



**“DETECCIÓN, IDENTIFICACIÓN POR SU FORMA Y  
MANIPULACIÓN DE OBJETOS EN TIEMPO REAL USANDO UN  
BRAZO DE ROBOT INDUSTRIAL”**

**Memoria de Residencias Profesionales**

Que para obtener el título de:  
**INGENIERO EN SISTEMAS COMPUTACIONALES**

Presenta:  
Norma Verónica Martínez Luévano

Con la asesoría de:  
Dr. Víctor Manuel Zamudio Rodríguez

León, Guanajuato. Febrero de 2013



## **Dedicatoria**

### **A Dios:**

*¡Bendito seas, Dios mío, por atender a mis ruegos!  
Tú eres mi fuerza; me proteges como un escudo. En ti confío de corazón, pues de ti recibo ayuda.  
El corazón se me llena de alegría, por eso te alabo en mis cantos. Salmos 28: 6-7 TLA*

Porque sin tu ayuda nada hubiera sido posible. Por tu incomparable amor y misericordia, por tu gracia y favor sobre mi vida, por la sabiduría que solo puede provenir de lo alto, por tu presencia en medio de las pruebas. Por ser mi Padre, mi amigo, mi Señor y Salvador. Porque todo es por ti y para ti.

### **A mi padre (Sergio Martínez Solís):**

Por ser un gran ejemplo a seguir, por proveer mis necesidades aun con sacrificios y esfuerzos, por estar a mi lado en cada momento importante de mi vida y en aquellos que no lo son, por cada noche orar por mí aún en medio de su cansancio, por las tantas risas provocadas con sus comentarios únicos y originales, por levantarse muy temprano para llevarme a tiempo a la escuela o solo acompañarme a tomar el camión.

### **A mi madre (Lourdes Luévano Rodríguez):**

Por ser un apoyo en todo tiempo, por acompañarme en las noches de desvelo y preparar el café para no dormir, por soportar el estrés del fin de semestre y todos mis cambios de humor, por las comidas tan deliciosas al llegar a casa después de un día pesado, por cada palabra de amor a pesar de mi frialdad, por estar presente en mis logros, compartir mis alegrías y llorar conmigo en los fracasos.

### **A mis hermanos (Israel y Josué):**

Por hacer mis días menos pesados, y ponerles alegría a cada uno de ellos, por ponerle música de fondo a mi vida y estar siempre a mi lado, por jugar, reír, pelear o solo platicar conmigo. Por amarme y ser mis hermanos.

### **A mis amigos:**

Que más que amigos han sido como una familia, por estar a mi lado siempre y mostrarme su amor, por sus sabios consejos y ayudarme a crecer. Por ser un apoyo y una pieza clave en mi vida.

### **A mis maestros y compañeros:**

Por cada enseñanza, por cada clase compartida, por toda la ayuda cuando no entendía algo, por la paciencia, por cada experiencia y por cada año compartido.



## Resumen

En este trabajo se muestran los avances del desarrollo de una aplicación creada en Matlab en conjunto con el sistema propio del Robot KUKA KR-16.

El proyecto simula una banda transportadora en la cual se encuentran diferentes productos cuya variante es su forma geométrica. Estos productos serán clasificados y en base a ello serán trasladados al lugar (empaquete) que les corresponda por medio de un brazo de robot industrial y una cámara digital.

El proceso inicia una vez que, por medio de la cámara el algoritmo de detección, encuentra un objeto dentro del área definida, cuando ya se detecto el objeto es necesario procesar la imagen para poder realizar el reconocimiento de la forma.

Después de reconocer la forma del producto es necesario identificar la posición dentro de la banda para que de este modo el brazo pueda moverse hasta el punto en el cual se encuentra el objeto.

Una vez que se tiene dicha información el brazo de robot se encargara de tomar el producto y llevarlo a su destino, el cual será una serie de cajas las cuales contendrán diferentes tipos de objetos y el brazo de robot será capaz de identificar la caja a la que corresponde el objeto.

El brazo de robot permanecerá inactivo siempre y cuando no detecte alguna pieza, una vez que una nueva pieza es detectada el brazo de robot sigue con la ejecución normal del programa.

# Índice

Objetivos .....	1
Objetivo General .....	1
Objetivos Específicos.....	1
Alcances y Limitaciones .....	2
Capítulo 1. Introducción .....	3
1.1 Introducción a la robótica .....	3
1.2 Automatización.....	4
1.3 Robots industriales.....	5
Capítulo 2. Sistema de Robot KUKA KR-16.....	10
2.1 Elementos del sistema de robot .....	10
2.1.1 Armario KR C2.....	10
2.1.2 KRC Software .....	13
2.1.3 Panel de Control KUKA (KCP).....	14
2.2 Sistema de Coordenadas.....	16
2.2.1 Especifico de ejes .....	16
2.2.2 WORLD.....	17
2.2.3 BASE .....	17
2.2.4 TOOL .....	18
2.3 Desplazamiento manual.....	18
2.3.1 Generalidades.....	18
2.3.2 Tipo de desplazamiento .....	19
2.3.3 Desplazamiento con Space-Mouse.....	19
2.3.4 Desplazamiento con teclas de desplazamiento .....	20
2.4 Programación.....	21
2.4.1 Movimiento Punto a Punto (PTP).....	21
2.4.2 Movimiento lineal (LIN) .....	22
2.4.3 Movimiento Circular (CIRC) .....	23

2.4.4 Funciones lógicas .....	25
Capítulo 3. Comunicación Robot – PC.....	26
3.1 Puerto Serial .....	26
3.1.1 Puerto serial virtual .....	28
3.1.2 USB CDC.....	28
3.2 USB – KUKA KR16 .....	30
3.2.1 Conector USB .....	30
3.2.2 Micro controlador .....	32
3.3 Entradas/ salidas.....	32
3.3.1 Entradas digitales.....	33
3.3.2 Salidas digitales .....	34
Capítulo 4. Reconocimiento de Piezas.....	35
4.1 Sistema de visión .....	35
4.1.1 Cámara digital .....	35
4.1.2 Iluminación.....	36
4.2 Procesamiento de imágenes.....	37
4.2.1 Binarización .....	38
4.2.2 Operaciones morfológicas .....	39
4.2.3 Segmentación .....	42
4.3 Reconocimiento .....	43
4.3.1 Redimensionar imagen .....	44
4.3.2 Suma de píxeles .....	44
4.3.3 Determinación de forma.....	45
4.3.4 Determinación de posición.....	46
Capítulo 5. Conclusiones.....	48
Conclusión del proyecto.....	48
Conclusión Personal .....	49
Bibliografía .....	50

## Índice de Figuras

Figura 1.1 Robot antropomórfico.....	4
Figura 1.2 Brazo manipulador de herramienta .....	5
Figura 1.3 Robot Inteligente capaz de tomar decisiones propia.....	7
Figura 2.1 Armario KR C2 .....	8
Figura 2.2 Interruptor principal .....	9
Figura 2.3 KRC Software .....	11
Figura 2.4 Entorno KRC Software .....	11
Figura 2.5 Panel de control (KCP).....	12
Figura 2. 6 Sistema específico de ejes.....	14
Figura 2. 7 Sistema de coordenadas WORLD .....	15
Figura 2.8 Coordenadas base referidas a WORLD .....	15
Figura 2. 9 Ejemplo coordenadas TOOL con una servopinza .....	16
Figura 2. 10 Mouse-Space .....	17
Figura 2. 11 Barra de teclas de estado .....	18
Figura 2.12 Movimiento PTP parada exacta .....	19
Figura 2.13 Movimiento PTP sin llegar a tocar el punto intermedio programado	20
Figura 2.14 Movimiento LIN con parada exacta .....	21
Figura 2.15 Movimiento LIN aproximado.....	21
Figura 2.16 Movimiento CIRC con parada exacta.....	22
Figura 2.17 Movimiento CIRC aproximado.....	22
Figura 3.1 Conexión computadora- microcontrolador .....	24



Figura 3.2 Cambios necesario para emigrar una aplicación de puerto RS-232 a USB.....	25
Figura 3.3 Visualización de la detección del dispositivo externo con un puerto serial virtual asignado por el sistema operativo.....	26
Figura 3.4 Ventana de Windows que muestra el estado de la conexión .....	27
Figura 3.5 Hilos de conexión .....	29
Figura 4.1: Sistema de visión .....	33
Figura 4.2 Imagen con alto contraste e Iluminación frontal .....	34
Figura 4.3 Selección de cámara.....	34
Figura 4.4 Sistema de iluminación .....	35
Figura 4.5 Imagen binarizada.....	36
Figura 4.6 Imagen con formas de menores de 100 pixeles sin eliminar y a la derecha dichas formas eliminadas .....	38
Figura 4.7 Circulo incompleto eliminado .....	38
Figura 4.8 Separación de imágenes unidas por algunos pixeles .....	39
Figura 4.9 Rellenado de agujeros .....	39
Figura 4.10 Región de interés segmentada .....	40
Figura 4.11 Matriz etiquetada.....	41
Figura 4.12 Imágenes segmentadas redimensionadas.....	42
Figura 4.13 Calculo del valor máximo del área para el circulo .....	43
Figura 4.14 Ejes de Cuadrado y rectángulo .....	44
Figura 4.15 Imagen segmentada donde se muestran las posibles combinaciones de coordenadas.....	44
Figura 4.16 Extracción de la malla para identificación de posición .....	45

## **Objetivos**

### **Objetivo General**

Desarrollar una aplicación con brazo de robot industrial de 6 grados de libertad marca KUKA para identificar objetos por su forma geométrica y color y manipularlos según su identificación en el espacio usando procesamiento de imágenes, creando con esto un sistema autónomo

### **Objetivos Específicos**

1. Implementar un algoritmo de detección de objetos y su posición en el espacio de la imagen adquirida con una cámara digital.
2. Programar la secuencia de instrucciones para que el brazo de robot pueda realizar tareas automatizadas con retroalimentación de una cámara digital.
3. Implementar un sistema de entradas salidas para la comunicación entre el algoritmo de detección en la PC y el centro de control del brazo del robot.
4. Se hará un análisis y reconocimiento de formas por medio de una cámara digital y realizando el procesamiento mediante un programa realizado en Matlab en el cual se identificara la forma o figura del objeto detectado y dicha información se la mandará, por medio del puerto serial, al brazo de robot industrial.

## **Alcances y Limitaciones**

La ejecución de este proyecto será como un prototipo debido a que se puede aplicar en muchas industrias que requieran automatizar sus procesos de producción y que cuenten con un brazo de robot industrial, solo realizándole ligeras modificaciones relacionadas con los espacios físicos.

En el presente reporte no se incluyen pruebas realizadas al sistema debido al tiempo para la realización, sin embargo durante la realización de las distintas etapas se fueron probando individualmente.

Este proyecto genera las bases para su utilización o modificación posterior y debido al tamaño del proyecto aun se puede seguir trabajando con el mismo.

## **Capítulo 1. Introducción**

Este reporte muestra los avances del desarrollo de la aplicación con el brazo de robot para identificar objetos por su forma geométrica y manipularlos según su identificación en el espacio usando procesamiento de imágenes.

El desarrollo esta creado por tres partes fundamentales, el robot manipulador, el modulo de comunicación y el proceso de detección. Dichas partes son presentadas en cada capítulo.

### **1.1 Introducción a la robótica**

Es importante iniciar describiendo lo que es un robot. La definición aceptada internacionalmente para robot es: manipulador multifuncional y reprogramable, diseñado para mover materiales, piezas, herramientas o dispositivos especiales mediante movimientos programables y variados que permiten llevar a cabo diversas tareas.

El termino robot se utilizo por primera vez en 1917 en una obra de teatro para denominar a unas máquinas construidas por el hombre y dotadas de inteligencia.

La ciencia que estudia este tipo de maquinas se denomina robótica, y podríamos definirla como el diseño, fabricación y utilización de maquinas automáticas programables con el fin de realizar tareas repetitivas.

La robótica ha cumplido 40 años de aplicación en procesos industriales, con más de medio millón de robots instalados en industrias de todo el mundo, principalmente en el sector automotriz; sin embargo es una tecnología que se ha introducido en la metalmecánica, química, papelera, cerámica, alimentación y que sigue abriendo caminos en muchos otros sectores incluyendo la medicina y los servicios.

La robótica tiene como intención final complementar o sustituir las funciones de los humanos, alcanzando en algunos sectores, aplicaciones masivas. En el contexto industrial, donde se utilizan con notable éxito desde hace varias décadas, sus beneficios empresariales y sociales se pueden resumir en 4:

1. **Productividad:** Significa aumento de la producción así como reducción de costos, mas significativamente laborales, de materiales, energéticos y de almacenamiento.
2. **Flexibilidad:** Permite la fabricación de productos del mismo tipo por mucho tiempo sin la necesidad de que cambie, por cansancio o cambio de turno la producción y, por consecuencia, no habrá paradas ni pérdidas de tiempo.
3. **Calidad:** Debido al alto nivel de repetitividad en las tareas realizadas por los robots, se asegura una calidad uniforme del producto final.
4. **Seguridad:** Debido a que los procesos de fabricación se llevan a cabo con un número mínimo de personas, disminuyen las posibilidades de accidentes laborales, además de reemplazar a los operadores de tareas tediosas y cansadas.

## 1.2 Automatización

Se puede definir automatización como el uso de sistemas o elementos computarizados y electromecánicos para controlar maquinarias y/o procesos industriales sustituyendo a operadores humanos.

La automatización no solo es un sistema de control, sino que también incluye el uso de sensores, instrumentación industrial, supervisión y aplicaciones de software.

Desde tiempos antiguos el hombre ha buscado simplificar los trabajos y el esfuerzo al realizarlos, primeramente lo hizo con la creación de maquinas, las cuales ellos solo dirigían. Ahora busca reducir la intervención humana en los procesos por medio de nuevas maquinas que controlen otras maquinas. Es decir, busca trasladar las tareas realizadas normalmente por operadores humanos a un conjunto de elementos tecnológicos.

Actualmente la parte más importante o la que más se nota en la automatización es la robótica industrial, y esto debido a algunas ventajas tales como la repetitividad en procesos, mejoría de la calidad en productos, mayor eficiencia, incremento de productividad y reducción de trabajo.

Una de las principales razones para incluir robots en el desarrollo industrial y sustituir al ser humano es debido a las condiciones de riesgo dadas en ciertas actividades, como por ejemplo las altas o bajas temperaturas que pueden afectar en gran manera a un humano y no afectar en lo absoluto a un robot.

Otra razón puede ser que en la realización de trabajos cíclicos el ser humano tiende a reducir la eficiencia debido al cansancio físico o mental y un robot no se va a cansar y puede estar realizando el mismo proceso por horas o incluso solo dedicarse a eso durante su tiempo de vida, y esto lo puede hacer sin reducir la calidad o eficiencia y aun con mayor rapidez que el ser humano.

Un pensamiento común respecto a la automatización es que es una fuerte causa por la cual se genera desempleo. Pero esto se da porque las empresas eligen despedir a un operario en lugar de cambiar sus tareas, por ejemplo a la supervisión de dichas tareas.

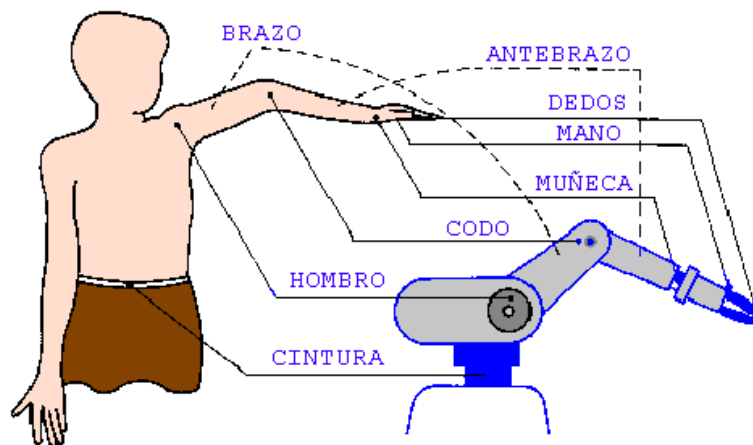
### **1.3 Robots industriales**

Los robots industriales surgen de la convergencia de tecnologías del control automático y en particular, del control de maquinas y herramientas, de los

manipuladores tele-operados, y de la aplicación de computadoras en tiempo real. Mediante el control automático de procesos se pretende concebir y realizar ingenios que permitan gobernar un proceso sin la intervención de agentes exteriores, especialmente el hombre. En la actualidad el uso de los robots industriales está concentrado en operaciones muy simples, como tareas repetitivas que no requieren tanta precisión.

Con esto diríamos que una posible definición de robot industrial podría ser: “una máquina programable, de propósito general, que posee ciertas características antropomórficas, es decir, con características basadas en la figura humana”.

Cabe destacar que la característica antropomórfica más común en nuestros días es la de un brazo mecánico (Figura 1.1)



*Figura 1.1 Robot antropomórfico*

Existen en el mercado diversas empresas dedicadas a la fabricación de robots industriales por lo que existen diferentes marcas y modelos. Las más conocidas son las empresas europeas tales como ABB, KUKA, COMAU, STAUBLI; que casi dominan la totalidad del mercado.

Dentro de los robots industriales podemos encontrar distintas clasificaciones dependiendo de sus características y/o funcionalidad. Dicha clasificación se cita a continuación:

## *Manipuladores*

Son sistemas mecánicos multifuncionales, con un sencillo sistema de control, que permite gobernar el movimiento de sus elementos, de los siguientes modos:

- Manual: Cuando el operario controla directamente la tarea del manipulador.
- De secuencia fija: cuando se repite, de forma invariable, el proceso de trabajo preparado previamente.
- De secuencia variable: Se pueden alterar algunas características de los ciclos de trabajo.

Existen muchas operaciones básicas que pueden ser realizadas óptimamente mediante manipuladores, por lo que se debe considerar seriamente el empleo de estos dispositivos, cuando las funciones de trabajo sean sencillas y repetitivas.



*Figura 1.2 Brazo manipulador de herramienta*

## *Robots de repetición o aprendizaje*

Son manipuladores que se limitan a repetir una secuencia de movimientos, previamente ejecutada por un operador humano, haciendo uso de un controlador manual o un dispositivo auxiliar. En este tipo de robots, el operario en la fase de enseñanza, se vale de un panel de programación con diversos pulsadores o teclas, o bien, de joysticks, o bien utiliza un maniquí, o puede desplazar directamente la mano del robot.



Los robots de aprendizaje son los más conocidos, hoy día, en los ambientes industriales y el tipo de programación que incorporan, recibe el nombre de "gestual"

### *Robots controlados por computadora*

Son manipuladores o sistemas mecánicos multifuncionales, controlados por una computadora, que habitualmente suele ser una microcomputadora.

En este tipo de robots, el programador no necesita mover realmente el elemento de la maquina, cuando la prepara para realizar un trabajo. El control por computadora dispone de un lenguaje específico (como en el caso de KUKA KR-16), compuesto por varias instrucciones adaptadas al robot, con las que se puede confeccionar un programa de aplicación utilizando solo la salidas de la computadora, no el brazo. A esta programación se le denomina textual y se crea sin la intervención del robot manipulador.

Las grandes ventajas que ofrecen este tipo de robots, hacen que se vayan imponiendo en el mercado rápidamente, lo que exige la preparación urgente de personal calificado, capaz de desarrollar programas similares a los de tipo informático.

### *Robots inteligentes*

Son similares a los del grupo anterior, pero además, son capaces de relacionarse con el mundo que les rodea a través de sensores y tomar decisiones en tiempo real (auto programable). De momento, son muy poco conocidos en el mercado y se encuentran en fase experimental, en la que se esfuerzan los grupos de investigación por potenciarles y hacerles más efectivos, al mismo tiempo que más accesibles en cuanto a precio.

La visión artificial y otras áreas de investigación parte de la inteligencia artificial, son las ciencias que más están estudiando para su aplicación en los robots inteligentes.



*Figura 1.3 Robot Inteligente capaz de tomar decisiones propias*

### *Microrobots*

Con fines educacionales, de entretenimiento o investigación, existen numerosos robots de formación o micro-robots a un precio muy accesible, cuya estructura y funcionamiento son similares a los de aplicación industrial.

## Capítulo 2. Sistema de Robot KUKA KR-16

### 2.1 Elementos del sistema de robot

El robot manipulador KUKA KR-16, es de tipo industrial y cuenta con 6 grados de libertad tomando en cuenta el actuador final (Gripper) encargado de realizar la sujeción.

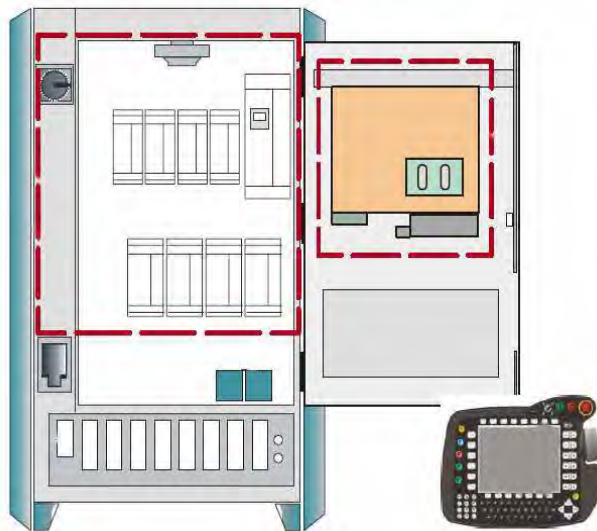
La carga nominal del manipulador es de 16 kg aplicados en la muñeca y una carga adicional de 10 kg colocada en el brazo.

Los manipuladores no son muy precisos, pero si tienen una repetitividad alta, en este caso es de  $\pm 0.1$  mm.

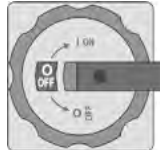
Los elementos básicos que integran un sistema KUKA, como cualquier otro tipo de sistemas robotizados; son el armario de control, el panel de control (KCP) y un prototipo de brazo. Todos estos comunicados por un software (KRC)

#### 2.1.1 Armario KR C2

El armario de control es el elemento principal donde se integran todos los componentes que suministran potencia y controlan los mecanismos del KR-16.



*Figura 2.1 Armario KR C2*



*Figura 2.2 Interruptor principal*

Con el interruptor principal se conectan o desconectan el sistema de robot y la unidad de control. Colocando un candado en el interruptor principal, se puede impedir de forma segura una conexión no autorizada.

En la parte inferior del armario se encuentran las conexiones; esto es, cables de alimentación/RED, cables de motor a robot, cables de señales del robot y conexión a KCP. El equipamiento estándar incluye además en el interior del armario, puerto Device Net y Ethernet para los sistemas de BUS habituales. Desde la KCP y a través del ordenador de a bordo, se maneja y opera la unidad de control del robot

### *PC Control*

El PC de control opera bajo aplicación de Windows que está destinado única y exclusivamente a la aplicación KRL (software de KUKA) y debe estar dimensionado con un mínimo de frecuencia de reloj de 733 MHz y una memoria RAM de 256 MB. A su vez incorpora las unidades de disco anteriormente mencionadas accesibles cuando la puerta del armario está abierta.

El resto del equipo es el Hardware estándar (fuente de alimentación, tarjeta gráfica KUKA con conexión a KCP). Existe la posibilidad de conectar un ratón estándar a través del puerto de comunicaciones COM1, así como un ratón/teclado con Interface PS/2. Se integra a modo de tarjeta PCI la tarjeta multifunción MFC con tarjeta digital Servo DSE.

La MFC contiene el controlador de todas las entradas y salidas a modo de diferentes tipos de conexión Device Net/CAN formando la Interface entre la KCP y el ordenador, mientras que la DSE toma los valores nominales de los servos en

cuanto a consumo y parametrización se refiere ; procesa por lo tanto, el estado de los servo controladores.

### *Acumuladores*

Estas baterías mantienen activa la memoria intermedia del sistema, de manera que los datos del ordenador se salven correctamente. Esto será siempre así, mientras la carga de los acumuladores no baje por debajo de los 19V, situación que será advertida mediante mensaje en el próximo arranque del equipo.

### *Servo convertidores*

Los servo convertidores obtienen la etapa final rectificada para el control de cada uno de los ejes mediante regulación de corriente. Todos los KSD están conectados por interbus para la comunicación y conmutación entre ejes.

### *Conector*

A través del conector de Interface X11, se conectan seguridades y setas de emergencia de la periferia concatenada con el sistema de KUKA. Dicho conector consta de una serie de pines desde lo que se interconectan los elementos de seguridad y se puntean aquellos que son necesarios para manejar el sistema.

En la Interface X11 se puede conectar la entrada de protección del operario, que sirve para el ínter bloqueo de dispositivos seccionadores de protección. A esta entrada bicanal pueden conectarse dispositivos de protección tales como puertas de protección. Si a esta entrada no se conecta nada, no es posible el modo de servicio automático. Para los modos de servicio T1 y T2 la protección del operario no se encuentra activa.

## 2.1.2 KRC Software

Después de conectar el sistema de robot con el interruptor principal en el armario de control, el ordenador comienza a poner en marcha (cargar) el sistema operativo y el software de control.



Figura 2.3 KRC Software

Una vez que iniciamos en ordenador el display muestra el menú básico que permite crear, seleccionar y ejecutar programas (figura 2.3). El manejo del software es fácil de usar debido a que es usado en plataforma Windows y el sistema de ventanas es muy similar al de una pc normal.

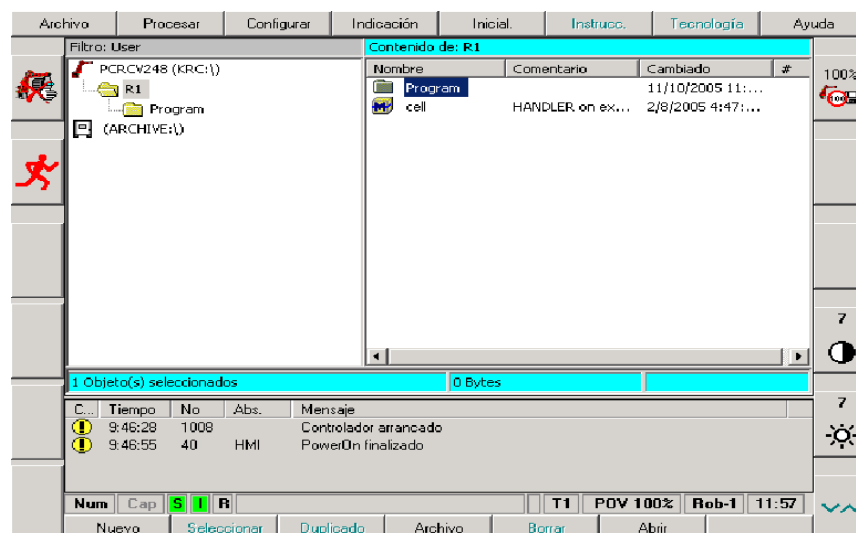


Figura 2.4 Entorno KRC Software

Si los datos de máquina cargados no corresponden con el tipo de robot, durante el arranque de la unidad de control se emite el correspondiente mensaje de fallo. En este caso, no se debe poner el robot movimiento.

### 2.1.3 Panel de Control KUKA (KCP)

El “KUKA Control Panel” (en adelante, “KCP”) es la interfaz entre la persona y la máquina, y permite manejar con facilidad la unidad de control del robot KRC. Todos los elementos de programación y manejo del sistema de robot, con excepción del interruptor principal, se encuentran directamente ubicados en el KCP. El display de cristal líquido visualiza las operaciones de manejo y programación.



*Figura 2.5 Panel de control (KCP)*

#### *Pulsador parada de emergencia*



Este pulsador rojo, accionado por golpe, se utiliza en casos de peligro. Provoca una detección inmediata de los accionamientos del robot.

Para volver a conectar los accionamientos, es preciso desenclavar el pulsador, no sin antes corregir la causa que origino la parada, y en su caso, el efecto que dicha parada ha provocado.

## Accionamientos CON



Accionando este pulsador, se conectan los accionamientos del robot. Estos accionamientos solamente se pueden conectar en condiciones de servicio normales.

## Accionamientos DESCON



Accionando este pulsador, se desconectan los accionamientos del robot. Al hacerlo también se activan, con un breve retardo, los ejes en su posición. La función “Accionamientos desconectados” provoca una parada por efecto generador.

## Selector de modos de servicios



Este interruptor de llave permite conmutar entre diferentes modos de servicio. La posición actual del selector de modos de servicio aparece indicada en la línea de direcciones. Los modos de funcionamiento establecidos son: Automático, Automático Externo o Controlado por PLC, Manual (T1) y (T2).



## Teclas de arranque y parada

Mediante la tecla de arranque (+) se inicia la ejecución de un programa hacia adelante, en los modos de servicio automático; siempre y cuando los accionamientos estén conectados.

En los modos T1 o T2 se ejecuta el programa hacia delante paso; siempre y cuando se mantengan pulsada la tecla a la vez que el hombre muerto, mientras



que el arranque (-) procesa la ejecución pasa a paso del programa en sentido contrario.

La tecla de Stop o parada del programa, detiene el puntero de programa en modo automático, provocando una parada sobre la trayectoria.

### *Teclas de estado y de función*



Seleccionaremos diferentes opciones dependiendo del modo de servicio en el que nos encontremos, para conmutar funciones y definir valores. Las teclas de función variarán su significado en función de lo que se nos vaya mostrando en pantallas y las teclas de menú correspondiente.

## **2.2 Sistema de Coordenadas**

Durante el desplazamiento del robot los movimientos se realizan con respecto a un sistema de coordenadas.

### **2.2.1 Especifico de ejes**

En el sistema de coordenadas específico de los ejes que se muestra en la figura 2.7, se puede desplazar individualmente cada uno de los ejes del robot en la dirección axial positiva o negativa. Este tipo de sistema se usa generalmente para movimientos largos y no tan exactos.



*Figura 2. 5 Sistema específico de ejes*

## 2.2.2 WORLD

El sistema de coordenadas WORLD (coordenadas universales) es un sistema de coordenadas cartesiano situado en un punto fijo. Funciona para movimientos más exactos como la calibración del *gripper* a diferentes sistemas. Además de servir como sistema de coordenadas de origen para los sistemas de coordenadas BASE y TOOL el cual se define más adelante.



Figura 2. 6 Sistema de coordenadas WORLD

## 2.2.3 BASE

El sistema de coordenadas BASE es un sistema de coordenadas cartesiano. Este sistema se mide de tal modo que su origen se encuentra en una pieza o en un dispositivo.

El sistema de coordenadas BASE está referido al sistema de coordenadas WORLD.

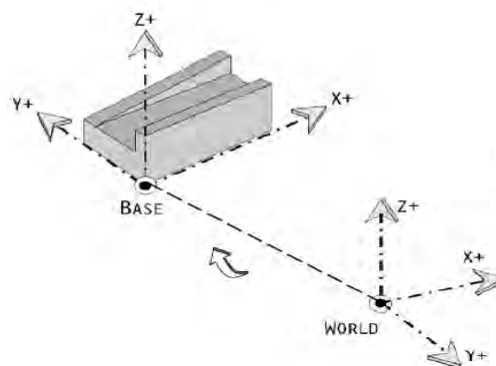
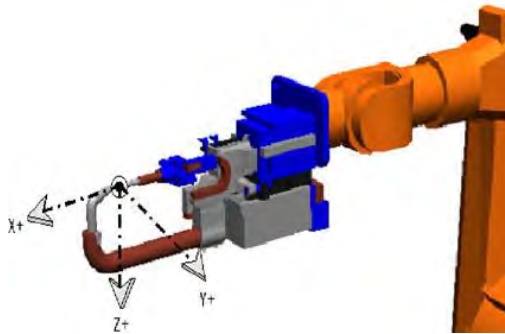


Figura 2.7 Coordenadas base referidas a WORLD

## 2.2.4 TOOL

El sistema de coordenadas TOOL es un sistema de coordenadas cartesiano. Este sistema se mide de tal manera que su origen está situado en la herramienta que está usando el robot.

Por lo general, la orientación del sistema de coordenadas TOOL se selecciona de modo que su eje X coincida exactamente con la dirección de trabajo de la herramienta (dirección de avance).



*Figura 2.8 Ejemplo coordenadas TOOL con una servopinza*

## 2.3 Desplazamiento manual

### 2.3.1 Generalidades

La función de desplazamiento manual permite mover el robot manualmente. El robot estándar dispone de un total de 6 ejes como se mostró en la figura 1.7

Para desplazar el robot manualmente, el selector de modos de servicio debe estar en la posición “Servicio en régimen discontinuo” (T1 o T2).

### 2.3.2 Tipo de desplazamiento

Para desplazar manualmente el robot, se puede elegir entre los siguientes tipos de desplazamiento:

#### Desplazamiento manual desconectado

Solamente para ejecutar un programa o para “automático”



#### Desplazamiento con el “Space-Mouse”

Desplazamiento simultáneo de 3 o 6 ejes, dependiendo del número de grados de libertad



#### Desplazamiento con las teclas de desplazamiento

Para poder mover cada uno de los ejes por separado



### 2.3.3 Desplazamiento con Space-Mouse

Si con la tecla de estado “Tipo de desplazamiento” ha seleccionado el Space-Mouse como elemento de mando, se puede mover el robot simultáneamente en 3 o en 6 ejes (grados de libertad). En este caso, se asigna al Space-Mouse un sistema de coordenadas que se mantiene igual en todos los sistemas de coordenadas de referencia.



Figura 2. 9 Mouse-Space

### 2.3.4 Desplazamiento con teclas de desplazamiento

Si con la tecla de estado “Tipo de desplazamiento”, ha elegido las teclas de desplazamiento como instrumento para introducir datos y ha seleccionado el sistema de coordenadas de referencia deseado, podrá mover el robot accionando las teclas de estado “+/-”.

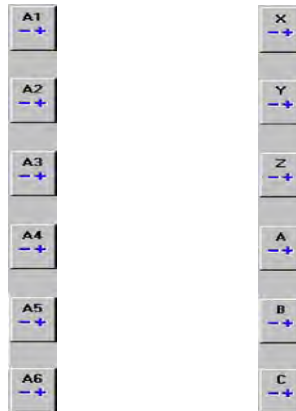


Figura 2. 10 Barra de teclas de estado

Dependiendo del sistema de coordenadas elegido la barra de estado muestra diferentes denominaciones. Cuando se elige el sistema de coordenadas específico de los ejes, la barra de teclas de estado situada a la derecha indica los ejes principales y los ejes de la muñeca A1 a A6 en cuanto el usuario acciona uno de los pulsadores de hombre muerto situados en la parte trasera del KCP. Pulsando la tecla de estado +/- correspondiente, se mueve el eje asignado.

Si se encuentra en el sistema de coordenadas “TOOL”, “BASE” o “WORLD”, el panel de control indicará los ejes principales X, Y y Z, así como los ejes de muñeca A, B y C. Por lo general, al desplazar el robot se mueven sincrónicamente varios ejes.

## 2.4 Programación

El movimiento del robot controlado por un programa, requiere de instrucciones de movimiento. Las instrucciones contienen información sobre el tipo de movimiento y su velocidad, la definición del punto de arranque y de destino (en recorridos circulares, también de un punto auxiliar), así como de otros ajustes que, a su vez, dependen del tipo de movimiento.

### 2.4.1 Movimiento Punto a Punto (PTP)

El posicionamiento del sistema de robot se produce aquí por el recorrido más corto entre dos puntos. Debido a que el movimiento en todos los ejes comienza y finaliza al mismo tiempo, los ejes deben ser sincronizados. Por este motivo, la trayectoria del robot no puede predecirse con exactitud.

#### *Movimiento PTP con parada exacta*

En los movimientos PTP con parada exacta, el robot se desplaza con exactitud hasta cada punto de destino como se observa en la figura 2.13

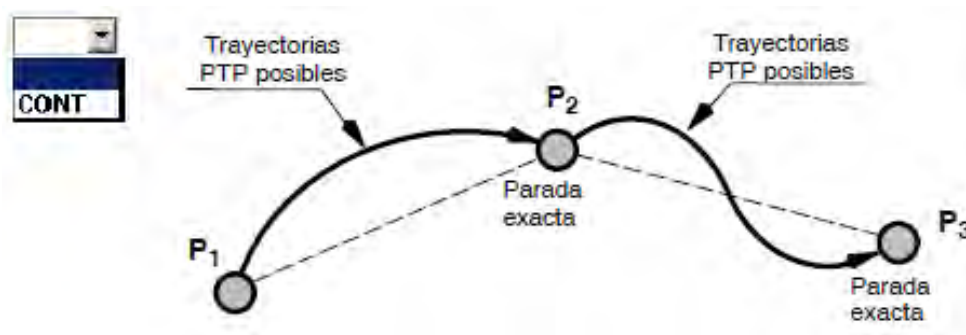
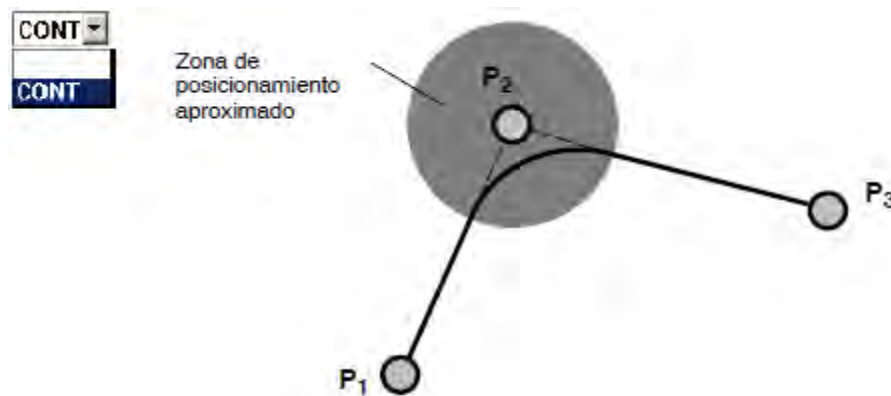


Figura 2.11 Movimiento PTP parada exacta

## *Movimiento PTP con posicionamiento aproximado*

En el movimiento con posicionamiento aproximado, la unidad de control supervisa una zona de aproximación alrededor del punto de destino. En la figura 2.14, este es el punto P2. Cuando el punto de referencia de la herramienta se introduce en esta zona de aproximación, el movimiento del robot pasa al punto de destino de la instrucción de movimiento siguiente.



*Figura 2.12 Movimiento PTP sin llegar a tocar el punto intermedio programado*

### **2.4.2 Movimiento lineal (LIN)**

En un movimiento lineal los ejes del robot se coordinan entre sí, de tal manera que el punto de referencia de la herramienta o de la pieza, se mueva a lo largo de una recta hacia el punto de destino. Los movimientos lineales se utilizan cuando se requiere para la aproximación de un punto un guiado de recorrido exacto con velocidad predeterminada.

#### *Movimientos LIN con parada exacta*

En movimientos LIN con parada exacta, el robot se desplaza a cada uno de los puntos de destino con posicionamiento exacto.

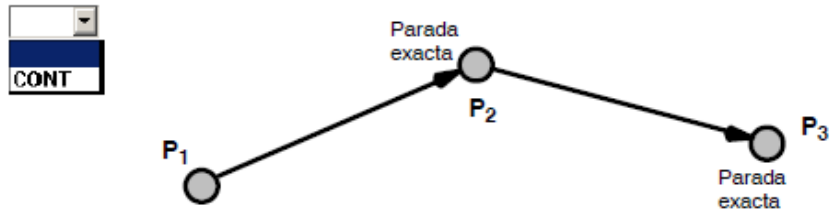


Figura 2.13 Movimiento LIN con parada exacta

### Movimientos LIN con posicionamiento aproximado

En el movimiento con posicionamiento aproximado, la unidad de control supervisa una zona de aproximación alrededor del punto de destino. En el ejemplo de la figura 2.16, este es el punto P2. Cuando el punto de referencia de la herramienta se introduce en esta zona de aproximación, el movimiento del robot pasa al punto de destino de la instrucción de movimiento siguiente.

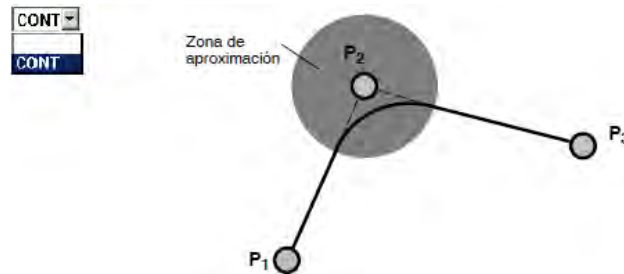


Figura 2.14 Movimiento LIN aproximado

### 2.4.3 Movimiento Circular (CIRC)

Aquí se mueve el punto de referencia de la herramienta o de la pieza sobre un arco circular al punto de destino. El recorrido se describe mediante puntos de arranque, auxiliar y final. Como punto de arranque vale en este caso el punto de destino de la parada exacta de la instrucción del movimiento anterior. La orientación se modifica uniformemente sobre el recorrido completo.

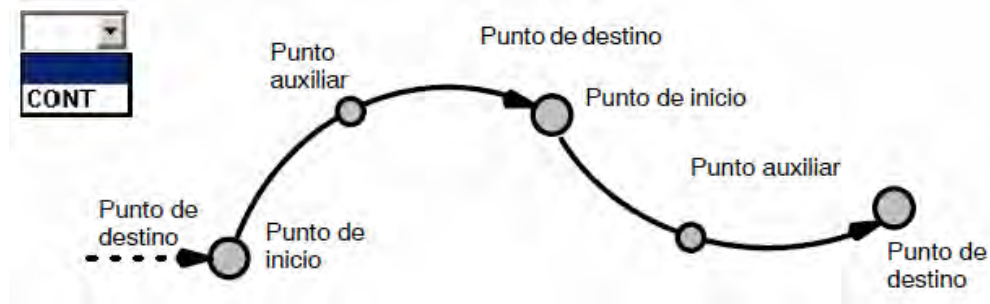
Movimientos CIRC se usan cuando procedimientos de procesamiento deben efectuarse sobre un trayecto circular con velocidad predeterminada. El punto de



arranque, auxiliar y final, se encuentran sobre un plano en el espacio. Para que la unidad de control pueda determinar, en lo posible, con exactitud ese plano, estos 3 puntos deberían en lo posible estar ampliamente distanciados entre sí.

### *Movimientos CIRC con parada exacta*

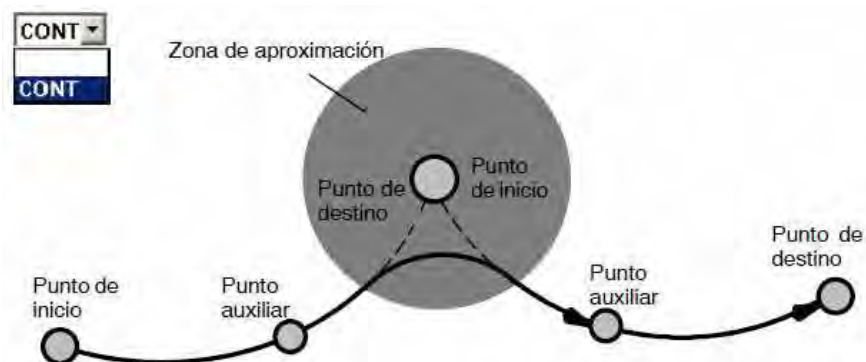
En movimientos CIRC con parada exacta, el robot se desplaza a cada uno de los puntos destino con posicionamiento exacto.



*Figura 2.15 Movimiento CIRC con parada exacta*

### *Movimientos CIRC con posicionamiento aproximado*

En el movimiento CIRC con posicionamiento aproximado, sucede lo mismo que en caso de LIN y PTP aproximado, de modo que el movimiento del robot pasa al punto de destino de la instrucción siguiente.



*Figura 2.16 Movimiento CIRC aproximado*

## 2.4.4 Funciones lógicas

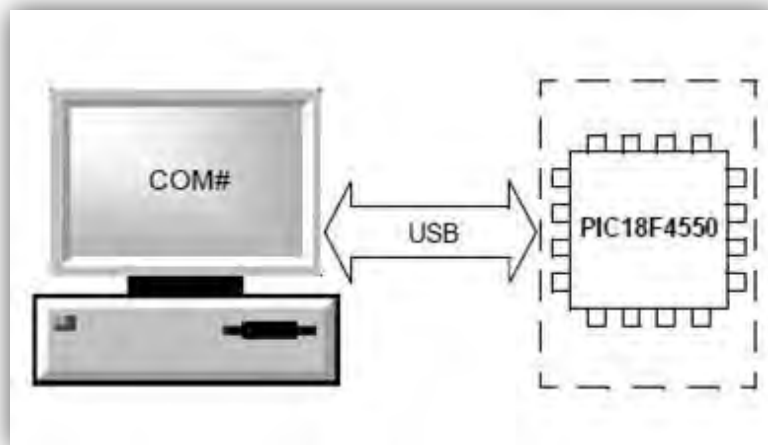
Las entradas y salidas de la unidad de control sirven para la comunicación del sistema de robot con los equipos periféricos. Para consultar entradas y activar salidas se dispone de instrucciones especiales.

Además, pueden ser programados tiempos de espera, para asegurar que los procedimientos hayan sido ejecutados hasta su finalización antes que el robot prosiga con su movimiento.

- a) **Función de espera (WAIT).** Se refiere a espera (retardos) dependientes del tiempo, con valores de paso de 0.05 seg.
  
- b) **Función WAITFOR.** Es una función de espera dependiente del estado de señales, las señales pueden ser de entrada o de salida.
  
- c) **Función de conmutación (OUT).** Esta función está integrada por varias opciones, como son SYNOUT para activar y desactivar salidas. SYN PULSE, envía un pulso simple a la salida. SYN OUT es una función de conmutación tiempo-distancia, dependiente de la trayectoria. SYN PULSE, función de impulso dependiente de la trayectoria.

## Capítulo 3. Comunicación Robot – PC

Para que el robot pueda establecer comunicación con el sensor encargado de detectar la forma y posición se usa el puerto USB incluido en una computadora, pero cuando se le conecta a ella misma el módulo de comunicación (microcontrolador y relevadores para acoplar), que se explicara más adelante, este puerto es detectado (con la instalación previa de un driver específico para el sistema operativo) como un puerto serie virtual, el cual a su vez es controlado por el programa de reconocimiento de forma realizado en MATLAB al cual se le programa el algoritmo para enviar los datos como si se tratara de un puerto serial.



*Figura 3.1 Conexión computadora- microcontrolador*

### 3.1 Puerto Serial

El puerto serial RS-232 también conocido como puerto COM, ya no es tan común en las nuevas generaciones de computadoras personales (PC). Debido a esto surgen grandes problemas para la comunicación con dispositivos externos y diversas aplicaciones desarrolladas y configuradas para este tipo de puerto. La solución está en emigrar la aplicación hacia el puerto USB.

Hay muchas diferentes maneras de convertir de puerto RS-232 a puerto USB, la más simple de ellas es la de emular un puerto serial sobre el bus de USB. Una ventaja de este método es que la PC reconocerá la conexión USB como un puerto RS-232 COM y por eso no se requieren cambios en la aplicación ya establecida.

Otra ventaja que se puede mencionar de este método es que utiliza un driver de Windows incluido en las versiones de Windows 98 y versiones posteriores hasta llegar al Windows vista en el cual se tiene que descargar el driver especial para esta versión, como es el caso de este proyecto, haciendo innecesario el desarrollo de alguno de ellos.

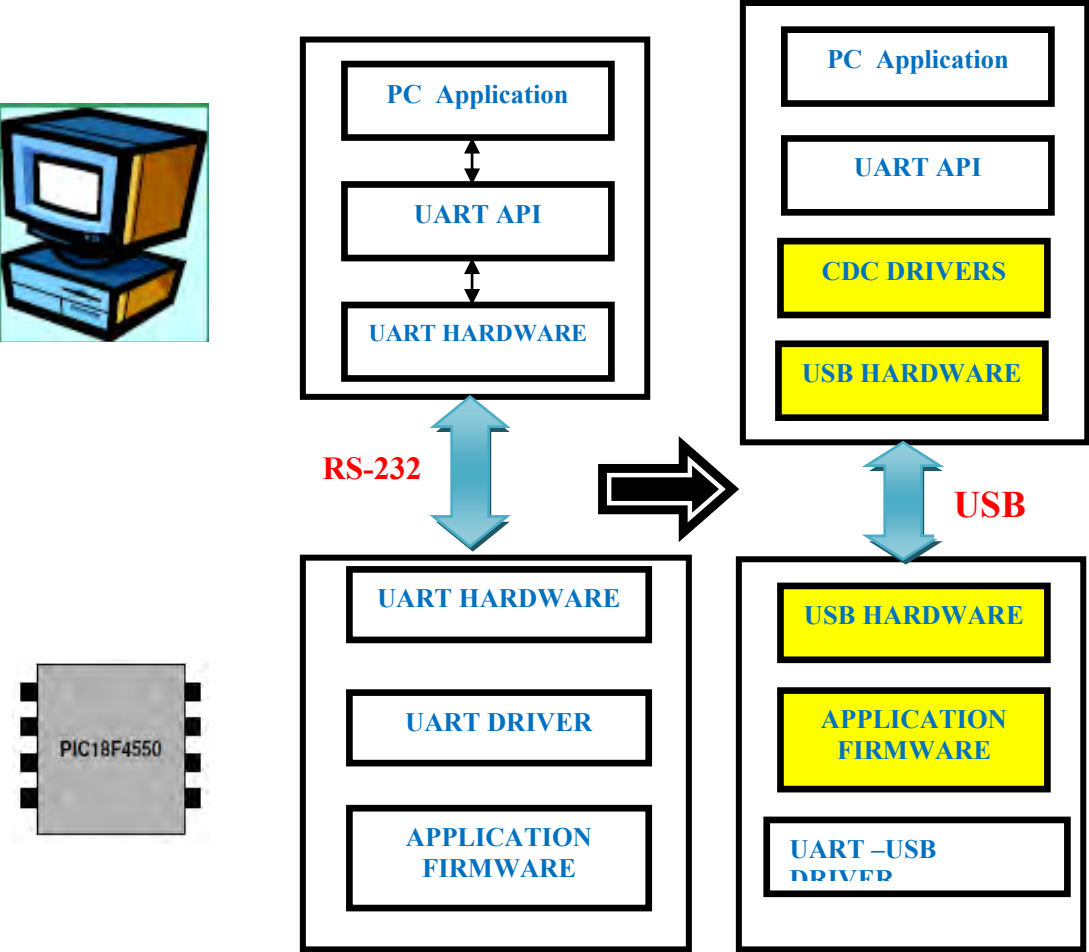


Figura 3.2 Cambios necesario para emigrar una aplicación de puerto RS-232 a USB

### 3.1.1 Puerto serial virtual

La clase del dispositivo de comunicación (CDC Communication Device Class) define muchos modelos de comunicación incluyendo la emulación del puerto serial. El driver de Windows `usbser.sys` forma parte de dicha especificación. Por lo que “Embedded device” (modulo de comunicación) debe ser asignado en el driver instalado para que a la hora de la conexión pueda ser detectado y visualizado en el panel de control Windows como en la Figura 3.3

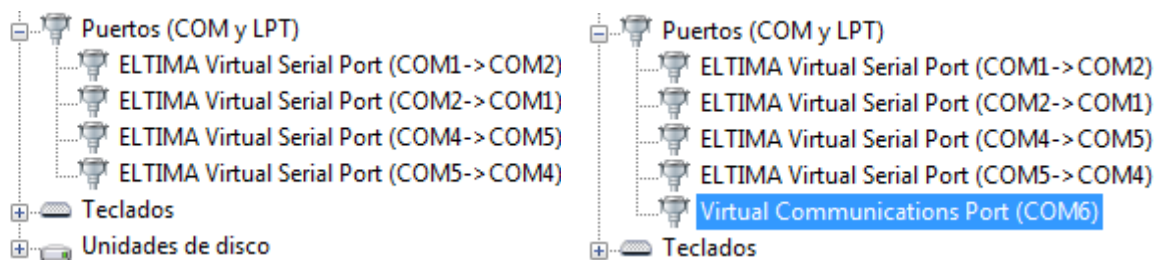


Figura 3.3 Visualización de la detección del dispositivo externo con un puerto serial virtual asignado por el sistema operativo

### 3.1.2 USB CDC

La especificación CDC requiere 2 interfaces. La primera es la interface de la clase de comunicación (Communication Class Interface), la cual utiliza un Endpoint de interrupción IN. Esta interface es usada para notificar al USB host de el estado de la conexión actual de RS-232 de el dispositivo emulado USB RS-232. La segunda es la interfaz de la clase de datos(Data Class Interface),en la que se emplea un “OUT BULK Endpoint” y un “IN BULK Endpoint “ la cual es usada para transferir los bytes de datos RAW que serian normalmente transferidos a través de el puerto serial RS-232 real.

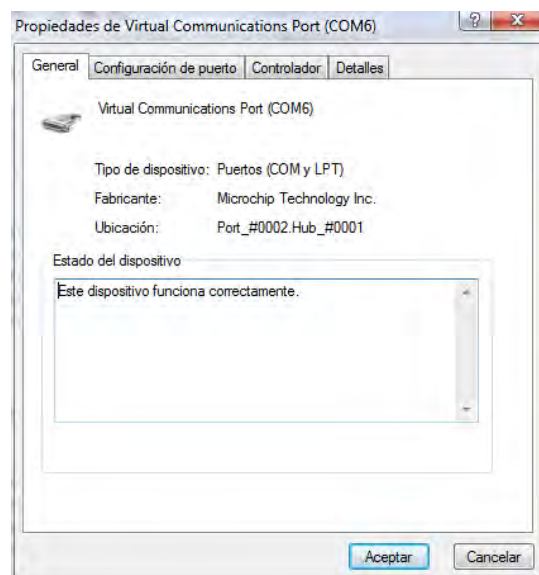
#### *Firmware USB CDC*

El firmware es un término comúnmente utilizado para referirse a los servicios fijos, por lo general pequeños programas y estructuras de datos para el control interno

de varios dispositivos electrónicos. Algunos ejemplos típicos de dispositivos que contengan firmware son los productos de usuario final, tales como calculadoras, PLC's, microcontroladores así como partes de computadoras y dispositivos como discos duros, tarjetas de memoria, todo enfocado hacia la instrumentación científica y robótica industrial. También los dispositivos más complejos, tales como teléfonos móviles, cámaras digitales, sintetizadores, contienen firmware.

El driver para esta aplicación (Archivo con extensión .INF) es el que habilita aplicaciones de la PC para que se pueda comunicar con el dispositivo externo que en este caso es el microcontrolador PIC18F2550. Cada dispositivo sobre el bus debe tener un driver, algunos periféricos utilizan los drivers que trae Windows pero para este caso en particular se instaló el driver específico para la comunicación USB CDC que fuera compatible con el sistema operativo Windows Vista Home Premium.

Cuando se instala correctamente el driver compatible se puede visualizar en el administrador de dispositivos (Figura 3.4) el número de puerto virtual asignado (que en este caso es el COM 6) el cual puede variar según el lugar donde se conecte la interfaz, además de mostrar si la conexión está funcionando correctamente, así como presentar la dirección asignada por el HUB en el bus.



*Figura 3.4 Ventana de Windows que muestra el estado de la conexión*

## 3.2 USB – KUKA KR16

Cuando la extensión USB se inserta en la PC se activa un voltaje de 5v en la interfaz y el programa empieza a funcionar en un modo en el cual emite un sonido intermitente desde uno de los relevadores, siguiendo así hasta que la interfaz de usuario (desde la PC en el botón seleccionar puerto COM Virtual) se sincroniza con la interfaz USB. Teniendo en cuenta que todos los circuitos de control de esta interfaz USB-KUKA KR16 se alimentan del voltaje suministrado por el puerto USB debido a que es solamente un circuito de control y acoplamientos por lo que requiere menos de 100 mA de corriente (esto es la cantidad máxima de consumo de corriente permitido en un solo puerto USB). La alternativa si se requiere utilizar circuitos externos es utilizar un regulador de voltaje estándar, como puede ser el LM7805, el cual proporciona +5 V a 1000mA.

Para poder sincronizar la interfaz USB con el programa de reconocimiento se debe insertar el número de puerto virtual (observar figura 2.6) que se asigna automáticamente por el sistema operativo cuando se conecta dicha tarjeta, en el botón con el nombre “Puerto COM Virtual” se le inserta el numero mencionado de puerto para así poder establecer la comunicación entre la PC y la tarjeta.

### 3.2.1 Conector USB

El USB (Bus de serie universal), como su nombre lo dice, se basa en una arquitectura de tipo serial. Sin embargo, es una interfaz de entrada/salida mucho más rápida que los puertos seriales estándar. La arquitectura serial se utilizó para este tipo de puerto por dos razones principales:

- La arquitectura serial le brinda al usuario una velocidad de reloj mucho más alta que la interfaz paralela debido a que este tipo de interfaz no admite frecuencias demasiado altas (en la arquitectura de alta velocidad,

los bits que circulan por cada hilo llegan con retraso y esto produce errores);

- Los cables seriales resultan mucho más económicos que los cables paralelos.

Una característica de la arquitectura USB es que puede proporcionar fuente de alimentación a los dispositivos con los que se conecta, con un límite máximo de 5 V por dispositivo. Para poder hacerlo, utiliza un cable que consta de cuatro hilos (la conexión a tierra GND, la alimentación del BUS y dos hilos de datos llamados D- y D+).

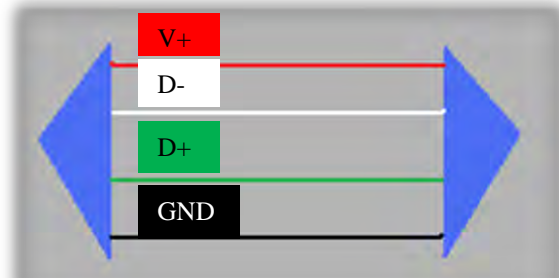


Figura 3.5 Hilos de conexión

Los puertos USB admiten dispositivos Plug and play de conexión en caliente. Por lo tanto, los dispositivos pueden conectarse sin apagar el equipo (conexión en caliente). Cuando un dispositivo está conectado al host, detecta cuando se está agregando un nuevo elemento gracias a un cambio de tensión entre los hilos D+ y D-. En ese momento, el equipo envía una señal de inicialización al dispositivo durante 10 ms para después suministrarle la corriente eléctrica mediante los hilos GND y VBUS (hasta 100 mA). A continuación, se le suministra corriente eléctrica al dispositivo y temporalmente se apodera de la dirección predeterminada (dirección 0). La siguiente etapa consiste en brindarle la dirección definitiva (éste es el procedimiento de lista). Para hacerlo, el equipo interroga a los dispositivos ya conectados para poder conocer sus direcciones y asigna una nueva, que lo identifica por retorno. Una vez que cuenta con todos los requisitos necesarios, el host puede cargar el driver adecuado.



El conector USB usado para este proyecto es un conector estándar de "tipo A" al cual llega la extensión de USB proveniente del puerto de la PC. Como se mencionó antes existen cuatro hilos de conexión de un cable USB, donde dos de ellos proviene la fuente de alimentación de la placa, mientras que los otros dos son las líneas de comunicaciones D + y D-. De esta manera es como se transfiere la información entre la PC y el PIC cuando la PC envía los códigos de detección.

### **3.2.2 Micro controlador**

El PIC usado es el PIC18F2550 este es un microcontrolador programable con 32Kbytes de memoria de programa y una RAM 2kBytes de propósito general. Cuenta con 10 canales A / D y 3 puertos cada uno con 8 entradas o salidas según se configure. Además de contar con sus terminales específicas para la conexión de los cuatro hilos desde el puerto USB.

### **3.3 Entradas/ salidas**

La electrónica digital utiliza magnitudes con valores discretos, mientras que la electrónica analógica emplea magnitudes con valores continuos.

Un sistema digital es cualquier dispositivo destinado a la generación, transmisión, procesamiento o almacenamiento de señales digitales.

Los sistemas digitales pueden ser de dos tipos:

- Sistemas digitales combinacionales: Son aquellos en los que la salida del sistema sólo depende de la entrada presente. Por lo tanto, no necesita módulos de memoria, ya que la salida no depende de entradas previas.
- Sistemas digitales secuenciales: La salida depende de la entrada actual y de las entradas anteriores. Esta clase de sistemas necesitan elementos de memoria que recojan la información de la ' historia pasada' del sistema.

### 3.3.1 Entradas digitales

Los módulos de entrada digitales permiten conectar a un robot, los sensores que envían señales digitales, como sensores de fin de carrera, presencia, contacto, etc. Los módulos de entrada digitales trabajan con señales de voltaje, por ejemplo cuando por una entrada llegan 24 volts se interpreta como un "1" y cuando llegan cero volts se interpreta como un "0".

Las entradas digitales de +24v del modulo de E/S WAGO son encargadas cada una de ellas de captar la señal enviada por la interfaz USB-KUKA la cual activa una combinación de ellas a la vez dependiendo del tipo de figura que sea detectada.

El modulo cuenta con 12 entradas (amarillas) de las cuales se utilizan se utilizan 8 para realizar las combinaciones en binario obteniendo con esto 252 posiciones posibles que puede tener el objeto. Y las tres restantes se utilizan para determinar la forma de la pieza.

Para poder enviar los datos de que figura fue detectada y en qué posición se encuentra, después de realizar el procesamiento por medio del programa de reconocimiento hacia la interfaz USB para que posteriormente esta misma tarjeta pueda enviarlos hacia el Robot, se presiona el botón "Enviar" el cual activa la función que activa las señales que serán el identificador de cada figura y posición programado en MATLAB para manejarlos de forma serial con la instrucción "fwrite" pero siendo enviados por el puerto USB.

### **3.3.2 Salidas digitales**

El modulo también cuenta con 12 salidas digitales (rojas) de las cuales solo se ocupan dos para controlar la apertura o cierre del *gripper*. En sí, para este proyecto las salidas no afectan mucho, puesto me el robot toma las decisiones en base a las entradas activadas y las salidas únicamente se usan para controlar las acciones del robot.

## Capítulo 4. Reconocimiento de Piezas

### 4.1 Sistema de visión

El sistema que permite la captura y/o adquisición de la imagen, está formado por los elementos que se muestran en la figura 4.1

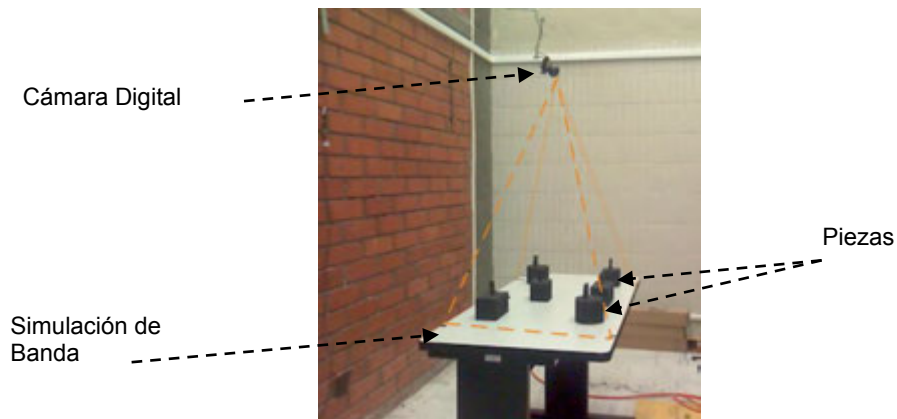


Figura 4.1: Sistema de visión

#### 4.1.1 Cámara digital

Es el dispositivo encargado de transformar las señales luminosas que aparecen en la escena, en señales digitales capaces de ser transmitidas por un cable serial. Se divide en dos partes, el sensor, que captura las propiedades del objeto en forma de señales luminosas y lo transforma en señales digitales, y la óptica que se encarga de proyectar los elementos presentes en la escena a la superficie del sensor. En la figura 4.2 se muestra una imagen capturada en la que los parámetros de contraste y brillo fueron ajustados para obtener una imagen que sea más fácil de reconocer.



*Figura 4.2 Imagen con alto contraste e Iluminación frontal*

Una vez que se está ejecutando el programa se selecciona la cámara que se va a utilizar para adquirir la imagen, primero se presiona el botón “seleccionar cámara” (Figura 4.3 ), posteriormente se activa una nueva ventana en la que se podrá seleccionar la cámara a usar, en la que la resolución programada para trabajar es 320x240 debido a que disminuye el tiempo de procesamiento y una resolución alta no es tan necesaria para esta aplicación debido a que el enfoque está en la forma de los objetos a partir del contraste y no en sus detalles .



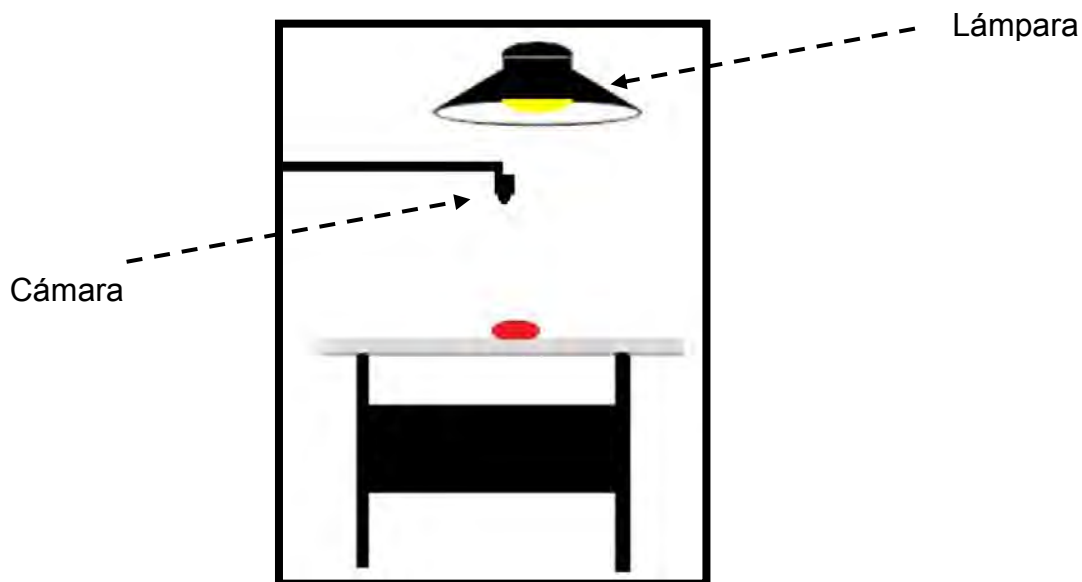
*Figura 4.3 Selección de cámara*

### **4.1.2 Iluminación**

La iluminación de la escena es un aspecto decisivo en el desarrollo de un sistema visual. Antes de intentar corregir un problema de iluminación por medio de algoritmos muy complicados, es mejor prestar atención e implantar un sistema de

iluminación adecuado, para que la captura de la imagen sea correcta. Es mejor un buen sistema de iluminación, que intentar corregir ese problema por software, pues la velocidad de procesamiento será mayor con algoritmos más sencillos. Por lo tanto, se menciona a continuación el sistema de iluminación usado para esta aplicación, exponiendo una breve descripción.

Es la más usada, y como se muestra en la figura 4.4 consiste en iluminar frontalmente la pieza. Presenta más problemas para obtener un buen contraste entre la pieza y el fondo debido a la aparición de brillos y sombras que alteran las propiedades de las piezas a estudio lo que se puede evitar seleccionando colores contrastantes de pieza y superficie, además de pintar las piezas con pintura poco reflectora para evitar los brillos.



*Figura 4.4 Sistema de iluminación*

## **4.2 Procesamiento de imágenes**

El procesamiento de imágenes tiene como objetivo mejorar el aspecto de una imagen o hacer más notoria alguna característica que nos interese. La imagen

puede ser obtenida fotográfica o electrónicamente. El procesamiento de la imagen se realiza utilizando las operaciones digitales con las cuales cuenta la computadora.

Para el procesamiento se toma la imagen y se divide en múltiples unidades llamadas píxeles y cada pixel tiene un valor específico.

### 4.2.1 Binarización

La mayor parte de los algoritmos para reconocer formas están escritos a partir de imágenes binarias, por lo que se hace conveniente el paso de una imagen en niveles de gris (o color) a una binaria, además esto permite reducir el volumen de los datos a tratar.

La binarización de una imagen digital consiste en convertir la imagen digital en una imagen en blanco y negro, de tal manera que se preserven las propiedades esenciales de la imagen, como se muestra en la figura 4.5



*Figura 4.5 Imagen binarizada*

Uno de los métodos para poder binarizar una imagen digital es mediante el histograma de dicha imagen. A través del histograma obtenemos una gráfica donde se muestran el número de píxeles por cada nivel de gris que aparecen en la imagen.

Todos los niveles de grises menores al umbral calculado se convertirán en negro y todos los mayores en blanco.

## **4.2.2 Operaciones morfológicas**

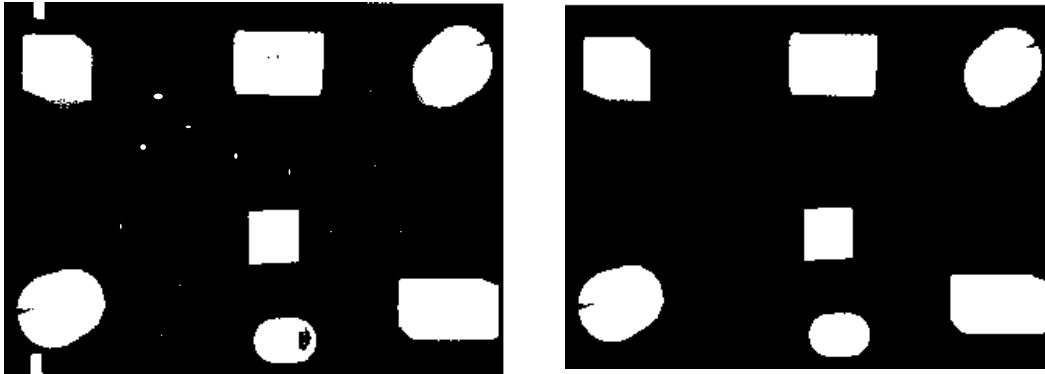
Las operaciones morfológicas pueden simplificar los datos de una imagen, preservar las características esenciales y eliminar aspectos irrelevantes. Teniendo en cuenta que la identificación y descomposición de objetos, la extracción de rasgos, la localización de defectos e incluso los defectos en líneas de ensamblaje están sumamente relacionados con las formas, es obvio el papel de la morfología matemática. La morfología matemática se puede usar, entre otros, con los siguientes objetivos:

- Preprocesamiento de imágenes (supresión de ruido, simplificación de formas).
- Destacar la estructura de objetos (extraer el esqueleto, marcado de objetos, envolvente convexa, ampliación, reducción).
- Descripción cualitativa de objetos (área, perímetro, diámetro, etc).

### *Eliminar áreas menores a 100 pixeles*

Este paso que se puede visualizar en la figura 4.6, se ejecuta con la intención de eliminar objetos intrascendentes que pueden interferir a la hora de ejecutar el algoritmo de detección, esto se hace también con la finalidad de que el objeto a detectar quede totalmente aislado y no exista ningún otro objeto que lo pueda que pueda deformar la imagen.

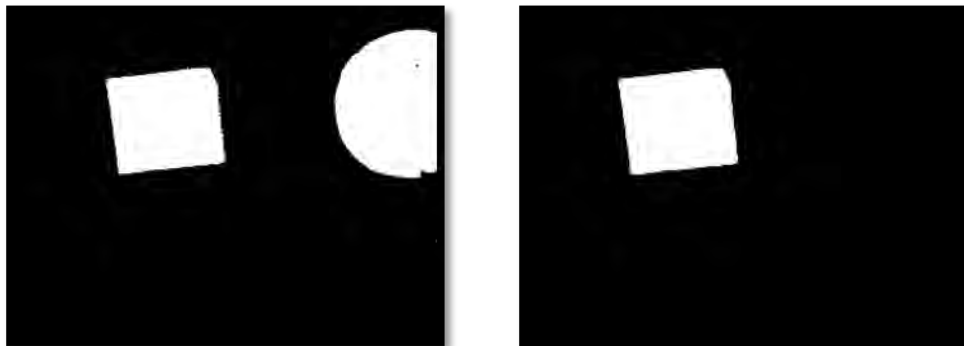




*Figura 4.6 Imagen con formas de menores de 100 pixeles sin eliminar y a la derecha dichas formas eliminadas*

### *Eliminar objetos incompletos*

Con esta operación se eliminan los objetos que se encuentran unidos al borde como triángulos, cuadrados, círculos y rectángulos incompletos, los cuales no tienen una forma definida que pudiera ser detectada para ser seleccionada como se puede observar en la figura 4.7



*Figura 4.7 Circulo incompleto eliminado*

## *Separar objetos*

El proceso que se muestra la figura 4.7 desarrolla una apertura morfológica, esto quiere decir que separa dos objetos que se encuentren unidos por unos cuantos píxeles, primero ejecuta una erosión seguida de una dilatación.



*Figura 4.8 Separación de imágenes unidas por algunos píxeles*

## *Suavizar bordes*

Para eliminar píxeles que se pueden visualizar como picos que se encuentren en el borde de las imágenes, se ejecuta una operación morfológica de cierre en la que primero se dilata la imagen y después se erosiona.

## *Rellenar agujeros*

Debido a que este tipo de reconocimiento se realiza tomando en cuenta el área de las figuras, se rellenan agujeros que se encuentran en el interior de las formas a detectar.



*Figura 4.9 Rellenado de agujeros*

### 4.2.3 Segmentación

La segmentación en el campo de la visión por computadora es el proceso de dividir una imagen digital en varias partes (grupos de píxeles) u objetos. El objetivo de la segmentación es simplificar y/o cambiar la representación de una imagen en otra más significativa y más fácil de analizar. La segmentación se usa tanto para localizar objetos de interés, como para encontrar los límites de estos dentro de una imagen (Figura 4.10). Más precisamente, la segmentación de la imagen es el proceso de asignación de una etiqueta a cada píxel de la imagen de forma que los píxeles que compartan la misma etiqueta también tendrán ciertas características visuales similares.



*Figura 4.10 Región de interés*

El resultado de la segmentación de una imagen es un conjunto de segmentos que cubren en conjunto a toda la imagen, o un conjunto de contornos extraídos de la imagen. Cada uno de los píxeles de una región son similares en algunas características, como el color, la intensidad o la textura. Regiones adyacentes son significativamente diferentes con respecto a la misma característica.

A través de propiedades básicas como: área, centroide, se puede definir una región dentro de una imagen. Dicho proceso fue realizado en Matlab.

El primer paso que se realiza es convertir dicha imagen a una representación a través de una matriz etiquetada (Labels) como se muestra en la figura 4.11. La

siguiente instrucción toma como elemento de entrada a esa matriz etiquetada y calcula las propiedades necesarias como longitud de ejes, área y centroide.

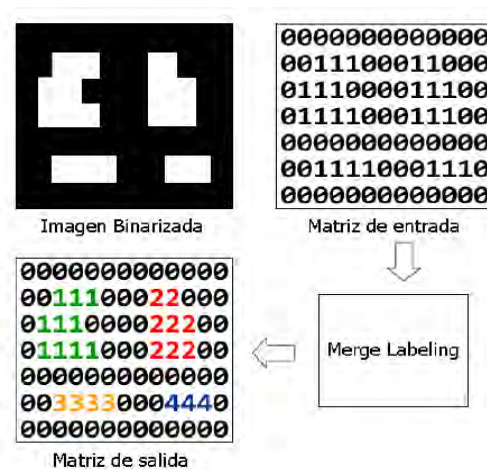


Figura 4.11 Matriz etiquetada

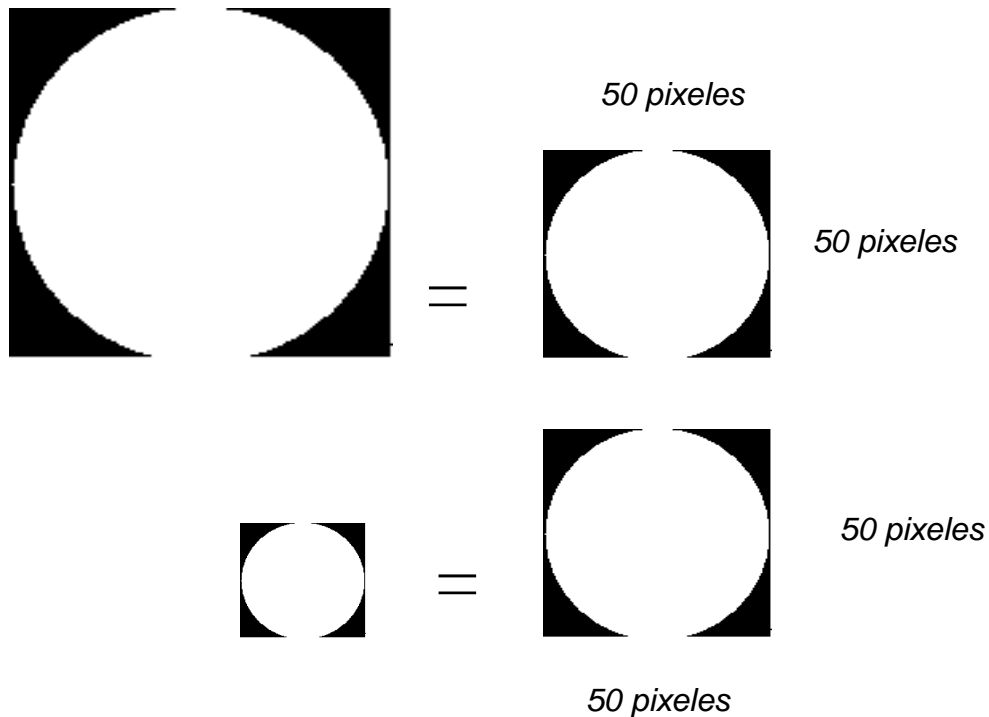
Si analizamos los resultados obtenidos podemos mencionar que el algoritmo de segmentación basado en etiquetación de píxeles propuesto resulta muy efectivo en condiciones en las cuales sabemos que los objetos a segmentar no están juntos en la escena.

### 4.3 Reconocimiento

Un método (para el caso del círculo) de los dos utilizados para reconocer las tres formas contempladas para este proyecto es tomar decisiones a partir de un área determinada, la base de la diferenciación radica en que una figura tendrá un área dentro de un rango de valores y la otra figura otro rango diferente a partir de una referencia que se establece redimensionando todas las imágenes segmentadas (para este caso a 50x50 píxeles). El otro método (para el caso del cuadrado y rectángulo) radica en conocer la diferencia entre longitudes de sus ejes por medio de la función *regionprops*.

### 4.3.1 Redimensionar imagen

Para poder establecer una referencia de un rango de valores de área al momento de clasificar las formas, se redimensionan todas las imágenes que fueron resultado de la segmentación a 50x50 píxeles como se muestra en la figura 4.12.



*Figura 4.12 Imágenes segmentadas redimensionadas*

### 4.3.2 Suma de píxeles

Después de redimensionar la imagen se suman, y se almacenan en una variable los píxeles que se encuentran en "1" y se descartan los píxeles en "0", determinando con esto el área total de la figura en píxeles, que después pasara a comparación para establecer de que figura se trata.

### 4.3.3 Determinación de forma

Se calcula el rango máximo en pixeles que podría tener el área de un círculo en las dimensiones de 50x50 como se muestra en la figura 4.13. Entonces el valor a comparar será el máximo posible.

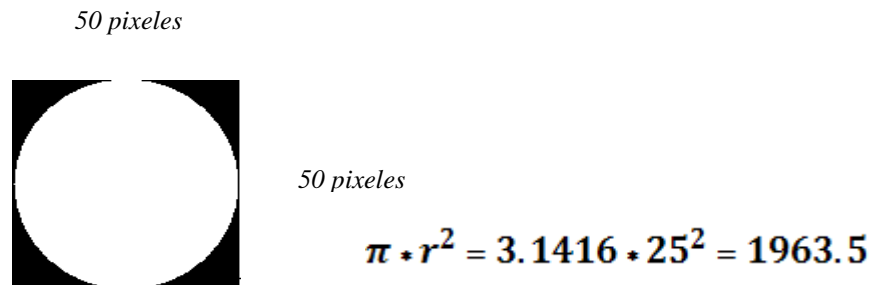


Figura 4.13 Calculo del valor máximo del área para el círculo

Si el área de la imagen es mayor a 1260 (umbral de 10 pixeles) y menor a 1963 entonces es un círculo.

Después de haber determinado los valores, si no concuerda con ninguno de ellos es que es un cuadrado o un rectángulo y pasa a la siguiente comparación mostrada en la figura 4.14, la cual hace diferencia entre las dos figuras mencionadas mediante el siguiente procedimiento:

- 1.- Se guarda un valor mínimo de diferencia, por ejemplo 15 pixeles.
- 2.- Se obtiene el valor en pixeles de eje mayor.
- 3.- Se obtiene el valor en pixeles de eje menor.
- 4.- Se restan el valor del eje mayor menos el eje menor.
- 5.- Si la diferencia entre eje mayor y eje menor nos es mas a 15 pixeles entonces se trata de un cuadrado, de lo contrario se trata de un rectángulo.

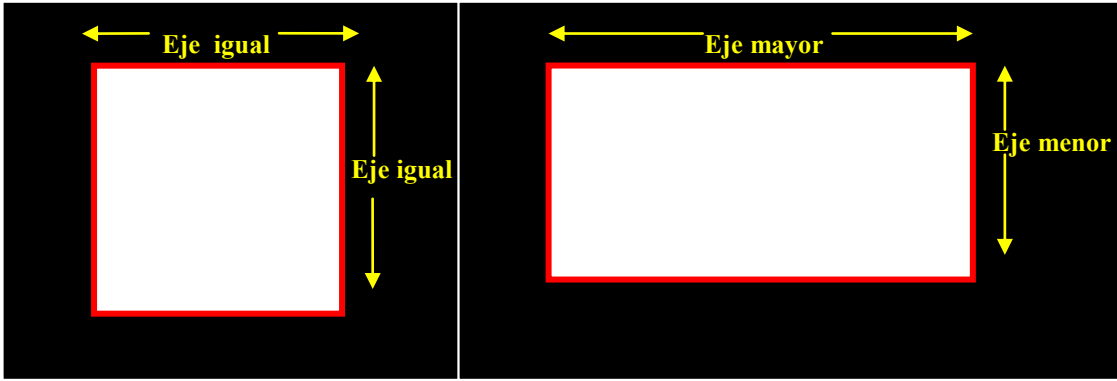


Figura 4.14 Ejes de Cuadrado y rectángulo

***Eje mayor – Eje menor  $\leq 15$ pixeles  $\therefore$  Es cuadrado***

***Eje mayor – Eje menor  $> 15$ pixeles  $\therefore$  Es Rectangulo***

#### 4.3.4 Determinación de posición

Para determinar la posición de la pieza, lo primero que se hizo fue tomar la imagen almacenada y dividir la imagen visible por la cámara en secciones más pequeñas a manera de malla (figura 4.15) y a cada cruce se le asigna un valor en binario de 8 dígitos, el cual coincide con las entradas utilizadas, de esta manera se tendrá un registro de las posibles combinaciones de posición en las cuales podrá estar el objeto, así, no importará en que parte de la banda se encuentra el objeto, el sistema asignará la coordenada más cercana con buena precisión (Figura 4.16).

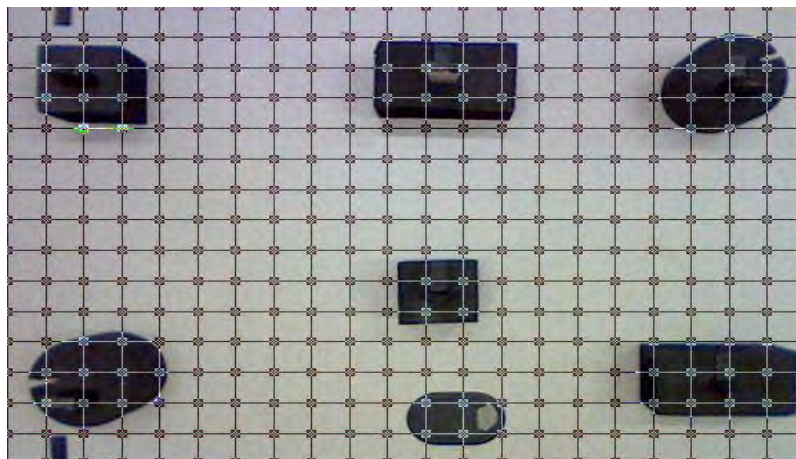
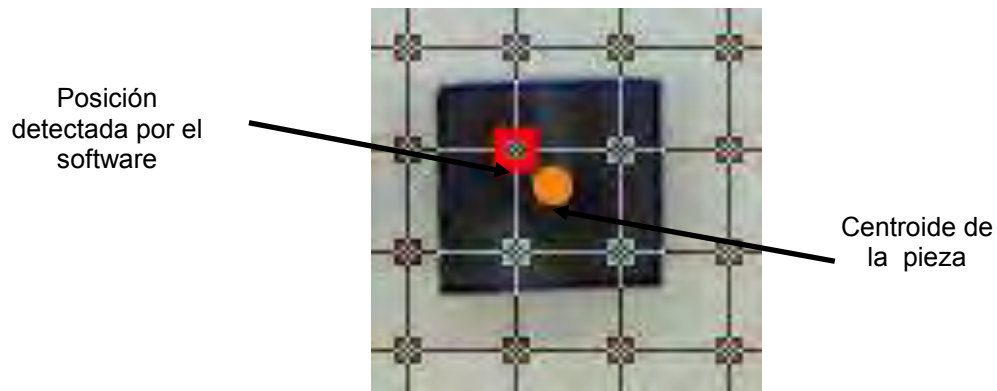


Figura 4.15 Imagen segmentada donde se muestran las posibles combinaciones de coordenadas.

Una vez que la figura ha sido identificada se calcula su centro de masa, este le servirá como referencia al sistema al momento de decidir cuál es el punto o nodo más cercano para poder tomar el objeto.

En el caso de que se encuentren distintos puntos cerca del centroide el sistema será capaz de determinar cuál es el más cercano de acuerdo a los criterios de selección.



*Figura 4.16 Extracción de la malla para identificación de posición*

Una vez que se han detectado la ubicación del objeto, se compara la coordenada con el valor de la combinación correspondiente, una vez que ya se tiene la coordenada se envían las señales al microcontrolador para que a su vez se el robot las reciba y en base a las entradas activadas pueda dirigirse a la posición programada.



## **Capitulo 5. Conclusiones**

### **Conclusión del proyecto**

Para el desarrollo de este proyecto se han hecho uso de distintas áreas de investigación tal y como lo son la óptica, la electrónica, la computación, entre otras, para que de este modo el sistema pueda ser más completo y eficiente.

En primer lugar fue necesario aprender las características básicas de robot KUKA KR-16 para poder manipularlo de acuerdo a lo requerido ya que si esto no se podría trabajar del lado del robot.

Se usaron las técnicas de programación conocidas, debido a que el software del robot tiene características muy similares a las de un entorno de programación de los ya utilizados.

Puesto que el sistema de detección debía ser realizado en MatLab, se aprendió a manejar el software, el modo de programación, el diseño de interfaces y una parte muy importante el uso de las funciones, ya que MatLab cuenta con librerías para el procesamiento de imágenes, y esto facilitó el desarrollo del sistema.

Se reforzaron los conocimientos de electrónica para poder desarrollar el modulo de conexión entre la computadora y el robot.

## Conclusión Personal

Al realizar este proyecto pude aprender a aterrizar los conocimientos adquiridos en las materias vistas, puesto que es muy diferente tener el conocimiento teórico que ya realizarlo en la práctica, además adquirí conocimientos de otras herramientas existentes para desarrollar el trabajo que se iba necesitando.

Por otro lado me ayudo a conocer las diferentes situaciones y problemas que se dan cuando se realiza un proyecto y que son cosas que no se enseñan teóricamente, sino que van surgiendo y se tiene que ir buscando la manera de resolverlos ya sea buscando nuevas herramientas o métodos. También me ayudo a organizar y cumplir con los tiempos, ya que se tuvieron que planear las actividades ya sea por semana o por mes y de este modo fue más fácil avanzar con el proyecto.

Los principales dificultades que se me presentaron fueron respecto a la electrónica y a la realización del circuito de comunicación entre la pc y el robot, debido a que es un área no muy vista en la carrera, y que en lo personal me resulta complicado, pero con un poco de investigación y dedicación se pudo realizar esa parte. Otro fue el área de la óptica para implementar el sistema de visión, para esto se me permitió tomar algunas clases en el centro de investigación para obtener los conocimientos requeridos.

Y un reto más fue la presentación del proyecto en la VIII participación de la mujer en la ciencia, debido a que se tuvieron que acelerar algunas partes del desarrollo para poder presentarlo, pero esto sirvió para ver la aceptación y el interés de la sociedad por el tema.

## **Bibliografía**

GARCÍA Higuera, Andrés; CASTILLO García, Fernando J.; CIM: el computador en la automatización de la producción; Universidad de Castilla La Mancha; 2007; 270 p.p.

GARCÍA Moreno, Emilio; Automatización de procesos industriales: robótica y automática; Universidad Politécnica Valencia; 1999; 377 p.p.

GOMEZ De Silva Garza, Andrés; ANIA Briseño, Ignacio de Jesús; Introducción a la computación; Cengage Learning, 2008, 522 p.p.

OLLERO Baturone, Aníbal; ROBÓTICA, Manipuladores y Robots móviles; Marcombo Boixareu Editores; 2001; 288 p.p.

[http://www.kuka-robotics.com/mexico/es/products/industrial\\_robots/low/kr16\\_2/start.htm](http://www.kuka-robotics.com/mexico/es/products/industrial_robots/low/kr16_2/start.htm);  
manual del robot kuka kr16.