleccionada en la cámara de observación es alcanzada en alrededor de 3 estimaciones. La primera de estas estimaciones, es obtenida en aproximadamente 15 iteraciones², mientras que en las subsiguientes estimaciones este número se reduce drásticamente hasta llegar a solo dos iteraciones. El diagrama de flujo de la Figura 2.12 muestra el procedimiento utilizado para alcanzar la posición deseada del manipulador en más de una aproximación.

Θ	1	2	3
θ_1	1.814	2.007	2.027
θ_2	30.910	31.553	31.559
θ_3	-52.061	-21.594	-21.581
θ_4	0.408	0.230	0.212
θ_5	77.766	77.298	77.285
θ_6	-0.674	-0.555	0.542

Tabla 3.2: Estimación de la configuración interna Θ en tres aproximaciones. Los valores están en grados ($^{\circ}$).

La Tabla 3.2 muestra las tres aproximaciones realizadas para la estimación de la posición del manipulador sobre el punto marcado con el número tres en la superficie que aparece en la Figura 3.2, en la cual se aprecia la primera aproximación; en este caso, el error es notable a simple vista. La Figura 3.3 muestra las segunda y tercera aproximaciones respectivamente. En el apéndice C

²El número de correcciones $\Delta\Theta$ (ec. 2.32) realizadas para obtener la convergencia del vector Θ .

aparecen algunas fotografías extra del desarrollo de una tarea de posicionamiento del manipulador.

3.2.1 Segunda aproximación: Corrección mediante flattening

Como se ha mencionado, durante el posicionamiento del manipulador en el vector Θ estimado, es obtenido un nuevo conjunto de muestras $(x_c, y_c; \Theta)$, con el fin de aplicar las ecuaciones enumeradas en la sección 1.4.6 para el refinamiento de los parámetros de visión utilizando la técnica de *flattening*. En la Tabla (3.3) aparecen los parámetros de visión estimados durante la primera, segunda y tercera aproximación del experimento descrito en esta sección.

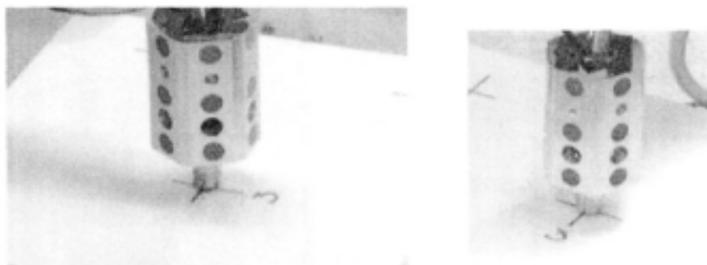


Figura 3.3: Segunda y tercera aproximación.

Θ	2	3
<i>Cámara 1</i>		
C_1	0.958	0.958
C_2	0.564	0.565
C_3	0.112	0.113
C_4	0.171	0.171
C_5	-1461.113	-1460.348
C_6	195.170	145.349
<i>Cámara 2</i>		
C_1	0.494	0.493
C_2	0.321	0.320
C_3	0.439	0.439
C_4	0.799	0.800
C_5	571.602	576.303
C_6	586.549	586.583

Tabla 3.3: Corrección de los parámetros de visión al usar *flattening* durante las aproximaciones 2 y 3.

3.2.2 Error en el posicionamiento de la herramienta

Dentro de las pruebas de posicionamiento realizadas, se encontró un error al determinar la configuración interna del manipulador para un punto especificado sobre la superficie de trabajo. La Tabla (3.4) muestra un conjunto de coordenadas seleccionadas en la imagen de la cámara de observación para ser alcanzadas por el manipulador.

(x_c, y_c)	$x_i(\text{mm})$	$y_i(\text{mm})$	$z_i(\text{mm})$	ERR(mm)	$\Delta(\text{mm})$
(-166, -48)	1367.681	-61.421	45.125	10.0	11.894
	1379.299	-62.405	45.717	0.5	
	1379.539	-62.135	45.717	0.5	
(71, 46)	1141.273	19.511	46.373	17.0	14.684
	1133.284	31.025	43.724	5.0	
	1132.812	21.245	43.849	3.0	
(-108, 33)	1235.721	-82.935	44.876	3.0	0.782
	1236.224	-83.297	44.992	2.0	
	1236.203	-83.276	45.033	2.0	
(127, -28)	1211.309	127.634	42.545	1.0	2.037
	1212.252	127.680	44.546	1.0	
	*	*	*	*	
(-201, 66)	1233.201	-173.170	45.046	4.0	1.028
	1234.173	-173.507	45.050	3.0	
	*	*	*	*	
(76, -125)	1373.871	167.517	45.421	8.0	2.504
	1375.507	166.335	46.903	6.0	
	*	*	*	*	
(-10, -68)	1334.912	62.905	46.232	2.0	0.314
	1335.003	62.643	46.446	1.5	
	*	*	*	*	

Tabla 3.4: Resultados experimentales. La columna **ERR** representa el error medido de la aproximación con respecto al punto seleccionado. Δ es la máxima corrección real aplicada.

Experimentalmente, después de tres aproximaciones (incluida la corrección de parámetros por *flattening* en cada una de ellas), el manipulador alcanzó su mínimo error, es decir, las siguientes estimaciones no producen una corrección significativa en el posicionamiento del manipulador. Son estas tres aproximaciones las presentadas en la Tabla 3.4. En algunos de los casos la tercera aproximación no representó una variación medible, este caso es representado en la tabla por el símbolo “*”.

Como puede apreciarse, el error más grande se presenta al hacer la primera aproximación, teniendo este un promedio de 7 mm. Esta desviación es causada por el conjunto de parámetros de visión, que al no estar estimados mediante *flattening*, no describen correctamente el modelo

ortográfico de cámara. Durante la trayectoria de aproximación, este error es corregido, como ya se ha mencionado, para lograr un error promedio de 2.5 mm. Este procedimiento es aplicado sucesivamente hasta obtener errores del orden de 1 mm.

Las variaciones tan grandes encontradas en el error de la primera aproximación de la Tabla 3.4 son debidas a que todas las estimaciones incluidas en ésta fueron realizadas subsecuentemente, de tal manera que la corrección *flattening* aplicada a la estimación anterior beneficia a la siguiente si el punto destino siguiente se encuentra lo suficientemente cerca.

3.2.3 Mejora de la maniobra de posicionamiento

El procedimiento de aproximaciones no es la mejor opción a seguir al momento de ejecutar una tarea de posicionamiento dentro de un entorno real. Un hipotético usuario del sistema requeriría colocar el manipulador en el lugar seleccionado a través de un solo paso, de tal manera que este procedimiento no sea lento y tedioso.

Un procedimiento a seguir para hacer menos evidente el manejo de tres etapas dentro del proceso de estimación de la posición exacta del manipulador es el de limitar la posición dada por el primer vector Θ estimado a un porcentaje del mismo, de tal manera que el extremo del manipulador no llegue completamente a esta primera aproximación. Subsecuentemente, las siguientes aproximaciones irán aumentando dicho porcentaje, hasta el momento en que se considere que se ha llegado a una configuración relativamente precisa (después de tres estimaciones), la cual ya se puede aplicar completamente o al 100%. El diagrama de flujo de la Figura 3.4 describe el procedimiento de una forma más general.

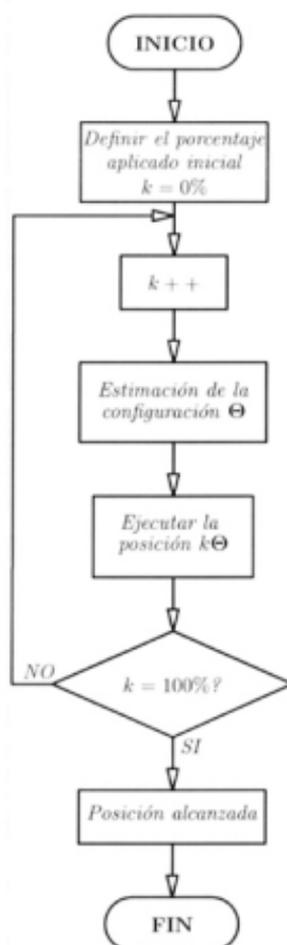


Figura 3.4: Procedimiento de estimación de la configuración interna en un solo paso.

Conclusiones y trabajo futuro

A lo largo de los capítulos anteriores se presentó el diseño e implementación de un sistema de posicionamiento de un manipulador industrial basado en CSM. Los objetivos planteados inicialmente han sido ampliamente alcanzados, dentro de los cuales se pueden enumerar los siguientes:

1. Se ha implementado con éxito y de manera experimental el método de CSM al posicionar la herramienta colocada en el extremo de un robot industrial sobre superficies planas cuya posición y orientación dentro del espacio de trabajo del manipulador es arbitraria. Los algoritmos desarrollados permiten seleccionar esta posición, de manera confiable, a través de una imagen proveniente de una cámara de video orientada hacia el espacio de trabajo del manipulador.

El uso de un solo conjunto de parámetros de visión asociado a cada cámara relacionado al sistema coordinado fijo en la base del manipulador, evita la necesidad de conocer con detalle la descripción geométrica del objeto no manipulable (que en este caso particular se trata de una superficie plana y arbitraria).

2. Queda eliminado completamente el uso de marcas visuales físicas sobre el objeto no manipulable. El uso de puntos láser permite al sistema alcanzar una mayor funcionalidad y versatilidad ante el desempeño de tareas, evitando la necesidad de preparar la superficie de trabajo antes de realizar una tarea de posicionamiento.

El uso de un conjunto puntos láser sobre la superficie de trabajo permite su caracterización de una manera precisa, aumentando la exactitud con la que el algoritmo de estimación de la configuración interna arroja sus resultados. A diferencia de los enfoques utilizados en trabajos anteriores [9] [3] en donde el número de puntos objetivo era establecido previamente, la implementación aquí presentada admite un número indeterminado de éstos.

3. Se han realizado experimentos de posicionamiento del manipulador a través de una interfaz gráfica implementada en una PC. Este programa presenta al usuario la imagen obtenida de la

cámara de observación, en donde puede seleccionar directamente el punto sobre la superficie al cual desea llevar el manipulador además de implementar en su totalidad los algoritmos de estimación basados en CSM.

Lo anterior constituye las bases en las cuales se sustenta el sistema de visión que se encarga de posicionar el extremo del manipulador sobre algún punto seleccionado sobre una superficie plana.

En general, el sistema presenta un comportamiento estable y robusto. A lo largo del desarrollo experimental presentado en el capítulo anterior, la estimación de la configuración interna del manipulador es alcanzada con un error promedio de 1 mm. La convergencia del algoritmo en ningún momento ha producido resultados fuera de un margen aceptable. Esta variación es producida por diversos factores, dentro de los cuales se pueden mencionar los siguientes¹:

- El modelo que describe la relación entre las imágenes obtenidas de cada una de las cámaras y la cámara de observación (sección 2.4) es obtenido a partir de un conjunto de muestras. Cada una de estas muestras define la posición de un punto sobre la pieza no manipulable dentro del espacio bidimensional de las cámaras que integran el sistema de visión.

Las condiciones bajo las cuales fue adquirida esta información pueden verse afectadas o modificadas (por ejemplo, la superficie de trabajo puede ser movida accidentalmente). Bajo estas nuevas condiciones los parámetros que describen el modelo no describen satisfactoriamente la relación entre las superficies observadas por cada una de las cámaras y puede inducir errores al determinar la posición tridimensional de los puntos láser proyectados sobre la superficie de trabajo. Este error puede verse reflejado en la estimación final de la configuración interna del manipulador.

- La densidad de los puntos proyectados por el láser no es uniforme a lo largo de la superficie de trabajo. Experimentalmente se observó que el error es un poco mayor en las áreas donde los puntos láser están más separados uno del otro.
- La especificación del objetivo final de la maniobra dentro de una imagen digitalizada está limitada por el elemento mínimo de la imagen llamado *pixel*. Bajo estas condiciones no es posible seleccionar *directamente*, una coordenada intermedia entre dos pixels contiguos. Es así como un error es inducido al no poder seleccionar exactamente la posición deseada dentro de la imagen.
- La velocidad de la comunicación entre la PC y el controlador del robot induce un retardo al obtener las muestras utilizadas para la estimación de los parámetros de visión de cada cámara. Como se presentó en la sección 1.4.5, la estimación de los parámetros \mathbf{C} requiere de la posición, dentro del espacio de cámara, de las marcas visuales colocadas en el extremo del manipulador además de la posición, dentro del espacio cartesiano del robot de estas mismas marcas visuales. Las características de la comunicación entre el controlador del robot y la PC solo permiten la obtención de un ángulo a la vez (de los seis que constituyen las articulaciones del manipulador). La obtención de cada uno de estos ángulos implica un cierto retardo

¹En la sección 4.2, el trabajo futuro presenta algunas posibles soluciones para estos problemas.

(experimentalmente alrededor de 60 mseg.), de tal manera que el último ángulo obtenido corresponde, en una escala de tiempo, a 360 mseg después de que se inició la obtención de la configuración actual del manipulador.

Lo anterior reduce drásticamente la velocidad máxima utilizada durante el proceso de estimación de los parámetros de visión. Dentro de la implementación de los algoritmos de comunicación presentada en el apéndice A se presenta el procedimiento que permite reducir el error inducido por estas circunstancias.

En resumen, la presente implementación de CSM propone una mejora en el desempeño y funcionalidad del método, haciéndolo mas flexible en el desarrollo de tareas de posicionamiento sobre superficies arbitrarias.

4.1 Aplicaciones

En el comienzo de esta tesis, se hizo mención del método tradicional de programación, que consiste en el uso casi exclusivo de una *terminal de enseñanza* o *teach-pendant*. Este procedimiento implica grabar cada uno de los puntos que componen una trayectoria a lo largo de la cual se debe desplazar el manipulador, así como la necesidad de complicados sistemas de posicionamiento de la pieza no manipulable, de tal manera que se colocada exactamente en el espacio que ocupaba durante la programación de la tarea, estos sistemas deben estar muy bien calibrados y difícilmente se adaptan a nuevas piezas de trabajo. Es debido a esta rigidez que algunos sectores de la industria han optado por evitar los problemas relacionados con este estilo de programación de robots, reduciendo su utilización, en el mejor de los casos, a simples tareas de carga y descarga de máquinas herramienta.

El sistema aquí propuesto permite a un usuario visualizar la pieza de trabajo a través de una terminal de computadora. Tareas como soldadura por arco eléctrico (MIG/TIG) y por punto, pueden ser ejecutadas por un robot sin la necesidad de complicados mecanismos de sujeción de la pieza no manipulable. A través de esta interface, un operador puede ser capaz de diseñar trayectorias para el cordón de soldadura en la unión de dos placas metálicas o seleccionar puntos de aplicación de soldadura en una estructura metálica.

Este sistema se puede migrar fácilmente desde la *teleoperación* hasta la completa automatización utilizando técnicas de visión artificial para la selección de las tareas de posicionamiento, creando de esta manera un sistema *híbrido* de control mediante visión completamente autónomo que, entre otras características, no requiere de la calibración de las cámaras de video ni del modelo cinemático del manipulador.

La exploración espacial constituye otro campo de aplicación importante para la *teleoperación guiada visualmente* basada en CSM. Las grandes distancias a las que son enviados los dispositivos y equipos implicados en esta área de la tecnología hacen prácticamente imposible el uso de técnicas tales como *visual servoing* dado el retardo relacionado con la transmisión de la cantidad de imágenes requeridas por este procedimiento de control. El uso de CSM y la teleoperación permite la utilización de una sola imagen para la especificación de alguna tarea (como puede ser la recolección de una muestra de material localizado en el campo de la imagen en la cámara). La información sobre el punto en la imagen que se desea alcanzar puede ser enviada de regreso para que, a partir de este momento, el sistema realice la tarea de una forma completamente autónoma.

Las aplicaciones del sistema son amplias y variadas. Éstos son solo algunos ejemplos de sus capacidades y alcances. Como ya se ha mencionado, su robustez y relativa inmunidad a la variación de los parámetros relacionados con las cámaras de video y cinemática del manipulador, así como la versatilidad que conlleva hacen de este sistema una buena opción para la implementación de sistemas de visión dentro de diversas áreas de la industria.

4.2 Trabajo futuro

El modelo de control desarrollado en el transcurso de la investigación es aún perfectible. El sistema cuenta con algunas deficiencias ya que algunas secciones del sistema, al no ser esenciales para el desempeño de tareas de posicionamiento, han sido omitidas o, en el mejor de los casos, reducidas a su mínima expresión por falta de tiempo. Se pueden listar los siguientes elementos, que de ser implementados pueden incrementar la eficiencia y optimización del método de posicionamiento:

- El modelo que describe la relación entre las imágenes obtenidas de las cámaras es definido lineal y por lo tanto está limitado al uso de superficies planas. Es necesario el desarrollo de un modelo que describa superficies curvas (orden superior), aunque idealmente este modelo debe ser capaz de relacionar superficies de cualquier característica dentro de la imagen de las cámaras.
- Como fue apuntado en la sección anterior, el movimiento accidental de la superficie de trabajo puede inducir pequeños errores en el modelo que describe la relación lineal entre las imágenes provenientes de dos cámaras (Sección 2.4). Es posible utilizar los puntos proyectados sobre la superficie por la matriz láser para actualizar el modelo dado por (2.37), de la misma manera que son utilizadas las muestras obtenidas durante la trayectoria de aproximación para actualizar los parámetros de visión.
- Es necesaria la optimización de la comunicación entre el robot y el programa de control implementado en la computadora con el fin de agilizar el proceso de obtención de muestras durante la trayectoria de aproximación hacia el punto seleccionado sobre la superficie. El error introducido por la velocidad de la comunicación puede ser reducido en gran medida, para poder, al mismo tiempo, aumentar la velocidad del robot durante la ejecución de las maniobras de posicionamiento, y hacer el proceso más eficiente.
- El uso de más de una aproximación para poder determinar con exactitud la posición final del manipulador en una maniobra no es deseable dentro de un entorno práctico. Es posible realizar estas correcciones gradualmente antes de que el manipulador llegue físicamente a la posición deseada, de tal manera que no sean tan evidentes y no requieran de ninguna intervención interna (ver Sección 3.2.3).
- La interfaz gráfica que implementa los algoritmos de control se encuentra aún en una etapa muy primitiva. El programa debe ser capaz de agregar nuevas cámaras al sistema, diseñar trayectorias preplaneadas para cada una de ellas y hacer cálculos iniciales de los parámetros de visión, definir la cámara de observación, entre otras configuraciones generales.

Es deseable además contar con la característica de *zoom* [15] para la selección del objetivo final de la maniobra, reduciendo con ello el error inducido por la naturaleza digital de las imágenes obtenidas desde las cámaras de video.

Por otra parte, el robot Fanuc®ArcMate 100i cuenta con un software interno² que implementa todas sus funciones básicas así como las funciones especiales de soldadura por arco. Este software es llamado ArcTool y está desarrollado bajo un lenguaje de programación de robots usado por Fanuc llamado Karel. Un proyecto más ambicioso consiste en implementar la metodología de control presentada en esta tesis dentro del CPU del manipulador utilizando este lenguaje de programación, eliminando así la necesidad de establecer una comunicación continua entre el robot y la PC. Dado que los algoritmos de visión utilizados aquí son de muy bajo nivel, los datos requeridos a partir de las imágenes obtenidas de las cámaras podrían ser enviados en respuesta a una señal del robot. Esta sería la única función de la computadora, procesar las imágenes que recibe a través de la tarjeta de digitalización de video y enviar los resultados al programa residente dentro del controlador del robot. Para este caso, los datos consistirían únicamente en la ubicación de las marcas visuales dentro de cada imagen proveniente de las cámaras.

²Los programas incluidos dentro de un sistema en particular son conocidos como *firmware*.

Programación Orientada a Objetos

Esta sección incluye una pequeña introducción a los conceptos básicos de la programación orientada a objetos (POO) particularmente del lenguaje C++. Además, es presentado someramente el desarrollo de la metodología de diseño de los objetos involucrados en una interfaz gráfica para los algoritmos presentados anteriormente junto con su justificación.

Las características de esta pequeña aplicación, que permite interactuar gráficamente con los algoritmos de control, son mostradas al final de este apéndice, haciendo énfasis en la sencillez con que el modelo de POO utilizado puede adaptarse al desarrollo de nuevas interfaces de control y teleoperación.

A.1 Breve introducción a la teoría de objetos

Todos los lenguajes de programación están basados en abstracciones. En cierta forma, la complejidad de los problemas que se pueden resolver por medio de un lenguaje depende del tipo de abstracción de éstos. Por ejemplo, el lenguaje Ensamblador es una pequeña abstracción del lenguaje de máquina que entiende el procesador, y de la misma manera, lenguajes de más alto nivel como BASIC, C, Pascal y Fortran son abstracciones del lenguaje Ensamblador. Estos lenguajes proveen de grandes mejoras sobre Ensamblador, pero su principal abstracción aún requiere de la necesidad de pensar en términos de la estructura de la computadora, en vez de la estructura del problema que se está tratando de resolver.

Este primer modelo de programación se puede dividir en dos secciones principales: *El espacio de la máquina* (o espacio solución del problema), que es el lugar en donde el programador está tratando de modelar el problema, y el *espacio del problema*, que es en sí, el modelo del problema que se está tratando de resolver. El objetivo de un programador bajo estas circunstancias es el de encontrar la asociación entre estos dos espacios. Debido a que el alto esfuerzo requerido para establecer esta relación y el hecho de que éste no es debido a las características propias del

lenguaje, da como resultado programas difíciles de escribir, complicados de interpretar y con un costo de mantenimiento muy elevado.

Una alternativa al modelo de la máquina es el *el modelo del problema* que se está tratando de resolver. Los primeros lenguajes como LISP y APL utilizaban enfoques muy particulares del mundo que estaban tratando de modelar (“todos los problemas son, en última instancia, listas” o “todos los problemas son de naturaleza algorítmica”), PROLOG, en cambio, manejaba cadenas de decisiones. Cada uno de estos puntos de vista constituye una buena solución para la clase de problema en particular que se está tratando de resolver, pero se vuelven ineficientes cuando se salen de este dominio.

El enfoque POO da un paso más allá al proporcionar al programador las herramientas necesarias para representar elementos en el espacio del problema. Esta representación es lo suficientemente general para que el programador no se vea restringido a un tipo de problema en particular. Estos elementos del programa son llamados **objetos**. La idea es que el programa pueda adaptarse por sí mismo a la estructura del problema mediante la adición de nuevos objetos, de tal manera que al estar leyendo el código que describe la solución, se están leyendo *palabras* que también describen el problema. De esta manera, la POO permite describir un problema en los términos del propio problema, en vez de los términos de la computadora donde la solución se va a ejecutar. Aún así, existe una conexión con la computadora: Cada objeto aparece de alguna forma como una pequeña computadora; tiene un *estado*, y tiene un conjunto de operaciones que se le pueden pedir al objeto que realice. Además de esto, como cualquier objeto del mundo real, cuenta con características y comportamientos.

Un enfoque muy puro de la POO contempla las siguientes características que todo objeto debe tener [5]:

1. **Todo es un objeto.** Se puede pensar en un objeto como en una variable comodín. Puede almacenar datos, pero además se le puede pedir que realice operaciones sobre sí mismo. En teoría, se puede tomar cualquier componente conceptual en el problema que se está tratando de resolver (perros, autos, etc.) y representarlo como un objeto dentro de un programa.
2. **Un programa es un conjunto de objetos comunicándose y dándose órdenes entre ellos mediante mensajes.** Para hacer una petición a un objeto es necesario mandarle un mensaje. Concretamente, un mensaje es una petición para llamar a una función que pertenece a un objeto.
3. **Cada objeto tiene su propio espacio de memoria constituida por otros objetos.** Es decir, se puede crear un nuevo objeto haciendo un paquete de objetos que ya existen. De esta forma, se puede establecer una complejidad en un programa *escondiéndola* detrás de la simplicidad de los objetos.
4. **Cada objeto corresponde a un tipo de objeto.** Cada objeto es una *instancia de clase*, donde “clase” es sinónimo de “tipo”.
5. **Los objetos de un tipo en particular reciben los mismos mensajes.** Debido a que un objeto de tipo “círculo” es también un objeto de tipo “figura”, es seguro que un círculo es

capaz de recibir los mensajes dirigidos a alguna figura. Esto significa que es posible escribir un programa que se refiera a las figuras y automáticamente maneje cualquier cosa que concuerde con la descripción de una figura. Esta *habilidad de sustitución* es uno de los conceptos más poderosos en la POO.

A.1.1 Interfaces

Todos los objetos, aunque sean únicos, son también parte de una clase de objetos que tienen características y comportamientos en común (El primer lenguaje orientado a objetos, Simula-67, introdujo este concepto con su palabra clave `class`, que crea un nuevo tipo de objeto dentro de un programa).

Los tipos de datos abstractos (como a los que pertenecen los objetos) funcionan casi exactamente como lo hacen los tipos de datos nativos de los lenguajes: Se pueden crear variables de un tipo (llamados *objetos* o *instancias*) y manipular esas variables (*mandando mensajes* o *peticiones*). Prácticamente todos los lenguajes orientados a objetos utilizan la palabra clave `class`, y es utilizada en el mismo sentido (pero no con el mismo fin) que `typedef` o `struct` del lenguaje C.

Ya que una clase describe un grupo de objetos con características y comportamientos (funciones) idénticos, una clase, al igual que un tipo de dato común (`float`, `int`, etc), es también un tipo de dato. La diferencia consiste en que el programador define una clase para satisfacer las necesidades de un problema específico, en vez de verse forzado a usar un tipo de dato ya existente que fue diseñado para representar una unidad de almacenamiento de la computadora. De esta manera, se extiende el lenguaje de programación nuevos tipos de datos que se adecúan a las necesidades del programador.

Una importante característica de la POO es que una vez que se ha diseñado una clase se pueden crear tantos objetos de esa clase como se desee, para después manipular estos objetos tal y como se tratase de los elementos que constituyen el problema que se está tratando de resolver.

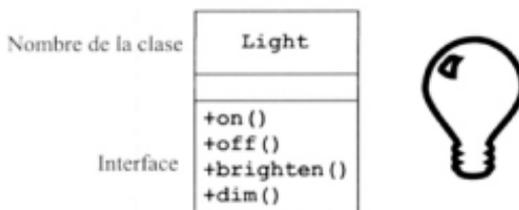
Hasta ahora se ha hablado de la utilidad de los objetos para modelar problemas, pero aún es necesario saber como es que se manejan. Una *interface* es el medio a través del cual se realizan las peticiones a un objeto para que éste haga algo (realizar una transacción, dibujar algo en la pantalla o accionar un interruptor).

La *interface* establece qué peticiones se pueden hacer a un objeto en particular. Sin embargo, debe haber código en algún lugar que realice esta petición. Este código, junto con los datos ocultos constituyen la *implementación*. Una clase contiene una función asociada a cada posible petición, y cuando se realiza una en particular, esa función es llamada. El proceso se conoce como "enviar un mensaje" a un objeto, mientras que el objeto decide que hacer con tal mensaje (ejecuta el código de la implementación).

La Figura A.1 muestra el diagrama UML¹ de una clase que representa el funcionamiento de una lámpara incandescente.

Para este caso, el nombre de la clase/tipo es **Light**. Se puede definir un *objeto* de la clase `Light` y las peticiones que se le pueden hacer son encender (on), apagar (off), incrementar su brillo

¹ *Unified Modeling Language*. En este esquema, una clase es representada por un rectángulo, con el nombre de la clase en la parte superior, los *datos miembro* en la parte central y *funciones miembro* en el inferior. El signo + antes de cada función miembro indica que son públicas.

Figura A.1: Representación UML de la clase `Light`.

(`brighten`) o reducirlo (`dim`). Como lo muestra el siguiente listado, un objeto de la clase `Light` es declarado simplemente asignando un nombre o identificador al objeto (Foco) y precedido del nombre de la clase. Para mandar un mensaje al objeto es utilizado este identificador seguido del nombre del mensaje, conectado a través de un punto.

```
Light Foco;
Foco.on();
```

A.1.2 La implementación oculta

En POO es posible ocultar porciones del código que implementa una clase. Existen muchas razones por las cuales es necesario hacer esto [5], pero fundamentalmente es debido a que un *usuario de clases* (al contrario de los *programadores de clases*) no requiere conocer mas información acerca de una clase que sus interfaces. Es deseable que los procesos internos que realiza un objeto se mantengan intactos para que futuras actualizaciones continúen siendo compatibles con las anteriores implementaciones de la clase. Además, algunos miembros de la clase pueden estar restringidos a cierto rango, y por consecuencia no pueden ser modificados directamente desde el exterior sin ser validados antes por alguna procedimiento extra.

Así que la principal razón para controlar el acceso de los usuarios de las clases a ciertas porciones del código es evitar que utilice elementos que no debe tocar, partes que son necesarias para los mecanismos internos del tipo de dato pero que no forman parte de la interface necesita para resolver sus muy particulares problemas. Esto en realidad es de gran ayuda, ya que el usuario puede ver claramente que es importante para el y que no lo es.

C++ utiliza tres palabras clave para establecer la estructura de una clase: `public`, `private` y `protected`. Estos *especificadores de acceso* determinan quién puede utilizar las definiciones que preceden.

public. Estas definiciones pueden ser accedidas por cualquiera. Es decir, una declaración pública puede ser utilizada tanto desde afuera del objeto (p. ej., ya que la función `Light::on()` es pública se puede acceder escribiendo `Foco.on()`) como desde adentro (p. ej. la función `Light::dim()` puede hacer uso de la función `Light::on()`).

private. A excepción de las funciones miembro que constituyen la clase, estas declaraciones no pueden ser utilizadas desde ninguna otra parte. Si se intenta acceder a un dato o función miembro privado se obtiene un error de compilación.

protected. Actúa exactamente igual que **private**, con la excepción de que una clase *heredada* o *derivada* puede acceder a los datos protegidos pero no a los datos privados. El concepto de herencia será introducido más adelante.

A.1.3 Reutilización de la implementación

Una vez que la clase ha sido creada y probada, representa una útil pieza de código. La reutilización de código constituye una de las más grandes ventajas que los lenguajes de POO proveen.

La forma más sencilla de reutilizar una clase es simplemente usar un objeto de esa clase directamente, pero además se puede colocar este objeto dentro de una nueva clase. Esto es conocido como *crear un objeto miembro*. Una nueva clase puede estar constituida por cualquier número y tipo de otros objetos, dependiendo de las necesidades y funcionalidad deseada. Este concepto es llamado *composición* o *agregación*, ya que se están creando nuevas clases a partir de otras ya existentes. Algunas veces la composición es referida como una relación “tiene-un”, de la misma manera en que un “automóvil *tiene-un* motor”.



Figura A.2: Diagrama UML donde la composición Auto-Motor se indica con un diamante.

La composición presta un alto grado de flexibilidad. Los objetos miembro de una nueva clase comúnmente son declarados privados, haciéndolos inaccesibles para los usuarios de la clase. Esto permite cambiar estos objetos miembro sin perturbar el código externo ya existente. Es posible además cambiar o agregar objetos miembro en tiempo de ejecución, esto es, modificar dinámicamente el comportamiento del programa.

A.1.4 Herencia: Reutilización de la interface

Las ventajas de la POO en cuanto a la facilidad de modelar un problema en sus propios términos se pueden ampliar mediante el concepto de *herencia*. En algunas ocasiones, después de crear una clase para satisfacer determinada circunstancia, puede surgir la necesidad de crear todo un conjunto de clases que presenten un comportamiento muy similar a la primera. En estos casos, es preferible tomar la clase existente y hacer un “clon” de ésta para posteriormente hacer las adiciones y modificaciones necesarias. Esto es efectivamente lo que significa la herencia, con la excepción de que si se modifica la clase original (también llamada *clase base*, *superclase* o *clase padre*), el “clon” modificado (conocido como *clase derivada*, *clase heredada*, *subclase* o *clase hija*) también reflejará esos cambios.

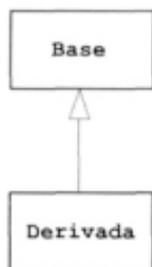


Figura A.3. Representación UML de una clase derivada de su clase base.

Dos tipos o clases pueden tener características y comportamientos en común, pero uno de ellos puede contener más características que el otro y puede incluso manejar más mensajes (o manejarlos diferente). La herencia expresa estas similitudes entre tipos con el concepto de *tipos base* y *tipos derivados*. Un tipo base contiene todas las características y comportamientos que son compartidos entre sus tipos derivados. Un tipo base es creado para representar una estructura general de una idea acerca de algún objeto dentro de un sistema. A través de el tipo base, otros más son derivados para expresar las diferentes formas en que esta estructura puede ser presentada.

Un ejemplo clásico de este concepto es el de las figuras geométricas. El tipo base es "figura", y cada una de ellas posee tamaño, color, posición, etc (propiedades). Además pueden ser dibujadas, borradas, movidas, coloreadas, etc (comportamientos o mensajes). A partir de "figura", tipos muy específicos pueden ser derivados: círculo, cuadrado, triángulo, etc., cada uno de los cuales puede tener características y comportamientos adicionales.

Cuando se deriva desde un tipo existente, se crea un nuevo tipo. Este nuevo tipo no solo contiene los miembros del tipo existente (aunque los miembros privados permanecen escondidos y son inaccesibles), sino que además duplica la interface de la clase base. Esto es, todos los mensajes que se puedan mandar a los objetos de la clase base pueden ser enviados también a los objetos de la clase derivada. Ya que se conoce el tipo de una clase por los mensajes que se le pueden mandar, esto significa que la clase derivada es *del mismo tipo que la clase base*. En el ejemplo anterior, un círculo sigue siendo una figura. Esta equivalencia de tipos a través de la herencia es uno de los conceptos más importantes dentro de la POO.

Existen más temas relacionados con la herencia dentro de la POO como lo son sobrecarga de funciones (*overriding*), *upcasting*, polimorfismo y funciones virtuales [5] que exceden los objetivos de este apéndice. Cabe mencionar que dentro del desarrollo de los algoritmos planteados por esta tesis no ha sido necesario introducir el concepto de herencia.

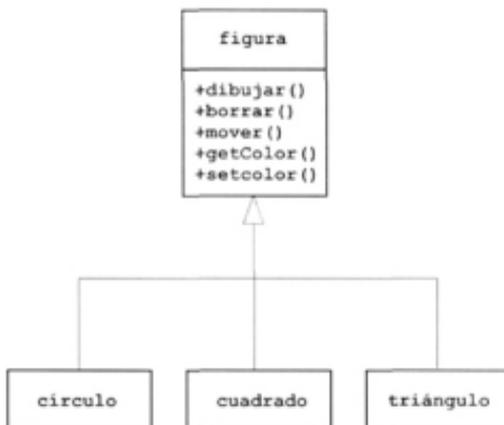


Figura A.4: Representación de la clase base "figura" y sus derivadas.

A.2 Aplicación de la POO en el proyecto

El primer paso dentro del diseño orientado a objetos es contestar las siguientes dos preguntas:

- ¿Cuáles son los objetos? o ¿Cómo se dividirá el proyecto en sus diversos componentes?.
- ¿Cuáles son las interfaces de dichos objetos? ó ¿Qué mensajes es necesario ser capaz de mandar a cada objeto?.

Generalmente esto es suficiente para comenzar a escribir el código de un programa aunque es posible necesitar más descripciones y documentación que esto, sin embargo, es la mínima información requerida para iniciar.

A.2.1 Objetivo general del sistema

La primera fase dentro del diseño orientado a objetos es determinar lo que se supone que el sistema debe ser capaz de hacer. Los requerimientos generales del sistema de control de posición basado en CSM propuesto en los capítulos centrales de esta tesis, consideran los siguientes aspectos:

1. La comunicación entre el programa y el controlador del manipulador, que fundamentalmente debe proveer al sistema de la información acerca de la posición actual del robot además de ser capaz de enviar órdenes de movimiento al robot.
2. La implementación del modelo de control basado en CSM desarrollado en el capítulo 2.
3. El programa debe proveer de un medio de comunicación amigable con el usuario, que permita la ejecución de los objetivos de una forma intuitiva.

A.2.2 Diseño de clases

La segunda fase del diseño de un programa orientado a objetos constituye el diseño que describe la apariencia de las clases así como su interacción. Entre otras cosas, en esta fase es necesario especificar:

- El nombre de las clases. Es importante que este nombre capture la esencia de lo que la clase realiza.
- Las “responsabilidades” de la clase, es decir, qué debe hacer. Típicamente esto puede ser resumido estableciendo el nombre de sus funciones miembro, tomando en consideración que estos nombres deben ser lo suficientemente descriptivos.
- Los “colaboradores” de estas clases: ¿Qué otras clases interactúan con ellas? Estos colaboradores constituyen las herramientas utilizadas para el desarrollo de las clases básicas del sistema.

Debe tenerse en cuenta que en esta etapa es necesario mantener estos conceptos en su mínima expresión.

El sistema propuesto ha sido resumido en las siguientes clases. El detalle de su implementación es presentado mas adelante:

CFanuc. Ésta es la clase que implementa la comunicación con el robot Fanuc. Debe proveer de las funciones básicas para la interacción entre el manipulador y el programa implementado en la PC.

El sistema de comunicación incluido en el robot está basado en el protocolo de comunicación TCP/IP-Ethernet, esto permite la conexión de un grupo de robots en una configuración semejante a una red de computadoras, todos ellos administrados por una computadora principal, que es donde reside el programa. La implementación de esta clase se justifica ante la necesidad de establecer parámetros muy específicos para cada uno de estos robots, relacionados con su dirección IP asignada². De esta manera un programa puede manejar cualquier número de objetos CFanuc, cada uno de ellos relacionados con un robot diferente conectado al sistema.

CCamara. El elemento básico, desde el punto de vista de objetos, de un sistema de visión basado en CSM lo constituye la cámara de video (esta consideración se podría ampliar a cualquier sistema de visión). Este dispositivo es el encargado de introducir al sistema la información necesaria para el desarrollo de las técnicas de control. El número de cámaras utilizado puede variar, pero en general cada una de ellas posee un conjunto de parámetros que la describen. La clase CCamara permite definir el número de objetos cámara necesarios, siendo este número completamente ajustable a las necesidades del sistema.

CConfig. Esta clase constituye la estructura principal del sistema. De acuerdo al concepto de *composición* presentado anteriormente, esta clase contiene los objetos de las clases anteriores

²La dirección IP constituye el número que identifica a una terminal o dispositivo dentro de una red.

necesarios para la implementación. Una de sus funciones es manipular dichos objetos con el fin de aplicar las técnicas de control desarrolladas.

Inicialmente esta clase ha sido conceptualizada con el único fin de integrar las piezas que conforman el sistema. Una definición más apegada a la POO puede derivarse de la necesidad de adaptar la clase a cualquier modelo de robot. Como es sabido, un manipulador está definido por su modelo cinemático; utilizando los conceptos de herencia vistos anteriormente, se pueden derivar clases específicas para cada tipo y modelo de robot sin la necesidad de reescribir más código que el que define el modelo cinemático. Esto da una idea de las posibilidades y alcances de este modelo de programación.

En los siguientes segmentos se presenta con más detalle la descripción de cada una de estas clases, aunque por razones de espacio, el código fuente relacionado es omitido.

A.2.3 Descripción general de la clase CFanuc

Esta clase ha sido desarrollada con base en la *Interfaz R-J2 MOTET*, que es una aplicación opcional común al SISTEMA R-J2 de FANUC. Esta interfaz provee el protocolo MOTET para la comunicación entre el controlador R-J2 y un dispositivo externo utilizado como computadora host.

La especificación MOTET es en realidad una pila de protocolos, lo cual significa que es un conjunto predefinido de protocolos acoplados uno arriba de otro. La pila de protocolos MOTET consiste en las siguientes capas³:

- El protocolo MMS.
- El protocolo TCP/IP
- Ethernet.

MMS es un lenguaje que define un conjunto de comandos adecuado para equipo funcionando en un ambiente de manufactura. La interfaz MOTET R-J2 utiliza los siguientes comandos llamados servicios:

- Servicios del medio de comunicación.
 - Initiate
 - Conclude
 - Abort
 - Reject
- Servicios de información.
 - Identify
 - Status

³El término utilizado en inglés es *layers*.

- Servicios de transferencia de archivos.
 - Obtainfile
 - FileOpen
 - FileRead
 - FileClose
- Servicios de directorio.
 - FileDirectory
 - FileDelete
 - FileRename
- Servicios de acceso a variables.
 - Read
 - Write
 - InformationReport

MMS es parte del layer de aplicación del modelo de comunicaciones OSI (Open System Interconnection). Está definido en el estándar internacional "Industrial Automation Systems - Manufacturing Message Protocol", especificación ISO 9506.

El protocolo TCP/IP y el estándar Ethernet constituyen la última parte de MOTET, su descripción puede ser encontrada en cualquier libro especializado.

El software requerido para la implementación del protocolo MOTET en una PC están desarrollados bajo el modelo COM/OLE⁴. Se ha implementado una clase básica **CMotet**, la cual implementa todos los procedimientos y eventos contenidos dentro el objeto OLE suministrado por Fanuc. Básicamente estos procedimientos son los descritos por el protocolo MMS mencionados anteriormente. Es decir, **CMotet** se puede ver como un conjunto de funciones de bajo nivel encargada de comunicar cualquier aplicación desarrollada bajo Visual C++ con el objeto OLE.

La clase **CFanuc** es introducida con la finalidad de reducir la complejidad en el uso de las funciones contenidas en **CMotet**. **CFanuc** está constituido por un objeto miembro **CMotet** (privado) y un conjunto de funciones miembro (públicas) que hacen uso de este objeto para implementar operaciones más específicas y relacionadas con el control de posición del robot.

A continuación se presentan algunas de las principales funciones desarrolladas dentro de **CFanuc**:

GetVariable(). Regresa la variable del sistema (Fanuc) solicitada.

SetVariable(). Establece el valor numérico de una variable del sistema.

⁴Modelo de Componentes Objeto (Component Object Model) es una arquitectura software estándar del mercado apoyada principalmente por Microsoft, Digital Equipment y muchas otras compañías. Este modelo se utiliza como una manera estándar de comunicación entre módulos de programa del sistema operativo Windows.

GetConfigRobot(). Regresa un vector de seis elementos correspondientes a los ángulos (en radianes) actuales de las juntas del robot.

GoToJointPos(). Recibe como argumento seis valores de tipo punto flotante correspondientes a la posición a la que se quiere llevar cada ángulo del robot.

Ready(). Mientras el robot esté realizando un movimiento regresa una constante booleana FALSE. De lo contrario el valor de retorno es TRUE.

ShowError(). Muestra el último error generado.

El diagrama UML de la Figura A.5 muestra la implementación de la clase **CFanuc**.

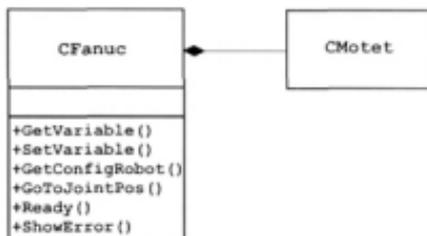


Figura A.5: Esquema UML de la clase **CFanuc**.

Variables de sistema

Para realizar la mayoría de sus funciones el controlador del robot cuenta internamente con un conjunto de variables que utiliza para almacenar parámetros, status, configuraciones y demás parámetros propios del sistema. Estas variables pueden ser de solo lectura (Read Only) o de lectura-escritura (R/W), y pueden ser accedidas a través de las funciones MMS arriba mencionadas.

Específicamente la posición angular actual de las juntas del robot se encuentran en las variables “**\$MOR_GRP [1] . \$PERCHPOS [1 . . 9]**”, donde el último índice corresponde a cada uno de los seis grados de libertad del brazo. Esta variable es de solo lectura y contiene el valor del ángulo en radianes.

Existe un segundo grupo de variables utilizado por la clase **CFanuc**. La variable “**\$REFPOS2 [1] . \$PERCHPOS [1 . . 9]**” originalmente está diseñada para que los programas desarrollados dentro del controlador almacenen posiciones de referencia. Pero aprovechando su atributo R/W en la clase **CFanuc** los primeros seis elementos (“**\$REFPOS2 [1] . \$PERCHPOS [1 . . 6]**”) son utilizados como un medio de comunicarle al robot la posición a la cual debe colocar cada ángulo del brazo. “**\$REFPOS2 [1] . \$PERCHPOS [7]**” es utilizada como una bandera para indicarle al robot que ya se ha transmitido una configuración completa y que ya es posible que la lea para posteriormente alcanzarla.

Para que el robot realice los movimientos indicados por la PC a través de MOTET es necesario que dentro de su controlador se encuentre ejecutando un programa "receptor". Este programa tiene como única función esperar a que la bandera arriba mencionada sea activada (Esta espera la realiza mediante una sentencia "IF...THEN". Una vez activada esta bandera el programa lee la posición recién enviada (y almacenada en las variables de sistema correspondientes) y ejecuta el movimiento correspondiente a dicha posición angular.

En el listado A.1 se muestra el programa encargado de recibir las órdenes desde el programa en la PC. Aunque el código original no puede incluir comentarios, estos son presentados en el estilo C++ (\(\dots\)) y las líneas en blanco son para hacer el programa mas legible.

Limitaciones

Este medio de comunicación posee una limitación seria en cuanto al tiempo de lectura del valor angular de las juntas del robot. Experimentalmente, la lectura de cualquier variable de sistema a través de MOTET lleva aproximadamente 60 ms y no es posible leer mas de una variable a la vez. Esto se traduce en un error en la lectura de la posición actual, ya que cuando se lee el sexto ángulo han pasado 300 ms o mas (experimentalmente se han registrado lapsos hasta de 500 ms), y esta posición no corresponde al tiempo en que se midió el primer ángulo.

Para salvar este problema se recurrió a la ponderación basada en dos posiciones leídas continuamente. La Figura A.6 muestra una gráfica de los tiempos en los que es leído cada ángulo. La imagen que se desea relacionar con la posición instantánea de los ángulos es tomada entre los dos conjuntos de ángulos leídos.

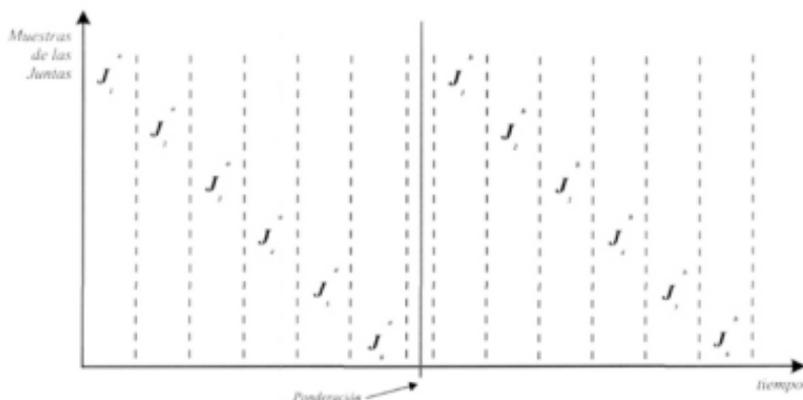


Figura A.6: Ponderación de la configuración interna leída del manipulador a partir de dos mediciones subsiguientes.

Por ejemplo, si se desea conocer el valor ponderado de J_1 , éste tendrá una ponderación mas alta en J_1^b ya que este se encuentra mas cercano (en términos de tiempo) al momento en que se

```

1 CALL HOME // Envía al robot a su posición inicial.
2 PR[1]=JPOS // Almacena la configuración del robot en
3 // un registro de posición.
4 R[1]=PR[1,1] // Copia la posición a un registro común.
5 R[2]=PR[1,2]
6 R[3]=PR[1,3]
7 R[4]=PR[1,4]
8 R[5]=PR[1,5]
9 R[6]=PR[1,6]
10
11 $REFFPOS2[1].$PERCHPOS[1]=R[1] // Almacena los valores de la configuración
12 $REFFPOS2[1].$PERCHPOS[2]=R[2] // actual en las variables del sistema.
13 $REFFPOS2[1].$PERCHPOS[3]=R[3]
14 $REFFPOS2[1].$PERCHPOS[4]=R[4]
15 $REFFPOS2[1].$PERCHPOS[5]=R[5]
16 $REFFPOS2[1].$PERCHPOS[6]=R[6]
17 $REFFPOS2[1].$PERCHPOS[7]=0 // Este valor indica que no ha sido enviada
18 // ninguna nueva posición.
19 LBL[1] // Etiqueta UNO.
20 $REFFPOS2[1].$PERCHPOS[8]=1 // Este valor indica que se está ejecutando
21 // un movimiento.
22 LBL[3] // Etiqueta TRES.
23 R[7]=$REFFPOS2[1].$PERCHPOS[7] // Verifica si ha sido enviada una nueva posición.
24 IF R[7]=1, JMP LBL[2]
25 JPM LBL[3]
26
27 LBL[2] // Etiqueta DOS.
28 $REFFPOS2[1].$PERCHPOS[8]=0 // Tarea Ejecutándose.
29
30 R[1]=$REFFPOS2[1].$PERCHPOS[1] // Lee la nueva posición enviada.
31 R[2]=$REFFPOS2[1].$PERCHPOS[2]
32 R[3]=$REFFPOS2[1].$PERCHPOS[3]
33 R[4]=$REFFPOS2[1].$PERCHPOS[4]
34 R[5]=$REFFPOS2[1].$PERCHPOS[5]
35 R[6]=$REFFPOS2[1].$PERCHPOS[6]
36
37 $REFFPOS2[1].$PERCHPOS[7]=0 // Indica que ya se ha leído la posición
38 // y que ya puede ser enviada otra.
39 PR[1,1]=R[1] // Almacena la posición en un Registro de Posición.
40 PR[1,2]=R[2]
41 PR[1,3]=R[3]
42 PR[1,4]=R[4]
43 PR[1,5]=R[5]
44 PR[1,6]=R[6]
45
46 L PR[1] 500 MM/SEC CNT100 // Alcanza la posición.
47
48 JMP LBL[1] // Regresa a la etiqueta UNO.

```

Listado A.1: Programa almacenado dentro del controlador del robot encargado de recibir las órdenes de movimiento desde la PC.

desea conocer la posición. De esta manera los valores ponderados se encontrarán alineados como si hubieran sido leídos al mismo tiempo.

Otra forma de reducir el error en la lectura de las posiciones angulares es disminuyendo la

velocidad a la que se mueve el brazo, ya que esto evita un desplazamiento significativo entre la lectura del primer y último ángulo. La gráfica de la Figura A.7 muestra el comportamiento lineal del error promedio encontrado experimentalmente al variar el porcentaje de velocidad. Se encontró un error muy aceptable a una velocidad de 5mm/seg.

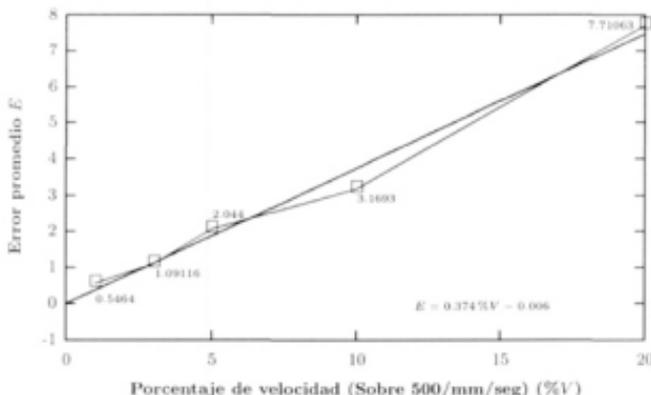


Figura A.7: Error promedio vs. porcentaje de velocidad.

A.2.4 La clase CCámara

El elemento principal del sistema de visión es la cámara de video. De acuerdo a CSM y al desarrollo propuesto en el capítulo 2, la clase **CCámara** que describe este dispositivo debe poseer las siguientes características intrínsecas (en POO se manejan como variables miembro de clase):

- Un conjunto de *parámetros de visión*, los cuales describen la relación entre el espacio bidimensional de la cámara y el sistema coordinado en la base del robot además de las matrices de momentos involucradas en su estimación [8] y los vectores \mathbf{b} y $\frac{\partial \mathbf{b}}{\partial \mathbf{C}}$.
- Una matriz de 640×480 elementos para almacenar la imagen actual obtenida de la cámara así como un arreglo para almacenar las coordenadas de las marcas visuales encontradas en ella.
- Las matrices de momentos relacionadas con la estimación de la configuración interna del manipulador (ecs. (2.20), (2.23), (2.29) y la matriz de diagonalización \mathbf{L}^i).
- Un arreglo que almacene las coordenadas del conjunto de puntos objetivo observados por cada cámara.
- Un conjunto de parámetros para el mapeo con la cámara de observación (sección 2.4).

- Algunos otros parámetros como el canal de video al cual está conectada la cámara en la tarjeta de digitalización de video, el parámetros Z_0 usado en *flattening* y el umbral utilizado en el análisis de las imágenes.

El comportamiento de la clase **CCamara** está descrito por sus funciones miembro (se muestran solo las funciones principales, aunque existen otras funciones auxiliares de menor importancia en la descripción del objeto):

GetImage(). Obtiene el arreglo de 640×480 de una imagen.

Centroides(). Analiza la imagen para obtener las coordenadas de las marcas visuales.

AddCentroidePC() y **AddCentroidePCFlat()**. Agregan una muestra a las matrices de momentos (para las estructuras con y sin *flattening* respectivamente) utilizadas para la estimación de los parámetros de visión.

SetParamC() y **SetParamCFlat()**. Estima los parámetros de visión con y sin *flattening*.

Mapear(). Aplica las ecuaciones de mapeo (ecs. 2.37) a un punto para obtener su correspondencia en la cámara de observación.

SetChannel(). Establece el canal de video asignado a la cámara.

La clase **CCamara** queda así descrito en la Figura A.8 por su correspondiente diagrama UML.

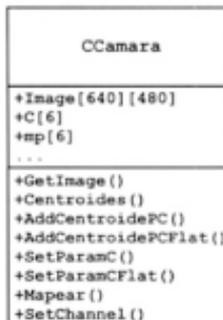


Figura A.8: La clase CCamara.

A.2.5 La clase CConfig

La clase **CConfig** se encarga de integrar todas las piezas y hacer uso de cada una de ellas para la implementación de los algoritmos de estimación de la configuración interna del manipulador. El diseño de esta clase permite la definición de un número indefinido de objetos **CCamara** (número limitado solamente por la capacidad de la tarjeta de adquisición de video). Estos objetos **CCamara** pueden ser agregados en tiempo de ejecución, es decir, un sistema terminado puede incluir la opción al usuario de agregar más cámaras según lo requiera su sistema de visión.

El uso de un arreglo de objetos **CCamara** creado dinámicamente permite referirse a una cámara específica mediante un código semejante al del listado A.2 (los puntos suspensivos indican la existencia de más código del que se muestra).

En este listado, **pCam** es un puntero que define el arreglo de objetos **CCamara** (línea 9 de código) y el índice **i** (líneas 35-36) especifica el número de cámara. El tamaño del arreglo es definido dentro de la implementación del constructor de la clase (línea 23)⁵.

En el ejemplo anterior la función **CConfig::GetCentroides()** (línea 29) obtiene la imagen de cada una de las cámaras definidas en el sistema y las analiza para obtener la posición (centroides) de las marcas luminosas que aparecen en cada una de ellas.

Los datos miembro utilizados para definir las características de la clase **CConfig** se muestran a continuación:

- el arreglo de objetos cámara ya mencionado **CCamara *pCam**.
- El número de cámaras instaladas en el sistema **Nc**.
- La cámara definida de observación **Oc**. Este valor es un entero positivo que especifica el número de la cámara instalada en el sistema que es utilizada como cámara de selección.
- Un vector que contiene la configuración actual de ángulos del manipulador. Este vector es utilizado también para almacenar los resultados de los procesos de estimación de la configuración interna.
- La matriz **Gamma** mencionados en el desarrollo de CSM (ec. (2.17)).

Las funciones miembro básicas de esta clase implementan las funciones de estimación de la configuración interna, mapeo de superficies, determinación de leds visibles en la herramienta, control de encendido y apagado de las marcas visuales de la herramienta y del laser, etc.

Estimate(). Obtiene una estimación de la configuración interna del manipulador.

Targets(). Obtiene los puntos objetivo que observa cada cámara.

Laser(), **RefLed()** y **Leds()**. Controla el encendido y apagado del láser que proyecta los puntos objetivo sobre la superficie de trabajo, y los leds que producen las marcas visuales en la herramienta.

⁵El capítulo 13 de [5] presenta más detalles acerca de creación dinámica de objetos.

```

1 class CConfig
2 {
3 public:
4     CConfig();           // Constructor de clase
5     ~CConfig();        // Destructor de clase
6
7     // ...
8
9     CCamara *pCam;     // Puntero que define el arreglo de objetos CCamara
10
11    // ...
12
13 private:
14    // ...
15 }
16
17 CConfig::CConfig()    // Implementación del constructor de clase.
18 {
19
20    // ...
21
22    pCam = new CCamara[Nc+1]; // Definición de la longitud del arreglo
23                               // de objetos CCamara.
24                               // Nc define el número de cámaras.
25
26    // ...
27 }
28
29 CConfig::GetCentroides() // Función miembro de prueba.
30 {
31     int i;
32
33     for(i=1; i<=Nc; i++)
34     {
35         pCam[i].GetImage();
36         pCam[i].Centroides();
37     }
38 }

```

Listado A.2: Declaración dinámica del arreglo de objetos **CCamara** dentro de la clase **CConfig**. Se muestra la manipulación del arreglo dentro de la función miembro de ejemplo **CConfig::GetCentroides**.

GetPosXYZ(). Utiliza el modelo cinemático del robot para conocer su posición en el espacio tridimensional.

Mapeo(). Obtiene los parámetros del mapeo que relaciona cada una de las cámaras con la cámara de selección.

La Figura A.9 presenta el esquema UML general del sistema. En ella aparecen los objetos **CCamara** y **CFanuc** agregados.

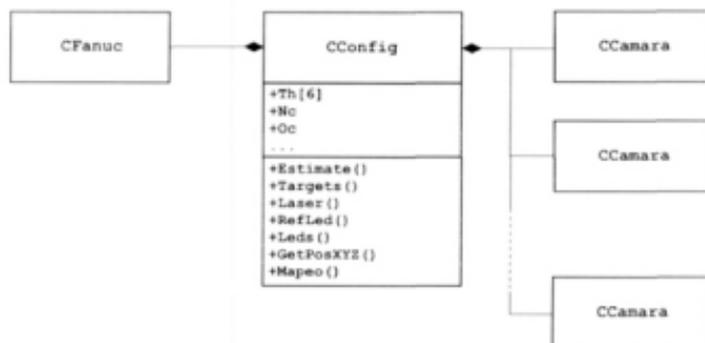


Figura A.9: Composición de la clase CConfig. diagrama UML.

A.3 Aplicación: La interfaz gráfica

El objetivo principal de una interfaz gráfica es el de presentar ante un hipotético usuario una forma sencilla, práctica, amigable e intuitiva de utilizar los algoritmos de control desarrollados para el posicionamiento de un robot sobre superficies arbitrarias a través de CSM.

El diseño actual de la interfaz es presentado como una propuesta y no es definitivo. Como ya ha sido señalado anteriormente, cuenta aún con serias deficiencias en cuanto a su desempeño. Este prototipo pretende mostrar la facilidad de la implementación de las clases desarrolladas mediante la POO en un programa que cumpla con los estándares de un producto de software comercial, como pueden ser menús, barras de herramientas, interacción con el usuario a través de un apuntador o mouse, barras de desplazamiento, etc.

La Figura A.10 muestra la apariencia inicial del programa. Se puede apreciar un menú de opciones con la barra de herramientas o accesos directos a las funciones más utilizadas. La barra de herramientas ofrece las siguientes funciones:

- **Abrir imagen.** Abre una imagen previamente guardada en el disco duro.
- **Guardar imagen.** Almacena en el disco duro la imagen que aparece en primer plano.
- **Centroides.** Analiza la imagen actual para localizar el centroide de las marcas luminosas que aparezcan dentro de ella.
- **Imagen cámara uno e Imagen cámara uno.** Obtiene una imagen (640 × 480 píxeles) desde la cámara uno o dos y la despliega en la pantalla.
- **Imagen continua cámara uno e Imagen continua cámara uno.** Despliega aproximadamente 10 imágenes por segundo en la pantalla. Permite instalar las cámaras y orientarlas hacia el área de interés de una manera sencilla.

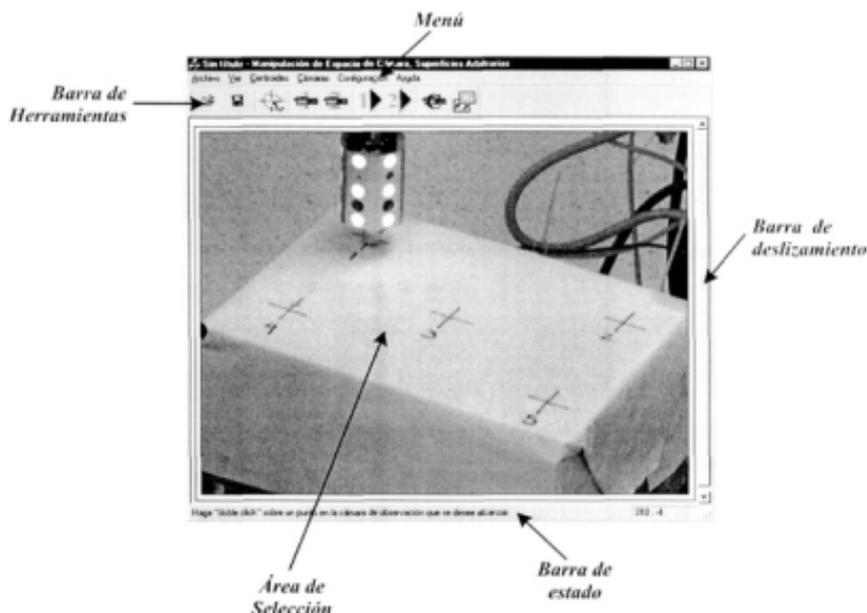


Figura A.10: Interface gráfica para el posicionamiento de robots basado en CSM.

- **Mapeo** Despliega un cuadro de diálogo para facilitar el procedimiento de obtención de muestras para establecer los parámetros que definen la relación entre las superficies observadas por cada cámara.
- **Trayectoria preplaneada.** Presenta un cuadro de diálogo en donde es posible seleccionar la cámara a la cual se aplicará la estimación preliminar de los parámetros de visión mediante una trayectoria preplaneada.

La superficie de trabajo hacia la cual se desea dirigir el manipulador aparece en primer plano. Mediante el apuntador del mouse, el usuario puede seleccionar (mediante un "doble-click") el punto sobre la superficie en el cual desea posicionar el manipulador. Inmediatamente después aparece un cuadro de diálogo (Figura A.12) en donde se despliega la posición en el espacio de cámara seleccionada y la configuración de ángulos internos del manipulador estimados. Es posible visualizar también el conjunto de parámetros de visión asociado a cada cámara. El usuario tiene la opción de ejecutar la tarea de posicionamiento con el botón "Ir a Posición". Información acerca del proceso como número de iteraciones efectuadas para llegar a la convergencia y tiempo de procesamiento en milisegundos aparece en la parte inferior derecha del cuadro de diálogo.

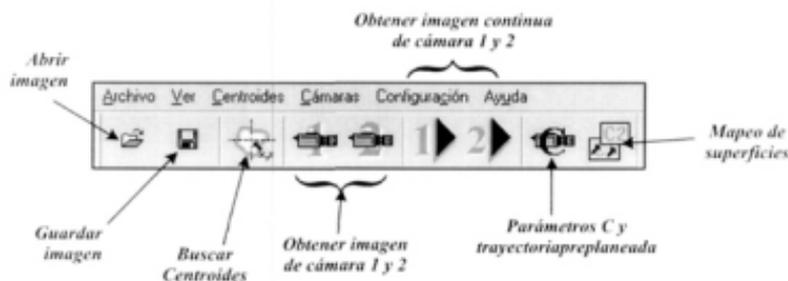


Figura A.11: Barra de herramientas del programa.

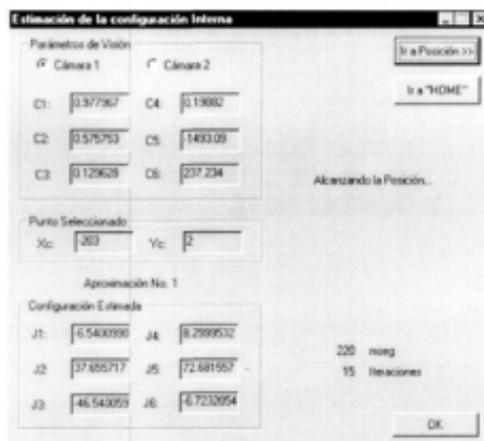


Figura A.12: Cuadro de diálogo para el posicionamiento del manipulador.

APÉNDICE B

Esquemas y diagramas

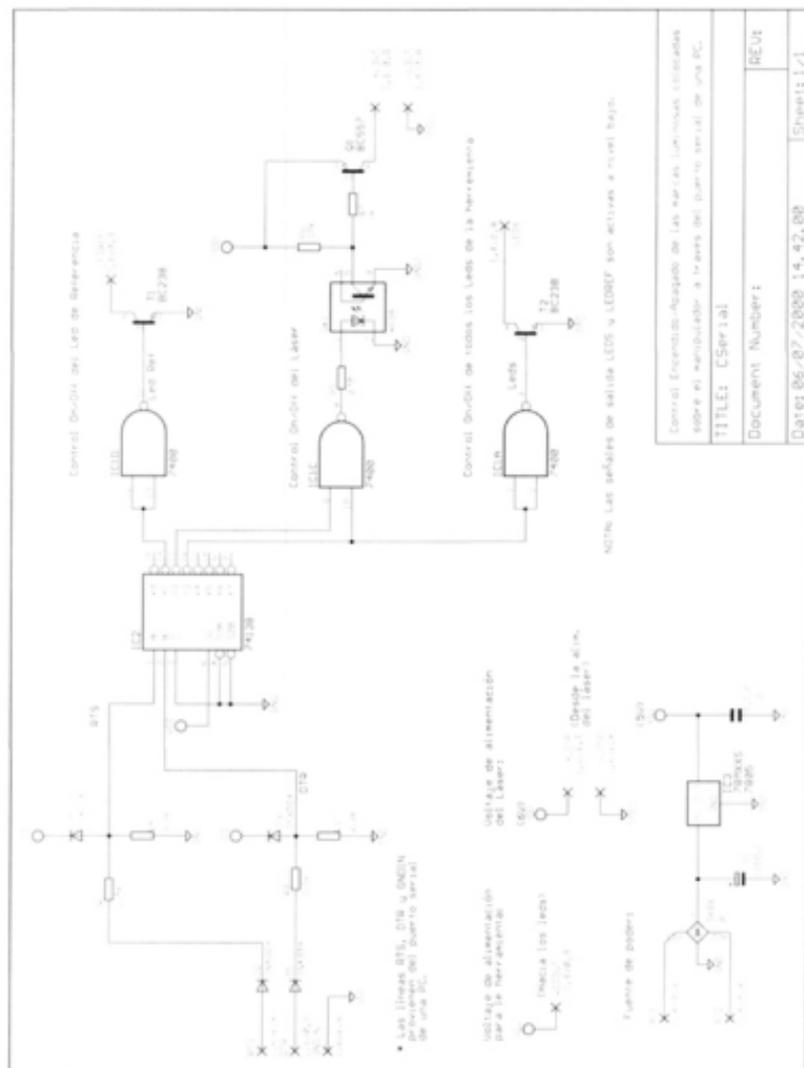


Figura B.3: Diagrama electrónico del circuito de control On/Off de las marcas luminosas en el manipulador.

Fotografías

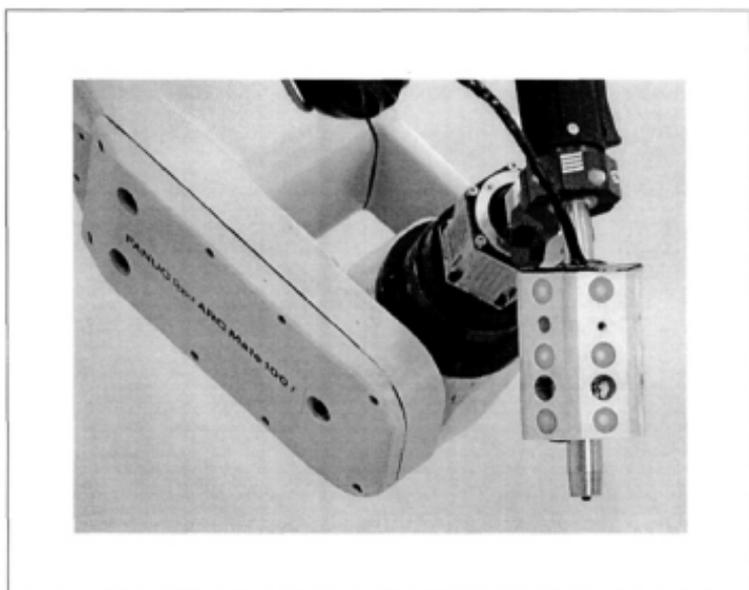


Figura C.1: Primer plano del robot Fanuc ArcMate 100i utilizado en los experimentos de posicionamiento. El aditamento para el uso de marcas luminosas esta colocado en la antorcha de soldadura.

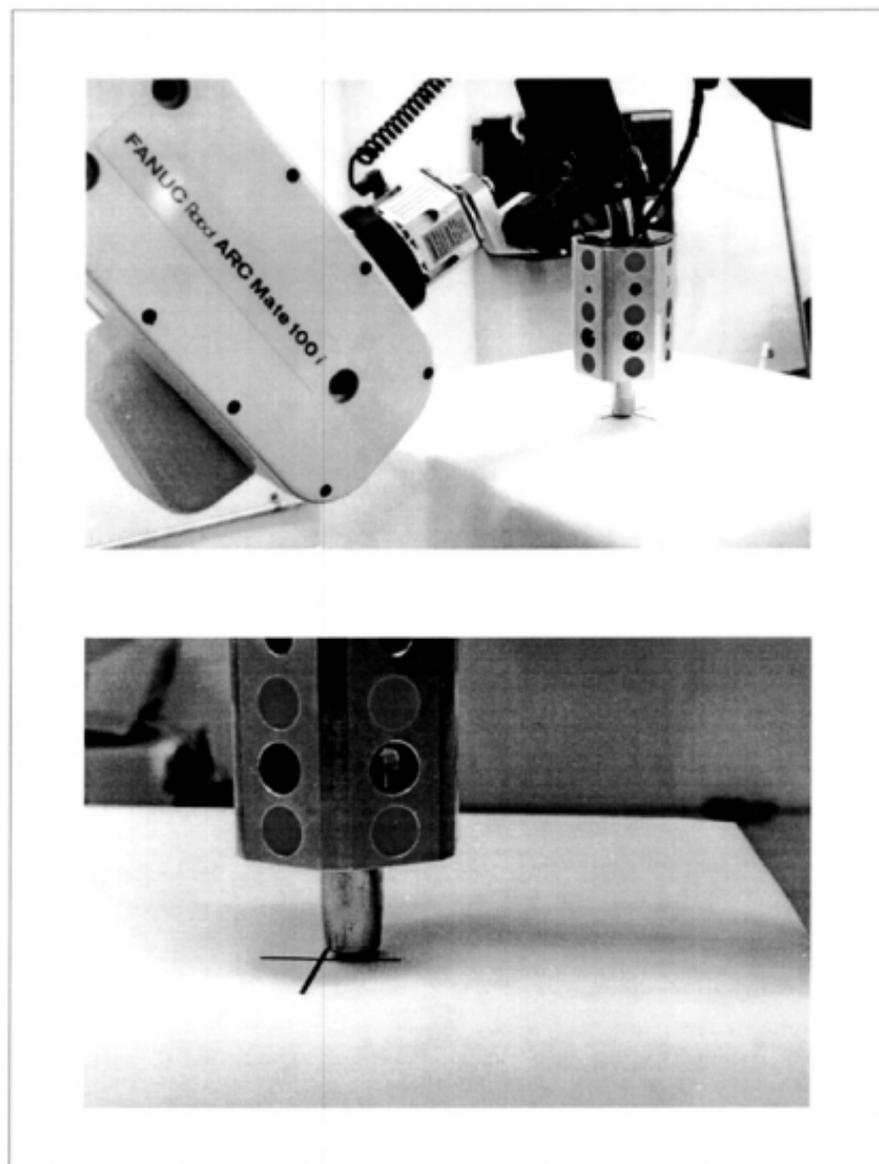


Figura C.2: Primera aproximación al punto seleccionado sobre la superficie de trabajo.

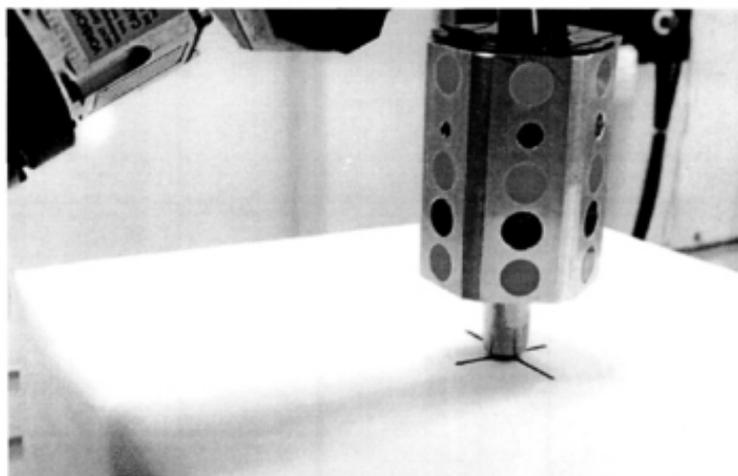


Figura C.3: Última aproximación al punto seleccionado.

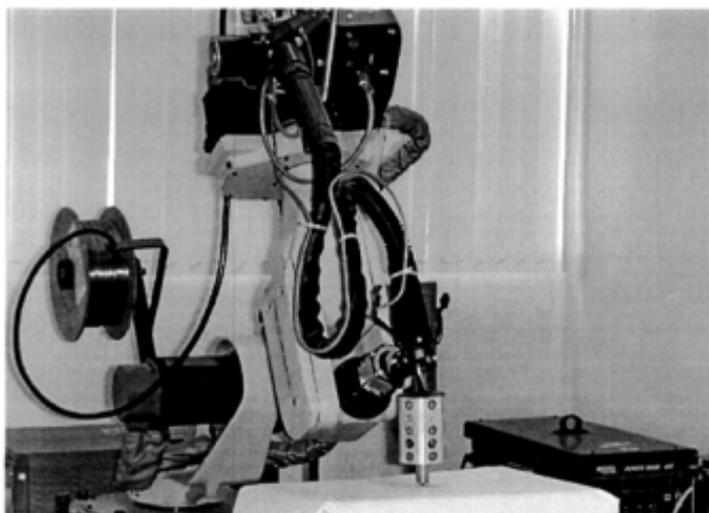


Figura C.4: Robot Famic®ArcMate 100i en una tarea de posicionamiento.

Índice de Figuras

1.1	Segmentación de imagen, (a) modelo de imagen con cuadrícula y (b) imagen segmentada.	7
1.2	Procedimiento de crecimiento de región para el análisis de una imagen.	8
1.3	Procedimiento de seguimiento de bordes para detectar el límite de una imagen binaria.	9
1.4	Sistema de referencia unido a la cámara.	15
1.5	Error en el modelo ortográfico inducido por $Z_0 \rightarrow \infty$	21
1.6	Proyección de acuerdo al modelo de orificio de cámara.	22
1.7	Sistemas de referencia relacionados con <i>flattening</i>	24
1.8	Esquema de los parámetros involucrados en el procedimiento de Denavit-Hartenberg.	28
1.9	Ejemplo de robot articulado configurado en paralelogramo.	29
1.10	Emulación de la configuración paralelogramo en el robot Fanuc®.	30
1.11	Modelo Cinemático del robot Fanuc® ArcMate 100i.	33
2.1	Detección de centroides en una imagen.	36
2.2	Esquema del aditamento utilizado para agregar el conjunto de marcas luminosas a la herramienta.	37
2.3	Marcas visuales colocadas sobre la herramienta del manipulador.	39
2.4	Superposición de las marcas luminosas dentro de la imagen bidimensional.	41
2.5	Vector \mathbf{V}_k asociado a cada uno de los leds	42
2.6	Sistemas coordenados (X, Y, Z) , (x_i, y_i, z_i) y (X_E, Y_E, Z_E)	43
2.7	Diagrama de flujo para la identificación de múltiples marcas luminosas en la herramienta	44
2.8	Sistemas coordenados descritos por las matrices \mathbf{D} y $\mathbf{\Gamma}$	46
2.9	Mapeo de los puntos observados por la cámara dos hacia la cámara de observación	53
2.10	Arreglo de 7×7 puntos láser proyectados sobre la superficie de trabajo	55
2.11	Procedimiento para encontrar las coordenadas tridimensionales de los puntos objetivo encontrados en la cámara de observación.	57

2.12 Descripción del procedimiento general para la determinación de la configuración interna del manipulador.	59
3.1 Trayectoria preplaneada para la cámara uno.	62
3.2 Primera aproximación dentro del proceso de estimación de la configuración interna.	64
3.3 Segunda y tercera aproximación.	65
3.4 Procedimiento de estimación de la configuración interna en un solo paso.	68
A.1 Representación UML de la clase Light.	78
A.2 Diagrama UML donde la composición Auto-Motor se indica con un diamante.	79
A.3 Representación UML de una clase derivada de su clase base.	80
A.4 Representación de la clase base "figura" y sus derivadas.	81
A.5 Esquema UML de la clase CFanuc.	85
A.6 Ponderación de la configuración interna leída del manipulador a partir de dos mediciones subsiguientes.	86
A.7 Error promedio vs. porcentaje de velocidad.	88
A.8 La clase CCamara.	89
A.9 Composición de la clase CConfig. diagrama UML.	92
A.10 Interface gráfica para el posicionamiento de robots basado en CSM.	93
A.11 Barra de herramientas del programa.	94
A.12 Cuadro de diálogo para el posicionamiento del manipulador.	94
B.1 Diagrama mecánico del aditamento colocado en la herramienta. <i>Página 1</i>	96
B.2 Diagrama mecánico del aditamento colocado en la herramienta. <i>Página 2</i>	97
B.3 Diagrama electrónico del circuito de control On/Off de las marcas luminosas en el manipulador.	98
C.1 Primer plano del robot Fanuc ArcMate 100i utilizado en los experimentos de posicionamiento. El aditamento para el uso de marcas luminosas está colocado en la antorcha de soldadura.	99
C.2 Primera aproximación al punto seleccionado sobre la superficie de trabajo.	100
C.3 Última aproximación al punto seleccionado.	101
C.4 Robot Fanuc® ArcMate 100i en una tarea de posicionamiento.	102

Índice de Tablas

1.1	Parámetros de Denavit–Hartenberg para el modelo cinemático del robot Famic [®] ArcMate 100i.	28
2.1	Traslaciones en el sistema coordenado de la herramienta correspondientes a cada LED. 40	
3.1	Parámetros de visión experimentales obtenidos para las cámaras uno y dos.	63
3.2	Estimación de la configuración interna Θ en tres aproximaciones. Los valores están en grados ($^{\circ}$).	64
3.3	Corrección de los parámetros de visión al usar <i>flattening</i> durante las aproximaciones 2 y 3.	65
3.4	Resultados experimentales. La columna ERR representa el error medido de la aproximación con respecto al punto seleccionado. Δ es la máxima corrección real aplicada. 66	

Bibliografía

- [1] BARRIENTOS C., PEÑÍN L. F., BALAGUER C., Y ÁRACIL R., *Fundamentos de Robótica*, McGraw Hill, Universidad Politécnica de Madrid, España, 1o Ed., 1997.
- [2] BROKKEN F. B. Y KUBAT K., *C++ Annotations Version 4.4.0m*, ICCE, University of Groningen, Holanda, 1999.
- [3] CHEN W. Z., KORDE U. A., Y SKAAR S. B., *Position control experiments using vision*, The Int. J. of Robotics Res., 13 (1994), pp. 199-208.
- [4] DAVIES R. B., *NEWMAT09: A Matrix Library in C++*. Disponible gratis en internet: <http://webnz.com/robert>, 1997.
- [5] ECKEL B., *Thinking in C++: Introduction to Standard C++*, Vol. 1, Prentice Hall, EUA, 2o Ed.
- [6] ECKEL B., *Thinking in C++: Standard Libraries & Advanced Topics*, Vol. 2, Prentice Hall, EUA, 2o Ed.
- [7] FU K. S., GONZÁLEZ R. C., Y LEE C. S., *Robótica - Control, detección, visión e inteligencia*, McGraw-Hill, EUA, 1988.
- [8] GONZÁLEZ GALVÁN E. J., *Introducción al Curso de Robótica y Visión por Computadora*. Sin publicar, 1999.
- [9] GONZÁLEZ GALVÁN E. J., SKAAR S. B., Y SEELINGER M. J., *Efficient camera-space target disposition in a matrix of moments structure using camera-space manipulation*, The Int. J. of Robotics Res., 18 (1999), pp. 809-818.
- [10] GROOVER M. P., WEISS M., NAGEL R., Y ODREY N. G., *Robótica Industrial - Tecnología, programación y aplicaciones*, McGraw-Hill, EUA, 1992.
- [11] JUNKINS J. L., *An Introduction to Optimal Estimation of Dynamical Systems*, Alphen Aan Den Rijn, Sijthoff & Noordhoff, 1978. Universidad de Virginia, EUA.

- [12] KRUGLINSKI D. J., SHEPHERD G., Y WINGO S., *Programación Avanzada con Microsoft Visual C++*, Serie de Programación Microsoft, Microsoft Press - McGraw Hill, EUA, 1999.
- [13] LAMPORT L., *TEX_ε - A document preparation system*, Addison Wesley, EUA, 1994.
- [14] MATA V., VALERO F., Y CUADRADO J. I., *Mecánica de Robots*, Vol. 16 de la Serie Libro-Apiunte número, Universidad Politécnica Valencia, España.
- [15] MIRANDA GÓMEZ V. H., *Desarrollo de una interface gráfica para el control de robots por medio de manipulación de espacio de cámara*, Tesis de Maestría, Universidad Autónoma de San Luis Potosí, Facultad de Ingeniería, México, 1999.
- [16] PRESS W. H., VETTERLING W. T., TEUKOLSKY S. A., Y FLANNERY B. P., *Numerical Recipes in C - The Art of Scientific Computing*, Ed. Cambridge University Press, EUA, 2o Ed.
- [17] SKAAR S. B. Y RUOFF C. F., *Teleoperation and Robotics. Chap. 9: Versatile and Precise Vision-Based Manipulation*, American Institute of Aeronautics and Astronautics, Inc., EUA.
- [18] SPHAR C., *Aprenda Microsoft Visual C++ Ya*, Microsoft Press - McGraw Hill, EUA, 1999.
- [19] ZARATIAN B., *Microsoft Visual C++, Manual del Programador*, Microsoft Press - McGraw Hill, EUA, 1999.

Este documento fue escrito utilizando
la distribución MikTeX Ver. 1.20e de L^AT_EX2_ε
y dvipdfm Ver. 0.12.6e.

Fue generado por ultima vez el
14 de octubre de 2000.

darioglz@ciep.ing.uaslp.mx
darioglz@hotmail.com
dglizcano@yahoo.com