



INSTITUTO POLITECNICO NACIONAL

Centro de Investigación en Computación

**Laboratorio de Reconocimiento de Patrones y
Microtecnología y Sistemas Embebidos**

**Tele-manipulación de objetos mediante un robot Khepera II
implementado en un circuito reconfigurable**

T E S I S

**Que para obtener el grado de
Maestro en Ciencias de la Computación**

P R E S E N T A :

Lic. Jaime Alfonso Pérez León

Directores de tesis:

Dr. Juan Humberto Sossa Azuela

Dr. Luis Alfonso Villa Vargas



México D.F., diciembre 2009

*A mis padres:
Norma León Quintero
y Jaime Pérez Rocha*

*A mis tíos y primos:
Ana María Jiménez,
Eleazar Pérez Rocha,
Paola, Edgar y Nancy*

A Nataly López Soto

Agradecimientos

A Dios, por darme la fuerza cuando ya no me quedaban.

A mis asesores, el Dr. Juan Humberto Sossa Azuela y Dr. Luis Alfonso Villa Vargas por darme la oportunidad de trabajar con ellos.

A mi comité por sus valiosas observaciones, Dr. Marco Antonio Ramírez Salinas, Dr. Victor Hugo Ponce Ponce, Dr. Herón Molina Lozano y especialmente al Dr. Marco Antonio Moreno Armendáriz; y a todas aquellas personas que directa o indirectamente intervinieron en el desarrollo de esta tesis.

Al Instituto Politécnico Nacional, al Centro de Investigación en Computación por permitirme continuar con mi formación académica.

Este documento se ha realizado con la ayuda financiera de la Comunidad Europea, la Unión Europea, la Comisión Europea y el CONACYT bajo el proyecto FONCICYT 93829.

El contenido de este documento es responsabilidad exclusiva del Instituto Politécnico Nacional y en modo alguno debe considerarse refleja la posición de la Unión Europea.

A todos los compañeros y amigos del Laboratorio de Reconocimiento de Patrones y Laboratorio de Micro tecnología y Sistemas Embebidos por sus aportaciones y desinteresada cooperación; en particular a Israel Román, Cecilia Albortante Morato, Liliana Sánchez Romero, Julio Jiménez Vielma, Laura Gómez Sánchez, Omar Huerta Moreno, Juan Villegas Cortez, Eduardo Lavín, Guadalupe Sáenz, Bryan Santos Pérez, Jeaninne Razo Gil, Roberto Vázquez, Raúl García, Itzamá López Yáñez, Edgar Catalán, Roberto Mercado, Vanesa González, Miguel A. Martínez.

A toda mi familia por haberme apoyado en el transcurso de estos años.

A los que de una manera u otra, intervinieron en mi formación...

A ti.

Resumen

En este trabajo, se llevó a cabo la manipulación de objetos usando un robot Khepera II y como interfaz un circuito reconfigurable, en este caso un FPGA.

Todo este sistema, fue hecho sobre un circuito reconfigurable, el cual, nos da una mayor respuesta y flexibilidad que una computadora personal, por lo que, es necesario configurarla completamente y por un lado, nos da mayor rapidez de respuesta en cuanto a tiempo de ejecución, pero por otro lado, nos da un mayor número de problemas, puesto que el circuito se tiene que configurar completamente desde cero, esto es, crear un controlador que sea capaz de enviar datos por el puerto serie, que sean reconocibles para el robot y también un decodificador para el uso del teclado por el puerto PS/2.

El teclado que se utilizó para este trabajo es un tipo de teclado estándar con conector PS/2 color purpura. Con éste, tenemos todo el control del robot y de sus pinzas, usando las teclas de arriba, abajo, izquierda y derecha, tenemos un movimiento del robot. Con la tecla V, se tiene un movimiento en reversa. Con la tecla A se tiene que el robot suba las pinzas, con la S que las baje, con la D que las cierre y con la F que las abra. Con estos comandos, ya tenemos un control total sobre todo lo que se podría realizar con el robot.

El puerto de comunicación serial es un conector DB9 que a su vez va conectado a un adaptador DB25 que lo trae el robot en su caja. Este cable tiene que ser cruzado para que pueda funcionar el envío de comandos al robot, puesto que el FPGA envía los datos por el pin 3 y el robot envía también por el 3, por lo tanto, hay que conectar la salida 3 del FPGA a la 2 del robot y la 3 del robot a la 2 del FPGA.

En la visión, tenemos el módulo de cámara montada sobre el robot y con ella, vemos el área de trabajo en la cual se está moviendo el robot. Esta cámara está conectada a un modulo inalámbrico de recepción de video. El espacio de trabajo lo observamos desde un monitor con entrada RCA.

Índice general

Índice de figuras	X
Glosario de términos	XII
1 Introducción	1
1.1 Los primeros robots móviles.....	2
1.2 Los inicios del FPGA.....	6
1.3 Justificación	8
1.4 Objetivo general.....	9
1.5 Objetivos particulares	10
1.6 Consideraciones del problema	11
1.7 Aportaciones	12
1.8 Organización de la tesis	12
2 Estado del arte	15
2.1 Manipulación robotizada de objetos basada en visión por computadora (2009).....	15
2.2 Generación de trayectorias para un robot móvil Khepera II usando técnicas de aprendizaje automático (2007)	17
2.3 Control visual de un robot móvil Khepera II (2007)	18
2.4 Control visual reactivo de múltiples agentes robóticos no holonómicos (1998).....	19
2.5 Soccer Teleworkbench for Development and Analysis of Robot Soccer.....	20
2.6 Desarrollo de un robot móvil con control inteligente usando algoritmos evolutivos y un fpga.....	22
2.7 Resumen	24
3 Marco teórico	25
3.1 Robots manipuladores	25
3.1.1 Articulaciones y grados de libertad	26
3.1.2 Espacio de trabajo	27
3.2 Robots móviles	29

3.2.1	El robot Khepera II	30
3.2.2	Programación de un robot	31
3.2.3	Movimientos del robot	32
3.3	Circuitos reconfigurables	33
3.3.1	FPGA.....	34
3.4	Control visual	39
3.4.1	Captor de imágenes	40
3.5	Controlador del robot manipulador.....	41
3.5.1	Tipo de teclado	41
3.5.2	Puerto serie RS-232	42
3.6	Resumen	45
4	Metodología propuesta	47
4.1	Condiciones iniciales	48
4.2	Etapa de adquisición	48
4.3	Etapa de detección.....	50
4.4	Etapa de control.....	50
4.4.1	Entrada de datos	50
4.4.2	Salida de datos.....	52
4.5	Resumen	59
5	Resultados experimentales	61
5.1	Experimento 1:Cubo rojo	61
5.2	Experimento 2: Agrupar objetos en una caja	62
5.3	Experimento 3: Poner objetos sobre una marca	65
5.4	Experimento 4: Cambiar de lugar cubo	67
5.5	Experimento 5: Intercambiar lugares.....	69
5.6	Resumen	73
6	Conclusiones y trabajos futuros	75
6.1	Conclusiones	75
6.2	Trabajos futuros.....	76
6.3	Conclusión final	77
	Referencias	79

Índice general	IX
A. Equipo empleado	83
B. Características del robot Khepera II	87
C. Características del FPGA Altera DE2	91

Índice de figuras

Figura 1.1. Diagrama de acción del sistema de tele-control a diseñar en esta tesis.....	10
Figura 2.1: Robot manipulador modelo CRS	16
Figura 2.2: Diagrama de soccer teleoperado	20
Figura 2.3: Robot Khepera II equipado con un módulo FPGA y una cámara en 2D a color	21
Figura 3.1: Manipulador cartesiano	27
Figura 3.2: Manipulador articulado	28
Figura 3.3: Manipulador tipo SCARA.....	28
Figura 3.4: Manipulador polar	29
Figura 3.5: Manipulador cilíndrico.....	29
Figura 3.6: Ángulo de movimiento del robot base	32
Figura 3.7: Angulo de movimiento de la pinza.....	33
Figura 3.8: Movimiento de la pinza del robot.	33
Figura 3.9: Arquitectura de un ASIC Actel.....	34
Figura 3.10: Arquitectura general de un FPGA.	36
Figura 3.11: Diagrama de bloques del FPGA arquitectura de Altera	40
Figura 3.12: Cámara del robot Khepera II.....	42
Figura 3.13: Esquemático de la conexión PS/2 en el FPGA	43
Figura 3.14: Esquema del puerto serial RS-232	43
Figura 3.15: (a) Conexión del adaptador tipo DB9 a DB25. (b) Un	

ejemplo de cómo se ven físicamente los conectores	44
Figura 3.16: Pines asignados en el conector DB25 del robot Khepera II	44
Figura 4.1: Torreta de visión cámara CMOS inalámbrico y receptor inalámbrico de audio y video	49
Figura 4.2: Imagen que se recibe desde la cámara observada por el humano	49
Figura 4.3: Envío de los datos desde el vector	51
Figura 4.4: El FPGA conectado al transmisor XBee para el envío de los datos por el puerto serie y el puerto PS/2 en color purpura al lado derecho que es la entrada en el circuito reconfigurable del teclado	53
Figura 4.5: De derecha a izquierda. Receptor de datos DB9 conectado al cable cruzado. Cable cruzado conectado al cable adaptador de DB9 a DB25. Conector DB25 conectado al adaptador del robot. Cable de datos conectado directo al robot.....	54
Figura 4.6: En esta imagen se observa el receptor de datos cerca del área de trabajo del robot que se encuentra realizando una tarea.	56
Figura 4.7: Envío de los datos desde el vector.....	56
Figura 4.8: El reporte de RTL que nos envía el software Quartus. .	57
Figura 4.9: Reporte de Technology Map Viewer dado por el Quartus II.	58
Figura 5.1: En esta figura se muestra el monitor de observación, el usuario. A través de la cámara montada sobre el robot, se observa un	

cilindro rojo.....	62
Figura 5.2: a) Objeto cubo amarillo.....	63
Figura 5.2: b) Cubo amarillo por ser insertado dentro de la caja	63
Figura 5.2: c) cubo azul siendo tomado por el robot.....	64
Figura 5.2: d) cubo azul dentro de la caj. Tarea completada con los dos objetos.....	64
Figura 5.3: a) El robot levantando el cubo verde.....	65
Figura 5.3: b) Se observa en el monitor el cubo verde sobre la marca verde	66
Figura 5.3: c) Se tiene al robot tomando el cubo azul	66
Figura 5.3 d) Se observa el cubo azul sobre la marca roja 	67
Figura 5.4: a) El robot está levantando el objeto a manipular.....	68
Figura 5.4: b) Tenemos al objeto sobre la posición final que se desea	68
Figura 5.5: a) robot aproximándose a tomar el cubo amarillo	69
Figura 5.5: b) tenemos el objeto puesto en la posición temporal	70
Figura 5.5: c) El robot tomando el segundo objeto	70
Figura 5.5: d) se posicionó el segundo objeto sobre el primero	71
Figura 5.5: e) tenemos el segundo objeto cambiarlo de la posición temporal, llevarlo a la final	71
Figura 5.6: a) El usuario moviendo el robot hacia el cilindro rojo .	72
Figura 5.6: b) El usuario ha logrado tele-operar el robot para situarlo lo suficientemente cerca del objeto a manipular.....	72

Figura 5.7: Robot Khepera II en el área de trabajo. 73

Figura B.1: Robot móvil Khepera II con pinzas y cámara 88

Figura B.2: Control de velocidad de las ruedas del robot Khepera II
..... 89

Resumen

En la presente tesis se describe una metodología para la manipulación a distancia de objetos mediante un robot móvil diferencial, mediante un control implementado en un circuito reconfigurable. El objetivo particular de esta tesis es lograr que un usuario humano tele-opere el robot manipulador, para que busque un objeto de geometría sencilla, lo tome mediante la pinzas del robot y realice una tarea dada dentro del espacio de trabajo del robot. En particular se probó el sistema con un robot móvil Khepera II y un FPGA de Altera DE-2.

El sistema desarrollado envía los comandos al robot para moverlo, accionar su brazo manipulador, tomar el objeto deseado y realizar así la tarea especificada, teniendo el control manual el usuario mediante un teclado como elemento de interface y un circuito reconfigurable como controlador.

Abstract

In this thesis we describe a methodology for the tele-operation a robot manipulator of objects by means of a differential mobile robot, controlled through a reconfigurable circuit. The particular goal of this thesis is that a human user tele-operate the robot to search for a given object, take it, the robot takes by means of the manipulator of the robot, and performs a desired task inside the working place of the robot. In particular, the system has been tested with a Khepera II mobile robot and the FPGA from Altera DE-2 model.

The developed system send the commands to robot to move it, to action the robot's manipulator, to take the object and to make thus the desired task. The user has the control of the robot by means of a keyboard acting as a joystick connected to the reconfigurable circuit acting as a like controller.

Capítulo 1

Introducción

El campo de la robótica, desde sus inicios fue concebido para el apoyo en las tareas que el ser humano realiza, muchas veces en entornos peligroso como son la limpieza de residuos tóxicos, el trabajo en plantas nucleares, las misiones de rescate, la vigilancia y el reconocimiento. Es deseable reducir el riesgo para las personas e introducir robots para dicha tarea, aunque el costo de éstos sea elevado.

Una de estas tareas es la manipulación de objetos, como lo hacemos con las manos o los pies; para esto, necesitamos una fuerza ejercida sobre dicho objeto para tomarlo, sostenerlo y transportarlo de un lugar a otro.

También, es deseable que dichos robots sean precisos y de pronta respuesta, por ejemplo, para el desarme o transportación de una bomba, se necesita una precisión milimétrica para desarmarla, por lo cual, se tiene un

circuito dedicado para el control del robot.

Las aplicaciones donde se están usando robots móviles van aumentando a medida que se optimizan los sistemas que los componen, entre otras se pueden mencionar las siguientes:

- Ayuda a personas con capacidades diferentes.
- Exploración de entornos hostiles como desiertos, volcanes, etc.
- Realización de tareas de forma remota a través de Internet.
- Guía en museos o expos.
- Seguir la trayectoria dada por una carretera.
- Llevar al robot de un punto a otro de una región poblada con obstáculos, evitando chocar con ellos.
- Tareas del hogar como barrer, limpiar, lavar, etc.
- Interceptar un objeto.
- Manipulación de un objeto dado dentro de un entorno cerrado.

1.1 Los primeros robots móviles

La inteligencia artificial y la robótica siempre han estado ligadas. Aún antes de la conferencia de 1956 en el Dartmouth College, en la cual el término inteligencia artificial fuera acuñado. Era bien sabido que la robótica podría ser

utilizada para realizar tareas interesantes así como para aprendizaje. William Grey Walter construyó un par de robots móviles a principios de los años 50 del siglo pasado [7]. Estos artefactos eran capaces de aprender tareas tales como la evasión de obstáculos y fototaxis al variar las cargas en el capacitor del robot, el cual controlaba el comportamiento del mismo. A finales de los 50's, pioneros en la inteligencia artificial, como Marvin Minsky y Jhon McCarthy, empezaron a interesarse en la robótica poco después de la conferencia de Dartmouth. Al final de los 50's, Minsky, junto con Richard Greenblood y William Gosper, intentaron construir un robot capaz de jugar ping-pong, pero debido a dificultades técnicas con el hardware de la máquina, se construyó solo un robot que atraparía una pelota usando una canasta en lugar de un dispositivo prensil.

En Stanford, Nils Nilson desarrolló el robot móvil SHAKEY [1] en 1969. Este robot disponía de un buscador visual de rango, una cámara y sensores táctiles binarios. Estaba conectado una computadora DEC PDP 10 a través de radio frecuencia. Las tareas de SHAKEY incluían la evasión de obstáculos y el desplazamiento de objetos en un entorno altamente estructurado. Todos los obstáculos eran bloques coloreados de manera uniforme y apilada. SHAKEY tenía almacenada una lista de fórmulas que representaban los objetos de su entorno y, usando una versión del sistema de demostración de teoremas llamado "STRIPS", determinaba los planes de las acciones que iban a ser ejecutadas.

Aunque el sistema de razonamiento trabajaba adecuadamente, SHAKEY a menudo tenía problemas para generar, a partir de los datos crudos obtenidos del sensor, la información simbólica necesaria para el planeador. El

hardware era la parte que dificultaba el proyecto. Hans Moravec, entonces estudiante de Stanford decía: “Una corrida entera de SHAKEY podía involucrar al robot dentro de un cuarto, buscar un bloque, solicitar mover el bloque hacia lo más alto de una plataforma, arrastrarlo a la rampa, y subirlo hasta la parte más alta. SHAKEY nunca hizo esto en una secuencia completa. Lo hizo en varios intentos diferentes, de los cuales en cada uno tenía una alta probabilidad de fallar. Se pudieran poner todas las piezas juntas en una película, pero sería poco fiable.”

También en Stanford, Jhon McCarthy empezó un proyecto a principios de 1970 para construir un robot, el cual pudiera ensamblar un kit de televisión de color. Nuevamente el hardware del robot – el acto físico de insertar componentes en tarjetas de circuitos impresos con suficiente precisión – demostró ser la parte difícil del problema. Muchos investigadores en los primeros días de la inteligencia artificial que estaban interesados, dejaron el aspecto de hardware de la robótica de lado concentrándose nuevamente en el software y los componentes de razonamiento del sistema de control.

El problema de diseño de un robot durante mucho tiempo recayó en la estructura del control del sistema. Una vez que los componentes necesarios de hardware fueran capaces de ayudar al elemento de razonamiento inteligente del controlador, el comportamiento inteligente sería inevitable y mucha de la investigación fue enfocada en los paradigmas de control. Más aún, un número influyente de proyectos de robótica fueron dedicados durante la época de los 70's.

El JPL Rover, desarrollado durante la década de 1970 en el Jet

Propulsion Laboratory en Pasadena, fue diseñado para la exploración planetaria, valiéndose de una cámara de TV, un buscador tipo láser de rango y sensores táctiles. El robot categorizaba su entorno como “reconocible”, “irreconocible” y “desconocido”. La navegación fue realizada mediante odometría usando un compás inercial.

En Stanford, Hans Morevac desarrolló CART a finales de la década de 1970. La tarea de este robot fue la evasión de obstáculos mediante una cámara como captor. El robot tomaba nueve fotos en una ubicación específica para crear un modelo del mundo bidimensional; entonces se movía un metro hacia adelante y repetía el proceso. El procesar las nueve imágenes le tomaba 15 minutos divididos en: 5 minutos para digitalizar las nueve fotos; 5 minutos para realizar una reducción de bajo nivel de la imagen, en las cuales, los obstáculos serán representados como círculos; y 5 minutos para el modelado del mundo y la planeación de trayectoria. CART evadía obstáculos con éxito, no obstante, era muy lento. Sin embargo, el robot tenía problemas para calcular correctamente su propia posición y para ver obstáculos con poco contraste.

En Europa, a finales de la década de 1970, HILARE fue desarrollado por la LAAS de Toulouse. Se trataba de uno de los primeros proyectos europeos en robótica móvil. HILARE usaba visión por computadora, buscadores tipo láser de rango y sensores ultrasónicos para navegar en su entorno. Un lento proceso de análisis de escenas, que operaba cada 10 segundos, y un rápido proceso de visión dinámico, que operaba cada 20 centímetros de movimiento, eran los principios subyacentes de control. Los obstáculos cercanos eran evadidos mediante los sensores ultrasónicos. La

navegación y la planeación de trayectorias eran almacenadas mediante polígonos bidimensionales, para la representación del espacio, también se usaba un sistema coordenado global.

1.2 Los inicios del FPGA

Los FPGA's (por sus siglas en inglés, Field Programmable Gate Array) son dispositivos semiconductores que contienen bloques de lógica cuya interconexión se puede programar. La lógica programable puede reproducir desde funciones tan sencillas como una compuerta lógica o un sistema combinatorio, hasta complejos sistemas en un chip.

Los FPGA's tienen la ventaja de ser reprogramables, sus costos de desarrollo y adquisición son mucho menores para tipos de dispositivos y el tiempo de desarrollo es mucho menor, sin embargo son mas lentos que los ASIC's (ASIC Application Specific Integrated Circuit), tienen mayor consumo de potencia y no pueden abarcar sistemas tan complejos como ellos. Son el resultado de la combinación de dos tecnologías diferentes, como son los dispositivos lógicos programables (PLD's Programmable Logic Devices) y los circuitos integrados de aplicación específica. Fueron creados en 1984 por Ross Freeman y Bernard Vondershmitt, quienes son co-fundadores de la empresa Xilinx y surgieron como la evolución de los CPLD's. Contienen un gran número de elementos lógicos programables. Si medimos la densidad de elementos lógicos programables en compuertas lógicas equivalentes (número de puertas NAND equivalentes que podríamos programar en un dispositivo) podríamos decir que en el FPGA hallaríamos del orden de cientos de miles

hasta millones de ellas. La comunicación es de gran importancia en los circuitos integrados, lo normal, es que el sistema pueda comunicarse mediante interfaces estándar de cable o inalámbricas, como pueden ser los puertos RS-232, USB, Wi-Fi, GSM, GPRS, etc. También, pueden contar con pantallas gráficas, táctiles, LCD, alfanuméricos, entre otras muchas.

Las aplicaciones donde se están usando este tipo de circuitos integrados van aumentando a medida que se optimizan los sistemas que los componen, entre otras se pueden mencionar las siguientes:

- En fábricas, para controlar un proceso de montaje o producción. Una máquina que se encargue de una determinada tarea hoy en día contiene numerosos circuitos electrónicos y eléctricos para el control de motores, hornos, etc. que deben ser gobernados por un procesador, el cual ofrece un interfaz persona – máquina para ser dirigido por un operario e informarle al mismo de la marcha del proceso.
- Puntos de información al ciudadano. En oficinas de turismo, grandes almacenes, bibliotecas, etc. existen equipos con una pantalla táctil donde se puede pulsar sobre la misma y elegir la consulta a realizar, obteniendo una respuesta personalizada en un entorno gráfico amigable.
- Sistemas radar de aviones. El procesado de la señal recibida o reflejada del sistema radar embarcado en un avión requiere alta potencia de cálculo además de ocupar poco espacio, pesar poco y soportar condiciones extremas de funcionamiento (temperatura, presión atmosférica, vibraciones, etc.).

- Equipos médicos, ambulancias, entre otros.
- Revelado automática de fotografías.
- Cajeros automáticos.
- Televisiones, teléfonos, radiograbadoras, reproductores, lavadoras, videos, monitores, impresoras.
- En un automóvil se pueden encontrar varios cientos de microprocesadores y micro-controladores que se encargan de la ignición, transmisión, dirección asistida, frenos anti bloqueo (ABS), control de tracción, seguridad, inyección de gasolina.

Muchos de estos sistemas son en tiempo real, esto es, que deben responder, dentro de un intervalo muy restringido de tiempo, a eventos externos mediante la ejecución de la tarea asociada a cada evento. Los sistemas en tiempo real se dividen en dos categorías: blandos y duros. Los blandos son aquellos en los cuales no se cumple con las restricciones de tiempo, simplemente se degrada el rendimiento del sistema, en cambio, los duros, si no se cumple con la restricción del tiempo, el sistema fallará.

1.3 Justificación

La manipulación a distancia de objetos ha sido de gran ayuda al humano en situaciones de peligro, como son la operación y desarmado de bombas, el trabajo en una planta nuclear, la toma de muestras en el cono de un volcán, la

revisión de la plataforma continental en profundidades que el hombre no soportaría; también en operaciones quirúrgicas a distancia, esto es un médico en lugar y el operado en otro. Hacia esta dirección está enfocada la presente investigación, en la cual, se probará con la manipulación a distancia de objetos de geometría sencilla en un entorno estático dentro de laboratorio, usando un robot móvil.

1.4 Objetivo general

Poner en operación una metodología, la cual se describe en el capítulo 4, que permita la navegación del robot Khepera II en un entorno estático con el objetivo de poder manipular objetos dentro del área de trabajo, usando un circuito reconfigurable como interfaz.

En la figura 1.1 se muestra en forma sencilla la idea de solución de tele-control a desarrollar en el contexto de esta tesis. Como se puede ver en la figura, un usuario humano al mirar un monitor conectado en forma inalámbrica a la cámara montada sobre el robot móvil analiza la escena observada. Con base en esta información, y la tarea a resolver, se accionan los botones del teclado conectado al robot a través del circuito reconfigurable para mover el robot y realizar así la tarea especificada.

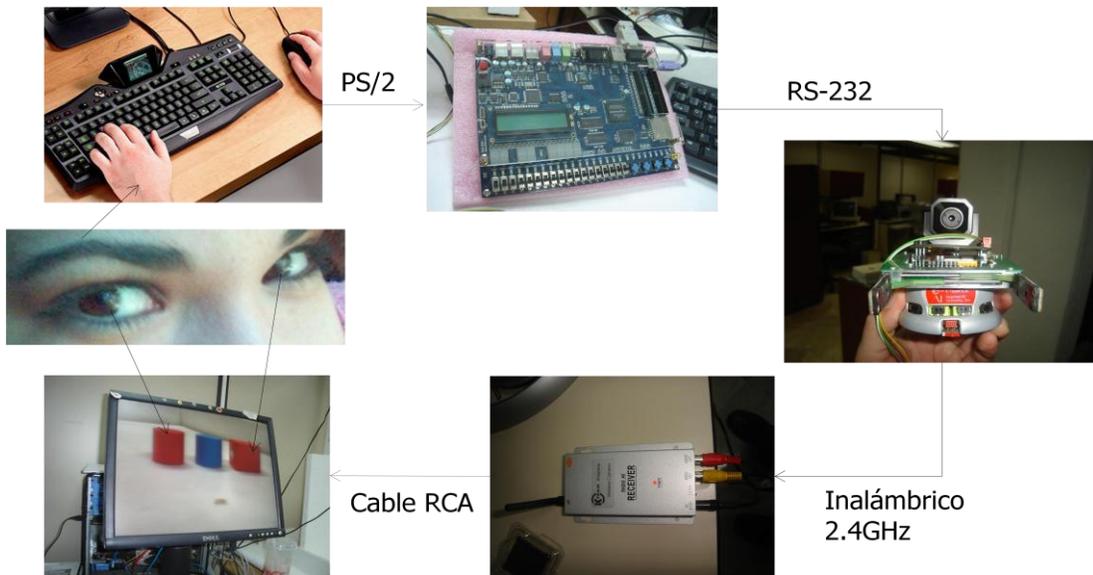


Figura 1.1. Diagrama de acción de la metodología propuesta para el sistema de tele-control de esta tesis.

1.5 Objetivos particulares

Los objetivos particulares de esta tesis son cuatro, a saber:

- Poner en operación un sistema para mover un robot Khepera II y poder realizar una tarea dada con un objeto previamente especificado dentro del espacio de trabajo del robot.
- Implementar en un circuito reconfigurable la comunicación entre un teclado y el robot a través de dicho circuito reconfigurable.
- Probar el control de movimiento del robot así como el de su elemento prensil para poder desplazar al robot y así tomar objetos especificados.

- Probar la comunicación entre la cámara montada en el robot y el monitor que observará el usuario para poder utilizar el teclado para manipular el robot.

1.6 Consideraciones del problema

Las siguientes son las consideraciones que se tomarán en cuenta para el diseño y puesta en operación del sistema que permitirá tele-controlar a un robot Khepera II para manipular objetos dentro de su espacio de trabajo:

- Se utilizará la cámara montada sobre el robot, la cual se conectará vía inalámbrica con un monitor que permitirá al usuario ver a través de dicha cámara dentro del espacio de trabajo del robot.
- La superficie sobre la que se desplace el robot será lo más plana y regular posible y con un color homogéneo.
- Los objetos a manipular serán de geometría sencilla como cubos y cilindros.
- Si hubiere obstáculos dentro del espacio de trabajo del robot, estos serán de color distinto a la superficie de trabajo del robot.
- Como teclado se utilizará uno con conexión PS/2.
- Finalmente, el FPGA contará con un cable serial RS-232 con salida inversa.

1.7 Aportaciones

Una metodología para la manipulación a distancia de objetos sencillos y en tiempo real mediante el elemento terminal de un robot Khepera II dentro del espacio de trabajo del robot. A mi conocimiento es la primera vez que se intenta el control de un robot de este tipo a través de un circuito reconfigurable.

1.8 Organización de la tesis

El resto de la tesis se divide en los siguientes capítulos, como sigue:

Capítulo 2: En este capítulo se presenta un estado del arte sobre los trabajos más importantes publicados que tienen una relación con el control de robots móviles, manipuladores y circuitos reconfigurables. En particular se hará énfasis sobre aquellos trabajos que aborden el problema del control a distancia de robots.

Capítulo 3: En este capítulo se presentarán los conceptos y fundamentos necesarios para la comprensión del material detallado en los siguientes capítulos.

Capítulo 4: En este capítulo se describen con detalle cada uno de los módulos del sistema diseñado, puesto en operación y probado.

Capítulo 5: Este capítulo contiene los resultados experimentales a los que se llegó, así como una discusión sobre los mismos.

Capítulo 6: Finalmente, en este capítulo se dan las conclusiones del trabajo y las posibles líneas futuras de investigación.

Capítulo 2

Estado del arte

En este capítulo se describen brevemente los trabajos más importantes relacionados con el control de robots móviles manipuladores, robots no holonómicos, tele-operación y circuitos reconfigurables que sirvieron de punto de partida de la presente tesis.

2.1 Manipulación robotizada de objetos basada en visión por computadora (2009)

En esta tesis de maestría desarrollada en el CIC-IPN [4], el autor implementa algoritmos de manipulación y visión por computadora para un robot CRS Robotics A465. Este robot es de seis grados de libertad (figura 2.1) y cuenta con un controlador articular de lazo cerrado, al cual, se le agregó un lazo

externo para la adquisición de imágenes y su procesamiento.



Figura 2.1: Robot manipulador modelo CRS.

El sistema diseñado a través del desarrollo de esta tesis, incorpora algoritmos para el procesamiento de imágenes como son: segmentación, extracción de características y clasificación. El resultado es la clase a la cual pertenecen los objetos a reconocer, ubicar y manipular. El objetivo de esta tesis consistió en solucionar el juego del Sudoku en un tablero montado al alcance del robot. El ambiente es controlado: las piezas a manipular son estáticas, cuentan con una marca para ser reconocidas en caso de traslaciones,

rotaciones y otras transformaciones tipo imagen. El fondo de trabajo es contrastante con respecto a los objetos a manipular, en este caso, números.

El tablero donde se colocan las piezas también es de color negro con divisiones color blanco para diferenciar los cuadrantes y tener una segmentación de las casillas.

Los resultados preliminares para probar la comunicación y movimiento de las piezas, control y manipulación del robot fue hecho con unos cubos que se encontraban debajo del brazo del robot con la cámara viendo hacia abajo, donde se encuentran los objetos. La tarea a realizar era intercambiar la posición de los cubos.

2.2 Generación de trayectorias para un robot móvil Khepera II usando técnicas de aprendizaje automático (2007)

En esta tesis de maestría desarrollada también en el CIC-IPN [5], el autor presenta una metodología para la generación de mapas al representar el área de trabajo como una malla regular de celdillas, de la cual se tiene una vista aérea. A partir de esta vista, se segmenta la imagen en espacio libre y obstáculos.

Se hace una planificación de la ruta a seguir, tomando en cuenta la

imagen obtenida. También se probaron varios algoritmos para la obtención de rutas.

Se propuso un planificador de trayectorias a partir de *splines cúbicos*, el cual suaviza las curvas. Puesto que el robot es tipo no holonómico, en la especificación de la ruta se obvian las características y dinámicas del vehículo, ya que el cómputo de una referencia adecuada que cumpla con estos atributos, es tarea del generador de trayectorias. Por lo tanto, la ruta solo asegura continuidad en posición y en general, únicamente los robots móviles omnidireccionales pueden seguir una referencia con esas características.

2.3 Control visual de un robot móvil Khepera II (2007)

Esta tesis de maestría se hizo en el CIC – IPN [2], en la cual, el autor presenta un modelo de control de un robot Khepera II para interceptar objetos en movimiento, en este caso, una pelota. En esta tesis, el robot se utiliza como portero, para lo cual, implementaron un lanzador de pelotas para lanzarlo a casi a la misma velocidad, una portería y se midió la cantidad de veces que la pelota entró.

Con este trabajo, se hizo el control visual, esto es ver el área de trabajo, reconocer los objetos, como son el robot y la pelota, y hacerlo en tiempo real para el movimiento de la pelota y del robot, que sepa dónde está, donde va a estar la pelota y hacia donde se tiene que mover el robot para interceptar la pelota.

2.4. Control visual reactivo de múltiples agentes robóticos no holonómicos (1998)

En este trabajo [6], los autores muestran el desarrollo de un sistema robótico multi-agente, el cual, fue usado en la RoboCup-1997, en particular en la liga de robots pequeños. El equipo consiste de cinco robots no holonómicos de tamaño 15cm x 12cm x 10cm, con un mecanismo de conducción diferencial.

Este artículo presenta un enfoque de localización por medio de asociación, que permite la identificación de los robots que visualmente son homogéneos.

Los autores introducen un control no holonómico que de manera reactiva planea la trayectoria del robot por medio de parámetros de movimiento de rotación y de traslación al utilizar la siguiente fórmula:

$$(v, w) = \{(\alpha \cdot \cos \theta, \beta \cdot \sin \theta)\} \text{ si } |\theta| < \frac{\pi}{4} \text{ o } |\theta| > \frac{3\pi}{4}$$

$$\{(0, \sin \theta \times \beta_0)\} \text{ de otro modo} \quad (2.1)$$

donde α y β son constantes, v y w son las velocidades de traslación y rotación respectivamente y π el error de orientación.

Esta regla de control, de manera implícita, permite una independencia de orientación, es decir, la regla de control permite movimientos hacia adelante y hacia atrás, eligiendo siempre el que sea más eficiente de realizar.

2.5. Soccer Teleworkbench for Development and Analysis of Robot Soccer (2006)

En este artículo de la Universidad de Paderborn, Alemania [8], los autores describen una plataforma para realizar y validar experimentos, usando robots reales en un partido de fútbol. Algunas de las características más importantes en este trabajo son: video en tiempo real, grabación de los videos, sistema de posicionamiento del robot, programación de cada robot de manera independiente y análisis de las herramientas después de los experimentos. Dicha plataforma puede ser usada o vía Internet (figura 2.2).

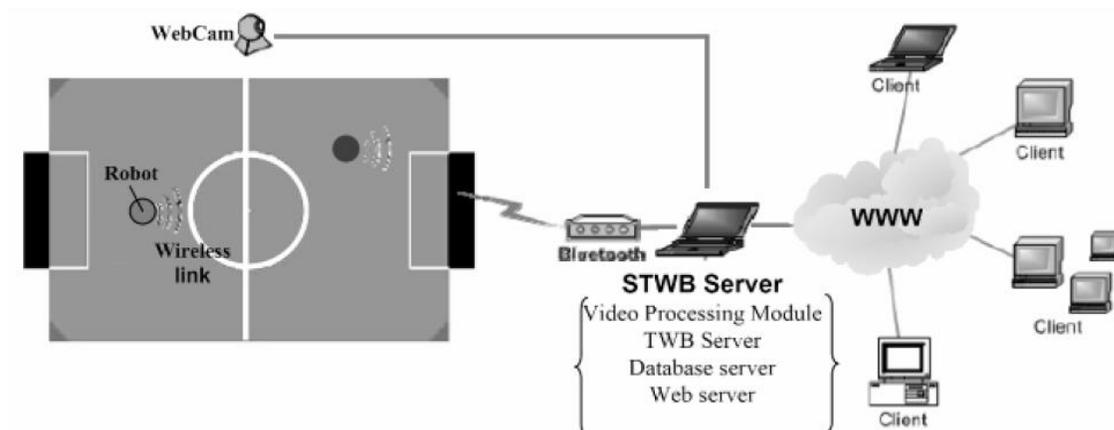


Figura 2.2: Diagrama de soccer teleoperado.

Como se muestra en la figura anterior, se cuenta con una cámara web puesta sobre el campo enviando datos en crudo para el procesamiento de imágenes y el video en tiempo real. Con el video se tienen dos situaciones: En la primera se cuenta con la posición y orientación en la que se encuentra el robot en el campo; en la segunda se emite video al servidor. Otros módulos del

sistema son:

- Un servidor web, el cual, está programado en PHP; con él se puede ver el video en tiempo real del campo, enviar comandos al sistema y descargar el programa de futbol al robot.
- Un servidor de video, el cual transmite el video en tiempo real sobre la Internet con el protocolo Real Transport Protocol (RTP).
- Análisis después del experimento y herramienta de visualización: es un archivo de registro que guarda el archivo log con toda la información del experimento, como podrían ser los sensores, datos y mensajes entre los robots, posiciones del robot, entre otras.

Estos experimentos los realizan con el robot Khepera II equipado con una cámara sobre el robot (figura 2.3).



Figura 2.3: Robot Khepera II equipado con un módulo FPGA y una cámara

en 2D a color.

Se usa el FPGA para el control de la cámara que adicional mente captura y procesa las imágenes.

2.6 Desarrollo de un robot móvil con control inteligente mediante algoritmos evolutivos y un FPGA (2006)

En este artículo [9], se presenta un método para navegación del robot XC-3. Se describe una serie de algoritmos conformado por sub-algoritmos con el fin de que se complemente la biología y la electrónica para formar un ser viviente digital. Se dotó a este de algo que sus autores llaman “instinto electrónico” y que no le permite al robot realizar acciones que le hagan daño. Es capaz de desechar soluciones que ya haya aprendido y guardar nuevas.

Cada parte electrónica del robot está conectada directamente a el FPGA, ellos usan el SPARTAN 3 XC3S200, el cual cuenta con 200,000 compuertas lógicas programables. Sobre el FPGA, se tienen los módulos de algoritmos genéticos que son:

- **GENECOD:** maneja una matriz dinámica de 16 x 16 palabras y cada palabra compuesta de 4 bits. Su salida se pasa al “chip ADN emulado electrónica mente”.
- **PRINCI:** se encarga de guiar al robot hacia adelante, siempre y cuando los sensores no registren ningún obstáculo.

- CODIF: codifica la salida de ENTRADA o PRINCI en acciones, en este caso, movimientos hacia adelante, atrás, gira izquierda, gira derecha, quietud.
- ENTRADA: aquí es donde se encuentra el chip ADN emulado electrónicamente. Se encarga de registrar los sensores.
- MULTICODI: es un multiplexor que deja pasar el dato que se va a decodificar en CODIF.
- MULTISALIDA: es un multiplexor que deja pasar acciones que a generado CODIF o el modulo FIJO, es gobernado por AG.
- AG: es el encargado de pasar la acción que genera ENTRADA por el instinto electrónico.
- RETIENE: retiene las decisiones que toma el XC-3 cierta cantidad de tiempo, esto con el fin de que las acciones sean notables y lo dejen moverse con más libertad.
- FIJO: deposita las acciones buenas guardadas por el AG luego de haber aprendido.
- PWM: es el módulo encargado de enviar las señales a los servomotores.
- MENSAJE: visualiza el mensaje LEARNED cuando el robot a aprendido totalmente o individualmente.

Todos estos módulos se encuentran programados en VHDL dentro del FPGA.

2.7 Resumen

En este capítulo se describieron los trabajos más representativos sobre: manipulación de objetos, robots móviles, robots no holonómicos, tele operación de robots y un robot móvil implementado en un FPGA, que directamente tienen que ver cada uno de ellos, con una parte del desarrollo del presente trabajo de investigación.

La principal diferencia con dichos trabajos, es que, en esta investigación, se tendrá un control supervisado, donde el humano es el que controla los movimientos del robot y del elemento terminal.

La diferencia con el primero de ellos, es que en este trabajo, se desarrolla en un robot móvil. En el segundo y tercer trabajo, tenemos el mismo robot, pero en este caso, no se cuenta con el elemento prensil. En el cuarto artículo, se tiene el modelado del robot, el cual sirvió para la continuación y trabajos posteriores con este robot. El quinto trabajo, nos refiere a un control de los robots a distancia, en este caso por internet y en el sexto y último trabajo referido, se cuenta con un robot desarrollado en un fpga con control inteligente usando algoritmos evolutivos. Para estos últimos, tenemos que el envío de los comandos es por medio de cableado y con el último, el control es supervisado.

Estas serían las principales diferencias de los trabajos presentados en este apartado con el trabajo a realizar.

Capítulo 3

Marco teórico

En este capítulo se presentan las bases teóricas necesarias para el desarrollo de esta tesis. Se muestran las ecuaciones que describen la cinemática del robot Khepera II, así como los conceptos básicos de un robot manipulador. Se introducen los conceptos básicos de FPGA.

3.1. Robots manipuladores

Un manipulador o brazo mecánico es un conjunto de elementos mecánicos que mueven un elemento terminal con el que se realiza la tarea. Generalmente a este dispositivo se le conoce como robot, ya que realiza un trabajo.

La palabra robot proviene del checo *robota*, que significa trabajo. El Instituto Norteamericano de Robótica (INR) define a un robot industrial como un manipulador multifuncional y reprogramable, útil para mover materiales, piezas,

herramientas o dispositivos especiales, mediante movimientos programables y variables que permiten llevar a cabo diversas tareas [10].

La principal característica que distingue a un robot de otras máquinas es su capacidad para realizar diversas tareas en el mismo espacio, debido a que se puede re-programar mediante una computadora o un circuito reconfigurable.

3.1.1. Articulaciones y grados de libertad

Una articulación conecta dos elementos rígidos permitiendo un movimiento relativo entre ellos, la cantidad de movimientos que permite esta articulación se le conoce como grado de libertad. Existen diferentes tipos de articulaciones, entre ellos se tienen las siguientes:

- 1.** Articulación angular: conecta dos elementos por medio de un perno permitiendo que exista un movimiento relativo de rotación.
- 2.** Articulación planar: se conoce así, porque los movimientos que permite están contenidos en el plano.
- 3.** Articulación cilíndrica: esta articulación permite dos tipos de movimientos: una rotación y una traslación en un eje.
- 4.** Articulación prismática: permite un movimiento a lo largo de un elemento.
- 5.** Articulación esférica: tiene tres grados de libertad, cada uno de rotación. La implementación de los movimientos en una articulación esférica no es sencilla, por lo que no es común encontrarlo en los robots manipuladores.

3.1.2. Espacio de trabajo

Un manipulador mecánico está constituido por una serie de elementos rígidos unidos entre sí [10]. Cada una de las articulaciones cuenta con al menos un grado de libertad. Por ejemplo, para un manipulador cartesiano de N grados de libertad existirían N parejas de elementos rígido-unión.

Los robots manipuladores pueden tener diferentes configuraciones, que dependen del tipo de las articulaciones que usen o de la tarea a realizar.

Conocer el espacio de trabajo de un robot es importante, ya que de ahí se puede conocer las restricciones de movimientos que tiene el robot. Para que el robot pueda manipular un objeto, éste debe estar dentro del área de trabajo.

El espacio de trabajo de un robot manipulador es el volumen que es capaz de alcanzar el efector final [11]. A continuación se explican diferentes manipuladores y su respectivo espacio de trabajo.

Manipulador cartesiano: es la configuración más sencilla. Tiene tres articulaciones prismáticas que son ortogonales entre sí y corresponden a las direcciones \vec{x} , \vec{y} , \vec{z} de un sistema de coordenadas cartesiano (figura 3.1).

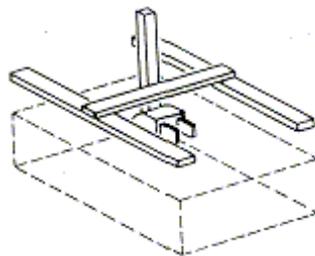


Figura 3.1: Manipulador cartesiano.

Manipulador articulado: también conocido como manipulador antropomórfico como se presenta en la figura 3.2. Las articulaciones que lo forman son angulares. Su característica principal es que es capaz de llegar a lugares o espacios confinados.



Figura 3.2: Manipulador articulado.

Manipulador SCARA: el manipulador está formado por articulaciones angulares paralelas. Ver figura 3.3, que le permite moverse en un plano. El uso principal es en el ensamble de tarjetas electrónicas, cuyos componentes se encuentran en un plano.



Figura 3.3: Manipulador tipo SCARA

Manipulador esférico o polar: está constituido por articulaciones angulares y una articulación prismática, que al moverse, el espacio que define es una esfera. Ver figura 3.4.

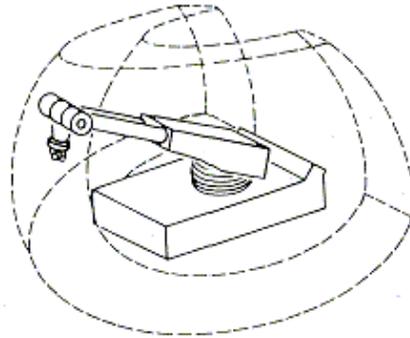


Figura 3.4: Manipulador polar.

Manipulador cilíndrico: tiene dos articulaciones prismáticas y una angular que permite dos traslaciones y una rotación, ver figura 3.5.

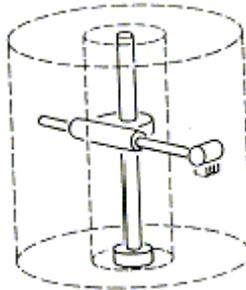


Figura 3.5: Manipulador cilíndrico.

3.2. Robots móviles

Un robot móvil es un dispositivo con ruedas, orugas, piernas, tentáculos y otros medios que le permiten desplazarse. En esta sección se habla principalmente de los móviles con ruedas, con un enfoque dirigido hacia el robot modelo Khepera II.

3.2.1. El robot Khepera II

El robot Khepera II es un robot con dos ruedas montadas sobre un eje común. Estas ruedas a través de dos motores le permiten moverse dentro de su espacio de trabajo. En esta sección se habla de este tipo de robot, se habla también de su cinemática y de su dinámica. La cinemática del robot trata los aspectos de movimientos sin tomar en cuenta aspectos de fuerza y/o torque que cause ese movimiento [12].

Para implementar las técnicas de control de movimiento del robot Khepera II se deben entender las leyes de la cinemática que gobiernan a este tipo de vehículos. Como ya se dijo, este tipo de robots se mueven mediante dos ruedas montadas sobre un eje común. Cada rueda es controlada a través de un motor por separado. Este mecanismo de manejo es conocido como *manejo diferencial*. Las ruedas de los robots mantienen un único punto de contacto con la superficie, de acuerdo a esto, los radios de la rueda son perpendiculares a la superficie, con lo cual se puede suponer que el vehículo no resbala en la dirección ortogonal a la del movimiento, ni tiene deslizamiento de traslación entre la rueda y el piso (rodamiento puro) en la dirección del movimiento.

El control de velocidades de las ruedas del robot determina el movimiento del robot, al variar las velocidades de las ruedas, se pueden variar también las trayectorias que el robot describe.

Para la derivación de las ecuaciones que describen la cinemática del robot Khepera II, se considera a éste como un sólido rígido moviéndose sobre un plano, más aún se supone que solo hace rotaciones y no traslaciones. Estas suposiciones están bien fundamentadas pues:

1. El robot no se deforma mientras se desplaza.

2. No hace traslaciones salvo el caso en que las velocidades de ambos motores sean idénticas. Sin embargo, en este caso, se puede considerar que hace una rotación alrededor de una circunferencia de radio infinito.
3. El robot se mueve sobre un plano, todos sus puntos se mueven paralelamente al plano de la superficie donde se desplaza.

3.2.2. Programación de un robot

La programación de un robot permite que éste sea flexible y se pueda adaptar a casi cualquier tarea que necesite desarrollar.

La programación de un robot se orienta al control de los motores así como a la generación de movimientos en sus articulaciones. Con el objetivo de llegar a un punto y realizar alguna tarea con el efector final; desde tareas sencillas como: manipular objetos hasta operaciones que requieren un alto grado de precisión, como por ejemplo, el fungir como asistente de un cirujano.

Los robots actualmente se pueden programar de diferentes maneras, desde comandos de pulsos modulados, hasta comando de voz.

Para que el programa pueda controlar al robot, es necesario que el robot sea calibrado, esto se realiza moviendo el robot a la posición de inicio y luego a la posición preparado.

Posición inicio: para poner en marcha el robot, es necesario mandarlo al punto conocido como inicio, es una posición donde los sensores (encodificadores incrementales) tienen la referencia inicial. Ya que los encodificadores incrementales no tienen una posición absoluta de cero. Este punto le sirve al controlador como

realimentación para conocer la posición del robot y controlar sus movimientos.

Posición preparado: una vez que el robot ya tiene la referencia del punto inicial, hay que verificar que el robot es capaz de moverse a los puntos que se le enviarán por comando. El robot tiene una posición de fábrica que ayudará a verificar la calibración del robot. Este punto se le conoce como punto listo. Al mover el robot al punto listo, el robot está preparado para realizar los comandos que le sean enviados.

3.2.3. Movimientos del robot

El controlador del robot tiene dos maneras de mover al robot que son las siguientes:

1. Movimiento en el plano. Este movimiento provoca que el robot se desplace en plano. Esto lo hace a través del movimiento de sus llantas al ser accionadas por los dos motores (figura 3.6).
2. Movimiento del brazo manipulador. El robot viene equipado con un brazo manipulador. Este robot cuenta con dos movimientos, uno circular en el plano vertical del robot, como se muestra en la figura 3.7.

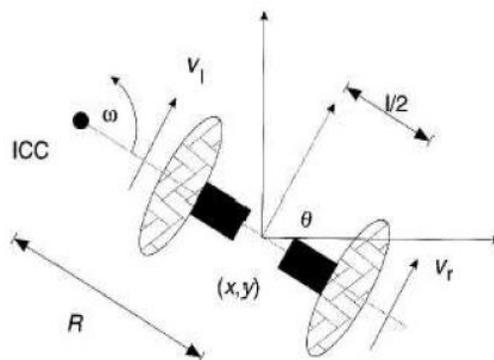


Figura 3.6: Ángulo de movimiento del robot base.

3. Apertura y cerradura del extremo terminal del brazo robótico. Como se muestra en la figura 3.8, el extremo terminal del robot se puede abrir o cerrar con el objeto de tomar los objetos a manipular.

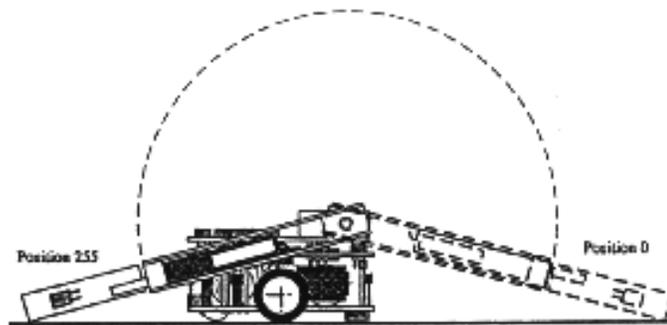


Figura 3.7: Ángulo de movimiento de la pinza.

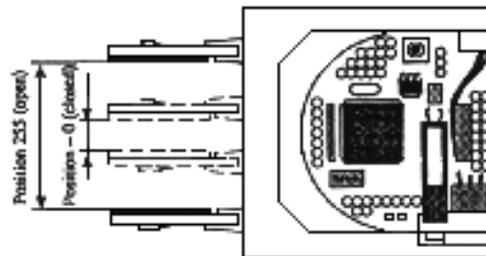


Figura 3.8: Movimiento de la pinza del robot.

3.3. Circuitos reconfigurables

Anteriormente, los dispositivos comúnmente usados en el diseño digital eran circuitos integrados de funcionalidad fija. La operación de estos componentes, estaba completamente definida por el fabricante y fuertemente restringida por el proceso de manufactura; tal es el caso de la conocida familia 7400 de circuitos integrados basados en lógica de transistor-transistor (TTL), que permitieron la construcción de los primeros equipos de computo.

Actualmente, en el ámbito de diseño y fabricación de prototipos, existe una clara tendencia hacia el uso de dispositivos basados en lógica reconfigurable. La razón principal del por qué son un recurso frecuentemente empleado, resulta del hecho de que la funcionalidad en este tipo de dispositivos no es fija, consecuentemente, tanto en los circuitos integrados de aplicación específica (ASIC) como en los arreglos de compuertas programables (FPGAs), la funcionalidad del dispositivo queda completamente en manos del diseñador.

3.3.1. FPGA

Los arreglos de compuertas programables, son dispositivos semiconductores principalmente estructurados por celdas lógicas interconectadas mediante una matriz de switches programables, figura 3.9. Típicamente, cada una de estas celdas es capaz de llevar a cabo tareas combinatorias o secuenciales, de manera que cualquier diseño puede ser implementado mediante la descripción funcional de cada celda lógica y la selección de los switches adecuados que permiten la interconexión entre los elementos involucrados.

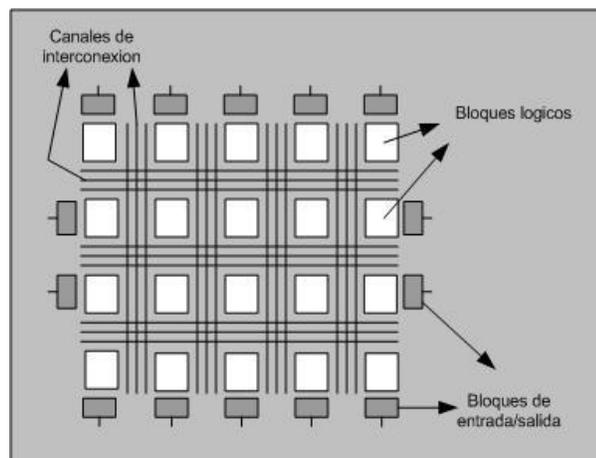


Figura 3.9: Arquitectura general de un FPGA.

Actualmente, los FPGAs comerciales emplean alguno de los siguientes elementos para fabricación de celdas lógicas:

- Pares de transistores
- Compuertas lógicas NAND o XOR de dos entradas
- Multiplexores
- Look-up tables (LUTs)

Existen tres tecnologías mayormente difundidas para la programación de los switches que interconectan cada celda lógica:

- Memoria estática de acceso aleatorio (SRAM), donde el proceso de conmutación lo realiza un transistor de paso, controlado por un bit de estado, almacenado en la memoria.
- Memoria de solo lectura, que se puede borrar y programar (EPROM), donde el proceso de conmutación puede anularse al aplicar cargas eléctricas.
- Antifusible, donde al programar el dispositivo, se forman rutas de baja resistencia.

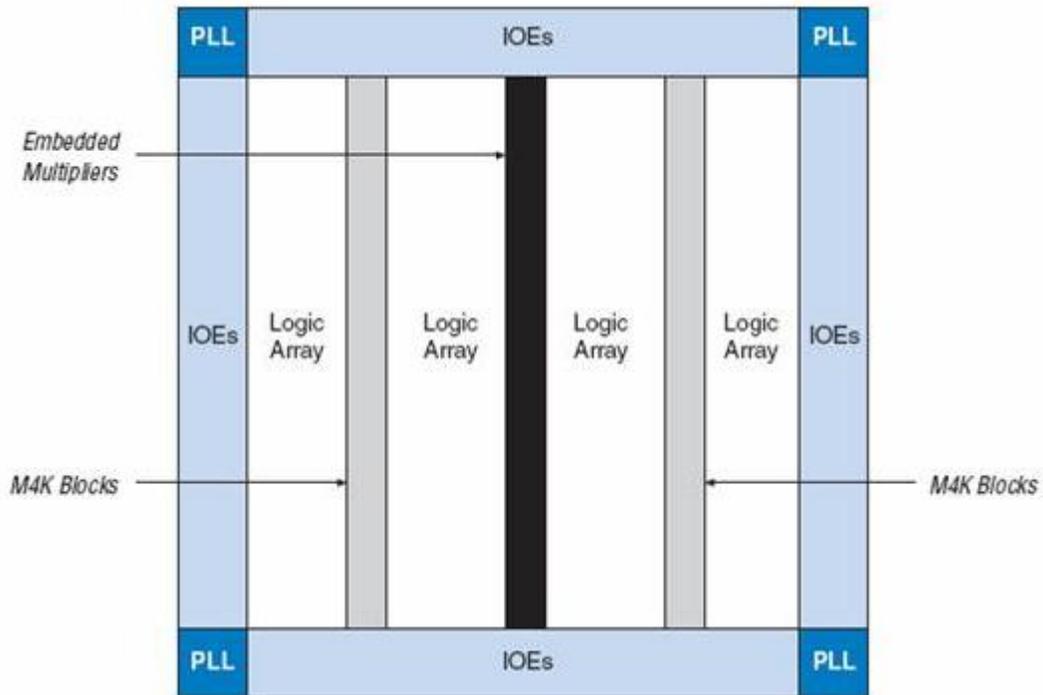


Figura 3.10: Diagrama de bloques del FPGA arquitectura de Altera.

Particularmente, el FPGA utilizado para la realización del presente trabajo de tesis, pertenece a la familia DE-2 del fabricante Altera, el cual, cuenta con bloques M4K interconectado. En la figura 3.10 se muestra el diagrama de bloques del circuito reconfigurable de Altera usado en el contexto de esta tesis.

Lenguajes de descripción de hardware. En un principio, los lenguajes de descripción de hardware (HDL) fueron usados meramente como un conjunto de sentencias simples, para definir la interconexión entre los elementos involucrados en el diseño de un sistema digital. A estos lenguajes se les llamó *Net list*, puesto que eran simplemente eso, un archivo con una lista de conexiones.

Años más tarde, el nivel de abstracción requerido para la verificación funcional de cada uno de estos elementos, puso de manifiesto la falta de herramientas apropiadas para la simulación, consecuentemente, los lenguajes de

descripción de hardware evolucionaron hacia niveles más altos de abstracción, facilitando la obtención de modelos sintetizables; en otras palabras, cuanto mayor es la funcionalidad de las sentencias disponibles en el lenguaje, mayor es la similitud entre la etapa de modelado y la de verificación funcional.

Aunque existen dos lenguajes principales para la descripción de hardware: VHDL y Verilog, en el presente trabajo de tesis se tomó la decisión de apegarse al primero de ellos.

El lenguaje de descripción VHDL. VHDL surge como una iniciativa del Departamento de Defensa de los Estados Unidos, para la aplicación de técnicas de diseño jerárquico que faciliten no solo el proceso de documentación de un circuito digital, sino que además, permita modelar un sistema digital completo desde diferentes niveles de abstracción, que van desde un panorama algorítmico hasta el nivel de compuertas lógicas.

Algunas ventajas que ofrece VHDL son:

- Gran cantidad de tipos de datos predefinidos por el lenguaje.
- Las unidades diseñadas pueden ser compiladas por separado.
- Los procedimientos y funciones pueden incluirse en paquetes separados.
- Los paquetes están disponibles para cualquier unidad.
- Mayor número de constructores para modelado de alto nivel.
- Permite tener unidades compiladas, almacenadas en librerías.
- Permite establecer llamadas concurrentes a procedimientos.

Unidades de diseño en VHDL. Para cualquiera de las dos etapas (diseño o simulación) previas a la implementación de un modelo sobre dispositivos basados en lógica reconfigurable, es necesario dividir el diseño en bloques abstractos, conocidos como componentes o unidades.

La unidad de diseño *entity*, proporciona la información concerniente a los puertos de entrada y/o salida, mientras que la unidad de diseño *architecture*, proporciona la descripción funcional de la entidad; cabe mencionar que, es posible tener múltiples descripciones funcionales de la misma unidad.

Estructura del código en VHDL. Existen tres tipos de enunciados que permites instanciar múltiples unidades de diseño:

- Declaraciones. Estas deben aparecer antes de ser instanciada una unidad de diseño.
- Postulados secuenciales. Este tipo de enunciados son ejecutados dependiendo de las condiciones del flujo de datos.
- Postulados concurrentes. Este tipo de enunciados son ejecutados en paralelo, es decir, en el mismo instante de tiempo e independientemente de otros postulados concurrentes o secuenciales.

Objetos de datos en VHDL. Existen cuatro tipos de objetos de datos en VHDL:

Constantes: una constante puede mantener un único valor. Este valor es asignado antes de iniciar el proceso de simulación, y no puede ser cambiado durante la ejecución. Ejemplo:

Constant prendido : integer := 1;

Variabes: Una variable puede ser usada para almacenar valores temporales dentro de un proceso; sin embargo, estos valores pueden cambiar en diferentes instantes de tiempo. Ejemplo:

Variable enviodata : bit_vector (3 downto 0);

Señales: Una señal puede mantener una lista de valores, que incluyen el valor actual y un conjunto de valores futuros. Las señales pueden ser vistas como conexiones (alambres en un circuito), elementos de almacenamiento simple (*flip-flops*) e incluso esquemas de almacenamiento agrupado (registros). Ejemplo:

Signal ledr:bit;

Signal busdatos : bit_vector (15 downto 0);

Signal registro : bit_vector (30 downto 0);

Signal tiempo : time := 10ns;

Archivos: un objeto de tipo archivo (*file*), hace referencia a un sistema de archivos. Este tipo de objeto no es sintetizable; sin embargo, es de vital importancia durante la etapa de simulación, ya que permite proporcionar los estímulos de entrada y obtener archivos de salida. Ejemplo:

File archivo: text open read_mode is “./arch1.txt”;

File arch : text open write_mode is “./arch2.txt”;

3.4. Control visual

El término control visual se refiere al uso de datos provenientes de un elemento

captor para el control de movimiento de un robot. Los datos de visión pueden ser adquiridos a través de una cámara montada directamente sobre un robot manipulador o sobre un robot móvil. En cada caso, el movimiento del robot induce un movimiento en la cámara. La cámara puede estar fija en el espacio de trabajo, de tal modo que el movimiento del robot puede ser observado desde una configuración fija. Otras configuraciones pueden ser consideradas tales como, varias cámaras montadas en cabezas de tipo *pan-tilt* observando el robot en movimiento.

3.4.1. Captor de imágenes

El primer componente con que cuenta un sistema de visión es el elemento captor (cámara). Mediante este dispositivo, se pueden adquirir la imagen de lo que tiene el robot frente a él. En el caso del robot Khepera II, éste viene equipado con un captor con una resolución de 380 x 450 píxeles a color. Figura 3.11.

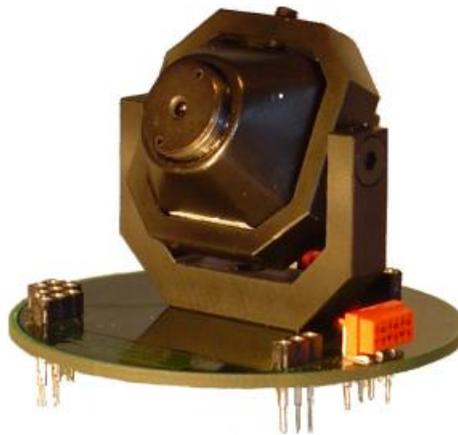


Figura 3.11: Cámara del robot Khepera II

La iluminación es un componente importante en la formación de una imagen. Si la iluminación no es homogénea, se tienden a formar sombras entre los objetos y a confundirse como parte de los mismos.

Si la intensidad de la luz es demasiada, tiende la imagen a ser blanca y confundirse con un solo objeto. Por el contrario, si la iluminación es escasa, la imagen tiende a ser negra y los objetos desaparecerían de la imagen.

Adicionalmente, el robot cuenta con una receptor que permite ser conectado a un monitor vía RCA para visualizar la información captada por el captor montado en el robot.

3.5. Controlador del robot manipulador

Como controlador para el robot se ha decidido utilizar un teclado que funge como un joystick para tener el control de los movimientos del robot y sus pinzas para manipular los objetos deseados. Esto permite al usuario controlar el robot al observar el monitor conectado a la cámara montada sobre el robot. Tanto el movimiento del robot como del brazo manipulador son controlados a través de algunas de las teclas del teclado.

3.5.1. Tipo de teclado

Existen muchos tipos de teclados, entre ellos, los matriciales que pueden ser hechos para la realización de tareas sencillas. En el contexto de esta tesis se usará un teclado con conexión PS/2 con estándar de 101 teclas. El esquemático de la conexión del teclado está dado por la figura 3.12.

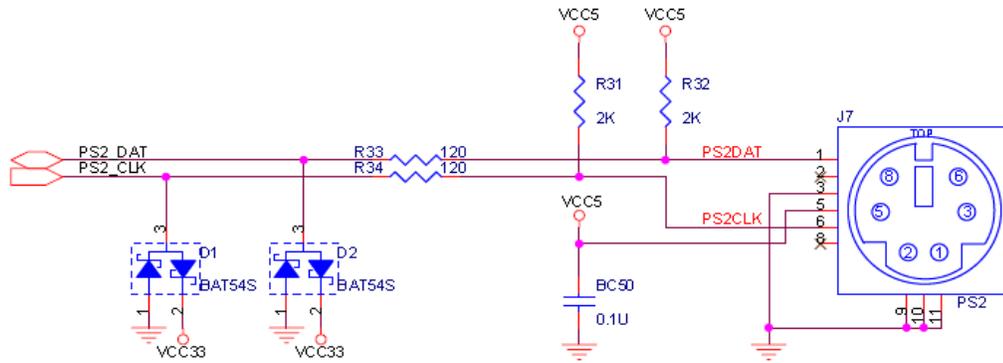


Figura 3.12: Esquemático de la conexión PS/2 en el FPGA.

3.5.2. Puerto serie RS-232

Para esta conexión existe un estándar de envío de datos, que es como se muestra la figura 3.13, en la cual se puede apreciar el bit más significativo (MSB), el menos significativo (LSB), un bus de datos de 8 bits, un bit de inicio y un bit de parada.

Para lograr la comunicación con otros dispositivos, se cuenta con los siguientes pines asignados. Cabe mencionar que existen más pines conectados, pero para este trabajo, solo ocuparemos los siguientes:

TXD, transmisor de los datos.

RXD, receptor de los datos.

GND, masa de señal o protección a tierra.

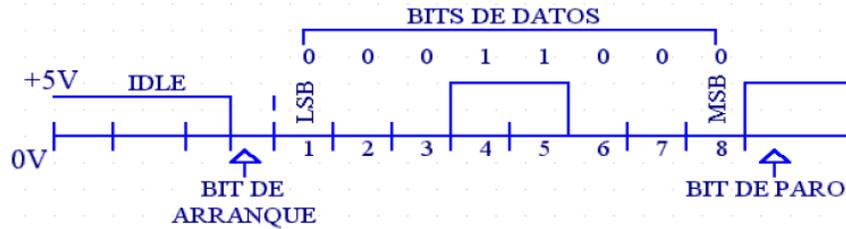


Figura 3.13: Esquema del puerto serial RS-232

El puerto DB9 cuenta con un adaptador a DB25 que es la conexión al robot. La conexión del robot se realiza mediante un conector DB25, por lo tanto, se tiene un adaptador conectado a la salida DB9-DB25, el cual se muestra en figura 3.14. Las conexiones del puerto serie del robot son las mostradas en la figura 3.15.

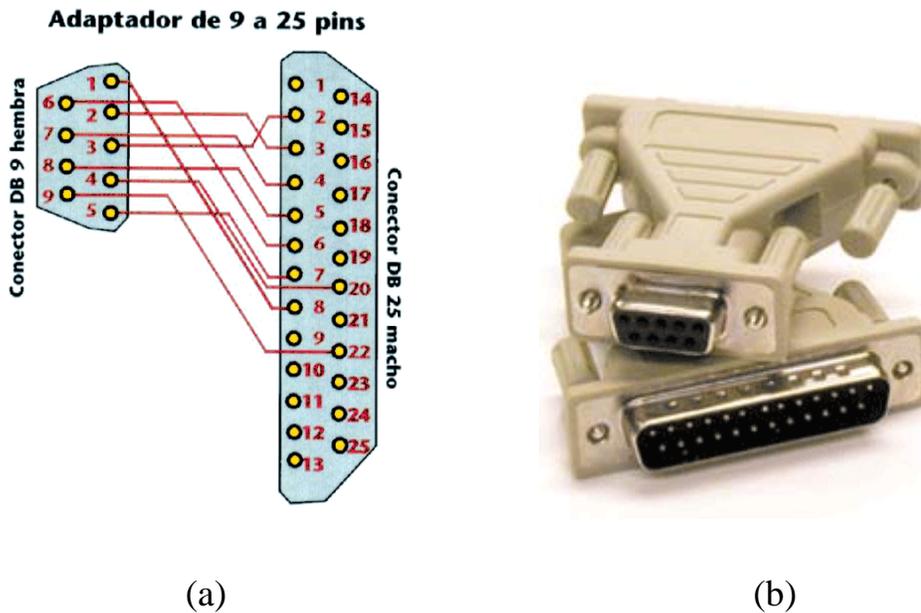


Figura 3.14: (a) Conexión del adaptador tipo DB9 a DB25. (b) Un ejemplo de cómo se ven físicamente los conectores.

Puesto que los pines del conector del robot tienen una conexión directa, se tiene que conectar un inversor de salidas para que el RX del FPGA quede en el TX del robot y viceversa. Como se muestra en la figura 3.16.

3.5.3. Módulo de comunicación inalámbrica

Con este equipo, se cuenta con una extensión de la comunicación entre los conectores DB9, el primero de los cuales está conectado directamente a la salida del puerto RS-232 del FPGA y el otro equipo al cable cruzado que conecta el adaptador del robot.

En la figura 3.17 podemos ver el módulo Xbee conectado al FPGA.

3.6. Resumen

En este capítulo, se presentaron los conceptos básicos y las definiciones que, como veremos en el siguiente capítulo, son base para plantear la metodología propuesta que permitirá resolver el problema planteado en el primer capítulo. En el siguiente capítulo se describen cada una de las etapas que componen la metodología propuesta.

Capítulo 4

Metodología propuesta

En este capítulo se describen todas y cada una de las etapas que se mostraron en la figura 1.1, realizadas para llevar a cabo una tarea con un robot móvil tele-operado y la manipulación de objetos a distancia, todo esto mediante un circuito reconfigurable como interface.

Por medio de una cámara tipo CMOS a color, se captura la imagen, la cual es presentada en el monitor. Posteriormente, se hace una búsqueda del objeto en el área de trabajo. Después, el robot es posicionado lo más cerca del objetivo. Esto se hace mediante los comandos del teclado, pasando como controlador el FPGA que va conectado directamente al robot.

4.1 Condiciones iniciales

Antes de proceder a explicar las etapas que componen la metodología propuesta, es necesario establecer las condiciones iniciales:

- El robot se encuentra en un área de trabajo plana y de color distinto al de los objetos.
- Los objetos son de cualquier color, excepto transparentes, y de forma sencilla, como cubos y cilindros.
- El teclado es del tipo estándar y con conexión PS/2.
- El conector DB9 debe estar invertido en sus entradas al conectarlo al robot.
- El monitor debe contar con una entrada RCA.
- Se cuenta con una cámara montada sobre el robot.
- La conexión entre el robot y el FPGA es cableada.

4.2 Etapa de adquisición

Se cuenta con una cámara CMOS de color, la cual está conectada de manera inalámbrica a el receptor de 2.4 GHz. La cámara tiene una resolución de 380 x 450 pixeles. El receptor, está conectado de forma cableada y por RCA a un monitor tipo plano de 19” para visualizar lo que el robot ve en un ángulo de 60 grados. En la figura 4.1 se muestra el receptor utilizado para conectar

inalámblicamente la cámara al monitor de visualización.



Figura 4.1: Torreta de visión cámara CMOS inalámbrico y receptor inalámbrico de audio y video.

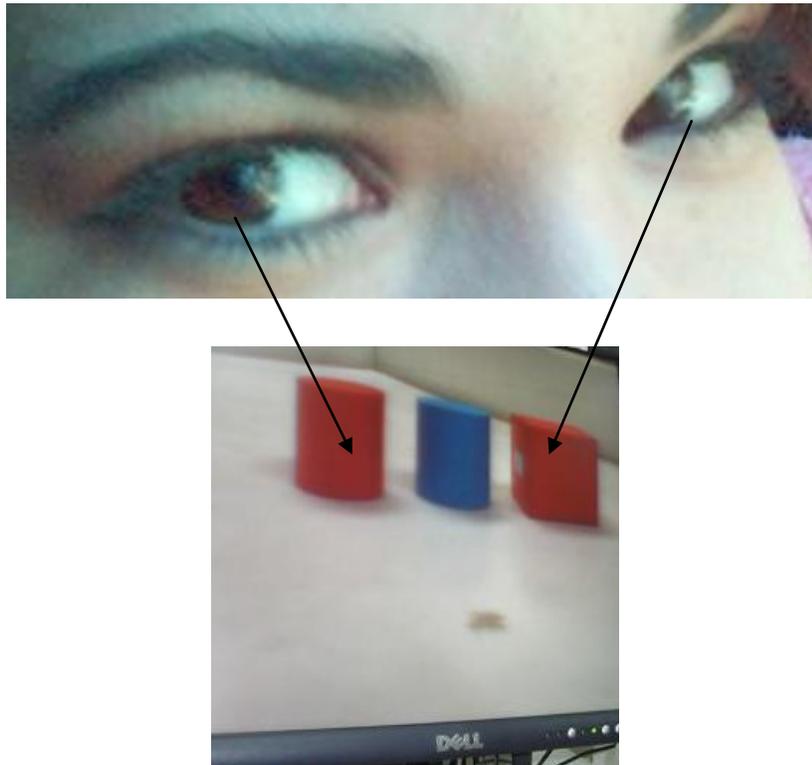


Figura 4.2: Imagen que se recibe desde la cámara observada por el humano.

4.3 Etapa de detección

En esta etapa, se recibe la imagen digital en color para localizar en el área de trabajo los objetos de interés. En la figura 4.2 se muestra una imagen capturada por el robot, donde se tienen 3 objetos frente éste. Esta es la información observada por el usuario y que permite manipular el robot. A través de este monitor es que el usuario tendrá el control del robot. Esto lo hace al analizar la escena observada de acuerdo a la tarea a realizar por el robot mediante su elemento manipulador.

4.4 Etapa de control

Este módulo consta de un FPGA que recibe los datos que le son enviados desde el teclado por el usuario, él cual al observar el monitor hace que el robot se desplace dentro de su espacio de trabajo. Los comandos pasan del teclado a través del FPGA al robot para moverlo y para mover también el manipulador. Si la tarea consiste en manipular un objeto dado, una vez que éste ha sido localizado, el usuario procede a mover el robot para llevarlo a la posición deseada y así realizar la tarea dada.

4.4.1 Entrada de datos

Para lograr esto, se tiene el teclado conectado por el puerto PS/2, como ya se explicó en el capítulo anterior y su esquemático de conexión entre los dispositivos. Con este dispositivo tenemos el control del robot y las pinzas.

Figura 4.3 nos muestra el teclado con el conector PS/2, utilizado para la entrada de datos del FPGA.



Figura 4.3: Teclado estándar de 102 botones con conexión PS/2.

Las teclas a utilizar son las siguientes:

Para el control del movimiento del robot (marcadas en el círculo rojo):



Flecha hacia arriba: este botón nos da un movimiento del robot hacia adelante.



Flecha a la derecha: hace que el robot haga movimiento y empiece a girar hacia la derecha.



Flecha a la izquierda: el robot empieza a girar hacia la izquierda.



Flecha hacia abajo: esta tecla hace que el robot detenga sus motores.

Tecla V: al presionar esta tecla, el robot avanza hacia atrás.

Para el control del brazo manipulador (marcadas por el rectángulo azul):

Tecla A: con esta tecla, las pinzas se suben gradualmente.

Tecla S: con este botón, las pinzas bajan gradualmente.

Tecla D: al presionar esta tecla, las tenazas del robot se cierran.

Tecla F: con este botón, el elemento prensil se abre.

4.4.2 Salida de datos

A la salida de datos del FPGA, se cuenta con el puerto RS232, el cual, se explicó su funcionamiento y manera de conectarlo en el capítulo anterior.

En este puerto, se tiene el envío de datos desde el FPGA al robot, pasando por los convertidores y adaptadores. El robot, solo acepta códigos en ASCII, se empieza con una letra en mayúscula y seguido, si es necesario, de un número o una literal, separados por comas y terminados con un retorno de línea (Enter). El robot, responde con la misma letra que se envió el comando pero en minúscula. En la figura 4.4 se puede ver el módulo de envío de datos inalámbrico, transparente para el usuario, que nos da una comunicación de un conector DB9 a otro igual. También, se puede observar el conector del FPGA DB9 y el PS/2 conectados a sus debidos puertos. En la figura 4.5 podemos observar el receptor conectado al robot Khepera II junto con el cable cruzado y los adaptadores para la conexión de DB9 a DB25.

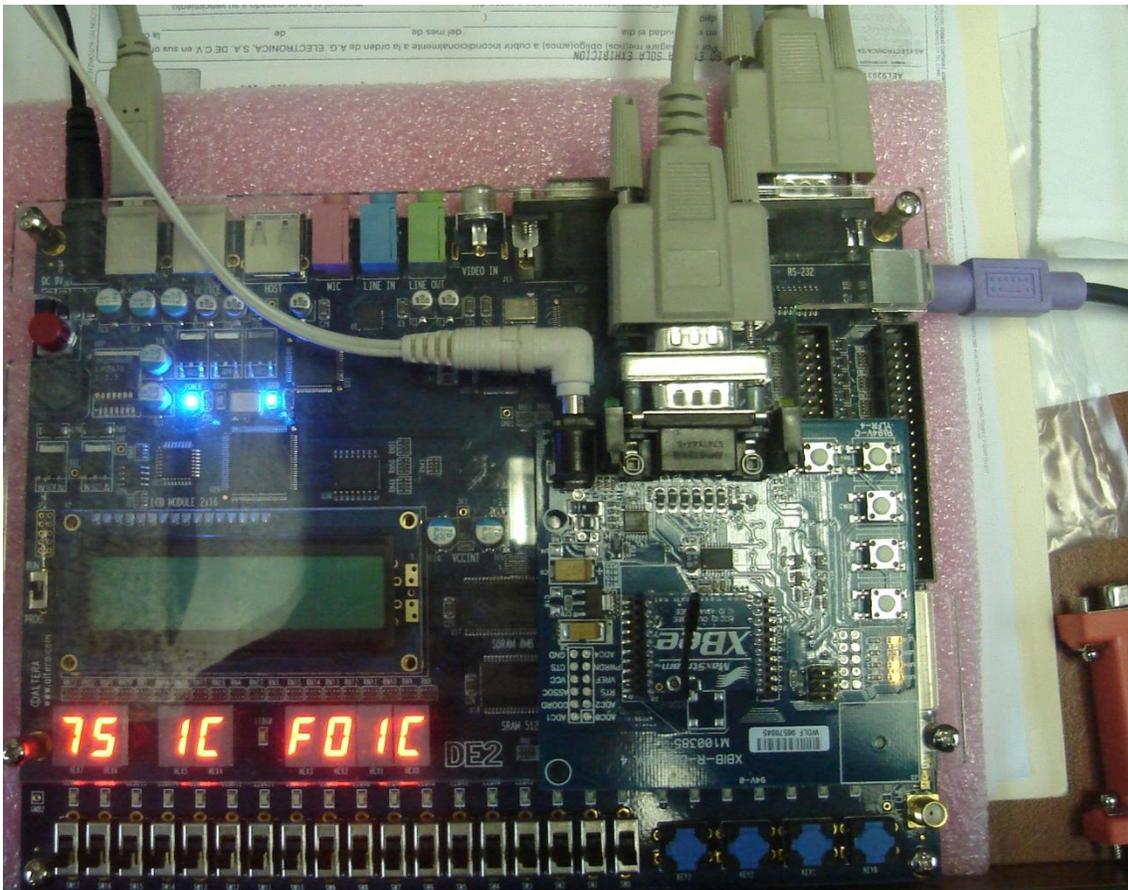


Figura 4.4: El FPGA conectado al transmisor XBee para el envío de los datos por el puerto serie y el puerto PS/2 en color purpura al lado derecho que es la entrada en el circuito reconfigurable del teclado.

Para la comunicación del protocolo de ambos dispositivos, el robot se configuró para trabajar a 9600 bauds de velocidad, que es lo que se puede configurar y cambiar, los demás datos son fijos como lo son: los 8 bits de datos, 1 bit de inicio, 2 bits de parada y sin paridad.

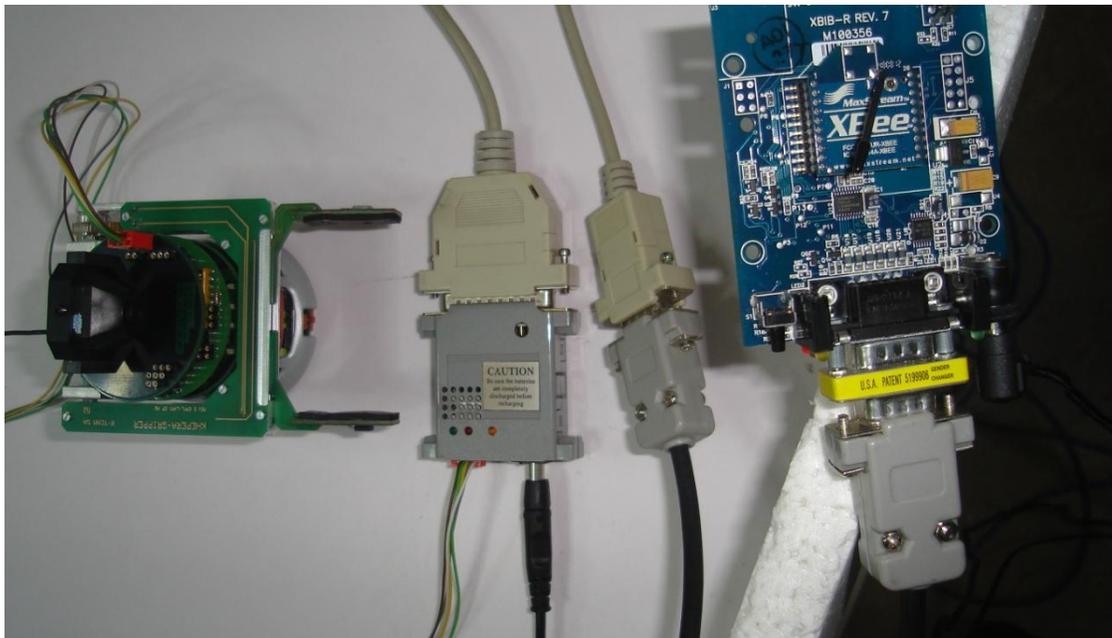


Figura 4.5: De derecha a izquierda. Receptor de datos DB9 conectado al cable cruzado. Cable cruzado conectado al cable adaptador de DB9 a DB25. Conector DB25 conectado al adaptador del robot. Cable de datos conectado directo al robot.

A continuación se muestra una lista de comandos que son usados para el control del robot y las pinzas:

- D,5,5:** este comando hace que el robot se desplace hacia adelante.
- D,0,5:** esta instrucción hace que el robot vaya hacia la izquierda.
- D,5,0:** esta orden manda al robot a dar la vuelta a la derecha.

- D,0,0:** el robot está en estado de parado.
- D,-5,-5:** el robot se desplaza hacia atrás.
- T,1,D,150,150:** esta orden provoca que las pinzas suban hasta la posición 150 dada por el servo motor incrustado en esta torreta.
- T,1,D,3,3:** lleva a las pinzas hacia abajo para tomar el objeto.
- T,1,E,1,1:** hace que el elemento prensil se abra.
- T,1,E,0,0:** hace que el elemento prensil se cierre.

Todas las instrucciones anteriores deben ir seguidas por el retorno de línea. Muchos otros comandos se encuentran inmersos en el micro controlador que trae el robot, pero para este tema de investigación, con los mencionados es suficiente.

En la figura 4.6 se observa el receptor con el robot en el área de trabajo, en esta imagen se ponen juntos, pero en la real no se observa esta zona del área de trabajo.

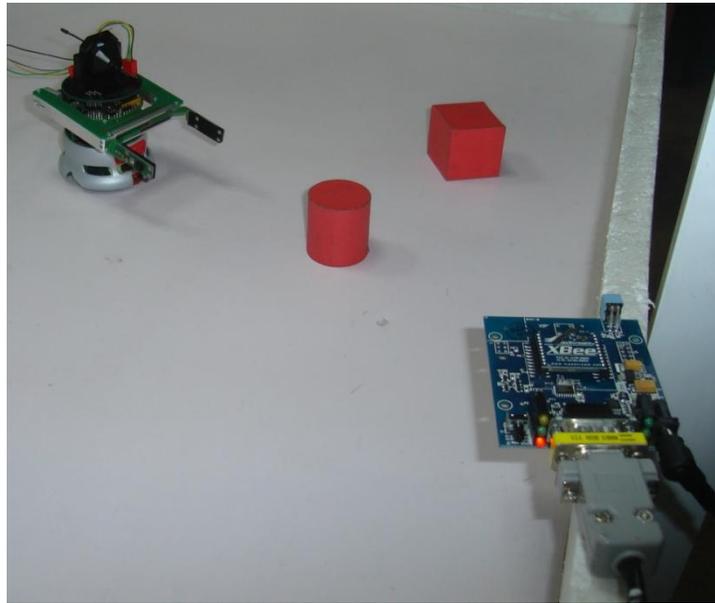


Figura 4.6: En esta imagen se observa el receptor de datos cerca del área de trabajo del robot que se encuentra realizando una tarea.

El FPGA se programó en vectores para enviar la instrucción que está instanciada a cada tecla. Puesto que no todos los comandos tienen la misma longitud, se tuvo que hacer un ajuste en los tamaños de los vectores y tener tres tipos de transmisores de distintas longitudes.

```

64      --8 bits de datos, 1 bit de inicio, 2 bits de parada, no parity
65      tx      <= '1'          when "0000000",
66              '0'          when "0000001", --bit de start
67              dato(40)     when "0000010", --inicia transmision de dato
68              dato(41)     when "0000011",
69              dato(42)     when "0000100",
70              dato(43)     when "0000101",
71              dato(44)     when "0000110",
72              dato(45)     when "0000111",
73              dato(46)     when "0001000",
74              dato(47)     when "0001001", --fin de transmision de dato
75              '1'          when "0001010", --bit de stop
76              '1'          when "0001011",

```

Figura 4.7: Envío de los datos desde el vector.

Como los circuitos electrónicos se toman frecuentemente desde el $n-1$ al 0, se hizo un ajuste de los valores del vector, haciendo que se iniciara por el bit $(n-1) - 7$. También, los dos primeros bits se pusieron para forzar a la entrada de cambio de 1 a 0 y así hacer que la entrada caiga en tensión y mande a 0 para el inicio de envío de datos. En la figura 4.7 tenemos una parte del código para el envío de datos.

Los reportes que nos envía el programa, en este caso, se realizó el sistema en Quartus II, son los siguientes: en la figura 4.8 tenemos el RTL.

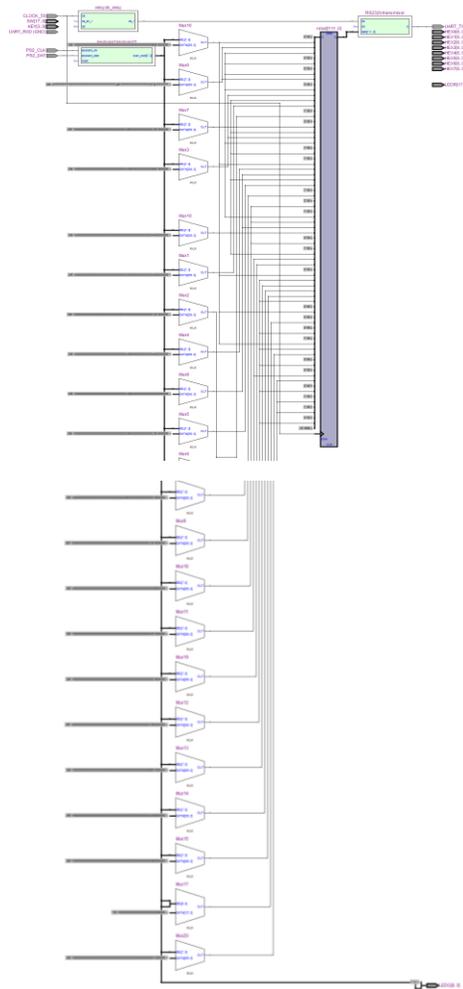


Figura 4.8: El reporte de RTL que nos envía el software Quartus.

En la siguiente imagen tenemos el reporte de Technology Map Viewer que arroja el Quartus II.

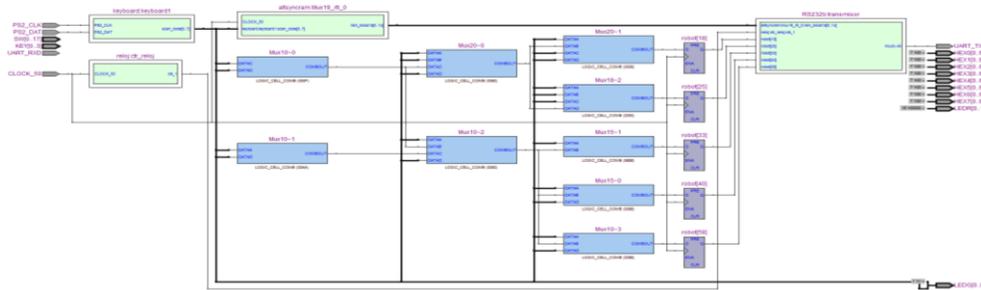


Figura 4.9: Reporte de Technology Map Viewer dado por el Quartus II.

El reporte de análisis y síntesis del Quartus II es el siguiente:

- Total de elementos lógicos: 133
 - ✓ Total de funciones combinatorias: 105
 - ✓ Registros lógicos dedicados: 58
- Total registros: 58
- Total pines: 110
- Total de pines virtuales: 0
- Total de bits de memoria: 3840
- Multiplicadores de 9 bits embebidos: 0
- Total de PLLs: 0

El análisis de tiempos del Quartus II para este sistema es la figura 4.10:

Tipo	Tiempo actual	Desde:	Hacia:	Desde el reloj:	Hacia el reloj:
Worst-case tsu	3.738 ns	PS2_DAT	keyboard: keyboard1 reg[21]	--	PS2_CLK

Worst-case tco	16.344 ns	RS232b:tr ansmisor t tx[3]	UART_T XD	CLOCK_50	--
Worst-case th	-3.508 ns	PS2_DAT	keyboard: keyboard1 reg[21]	--	PS2_CLK
Clock Setup: 'PS2_CLK'	200.40 MHz (period = 4.990 ns)	keyboard: keyboard1 reg[8]	keyboard: keyboard1 scan_cod e[7]	PS2_CLK	PS2_CLK
Clock Setup: 'CLOCK_50'	335.91 MHz (period = 2.977 ns)	reloj:ctr_r elobj cnttx[1]	reloj:ctr_r elobj cnttx[12]	CLOCK_50	CLOCK_50

Figura 4.10: Tiempos arrojados por el sistema, dados por el Quartus

II.

4.5 Resumen

En este capítulo, se describió detalladamente todos y cada uno de los pasos a seguir para llevar a cabo una tarea con un robot móvil manipulador para la transporte de objetos. En el siguiente capítulo, se muestran los resultados obtenidos con dicha metodología.

Capítulo 5

Resultados experimentales

En este capítulo se prueba el desempeño de la metodología descrita en el capítulo 4. Para esto, se posiciona al robot en el área de trabajo, se procede a recibir las imágenes desde la cámara montada sobre el robot para empezarlo a manipular y así desarrollar la tarea deseada. Se conectó el robot al FPGA mediante el uso de un cable cruzado y dos módulos receptor-emisor de Xbee, con el objeto de llevar a cabo la tarea de tele-operación sin mirar directamente el robot y los objetos dentro de su espacio de trabajo.

5.1. Experimento 1: Cubo rojo

Para proceder a tele-operar el robot, el usuario selecciona una tarea y al observar el monitor procede a realizar dicha tarea. En la figura 5.1 se puede observar a través del monitor una parte del área de trabajo donde se encuentra el objeto a modificar. De esta figura se puede ver que el robot tiene en frente de él un cilindro rojo. Si la tarea consistiera en tomar dicho objeto, entonces el

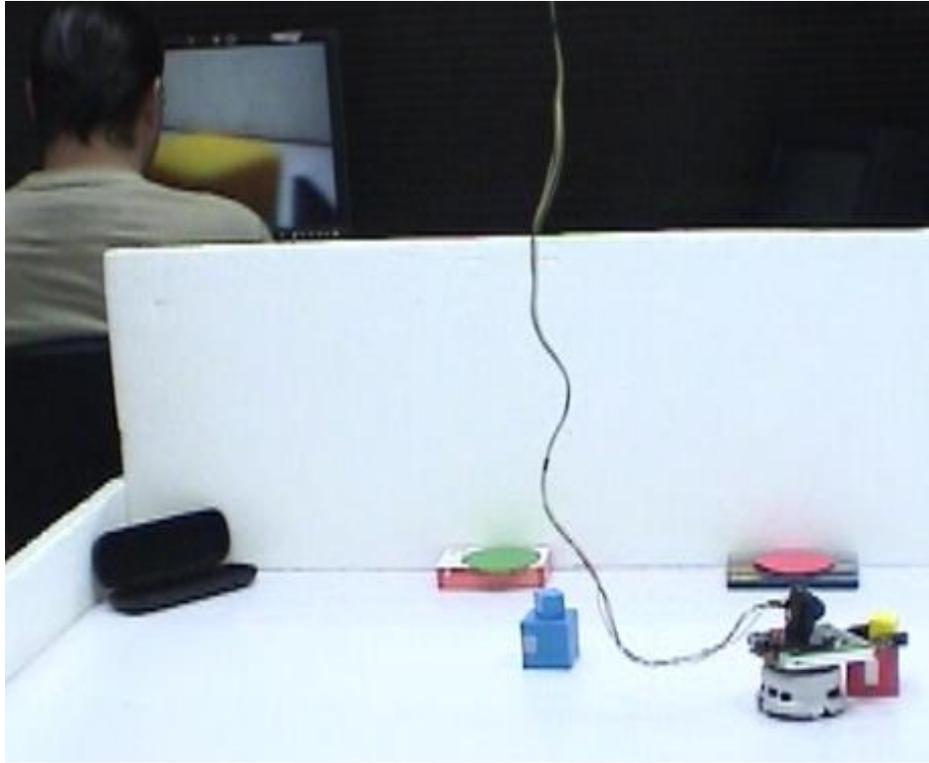
usuario procede a manipular las teclas correspondientes en el teclado para acercar el robot al cilindro; ya estando lo suficientemente cerca de éste, entonces, se acciona el brazo para posicionarlo a una distancia que las pinzas puedan tomarlo.



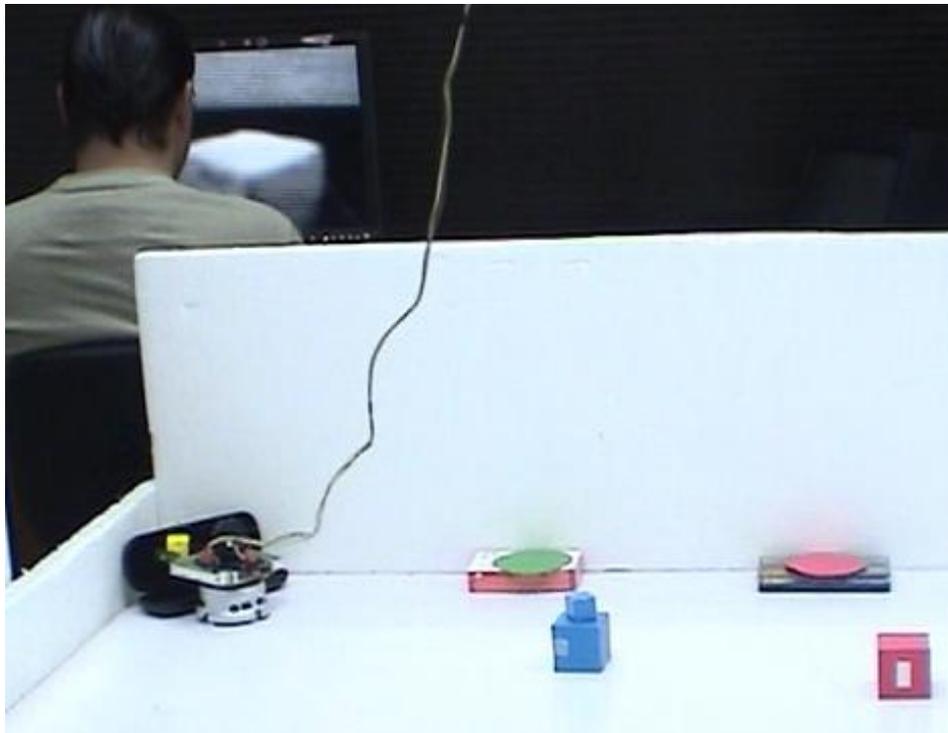
Figura 5.1: En esta figura se muestra el monitor de observación, el usuario. A través de la cámara montada sobre el robot, se observa un cilindro rojo.

5.2. Experimento 2: Agrupar objetos dentro de una caja

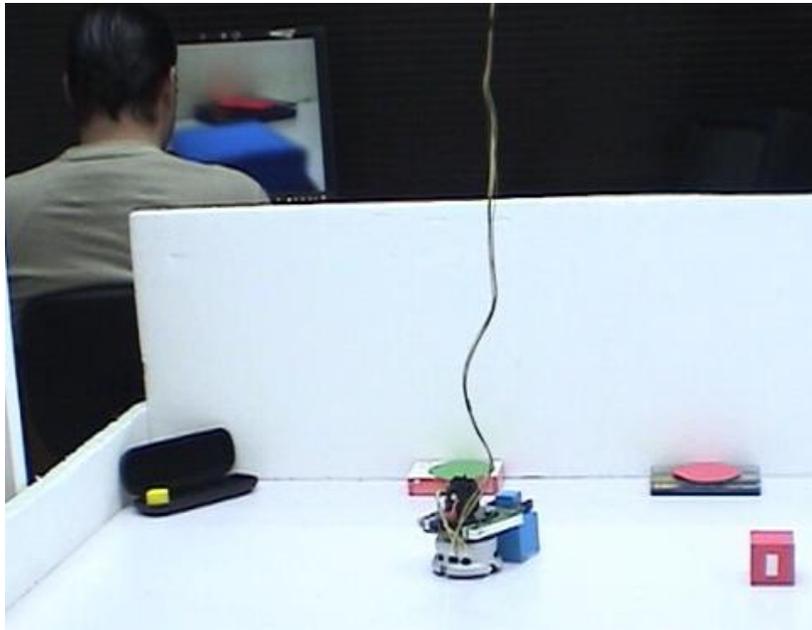
En la figura 5.2(a) se puede observar que ahora el robot está mirando un cubo de color amarillo, para después ponerlo dentro de la caja negra como se ve en la figura 5.2(b). En la figura 5.2(c) se puede ver el cubo azul cuando éste ya está a una distancia lo suficientemente cercana para ser tomado con la pinza del robot.



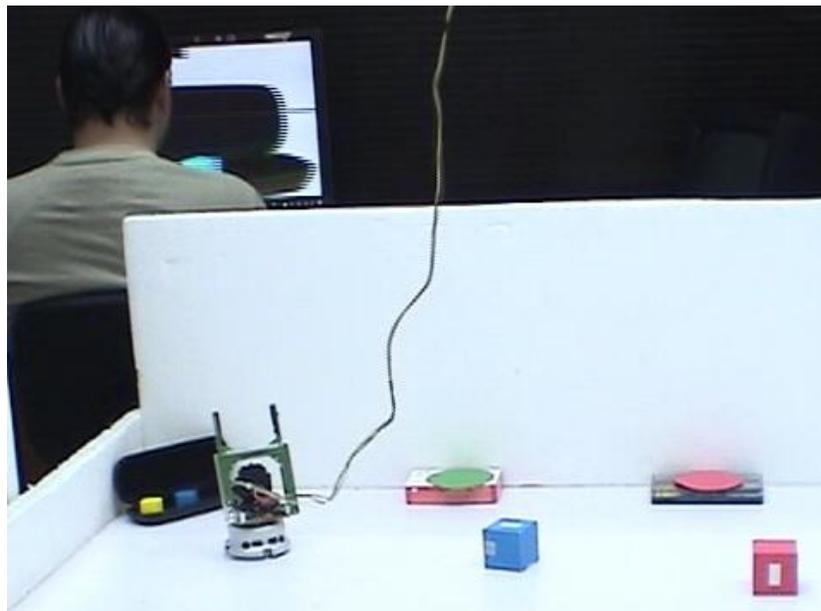
(a)



(b)



(c)

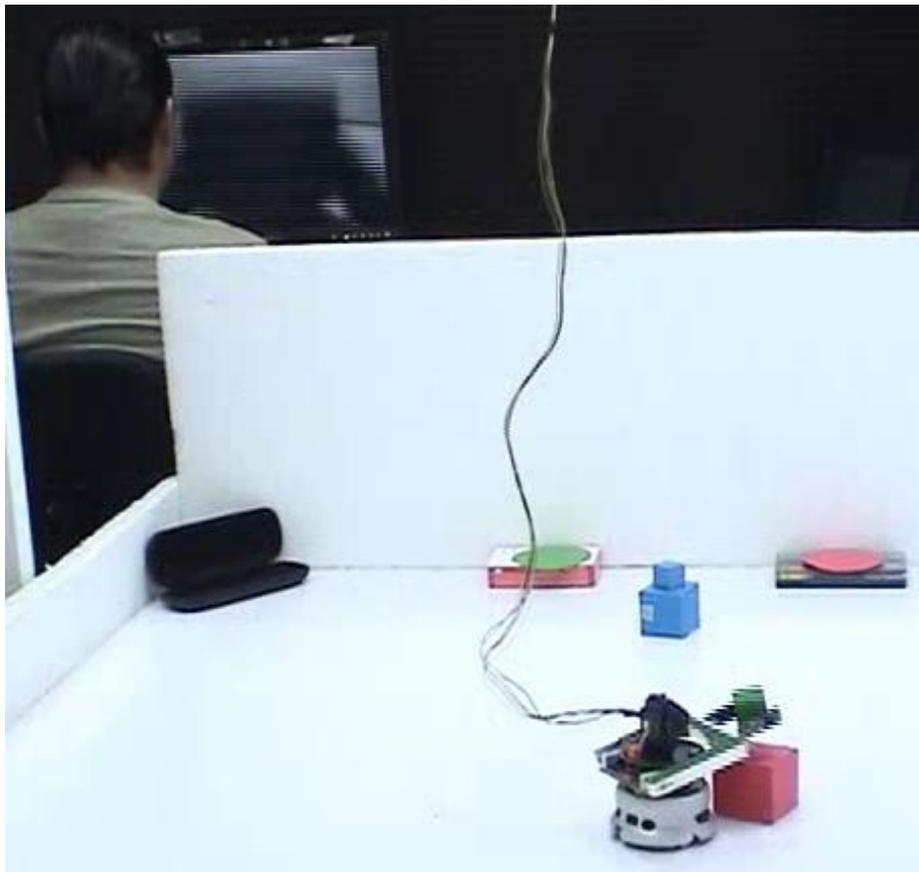


(d)

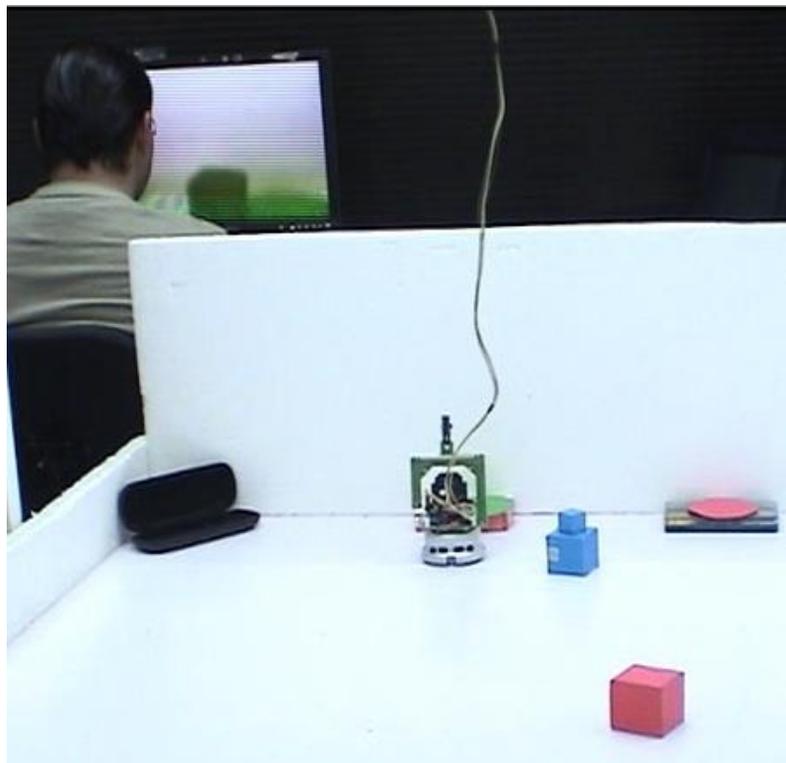
Figura 5.2: (a) Objeto cubo amarillo (b) Cubo amarillo por ser insertado dentro de la caja. (c) cubo azul siendo tomado por el robot (d) cubo azul dentro de la caj. Tarea completada con los dos objetos.

5.3. Experimento 3: Poner objetos sobre una marca

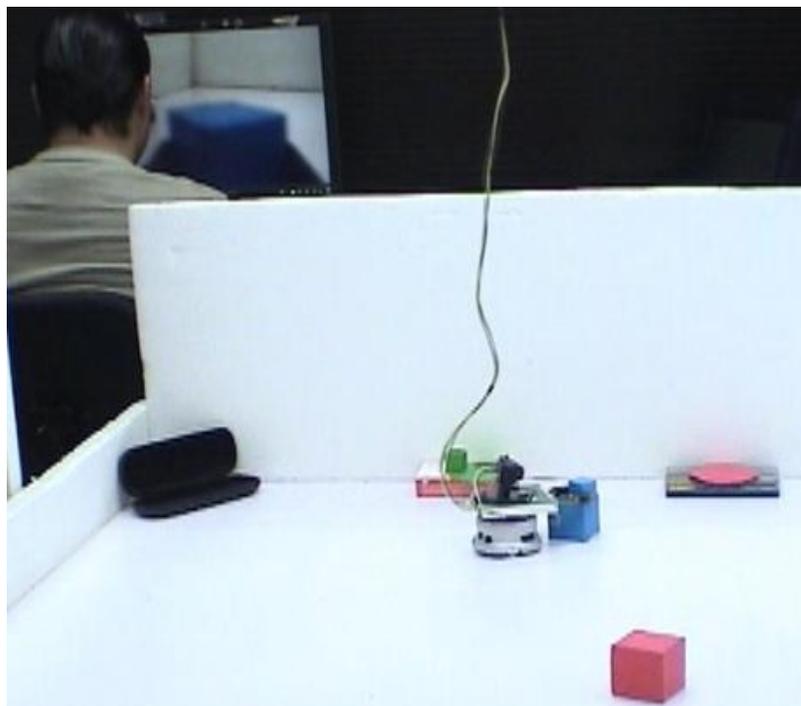
En la figura 5.3(a) se puede ver el cubo verde siendo levantado por el robot. Para la figura 5.3(b) vemos en el monitor el cubo verde sobre la marca verde. En la figura 5.3(c) tenemos al robot tomando el cubo azul para ponerlo sobre la marca roja. En la figura 5.3(d).



a)



b)



c)

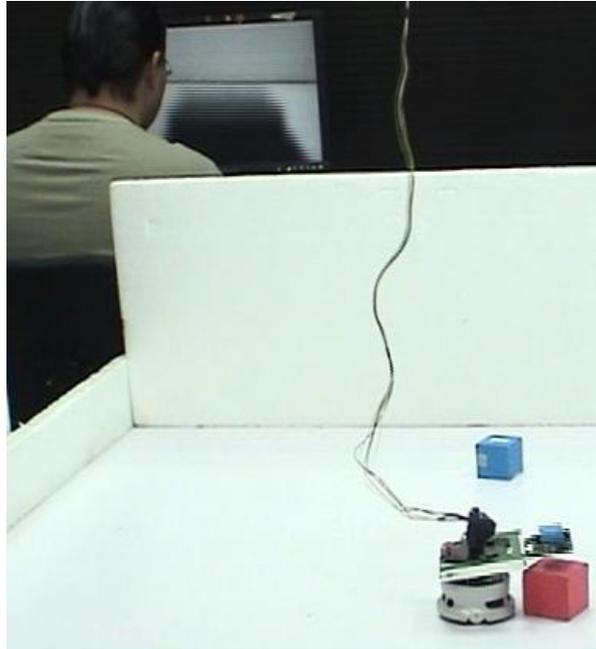


d)

Figura 5.3: (a) El robot levantando el cubo verde. (b) Se observa en el monitor el cubo verde sobre la marca verde. (c) Se tiene al robot tomando el cubo azul. (d) Se observa el cubo azul sobre la marca roja.

5.4. Experimento 4: Cambio de lugar cubo

En este experimento, la tarea consiste en cambiar de lugar el objeto, para este caso, el cubo azul, llevarlo de estar sobre el cubo rojo a estar sobre el cubo azul. Como se ve en la figura 5.4, tenemos dos cubos grandes y uno pequeño, los que nos servirán de base para el objeto que se está manipulando y como se observa, se toma el cubo azul pequeño que esta sobre el rojo para ponerlo sobre el azul grande.



a)

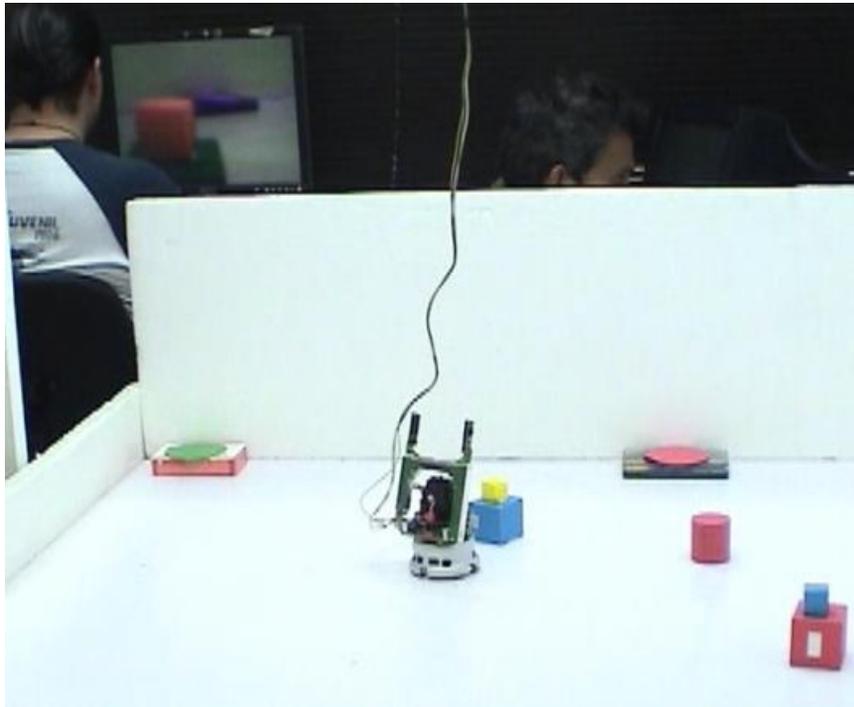


b)

Figura 5.4: (a) El robot está levantando el objeto a manipular. (b) Tenemos al objeto sobre la posición final que se desea; por lo tanto la tarea esta completada.

5.5. Experimento 5: Intercambiar lugares

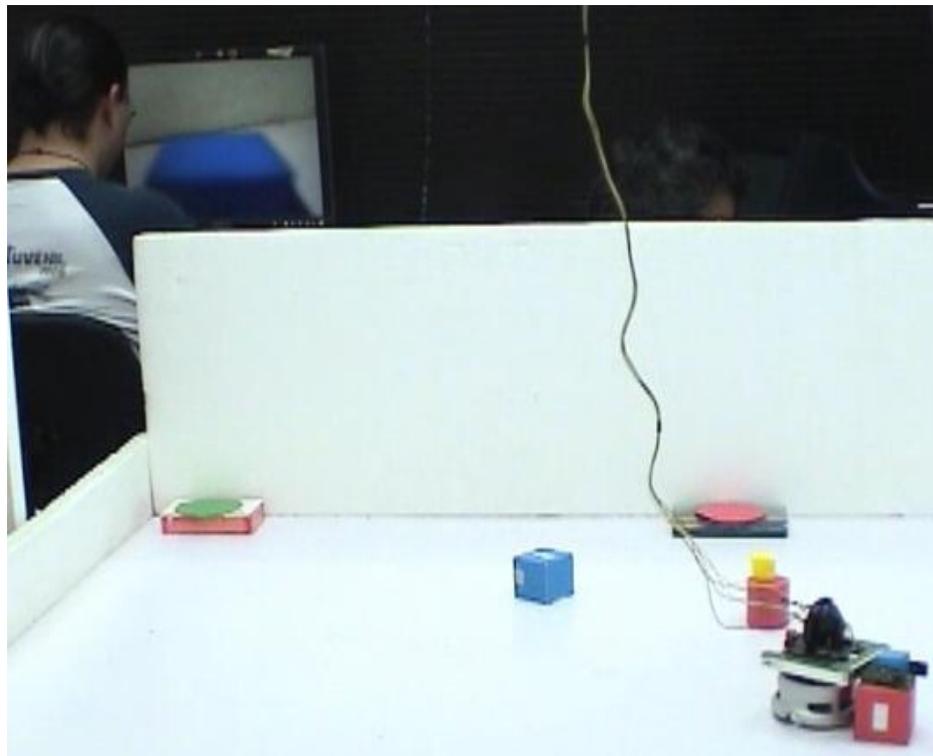
En la figura 5.5(a) tenemos como tarea realizar el intercambio de lugar de los objetos y observamos dos cubos, uno amarillo y otro azul. Necesitaremos un lugar temporal para posicionar el primer objeto para poder usar ese lugar después con el siguiente objeto. En la figura 5.5(b) está posicionado el cubo amarillo en el lugar temporal, en este caso, en el cilindro rojo. Para la figura 5.5(c) tenemos al robot tomando el cubo azul pequeño que esta sobre el cubo rojo, para después ponerlo sobre el cubo azul grande, como se observa en la figura 5.5(d). Para el último movimiento, se pone el cubo amarillo pequeño sobre el cubo rojo grande y se muestra en la figura 5.5(e).



a)



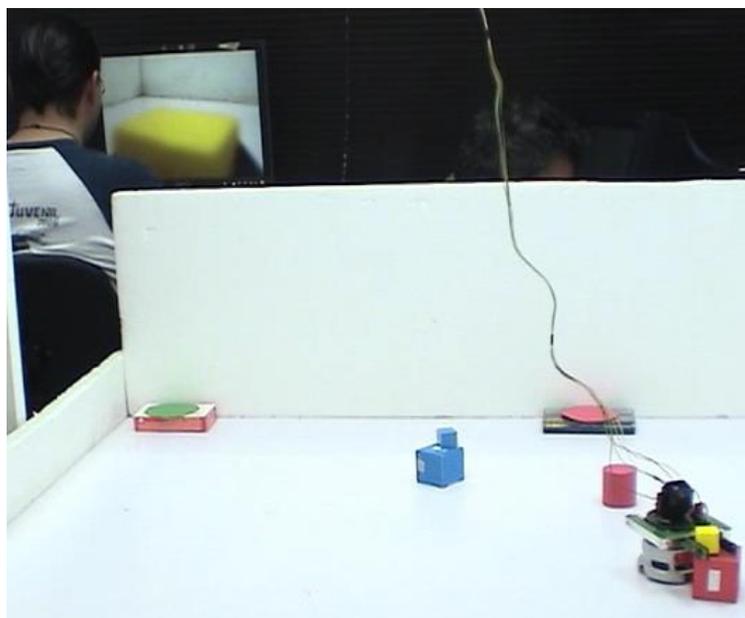
b)



c)



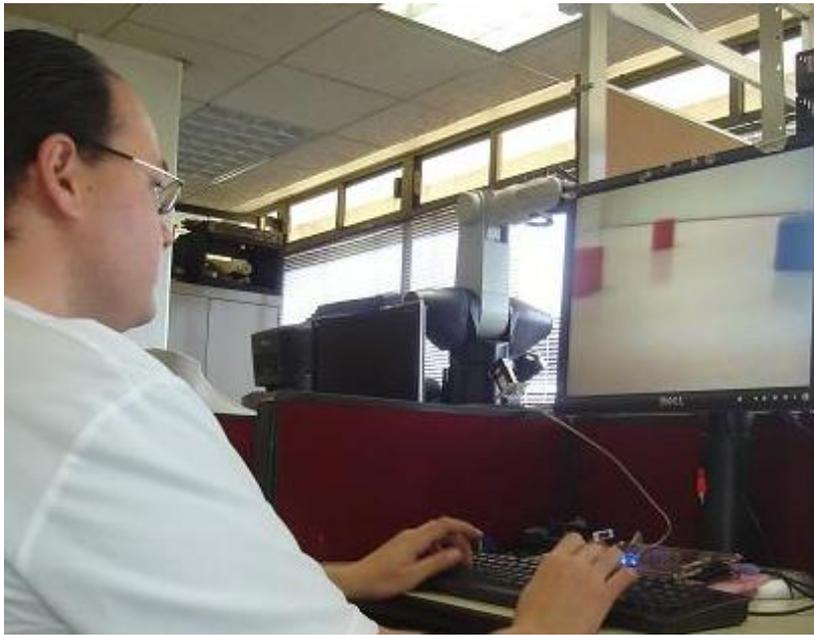
d)



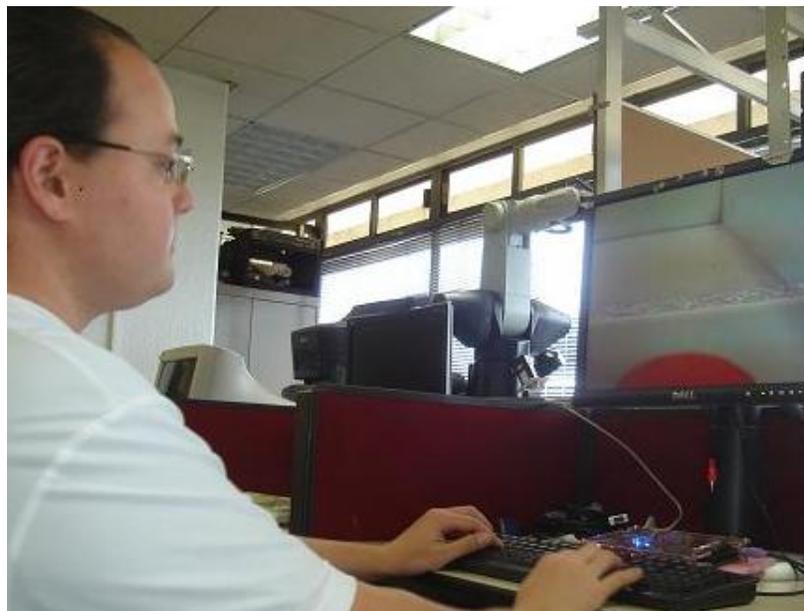
e)

Figura 5.5: en la figura (a) tenemos al robot aproximándose a tomar el cubo amarillo. En la (b) tenemos el objeto puesto en la posición temporal. (c) El robot tomando el segundo objeto. En (d) se posicionó el segundo objeto sobre el primero y para el último, tenemos el segundo objeto cambiarlo de la posición temporal, llevarlo a la final.

En la siguiente figura, vemos al usuario manipulando al robot hasta acercarlo al cilindro rojo para tomarlo.



a)



b)

Figura 5.6: (a) El usuario moviendo el robot hacia el cilindro rojo. (b) El usuario ha logrado tele-operar el robot para situarlo lo suficientemente cerca del objeto a manipular.

En la siguiente imagen, figura 5.7, se muestra una imagen del robot dentro de su área de trabajo, donde se encuentran varios objetos. En este caso el usuario está desplazando el robot para acercarlo al cilindro rojo a ser manipulado.

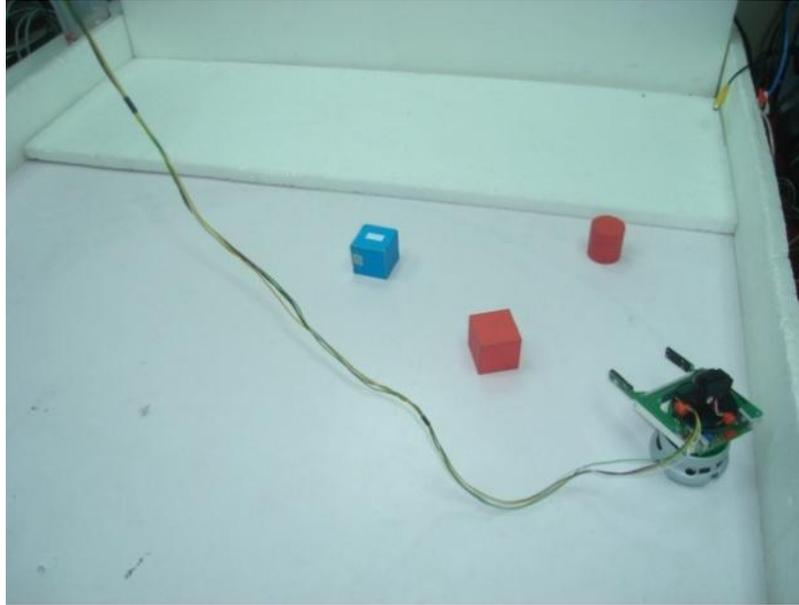


Figura 5.7: Robot Khepera II en el área de trabajo.

5.6. Resumen

En este apartado pudimos observar las diferentes tareas que se le asignaron por resolver al robot por medio de la tele-operación. En todos los casos la tarea seleccionada fue realizada exitosamente.

Capítulo 6

Conclusiones y trabajos futuros

En este capítulo se dan las conclusiones a las que se llegó después de haber realizado la presente investigación. Por otro lado, se enuncian algunas posibles directivas para futuras investigaciones.

6.1 Conclusiones

En esta tesis se mostró una metodología para tele-controlar un robot Khepera II y poder manipular objetos a distancia, para lo cual, se hizo un control supervisado usando como base el FPGA y conectándolo usando el teclado y como salida el puerto serie para comunicación con el robot.

Tomando todo lo anterior, tenemos un robot capaz de mover de lugar objetos de geometría sencilla y operación a distancia. Todo esto, gracias a distintas ramas del saber, como lo son los circuitos reconfigurables, la

electrónica, la robótica, el control y estabilización, los lenguajes de descripción de hardware. Ya que todo el sistema fue realizado sobre un FPGA, tenemos un mucho menor tiempo de respuesta comparado con una PC. Con esto, también probamos que con el uso de un circuito externo, podemos tener el control de este robot.

Gracias a los módulos de comunicación de esta tesis, se tiene que este mismo FPGA puede ser usada para diversos robots y diversas tareas, ya que, muchos robot, utilizan esta interfaz para comunicarse, por lo que se puede usar en otros dispositivos en el mercado.

Cuando se trabaja en tiempo real, es importante saber elegir la arquitectura, el lenguaje y sistema a utilizar. De lo contrario, la programación se puede volver muy complicada y engorrosa.

6.2 Trabajos futuros

Entre los trabajos futuros a desarrollar se pueden mencionar los siguientes:

1. Poner en operación el módulo de captura, transmisión y análisis automático de imágenes del robot al monitor.
2. Implementar el módulo anterior pero con visión sobre el área de trabajo con la cámara que cuenta el circuito reconfigurable.
3. Conectar de manera inalámbrica el circuito al robot, esto sería, por bluetooth.
4. Implementar las leyes de control que permitirán controlar al robot de

manera automática.

5. Desarrollar los módulos de control y visión para el FPGA que puede ser montado sobre el robot.
6. Conectar el FPGA a internet y llevar a cabo la recepción de datos desde la red.
7. Conectar un joystick tipo palanca para tener un mayor confort con respecto a los movimientos del robot y las pinzas.

6.3 Conclusión final

El surgimiento de nuevas tecnologías en la integración a gran escala de circuitos debe involucrar un arduo trabajo de investigación por parte de los fabricantes. El funcionamiento de tales dispositivos puede no coincidir con lo estipulado en la literatura que lo acompaña, por lo que, finalmente, se debe tener en mente “No todo se encuentra en los libros” ó manuales y que buena parte del éxito en el desarrollo de proyectos relacionados con nuevas tecnologías, radican en la capacidad del investigador por lidiar con esos pequeños caprichos de la electrónica.

Referencias

- [1] N.J. Nilsson (1969). A Mobile Automation: An Application of Artificial Intelligent Techniques, *Proc. IJCAI*, Washington D.C.
- [2] Alberto Elías Petrilli Barceló (2007). Control visual de un robot móvil Khepera II. Tesis de Maestría, CIC – IPN.
- [3] Corina Adrinana Sandoval García (2006). Sistema de programación remoto de un brazo robot mediante lógica programable vía internet. Tesis de Maestría, CIC – IPN.
- [4] César Vázquez Cianca (2009). Manipulación robotizada de objetos basada en visión por computadora. Tesis de Maestría, CIC – IPN.
- [5] Heriberto Casarrubias Vargas (2007). Generación de trayectorias para un robot móvil Khepera II usando técnicas de aprendizaje automático.

-
- [6] K. Han and M. Veloso (1998). *Reactive visual control of multiple non-holonomic robotic agents*, In Proceedings of the International Conference on Robotics and Automation, Leuven, Belgica.
- [7] W. G. Walter (1993). *The Living Brain*, Duckworth, London.
- [8] Emad Monier, Ulf Witkowski, and Andry Tanoto (2006). Soccer Teleworkbench for Development and Analysis of Robot Soccer, Paderborn, Germany.
- [9] Oscar Mauricio Ayala and Javier Gonzalez (2006). Desarrollo de un Robot Móvil con control inteligente usando Algoritmos Evolutivos y un FPGA, Meta, Colombia.
- [10] R.C. González K.S. Fu and Lee (1986). *Robotics control, Sensing, Vision and Intelligence*. McGraw Hill.
- [11] John J. Craig (2006). *Robótica*. Pearson, Prentice Hall.
- [12] Mario Aldape Pérez (2007). Implementación de los modelos ALFA-BETA con lógica reconfigurable. Tesis de Maestría, CIC – IPN.
- [13] Aníbal Ollero Baturone (2007) *ROBÓTICA: Manipuladores y robots móviles*. Alfaomega – Marcombo. ISBN: 987-84-267-1313-1.

- [14] Antonio Barrientos, Luis Felipe Peñín, Carlos Balaguer, Rafael Aracil (2007). *FUNDAMENTOS DE ROBÓTICA*. Mc Graw Hill. ISBN: 978-84-481-5636-7.
- [15] David G. Alciatore, Michael B. Hstand (2008). *Introducción a la MECATRÓNICA y los sistemas de medición*. Mc Graw Hill. ISBN: 978-970-10-6385-9.
- [16] Cuauhtémoc Peredo Macías (2001). Integración de un algoritmo de control difuso en un FPGA para control de nivel de líquido.

Apéndice A

Equipo empleado

En este apartado se describe el equipo empleado para el desarrollo de esta tesis. En la figura A.1 se muestra el espacio de trabajo usado en los experimentos realizados en esta tesis. A continuación se detalla el equipo empleado:

- Computadora.
- Monitor.
- Robot Khepera II.
- Torre con brazo manipulador y pinza para robot Khepera II.
- Torre con cámara a color para robot Khepera II.
- Mesa de trabajo.

- FPGA con puertos PS/2 y RS-232.
- Cable DB-9 cruzado.
- Teclado PS/2.
- Cable USB-A a DB9.
- Cable DB9 a DB25.

Las especificaciones del robot Khepera II se presentan en el apéndice B. Las especificaciones del FPGA en el C y del equipo restante se dan a continuación.

Computadora:

- Dell Optiplex GX620.
- Sistema operativo Windows XP profesional con Service Pack 3.
- Procesador Pentium 4ht a 2.8 Ghz.
- Memoria RAM 4 Gb.
- 16Mb de video integrado.
- Puerto RS-232.

Monitor:

- 19" a color.
- Resolución mínima: 1024 x 768.
- Entrada RCA.

- Entrada VGA.

Torreta tipo pinza para el robot Khepera II:

Este módulo de extensión del Khepera II permite la manipulación de objetos mediante sus tenazas, con las cuales, puede golpear, tomar y levantar un objeto. A continuación, algunas características de la torreta:

- Movimiento vertical de 200 grados.
- Apertura de la pinza de 5 cm.
- Levanta objetos de hasta 100 gr.

Torreta tipo cámara a color para el robot Khepera II:

Esta extensión del Khepera II, permite al usuario de manera inalámbrica, ver que es lo que tiene el robot enfrente o dentro de su ángulo de captura. A continuación se dan sus características:

- Comunicación vía inalámbrica con su receptor a 2.4 Ghz.
- Salida del receptor tipo RCA.
- 4 canales de comunicación.
- Toma corriente de 9V.

Mesa de trabajo:

Sobre la mesa de trabajo se montó una pista para la realización de los experimentos presentados en este trabajo. El material utilizado para la superficie de la pista es una lámina plastificada de 2 mm de grosor en color

blanco con dimensiones de 1.0 por 1.3 m. Para confinar al robot dentro de esta área de trabajo, se utilizaron paredes de unicel de 10 cm de altura.

FPGA con puertos RS-232 y PS/2:

Para el hardware se cuenta con un FPGA de Altera modelo 60 tipo DE-2 con los puertos PS/2 y RS-232 para comunicación con el robot y entrada de comandos.

Puerto RS-232:

Con este puerto se tiene la comunicación serial con el robot. El puerto es tipo DB-9, de este puerto se usan los siguientes pines:

- TX, para la transmisión de datos.
- RX, para la recepción de datos.
- GND, para la conexión a tierra.

Puerto PS/2:

Este protocolo de comunicación está hecho para conectar tanto teclado como ratón. A éste se conecta un teclado de 102 botones.

Apéndice B

Características del robot Khepera II

En este apéndice se dan las características más importantes del robot móvil Khepera II, utilizado para probar los métodos descritos en esta tesis.

El robot Khepera II es un robot miniatura desarrollado en el Instituto Federal Suizo de Tecnología en Lausanne con funcionalidades similares a los de robots grandes usados en investigación y educación. El dispositivo permite probar algoritmos en mundos reales desarrollados para planeación de trayectorias, evasión de obstáculos, entre otros. El robot Khepera II viene equipado con varias torretas que le permiten adaptarse a diversos escenarios. En la figura B.1, se muestra el robot Khepera II con la

torreta de manipulación y la cámara inalámbrica.

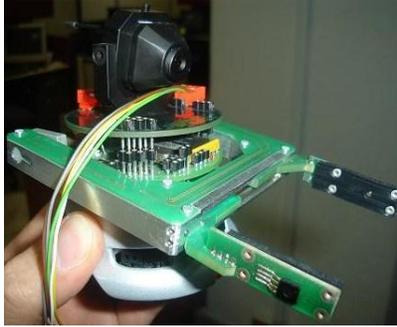


Figura B.1: Robot móvil Khepera II con pinzas y cámara.

B.1. Locomoción del robot Khepera II

El robot Khepera II se mueve mediante un par de ruedas que se controlan de manera independiente, este método es conocido como conducción diferencial. Cada una de las ruedas del robot Khepera II es impulsada por un motor DC acoplado con la rueda a través de una caja de reductora de 25:1. Un codificador incremental colocado sobre el eje del motor da 24 pulsos por revolución del motor. Esto permite una resolución de 600 pulsos por revolución de la rueda, que corresponde a 12 pulsos por milímetro de camino recorrido del robot.

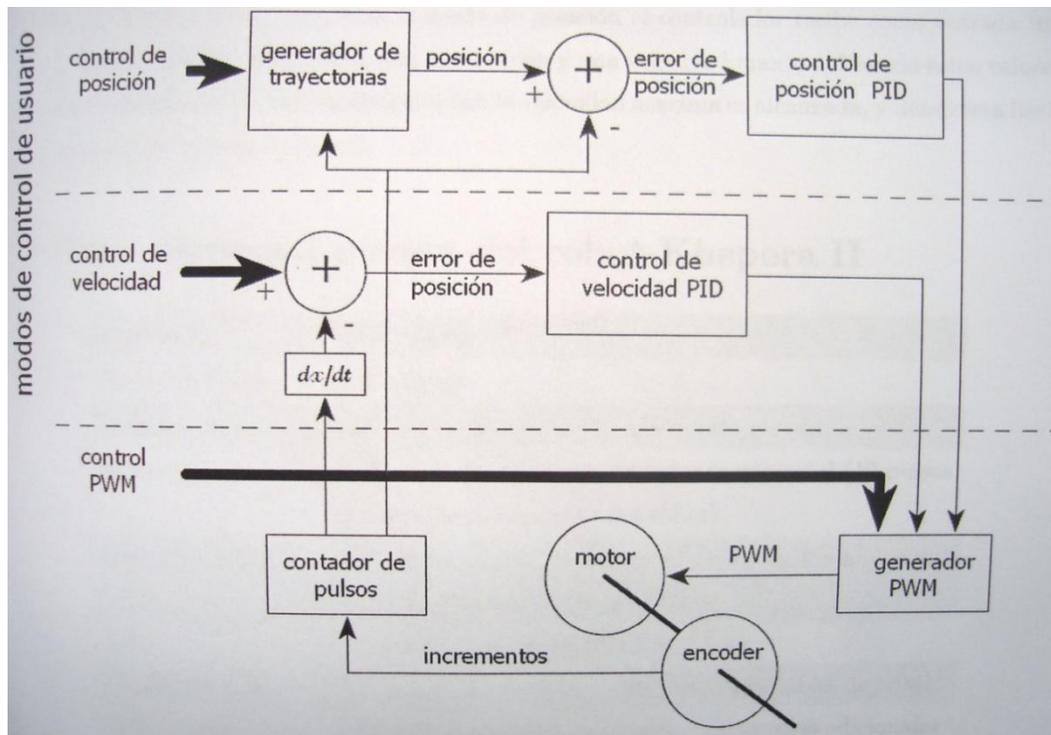


Figura B.2: Control de velocidad de las ruedas del robot Khepera II

El procesador principal del robot Khepera II tiene el control directo sobre los motores y su potencia es variada mediante modulación de ancho de pulso (PWM). Ambos motores son controlados por un controlador PID, que se ejecuta en una rutina de interrupción en el procesador del robot, ver figura B.2. El controlador del robot puede ser usado en dos modos de control:

1. Control por velocidad.
2. Control por posición.

El modo activo es de acuerdo al comando que se pide ejecutar. Al usar el control en modo velocidad, se recibe un valor correspondiente a la velocidad de las ruedas y el control del robot lleva las ruedas a esta velocidad tan rápido como le sea posible. Al usar el método de posición el controlador recibe como entrada una posición objetivo de la rueda, una aceleración y una velocidad

máxima. Al usar estos valores, el controlador acelera la rueda hasta que la velocidad máxima es alcanzada, y desacelera hasta alcanzar la posición objetivo.

B.2 Especificaciones del robot Khepera II

Procesador:	Motorola 68331, 25 MHz
Memoria RAM:	512 KBytes
Memoria flash:	512KBytes, programables vía puerto serial
Motores:	2 DC servo motores con codificador incremental (12 pulsos por mm de movimiento del robot)
Sensores:	8 sensores de proximidad infra rojo y de luz ambiental de hasta 100 mm de rango
I/O:	3 entradas analógicas (0-4.3V, 8 bits)
Fuente de energía:	Adaptador de corriente, o baterías recargables de NiHM
Autonomía:	1 hora de movimiento constante. Las torretas adicionales reducen la duración de la batería
Comunicación:	Hasta 115kbps vía puerto serial estándar
Extensión de bus:	Pueden ser añadidos módulos de expansión usando el <i>K.-Extension bus</i>
Dimensiones:	70mm x 30mm (aproximadamente)
Peso:	80 hrs (aproximadamente)

B.3 Especificaciones de la cámara del robot Khepera II

Sensores:	Cámara a color tipo CMOS (omnivision)
Resolución de la imagen:	380 x 450
Frecuencia máxima de escaneo:	50 Hz
Distancia máxima:	50 a 100 m
Salida de poder:	10 mW
Iluminación mínima:	1 LUX
Extensión del bus:	Ninguna otra torreta se puede poner encima de la cámara, por lo tanto, todas las demás deben estar debajo de ésta.
Tamaño:	Diámetro: 55mm Altura: 38 mm

Apéndice C

Características del FPGA Altera DE-2

En este apartado se describen las especificaciones del FPGA modelo Cyclone II serie 2C35.

Cyclone II 2C35:

- ❖ Contiene 35000 elementos lógicos.
- ❖ 475 entradas/salidas para el usuario.
- ❖ 104 bloques M4K y 483 Kbits de SRAM.
- ❖ 35 multiplicadores embebidos y 4 PLLs.

Configuración serial (EPCS16) y circuito USB Blaster:

- ❖ USB Blaster embebido para programación y controlador API para el

usuario.

- ❖ Modo JTAG y AS soportados.
- ❖ Configuración serial de EPCS16.

8 Mbyte SDRAM:

- ❖ Organizada en 1M x 4 x 16.
- ❖ Soporta el acceso de NIOS II y el controlador Terasic multi-puerto de alta velocidad.

1 Mbyte de memoria flash (ampliable hasta 4):

- ❖ Equipado con 1 Mbyte de memoria flash NAND.
- ❖ Las capas están diseñadas para soportar 4 Mbyte de SDRAM.
- ❖ 8 bits de bus de datos.

Entrada para tarjeta SD:

- ❖ Acceso a tarjetas en modo SPI.
- ❖ Soporta acceso de NIOS II con el driver de la SD de Terasic.

Botones para presionar:

- ❖ 4 botones de presión.
- ❖ Normalmente en alta y genera una señal de baja cuando es presionado.

DPDT switches:

- ❖ Contiene 18 switches para la introducción del usuario.

Relojes:

- ❖ 1 oscilador de 50 MHz.
- ❖ 1 reloj de 27 MHz.
- ❖ Contiene una conexión SMA para conectar un reloj externo.

Codificador de audio:

- ❖ Utiliza Wolfson WM 8731 de 24 bits.
- ❖ Trae línea de entrada, línea de salida y micrófono.
- ❖ Frecuencia de operación: 8 KHz a 96 KHz.
- ❖ Aplicaciones para reproductores MP3, PDA y teléfonos inteligentes.

Salida XSGA DAC:

- ❖ Utiliza ADI 7123 a 240 MHz triple a 10 bits video DAC de alta velocidad.
- ❖ Con conector de 15 pines D-sub.
- ❖ Soporta hasta 1600 x 1200 a 100 MHz.
- ❖ Puede ser empleado para implementar un codificador de TV.

Circuito decodificador de NTSC/PAL:

- ❖ Usa ADI 7181B multiformato decodificador de video SDTV.

- ❖ Soporta NTSC-M,J,4.43, PAL-D,B,G,H,I,M,N, SECAM.
- ❖ Integra tres 54MHz ADCs de 9 bits.
- ❖ Entrada de oscilador de 27 MHz.
- ❖ Múltiples formatos de entrada analógica: Video compuesto (CVBS), S-Video(Y/C) y componentes YPrPb.
- ❖ Aplicaciones en: grabadores de DVD, televisores LCD, televisión digital, dispositivos portátiles.

Controlador 10/100 ethernet:

- ❖ MAC y PHY interfaz integrada.
- ❖ Soporta aplicaciones 10Base-T y 100Base-T.
- ❖ Operaciones full dúplex a 10 y 100 Mb/s con auto MDIX.
- ❖ Compatible con las especificaciones 802.3u de IEEE.
- ❖ Soporta generación de checksum IP/TCP/UDP.

Controlador USB Esclavo/Maestro:

- ❖ Cumple completamente las especificaciones del Universal Serial Bus Rev. 2.0.
- ❖ Soporta transferencia de datos en alta y en baja velocidad.
- ❖ Soporta ambos tipos USB maestro y esclavo.
- ❖ Soporta dos puertos USB (un tipo A para maestro y un tipo B para

dispositivos conectados a la DE2).

- ❖ Provee una interfaz paralela de alta velocidad para gran mayoría de los CPUs.
- ❖ Soporta programación de entrada/salida (PIO) o Acceso directo a Memoria (DMA).

Puerto serial:

- ❖ **Provee de dos puertos seriales: un RS-232 y otro PS/2.**
- ❖ **Tiene un conector serial tipo DB-9 para el puerto RS-232.**
- ❖ **Trae conector PS/2 para conectar un ratón o un teclado a la tarjeta DE2.**

Transceptor IrDA:

- ❖ Contiene un transceptor de 115.2 Kb/s.
- ❖ LED de 32 mA.
- ❖ EMI blindado integrado.
- ❖ Ojos seguros clase 1 tipo IEC825-1.
- ❖ Detector de entradas de filo.

Dos cabezas de expansión de 40 pines con diodos de protección:

- ❖ Un total de 72 Cyclone II entradas/salidas en extensión de dos conectores de 40 pines.

- ❖ La cabeza de 40 pines fue diseñada para aceptar el cable estándar de 40 pines usado para el estándar de dispositivos IDE.

Configuración del dispositivo y circuito USB Blaster:

- ❖ Circuito USB Blaster embebido en la tarjeta.
- ❖ Provee los modos de programación JTAG y AS.
- ❖ Contiene 16 Mbit (EPCS16) para configuración del dispositivo.

Glosario de términos

Arreglo de compuertas programables (Field Programmable Gate Array):	Dispositivo lógico programable cuya arquitectura es de alta densidad con múltiples niveles de lógica e interconexión programable.
Cámara CMOS:	Cámara a color con omnivisión.
Configuración:	Un vector de parámetros independientes, los cuales definen de manera única la posición y dirección del robot.
Dispositivo lógico programable (Programmable Logic Device, PLD):	Término genérico para un dispositivo lógico, cuya función puede ser configurada por el usuario.
Entorno:	Conjunto de condiciones extrínsecas que necesita un sistema informático para funcionar, como el tipo de programación, de proceso, las características de la máquina que lo componen, etc.
Entorno estático:	Condiciones físicas de un entorno que no se modifican con el paso del tiempo.

Grado de libertad:	Cada uno de los movimientos (desplazamiento y/o rotación) que se pueden realizar con un robot.
Imagen:	Es una representación visual de un objeto a través de diferentes técnicas, por ejemplo, fotografía, video, etc.
Manipulador:	Mecanismo formado por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Puede ser gobernado por un operador humano o un dispositivo lógico.
Motor DC:	Motor de corriente continua que convierte la energía eléctrica en mecánica mediante movimiento rotativo.
Protocolo:	Conjunto formal de acuerdos que gobiernan el formato y temporización de mensajes.
Receptor-transmisor universal asíncrono (Universal Asynchronous Receiver-Transceiver):	Un dispositivo, generalmente un circuito integrado, que convierte del formato en paralelo a serie al transmitir y de serie a paralelo al recibir; también agrega bits de inicio, paro, paridad y comprueba errores.
Robot:	Manipulador automático servocontrolado, reprogramable, capaz de posicionar y orientar piezas, útiles o

dispositivos especiales, siguiendo trayectorias variables reprogramables, para la ejecución de diversas tareas. Su unidad de control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno.

- Robot diferencial:** Robot cuyo medio de locomoción consta de dos ruedas, las cuales son controladas de forma independiente.
- Robot no holonómico:** Es aquel que tiene más grados de libertad que grados de control.
- Robot omnidireccional:** Robot con grados de libertad igual al de parámetros de control; este tipo de robot puede moverse en cualquier dirección.
- Sensor:** Es un dispositivo que censa manifestaciones de cualidades o fenómenos físicos como la energía, velocidad, tamaño, cantidad, etc.
- Sistema de control en lazo cerrado:** Es un sistema que mantiene una relación determinada entre la salida y la entrada de referencia, comparándolas y usando la diferencia como medio de control.
- Sistema de control en lazo abierto:** Es un sistema en el cual la salida no tiene efecto sobre la acción de control.
- Tele-operación:** Extensión de las capacidades sensoriales

y destrezas humanas a una localización remota.

Tele-presencia:

Situación ideal en la que el operador se “siente” presente en la localización remota.

Tele-robótica:

Forma evolucionada de tele-operación, caracterizada por un aumento de autonomía en el sistema remoto manteniendo una intervención significativa del operador humano para supervisión o tele-operación directa.