



UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

CONTROL DE SISTEMAS CON
RETROALIMENTACIÓN DE
SEÑALES POR MEDIO DE VISIÓN

T E S I S

Para obtener el título de

**INGENIERO ELÉCTRICO – ELECTRÓNICO
E
INGENIERO EN COMPUTACIÓN**

PRESENTAN

Vidal García Uriel Santiago
Contreras Alcalde Enrique

Director de tesis Dr. Miguel A. Hernández López



México, D. F. 2008

Agradecimientos

Quiero agradecer a todos los que hicieron esto posible:

A mis padres Enrique y Sara y a mi hermano Daniel, por su apoyo incondicional. A mis abuelos Olivo y Rebeca, que siempre se preocuparon por mí. A mi novia Alba, que con amor y paciencia a estado a mi lado en este proyecto. A la familia tan grande que tengo, donde cada uno de los ellos, me apoyo a su manera para que yo este en este punto. A mi amigo y compañero de tesis Uriel, por brindarme su apoyo para la investigación y elaboración de esta tesis. A todos y cada uno de mis amigos, en especial a los del Canal Cultural, a mis amigos de Compu, a los de La Banca del CCH Azcapo, a mi equipo de la Secu, y a toda la Banda de los de Villas. A mis maestros de todos los niveles escolares, de los cuales aprendí muchas cosas. En memoria de los que ya no están, pero que nos cuidan desde donde estén. A la Universidad Nacional Autónoma de México, por la formación y valores adquiridos.

Enrique Contreras Alcalde

El presente trabajo representa el esfuerzo de muchas personas, que de una u otra forma, contribuyeron en la finalización de uno de los proyectos más grandes de mi vida. A esas personas quiero expresar mi más sincero agradecimiento. A mi padres Santiago e Irais, quienes con su apoyo y consejo siempre he podido llevar cualquier idea adelante y a quienes debo cuanto soy, mis hermanos Azael y Josué, que sin ellos mis ocurrencias no hubiesen llegado tan lejos y le dan brillo a mi existencia. A mi extensa familia, en especial a Gimmy(†) y Alan, por su afecto y comprensión en los buenos y amargos momentos. A mi asesor el Dr. Miguel Ángel Hernández y mi compañero de tesis Enrique, por su amistad y esfuerzo para sacar este proyecto adelante. A mis amigos, ExBT, preparatorianos, Canal Cultural, Protecós, FI's, secus y las personas que he conocido recientemente, que me presionaron para terminar y que sin ellos mi vida en definitiva "otra cosa sería". A mis profesores y coach's, dentro y fuera de la Universidad Nacional Autónoma de México, y a la misma UNAM, por su instrucción, consejo y paciencia.

Uriel Santiago Vidal García

Índice general

1. Introducción	1
1.1. Planteamiento del Problema	2
1.2. Estado del Arte	2
1.3. Objetivos Generales	4
1.4. Objetivos Particulares	4
1.5. Organización del Trabajo	4
2. Preliminares	6
2.1. Control Difuso	6
2.2. Lógica Difusa	6
2.3. Conjuntos Difusos	7
2.3.1. Control Difuso	12
2.4. Visión Artificial	14
2.4.1. Esquema de Visión	15
2.5. Calibración de la Cámara	32
3. Control de Sistemas por Visión	35
3.1. Implementación del Sistema de Visión	35
3.1.1. Adquisición de Imágenes	35
3.1.2. Acondicionamiento e Iluminación del Entorno	38

3.1.3. Calibración	38
3.1.4. Preprocesamiento	41
3.1.5. Segmentación de Bordes y Reconocimiento de Formas	41
3.2. Control Descentralizado Difuso	43
3.2.1. Diseño del Controlador Difuso	43
3.2.2. Implementación del Control Difuso	45
3.2.3. Conexión con el Robot Rhino	47
3.3. Resultados de Simulación y Experimentos	47
4. Conclusiones	54
4.1. Limitaciones	54
4.2. Alcances Futuros	55
A. Descripción del Robot Rhino	56
A.1. Modelo Matemático del Robot Rhino	59
A.2. Cinemática Inversa	61
B. Descripción Física de la Interface de Potencia	63
B.1. Conexiones	63
C. Conversor Digital/Analógico	64
D. Códigos Fuentes	69

Índice de figuras

1.1. Sistema Brazo-Ojo	2
2.1. Funcion de membresía de la variable joven	9
2.2. Función membresía del tipo trapezoidal	9
2.3. Función membresía del tipo triangular	9
2.4. Función membresía del tipo sigmoide	10
2.5. Función membresía del tipo gaussiana	10
2.6. Representación de la altura de los seres humanos con conjuntos difusos	11
2.7. Razonamiento difuso de una variable cualquiera	12
2.8. Esquema de visión	15
2.9. Modelo Pin-Hole	17
2.10. Modelo lente delgada	17
2.11. Formación de imágenes	18
2.12. Cámara digital	18
2.13. Sensor óptico	19
2.14. Frontal direccional	21
2.15. Ángulo bajo	21
2.16. Frontal difusa	22
2.17. Carpa de luz	22
2.18. Retroiluminación colimada	23

2.19. Retroiluminación difusa	23
2.20. Iluminación estructurada	24
2.21. Modelo RGB	26
2.22. Colores aditivos primarios y secundarios	26
2.23. Modelo CMY	27
2.24. Colores substractivos primarios y secundarios	27
2.25. Transformación geométrica en un plano	28
2.26. Ejemplo de segmentación	29
2.27. Segmentación	30
2.28. Esquema del modelo pin-hole	32
3.1. Figura patrón para la calibrar la cámara	39
3.2. Proceso de calibración de la cámara	40
3.3. Resultados de la calibración	40
3.4. Imagen RGB en MatLAB	42
3.5. Preprocesamiento a la imagen	43
3.6. Casos para Y y Y_d	44
3.7. Casos para Y , Y_d y e	46
3.8. Resultado de la simulación del sistema de control en el Hombro	48
3.9. Resultado de la simulación del sistema de control en el Codo	49
3.10. Resultados experimentales con una señal senoidal en el Hombro	50
3.11. Resultados experimentales con una señal senoidal en el Codo	51
3.12. Resultados experimentales con una señal triangular en el Hombro	52
3.13. Resultados experimentales con una señal triangular en el Codo	53
A.1. Robot Rhino XR-3	57
A.2. Dimensiones del Robot Rhino XR-3	58
A.3. Puntos identificados: hombro, codo y muñeca	62

B.1. Interfaz de control del Robot tipo Rhino	63
C.1. Esquemático del conversor analógico/digital	65
C.2. Conversor analógico/digital (vista superior)	66
C.3. Conversor analógico/digital (vista inferior)	67
C.4. Conversor analógico/digital (vista superior con componentes)	68

Índice de tablas

3.1. Eventos de MatLAB	38
3.2. Definición de salidas	45
3.3. Variables lingüísticas de Y y Y_d	45
3.4. Variables lingüísticas de Y , Y_d y e	45
A.1. Asignación de motores	57
A.2. Reducción del giro del engranaje	57

CAPÍTULO 1

Introducción

El uso de máquinas dentro de los procesos de producción ha reducido la participación del hombre de manera paulatina. Es a partir de esto, que surge la necesidad de crear procesos de forma automática.

El control automático se ha vuelto una parte importante e integral de los procesos industriales al aportar los medios para mejorar la productividad. Es así que la teoría desarrollada para el control de procesos, tiene su base en el conocimiento de la dinámica del proceso. Esta dinámica en muchas ocasiones no es exacta, ya que no siempre es posible considerar todos los parámetros del sistema, por la falta de información del mismo.

Ante esta situación, se puede recurrir al uso de otras técnicas de control como el “control inteligente” [1], el cual combina teoría de control con investigación de operaciones e inteligencia artificial para poder emular las capacidades del ser humano. Entre estas técnicas encontramos la lógica difusa, redes neuronales, algoritmos genéticos, etc.

En cuanto a la medición del sistema, se puede utilizar un medio óptico para obtener información de su estado, reconociendo patrones para poder actuar de la manera más adecuada conforme a las tareas que están realizando, como por ejemplo, mover una parte mecánica, ensamblar una pieza, cortar, etc. Estos sistemas son los sistemas denominados “brazo-ojo” [2], los cuales están formados como lo muestra la Figura 1.1.

Estos sistemas se hacen con la finalidad de dotar al sistema con adaptabilidad al medio ambiente debido a que la información que provee un sistema de visión es mucho más completa y rica en comparación a otro medio dedicado a obtener información del entorno.

El uso de este tipo de sistemas se incrementa en los procesos industriales gracias al desarrollo de sistemas de control inteligentes, siendo en la actividad la visión una de las principales áreas de investigación en robótica.

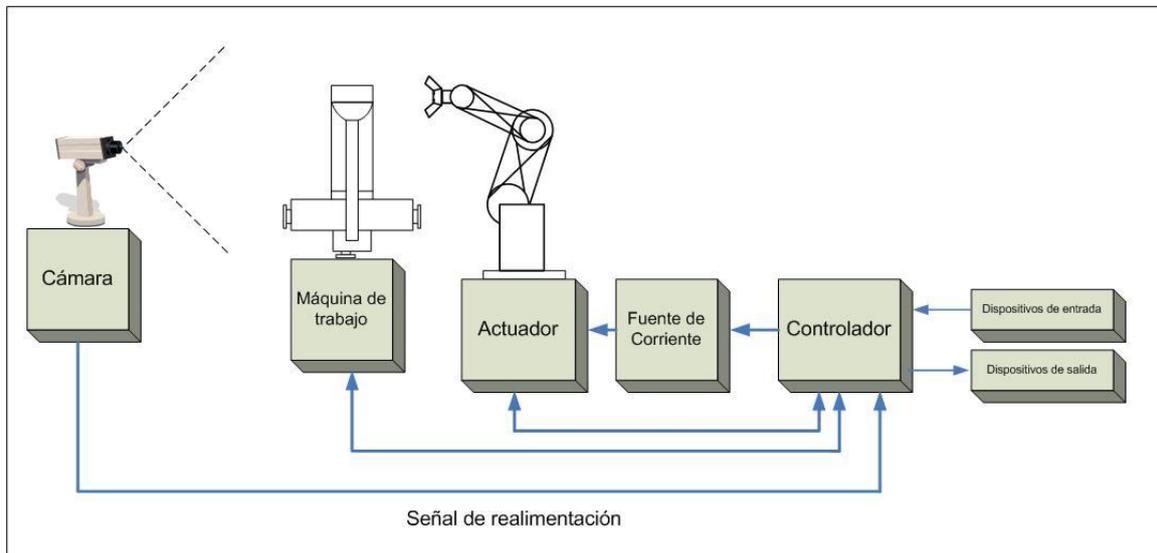


Figura 1.1: Sistema Brazo-Ojo

1.1. Planteamiento del Problema

En la actualidad existen una gran variedad de sistemas autónomos para la realización de tareas determinadas, que son bien aceptados en la industria, estos sistemas tienen una retroalimentación de señales que vienen inmersas en sus componentes electromecánicos, por esta razón la manufactura de este tipo de sistemas es de un costo considerablemente bajo.

El empleo de un sensor óptico para determinar el estado del sistema ha sido una solución y las tendencias están muy marcadas a que en un futuro la mayoría de los sistemas serán controlados por medio de sensores ópticos, por esta razón surge la necesidad de diseñar un mecanismo capaz de controlar un sistema por medio de la retroalimentación óptica.

1.2. Estado del Arte

La idealización de los sistemas “brazo-ojo” está presente desde la década de los 70’s, sin embargo, debido a las limitaciones en cuanto al procesamiento de cómputo para el reconocimiento de patrones, los hacia de muy poca aceptación en la industria.

A pesar de que ya ha pasado mucho tiempo de estas primeras incursiones en este campo, la investigación sigue siendo parte fundamental para el desarrollo del área, y se está utilizando con mayor frecuencia en la industria [3]. Normalmente las aplicaciones son variadas, sin embargo, el acceso a este tipo de sistemas está limitado por su alto costo económico tanto en la adquisición, instalación y mantenimiento, además al operarlo se debe tener un amplio conocimiento del fun-

cionamiento del sistema. Este tipo de sistemas utilizan hardware especial para la adquisición de imágenes, que va desde cámaras de lectura por área, que separan los colores en forma automática, así como aceleradoras gráficas que ayudan a tener una mayor velocidad de procesamiento de estas imágenes.

Las investigaciones en este campo se dividen en tres secciones:

1. Segmentación de imágenes y reconocimiento de patrones.
2. Calibración de la cámara.
3. Control por visión.

La segmentación de imágenes y el reconocimiento de patrones son las partes más importantes en un sistema de visión porque son las encargadas de extraer la información requerida de la imagen. Entre las formas habituales de segmentación, está la segmentación por color, la detección de contornos y la híbrida. Actualmente se trabaja en la segmentación y reconocimiento en 2D y en 3D, utilizando visión monoscópica o estereoscópica [4, 5].

La calibración de la cámara es otro de los aspectos fundamentales de un esquema de visión, ya que a partir de ella, es como podemos hacer la interpretación entre los píxeles de la imagen y las magnitudes reales en las que opera el sistema. Para poder manejar esta propiedad, existen muchos métodos como la calibración de una cámara utilizando un conjunto de puntos coplanares [6], utilizando un conjunto de puntos no coplanares [6], el método de calibración de Ayache [6], el método de calibración de Song De Ma [6], el método de calibración propuesto por Heikkila [7], el método propuesto por E. Trucco [8]. Entre los más utilizados está la propuesta por Tsai [9], sin embargo, a pesar de que los métodos propuestos son efectivos para algunas aplicaciones, tiene un defecto importante que es el que la cámara debe de quedar inmóvil una vez que se ha realizado la calibración. Para resolver esto, hay varias propuestas entre las que podremos destacar la calibración, utilizando redes neuronales como medio de aprendizaje y la calibración a partir de una base de datos de puntos conocidos [10]. Otro aspecto importante en la calibración es la cantidad de cámaras a utilizar en el proceso de visión, Tsai [9] propone métodos de calibración en 3D los resultados no son satisfactorios, por lo que es un campo amplio de investigación aún abierto [4, 5].

Finalmente en los esquemas de control por visión, que son la parte medular del sistema “brazo-ojo”, se hacen estudios actualmente con redes neuronales [11],[12] y con sistemas de bases de datos [10]. Cabe mencionar que, aunque son muchos los estudios, estos son limitados a una tarea o problema en específico.

Se han realizado investigaciones donde se utilizan las tres áreas de investigación antes mencionadas como las siguientes:

- El trabajo realizado por Chávez [13] el cual es un sistema manipulador de 5 grados de libertad, y un efector final que funciona por gravedad para cerrar o abrir su pinza, utiliza una

cámara CMOS para capturar la imagen. El sistema fue desarrollado para hacer una aplicación, que coordina el movimiento del brazo con la imagen recibida de la cámara.

- El trabajo realizado por Park [14] el cual utiliza un sistema "brazo-ojo" y coloca una cámara en el efector final y utiliza un tipo de iluminación especial para reconocer una superficie discontinua, y puede reconocer un defecto o una corrección en la superficie.
- El trabajo realizado por Zhao [15] utiliza un nuevo algoritmo de calibración, utilizando la teoría de movimiento de rotación para establecerlo en una ecuación matricial "brazo-ojo" y obtener un resultado simultáneo para la rotación y traslación para resolver la ecuación lineal. El algoritmo propuesto tiene una alta y estable eficiencia computacional.
- El trabajo realizado por Wang [16] el cual describe la configuración y la implementación de un sistema "brazo-ojo" usando la información visual de dos cámaras. Describe varias cuestiones con respecto a la detección, control y configuración del sistema, como pueden ser la segmentación de imágenes, la calibración, utilizando con mayor énfasis la detección de ejes y la segmentación de imágenes por color.

1.3. Objetivos Generales

Realizar un algoritmo que permita reconocer el estado en que se encuentra un sistema, por medio de una secuencia de imágenes obtenidas por una cámara en tiempo real, así mismo diseñar un algoritmo de control que permita manipular el sistema en todo instante, y crear un sistema de adquisición de datos económico y al alcance de un rango amplio de usuarios.

1.4. Objetivos Particulares

- Diseñar un algoritmo de visión, capaz de reconocer los elementos de interés de un sistema.
- Diseñar un algoritmo de control por medio de lógica difusa para el sistema, para que realice un seguimiento de trayectorias.

1.5. Organización del Trabajo

- En el segundo capítulo se proporcionan conceptos y antecedentes necesarios para el análisis del sistema propuesto.
- En el tercer capítulo se diseñará un algoritmo de reconocimiento de patrones y un esquema de control difuso para el sistema propuesto.

- En el cuarto capítulo se presentan las conclusiones obtenidas en el trabajo y algunos lineamientos para trabajos futuros.

CAPÍTULO 2

Preliminares

2.1. Control Difuso

El Control Difuso es una técnica del control que utiliza la Lógica Difusa (LD) para aproximar alguno de los elementos de la ley de control de un sistema. Buscando reproducir de alguna forma la experiencia de un usuario humano para controlar un sistema.

La Lógica Difusa permite tratar información imprecisa, como “caliente”, “tibio”, “frío”, “poco”, “mucho”, etc., en términos de conjuntos difusos. Estos conjuntos difusos se combinan en reglas para definir acciones, por ejemplo “*Si* la velocidad es baja, *entonces* frena moderadamente”. De esta manera, los sistemas de control basados en lógica difusa combinan variables de entrada, por medio de grupos de reglas que producen uno o varios valores de salida.

Mucho del lenguaje es ambiguo y tiene una considerable multiplicidad de significados. Los adjetivos, no son claramente especificados en términos de distinguir claramente el significado. Por ejemplo, si decimos “persona joven”, no podemos claramente determinar quien es joven o quien no es joven

2.2. Lógica Difusa

Las bases de la Lógica Difusa se presentan a principios de los años 60's por Lofti A. Zadeh, profesor de Ingeniería Eléctrica en la Universidad de California, en su primera publicación de 1965 [17], pero es hasta 1973 donde presenta teoría básica de controladores difusos. En 1974, Ebrahim Madmani demuestra una aplicación de la Lógica Difusa desarrollando el primer sistema de control difuso práctico, al regular un motor de vapor.

La lógica difusa, es una lógica alternativa a la lógica clásica que pretende introducir un grado de vaguedad en las cosas que califica. En el mundo real existe una gran cantidad de conocimiento vago, incierto o ambiguo. El razonamiento y pensamiento humano frecuentemente lleva información de este tipo, probablemente originada de la inexactitud inherente de los conceptos humanos.

Para la lógica clásica una proposición sólo admite dos valores, 0 ó 1 (pertenece o no pertenece), con lo que surge un problema debido a la poca capacidad de expresión dentro de la realidad en la que vivimos donde hay muchos conceptos vagos que manejamos comúnmente, por ejemplo el concepto de “juventud”. La solución que presenta la lógica clásica es definir un umbral de pertenencia (por ejemplo, un valor que todo el mundo considera que de ser alcanzado o superado, la persona en cuestión puede llamarse vieja). Si dicho umbral es de 35 años, todas las que tengan 35 años o más serán consideradas personas viejas, mientras que las otras serán consideradas personas jóvenes. Según esta manera de pensar, alguien que tenga 34 será tratado igual que otro que tenga 15 ya que ambos han merecido el calificativo de jóvenes. Sin embargo, no hay un valor cuantitativo que defina el término joven porque para algunas personas 25 años es joven, mientras que para otras 35 años es joven. Incluso el concepto puede ser relativo al contexto. Un presidente de 35 años es joven, mientras que un futbolista no lo es. A pesar de esto, existen valores claros: una persona de 1 año es joven, mientras que una persona de 100 años no lo es, pero una persona de 35 años tiene algunas posibilidades de ser joven (según el contexto). Para representar este hecho, definiremos el conjunto joven de modo que cada uno de sus elementos pertenezcan a él con cierto grado. De un modo más formal, un conjunto difuso A se caracteriza por una función de pertenencia:

$$\mu_A : U \rightarrow [0, 1] \quad (2.1)$$

Que asocia a cada elemento x de U un número $\mu_A(x)$ del intervalo $[0,1]$, que representa el grado de pertenencia de x al conjunto difuso A . A U se le llama universo de discurso. De este modo la Lógica Difusa permite a los sistemas manejar información que tiene un alto grado de imprecisión, al contrario de la lógica tradicional que maneja información precisa y definida, de tal manera que para la lógica difusa son posibles las condiciones intermedias entre los dos estados: “niño”, “adolescente”, “adulto”, etc.

En la actualidad, es un campo de investigación muy importante, tanto por sus implicaciones matemáticas o teóricas como por sus aplicaciones prácticas.

2.3. Conjuntos Difusos

En los conjuntos difusos, a diferencia de los conjuntos clásicos, se relajan la restricción de que la función de pertenencia valga 0 ó 1, y dejamos que tome valores en el intervalo $[0,1]$. La necesidad de trabajar con conjuntos difusos surge de un hecho: hay conceptos que no tienen límites claros. Por ejemplo: ¿Una persona que mide 1.80 es alta? ¿Una temperatura de 15 grados es baja?. A diferencia de lo que ocurre en el caso de un conjunto de frutas (no hay vaguedad, un alimento o

bien es una fruta o bien no lo es), en otras situaciones nos vemos obligados a tratar con ella.

Veremos unas definiciones útiles para después dar una definición de conjunto difuso.

Variables Lingüísticas

Llamaremos variable lingüística a aquella noción o concepto que vamos a calificar de forma difusa. Por ejemplo “la altura”, “la edad”, “el error”, “la variación del error” le aplicamos el adjetivo “lingüística” porque definiremos sus características mediante el lenguaje hablado.

Universo de Discurso

Se le llama Universo de Discurso al rango de valores que pueden tomar los elementos que poseen la propiedad expresada por la variable lingüística. En el caso de la variable lingüística “edad de los humanos”, sería el conjunto de valores comprendido entre 0 y 100 años.

Valor Lingüístico

Llamamos Valor Lingüístico a las diferentes clasificaciones que efectuamos sobre la variable lingüística, por ejemplo “edad”, podríamos dividir el universo de discurso en joven, adulto y viejo.

Funciones Membresía

Una función membresía¹ es aquella que asocia a la variable de entrada el grado de pertenencia² de la variable lingüística asociada. Por ejemplo, tenemos la variable lingüística “joven” con valores de pertenencia de 0 a 1, donde 0 significa ningún tipo de asociación y 1 significa completamente asociado. Podríamos pensar que 10 años de edad es “joven”, por lo que le asociamos a esta edad el 1, pero también sabemos que a los 40 años una persona es también “joven” por lo que le asociamos un valor de 0.3 de pertenencia, y así sucesivamente para cualquier edad. Esto se puede observar en la Figura 2.1.

Para la definición de las funciones membresía se utilizan ciertas funciones normalizadas por coincidir con el significado lingüístico de las etiquetas más utilizadas. Las más utilizadas son del tipo triangular, trapezoidal, sigmoide y gaussiana, sin embargo cualquier función puede ser una función membresía siempre que sus valores esten dentro del rango de $[0,1]$, o $[-1,1]$.

La función trapezoidal se emplea habitualmente en sistemas difusos sencillos, pues permite definir un conjunto difuso con pocos datos y calcular su valor de pertenencia con pocos cálculos.

¹También es conocida como función de pertenencia

²Con que grado se encuentra incluido en el concepto

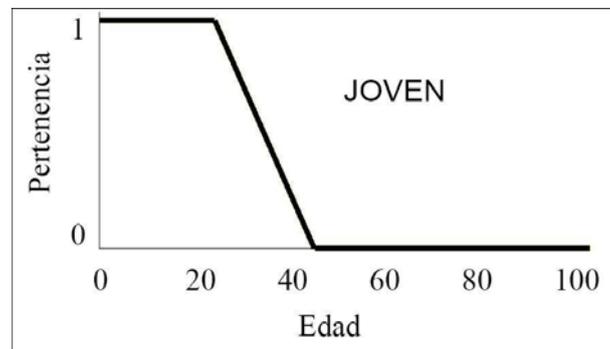


Figura 2.1: Funcion de membresía de la variable joven

Se emplea especialmente en sistemas donde se van a utilizar microprocesadores por su facilidad de programación. Se define como se observa en la Figura 2.2 :

$$y = \begin{cases} 0 & \text{si } x \leq a \\ \frac{x-a}{b-a} & \text{si } x \in (a, b] \\ 1 & \text{si } x \in (b, c) \\ \frac{d-x}{d-c} & \text{si } x \in (c, d) \\ 0 & \text{si } x \geq d \end{cases}$$

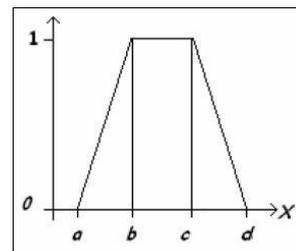


Figura 2.2: Función membresía del tipo trapezoidal

La función triangular es adecuada para modelar propiedades con valor de inclusión distinto de cero para un rango de valores estrecho en torno a un punto m. Se define como se observa en la Figura 2.3:

$$y = \begin{cases} 0 & \text{si } x \leq a \\ \frac{x-a}{m-a} & \text{si } x \in (a, m] \\ \frac{b-x}{b-m} & \text{si } x \in (m, b) \\ 0 & \text{si } x \geq b \end{cases}$$

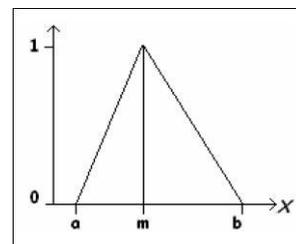


Figura 2.3: Función membresía del tipo triangular

La función sigmoide es adecuada para modelar propiedades con un amplio universo de discusión como “grande”, “positivo”, “mucho”, etc. Se define como se observa en la Figura 2.4:

La función gaussiana tiene forma de campana, y resulta adecuada para conjuntos definidos en torno a un valor m, como “medio”, “normal”, “cero”, etc. Se define como se observa en la Figura 2.5:

$$y = \frac{1}{1 + e^{d(x-m)}}$$

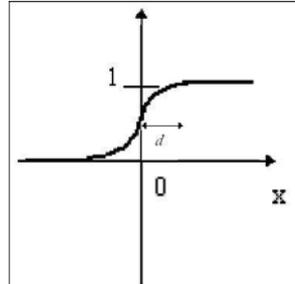


Figura 2.4: Función membresía del tipo sigmoide

$$y = e^{-k(x-m)^2}$$

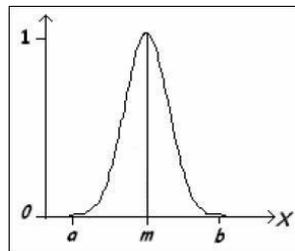


Figura 2.5: Función membresía del tipo gaussiana

Definición de los Conjuntos Difusos

Un conjunto difuso es un valor lingüístico junto a una función de membresía. Decimos que un conjunto es nítido si su función de membresía toma los valores 0 ó 1, y difuso si toma valores en el rango $[0,1]$.

Un conjunto difuso se define matemáticamente como:

$$A = \{x, \mu_A(x) | x \in U\} \quad (2.2)$$

Donde:

A es la variable lingüística

$\mu_A : x \rightarrow [0, 1]$ es la función de membresía

$\mu_A(x)$ es la función de membresía de la variable lingüística x

U es el universo de discurso

Los conjuntos difusos permiten agrupar objetos o sucesos por el valor de una cierta magnitud. Por ejemplo, consideramos la “Altura de los humanos” como una forma de agrupar a un conjunto

de personas, si definimos un conjunto clásico de las personas de estatura “baja” como las que miden 1.5 metros, resultaría que una persona que mide 1.49 metros es una persona baja mientras que una persona que mide 1.51 metros ya no lo es. Esta explicación que da la teoría de conjuntos clásica no es satisfactoria, ya que la diferencia de estaturas es de sólo 2 centímetros. En este caso una descripción en términos de conjuntos difusos resulta mas apropiada porque podemos considerar la variable ligüística “Altura de los humanos”, que toma valores en el universo de discurso de $U = [1.4, 2.5]$ y los términos “bajo”, “mediano” y “alto” que podemos representar mediante funciones de membresía que al variar de 0 a 1 nos indicarían si una persona es “baja”, “medianamente baja”, “ligeramente mediana”, etc. (Figura 2.6).

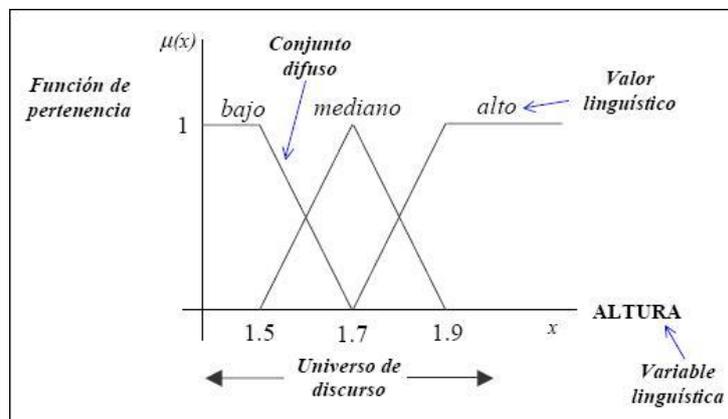


Figura 2.6: Representación de la altura de los seres humanos con conjuntos difusos

Operaciones Lógicas con Conjuntos Difusos

En la teoría clásica de conjuntos existen operaciones, AND (intersección), OR (unión), NOT (complemento), etc., los cuales son únicos y definidos. En Lógica Difusa existen también esta clase de operaciones con conjuntos, sin embargo, no todas las operaciones son únicas, particularmente en las operaciones AND y OR. Los más usados para estas operaciones son los propuestos por Zadeh (2.3, 2.4, 2.5), el producto difuso (2.6) y el operador OR de Lukasiewicz (2.7).

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (2.3)$$

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad (2.4)$$

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (2.5)$$

$$\mu_{A \cap B}(x) = \min(\mu_A(x) \times \mu_B(x)) \quad (2.6)$$

$$\mu_{A \cup B}(x) = \max(\mu_A(x) + \mu_B(x), 1) \quad (2.7)$$

2.3.1. Control Difuso

El control difuso es una aplicación de la teoría de la Lógica Difusa, dado que con ella se tratan de crear aproximaciones matemáticas para la resolución de cierto tipo de problemas y producir resultados exactos a partir de datos imprecisos.

Los controles difusos son utilizados cuando el proceso a controlar es complejo, no lineal y/o su modelo matemático es difícil de obtener, por estos motivos se utiliza la información obtenida a través de la experiencia para crear las reglas de control.

Razonamiento Difuso

El razonamiento difuso³ es un procedimiento matemático que asigna en el antecedente de cada declaración (regla) difusa, un grado de pertenencia entre 0 y 1 por la función de membresía asociada a cada elemento (operando) en dicho antecedente.

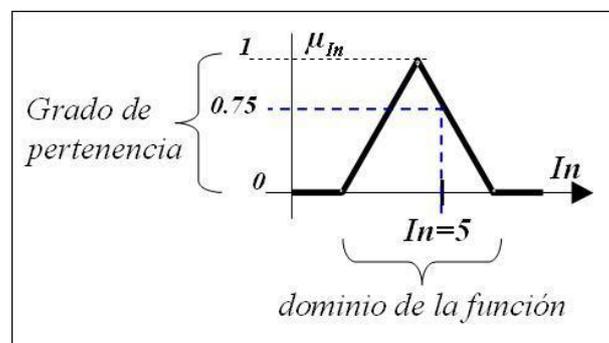


Figura 2.7: Razonamiento difuso de una variable cualquiera

Reglas Difusas

Las reglas difusas combinan uno o más conjuntos difusos de entrada, llamados antecedentes y se les asocia un conjunto de salida llamado consecuente. Los conjuntos difusos de la premisa se asocian mediante sentencias lingüísticas del tipo *SI-ENTONCES (IF-THEN)*, entonces para un sistema de control tendríamos “SI velocidad es *alta* Y SI curva es *pronunciada*, ENTONCES el frenado es *enérgico*”.

³La bibliografía en inglés lo describe como “Fuzzification” y en español como “Borrosidad”

Las reglas difusas permiten expresar el conocimiento que se dispone acerca de la relación entre antecedentes y consecuentes, para expresar este conocimiento de forma completa, generalmente se disponen de varias reglas.

Son dos los tipos de reglas difusas que existen: Mamdani y Takagi-Sugeno.

Reglas difusas del tipo Mamdani

La estructura de las reglas difusas del tipo Mamdani es:

$$\text{SI } v_1 \text{ es } S_1 \text{ Y } \dots \text{ Y } v_M \text{ es } S_M \text{ ENTONCES } z_1 \text{ es } \beta_1, \dots, z_p \text{ es } \beta_p \quad (2.8)$$

Donde $v_i, i = 1, \dots, M$ es una variable de entrada y $z_j, j = 1, \dots, P$ es una variable de salida. En este tipo de reglas, tanto S_j como β_p son funciones membresía. Por lo tanto el antecedente como el consecuente son expresiones difusas.

Reglas difusas del tipo Takagi-Sugeno

La estructura de las reglas difusas del tipo Takagi-Sugeno es:

$$\text{SI } v_1 \text{ es } S_1 \text{ Y } \dots \text{ Y } v_M \text{ es } S_M \text{ ENTONCES } z_1 = f_1(v_1, \dots, v_m), \dots, z_P = f_P(v_1, \dots, v_m) \quad (2.9)$$

Donde $v_i, i = 1, \dots, M$ es una variable de entrada, $z_j, j = 1, \dots, P$ es una variable de salida y $f_P()$ es una función de cualquier tipo, aunque, generalmente se suele utilizar una combinación lineal de las entradas o bien una constante. El antecedente son expresiones difusas, pero el consecuente son funciones determinísticas.

Decodificación

Existen diversas alternativas para transformar un valor difuso en nítido ⁴, las más empleadas son[18]:

1. **Centro del área:** Es el más conocido método de decodificación, donde u_i son las variables

⁴Este proceso que en inglés se llama defuzzification, en español podríamos llamar decodificación

de entrada y $\mu_U(u_i)$ son las salidas de control:

$$z = \frac{\sum_{i=1}^l u_i \cdot \mu_U(u_i)}{\sum_{i=1}^l \mu_U(u_i)} \quad (2.10)$$

2. **Centro de Sumas:** Un método más rápido ya que se trata de evitar el cómputo de la unión de las salidas de control (\tilde{U})

$$z = \frac{\sum_{i=1}^l u_i \cdot \sum_{k=1}^n \mu_k(u_i)}{\sum_{i=1}^l \sum_{k=1}^n \mu_k(u_i)} \quad (2.11)$$

3. **Peso:** Este método toma el valor del pico más alto de cada salida de control y construye la suma ponderada de esos valores pico. Donde c^k son los valores pico y f_k es el peso de las salidas de control

$$z = \frac{\sum_{k=1}^m c_k \cdot f_k}{\sum_{k=1}^n f_k} \quad (2.12)$$

4. **Primera máxima:** Usa las salidas de control y toma el menor valor en el dominio de U con máximo grado de pertenencia en \tilde{U}

$$z = \inf_{u \in U} \{u \in U | \mu_U(u) = \text{hgt}(U)\} \quad (2.13)$$

5. **Última máxima:** Es un método alternativo al anterior

$$z = \sup_{u \in U} \{u \in U | \mu_U(u) = \text{hgt}(U)\} \quad (2.14)$$

2.4. Visión Artificial

La visión artificial tiene como objetivo la reproducción artificial del sentido de la vista humana. Los humanos obtienen la mayor parte de su entrada sensorial a través del sentido de la vista, y se ha hecho un enorme esfuerzo para mejorar de forma artificial este sentido. Lentes, binoculares, telescopios, radares, sensores infrarrojos; todos funcionan para mejorar nuestra vista del mundo y del universo en el que nos encontramos. A raíz de esto surge la idea de utilizar computadoras para incrementar la capacidad en este sentido[19].

En la visión artificial se distinguen tres niveles de procesamiento de imágenes, de bajo, intermedio y alto nivel. El procesamiento de imagen de bajo nivel, cuenta con procesos que son primarios en el sentido que pueden ser considerados “reacciones automáticas” sin requerir de ningún tipo de inteligencia, en esta etapa se encuentra la adquisición de imágenes, transformadas para las imágenes, la restauración y la modificación geométrica de las imágenes. En el procesamiento de nivel intermedio se encuentra lo que es la detección de contornos y fronteras para las imágenes y la segmentación espacial de imágenes. Por último, el procesamiento de alto nivel se refiere a procesamientos que tratan de emular la cognición y la compresión de imágenes fijas y en movimiento.

2.4.1. Esquema de Visión

En esta sección se describe el esquema de visión utilizado en este trabajo, en la Figura 2.8 se muestra el diagrama de bloques empleado.

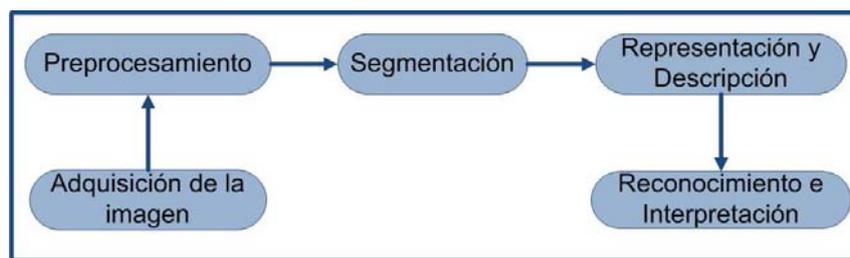


Figura 2.8: Esquema de visión

- El primer paso es la adquisición de la imagen; para lograrlo, se requiere de una cámara y la capacidad de digitalizar la señal producida por ella.
- Después de que una imagen digital ha sido obtenida, el siguiente paso es el preprocesamiento de la imagen.
- La siguiente etapa es la segmentación en donde se realiza una partición de la imagen en sus partes constituyentes u objetos.
- Los datos obtenidos en la etapa de segmentación deben ser convertidos a una estructura apropiada para su posterior procesamiento. La descripción, también llamada selección de características, extrae información cuantitativa o características relevantes para diferenciar un objeto de los demás.
- La última etapa es el reconocimiento e interpretación. El reconocimiento es el proceso que asigna un identificador a un objeto basado en la información provista por sus descriptores. La interpretación es asignar un significado al objeto reconocido. En general, los sistemas de procesamiento que incluyen reconocimiento e interpretación están asociados a aplicaciones de análisis de imágenes cuyo objetivo es la extracción automática de información

Adquisición de Imágenes

En esta etapa se convierte la información visual en señal eléctrica por medio de sensores. Cuando las señales eléctricas se muestrean espacialmente y se cuantifican en amplitud, obtenemos de ellas una imagen digital.

Se debe tener en cuenta que para alcanzar el objetivo de adquirir imágenes digitales se requieren elementos de hardware y software. Los elementos de hardware mínimos que se necesitan para un sistema de visión artificial son los siguientes:

- Óptica
- Cámara
- Sensores Visuales
- Iluminación del entorno
- Transmisión de la señal
- Digitalizadores
- Computadora

Estos elementos funcionan en conjunto para la adquisición de imágenes de la siguiente forma. El objetivo de la iluminación es proporcionar luz suficiente para obtener una buena imagen de la escena, los rayos de luz provenientes de los objetos pasan a través de una óptica y son captados por un sensor visual (esto es realizado por la cámara), la información de luz es transformada a una información eléctrica analógica, la información eléctrica es capturada por un digitalizador y transformada en una matriz de valores, con lo cual se obtiene una matriz de $M \times N$ que representa la imagen adquirida.

Óptica

La óptica es captar los rayos luminosos para proyectarlos en el sensor de imagen. Los modelos de óptica comúnmente usados son:

- Modelo Pin-Hole
- Modelo de lente delgada

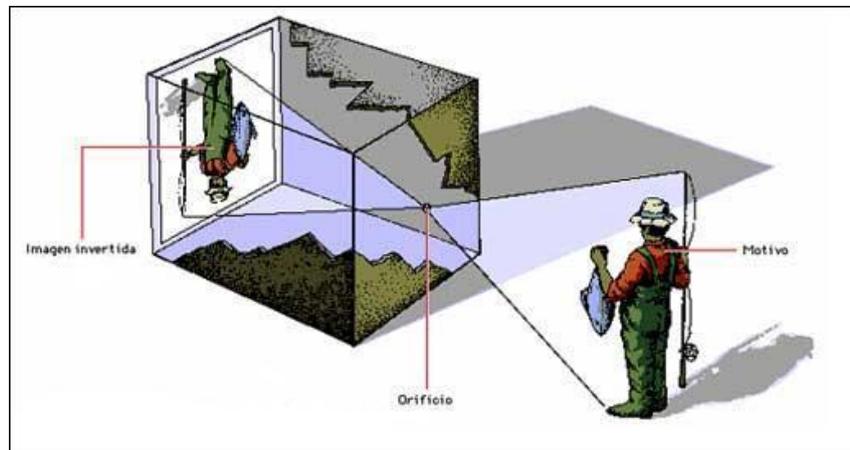


Figura 2.9: Modelo Pin-Hole

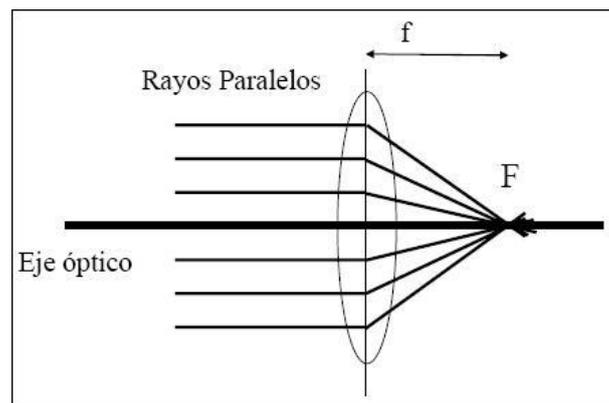


Figura 2.10: Modelo lente delgada

Modelo Pin-Hole

En este modelo se supone que un rayo luminoso parte de cada punto 3D del mundo y se proyecta en un único punto del plano de imagen, como se muestra en la Figura 2.9.

Se supone que un rayo luminoso parte de cada punto 3D del mundo y se proyecta en un único punto del plano imagen, también se supone una abertura infinitesimal, y aparte este modelo es de los más utilizados por las cámaras y nos permite entender el proceso de formación de imágenes, este modelo se describirá más a profundidad en la sección de calibración de una cámara.

Modelo lente delgada

En este modelo todos los rayos paralelos al eje óptico convergen al foco F del lente, un rayo que atraviesa el centro de la lente no sufre desviación, Figura 2.10.

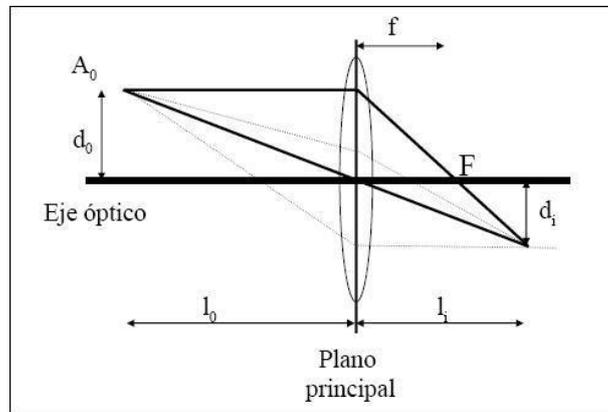


Figura 2.11: Formación de imágenes

Para la formación de las imágenes, los rayos de luz que atraviesan el centro de la lente no sufren desviación, en cambio los rayos que son paralelos al eje óptico, convergen al foco, en la Figura 2.11 se muestra como se genera una imagen.

Cámara de Video

Las cámaras de video son los dispositivos más empleados para la visión artificial. Cuentan con una lente, una abertura y un sensor óptico. La lente es de vidrio y toma los rayos de luz y los enfoca en la cámara. La abertura es un pequeño orificio que controla la cantidad de luz que entra a la cámara. El sensor óptico que transforma la cantidad de iluminación a señales de voltaje estandarizada, este sensor puede ser un Dispositivo de Acoplamiento de Carga (CCD) o un Active Píxel Sensor (APS).

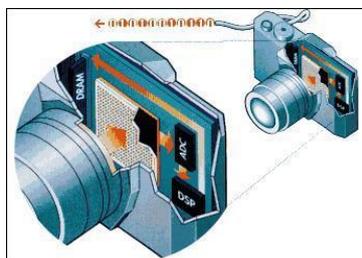


Figura 2.12: Cámara digital

Sensores Visuales

Los CCD son circuitos integrados que contienen un cierto número de elementos fotosensibles, elementos de silicio llamados photosites, enlazados o acoplados, en forma de una matriz rectangu-

lar. Los photosites transforman la cantidad de iluminación en una carga de voltaje. Bajo el control de un circuito interno, cada celda fotosensible puede transferir su carga a uno o varios capacitores que están a su lado en el circuito impreso. Después se pasa los voltajes de los capacitores a un convertidor analógico-digital, el cual permite digitalizar la señal de video entrante.

El Active Píxel Sensor (APS) es un sensor que detecta la luz basado en tecnología CMOS (Complementary Metal Oxide Semiconductor) y por ello más conocido como Sensor CMOS. Por la tecnología CMOS es posible integrar más funciones en un chip sensor, como por ejemplo control de luminosidad, corrector de contraste, o un convertidor analógico-digital.

Los dos tipos de sensores son muy utilizados en las cámaras digitales, aunque los CCD son más utilizados en aplicaciones profesionales y los sensores CMOS en numerosas WebCam (Figura 2.13).

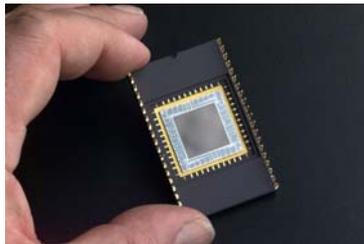


Figura 2.13: Sensor óptico

Ambos sensores funcionan de manera muy similar, aunque tienen algunas diferencias significativas como el hecho de que el sensor CMOS cuenta con un convertidor analógico-digital dentro del mismo encapsulado, en cambio el CCD envía las señales eléctricas al convertidor analógico-digital exterior al encapsulado.

Según los fabricantes de CCD's, los sensores CMOS tienen un elevado patrón de ruido fijo, pero sus defensores indican que tienen un bajo consumo de energía (lo cual redundaría en una mayor autonomía de la cámara). Al parecer, el 'ruido' mencionado se debe al que los sensores CMOS convencionales tienen un amplificador por separado en cada píxel y estos amplificadores normalmente no serán uniformes por todo el chip y la desigualdad residual será la que genere el ruido. Por el contrario, todos los píxeles de un CCD se activan a través de una etapa común del amplificador, de modo que se evita este problema.

Por otro lado, los fabricantes de CMOS argumentan que los sensores CCD necesitan una electrónica externa compleja por lo cual se eleva el costo de estos sensores.

Iluminación del Entorno

El objetivo de la iluminación es resaltar aquellas características del objeto (defectos) de manera que sean fácilmente detectables por el sistema de visión. La iluminación depende principalmente de la aplicación.

Tipos de fuentes luminosas

- Iluminación solar
- Lámparas de filamento incandescente
- Lámparas halógenas
- Tubos fluorescentes
- LED
- Láser
- Fibra óptica

Existe una gran variedad de formas de iluminar un objeto, cada una de ellas facilita el procesado de determinados aspectos a inspeccionar. Aunque inicialmente un dispositivo esté especialmente pensado para originar una determinada iluminación, un mismo sistema puede ser usado de múltiples formas para iluminar, dependiendo de su colocación final.

Tipos básicos de iluminación:

- Direccional
 - Frontal direccional
 - De ángulo bajo
- Difusa
 - Frontal difusa
 - Carpa de luz
- Retroiluminación
 - Retroiluminación colimada
 - Retroiluminación difusa
- Estructurada

Frontal Direccional

La luz producida por la lámpara llega directamente al objeto (Figura 2.14). Produce un gran contraste y resalta las texturas, relieves y fisuras del objeto iluminado debido a que cualquier relieve, por pequeño que sea, produce una sombra muy definida. El ángulo de incidencia de la luz respecto al plano de iluminación determinará el grado de resalte de los relieves. Para ángulos muy pequeños respecto a la horizontal, la luz producirá sombras en los relieves de la pieza. Para ángulos cercanos a 90 grados respecto a la horizontal, sólo será detectable la sombra en los grandes relieves.

Su uso está indicado en la detección de presencia de piezas y objetos y su correcta colocación y en la detección de bordes, rayas y fisuras en una dirección determinada.

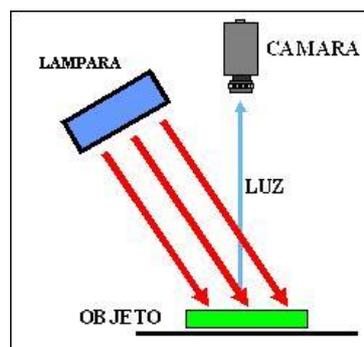


Figura 2.14: Frontal direccional

De Ángulo Bajo

Incide en el mismo plano que la pieza. Se utiliza para detectar pequeños defectos, rasguños. Típicamente se realiza con coronas luminosas y líneas de luz. Las ventajas de este tipo de iluminación es que resalta pequeños defectos del objeto, pero las desventajas es que produce una iluminación irregular (Figura 2.15).

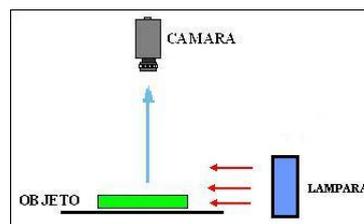


Figura 2.15: Ángulo bajo

Frontal Difusa

Este sistema de iluminación está especialmente indicado para atenuar brillos. Suaviza las texturas de los objetos, iluminando éstos uniformemente. Las lámparas están diseñadas para ser colocadas en cualquier posición excepto en el eje de la cámara. Permite así variar la sombra proyectada dependiendo de las necesidades de la pieza a inspeccionar. Indicado para poder detectar fácilmente variaciones del color como las que se producen al imprimir texto en un objeto (Figura 2.16).

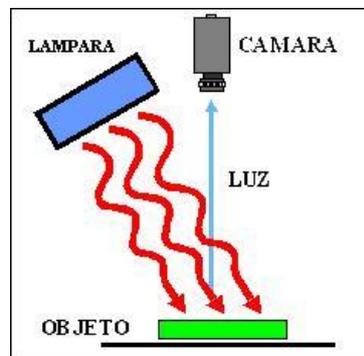


Figura 2.16: Frontal difusa

Carpa de Luz

El objeto es iluminado desde todas las direcciones con luz difusa, eliminando las sombras y reflejos, suavizando las texturas y minimizando la influencia de las rayas, el polvo, y los relieves y curvaturas que pueda tener el objeto inspeccionado. Indicado para la detección de marcas de diferentes colores, caracteres y detección de todo lo que suponga un cambio de color tanto en superficies lisas, esféricas, rugosas o con brillo (Figura 2.17).

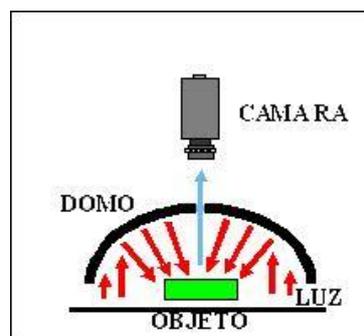


Figura 2.17: Carpa de luz

Retroiluminación Colimada

La cámara se dirige directamente a la fuente de luz, se coloca una lente delante de la fuente de luz para que despida rayos de luz paralelos al eje de la cámara, la ventaja que tiene es que produce bordes muy resaltados, y la desventaja es su difícil implementación. (Figura 2.18)

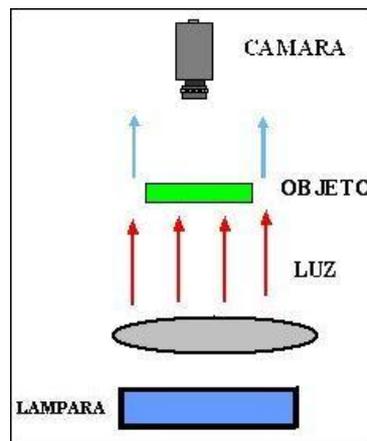


Figura 2.18: Retroiluminación colimada

Retroiluminación Difusa

La cámara se dirige directamente a la fuente de luz, observando una superficie blanca totalmente uniforme, cualquier objeto interpuesto entre la cámara y la fuente de luz produce una sombra que es detectada por la cámara como una forma en negro sobre el fondo blanco producido por el iluminador, obteniendo el máximo contraste posible. Aplicado en la medición precisa de los perfiles de las piezas. Se emplea también en la medición de transmitancia y de impurezas en los objetos transparentes o translúcidos (Figura 2.19).

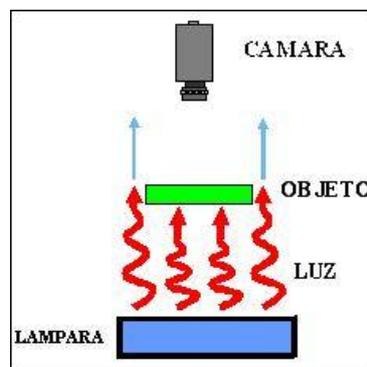


Figura 2.19: Retroiluminación difusa

Iluminación Estructurada

Con este tipo de iluminación el plano de luz es generado por un láser, es útil para obtener una percepción 3D de la escena, y obtienes un alto contraste de los objetos, la desventaja que tiene es el alto costo (Figura 2.20).

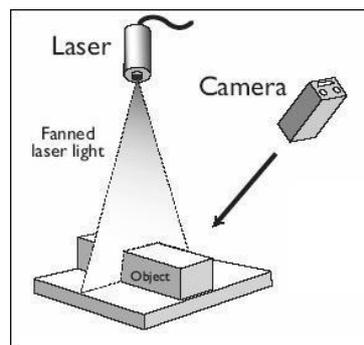


Figura 2.20: Iluminación estructurada

Transmisión de la señal

La transmisión de la señal de video puede darse de dos maneras, analógica o digital.

Digitalizadores

El propósito de los digitalizadores es la conversión de una señal analógica a una señal digital almacenable en una computadora. Algunos fabricantes son MATROX, IMAGING TECHNOLOGY, DATACUBE, NATIONAL INSTRUMENTS, etc.

- Características:
 - Señales de entrada
 - Tipo de señal:
 - ◇ Video compuesto, Señal estándar (PAL, NTSC, etc.)
 - ◇ B/N o Color, Entrada de video digital
 - ◇ N° Canales de entrada
 - Velocidad de transferencia
 - Bus de conexión
 - Memoria disponible en el digitalizador
 - Capacidad de procesamiento
 - Software de programación

Computadora

Es el último elemento de hardware empleado en la adquisición de imágenes. Una vez digitalizada la imagen, debe ser almacenada en la memoria principal de una computadora para su posterior procesamiento y manipulación por medio de software específico.

Imágenes Blanco/Negro y Color

Independiente al tipo de sensor, la imagen que va a ser manipulada por la computadora se presenta digitalizada espacialmente en forma de matriz con una resolución de $M \times N$ elementos. Cada elemento de la matriz o píxel tiene un valor de luminosidad correspondiente al tipo de escala utilizada.

Si la imagen es en blanco y negro, el valor almacenado por cada píxel es un nivel de intensidad o nivel de gris. Generalmente se utilizan 8 bits para representar cada nivel de gris teniendo 2^8 combinaciones, que son 256 niveles de gris, esto es una gama del 0 decimal o 00000000 binario para el negro y 255 decimal o 11111111 para el blanco.

Si la imagen es a color, los elementos de la matriz son representados por tres valores, los cuales representan cada uno de los componentes básicos del color. Los componentes son el Rojo (R), Verde (G) y el Azul (B), mejor conocido como código RGB. También se utilizan 8 bits para cada componente del color esto es 8 bits para el rojo, 8 para el verde y 8 para el azul, teniendo así 16, 777, 216 [20] combinaciones de colores, como por ejemplo (0, 0, 0) negro, (255, 255, 255) blanco, (255, 0, 0) rojo, (0, 255, 0) verde y (0, 0, 255) azul.

Fundamento de Color

El uso del color en el procesamiento de imágenes es usado principalmente porque el color es un potente descriptor. Un descriptor, en el análisis de imágenes, es un elemento que simplifica la identificación y extracción de objetos en una escena.

Modelo de Color

Un modelo de color consiste en la especificación de un sistema de coordenadas tridimensionales y de un subespacio dentro de ese sistema, donde cada color está representado por un único punto.

Modelo RGB

Este modelo consiste en usar los colores rojo (R), verde (V) y azul (B), para formar toda una gama de colores, y se representa mediante un sistema cartesiano cuyo subespacio es un cubo de unidad 1 por lado, como se muestra en la Figura 2.21.

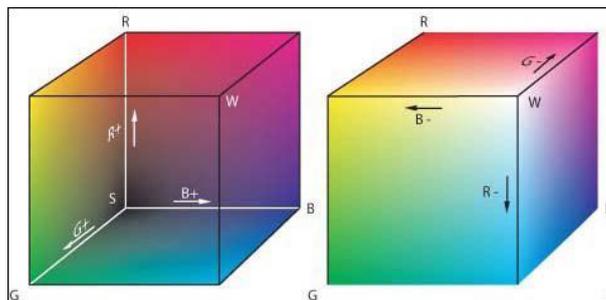


Figura 2.21: Modelo RGB

El origen de coordenadas representa el negro, cada eje esta asociado a un color (rojo, verde, azul), al superponerse los colores primarios se crean los colores secundarios Cian, Magenta y Amarillo (Figura 2.22).

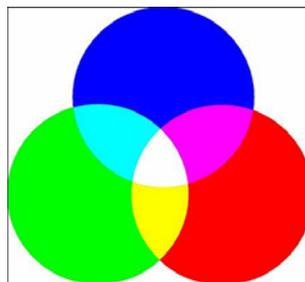


Figura 2.22: Colores aditivos primarios y secundarios

El sistema RGB será útil siempre que se quiera visualizar imágenes mediante adición de colores, como es el caso de los monitores y sistemas de televisión o las imágenes multispectrales.

Presenta una desventaja para aplicaciones de visión por computadora, debido a que no se puede separar la luminosidad del color, es decir, se necesita de los tres componentes para poder obtener la luminosidad, haciendo difícil la adaptación de la aplicación a condiciones de luz variable.

Modelo CMY

En este modelo, cuenta con un subespacio con un cubo de unidad sobre el sistema de coordenadas cartesianas.

Este modelo en diferencia al modelo RGB, el origen del sistema representa el color blanco y cada uno de los ejes esta asociado a los colores cian, magenta y amarillo, como se muestra en la Figura 2.23.

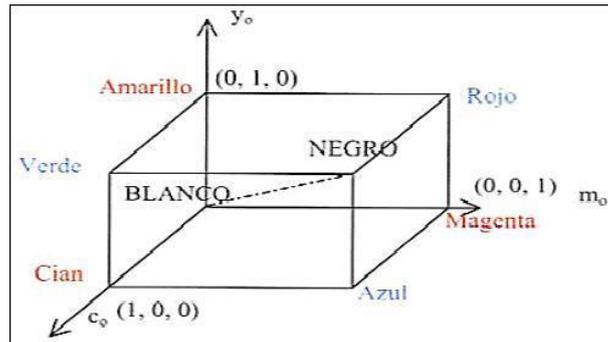


Figura 2.23: Modelo CMY

Se basa en las propiedades de absorción de luz de la tinta impresa en papel y se utilizan los colores substractivos.

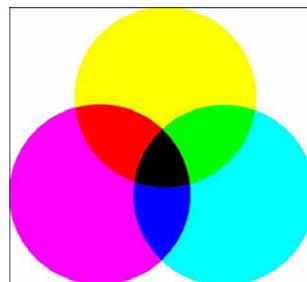


Figura 2.24: Colores substractivos primarios y secundarios

Los colores aditivos y los substractivos son complementarios:

$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (2.15)$$

Preprocesamiento

El objetivo de esta etapa es mejorar las imágenes obtenidas por la cámara para aumentar la probabilidad de éxito en sus posteriores etapas, se considera un procesamiento de imagen de bajo nivel. En el preprocesamiento se intenta reducir la información que no es importante para los objetivos particulares de análisis de las imágenes para cada caso en particular.

La importancia del preprocesamiento es considerar el hecho de que se puede remover el ruido introducido a la imagen ya sea por el sensor, el convertidor analógico-digital, o el canal de transmisión (cable de conexión entre la cámara y la computadora), aplicando a la imagen digitalizada un filtro paso bajas (porque el ruido se presenta con una frecuencia alta).

También se puede hacer una conversión a niveles de gris para ver con más detalle los contornos de la imagen, y poder determinar las zonas de interés en las etapas posteriores.

Otra técnica utilizada en el pre procesamiento es emplear las transformaciones geométricas, que permiten la eliminación de distorsiones geométricas de la imagen, que son provocadas por las lentes de las cámaras.

Una transformación geométrica es una función T que coloca al píxel (x, y) en una nueva posición (x', y') . En la Figura 2.25 se puede observar este concepto.

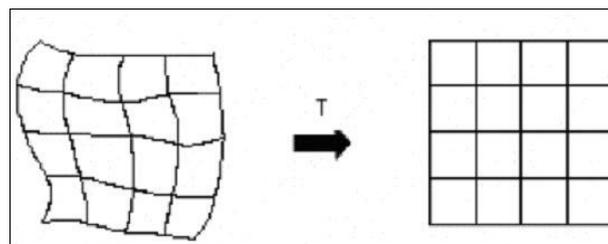


Figura 2.25: Transformación geométrica en un plano

Segmentación

El objetivo de esta etapa es realizar un proceso orientado a particionar la imagen digital en zonas disjuntas con un significado propio. Este significado dependerá de la imagen, así como de los objetivos que se persigan con su interpretación.

En la práctica se tienen dos tipos de segmentación: la basada en regiones o zonas y la de objetos. Ambas técnicas son semejantes, pero su utilización dependerá de la imagen a utilizar y de lo que se desee segmentar. Un ejemplo de esto se muestra en la Figura 2.26, donde se muestra la imagen original (a), la imagen (b) es obtenida utilizando un umbral cercano a 255, la imagen (c) se obtiene utilizando un umbral cercano a 0 y cercano a 255, y la imagen (d) es obtenida utilizando un umbral cercano a 0.

Se utilizan algoritmos de segmentación que se basan en:

1. Discontinuidad del nivel de gris: La cual consiste en dividir una imagen, basándose en los cambios bruscos de nivel de gris. Esto es aplicable a imágenes estáticas y dinámicas. Las principales técnicas son:
 - Detección de puntos aislados.

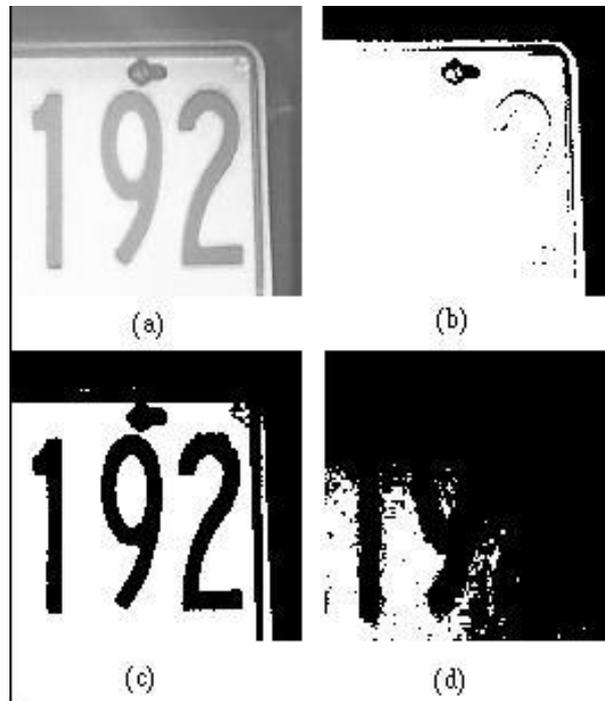


Figura 2.26: Ejemplo de segmentación

- Detección de líneas de una imagen.
 - Detección de bordes en una imagen.
2. Similitud del nivel de gris: Esto se obtiene utilizando las propiedades de distribución de los píxeles, como la intensidad o el color; o en encontrar directamente las regiones, utilizando para ello alguno de los siguientes métodos:
- Umbralización.
 - Crecimiento de regiones.
 - División y fusión de regiones.

La segmentación termina cuando los objetos o regiones de interés son identificados. Esta parte del procesamiento digital de imágenes es la más complicada y la que utiliza la mayoría de los recursos de hardware y software empleados por el equipo.

En la figura 2.27 se muestra un ejemplo de una imagen segmentada en regiones, dependiendo del nivel de gris.

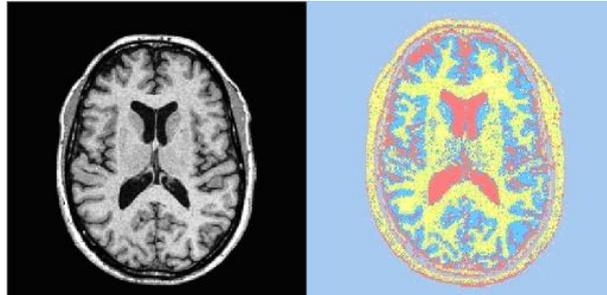


Figura 2.27: Segmentación

Representación y Descripción

Una vez segmentada la imagen en diferentes regiones, pasamos al siguiente paso que es representarla y después describir esa representación de una forma que facilite su posterior procesamiento, y sólo procesar la región de interés ya que se cuenta con menor cantidad de información, que la imagen completa.

La representación puede llevarse a cabo mediante dos enfoques distintos:

1. En términos de sus características externas (fronteras): la forma del objeto.
2. En términos de sus características internas (los píxeles que conforman la región): color y textura (propiedades de reflectividad).

Estas representaciones son invariantes a cambios en el tamaño, así como a traslaciones y rotaciones.

Las técnicas de segmentación entregan datos en bruto en forma de píxeles de una frontera o región. Algunas veces esos datos se utilizan directamente para obtener descriptores (Información específica sobre cada objeto). Lo habitual es utilizar esquemas que compacten los datos en representaciones que son bastante más útiles en el cálculo de descriptores.

Algunos métodos de representación son:

- Códigos de cadena.
- Aproximaciones poligonales.
- El esqueleto de una región.

Algunas técnicas para los descriptores son:

- Longitud del contorno.

- Número de forma.
- Descriptores de Fourier.

Reconocimiento e Interpretación

El reconocimiento tiene como objetivo identificar el patrón, es decir, identificar el objeto que representa. En la interpretación se asigna un significado al objeto reconocido.

En el reconocimiento de imágenes se seleccionan los rasgos discriminantes de una imagen, construyendo un vector de características que la represente. Las características que se utilicen tienen que cumplir las siguientes propiedades:

1. Capacidad discriminante: deben separar lo más nítidamente posible las clases existentes.
2. Fiabilidad: los objetos de una misma clase tendrán la menor dispersión posible.
3. Incorrelación: las características no dependerán fuertemente de otras, ya que si fuera así, no aportarían información.
4. Rapidez de cálculo.
5. Economía de cálculo.

La interpretación automática de imágenes es compleja. Entre las dificultades que presentan, están:

- El número elevado de datos que se tienen que procesar.
- La falta de herramientas básicas de procesado (que conviertan una matriz de píxeles en una lista del contenido).

La solución se basa en el uso de restricciones:

- Limitando la generalidad del problema: se simplifica el problema limitando el número de circunstancias desconocidas. Si no se pueden controlar las variables del problema, se limitan las posibilidades y la precisión de resultados.
- Intentando incorporar el conocimiento humano al proceso: eligiendo un formalismo en el que se exprese el conocimiento.

2.5. Calibración de la Cámara

La calibración de la cámara es un proceso que relaciona el modelo real (mundo real) con el dispositivo físico (computadora) y determina la posición y orientación de ésta con respecto al sistema de referencia del mundo.

En la calibración geométrica de una cámara el objetivo es determinar un conjunto de parámetros tanto internos como externos de la cámara que describe el mapeo, entre el sistema de coordenadas de referencia 3D y las coordenadas de la imagen 2D. Existen varios métodos para la calibración de cámaras. En este trabajo se utilizará un modelo similar al propuesto por Heikkilä de su trabajo: “Procedimiento de Calibración de Cámaras en Cuatro Pasos” [7].

Modelo Pin-Hole

En este modelo se toma en cuenta que todos los rayos provenientes de un cierto objeto atraviesan un fino agujero para impactar en el sensor de imagen. El sistema de referencia de la cámara se sitúa en el centro de la proyección, coincidiendo el eje z de dicho sistema con el eje óptico, también denominado eje axial. En esta disposición de ejes, el plano de imagen, de coordenadas (u, v) , se encuentra situado a una distancia igual a la longitud focal del objetivo de forma perpendicular al eje óptico. La intersección del eje óptico con el plano imagen se denomina punto principal.

El centro de proyección C de la cámara se supone constante pero es a priori desconocido. El plano imagen normalmente se sitúa por delante del centro de proyección C para tener una imagen sin inversión. En la Figura 2.28 se muestra un esquema del modelo pin-hole.

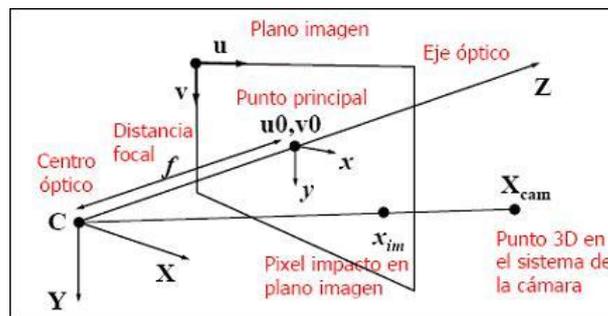


Figura 2.28: Esquema del modelo pin-hole

La relación que existe entre el punto $X_{cam}(X, Y, Z)$ de coordenadas 3D del sistema de referencia de la cámara y el punto $X_{in}(u, v)$ impactado en el plano de imagen, esta dado por la formula:

$$X_{in} = A \cdot X_{cam} \quad (2.16)$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.17)$$

Donde la matriz A es conocida como la matriz de parámetros intrínsecos.

Los parámetros físicos de la cámara son comúnmente divididos en:

- Parámetros intrínsecos, que son las características internas de la cámara.
- Parámetros extrínsecos, que son los parámetros que describen la geometría de la cámara.

Dentro de los parámetros intrínsecos se encuentran: La longitud focal efectiva (f), que es la distancia entre el plano de la imagen y el centro de proyección o centro óptico. El coeficiente de distorsión de las lentes (k_c), que modela la distorsión radial de la lentes en las direcciones x e y. Las coordenadas del punto principal o punto central de la imagen en píxeles (c_x, c_y). El factor de escala (s), entre las coordenadas calculadas de los píxeles y las coordenadas reales, debido a errores en el proceso de digitalización.

Dentro de los parámetros extrínsecos se encuentran los elementos de la matriz de rotación (R) y los elementos de la matriz de traslación (T).

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{32} \\ r_{31} & r_{23} & r_{33} \end{bmatrix} \quad (2.18)$$

$$\mathbf{T} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (2.19)$$

Con los parámetros extrínsecos podemos pasar coordenadas del punto P_W en coordenadas de un sistema de referencia externo, a coordenadas del sistema de referencia de la cámara P_{cam} , con la ecuación:

$$P_{cam} = R \cdot P_W + T \quad (2.20)$$

Distorsión de la lente

Una cámara generalmente exhibe una distorsión producida por la lente. Esta distorsión es la distancia existente entre el impacto ideal dado por el modelo pin-hole (modelo de proyección perspectiva) y la posición real que ocupa cada píxel en la imagen.

Existen dos tipos de distorsión producidas por las lentes de las cámaras. La distorsión radial cuyos parámetros son k_1 y k_2 , que provocan que un punto en la imagen sea desplazado radialmente del plano de imagen. La distorsión tangencial cuyos parámetros son p_1 y p_2 ; esta distorsión es provocada porque no siempre los centros de la curvatura de la superficie del lente son estrictamente colineales.

Si (u, v) es la coordenada de un píxel de la imagen, es decir, la coordenada de la proyección ideal, y (\hat{u}, \hat{v}) es la correspondiente coordenada de la imagen con distorsión y similarmente, (x, y) es la coordenada física ideal (sin distorsión) y (\hat{x}, \hat{y}) es la coordenada física real (con distorsión). Las coordenadas físicas con distorsión y sin distorsión se relacionan por:

$$\hat{x} = x + x [k_1 \cdot r^2 + k_2 \cdot r^4] + [2 \cdot p_1 \cdot x \cdot y + p_2 (r^2 + 2 \cdot x^2)] \quad (2.21)$$

$$\hat{y} = y + y [k_1 \cdot r^2 + k_2 \cdot r^4] + [2 \cdot p_1 \cdot x \cdot y + p_2 (r^2 + 2 \cdot y^2)] \quad (2.22)$$

donde:

$$r^2 = x^2 + y^2 \quad (2.23)$$

El segundo término de las ecuaciones anteriores describe la distorsión radial y el tercer término la distorsión tangencial. Si el centro de la distorsión radial es igual al punto principal (c_x, c_y) , entonces:

$$\hat{u} = u + (u - c_x) \left[k_1 \cdot r^2 + k_2 \cdot r^4 + 2 \cdot p_1 \cdot y + p_2 \left(\frac{r^2}{x} + 2 \cdot x \right) \right] \quad (2.24)$$

$$\hat{v} = v + (v - c_y) \left[k_1 \cdot r^2 + k_2 \cdot r^4 + 2 \cdot p_2 \cdot x + p_1 \left(\frac{r^2}{y} + 2 \cdot y \right) \right] \quad (2.25)$$

CAPÍTULO 3

Control de Sistemas por Visión

3.1. Implementación del Sistema de Visión

La implementación de algoritmos de visión artificial es un trabajo exhaustivo, ya que se requiere de la manipulación de punteros, gestión de memoria, conocimiento amplio del lenguaje de programación, etc. Estos problemas pueden ser resueltos si la implementación de prueba es realizada en MatLAB, utilizando su toolbox de procesamiento de imágenes, con ello el tiempo de implementación se convierte en el mínimo con la confianza de utilizar algoritmos probados y robustos.

El esquema de visión propuesto se aplica en el control de un robot tipo Rhino (Apéndice A) que es un robot manipulador de 5 grados de libertad (cintura, hombro, codo, muñeca y giro de la muñeca) y un efector final. Aunque, en este proyecto se utilizaron dos de las cinco articulaciones (Hombro y Codo) de forma que el robot se mueva en un plano. Debido a que las piezas que conforman el robot son rígidas, podemos saber la posición de cada una de las articulaciones conociendo tres puntos del robot: hombro, codo y muñeca.

3.1.1. Adquisición de Imágenes

La adquisición de imágenes sigue un proceso en el cuál se convierte la información visual en información digital para un procesamiento posterior (Ver 2.4.1). Los rayos de luz que provienen de los objetos pasan a través de una óptica (Ver 2.5) y son captados por un sensor visual. La información de luz es transformada a información eléctrica analógica (salida de la cámara), esta información eléctrica es capturada y con un digitalizador es transformada en una matriz de valores la cual es procesada para obtener cierta información (objetos presentes, distancia a la cámara, etc).

Cámara

La cámara utilizada es una *StarCam370i* de MSI la cual cuenta con las siguientes características:

- Sensor: CMOS 1/4"
- Resolución: 640x480, 352x288, 320x240, 176x144, 160x120
- Cuadros por seg: 15fps a 640x480 y 30fps a 352x288
- Ángulo de distorsión del lente: 56 grados en diagonal
- Brillo y control de exposición: Automático
- Formato de la imagen: RGB24
- Consumo de potencia: 2 W
- Interfaz IO: USB 1.1

Esta cámara es conveniente para los propósitos de este trabajo puesto que es de fácil adquisición, económica y fácil de utilizar.

Adquisición de Imagen Mediante MatLAB

MatLAB es conveniente ya que no necesita ninguna configuración especial para soportar distintos tipos de cámaras, siempre y cuando los drivers de las tarjetas de adquisición sean compatibles con esta herramienta de adquisición. El Toolbox de adquisición de imágenes de MatLAB es válido para equipos matrox, data translation y video de Windows genérico. Para identificar el tipo de dispositivo de video que se posea se utiliza el siguiente comando:

```
>> info = imaqhwinfo
info =
    InstalledAdaptors: {'winvideo'}
    MatLABVersion: '7.0 (R14)'
    ToolboxName: 'Image Acquisition Toolbox'
    ToolboxVersion: '1.5 (R14)'
```

Este comando devuelve los drivers disponibles para usarse con MatLAB. El comando *winvideo* indica que únicamente se tienen instalados los drivers de video genéricos de windows, sin embargo, estos son suficientes para la cámara utilizada en este proyecto.

Para poder trabajar con la cámara, se tiene que crear un objeto que pueda controlar la cámara:

```
>>objetovideo=videoinput('winvideo')
```

Este comando da como resultado una lista de las propiedades del objeto de video creado, el número de imágenes adquiridas, el tipo de disparo de la adquisición, el estado del objeto, etc. Estas propiedades pueden ser ajustadas usando el comando:

```
>>set(objetovideo,'propiedad',valor_propiedad)
```

Sin embargo, esto implica saber las propiedades de antemano. De este modo una de las formas más sencillas de ajustar las propiedades es crear un canal de video y ajustarlas en tiempo real:

```
>>preview(objetovideo)
```

Una vez que se ha ajustado las propiedades se puede trabajar con la cámara. Surge en este momento la problemática de como trabajar en *tiempo real*, puesto que el robot es un sistema físico que necesita las entradas de control para poder efectuar su movimiento. El concepto de *tiempo* es una idea sumamente discutida por filósofos y matemáticos [21], pero para los fines que perseguimos consideraremos como que un *sistema en tiempo real es aquel en el que para que las operaciones computacionales estén correctas no depende sólo de que la lógica e implementación de los programas computacionales sea correcto, sino también en el tiempo en el que dicha operación entregó su resultado*[22]. De esta forma se debe asegurar que se obtenga una salida para cada una de las imagenes obtenidas antes de adquirir la siguiente imagen.

Eventos en MatLAB

MatLAB posee “Events and Callbacks” que son eventos que ocurren mientras la adquisición de una imagen esta en progreso. MatLAB define un conjunto de eventos que incluyen *inicio*, *paro*, o *adquisición de imágenes*.

Un *Callback*¹ es una función específica que se ejecuta cuando ocurre un determinado evento. Los eventos que maneja MatLAB son los mostrados en la Tabla 3.1.

Usando el *callback* TriggerFcn podemos agregar una función que analice, procese y controle el sistema mecánico y se obtenga una salida antes de que la siguiente entrada sea procesada

```
set(vid,'FramesPerTrigger',1)
set(objetovideo,'TriggerFcn',{'display_frame'})
```

¹Información obtenida de la documentación de MatLAB

Evento	Propiedad	Descripción
Error	ErrorFcn	Se genera cuando ocurre un error de hardware o de tiempo
Frames	AcquiredFramesAcquiredFcn	Se genera cuando un número especificado de cuadros han sido capturados
Start	StartFcn	Se genera cuando el objeto de video es iniciado
Stop	StopFcn	Se genera cuando el objeto de video es detenido
Timer	TimerFcn	Se genera después de que ha pasado un tiempo previamente determinado
Trigger	TriggerFcn	Se genera después de que es ejecutado un disparo

Tabla 3.1: Eventos de MatLAB

En donde *displayframe* es el nombre de la función que hará el análisis de la imagen. Este proceso resulta de vital importancia ya que aquí se obtiene la frecuencia con que el sistema operará, razón por la cual los algoritmos deben de ser rápidos de forma que el único retraso sea el tiempo de adquisición de una nueva imagen.

Con esta configuración sería suficiente, sin embargo, cada que se ejecuta el programa sigue un flujo latente del objeto que controla la cámara así usando el *Callback* StopFcn se hizo una función que detenga de forma correcta la ejecución del programa.

```
set(objetovideo, 'StopFcn', {'stop'})
```

3.1.2. Acondicionamiento e Iluminación del Entorno

Un proceso propuesto para una correcta adquisición de la imagen conllevaría una correcta iluminación, cuya finalidad es proporcionar luz suficiente para obtener una buena imagen de la escena. La cámara con que se cuenta es muy sensible a los cambios de luz debido a que cuenta con un control automático de brillo y exposición por lo que cualquier reflejo impacta mucho en la adquisición de la imagen. Para contrarrestar este efecto se colocó un fondo negro y se utilizó la iluminación frontal direccional con lámparas fluorescentes de forma que los puntos de colores reflejaran la mayor cantidad de luz.

3.1.3. Calibración

La calibración de la cámara es un proceso con el que se determinan los parámetros internos y externos de la cámara (distancia focal, el centro del plano del sensor y la distorsión del lente) para poder determinar la geometría de los objetos captados por la cámara (Ver 2.5).

Debido a la calibración realizada en la adquisición de imágenes la cámara deberá permanecer invariante en sus parámetros, es decir que se tiene que mantener un mismo enfoque, una misma rotación y un mismo lugar ya que cualquier cambio de esos parámetros conllevará a una nueva calibración; la calibración se realiza sólo una vez que se haya acomodado la cámara.

Para efectuar la calibración de la cámara se toma un objeto patrón el cual debe de tener puntos significativos fácilmente identificables. En nuestro caso una rejilla facilita la búsqueda de puntos significativos que son las esquinas de los cuadrados.

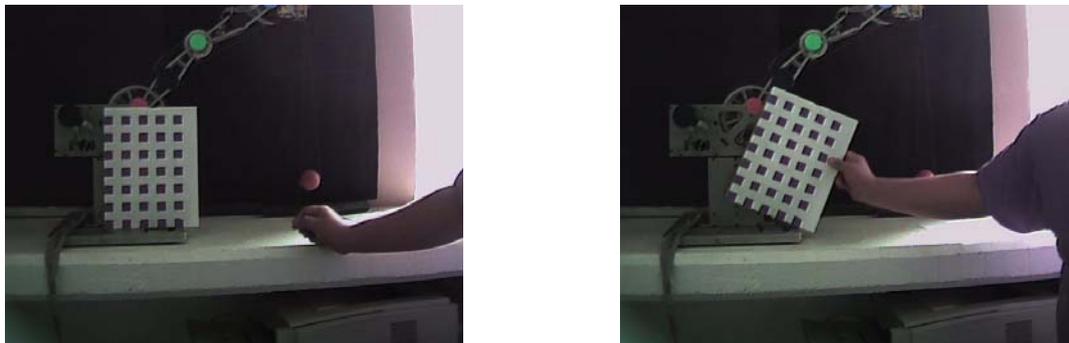


Figura 3.1: Figura patrón para la calibrar la cámara

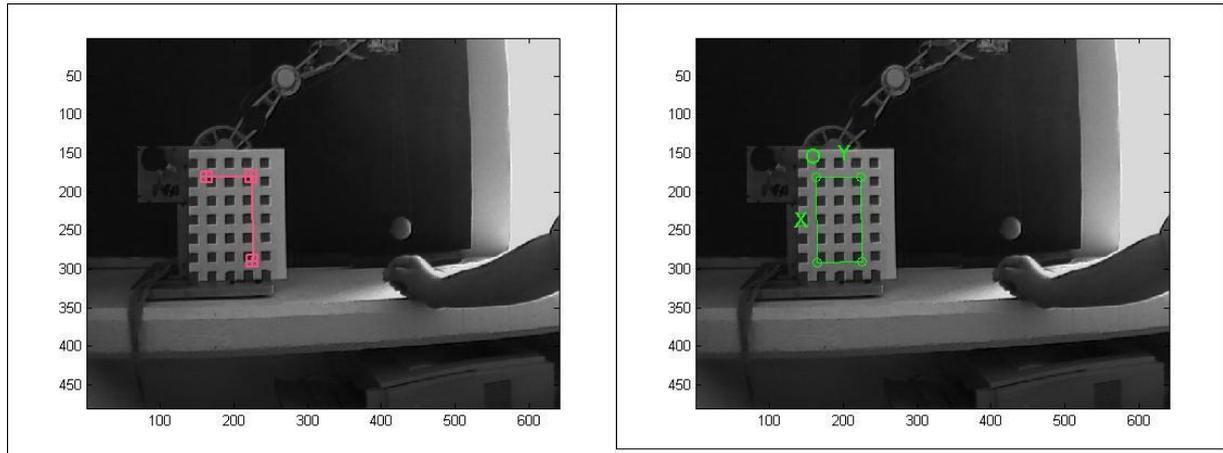
La calibración de la cámara se efectúa a través un programa de calibración, basado en las imágenes ya captadas de la imagen patrón.

Debemos de saber la cantidad de imágenes que se utilizarán para el mallado y la longitud entre los puntos significativos. Para realizar la calibración se utilizan 4 puntos en cada imagen para formar un cuadrado, de tal forma que estos pueden ser un pequeño cuadrado o la suma de varios dentro del patrón. Debemos tener en cuenta que los puntos deberán estar lo más próximo a las esquinas de los cuadrados y haciendo un rectángulo lo más regular posible (Figura 3.2a).

A continuación, nos pedirá también el número de cuadrados que integran el rectángulo formado, tanto según el eje x y el eje y (Figura 3.2b).

Con los puntos obtenidos y con los datos de longitud asociados a cada una de estas imágenes, se puede utilizar el algoritmo de calibración de los cuatro pasos [7], que consiste en la iteración de los parámetros obtenidos para obtener la matriz de rotación y traslación, además de ciertos parámetros como la distancia focal y el error obtenido.

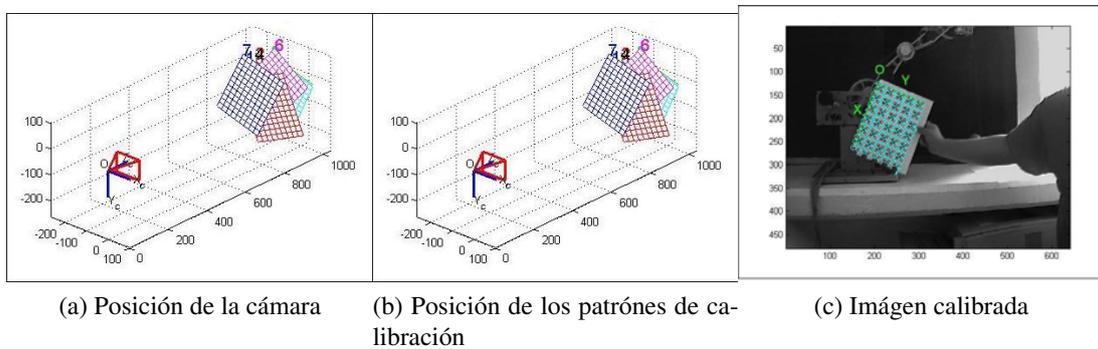
Así con MatLAB se puede obtener una representación que nos muestre la posición de cada una de las figuras conforme a la vista que tiene con la cámara (Figuras 3.3a y 3.3b) y una figura que represente la inclusión de cada uno de los puntos de la imagen patrón (Figura 3.3c).



(a) Juego de puntos para realizar la calibración

(b) Logitud para calibración

Figura 3.2: Proceso de calibración de la cámara



(a) Posición de la cámara

(b) Posición de los patrones de calibración

(c) Imágen calibrada

Figura 3.3: Resultados de la calibración

3.1.4. Preprocesamiento

Una imagen de color RGB es representada por una matriz tridimensional $m \times n \times p$, donde n representa el número de píxeles de ancho y m el número de píxeles de largo y p representa el plano que puede ser 1 para el rojo, 2 para el verde y 3 para el azul (Figura 3.4). Puesto que buscamos tres puntos, se colocó un punto rojo, uno verde y uno azul en cada una de las uniones de las articulaciones y con esto se redujo la búsqueda de un solo punto en cada uno de los planos de una imagen RGB.

Antes de comenzar con la búsqueda debemos de eliminar el ruido de la imagen que tenemos, es decir, reflejos, puntos blancos e imperfecciones de la imagen. Debido a que se trabajó en tiempo real, este algoritmo debe ser lo bastante rápido para que no interfieran con la frecuencia de adquisición de la cámara con la que se hace la adquisición. También la unión de los conjuntos de los planos RGB da por resultado el color blanco, un píxel de valor 255 en cada uno de los planos, una forma de eliminar gran parte de ruido es restar matricialmente un plano de la imagen, por ejemplo Verde ($I_G(m, n, 2)$), a otro plano distinto de la misma imagen, por ejemplo del plano Azul ($I_B(m, n, 3)$), con lo que obtenemos un plano sin los puntos de mayor valor de píxel (Figura 3.5).

3.1.5. Segmentación de Bordos y Reconocimiento de Formas

En este punto la imagen tiene varios niveles de gris en distintos rangos, uno de los cuales es el valor del punto que nos interesa, por lo tanto podemos comparar los niveles de gris con algún valor de referencia para realizar la segmentación.

La segmentación basada en el método de umbralización permite que al obtener una imagen tengamos dos regiones definidas: la primera contiene los píxeles que componen el punto que se encuentra en cada una de las articulaciones y la segunda contiene todos los demás píxeles que componen el fondo de la imagen. Para determinar el valor del umbral se utilizó el valor máximo, la moda y un análisis de los histogramas de cada una de las componentes de la imagen RGB; al final se unen para formar el umbral buscado.

Para comenzar con la búsqueda de los puntos de interés se obtienen dos matrices (3.1) y (3.2) con los máximos valores de cada fila y columna de la imagen.

$$Col_{max}(1, n) = [col_{11} \quad col_{12} \quad \dots \quad col_{1n}] \quad (3.1)$$

$$Fil_{max}(1, n) = [fil_{11} \quad fil_{12} \quad \dots \quad fil_{1n}] \quad (3.2)$$

Con estas matrices y con la definición del umbral se hace un barrido buscando una serie de píxeles que cumplan con:

1. Superar el umbral definido

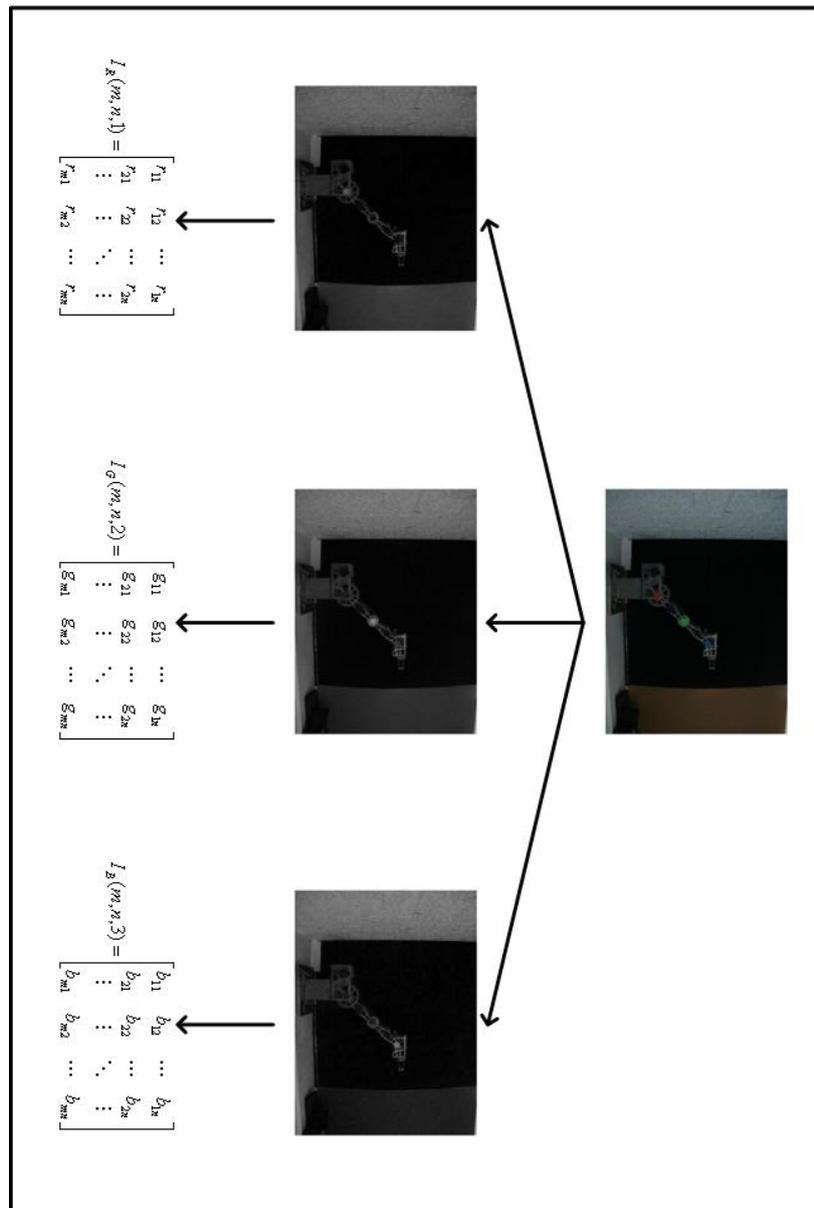


Figura 3.4: Imagen RGB en MatLAB

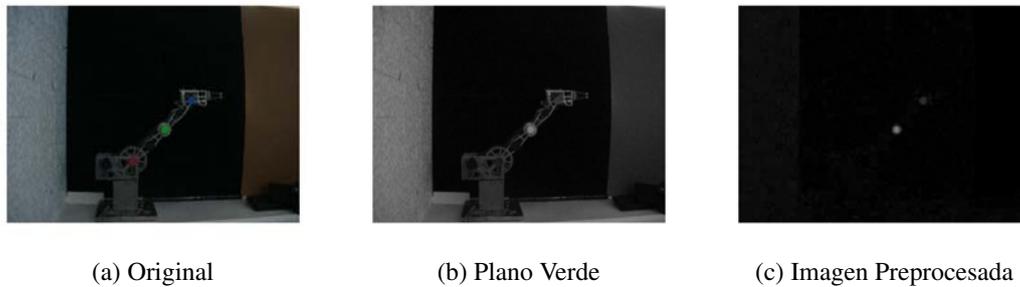


Figura 3.5: Preprocesamiento a la imagen

2. Que cubran el 80 % del valor del ancho o largo del punto buscado

Con esto tenemos las proyecciones en X_{ancho} y Y_{largo} de cada uno de los puntos RGB de la imagen y con lo que podemos obtener las coordenadas del centro de cada uno de los puntos (X_0, Y_0) , (X_1, Y_1) y (X_2, Y_2) en la Figura A.3.

3.2. Control Descentralizado Difuso

3.2.1. Diseño del Controlador Difuso

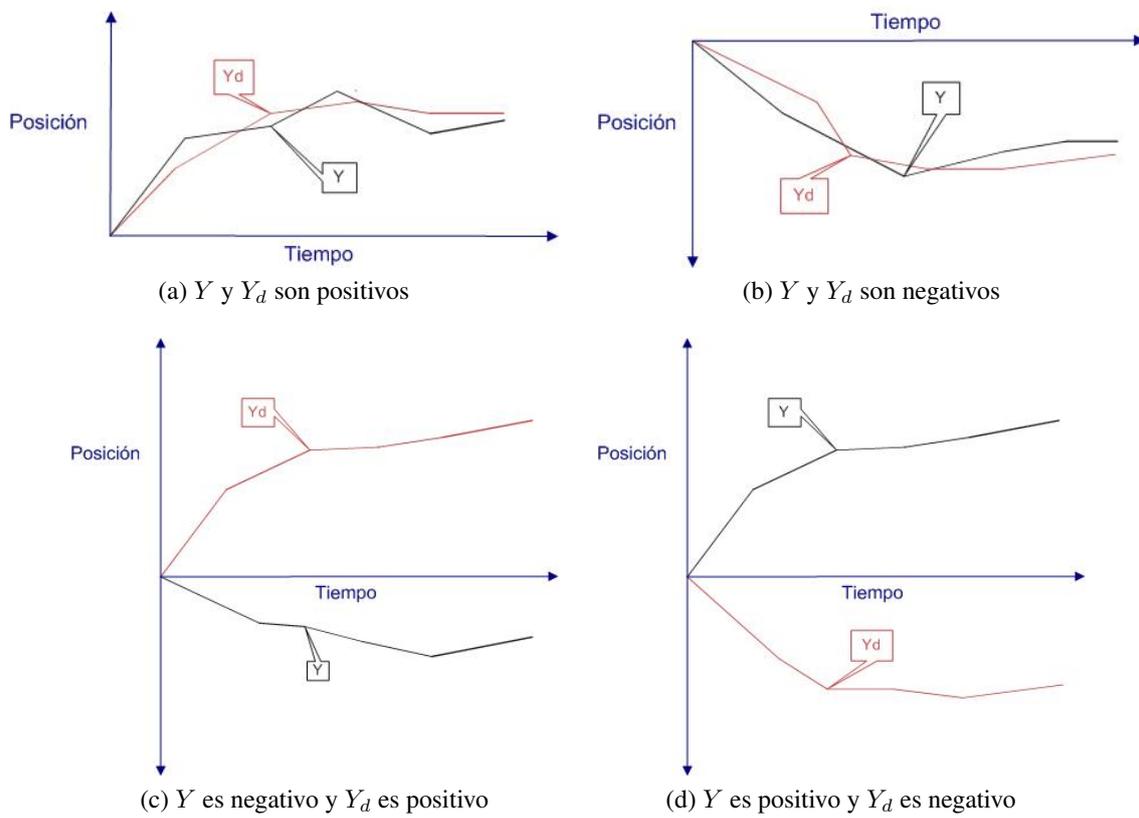
El controlador difuso se basó en las reglas lingüísticas obtenidas a partir de la experiencia que se tuvo con el control del movimiento del robot Rhino. El robot Rhino cuenta con una etapa de potencia, la cual recibe como entrada voltajes de corriente directa de -10 V a + 10 V, que le genera el par de los motores en distintas magnitudes y en ambos sentidos (Apéndice B). En base en lo anterior y conociendo los estados del sistema, podemos tener los casos en que con la posición deseada (Y_d) y la posición real (Y) sean positivas (Figura 3.6a), negativas (Figura 3.6b) o se encuentren en un plano distinto (Figuras 3.6c y 3.6d).

Dado que se genera cierto par en cada uno de los motores a través de una señal de voltaje, definimos en la tabla 3.2 las posibles salidas.

De este modo definimos en la tabla 3.3 para cada uno de los casos presentados en las gráficas su correspondiente variable lingüística.

Con estas variables el control resulta poco efectivo ya que si la separación entre la Y y Y_d es demasiada, el par que resulte será muy bajo para poder mover cualquiera de los eslabones, por este motivo definimos en (3.3) el error de seguimiento (e) entre las posiciones (Y y Y_d).

$$e = Y - Y_d \quad (3.3)$$

Figura 3.6: Casos para Y y Y_d

Constante	Definición
EP	Empuje Pequeño
EM	Empuje Mediano
EG	Empuje Grande
JP	Jalón Pequeño
JM	Jalón Mediano
JG	Jalón Grande

Tabla 3.2: Definición de salidas

Variable	Figura
Si $Y \in P$ & $Y_d \in P_d$ entonces $\mu = EP$	3.6a
Si $Y \in P$ & $Y_d \in N_d$ entonces $\mu = JM$	3.6c
Si $Y \in N$ & $Y_d \in P_d$ entonces $\mu = EM$	3.6d
Si $Y \in N$ & $Y_d \in N_d$ entonces $\mu = JP$	3.6b

Tabla 3.3: Variables lingüísticas de Y y Y_d

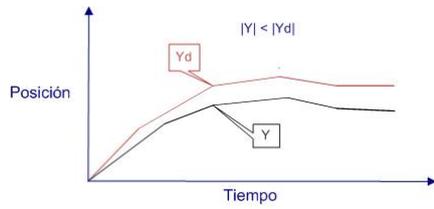
Con la definición del error, este puede ser positivo o negativo dependiendo de las magnitudes absolutas de Y y Y_d , se obtiene la señal de control que se aplica al sistema la salida (μ) al sumar a las variables lingüísticas definidas en la Tabla 3.3 otro singleton debido al error. Las variables lingüísticas completas se muestran en la Tabla 3.4, donde Y es la posición real, Y_d es la posición deseada y μ es la salida del sistema.

Variable	Figura
Si $Y \in P$ & $Y_d \in P_d$ & $e \in N_e$ entonces $\mu = EP + EM$	3.7a
Si $Y \in P$ & $Y_d \in P_d$ & $e \in P_e$ entonces $\mu = EP + JM$	3.7b
Si $Y \in P$ & $Y_d \in N_d$ & $e \in N_e$ entonces $\mu = 0$	Este caso no existe
Si $Y \in P$ & $Y_d \in N_d$ & $e \in P_e$ entonces $\mu = JM + JM$	3.7c y 3.7d
Si $Y \in N$ & $Y_d \in P_d$ & $e \in N_e$ entonces $\mu = EM + EM$	3.7e y 3.7f
Si $Y \in N$ & $Y_d \in P_d$ & $e \in P_e$ entonces $\mu = 0$	Este caso no existe
Si $Y \in N$ & $Y_d \in N_d$ & $e \in N_e$ entonces $\mu = JP + EM$	3.7g
Si $Y \in N$ & $Y_d \in N_d$ & $e \in P_e$ entonces $\mu = JP + JM$	3.7h

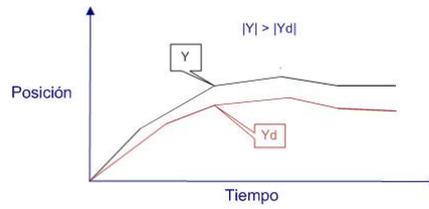
Tabla 3.4: Variables lingüísticas de Y , Y_d y e

3.2.2. Implementación del Control Difuso

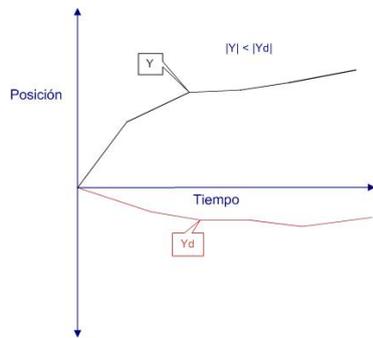
El control se implementó con los siguientes parámetros: La parte antecedente está definida a partir de Funciones Membresía sigmoideas Y [media, desviación estandar] = $([-5, -40], [5, 40])$, Y_d [media, desviación estandar] = $([-3, -40], [3, 40])$ y e [media, desviación estandar] = $([-10, -1], [10, 1])$. Los singletons utilizados fueron: EP = 2.5, EM = 25, JM = -10 y JP = -6.



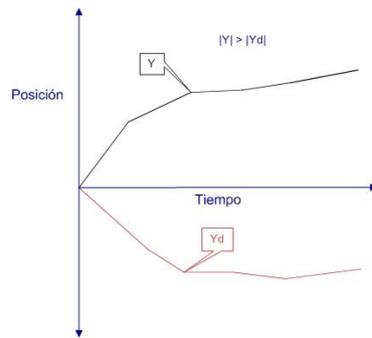
(a) Y, Y_d son positivos y e es negativo



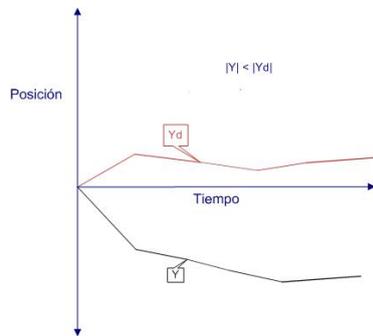
(b) Y, Y_d y e son positivos



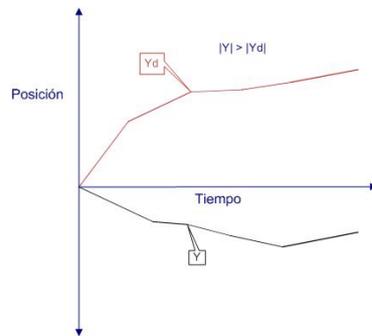
(c) Y es positivo, Y_d es negativo y e es positivo



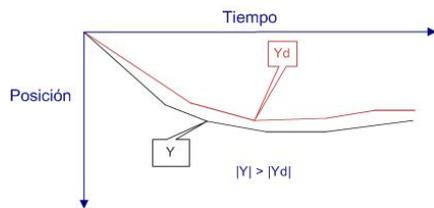
(d) Y es positivo, Y_d es negativo y e es positivo



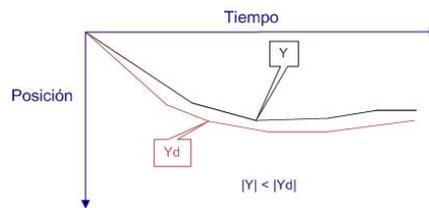
(e) Y es negativo, Y_d es positivo y e es negativo



(f) Y es negativo, Y_d es positivo y e es negativo



(g) Y, Y_d son negativos y e es negativo



(h) Y, Y_d son negativos y e es positivo

Figura 3.7: Casos para Y, Y_d y e

3.2.3. Conexión con el Robot Rhino

El robot Rhino cuenta con una interfaz (Apéndice B) la cual tiene dos objetivos principales:

1. Permitir la conectividad con dispositivos de control
2. La posibilidad de manejar señales tanto analógicas como digitales

Señales analógicas: Los voltajes deben de estar dentro de un intervalo de ± 10 V

Señales digitales: Los voltajes son de nivel TTL con Modulación de Ancho de Pulso (PMW) que determina la velocidad de giro y sentido.

Debido a que la salida del controlador son niveles lógicos de voltaje, se tuvo la necesidad de diseñar un convertor digital/analógico (Apéndice C) utilizando el puerto paralelo de la computadora para poder convertir y enviar las señales de control.

3.3. Resultados de Simulación y Experimentos

Los resultados obtenidos se presentan en el siguiente orden. En primer lugar, se muestran los resultados obtenidos por simulación del control difuso (Figuras 3.8 y 3.9). Posteriormente se presentan los resultados obtenidos de manera experimental introduciendo al sistema distintos tipos de señales.

1. En las gráficas 3.10 y 3.11, se muestran los resultados de la aplicación del control difuso por retroalimentación visual con una señal senoidal
2. En las gráficas 3.12 y 3.13, se muestran los resultados de la aplicación del control difuso por retroalimentación visual con una señal triangular

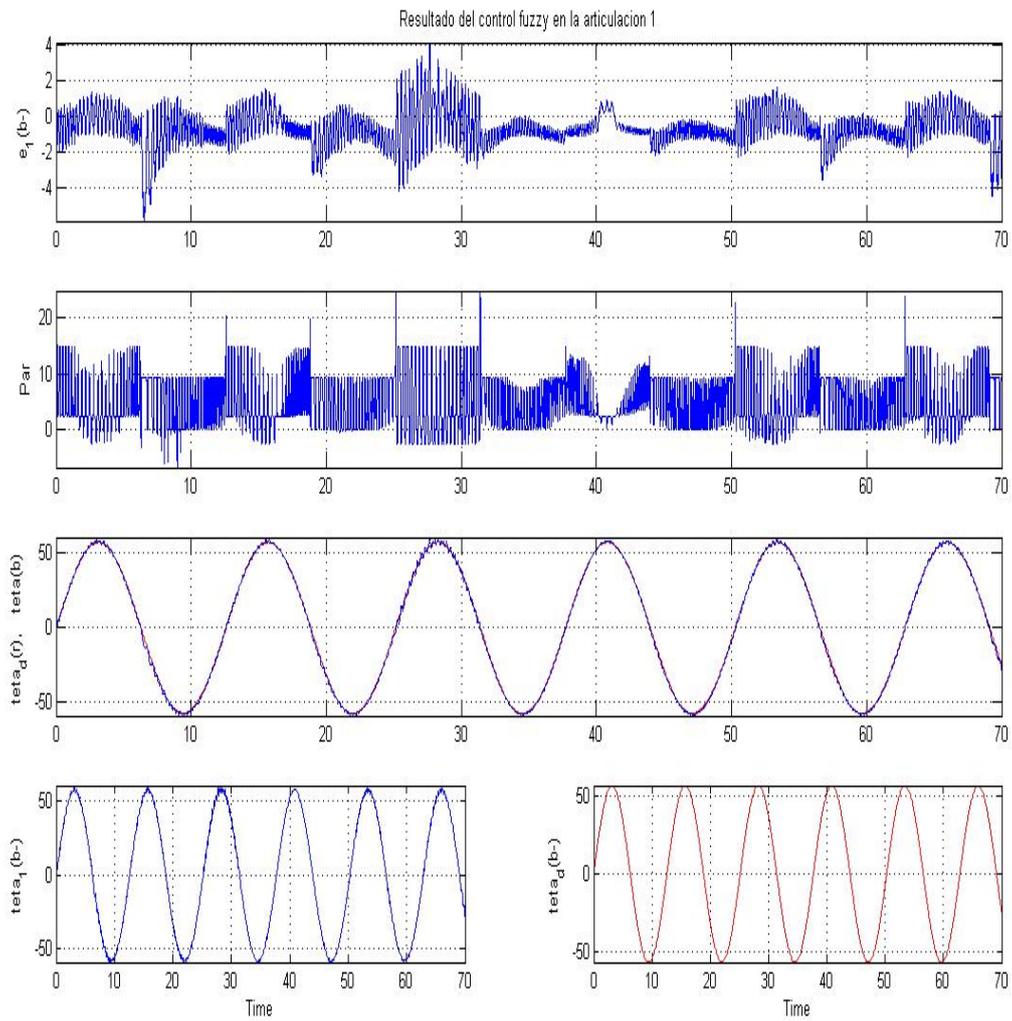


Figura 3.8: Resultado de la simulación del sistema de control en el Hombro

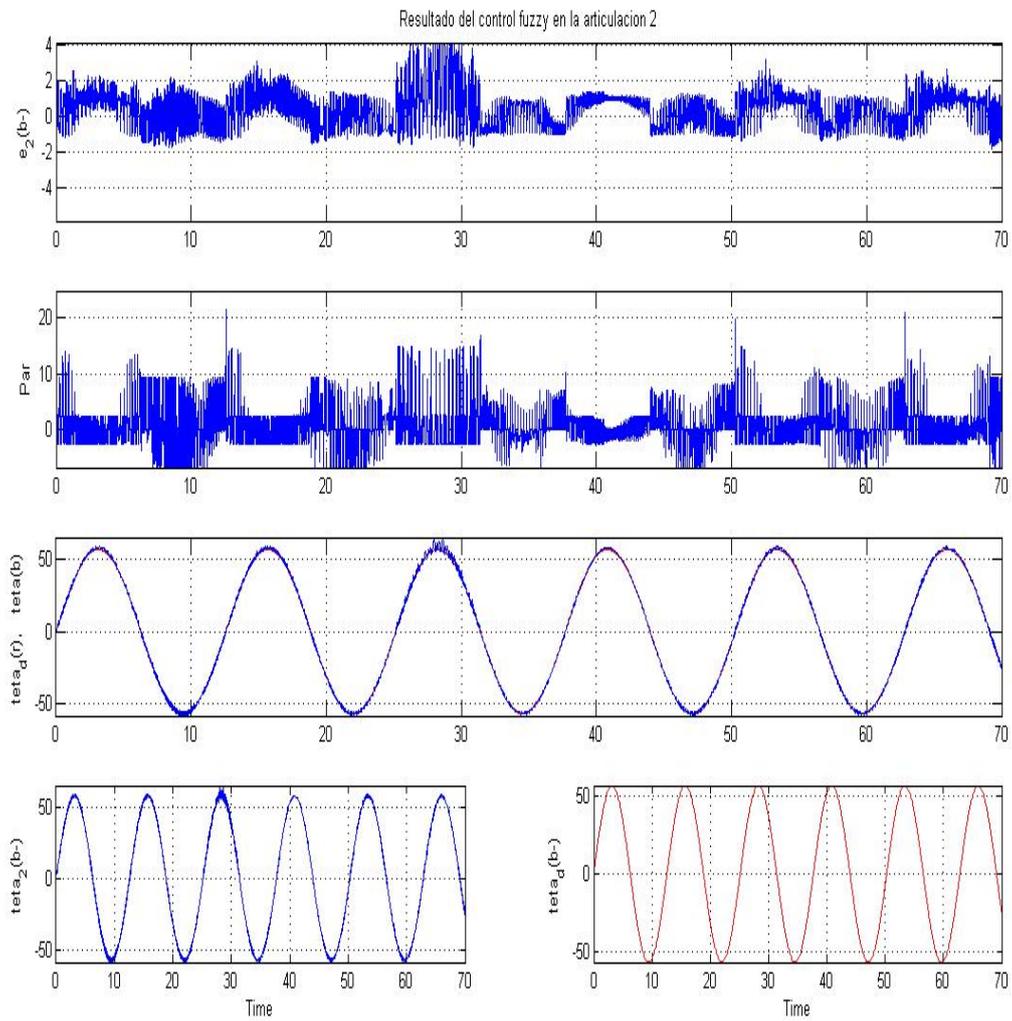


Figura 3.9: Resultado de la simulación del sistema de control en el Codo

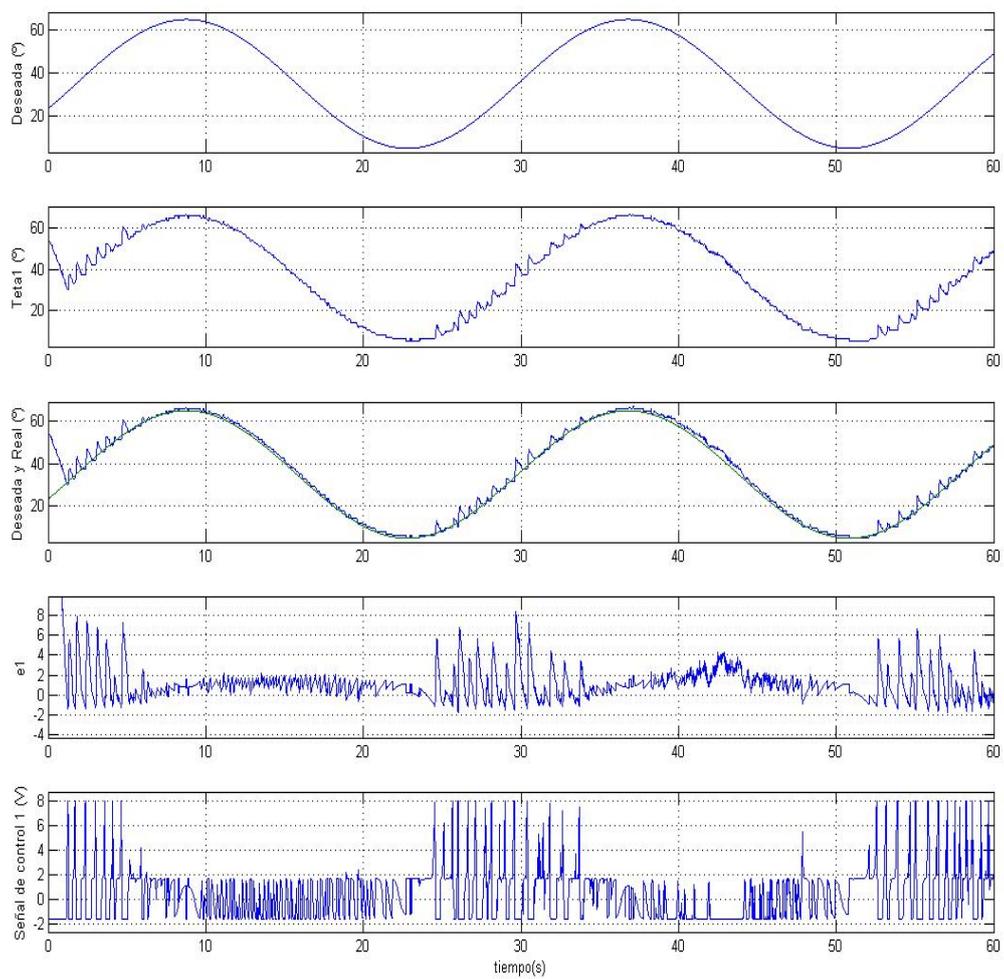


Figura 3.10: Resultados experimentales con una señal senoidal en el Hombro

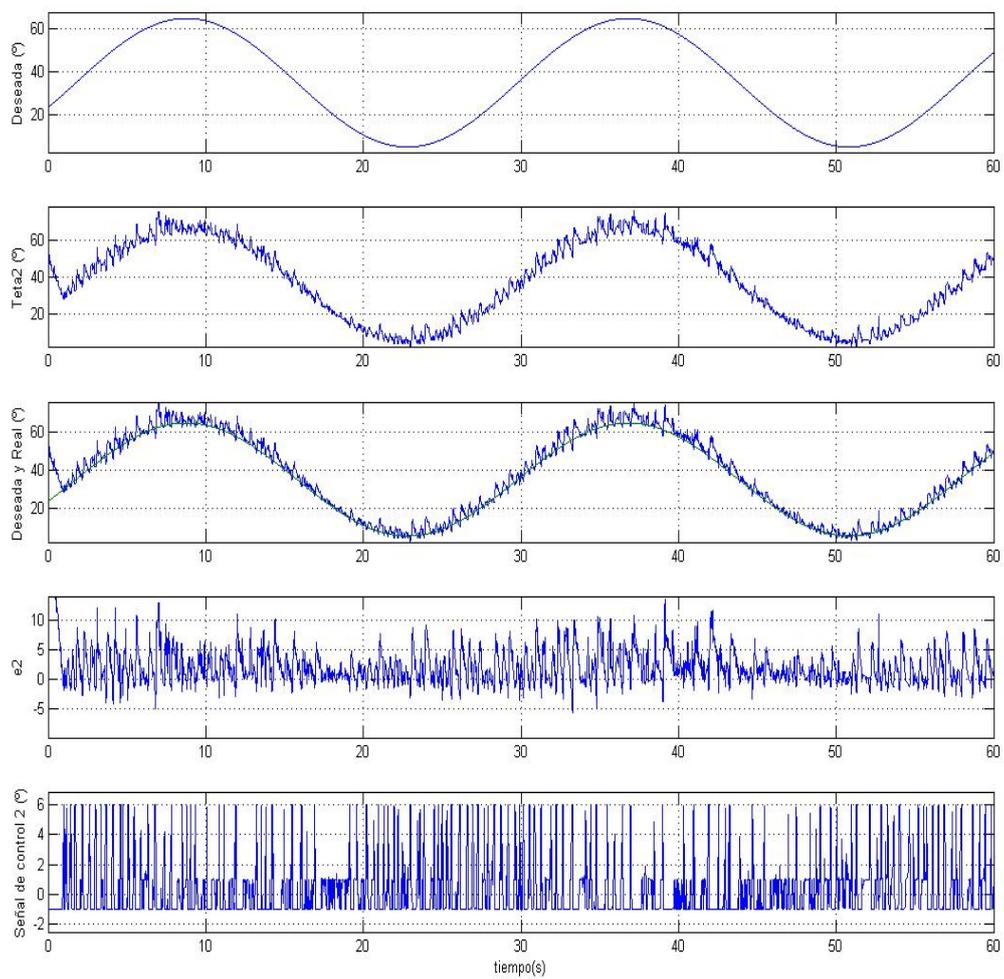


Figura 3.11: Resultados experimentales con una señal senoidal en el Codo

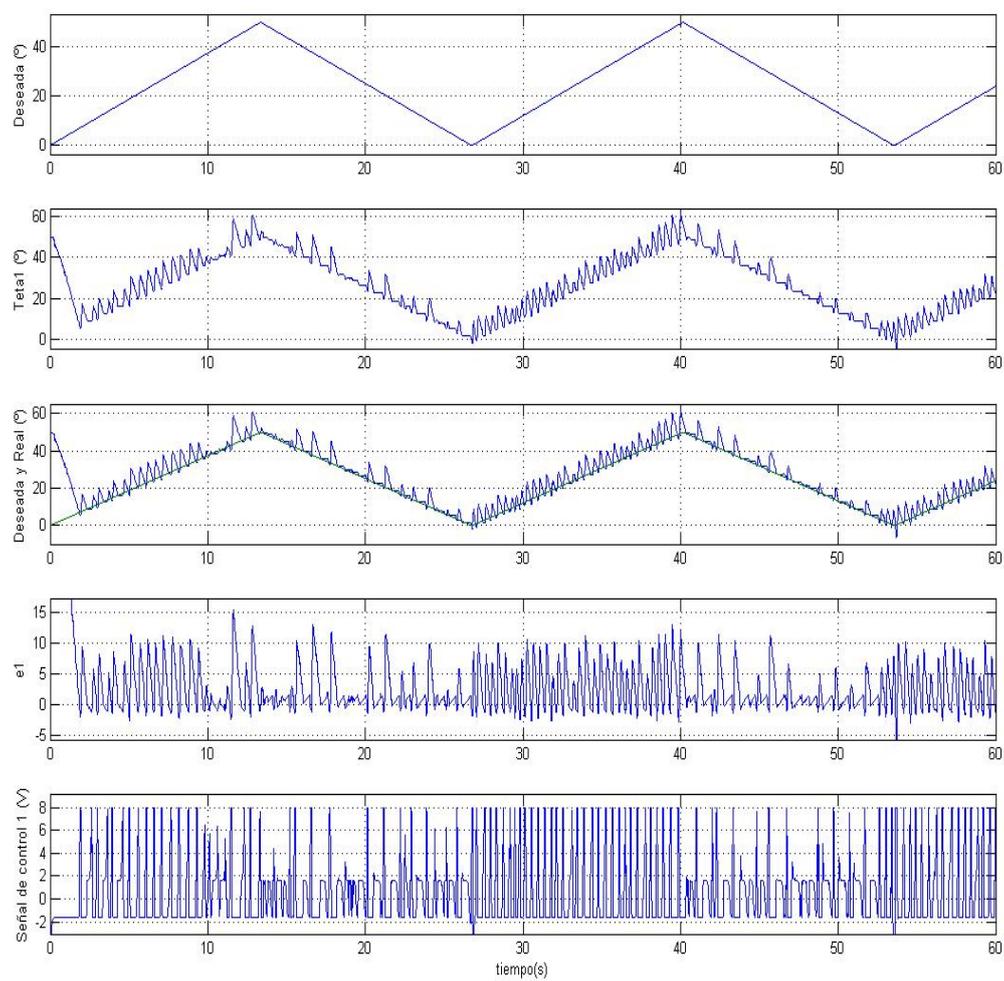


Figura 3.12: Resultados experimentales con una señal triangular en el Hombro

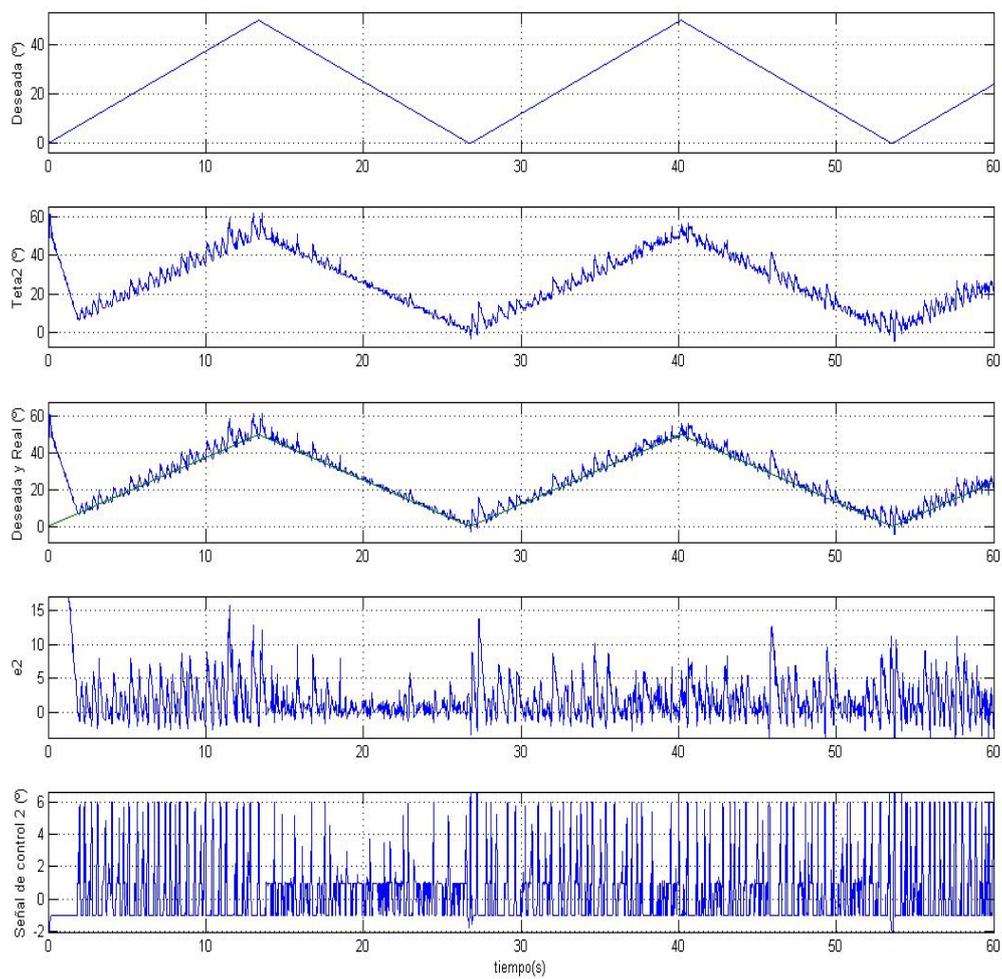


Figura 3.13: Resultados experimentales con una señal triangular en el Codo

CAPÍTULO 4

Conclusiones

En términos generales se cubrieron los objetivos planteados en este trabajo de tesis.

Se diseñó un algoritmo de visión eficiente y confiable, capaz de identificar los puntos de interés de la imagen capturada por la cámara, con una velocidad limitada sólo por la velocidad de cámara con el que se realizaron las pruebas y con una simple reconfiguración puede aceptar cámaras de mayor velocidad.

Se diseñó un algoritmo eficiente de control utilizando lógica difusa comprobándose en las simulaciones del sistema realizadas con MatLAB y en las pruebas experimentales. Un punto importante a considerar es que el error no es mínimo en las pruebas experimentales a comparación con la simulación del sistema, pero este se encuentra acotado, por lo que esta variación se debe a los actuadores y al desgaste del Robot Rhino.

Este trabajo se elaboró únicamente con el equipo con el que cuenta el Laboratorio de Control del Posgrado de la Facultad de Ingeniería, contando como las únicas inversiones realizadas la implementación de una tarjeta para la conversión de las señales de control digitales a señales analógicas y la adquisición de una cámara web de propósito general, por lo que la suma de ambos costos soluciona el problema del alto costo de la captura de datos por medio de visión en el diseño de prototipos.

4.1. Limitaciones

Este trabajo encuentra limitaciones en hardware y en software que pueden ser corregidos en trabajos futuros con lo que mejoraría la presentación de los resultados:

- El tipo de cámara que se utilizó cuenta con una velocidad de captura de 15 fps en la resolución 640x480 pixeles lo cual es lenta en comparación con las cámaras profesionales que capturan por lo menos a 60 fps a la misma resolución
- La transmisión de los datos de la cámara usada a la computadora utiliza el puerto USB 1.1, el cual tiene una velocidad máxima de 12 Mbps, en comparación con otro tipo de puertos como el IEEE 1394 ó FireWire, los cuales tienen velocidades superiores de 400 Mbps
- La cámara utilizada no tiene un control de la abertura del diafragma para moderar la cantidad de luz que incide sobre el sensor CMOS, por lo que se tuvo que adecuar el lugar y las condiciones de iluminación
- La programación se efectuó en MatLAB lo cual resulta en un costo considerable extra en la adquisición de una licencia, sin embargo, tanto el algoritmo de visión como el de control se pueden migrar a otro lenguaje de programación como C++, Java o C#

4.2. Alcances Futuros

Este trabajo se puede complementar con algunas de las propuestas sugeridas a continuación:

- Realizar las pruebas experimentales utilizando una cámara profesional para mejorar los algoritmos tanto de visión como de control y llevar el error a un estado mínimo
- Investigación para utilizar visión estereoscópica, para tener un reconocimiento del entorno en 3D
- Investigación para realizar el control de sistemas de manera remota, por medio de una red local LAN, o por medio de Internet
- Utilización de los algoritmos en otro tipo de sistemas electromecánicos

APÉNDICE A

Descripción del Robot Rhino

El Robot Rhino XR-3 (Figura A.1) es un brazo robot con cinco grados de libertad. Los ejes de movimiento son:

1. Rotación de la cintura
2. Rotación del hombro
3. Rotación del codo
4. Rotación de la muñeca
5. Flexión en la pinza

El movimiento es transferido desde el eje de los actuadores a las articulaciones a través de cadenas formadas por eslabones. La velocidad de rotación del eje de los actuadores es reducida a través de los engranes del actuador a las articulaciones

El robot rhino tiene las siguientes características físicas:

- Esta construido de aluminio, lo que le da resistencia y poco peso
- Pesa 7.3 kg sin base y 11.8 con ella
- Tiene un alcance de 0.6 m desde el centro hasta los dedos de la pinza
- El control de velocidad se hace a través de servomotores desde 0 V hasta 20 V
- Alcanza una velocidad máxima de 40°por segundo

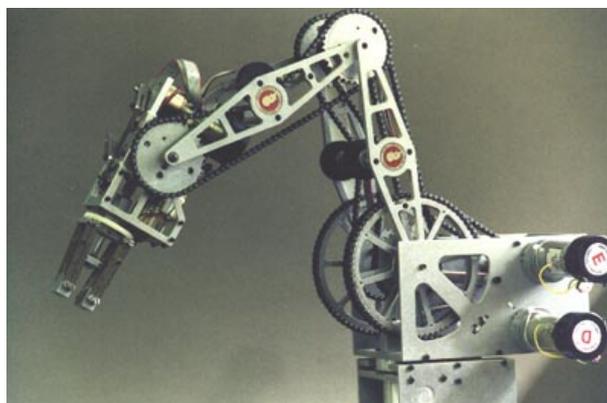


Figura A.1: Robot Rhino XR-3

Articulación	Motor	Rango de movimiento
Cintura	F	350°
Hombro	E	210°
Codo	E	265°
Muñeca	D	310°
Giro de Muñeca	B	356°
Movimiento pinza	A	1 s para abrir, 2 s para cerrar

Tabla A.1: Asignación de motores

- La corriente máxima es de 2A
- La asignación de motores es la mostrada en la tabla A.1

La reducción del radio de giro por el engranaje en cada articulación es la mostrada en la Tabla A.2.

Articulación	Relación del engranaje del motor	Relación del engranaje por encadenamiento	Reducción final
Cintura	66.1:1	4:1	264.4:1
Hombro	66.1:1	8:1	528:1
Codo	66.1:1	8:1	528:1
Muñeca	66.1:1	8:1	528:1
Giro de Muñeca	96:1	3	384:1
Movimiento pinza	96:1	No aplica	No aplica

Tabla A.2: Reducción del giro del engranaje

Las dimensiones del robot rhino XR-3 se muestran en la Figura A.2:

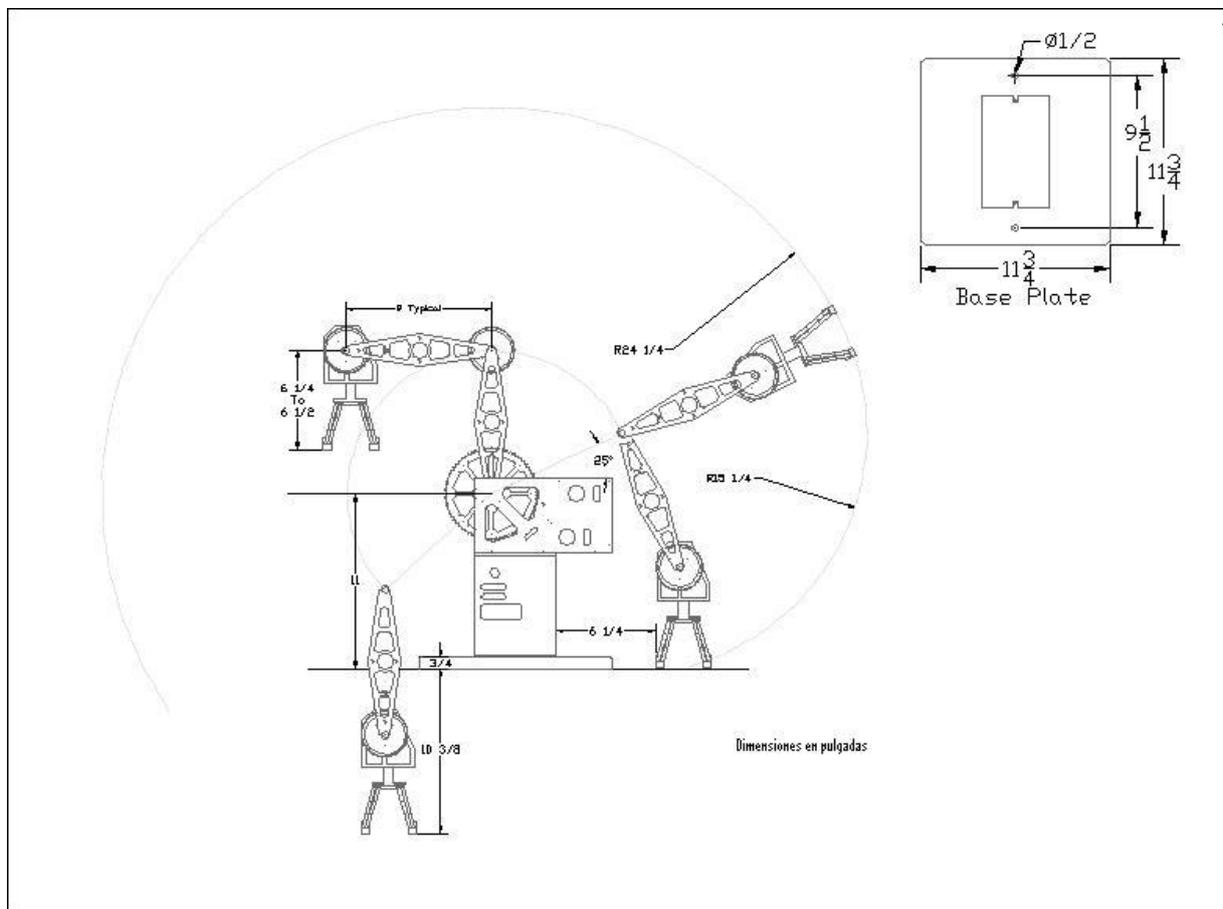


Figura A.2: Dimensiones del Robot Rhino XR-3

A.1. Modelo Matemático del Robot Rhino

El modelo matemático del robot Rhino describe la relación que existe entre las fuerzas y pares que actúan sobre cada grado de libertad. Un modelo matemático es de suma importancia, ya que a través de éste se pueden hacer simulaciones por computadora y diseñar métodos de control, sin embargo, la obtención de un buen modelo es complicado, de forma que aquí sólo se describirán las ecuaciones que satisfacen a cualquier robot manipulador.

El modelo matemático dinámico de un robot manipulador con n grados de libertad con eslabones rígidos sin fricción en las articulaciones y con actuadores ideales está dado por:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + g(\theta) = \tau \quad (\text{A.1})$$

Donde:

- $M(\theta)$ es la matriz de inercia
- $C(\theta, \dot{\theta})$ es el vector de fuerzas centrífugas y de Coriolis
- $g(\theta)$ es el vector de gravedad
- τ es el vector de fuerzas y pares aplicados a las articulaciones
- θ es la posición de las articulaciones
- $\dot{\theta}$ es la velocidad de las articulaciones
- $\ddot{\theta}$ es la aceleración de las articulaciones

A continuación se enlistan los elementos que componen a las matrices de la Ecuación A.1:

$$M_{11}(\theta) = m_2 l_{c2}^2 \cos^2 \theta_2 + 2m_3 l_2 l_{c3} \cos \theta_2 \cos(\theta_2 + \theta_3) + (m_3 l_{c3}^2 + m_f l_3^2) \cos^2(\theta_2 + \theta_3) + (m_3 + m_f) l_2^2 \cos^2(\theta_2) + 2m_f l_2 l_3 \cos \theta_2 \cos(\theta_2 + \theta_3) + I_1 + I_2 + I_3 \quad (\text{A.2})$$

$$M_{12}(\theta) = 0 \quad (\text{A.3})$$

$$M_{13}(\theta) = 0 \quad (\text{A.4})$$

$$M_{21}(\theta) = 0 \quad (\text{A.5})$$

$$M_{22}(\theta) = m_2 l_{c2}^2 + I_2 + m_3 l_{c3}^2 + m_3 l_2^2 + 2m_3 l_2 l_{c3} \cos \theta_3 + I_3 + m_f l_3^2 + m_f l_2^2 + 2m_f l_2 l_3 \cos(\theta_3) \quad (\text{A.6})$$

$$M_{23}(\theta) = m_3 l_{c3}^2 + m_3 l_2 l_{c3} \cos \theta_3 + I_3 + m_f l_3^2 + m_f l_2 l_3 \cos \theta_3 \quad (\text{A.7})$$

$$M_{31}(\theta) = 0 \quad (\text{A.8})$$

$$M_{32}(\theta) = m_3 l_{c3}^2 + m_3 l_2 l_{c3} \cos \theta_3 + m_f l_3^2 + m_f l_2 l_3 \cos \theta_3 + I_3 \quad (\text{A.9})$$

$$M_{33}(\theta) = m_3 l_{c3}^2 + m_f l_3^2 + I_3 \quad (\text{A.10})$$

$$\begin{aligned} C_{11}(\theta, \dot{\theta}) = & -(m_3 l_2 l_{c3} + m_f l_2 l_3) \sin \theta_2 \cos(\theta_2 + \theta_3) \dot{\theta}_2 - \\ & -(m_3 l_2 l_{c3} + m_f l_2 l_3) \cos \theta_2 \sin(\theta_2 + \theta_3) (\dot{\theta}_2 + \dot{\theta}_3) - \\ & -(m_3 l_{c3}^2 + m_f l_3^2) \sin(\theta_2 + \theta_3) \cos(\theta_2 + \theta_3) (\dot{\theta}_2 + \dot{\theta}_3) - (m_3 + m_f) l_2^2 \sin \theta_2 \cos \theta_2 \dot{\theta}_2 - \\ & -m_2 l_{c2}^2 \cos \theta_2 \sin \theta_2 \dot{\theta}_2 \end{aligned} \quad (\text{A.11})$$

$$\begin{aligned} C_{12}(\theta, \dot{\theta}) = & -(m_3 m_f) l_2^2 \sin \theta_2 \cos \theta_2 \dot{\theta}_1 - (m_3 l_2 l_{c3} + m_f l_2 l_3) \sin \theta_2 \cos(\theta_2 + \theta_3) \dot{\theta}_1 - \\ & -m_2 l_{c2}^2 \cos \theta_2 \sin \theta_2 \dot{\theta}_1 - (m_3 l_2 l_{c3} + m_f l_2 l_3) \cos \theta_2 \sin(\theta_2 + \theta_3) \dot{\theta}_1 - \\ & -(m_3 l_{c3}^2 + m_f l_3^2) \sin(\theta_2 + \theta_3) \cos(\theta_2 + \theta_3) \dot{\theta}_1 \end{aligned} \quad (\text{A.12})$$

$$C_{13}(\theta, \dot{\theta}) = -(m_3 l_2 l_{c3} + m_f l_2 l_3) \cos \theta_2 \sin(\theta_2 + \theta_3) \dot{\theta}_1 - (m_3 l_{c3}^2 + m_f l_3^2) \sin(\theta_2 + \theta_3) \cos(\theta_2 + \theta_3) \dot{\theta}_1 \quad (\text{A.13})$$

$$\begin{aligned} C_{21}(\theta, \dot{\theta}) = & -m_2 l_{c2}^2 \cos \theta_2 \sin \theta_2 \dot{\theta}_1 - (m_3 l_2 l_{c3} + m_f l_2 l_3) \cos \theta_2 \sin(\theta_2 + \theta_3) \dot{\theta}_1 - \\ & -(m_3 + m_f) l_2^2 \cos(\theta_2) \sin(\theta_2) \dot{\theta}_1 - (m_3 l_{c3}^2 + m_f l_3^2) \cos(\theta_2 + \theta_3) \sin(\theta_2 + \theta_3) \dot{\theta}_1 - \\ & -(m_3 l_2 l_{c3} + m_f l_2 l_3) \sin(\theta_2) \cos(\theta_2 + \theta_3) \dot{\theta}_1 \end{aligned} \quad (\text{A.14})$$

$$C_{22}(\theta, \dot{\theta}) = -(m_3 l_2 l_{c3} + m_f l_2 l_3) \text{sen}(\theta_3) \dot{\theta}_3 \quad (\text{A.15})$$

$$C_{23}(\theta, \dot{\theta}) = -(m_3 l_2 l_{c3} + m_f l_2 l_3) \text{sen}(\theta_3) \dot{\theta}_2 - (m_3 l_2 l_{c3} + m_f l_2 l_3) \text{sen}(\theta_3) \dot{\theta}_3 \quad (\text{A.16})$$

$$C_{31}(\theta, \dot{\theta}) = (m_3 l_2 l_{c3} + m_f l_2 l_3) \cos(\theta_2) \text{sen}(\theta_2 + \theta_3) \dot{\theta}_1 + (m_3 l_{c3}^2 + m_f l_3^2) \cos(\theta_2 + \theta_3) \text{sen}(\theta_2 + \theta_3) \dot{\theta}_1 \quad (\text{A.17})$$

$$C_{32}(\theta, \dot{\theta}) = (m_3 l_2 l_{c3} + m_f l_2 l_3) \text{sen}(\theta_3) (\dot{\theta}_2 + \dot{\theta}_3) \quad (\text{A.18})$$

$$C_{33}(\theta, \dot{\theta}) = -(m_3 l_2 l_{c3} + m_f l_2 l_3) \text{sen}(\theta_3) \dot{\theta}_2 \quad (\text{A.19})$$

$$g_1(\theta) = 0 \quad (\text{A.20})$$

$$g_2(\theta) = m_2 g l_{c2} \cos \theta_2 + (m_3 + m_f) g l_2 \cos \theta_2 + (m_3 l_{c3} + m_f l_3) g \cos(\theta_2 + \theta_3) \quad (\text{A.21})$$

$$g_3(\theta) = (m_3 l_{c3} + m_f l_3) g \cos(\theta_2 + \theta_3) \quad (\text{A.22})$$

A.2. Cinemática Inversa

Uno de los problemas que se tienen en las aplicaciones de robótica, es el cálculo de la posición de las articulaciones a partir de la posición del punto final conocido y se vuelve mas complejo cuando se incrementa el número de articulaciones que se deben calcular. Existen algunas técnicas que resuelven éste problema para configuraciones arbitrarias ([23], [11] y [24]), sin embargo, en este trabajo al identificarse tres puntos: hombro, codo y muñeca (Figura A.3) el problema se resuelve de manera geométrica, considerando que el eje Y va en sentido contrario, en una imagen se tienen los siguientes casos:

1. $0^\circ < \theta_1, \theta_2 < 90^\circ$

$$\theta_1 = \text{ArcTan} \frac{|Y_1 - Y_0|}{X_1 - X_0}, \quad \theta_2 = \text{ArcTan} \frac{|Y_2 - Y_1|}{X_2 - X_1} \quad (\text{A.23})$$

2. $\theta_1, \theta_2 < 0^\circ$

$$\theta_1 = \text{ArcTan} \frac{Y_0 - Y_1}{X_1 - X_0}, \quad \theta_2 = \text{ArcTan} \frac{Y_2 - Y_1}{X_2 - X_1} \quad (\text{A.24})$$

3. $\theta_1, \theta_2 > 90^\circ$

$$\theta_1 = 90^\circ + \text{ArcTan} \frac{X_0 - X_1}{Y_0 - Y_1}, \quad \theta_2 = 90^\circ + \text{ArcTan} \frac{X_1 - X_2}{Y_1 - Y_2} \quad (\text{A.25})$$

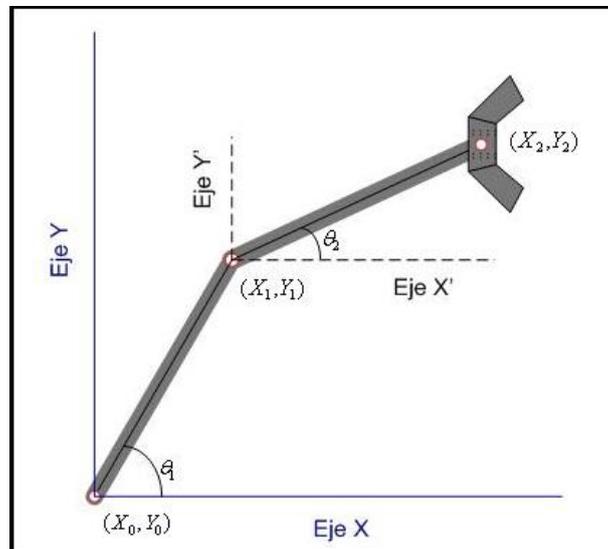


Figura A.3: Puntos identificados: hombro, codo y muñeca

Los ángulos obtenidos representan la posición en la cual se desea se encuentre el robot Rhino y se pueda hacer control de seguimiento del robot.

APÉNDICE B

Descripción Física de la Interface de Potencia

B.1. Conexiones

La interfaz de potencia empleada, fue la propuesta por Saint Martín [24]. Las conexiones de la interfaz se muestran en la Figura B.1. El significado de las letras es la siguiente:

- A, B, C, D, E, F son entradas de 10 pines para la conexión de los motores del robot Rhino.
- BA, BB, BC, BD, BE, BF son conectores BNC en donde se suministra el voltaje de control. Se conectan directamente a los canales de la tarjeta descrita en el apéndice C.
- EA, EB, EC, ED, EE, EF son conectores DB-9 de donde se obtiene la señal de salida de los encoders.

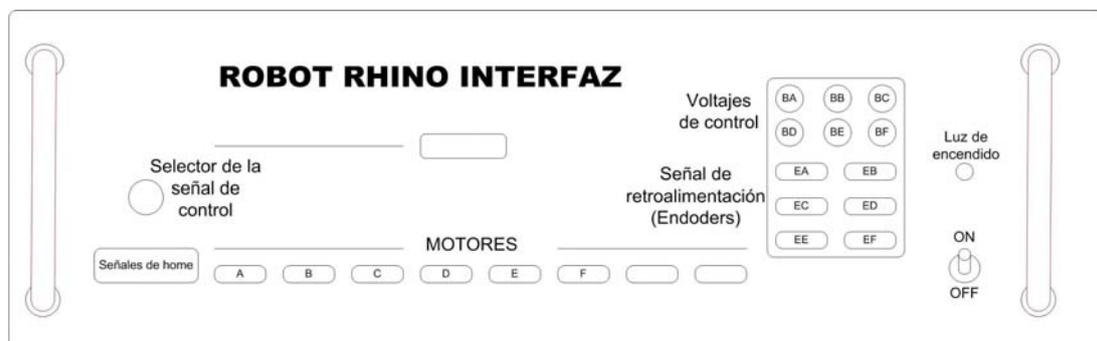


Figura B.1: Interfaz de control del Robot tipo Rhino

APÉNDICE C

Convertor Digital/Analógico

Las señales de control obtenidas son señales digitales lógicas, sin embargo, debido a que la etapa de potencia requiere de voltajes analógicos dentro de un rango de ± 10 V se diseñó un convertidor digital/analógico y que además además inhibe el movimiento del motor que no está en operación.

Se utilizó el puerto paralelo para enviar los datos digitales, flips-flops tipo D para hacer el cambio del motor que se quisiera controlar, convertidores DAC800N para hacer la conversión y una etapa de potencia para alcanzar los ± 10 V. A continuación se muestra el diagrama esquemático y las distintas vistas de la placa fenólica.

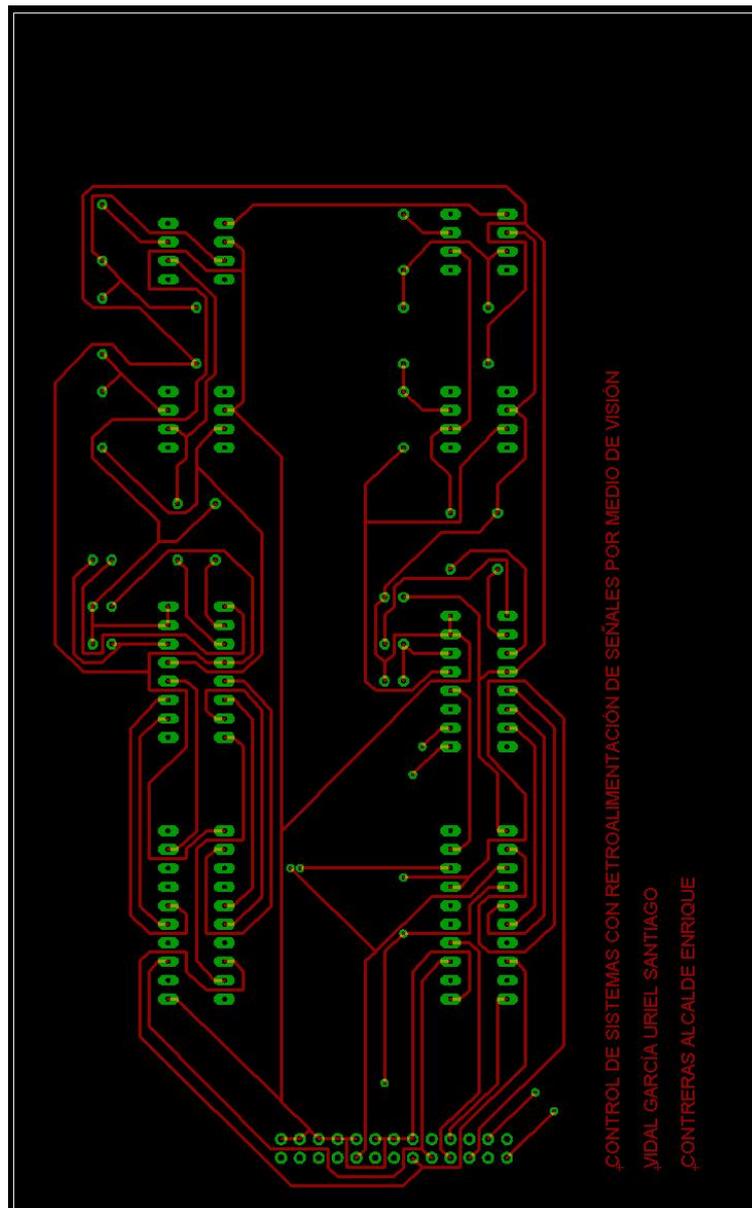


Figura C.2: Convertor analógico/digital (vista superior)

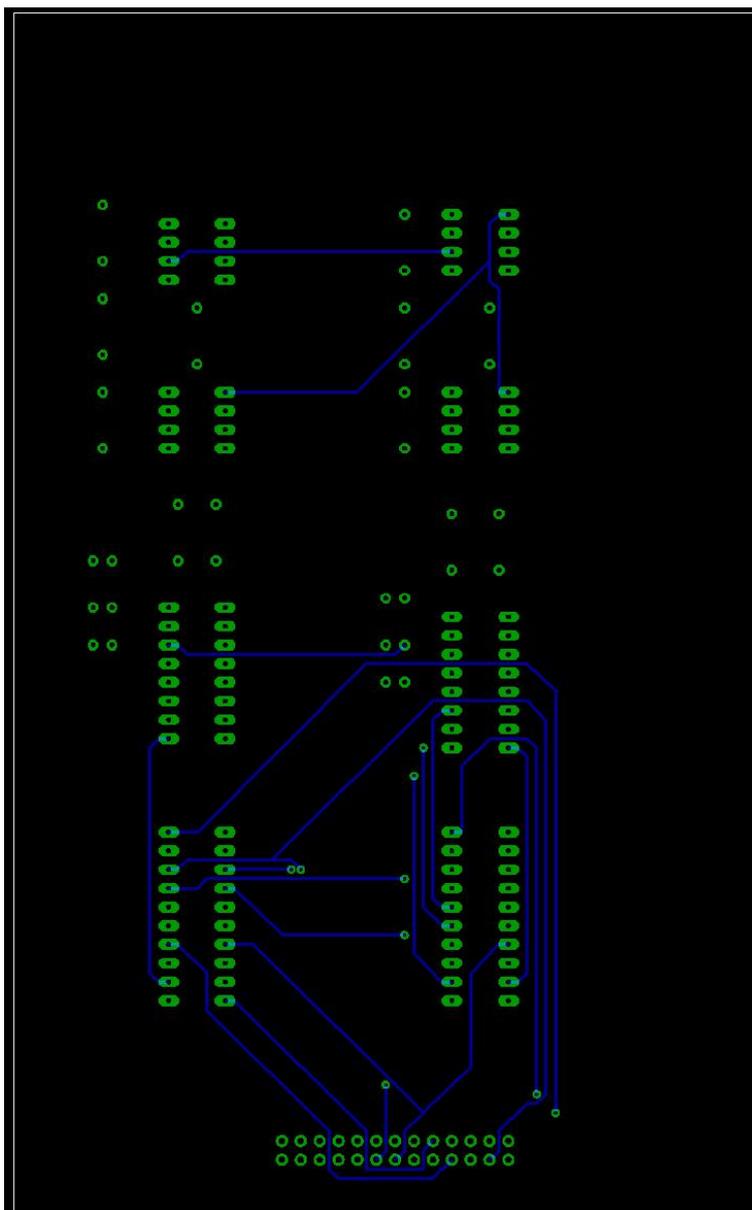


Figura C.3: Conversor analógico/digital (vista inferior)

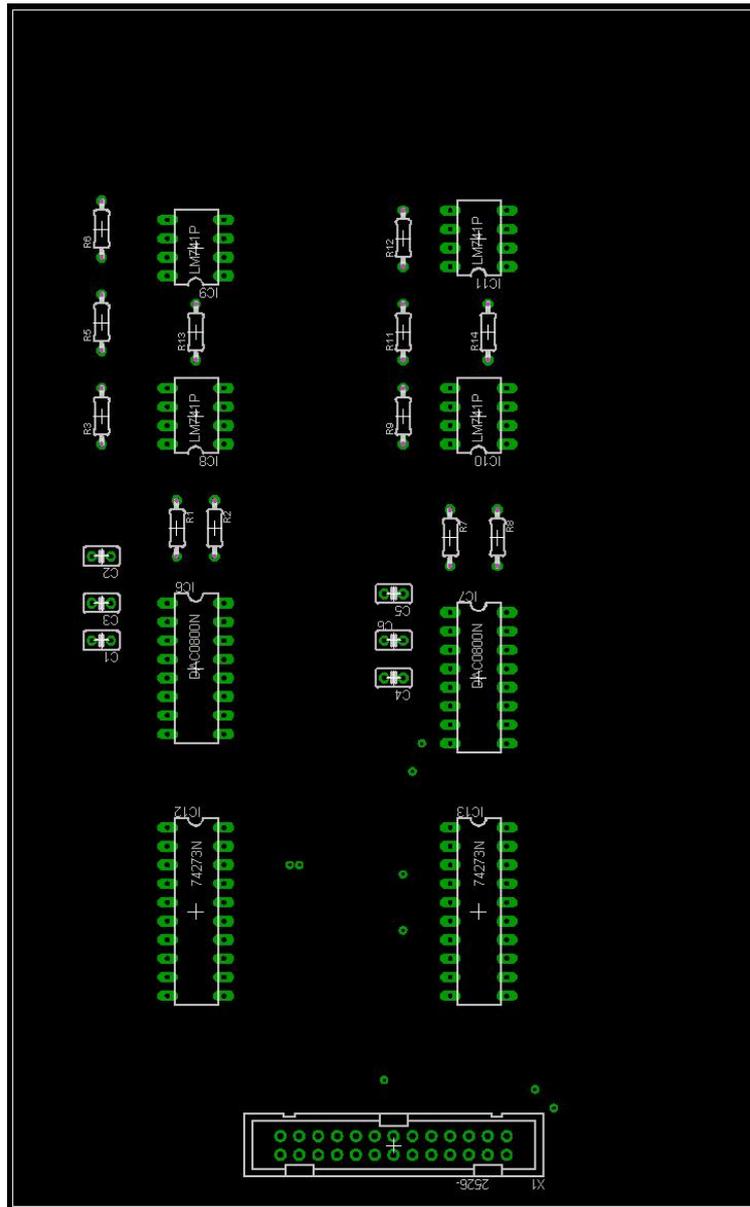


Figura C.4: Convertor analógico/digital (vista superior con componentes)

APÉNDICE D

Códigos Fuentes

Archivo de Inicio

```
%Camara
global dio;
dio = digitalio('parallel','LPT1');
addline(dio,0:7,0,'out');
addline(dio,0:3,2,'out');

dato1=dec2binvec(128,8);
mandarDatos273(dato1, dato1, dio);

global seno
j=1;
for i=0:0.14:450
    residuo=mod(i,0.14);
    if residuo == 0
        seno(j)=30*sin(0.03*i-0.4)+35;
        const=seno(j);
    else
        seno(j)=const;
    end
    j=j+1;
end
```

```
vid = videoinput('winvideo',1,'RGB24_640x480');
set(vid,'TriggerRepeat',3200);
set(vid,'FramesPerTrigger',1)
set(vid,'TriggerFcn', {'display_frame'});
set(vid,'StopFcn', {'parar'});

global rojo;
global azul;
global verde;
global xr;
global yr;
global xb;
global yb;
global xg;
global yg;
global count;
global teta1;
global teta2;
global tiempo;
global senalConHombro;
global senalConCodo;

global znm1;
global delta ;
global tnm1;
global qdnm1;
global qk2nm1;
global em1;

znm1=0;
delta =0;
tnm1=0;
qdnm1=0;
qk2nm1=0;
em1=0;

tiempo=1;
count=2;
xr=[10 10];
yr=[10 10];
xb=[10 10];
yb=[10 10];
xg=[10 10];
```

```
yg=[10 10];  
teta1=0;  
teta2=0;  
start(vid)
```

Función MandarDatos273

```
function [void] = mandarDatos273(dato1, dato2, dio)  
  
bvdata=[dato1 0 0 0 1];  
putvalue(dio,bvdata)  
  
bvdata=[dato1 0 0 0 0];  
putvalue(dio,bvdata)  
  
bvdata=[dato2 0 0 0 1];  
putvalue(dio,bvdata)  
  
bvdata=[dato2 0 0 1 1];  
putvalue(dio,bvdata)
```

Función Displayframe

```
function display_frame(obj,event)  
    %Obtener una imagen  
    im = getdata(obj,1);  
  
    global rojo;  
    global azul;  
    global verde;  
    global xr;  
    global yr;  
    global xb;  
    global yb;  
    global xg;  
    global yg;  
    global count;  
    global teta1;  
    global teta2;
```

```
global dio;
global tiempo;
%global seno;
%global triangulo
global sierra
global senalConHombro;
global senalConCodo;

rojo=imsubtract(im(:,:,1),im(:,:,3));
esc_rojo=round(max(max(rojo))*0.65);
%obtengo coordenada
[xrojo, yrojo]=detecpoint(max(rojo), max(transpose(rojo)), esc_rojo);
if (xrojo==0)
    xr(count)=xr(count-1);
else
    xr(count)=xrojo;
end
if (yrojo==480)
    yr(count)=yr(count-1);
else
    yr(count)=yrojo;
end
rojo(yr(count),xr(count))=255;

azul=imsubtract(im(:,:,3),im(:,:,2));
esc_azul=round(max(max(azul))*0.65);
[xazul, yazul]=detecpoint(max(azul), max(transpose(azul)), esc_azul);
if (xazul==0)
    xb(count)=xb(count-1);
else
    xb(count)=xazul;
end
if (yazul==480)
    yb(count)=yb(count-1);
else
    yb(count)=yazul;
end
azul(yb(count),xb(count))=255;

verde=imsubtract(im(:,:,2),im(:,:,1));
```

```
esc_verde=round(max(max(verde))*0.65);
[xverde, yverde]=detecpoint(max(verde), max(transpose(verde)),
esc_verde);
if (xverde==0)
    xg(count)=xg(count-1);
else
    xg(count)=xverde;
end
if (yverde==480)
    yg(count)=yg(count-1);
else
    yg(count)=yverde;
end
verde(yg(count), xg(count))=0;
```

```
[T1, T2]=invcinematica(xr(count), yr(count), xg(count),
yg(count), xb(count), yb(count));
teta2(count)=T2;
tetal(count)=T1;
```

```
tetad1 = seno(tiempo);
tetad2 = seno(tiempo);
```

```
tiempo = tiempo + 1;
```

```
torqueFuzzy1=fuzzy(tetal(count), tetad1);
torqueFuzzy2=fuzzy(teta2(count), tetad2);
```

```
torqueTotal1 = torqueFuzzy1;
torqueTotal2 = torqueFuzzy2;
```

```
voltajeTotal1 = torqueTotal1 * 0.65;
voltajeTotal2 = torqueTotal2 * 0.4;
```

```
if voltajeTotal1 > 8
    voltajeTotal1 = 8;
elseif voltajeTotal1 < -8
    voltajeTotal1 = -8;
```

```
end
if voltajeTotal2 > 8
    voltajeTotal2 = 8;
elseif voltajeTotal2 < -8
    voltajeTotal2 = -8;
end

senalConHombro(tiempo)=voltajeTotal1;
senalConCodo(tiempo)=voltajeTotal2;

voltBinario1=datobinario3(voltajeTotal1);
voltBinario2=datobinario3(voltajeTotal2);

mandarDatos273(voltBinario1, voltBinario2, dio)

count=count+1;

abstime = event.Data.AbsTime;

t = fix(abstime);

sprintf('%s %d:%d:%d','timestamp', t(4),t(5),t(6));
```

Función Detecpoint

```
function [xp, yp]=detecpoint(col, ren, maxescalar)
    contcol=0;
    contren=0;
    j=0;
    i=1;
    while i<640
        if col(i)>=maxescalar
            h=i+1;
            contcol=contcol+1;
            while col(h)>=maxescalar
                contcol=contcol+1;
                h=h+1;
                if h>640
```

```
                break
            end
        end
        if contcol>=15
            xp=i+11;
            h=640;
            j=1;
        end
        contcol=0;
        i=h;
    end
    i=i+1;
end
if j==0
    xp=0;
end
j=0;
i=1;
while i<480
    if ren(i)>=maxescalar
        k=i+1;
        contren=contren+1;
        while ren(k)>=maxescalar
            contren=contren+1;
            k=k+1;
            if k>480
                break
            end
        end
        if contren>=15
            yp=i+11;
            k=480;
            j=1;
        end
        contren=0;
        i=k;
    end
    i=i+1;
end
if j==0
    yp=480;
end
end
```

Función Invcinematica

```
function [teta1, teta2]= invcinematica(X0, Y0, X1, Y1, X2, Y2)
if (X0-X1)>0
    teta1=atand((X0-X1)/(Y0-Y1))+90;
elseif (Y1-Y0)<0
    teta1=atand((abs(Y1-Y0))/(X1-X0));
else
    teta1=atand((Y0-Y1)/(X1-X0));
end

if (X1-X2)>0
    teta2=atand((X1-X2)/(Y1-Y2))+90;
elseif (Y2-Y1)<0
    teta2=atand((abs(Y2-Y1))/(X2-X1));
else
    teta2=atand((Y1-Y2)/(X2-X1));
end
```

Función Fuzzy

```
function [volt] = fuzzy(teta, tetad)

e = teta - tetad;

EP = 2.5;
EM = 25;
JM = -10;
JP = -6;

P=1/(1+exp(-5*(teta-40)));
N=1/(1+exp(+5*(teta+40)));

PD=1/(1+exp(-3*(tetad-40)));
ND=1/(1+exp(+3*(tetad+40)));

EPG=1/(1+exp(-10*(e-1)));
```

```
ENG=1/(1+exp(+10*(e+1)));  
Z=1;
```

```
W = [P*PD*ENG P*PD*Z P*PD*EPG P*ND*ENG P*ND*Z P*ND*EPG N*PD*ENG  
N*PD*Z N*PD*EPG N*ND*ENG N*ND*Z N*ND*EPG ];
```

```
Ua =[EP+EM EP EP+JM 0 JM JM+JM EM+EM EM*0 0 JP+EM JP*0 JP+JM];
```

```
volt=W*transpose(Ua)/sum(W);
```

Bibliografía

- [1] Mohamed Jamshidi, Mohamed Vadiie, and Timothy Ross. *Fuzzy logic and control, software and hardware applications*. Prentice Hall, 1993.
- [2] Gómez-Allende y Dario Maravall. *Reconocimiento de formas y visión artificial*. Addison Wesley Iberoamericana, 1993.
- [3] Penwell. *Vision System Design*. december 2006, january and april 2007.
- [4] Y. Sumi, Y. Kawai, T. Yoshimi, and F. Tomita. Recognition of 3d free-form objects using segment-based stereo vision. *Intelligent Systems Division, Electrotechnical Laboratory*.
- [5] D. Park and J Kim. 3d hand-eye robot vision system using a cone-shaped structured light. *SICE-ICASE International Joint Conference*, 2006.
- [6] Gonzalo Pajares Martinsanz y Jesús Maria De la Cruz García. *Visor por Computador, Imágenes Digitales y Aplicaciones*. Alfaomega, 2002.
- [7] Janne Hekkilä and Olli Silvén. A four-step camera calibration procedure with implicit image correction. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997.
- [8] E. Trucco and A. Verri. *Introductory to techniques for 3-D computer vision*. Prentice Hall, 1998.
- [9] R. Y. Tsai. A versatile camera calibration technique for high accuracy 3d machine vision metrology using oof-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 1987.
- [10] S. Matsuda and Y. Maeda. Motion control system via knowledge database using visual information. *SICE-ICASE International Joint Conference*, 2006.
- [11] Sebastián Ibarra Rojas. Control de robot por retroalimentación visual utilizando redes neuronales, 2001.

-
- [12] B. Tuyen and P. Cat. Neural network based visual control. *Seventh International Conference on Canbol, Automation*, 2002.
- [13] J. Alberto Chavez Aragón. Diseño y construcción de un brazo robótico pedagógico jugador de gato, dotado de un sistema básico de visión artificial, 1999.
- [14] Dong-Jin Park and John-Hyeong Kim. 3d hand-eye robot vision system using a cone-shaped structured light. *SICE-ICASE International Joint Conference*, 2006.
- [15] Zijian Zhao and Yuncai Liu. Hand-eye calibration based on screw motions. *Inst. of image processing and pattern recognition*.
- [16] Jie Wang, Zongwu Xie, Haiying Hu, and Bin Wang. Research on hand-eye robot. *International Conference on Mechatronics and Automation*, 2006.
- [17] L. A. Zadeh. Fuzzy sets. pages 338–353, 1965.
- [18] Dimiter Driankov, Hans Hellendoorn, and Michael Reinfrank. *An introduction to fuzzy control*. Springer, 1996.
- [19] J.R. Parker. Algorithms for image processing and computer vision. *Wiley Computer Publishing*, 1997.
- [20] Gonzalo Pajares Martinsanz. *Imágenes Digitales: Procesamiento Práctico con Java*. Ra-Ma, 2003.
- [21] Xavier Zubiri. El concepto descriptivo del tiempo. *Realitas. II*, 1976.
- [22] Donald Gillies. What exactly is meant by real-time? <http://www.faqs.org/faqs/realtime-computing/faq/>.
- [23] Mark W. Spong and Vidyasagar. *Robot Dynamics and Control*. Wiley, 1989.
- [24] Ricardo Saint Martin de Maria y Campos. Control difuso para un robot tipo rhino : construcción de interfaz para computadora y resultados experimentales, 2002.