

**DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR  
SCARA (Selective Compliant Assembly Robot)**

**JAIME ANDRÉS GARCÍA NAVARRO  
MIGUEL ÁNGEL ROMERO AMOROCHO**

**UNIVERSIDAD AUTONOMA DE BUCARAMANGA  
ESCUELA DE CIENCIAS NATURALES E INGENIERIA  
FACULTAD INGENIERIA MECATRONICA  
BUCARAMANGA**

**2005**

**DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR  
SCARA (Selective Compliant Assembly Robot)**

**TESIS DE GRADO**

**JAIME ANDRÉS GARCÍA NAVARRO  
MIGUEL ÁNGEL ROMERO AMOROCHO**

**DIRECTOR  
Ing. JHON FABER ARCHILA DÍAZ**

**UNIVERSIDAD AUTONOMA DE BUCARAMANGA  
ESCUELA DE CIENCIAS NATURALES E INGENIERIA  
FACULTAD INGENIERIA MECATRONICA  
BUCARAMANGA  
2005 .**

**DIRECTOR**

---

**EVALUADOR**

---

**EVALUADOR**

---

**JURADO**

---

**JURADO**

---

## DEDICATORIAS

*A Dios quien es nuestra fuente de sabiduría y guía.*

*A mis padres por la confianza que me han brindado hasta este ciclo de mi vida, ya que ha sido un apoyo incondicional en todo el transcurso de mi carrera.*

*A mi hermana que es la mujer por quien yo realizo todos los esfuerzos posibles para que algún día tenga lo mejor.*

*A Miguel mi amigo y hermano, ya que el trabajo realizado fue exitoso por su entrega y dedicación.*

**JAIME A.**

*A Dios que nos da la fortaleza espiritual para no dejarnos vencer por nada.*

*A mis queridos padres por su amor, comprensión y confianza, brindados durante el transcurso de mi vida y en especial durante mi carrera.*

*A mis hermanos, por sus consejos y apoyo incondicional en esta etapa de mi vida.*

*A mi novia, por su entrega y ayuda durante todo este tiempo.*

*Y a mi compañero y mejor amigo Jaime, por su seguridad y voz de aliento que hicieron que este proyecto fuera exitoso.*

**MIGUEL A.**

## **AGRADECIMIENTOS**

Estos agradecimientos van dirigidos a todas las personas que han estado presentes brindándonos ayuda y apoyo cuando la necesitamos, también a quienes nos dieron alientos para seguir adelante.

A nuestro director y docente Ing. Jhon Faber Archila, por su paciencia, dedicación y consejos que nos permitieron dar solución a los inconvenientes presentados, a la Ing. Nayibe Chio Cho por su asesoría y colaboración.

A nuestra facultad por su aporte en la fabricación del proyecto

A los Ingenieros Mecatronicos Javier Jurado, Carlos Forero y Hernán Gonzáles, por ayuda, sugerencias y conocimientos expresados durante la realización de este proyecto.

A cada uno de los docentes que durante el transcurso de la carrera, nos aportaron conocimientos primordiales para el desarrollo de este proyecto.

A Marvin Torres y Mauricio Suárez, por sus aportes, sugerencias, conocimiento y ayuda incondicional durante la carrera y elaboración de este proyecto.

A nuestros amigos y compañeros durante el aprendizaje y trabajo en grupo.

## CONTENIDO

	Pág.
INTRODUCCIÓN	12
1. GENERALIDADES	13
1.1 PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN	13
1.2 OBJETIVOS	14
1.2.1 OBJETIVO GENERAL	14
1.2.2 OBJETIVOS ESPECIFICOS	14
1.3 MARCO TEORICO	15
1.3.1 ROBOT	15
1.3.2 ROBOTS MANIPULADORES	16
1.3.3 MORFOLOGÍA DE LOS ROBOTS	17
A. Tipos de articulaciones	17
B. Estructuras básicas	19
1.3.4 MODELO CINEMÁTICO DIRECTO DE LOS MANIPULADORES	21
1.3.5 MODELO CINEMÁTICO INVERSO DEL ROBOT	22
1.3.6 VELOCIDADES, FUERZAS ESTÁTICAS Y SINGULARIDADES	23
1.3.7 DINÁMICA	24
1.3.8 CONTROL CINEMATICO	25
A. Trayectorias	26
1.3.9 EFECTORES FINALES	28
A. Pinzas	28
B. Herramientas	29
1.3.10 COMUNICACIÓN SERIAL	30
Transmisión Asíncrona	32
1.3.11 COMUNICACIÓN INALAMBRICA Y CODIFICACION DE SEÑALES	33
1.3.12 ACTUADORES	34
Servomotores	34

<b>2. DISEÑO</b>	<b>35</b>
<b>2.1 METODOLOGIA DE DISEÑO MECATRÓNICO</b>	<b>35</b>
<b>2.2 DISEÑO MECATRONICO</b>	<b>39</b>
2.2.1 SCARA (Selective Compliance Assembly Robot Arm)	39
2.2.2 EVALUACIÓN DEL MANIPULADOR SCARA DE LA FACULTAD	40
2.2.3 PROPUESTA DE MEJORAMIENTO PARA EL MANIPULADOR SCARA DE LA FACULTAD	42
2.2.4 ETAPA DE CONTROL	46
A. Control de Actuadores	52
2.2.5 DISEÑO DE CIRCUITOS ELÉCTRICOS	57
2.2.6 DISEÑO DE LA ESTRUCTURA	60
Diseño del elemento Terminal	64
2.2.7 CINEMÁTICA DIRECTA	67
Parámetros de Denavit – Hartenberg	67
2.2.8 CINEMÁTICA INVERSA	71
2.2.9 MATRIZ JACOBIANA DIRECTA	73
2.2.10 MATRIZ JACOBIANA INVERSA	74
2.2.11 DINÁMICA	75
2.2.11.1 Método de Solución por Newton – Euler	75
2.2.12 ANÁLISIS DEL DISEÑO DEL MANIPULADOR Y SELECCIÓN DE MATERIALES	79
A. Selección de Materiales	87
B. Selección de actuadores	91
<b>3. CONSTRUCCION Y MONTAJE</b>	<b>93</b>
<b>4. CONCLUSIONES Y RECOMENDACIONES</b>	<b>97</b>
<b>5. BIBLIOGRAFIA</b>	<b>99</b>

## LISTA DE FIGURAS

	Pág.
FIGURA 1. ESQUEMA BÁSICO DE UN ROBOT	15
FIGURA 2. CADENA CINEMATICA ABIERTA	17
FIGURA 3. TIPOS DE ARTICULACIONES	18
FIGURA 4. ESTRUCTURAS BÁSICAS DE MANIPULADORES	20
FIGURA 5. EJEMPLO DE UN ROBOT SCARA	21
FIGURA 6. DIAGRAMA DE RELACIÓN ENTRE CINEMÁTICA DIRECTA E INVERSA	23
FIGURA 7. DIAGRAMA DE RELACIÓN ENTRE MATRIZ JACOBIANA DIRECTA E INVERSA	24
FIGURA 8. FUNCIONAMIENTO DEL CONTROL CINEMÁTICO	25
FIGURA 9. DIFERENTES TRAYECTORIAS ARTICULARES POSIBLES PARA UN ROBOT SCARA DE 2 GDL. A) MOVIMIENTO EJE A EJE, B) MOVIMIENTO SIMULTÁNEO DE EJES, C) TRAYECTORIA COORDINADA, D) TRAYECTORIA CONTINUA RECTILÍNEA	27
FIGURA 10. A) PINZA MECÁNICA. B) ELECTROIMÁN	28
FIGURA 11. A) PINZA PARA SOLDADURA POR PUNTOS. B) CAÑÓN LÁSER	29
FIGURA 12. TRANSMISIÓN DE DATOS PC – INSTRUMENTO (DIRECTO)	30
FIGURA 13. TRANSMISIÓN DE DATOS PC – PC (CRUZADO)	31
FIGURA 14. CONECTOR DB9	31
FIGURA 15. TRANSMISIÓN ASÍNCRONA DE DATOS	32
FIGURA 16. CODIFICACIÓN DE DATOS BINARIOS	33
FIGURA 17. PULSOS PARA MOVIMIENTO DEL SERVOMOTOR	34
FIGURA 18. METODOLOGÍA DE DISEÑO MECATRÓNICO	35
FIGURA 19. METODOLOGÍA DE DISEÑO MECATRÓNICO APLICADA AL DISEÑO Y CONSTRUCCIÓN DE UN MANIPULADOR	36
FIGURA 20. DIVERSOS MANIPULADORES TIPO SCARA	39
FIGURA 21. MANIPULADOR SCARA DE LA FACULTAD	40
FIGURA 22. PRUEBAS MANIPULADOR SCARA	41
FIGURA 23. REDISEÑO MANIPULADOR SCARA DE LA FACULTAD	42
FIGURA 24. PRUEBAS NUEVA PROGRAMACIÓN	43
FIGURA 25. MEJORAS FORMALES DEL MANIPULADOR	44
FIGURA 26. MEJORAS DE LAS CONEXIONES DE LOS ACTUADORES	45
FIGURA 27. DIAGRAMA DE BLOQUES CONTROL DE POSICIÓN	46

FIGURA 28. DIAGRAMA DE BLOQUES CONTROL DIRECTO	47
FIGURA 29. DIAGRAMA DE BLOQUES CONTROL INVERSO	47
FIGURA 30. DIAGRAMA DE BLOQUES CONTROL DE INERCIA	47
FIGURA 31. DIAGRAMA DE BLOQUES CONTROL DE TRAYECTORIAS	48
FIGURA 32. ALGORITMO DEL MICROCONTROLADOR MAESTRO	49
FIGURA 33. ALGORITMO DE LOS MICROCONTROLADORES ESCLAVOS	50
FIGURA 34. ALGORITMO INTERNO DE LOS MICROCONTROLADORES ESCLAVOS 1 Y 2	51
FIGURA 35. ALGORITMO DEL MICROCONTROLADOR ESCLAVO 3.	51
FIGURA 36. DIAGRAMA ESQUEMÁTICO DE UN SISTEMA DE SEGUIMIENTO O SERVOMOTOR	53
FIGURA 37. DIAGRAMA DE BLOQUES PARA EL SISTEMA	56
FIGURA 38. DIAGRAMA DE BLOQUES SIMPLIFICADO	56
FIGURA 39. DIAGRAMA DE CIRCUITOS ELÉCTRICOS	57
FIGURA 40. CIRCUITO ELÉCTRICO TARJETA TRANSMISORA	58
FIGURA 41. CIRCUITO ELÉCTRICO TARJETA RECEPTORA	59
FIGURA 42. PRIMER DISEÑO DEL MANIPULADOR	60
FIGURA 43. SEGUNDO DISEÑO DEL MANIPULADOR	61
FIGURA 44. TERCER DISEÑO DEL MANIPULADOR	62
FIGURA 45. DISEÑO FINAL DEL MANIPULADOR CON PINZA	63
FIGURA 46. DISEÑO FINAL DEL MANIPULADOR CON PORTA- HERRAMIENTA	63
FIGURA 47. DISEÑO DEL SISTEMA DE DESPLAZAMIENTO	64
FIGURA 48. PRIMER Y SEGUNDO DISEÑO DE LA PINZA DEL MANIPULADOR	65
FIGURA 49. DISEÑO FINAL DE LA PINZA DEL MANIPULADOR	66
FIGURA 50. DISEÑO DEL ELEMENTO TERMINAL PORTA-HERRAMIENTA	66
FIGURA 51. SISTEMA DE EJES COORDENADOS ROBOT SCARA	68
FIGURA 52. MATRICES DE TRANSFORMACIÓN HOMOGÉNEA	69
FIGURA 53. MODELO CINEMÁTICO INTEGRADO	69
FIGURA 54. ORIENTACIÓN Y POSICIÓN DEL ROBOT	70
FIGURA 55. FIGURA PARA EL ANÁLISIS DEL MÉTODO GEOMÉTRICO	71
FIGURA 56. ALGORITMO DE SOLUCIÓN DE LA CINEMÁTICA INVERSA	72
FIGURA 57. VARIABLES DE LA MATRIZ JACOBIANA	73
FIGURA 58. COMPONENTES DE LA MATRIZ JACOBINA	73
FIGURA 59. MATRIZ JACOBINA INVERSA	74
FIGURA 60. DIAGRAMA ESQUEMÁTICO DEL ROBOT SCARA	75
FIGURA 61. FUERZAS Y MOMENTOS EJERCIDOS EN EL ESLABÓN I	76
FIGURA 62. ALGORITMO DINÁMICO E NEWTON – EULER	78
FIGURA 63. ENMALLADO Y FUERZA APLICADA A LA BASE	79
FIGURA 64. DEFORMACIÓN Y ESFUERZO EQUIVALENTE DE LA BASE	80
FIGURA 65. FACTOR DE SEGURIDAD DE LA BASE	80
FIGURA 66. ENMALLADO Y FUERZA APLICADA AL ANTEBRAZO	81
FIGURA 67. DEFORMACIÓN Y ESFUERZO EQUIVALENTE DEL ANTEBRAZO	82

<b>FIGURA 68. FACTOR DE SEGURIDAD DEL ANTEBRAZO</b>	<b>82</b>
<b>FIGURA 69. ENMALLADO Y FUERZA APLICADA AL BRAZO</b>	<b>83</b>
<b>FIGURA 70. DEFORMACIÓN Y ESFUERZO EQUIVALENTE DEL BRAZO</b>	<b>84</b>
<b>FIGURA 71. FACTOR DE SEGURIDAD DEL BRAZO</b>	<b>84</b>
<b>FIGURA 72. ENMALLADO Y FUERZA APLICADA A LA PINZA</b>	<b>85</b>
<b>FIGURA 73. DEFORMACIÓN Y ESFUERZO EQUIVALENTE DE LA PINZA</b>	<b>85</b>
<b>FIGURA 74. FACTOR DE SEGURIDAD DE LA PINZA</b>	<b>86</b>
<b>FIGURA 75. SERVOMOTOR HITEC</b>	<b>92</b>
<b>FIGURA 76. MOTOREDUCTOR</b>	<b>92</b>
<b>FIGURA 77. MODELOS EN MADERA</b>	<b>93</b>
<b>FIGURA 78. PIEZAS EN FUNDICIÓN DE ALUMINIO</b>	<b>94</b>
<b>FIGURA 79. PIEZAS MECANIZADAS</b>	<b>94</b>
<b>FIGURA 80. CIRCUITO TARJETA TRANSMISORA</b>	<b>95</b>
<b>FIGURA 81. CIRCUITO TARJETA RECEPTORA</b>	<b>95</b>
<b>FIGURA 82. MANIPULADOR TIPO SCARA CONSTRUIDO</b>	<b>96</b>

## LISTA DE TABLAS

	Pág.
TABLA 1. PINES DEL CONECTOR DB9	31
TABLA 2. CONEXIONES DEL MANIPULADOR	45
TABLA 3. PARÁMETROS DE DENAVIT HARTENBERG DEL MANIPULADOR SCARA	68
TABLA 4. TABLA DE VARIABLES DE NEWTON EULER	76
TABLA 5. USOS DE ALEACIONES DE ALUMINIO	87
TABLA 6. PROPIEDADES DEL ALUMINIO	88
TABLA 7. USOS DE ACEROS	89
TABLA 8. PROPIEDADES DEL ACERO INOXIDABLE	89
TABLA 9. PROPIEDADES DEL ACRÍLICO	90

## INTRODUCCIÓN

A través del tiempo la tecnología ha sido muy influyente en la forma de vida de cada sociedad, tanto así que han buscado optar nuevas formas de vida con ayuda de esta. Uno de los mejores avances y logros obtenidos han sido los Robots y la ciencia de estos, la Robótica.

Hay una creciente necesidad de automatizar procesos para tener una variedad mas amplia de productos, con diferentes requerimientos tecnológicos, calidad mejorada, desarrollo en menor tiempo y a los mas bajos costos; esto con el único fin de satisfacer las necesidades del ser humano. Desde hace 30 años desde la invención del robot SCARA (Selective Compliant Assembly Robot Arm) los robots industriales han sido "caballos de trabajo" de la manufactura.

Actualmente existe un gran interés por la investigación y el aprendizaje en la Robótica; por ello que carreras como Ingeniería Mecatrónica han contribuido al conocimiento en dichos temas; y a su vez han impulsado ideas que se plasman en proyectos. En la actualidad existen muchas universidades que trabajan en este campo y que han obtenidos logros muy significativos.

Con este proyecto de grado se logra un avance en este campo dentro de la Facultad de Ingeniería Mecatronica, así como conjugar y aplicar valiosos conocimientos obtenidos durante la carrera en busca de ampliar el estudio de las ramas de la ingeniería aportando una herramienta que facilite su aprendizaje y apoye la creación de nuevas y mejores maquinas.

# **1. GENERALIDADES**

## **1.1 PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN**

En los últimos años se ha introducido el concepto robótica el cual ha venido a revolucionar la automatización de su clasificación denominada "fija" que consistía en la realización de la producción automática de piezas, elementos y productos en grandes cantidades o de manera repetitiva a su denominación actual "automatización flexible" que radica en adaptar la producción a la demanda de un mercado en constante cambio por medio de un sistema de producción programable y adaptable como lo es un robot.

Hoy en día se ve la robótica como un trabajo mucho más amplio, que trata temas además de la investigación y el desarrollo de una serie de áreas interdisciplinarias, con la cinemática, dinámica, planificación de sistemas, control, sensores, lenguajes de programación e inteligencia de máquina. Los robots son un elemento indispensable en los procesos de manufactura así como soldadura, paletización, etc.; dependiendo de las funciones de sus elementos terminales se hacen adaptables a diversos tipos de entorno.

Por esta razón surgió este proyecto de grado con el fin de diseñar, construir y programar un manipulador tipo SCARA para su integración en el Laboratorio de Robótica de la Facultad de Ingeniería Mecatrónica, en busca de aumentar el interés de los alumnos por la carrera y fomentar su entusiasmo por la investigación y el desarrollo de la misma.

## **1.2 OBJETIVOS**

### **1.2.1 OBJETIVO GENERAL**

Diseñar, construir y programar un manipulador tipo SCARA para su integración en el Laboratorio de Robótica de la Facultad de Ingeniería Mecatrónica.

### **1.2.2 OBJETIVOS ESPECIFICOS**

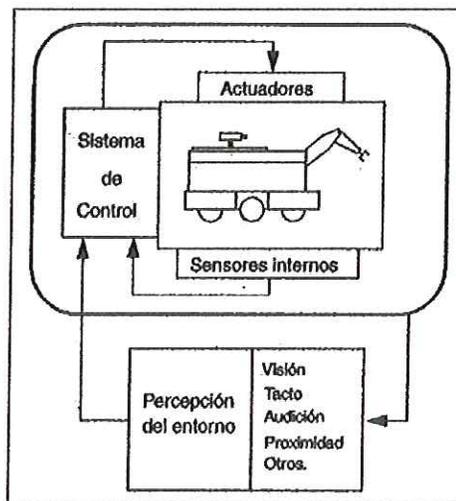
- Evaluación del manipulador tipo SCARA de la Facultad.
- Realizar un estudio del estado actual de los manipuladores tipo SCARA, para el nuevo diseño del mismo.
- Diseñar y construir un elemento terminal para la manipulación de objetos.
- Diseñar y construir circuitos eléctricos para la comunicación PC-Manipulador.
- Realizar cálculos cinemáticos y dinámicos del manipulador.
- Implementar un control de posición.
- Implementar un control de trayectorias punto a punto.
- Implementar un control de inercia.
- Realizar la programación necesaria para planeamiento de una tarea del manipulador.

## 1.3 MARCO TEORICO

### 1.3.1 ROBOT

La definición técnica adoptada por el Instituto Norteamericano de Robótica y aceptada internacionalmente es la siguiente: "Un robot es un manipulador multifuncional reprogramable, diseñado para mover materiales, piezas, herramientas o dispositivos especiales mediante movimientos programados y variables que permiten realizar diversas tareas". Suelen tener forma de brazo articulado, en cuyo extremo incorporan elementos de sujeción o herramientas. Realizan tareas repetitivas en industrias de automoción, fabricación mecánica o electrónica, en las que se emplean para montar y mover piezas o componentes, ajustarlos, soldar, pintar, etcétera.

Figura 1. Esquema básico de un robot



Ollero, A, Robótica: Manipuladores y robots móviles

Una vez comprendido el concepto de robot podemos avanzar hacia la definición de la ciencia que estudia este tipo de dispositivos, la cual se denomina "Robótica" y ha evolucionado rápidamente en estos últimos años.

Podríamos aproximarnos a una definición de Robótica como:

El diseño, fabricación y utilización de máquinas automáticas programables con el fin de realizar tareas repetitivas como el ensamble de automóviles, aparatos, etc. y otras actividades.

Básicamente, la robótica se ocupa de todo lo concerniente a los robots, lo cual incluye el control de motores, mecanismos automáticos neumáticos, sensores, sistemas de cómputos, etc.

### **1.3.2 ROBOTS MANIPULADORES**

La mayor parte de los robots industriales actuales son esencialmente brazos articulados. De hecho, según la definición del "Robot Institute of America", un robot industrial es un manipulador programable multifuncional diseñado para mover materiales, piezas, herramientas o dispositivos especiales, mediante movimientos variados, programados para la ejecución de distintas tareas.

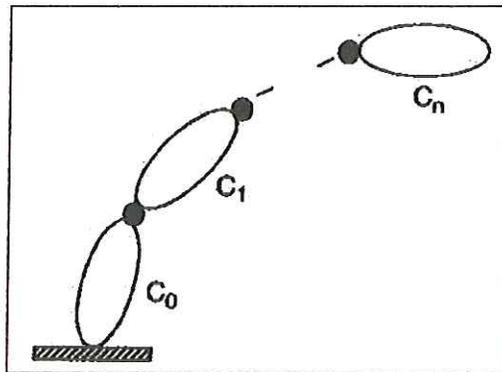
Dentro de la robótica industrial, los manipuladores industriales logran dotar de flexibilidad a los procesos productivos manteniendo al mismo tiempo la productividad que se consigue con una máquina automática especializada.

El objetivo no es sustituir la actividad directa de un trabajador en una cadena de producción; se trata de realizar tareas en lugares difícilmente accesibles, con riesgo de accidentes, en condiciones peligrosas para la salud, o trabajos que resultan difíciles por el tamaño de los objetos que es necesario manipular.

### 1.3.3 MORFOLOGÍA DE LOS ROBOTS

Las características básicas de la estructura de los robots están formadas por los tipos de articulaciones y configuraciones clásicas de brazos de robots industriales. Los robots manipuladores son esencialmente, brazos articulados. De forma más precisa, un manipulador industrial convencional es una cadena cinemática abierta formada por un conjunto de eslabones o elementos de la cadena interrelacionados mediante articulaciones o pares cinemáticos como lo esquematiza la figura 2. Las articulaciones permiten el movimiento relativo entre los sucesivos eslabones.

**Figura 2. Cadena cinemática abierta**



Ollero, A, Robótica: Manipuladores y robots móviles

#### **A. Tipos de articulaciones**

Existen diferentes tipos de articulaciones. Las más utilizadas en robótica son las que se indican en la figura 3.

La articulación de rotación suministra un grado de libertad consistente en una rotación alrededor del eje de la articulación. Esta articulación es, con diferencia, la más empleada.

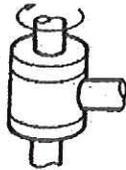
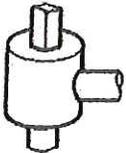
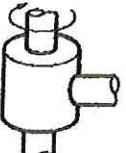
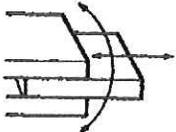
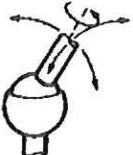
En la articulación prismática el grado de libertad consiste en una traslación a lo largo del eje de la articulación.

En la articulación cilíndrica existen dos grados de libertad: una rotación y una traslación.

La articulación planar está caracterizada por el movimiento de desplazamiento en un plano, existiendo por lo tanto, dos grados de libertad.

Por último, la articulación esférica combina tres giros en tres direcciones perpendiculares en el espacio.

**Figura 3. Tipos de articulaciones**

ESQUEMA	ARTICULACIÓN	GRADOS LIBERTAD
	ROTACIÓN	1
	PRISMÁTICA	1
	CILÍNDRICA	2
	PLANAR	2
	ESFÉRICA (RÓTULA)	3

Ollero, A, Robótica: Manipuladores y robots móviles

Los grados de libertad son el número de parámetros independientes que fijan la situación del órgano terminal. El número de grados de libertad suele coincidir con el número de eslabones de la cadena cinemática.

## **B. Estructuras básicas**

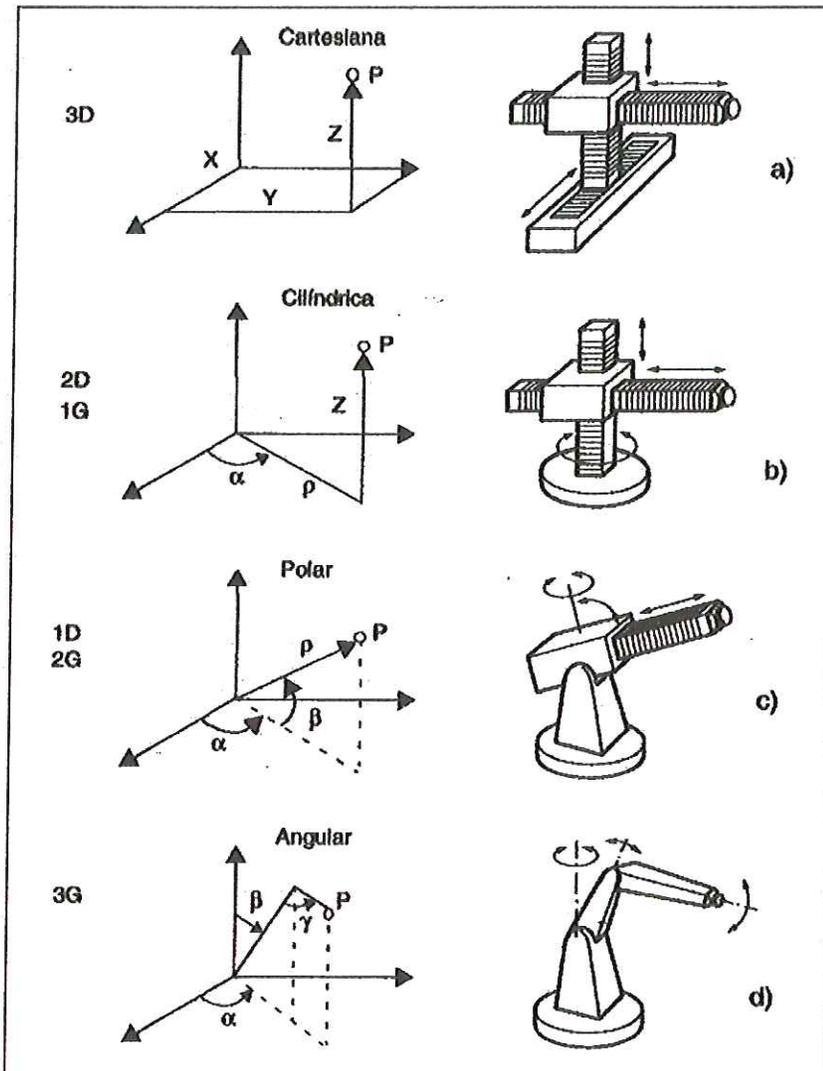
La estructura típica de un manipulador consiste en un brazo compuesto por elementos con articulaciones entre ellos. En el último enlace se coloca un órgano terminal o efector final tal como una pinza o un dispositivo especial para realizar operaciones.

Se consideran, en primer lugar, las estructuras más utilizadas como brazo de un robot manipulador. Estas estructuras tienen diferentes propiedades en cuanto a espacio de trabajo y accesibilidad a posiciones determinadas. En la figura 4 se muestran cuatro configuraciones básicas.

El espacio de trabajo es el conjunto de puntos en los que puede situarse el efector final del manipulador. Corresponde al volumen encerrado por las superficies que determinan los puntos a los que accede el manipulador con su estructura totalmente extendida y totalmente plegada.

Por otra parte, todos los puntos del espacio de trabajo no tienen la misma accesibilidad. Los puntos de accesibilidad mínima son los que las superficies que delimitan el espacio de trabajo ya que a ellos solo puede llegarse con una única orientación.

Figura 4. Estructuras básicas de manipuladores



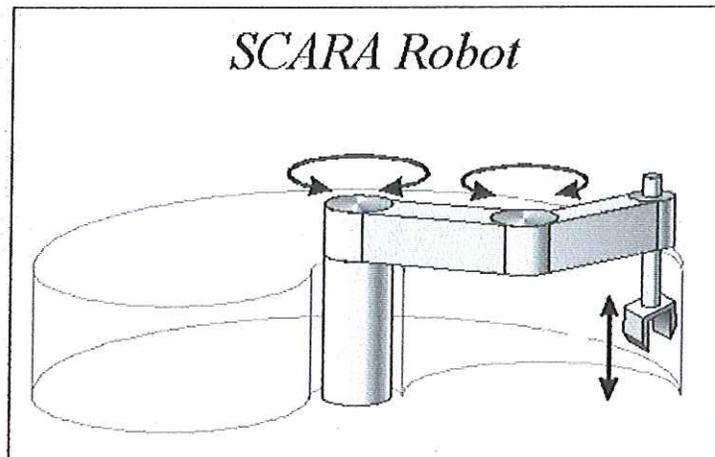
Ollero, A, Robótica: Manipuladores y robots móviles

### Configuración SCARA

Esta configuración está especialmente diseñada para realizar tareas de montaje en un plano. Está constituida por dos articulaciones de rotación con respecto a dos ejes paralelos, y una de desplazamiento en sentido perpendicular al plano. El

volumen de trabajo de este robot, suponiendo segmentos de longitud  $L$ , un radio de giro de 360 grados y un rango de desplazamiento de  $L$  es de  $4\pi L^3$ .

**Figura 5. Ejemplo de un robot scara**



<http://prime.jsc.nasa.gov/ROV/images/SCARA.GIF>

Para llevar a cabo los cálculos y de esta forma asegurar su correcto funcionamiento del robot en cuanto a la cinemática y dinámica se refiere, se toma en consideración la siguiente teoría que tiene por objeto crear las bases de un modelo matemático del sistema.

#### **1.3.4 MODELO CINEMÁTICO DIRECTO DE LOS MANIPULADORES**

La cinemática es la ciencia del movimiento que trata a éste sin importarle las fuerzas que lo causan. Dentro de la cinemática se estudia la posición, la velocidad, aceleración y todas las derivadas de las variables de posición de mayor orden con respecto al tiempo o cualquier otra variable.

El estudio de la cinemática de los manipuladores se refiere a todas las propiedades geométricas y basadas en el tiempo del movimiento.

Los robots consisten en un conjunto de eslabones conectados mediante articulaciones que permiten el movimiento relativo entre los eslabones vecinos. El número de grados de libertad que un robot posee es el número de variables de posición independientes que deberían ser especificadas para localizar todas las partes del mecanismo. En el caso de los robots industriales el número de grados de libertad suele equivaler al número de articulaciones siempre y cuando cada articulación tenga un solo grado de libertad. Al final de la cadena de eslabones del robot se encuentra el órgano terminal. Dependiendo de la aplicación del robot, el órgano terminal puede ser una pinza, un soldador, un electroimán o un gripper como en este caso.

Generalmente se describe la posición del robot dando una descripción del marco de la herramienta, la cual está unida al órgano terminal, relativo al marco de la base, el cual está a su vez unido a la base fija del robot.

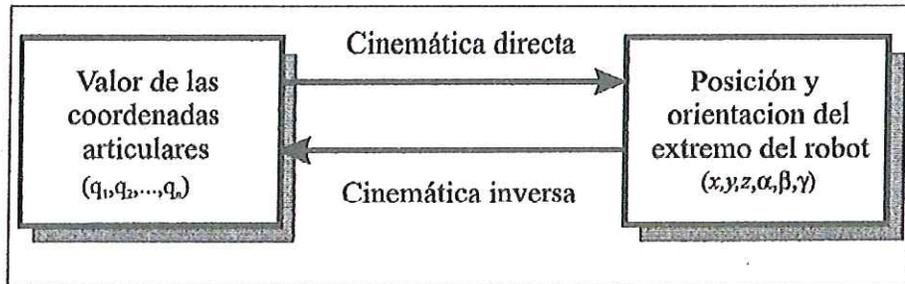
El modelo cinemático directo es el problema geométrico que calcular la posición y orientación del efector final del robot. Dada una serie de ángulos entre las articulaciones, el problema cinemático directo calcula la posición y orientación del marco de referencia del efector final con respecto al marco de la base.

### **1.3.5 MODELO CINEMÁTICO INVERSO DEL ROBOT**

Dada la posición y orientación del efector final del robot, el problema cinemático inverso consiste en calcular todos los posibles conjuntos de ángulos entre las articulaciones que podrían usarse para obtener la posición y orientación deseada. El problema cinemático inverso es más complicado que la cinemática directa ya que las ecuaciones no son lineales, sus soluciones no son siempre fáciles o incluso posibles en una forma cerrada. También surge la existencia de una o de diversas soluciones. La existencia o no de la solución lo define el espacio de

trabajo de un robot dado. La ausencia de una solución significa que el robot no puede alcanzar la posición y orientación deseada porque se encuentra fuera del espacio de trabajo del robot o fuera de los rangos permisibles de cada una de sus articulaciones.

**Figura 6. Diagrama de relación entre cinemática directa e inversa**



Barrientos Antonio. Fundamentos de Robótica.

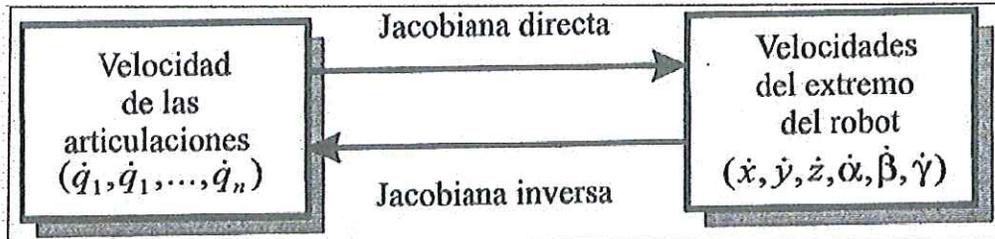
### 1.3.6 VELOCIDADES, FUERZAS ESTÁTICAS Y SINGULARIDADES

Además de tratar problemas de posicionamiento estáticos debemos analizar los robots en movimiento.

Cuando se analiza la velocidad de un mecanismo es conveniente definir una matriz llamada el Jacobiano del manipulador. El jacobiano establece una aplicación entre las velocidades del espacio de articulaciones y las velocidades del espacio cartesiano. En ciertos puntos (llamados singularidades) esta aplicación no es invertible.

Muchas veces los robots que no se mueven en el espacio, simplemente aplican una fuerza estática sobre alguna pieza o superficie de trabajo. En este caso se nos proporciona una fuerza de contacto y un momento de aquí que la matriz Jacobiana nos ayuda a encontrar la solución.

**Figura 7. Diagrama de relación entre Matriz Jacobiana directa e inversa**



Barrientos Antonio. Fundamentos de Robótica.

### 1.3.7 DINÁMICA

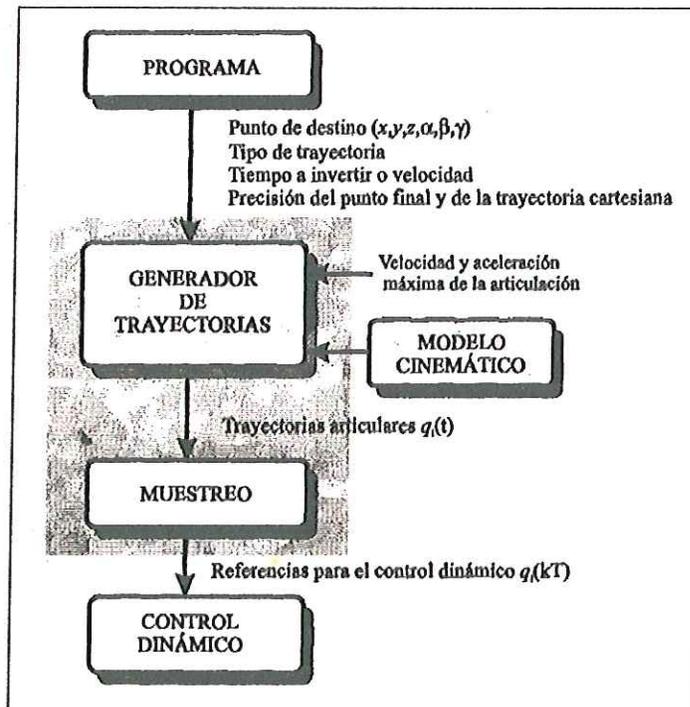
La dinámica es un campo de las ciencias dedicado al estudio de las fuerzas requeridas para producir el movimiento. Para acelerar un robot desde el reposo y finalmente desacelerarlo hasta una completa posición de reposo, los actuadores articulares (motores eléctricos, actuadores hidráulicos y neumáticos), deben aplicar un conjunto complejo de funciones de par.

Un método para controlar que un robot siga un camino determinado consiste en calcular estas funciones de par usando las ecuaciones dinámicas del robot. Un segundo uso de las ecuaciones dinámicas del movimiento es en la simulación. Reformulando las ecuaciones dinámicas de forma que la aceleración se calcule como una función del par actuador, es posible simular cómo un robot se movería bajo la aplicación de un conjunto de pares del actuador.

### 1.3.8 CONTROL CINEMATICO

El control cinemático establece cuáles son las trayectorias que debe seguir cada articulación del robot a lo largo del tiempo para lograr los objetivos fijados por el usuario (punto de destino, trayectoria cartesiana del efector final del robot, tiempo invertido por el usuario, etc.). Estas trayectorias se seleccionarán atendiendo a las restricciones físicas propias de los accionamientos y a ciertos criterios de calidad de trayectoria, como suavidad o precisión de la misma.

Figura 8. Funcionamiento del control cinemático



## **A. Trayectorias**

Para realizar una tarea determinada el robot debe moverse desde un punto inicial a un punto final. Este movimiento puede ser realizado según infinitas trayectorias espaciales. De todas ellas hay algunas que, bien por su sencillez de implementación por parte del control cinemático o bien por su utilidad y aplicación a diversas tareas, son las que en la práctica incorporan los robots comerciales. De este modo, puede encontrarse que los robots dispongan de trayectorias punto a punto, coordinadas y continuas.

### **Trayectorias punto a punto**

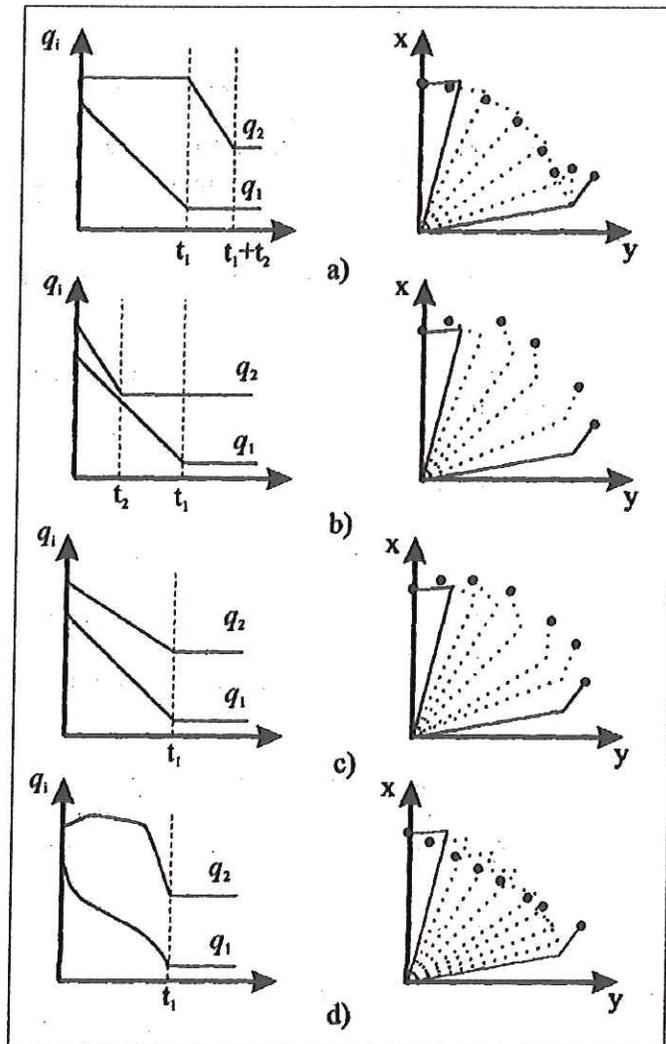
En este tipo de trayectorias cada articulación evoluciona desde su posición inicial a la final sin realizar una consideración alguna sobre el estado o evolución de las demás articulaciones. Normalmente, cada actuador trata de llevar a su articulación al punto de destino en el menor tiempo posible, pudiéndose distinguir dos casos: movimiento eje a eje y movimiento simultáneo de ejes.

#### **Movimiento simultáneo de ejes.**

En este caso todos los actuadores comienzan simultáneamente a mover las articulaciones del robot a una velocidad específica para cada una de ellas. Dado que la distancia a recorrer y las velocidades serán en general diferentes, cada una acabará su movimiento en un instante diferente. El movimiento del robot no acabará hasta que se alcance definitivamente el punto final, lo que se producirá cuando el eje que más tarde concluya su movimiento. De esta manera, el tiempo total invertido en el movimiento coincidirá con el del eje que más tiempo emplee en realizar su movimiento particular, pudiéndose dar la circunstancia de que el resto de los actuadores hayan forzado su movimiento a una velocidad y aceleración elevada, viéndose obligados finalmente a esperar a la articulación más lenta.

La Figura 9 representa las trayectorias articulares  $q_1(t)$  y  $q_2(t)$  y el resultado final en el espacio de la tarea  $(x, y)$  para un robot tipo SCARA correspondientes a los cuatro tipos de trayectorias indicadas.

**Figura 9. Diferentes trayectorias articulares posibles para un robot SCARA de 2 GDL. a) movimiento eje a eje, b) movimiento simultáneo de ejes, c) trayectoria coordinada, d) trayectoria continua rectilínea**



Barrientos Antonio. Fundamentos de Robótica.

### 1.3.9 EFECTORES FINALES

Por efector final, o elemento terminal, se entiende cualquier dispositivo que, colocado en el extremo del robot, permite a éste realizar una tarea concreta.

Los efectores finales se clasifican en dos grandes grupos:

Pinzas: para la sujeción y manipulación de piezas.

Herramientas: para la realización de operaciones sobre piezas.

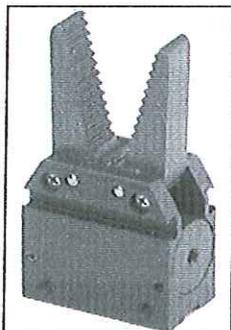
También es habitual encontrar como elemento terminal algún dispositivo sensor, como puede ser un palpador o una cámara de visión artificial.

#### A. Pinzas

Gran parte de las tareas de los robots industriales consiste en la manipulación de piezas, es decir, recoger una pieza, trasladarla y depositarla en otro lugar. Estas operaciones se realizan mediante pinzas; robots de prensas, empaquetado, paletizado.

Las pinzas pueden clasificarse según el modo de sujeción, pinzas mecánicas, pinzas succionadoras o ventosas, pinzas magnéticas o electroimanes, pinzas especiales (a medida de la aplicación).

**Figura 10. a) Pinza mecánica. b) Electroimán**



**a)**



**b)**

<http://www.gimatic.com>

## B. Herramientas

Las herramientas se utilizan en aplicaciones en las que el robot ha de realizar alguna operación sobre la pieza. Las herramientas más usadas en aplicaciones robotizadas son las siguientes:

Pinzas para soldadura por puntos.

Antorcha para la soldadura al arco.

Cañón de agua a alta presión.

Cañón láser.

Pistola de pintura.

Fresa-lija: se pueden montar cabezales fresadores de vacío, rectificadoras, lijadoras de banda o de disco, etc.

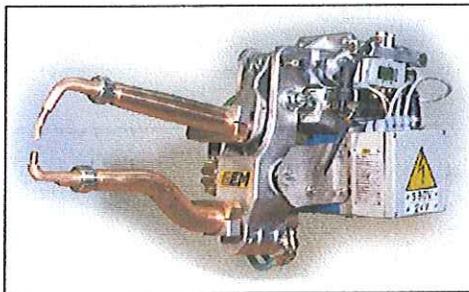
Atornillador: suele incluir la alimentación de tornillos.

Cucharón para colada: se utiliza en trabajos de fundición.

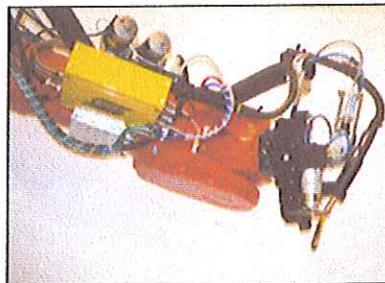
La herramienta está fijada rígidamente al extremo del robot, o utiliza un dispositivo de cambio automático de pinza.

La mayor dificultad en este tipo de aplicaciones es el transporte de fluidos o materiales de aporte (hilo de soldadura, agua, etc.). Suele ser necesario añadir al brazo soportes y/o colgadores extensibles.

**Figura 11. a) Pinza para soldadura por puntos. b) Cañón láser**



a)



b)

[http://gemwelding.com/img\\_genr/pinzas/G01n.jpg](http://gemwelding.com/img_genr/pinzas/G01n.jpg)

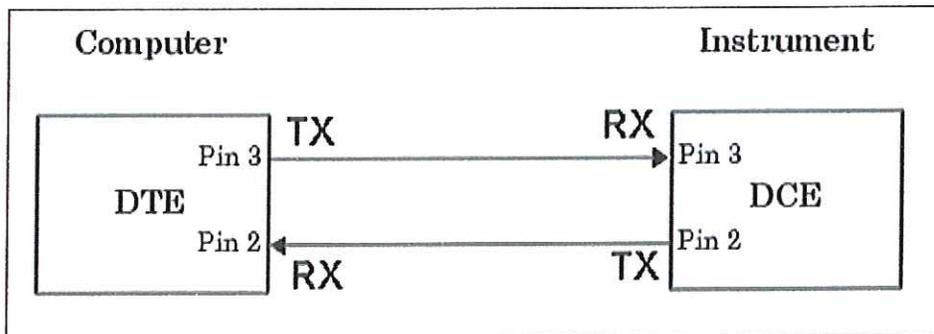
### 1.3.10 COMUNICACIÓN SERIAL

La comunicación serial es el protocolo más común para comunicarse entre dos o más dispositivos. Normalmente, un dispositivo es un computador, mientras que el otro dispositivo puede ser un módem, impresora, otra computadora, o un instrumento científico tal como un osciloscopio o un generador de funciones.

El puerto serial envía y recibe octetos de información en una manera serial, un bit a la vez. Estos octetos se transmiten usando un formato binario o un formato del texto (ASCII).

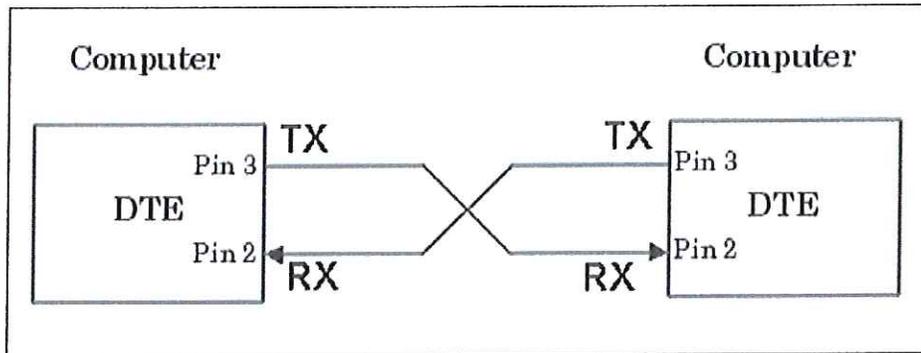
La norma serie RS-232 fue diseñada para conectar DTEs (Data Terminal Equipment) o equipos terminales de datos (como un terminal, un ordenador,...) con DCEs (Data Communication Equipment) o equipos de comunicación de datos, como módems, codecs, Pc's, etcétera.

**Figura 12. Transmisión de Datos Pc – Instrumento (Directo)**



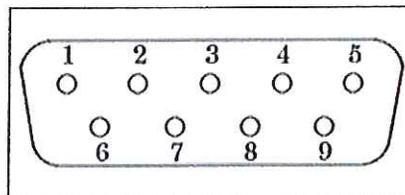
Matlab 7.0 – Help - Serial I/O Ports (External Interfaces)

**Figura 13. Transmisión de Datos Pc – Pc (Cruzado)**



Matlab 7.0 – Help - Serial I/O Ports (External Interfaces)

**Figura 14. Conector DB9**



Matlab 7.0 – Help - Serial I/O Ports (External Interfaces)

**Tabla 1. Pines del conector DB9**

Nº Pin	Señal
1	DCD (DATA CARRIER DETECT)
2	RX
3	TX
4	DTR (DATA TERMINAL READY)
5	GND
6	DSR (DATA SHEET READY)
7	RTS (REQUEST TO SEND)
8	CTS (CLEAR TO SEND)
9	RI (RING INDICATOR)

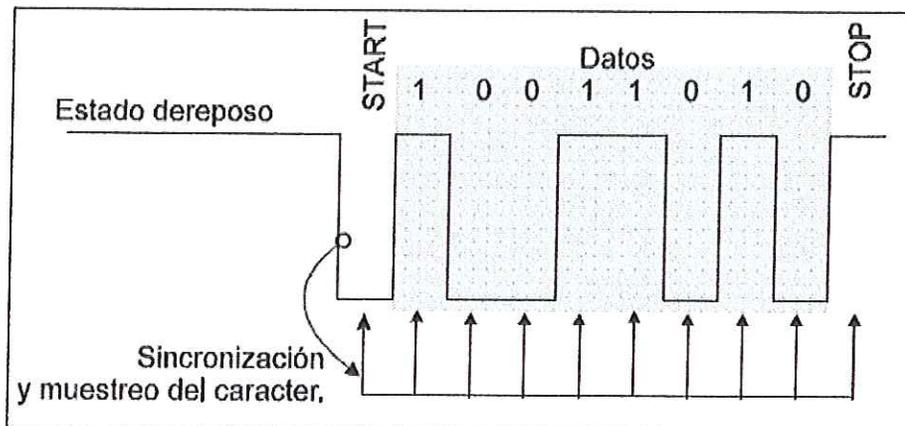
Matlab 7.0 – Help - Serial I/O Ports (External Interfaces)

La RS-232 permite la transmisión sincronía y asíncrona. La subnorma asíncrona es sin duda la más frecuente por lo que nos centraremos en ella.

### Transmisión Asíncrona

Cuando el emisor no transmite, en el periodo entre caracteres, la línea se mantiene a "1" lógico. Cuando decide transmitir un carácter, primero transmite un "0" que se denomina bit de START y sirve para que el receptor sincronice (empieza a contar tiempos desde ese momento). El instante de sincronismo es el flanco de bajada de la señal (ver figura 15.). Tras el bit de START se transmiten los bits de datos y después es obligatorio al menos un bit de STOP a "1" lógico. La secuencia se repite tantas veces como caracteres se transmitan. Obsérvese que este mecanismo de sincronización con el bit de START impide que la deriva de muestreo por diferencias entre los relojes continúe en el siguiente carácter. Se asume que la deriva de muestreo no debe ser tan grande que provoque un error de muestreo en los bits de cada carácter.

**Figura 15. Transmisión asíncrona de datos**

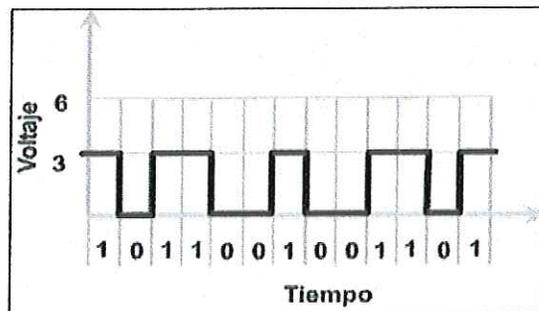


[www.htmlweb.net/redes/tcp\\_ip/capa\\_1/fisica\\_3.html](http://www.htmlweb.net/redes/tcp_ip/capa_1/fisica_3.html)

### 1.3.11 COMUNICACIÓN INALÁMBRICA Y CODIFICACIÓN DE SEÑALES

En toda comunicación se necesita un emisor y un receptor y unos mecanismos o sistemas de emisión-recepción. Por ejemplo las señales de TV se graban con una cámara en unos estudios, mediante sistemas electrónicos se convierte a señales de radiofrecuencia que se propagan en el espacio a la velocidad de la luz, y se envía mediante una antena al espacio. Posteriormente esta señal se recogerá por otra antena en el receptor y se invertirá el proceso. El problema que podría darse es que cuando se mandan muchas señales interfieran unas con otras por eso existen diferentes tipos de codificación de señales. Para que la información circule por las diferentes partes de nuestro ordenador es preciso una transformación de dígitos binarios en impulsos de electricidad continua. El mecanismo general de transformar información (datos) en "algo" que la represente y que sea apto para su transmisión por un medio cualquiera se denomina Codificación, y a esos "algo" que representan la información se les conoce con el nombre de señales.

**Figura 16. Codificación de Datos Binarios**



[www.htmlweb.net/redes/tcp\\_ip/capa\\_1/fisica\\_3.html](http://www.htmlweb.net/redes/tcp_ip/capa_1/fisica_3.html)

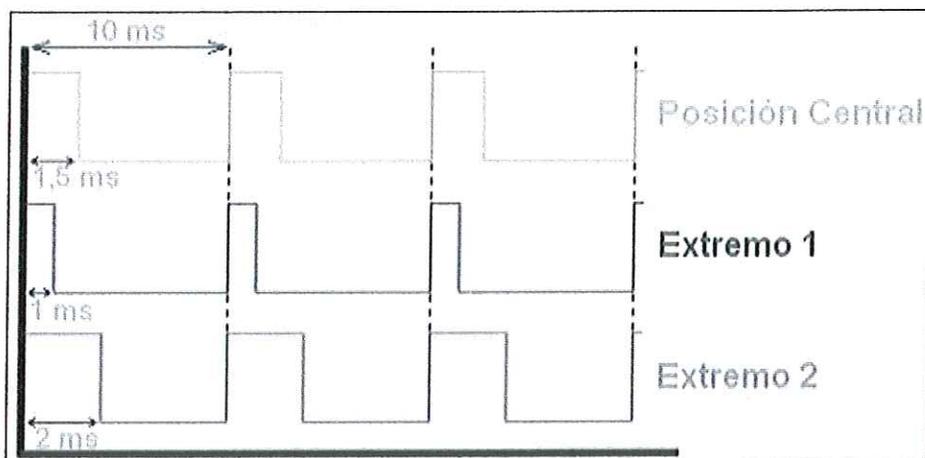
### 1.3.12 ACTUADORES

#### Servomotores

Los servos son un tipo especial de motor que se caracterizan por su capacidad para posicionarse de forma inmediata en cualquier posición dentro de su rango de operación. Para ello, el servo espera un tren de pulsos que se corresponden con el movimiento a realizar. Están generalmente formados por un amplificador, un motor, la reducción de engranaje y la realimentación, todo en un misma caja de pequeñas dimensiones. El resultado es un servo de posición con un margen de operación de 180° aproximadamente.

Disponen de tres conexiones electricas: Vcc (roja), GND(negra) y entrada de control (amarilla). Estos colores de identificación y el orden de las conexiones dependen del fabricante del servo. Es importante identificar las conexiones ya que un voltaje de polaridad contraria podría dañar el servo.

**Figura 17. Pulsos para movimiento del servomotor**



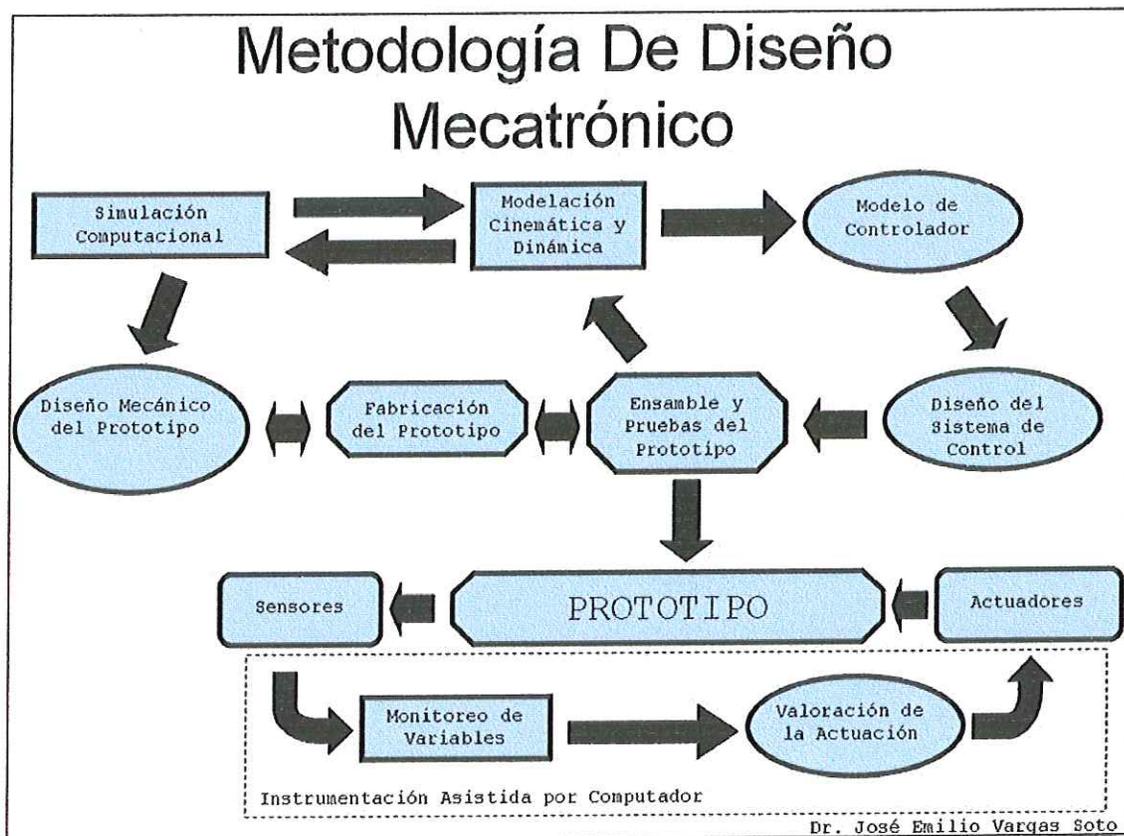
<http://autric.com/Microbotica%20y%20Mecatronica/servomotores.htm>

## 2. DISEÑO

### 2.1 METODOLOGIA DE DISEÑO MECATRÓNICO

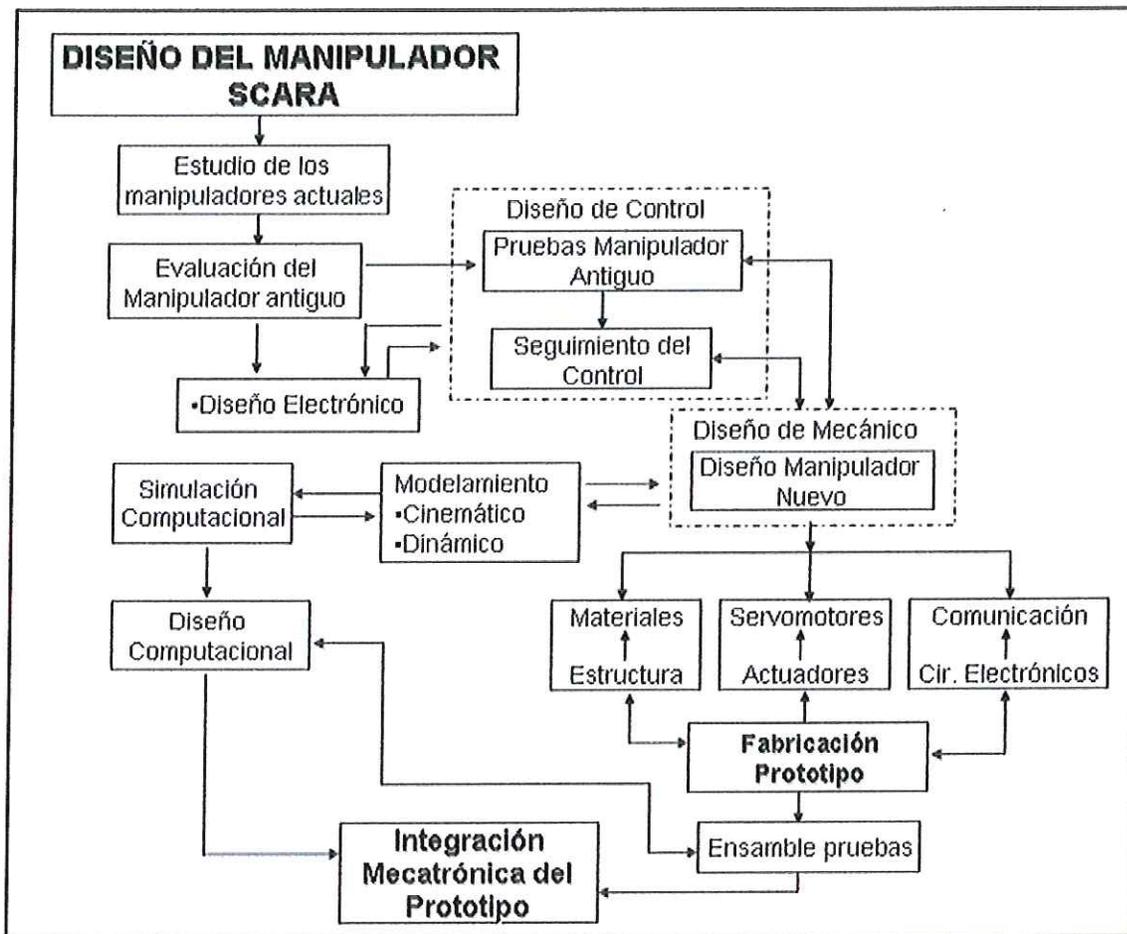
Para el desarrollo de este proyecto de grado, nos orientamos de la metodología de diseño mecatrónico planteada por el Doctor José Emilio Vargas Soto, para desarrollar la nuestra.

Figura 18. Metodología de Diseño Mecatrónico



Dr. José Emilio Vargas Soto

Figura 19. Metodología de Diseño Mecatrónico aplicada al diseño y construcción de un manipulador



Del Autor

Como podemos observar este diseño metodológico muestra todo el desarrollo del proyecto. Inicialmente fue la búsqueda de información y la evaluación del manipulador realizado en la tesis **“DISEÑO Y CONSTRUCCIÓN DE UN MANIPULADOR TIPO SCARA CON IMPLEMENTACION DE UN SISTEMA DE CONTROL”**.

La primera parte de diseño fue el control, desarrollada por medio de pruebas y seguimiento de las mismas, con el fin de encontrar así la mejor programación para

el mejor movimiento del manipulador, paralelamente se diseñó la etapa electrónica la cual hace parte del control, en la evolución del proyecto, nuestro objetivo final era diseñar y construir un manipulador el cual les permitiera a los estudiantes el aprendizaje rápido de la robótica como su programación, ya que contamos con máquinas CNC<sup>1</sup> para conformar un CIM<sup>2</sup>.

En cada etapa se realizaron pruebas para concluir cual era el mejor diseño, por ejemplo en la etapa del control se evaluó la curva de velocidad planeada y el paso del servomotor, en la etapa electrónica, se evaluaron las respuestas al ruido de los circuitos, el buen funcionamiento de los microcontroladores y la comunicación entre el PC-Microcontrolador, en la etapa mecánica se seleccionaron los materiales a través de los análisis de elementos finitos.

En esta metodología cada etapa de los procesos esta llevada a cabo por medio de nuevas tecnologías de hardware y software usadas en ingeniería.

Estas tecnologías son:

➤ **Software:**

- **SolidWorks 2004:** Software para diseño asistido por computador CAD. Con este software se busca hacer un diseño del prototipo.
- **Ansys 8.1:** Software para ingeniería asistida por computador CAE. Con este software se hará un análisis de esfuerzos y deformaciones para comprobar el comportamiento del prototipo diseñado con SolidWorks, y la selección adecuada del material de fabricación y los subcomponentes.

---

<sup>1</sup> CNC: Control Numerico Computarizado

<sup>2</sup> CIM: Computer Integrated Manufacturing

- **Mplab:** Software para programación y simulación de microcontroladores. Software para la simulación del algoritmo de control del manipulador.
- **Pic Ide Simulator:** Software para programación y simulación de microcontroladores en lenguaje Basic.
- **Matlab 7.0 :** Software para cálculos, análisis y comunicación entre el PC y el manipulador. Nos ayudara para el calcula de la cinemática y dinámica del manipulador.
- **Eagle 4.14:** Software para el diseño y simulación de circuitos electrónicos. Nos servirá para diseñar y simular la parte electrónica del manipulador.

➤ **Hardware:**

- Servomotores
- Computador
- Quemador universal de Microcontroladores
- Microcontroladores, Transmisores de RF e integrados
- Taller de manufactura de la facultad

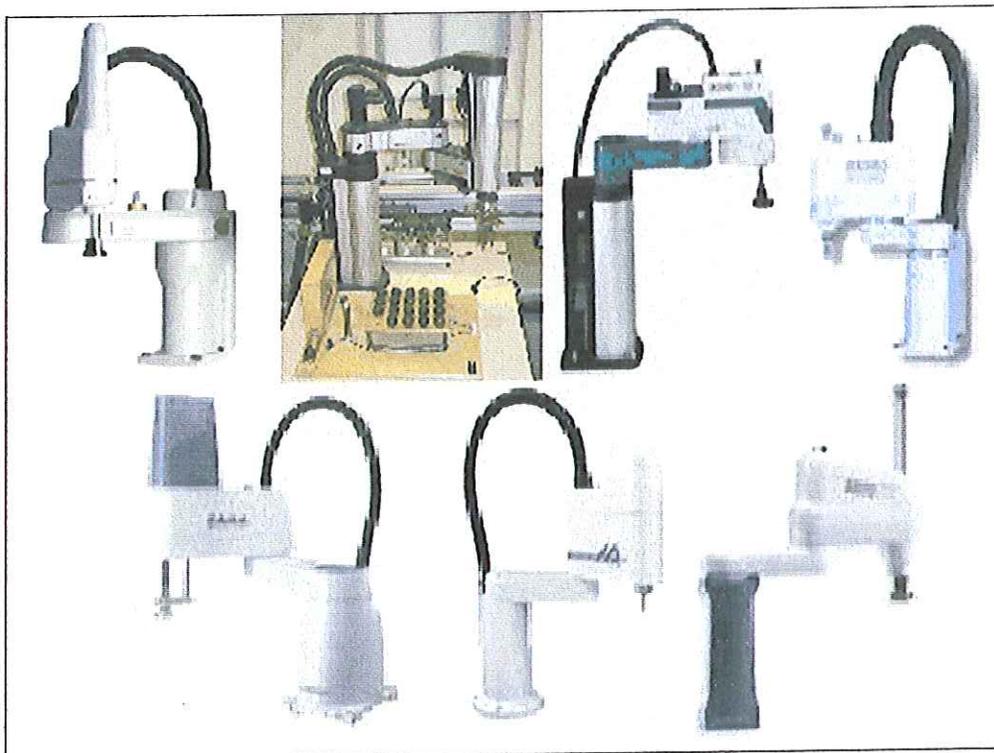
## 2.2 DISEÑO MECATRONICO

### 2.2.1 SCARA (Selective Compliance Assembly Robot Arm)

Estos manipuladores son dispositivos cuya configuración geométrica es angular de tipo cilíndrica, son los más usados en aplicaciones que requieren movimientos rápidos y uniformes; una característica particular es su adaptación selectiva, que es extremadamente útil en operaciones de ensamblaje que requieren la inserción de objetos en cavidades.

Debido a su construcción el Scara es extremadamente rígido en la dirección vertical; pero puede adaptarse lateralmente, facilitando el proceso de inserción. Son relativamente asequibles económicamente.

Figura 20. Diversos manipuladores tipo Scara

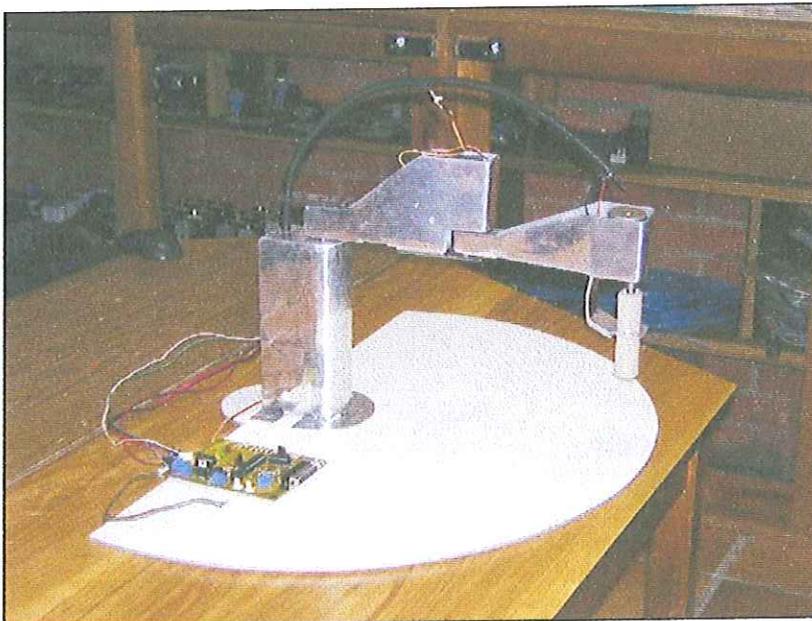


Del Autor

### 2.2.2 EVALUACIÓN DEL MANIPULADOR SCARA DE LA FACULTAD

El manipulador Scara de la facultad, que con anterioridad había sido diseñado y construido por los estudiantes SINDY MELISSA CARRASCAL CORREDOR y LEONARDO GIL AREIZA, como proyecto de grado, sin duda alguna fue en su momento un gran avance en este campo ya que conjugo valiosos conocimientos y un gran aporte para el fortalecimiento de la carrera en la línea de Robótica.

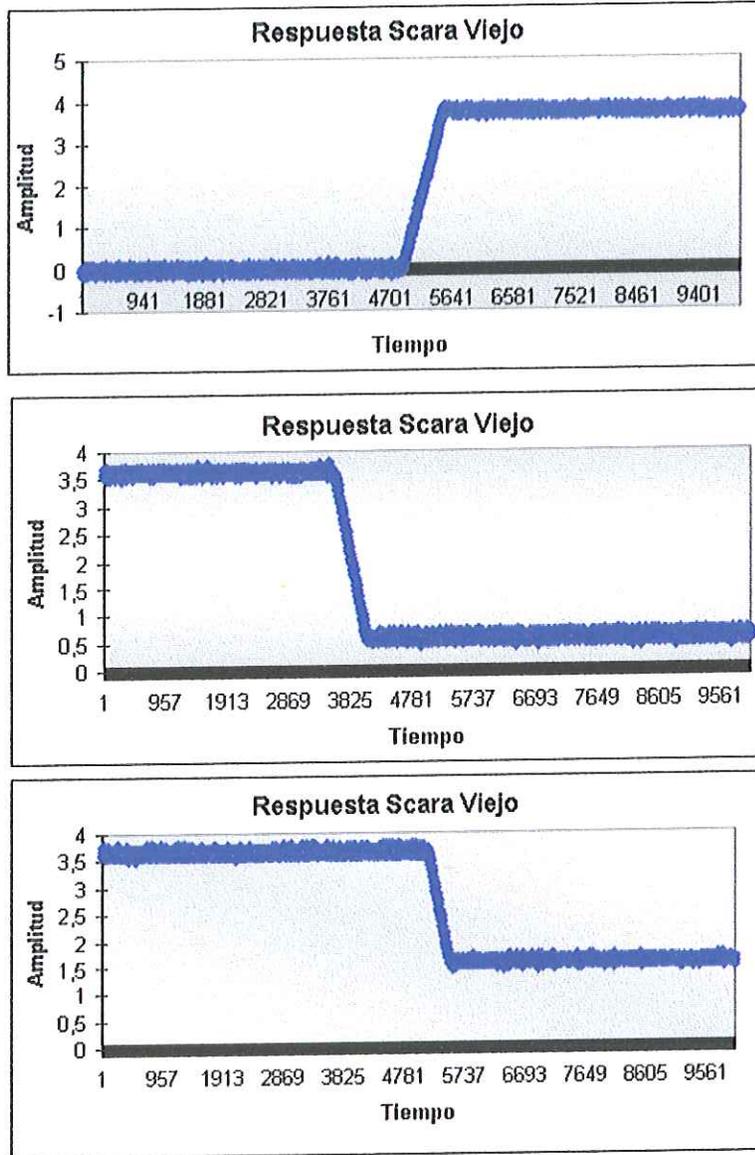
**Figura 21. Manipulador Scara de la facultad**



Tesis de grado, DISEÑO Y CONSTRUCCIÓN DE UN MANIPULADOR TIPO SCARA CON IMPLEMENTACION DE UN SISTEMA DE CONTROL

Al realizar un minucioso estudio y pruebas al manipulador, se observo que las articulaciones efectuaban un movimiento muy acelerado debido a la velocidad de trabajo de los actuadores, y por ello la inercia de las mismas generaba un movimiento vibratorio al posicionarse en el punto deseado, además sin exceptuar que una de ellas presentaba un alto fuego mecánico. De allí que se obtuviera un movimiento irregular del manipulador.

**Figura 22. Pruebas manipulador Scara**



Del Autor

Para realizar las pruebas al manipulador se acoplo un potenciómetro lineal al eje de una articulación con el fin de observar tanto el posicionamiento como la velocidad de movimiento de dicha articulación, verificando la señal de control enviada al actuador. Las graficas obtenidas de las respuesta al punto de consigna es muy rápida lo cual hace que el robot vibre al final del movimiento, el pwm implementado en la tesis anterior no era continuo lo cual hace que el servomotor

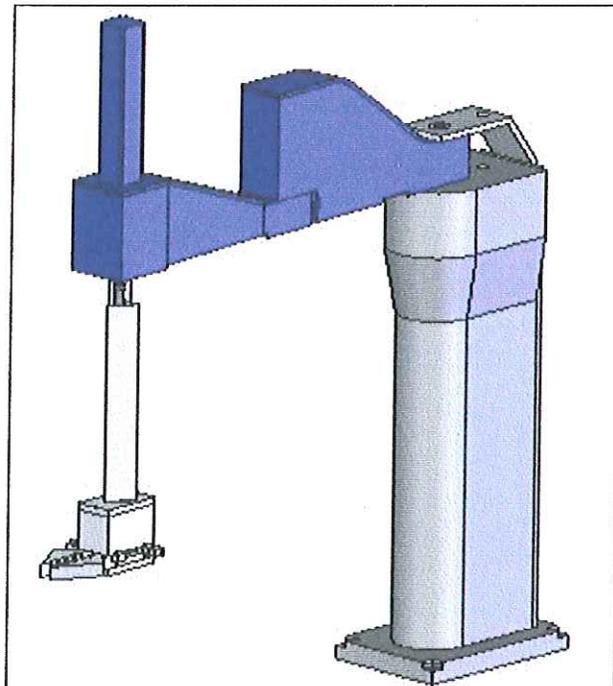
pierda la fuerza y esto a su vez ayuda a que el manipulador no tenga un movimiento bueno.

### 2.2.3 PROPUESTA DE MEJORAMIENTO PARA EL MANIPULADOR SCARA DE LA FACULTAD

En base a los resultados obtenidos anteriormente se comenzó a trabajar en los posibles cambios que se podían realizar para mejorar tanto el movimiento del manipulador como su exterior.

Primero se trabajo en un diseño en el cual se incluyera la construcción de un elemento terminal para la manipulación de objetos, ya que con el que contaba no cumplía con las características necesarias para la tarea que debía realizar el manipulador.

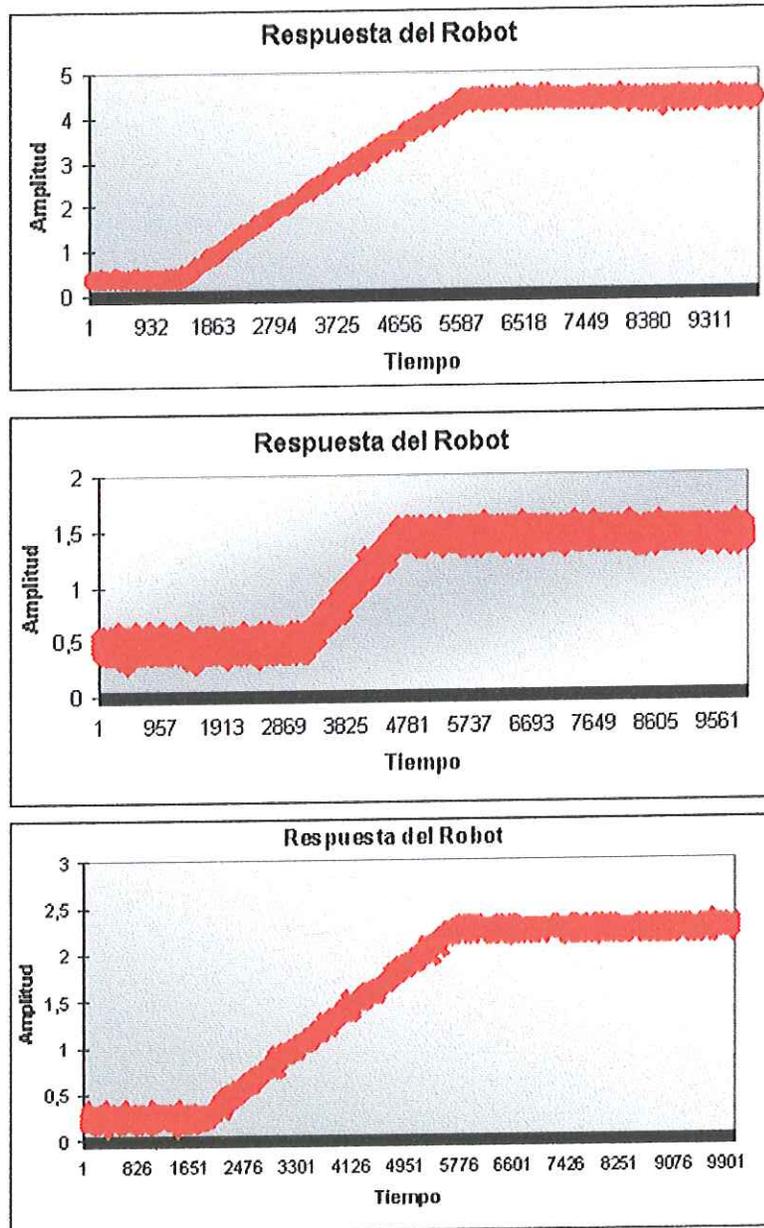
**Figura 23. Rediseño manipulador Scara de la facultad**



Del Autor

Una vez culminada la etapa de diseño, se inicio la implementación del control de inercia a través de mejoras en la parte mecánica y de programación. Estas mejoras incluirían, arreglos de los acoples de las articulaciones, cambio de posición de uno de los actuadores y control de velocidad de los servomotores.

**Figura 24. Pruebas nueva programación**



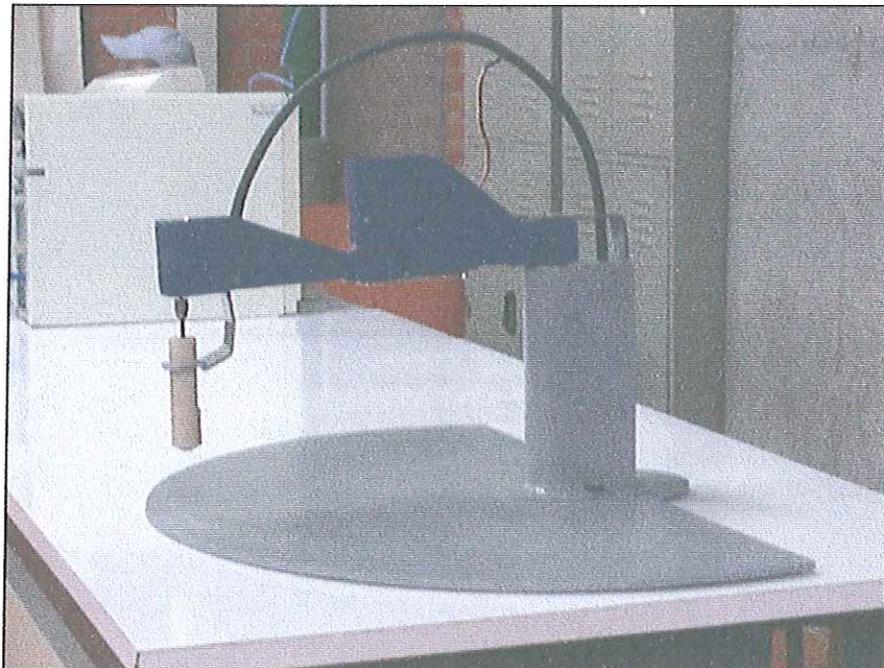
Del Autor

Luego de realizar dichos cambios y pruebas de la nueva programación, se encontraron bastantes mejorías del movimiento de las articulaciones y del posicionamiento de las mismas.

Las graficas obtenidas demuestran que la respuesta al punto de consigna es lenta, esto es por la curva de velocidad implementada, es visible que la curva cambia con respecto al punto de consigna, entre mayor sea la diferencia de las posiciones mas lenta va a hacer la respuesta, el pwm implementado es continuo, con el fin de que el actuador no pierda el troqué para el buen movimiento del manipulador.

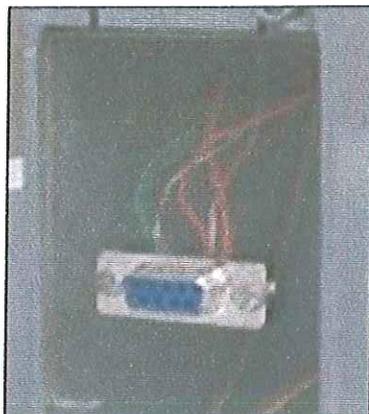
Una vez concluidas las pruebas, se iniciaron las mejoras formales del prototipo y se continúo con la construcción del nuevo manipulador.

**Figura 25. Mejoras formales del manipulador**



Del Autor

**Figura 26. Mejoras de las conexiones de los actuadores**



Del Autor

**Tabla 2. Conexiones del manipulador**

<b>Nº Pin</b>	<b>Señal</b>
1	PWM Servo 1
2	PWM Servo 2
3	Señal Encoder Motorreductor
4	VCC Servomotores
5	VCC-GND Motorreductor
6	GND Servomotores
7	GND-VCC Motorreductor
8	GND Encoder

Del Autor

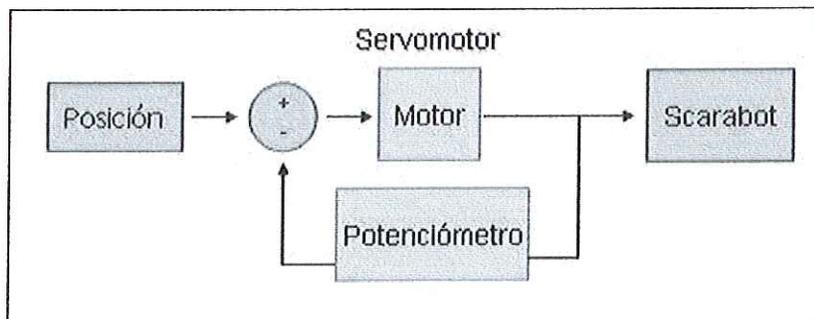
Estas mejoras buscaban brindar facilidad en las conexiones de los actuadores para un mejor control de los mismos por parte de los estudiantes.

## 2.2.4 ETAPA DE CONTROL

El control fue diseñado en base a pruebas realizadas al manipulador Scara diseñado y construido en la tesis de grado **“DISEÑO Y CONSTRUCCIÓN DE UN MANIPULADOR TIPO SCARA CON IMPLEMENTACION DE UN SISTEMA DE CONTROL”**.

De acuerdo a las pruebas realizadas con este prototipo se obtuvieron los siguientes diagramas de bloques, para las diferentes etapas de control.

**Figura 27. Diagrama de bloques Control de Posición**

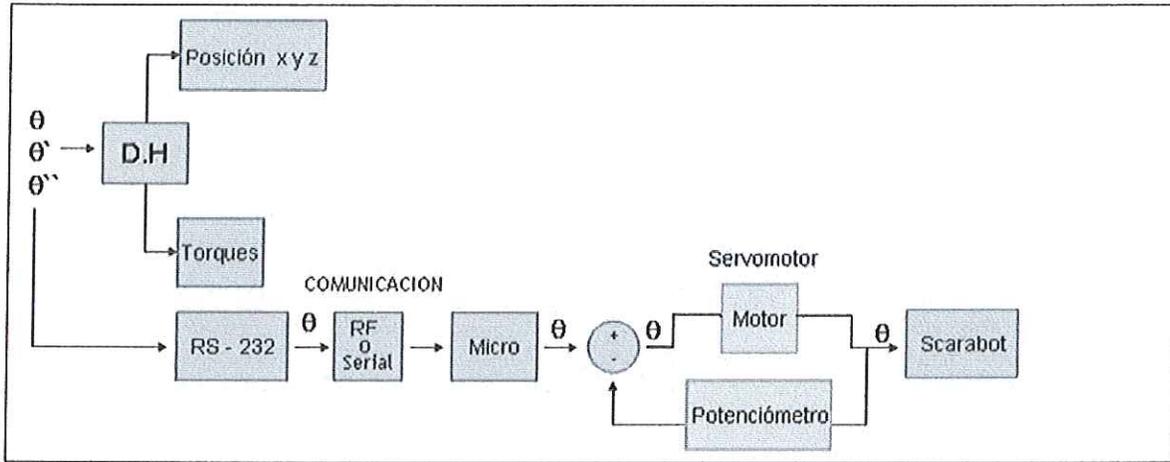


Del Autor

El diagrama de bloques del control de la cinemática, es de alguna manera el más eficaz, ya que los datos ingresados al algoritmo de Denavit Hartemberg, me van a indicar en la posición (X,Y,Z) en la cual se encuentra el manipulador.

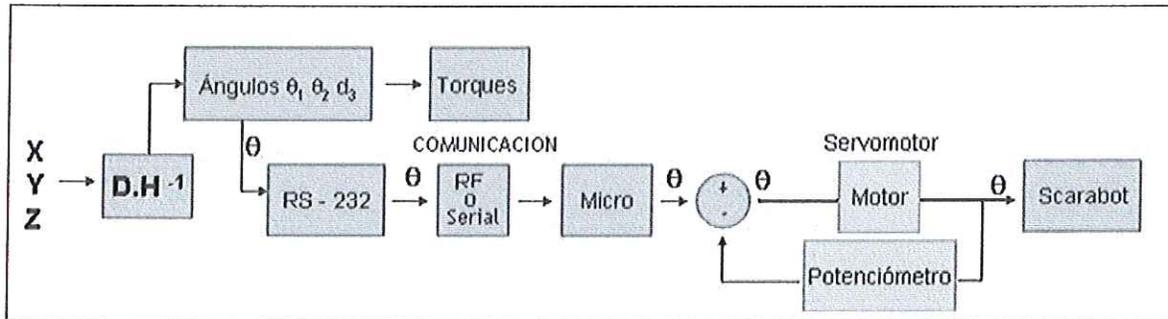
Los diagramas de bloques de control inverso y control de trayectorias realizan el cálculo de los ángulos a mover, por medio de un algoritmo diseñado el cual encuentra la posición más cercana a los puntos dados.

**Figura 28. Diagrama de bloques Control Directo**



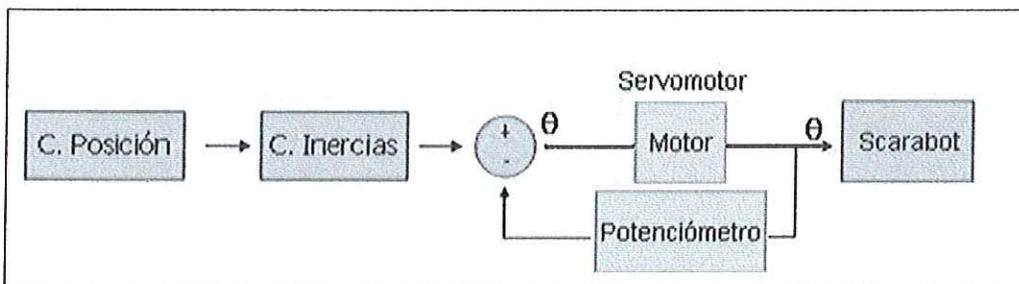
Del Autor

**Figura 29. Diagrama de bloques Control Inverso**



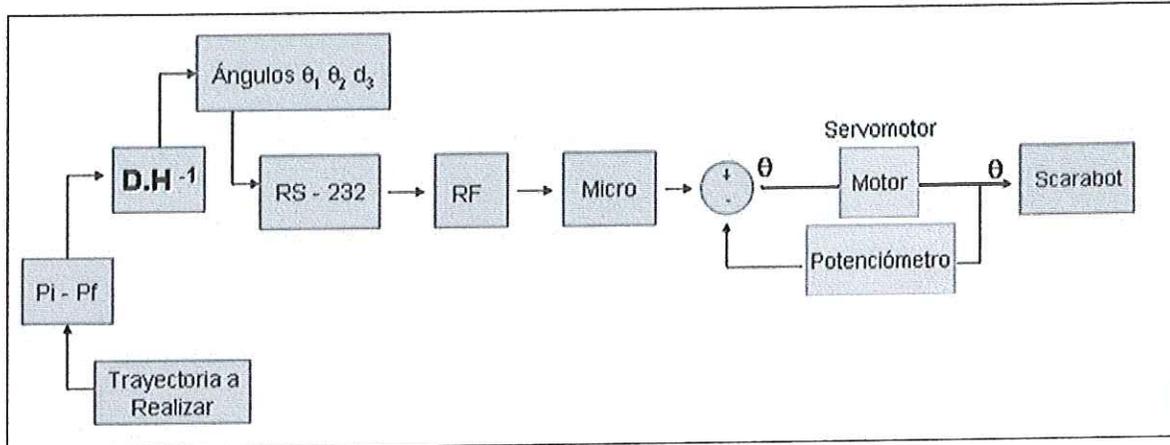
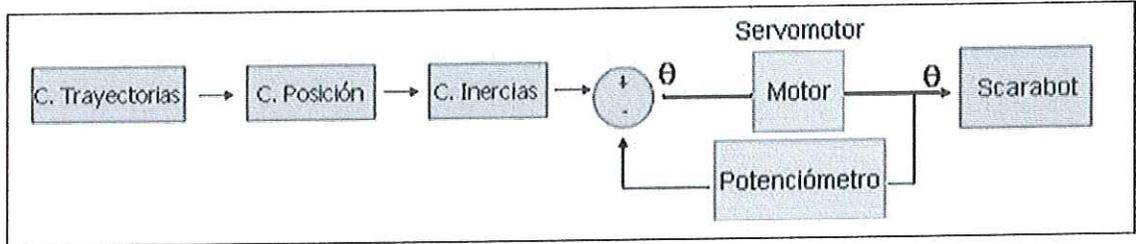
Del Autor

**Figura 30. Diagrama de bloques Control de Inercia**



Del Autor

**Figura 31. Diagrama de bloques Control de Trayectorias**



Del Autor

Los diagramas de control de trayectorias y control inverso son prácticamente los mismos ya que su única diferencia es que el control de trayectorias realiza un calculo de dos posiciones dadas ( $p_i$ = posición inicial,  $p_f$ = posición final), mientras en el inverso solo se calcula una posición dada.

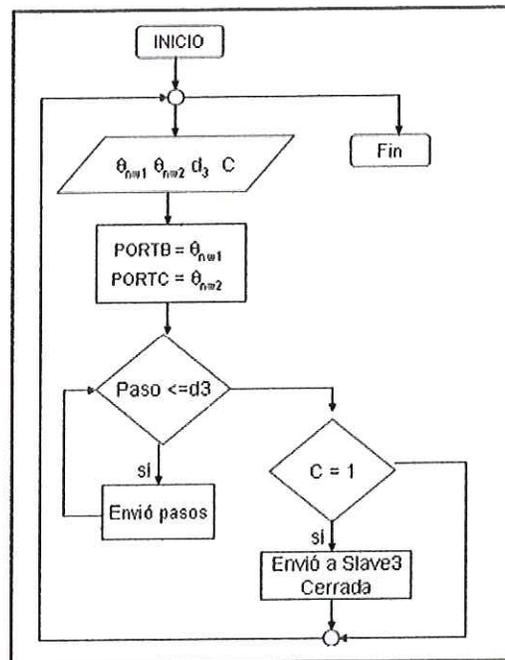
Como se puede observar los sistemas de control del manipulador se encuentran en lazo abierto ya que no hay retroalimentación de la señal de control recibida por el Robot. En cambio los servomotores internamente tienen un control en lazo cerrado el cual nos garantiza la posición de un pwm determinado.

Después de muchas pruebas realizadas para encontrar la programación adecuada que nos permitiera tener un movimiento seguro y sin inercia, se observó que el robot debería tener fuerza en todo momento especialmente en el cargue y

descargue de objetos. La solución planteada fue que los servomotores deberían mantener la señal de control siempre y cuando no hubiera cambio de posición, por lo tanto la tarjeta de control contiene un microcontrolador maestro y 3 esclavos. El microcontrolador maestro se encarga de recibir los datos y controlar el motor paso a paso del elemento *prismático*.

Los microcontroladores esclavos 1 y 2 se encargan de enviar las señales de control a los dos primeros ejes del robot, el microcontrolador esclavo 3 se encarga de enviar la señal de control al servomotor de la pinza.

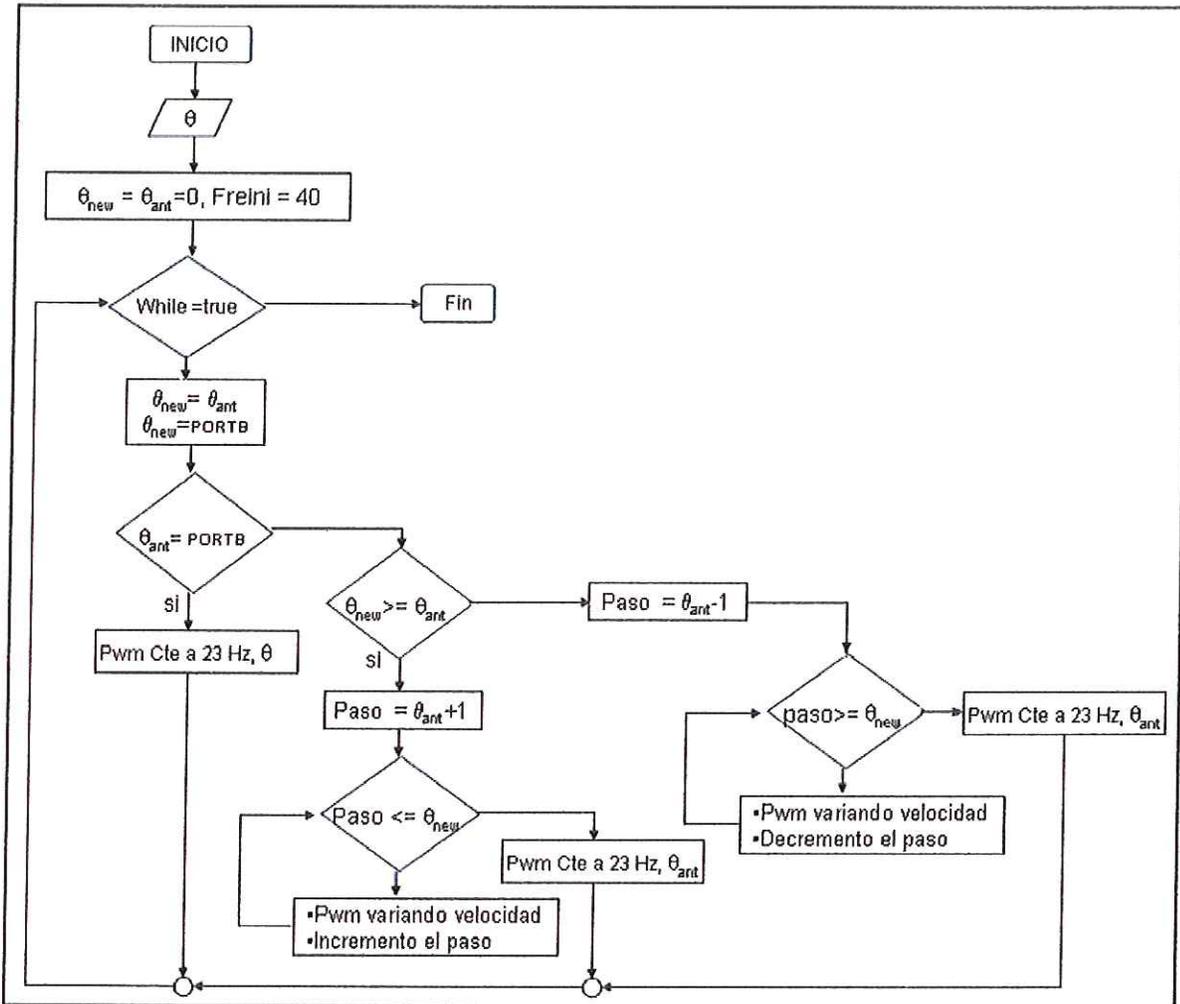
**Figura 32. Algoritmo del Microcontrolador Maestro**



Del Autor

El algoritmo de la figura 32 recibe los ángulos a girar y controla el motoreductor según la distancia de ubicación dada, envía señal de control al microcontrolador 3 el cual realiza el control de cierre de la pinza.

Figura 33. Algoritmo de los Microcontroladores Esclavos

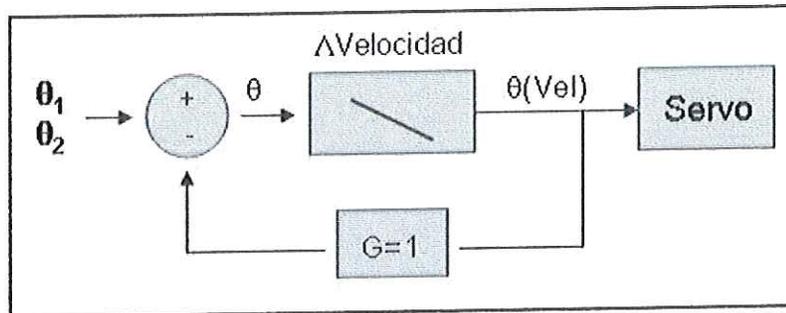


Del Autor

El algoritmo de la figura 33 es quizás el más importante ya que me permite realizar un control de posición y un control de inercia, ya que envía la señal de control necesaria para que el robot se mueva al principio con la máxima velocidad y vaya disminuyendo hasta encontrar la posición enviada por el microcontrolador maestro.

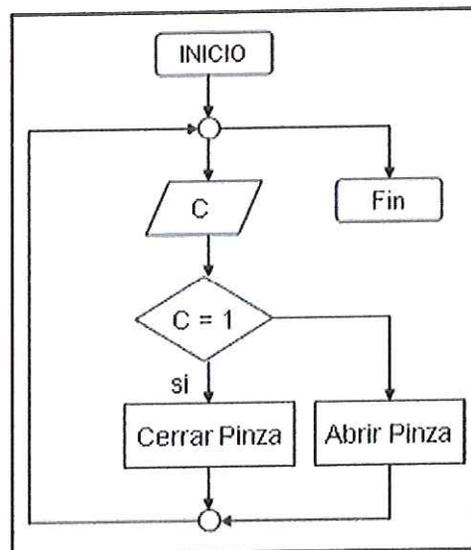
Este algoritmo tiene una caída de velocidad dependiente de la diferencia del ángulo anterior y el ángulo nuevo, ya que el servo se va moviendo grado a grado hasta la posición recibida.

Figura 34. Algoritmo interno de los Microcontroladores Esclavos 1 y 2



Del Autor

Figura 35. Algoritmo del Microcontrolador Esclavo 3.



Del Autor

El algoritmo de la figura 35 recibe una señal de microcontrolador 3 y cierra la pinza a una velocidad constante, manteniendo la señal de control, el cual le permite al servomotor tener buen torque.

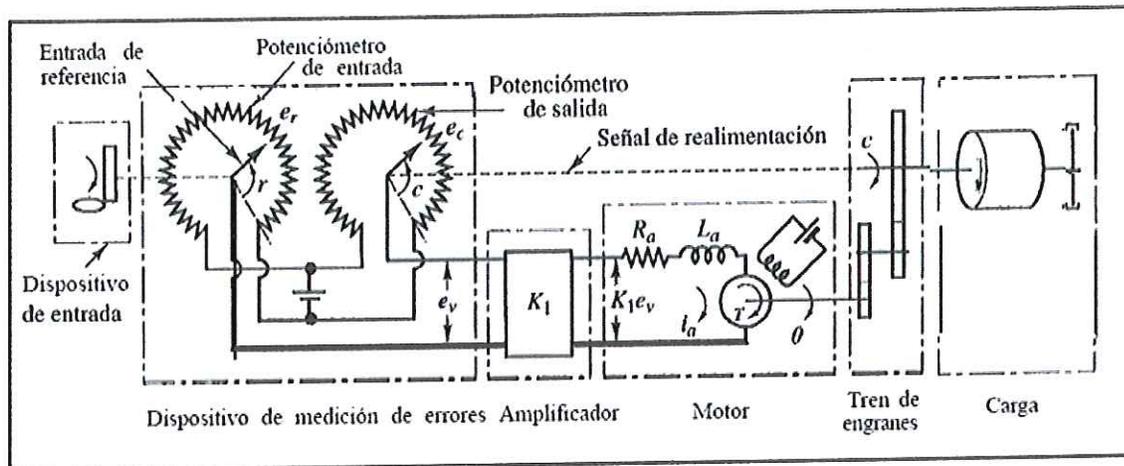
## A. Control de Actuadores

### Control del servomotor

Un servomotor de cd también es manejado por un controlador de movimiento electrónico, con frecuencia denominado servo manejador, como una combinación motor-manejador. El servo manejador determina el movimiento de un servomotor de cd y opera en diversos modos. Algunas de las características son el posicionamiento de punto a punto, el perfilado programable de velocidad y aceleración. El uso de un manejador de movimiento electrónico mediante un manejador de modulación por ancho de pulso para controlar un servomotor de cd se observa con frecuencia en los sistemas de control de robots, de control numérico y otros de posición y/o velocidad. A continuación analizaremos el control de la armadura de servomotores de cd.

El objetivo de este sistema es controlar la posición de la carga mecánica de acuerdo con la posición de referencia. La operación de este sistema es la siguiente: un par de potenciómetros funciona como un dispositivo de medición de error. Convierten las posiciones de entrada y salida en señales eléctricas proporcionales. La señal de entrada de comando determina la posición angular  $r$  del cursor del potenciómetro de entrada. La posición angular  $r$  es la entrada de referencia para el sistema y el potencial eléctrico del cursor es proporcional a la posición angular del cursor. La posición del eje de salida determina la posición angular  $c$  del cursor del potenciómetro de salida. La diferencia entre la posición angular de entrada  $r$  y la posición angular de salida  $c$  es la señal de error  $e$ , o bien.

**Figura 36. Diagrama esquemático de un sistema de seguimiento o servomotor**



Ogata , K. Ingeniería de control moderna.

$$\text{Error} = r - c$$

La diferencia potencial  $e_r - e_c = e_v$  es el voltaje de error, en donde  $e_r$  es proporcional a  $r$  y  $e_c$  es proporcional a  $c$ ; es decir,  $e_r = K_o r$  y  $e_c = K_o c$ , en donde  $K_o$  es una constante de proporcionalidad. El amplificador aumenta el voltaje de error que aparece en las terminales del potenciómetro y su constante de ganancia es  $K_1$ . El voltaje de salida de este amplificador se aplica al circuito de la armadura del motor de cd. (El amplificador debe tener una impedancia de entrada muy alta, debido a que los potenciómetros son esencialmente circuitos de alta impedancia y no toleran una variación de corriente. Al mismo tiempo, el amplificador debe tener una impedancia de salida baja, dado que se alimenta en el circuito de la armadura del motor.) Se aplica un voltaje fijo a la bobina de campo. Si existe un error, el motor desarrolla un par para rotar la carga de salida de tal forma que el error se reduzca a cero.

Para una corriente de campo constante  $i$  el par que desarrolla el motor es:

$$T = K_2 i_a$$

En donde  $K_2$  es la constante de par del motor e  $i_a$  es la corriente de la armadura. Observe que si se invierte el signo de la corriente  $i_a$  el signo del par  $T$  se invierte, y esto provocará que la dirección de giro del rotor se invierta.

Cuando la armadura gira, se induce en ella un voltaje proporcional al producto del flujo y la velocidad angular. Para un flujo constante, el voltaje inducido  $e_b$  es directamente proporcional a la velocidad angular,  $\frac{d\theta}{dt}$  o bien

$$e_b = k_3 \frac{d\theta}{dt}$$

En donde  $e_b$  es la fuerza contraelectromotriz,  $K_3$  es la constante de la fuerza contraelectromotriz del motor y  $\theta$  es el desplazamiento angular de la flecha del motor.

La velocidad del servomotor controlado por armadura está determinada por el voltaje de la armadura  $e_a$ . (El voltaje de la armadura  $e_a = K_1 e_v$  es la salida del amplificador.) La ecuación diferencial para el circuito de la armadura es:

$$L_a \frac{di_a}{dt} + R_a i_a + e_b = e_a$$

Reemplazando se obtiene:

$$L_a \frac{di_a}{dt} + R_a i_a + K_3 \frac{d\theta}{dt} = K_1 e_v$$

La ecuación para el equilibrio de pares es:

$$J_0 \frac{d^2\theta}{dt^2} + b_0 \frac{d\theta}{dt} = T = K_2 i_a$$

En donde  $J_0$  es la inercia de la combinación del motor, de la carga y el tren de engranes referida a la flecha del motor, y  $b_0$  es el coeficiente de fricción viscosa de la combinación del motor, la carga y el tren de engranes referida a la flecha del

motor. La función de transferencia entre el desplazamiento angular de la flecha del motor y el voltaje de error sería la siguiente:

$$\frac{\theta(s)}{E_v(s)} = \frac{K_1 K_2}{s(L_a + R_a)(J_o s + b_o) + K_2 K_3 s}$$

Suponemos que la relación de engranes del tren de engranes es tal que la flecha de salida gira  $n$  veces por cada revolución de la flecha del motor. Por tanto,

En donde  $C(S)$  es el  $C(s) = n\theta(s)$  desplazamiento angular de la flecha de salida. La relación entre  $E_v(S)$ ,  $R(s)$  y  $C(s)$  es:

$$E_v(s) = K_o [R(s) - C(s)] = K_o E(s)$$

La función de transferencia en la trayectoria directa de este sistema es:

$$G(s) = \frac{K_0 K_1 K_2 n}{s[(L_a s + R_a)(J_o s + b_o) + K_3 K_2]}$$

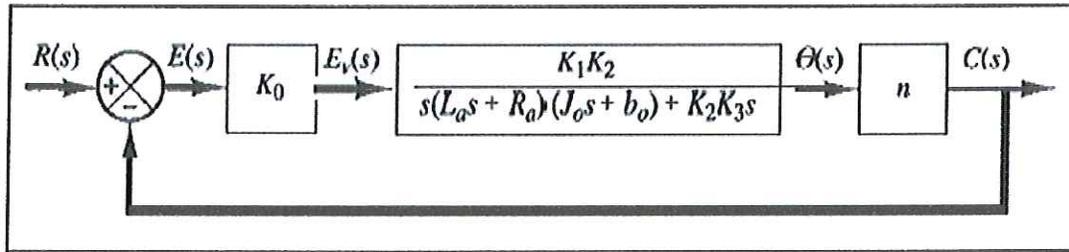
Dado que en la mayoría de los casos la variable  $L_a$  es muy pequeña puede pasarse por alto por lo tanto la función de transferencia sería:

$$G(s) = \frac{K_0 K_1 K_2 n}{s[R_a (J_o s + b_o) + K_3 K_2]}$$

Reacomodando la ecuación quedaría de la siguiente manera:

$$G(s) = \frac{K_0 K_1 K_2 n / R_a}{J_o s^2 + (b_o + \frac{K_3 K_2}{R_a})s}$$

Figura 37. Diagrama de Bloques para el sistema



Ogata , K. Ingeniería de control moderna.

El término  $(b_o + \frac{K_3 K_2}{R_a})s$  indica que la fuerza electromotriz del motor aumenta

Efectivamente la fricción viscosa del sistema. La inercia  $J_o$  y el coeficiente de fricción viscosa  $b_o + (K_2 K_3 / R_a)$  referida a la flecha del motor. Cuando  $J_o$  y  $b_o + (K_2 K_3 / R_a)$  se multiplican por  $1/n^2$ , la inercia y el coeficiente de fricción viscosa se expresan en términos de la flecha de salida. Introduciendo parámetros nuevos mediante:

$$J = J_o / n^2$$

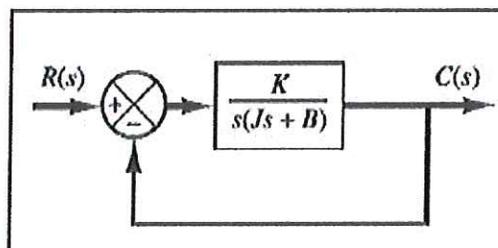
$$B = [b_o + (K_2 K_3 / R_a)] / n^2$$

$$K = K_o K_1 K_2 / n R_a$$

Todas estas ecuaciones son referidas a la flecha de salida, de tal manera que la función transferencia quedaría de la siguiente forma:

$$G(s) = \frac{K}{Js^2 + Bs}$$

Figura 38. Diagrama de bloques simplificado

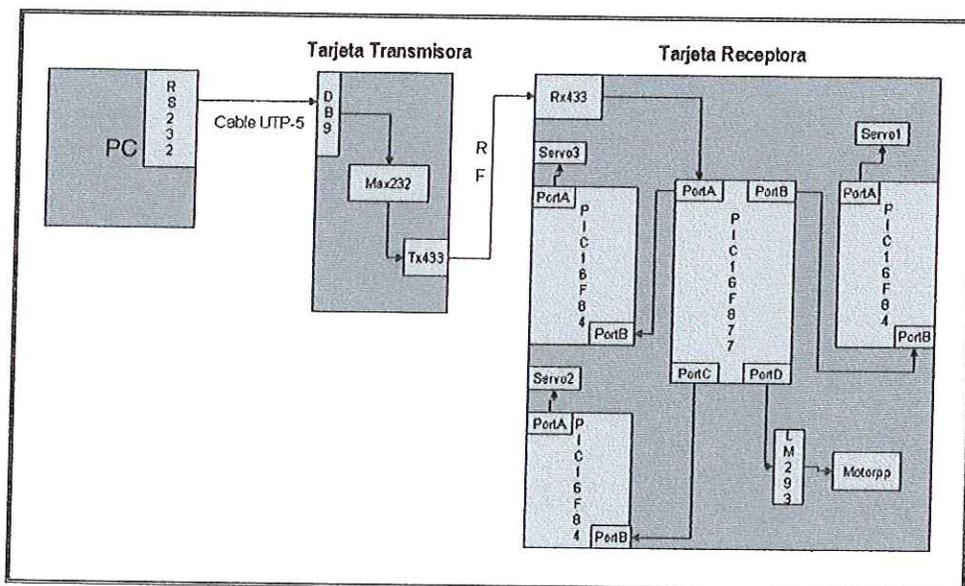


Ogata , K. Ingeniería de control moderna.

## 2.2.5 DISEÑO DE CIRCUITOS ELÉCTRICOS

Para el control del manipulador se diseñaron dos tarjetas electrónicas con funciones específicas, a fin de lograr una buena comunicación y óptimo control del manipulador. Estas fueron diseñadas con ayuda del software EAGLE, el cual posee una amplia librería de elementos e interfaz de fácil manejo.

Figura 39. Diagrama de circuitos eléctricos



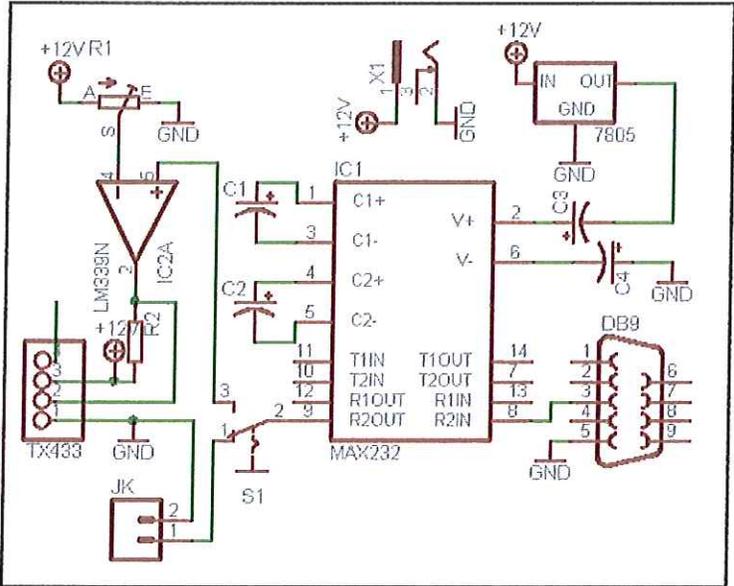
Del Autor

El computador envía los datos por el puerto serial a través de un cable directo el cual es conectado a la tarjeta transmisora, y esta se encarga de enviarlos a la tarjeta receptora mediante señales de radio frecuencia (RF).

El receptor de RF ubicado en la tarjeta es el encargado de suministrar los datos al microcontrolador maestro y este a su vez envía los datos correspondientes a cada microcontrolador esclavo para que estos generen el movimiento de cada una de las articulaciones del manipulador.

Para suministra el voltaje de la alimentación de la tarjeta transmisora se utilizo un adaptador universal de 12V.

Figura 40. Circuito eléctrico Tarjeta Transmisora

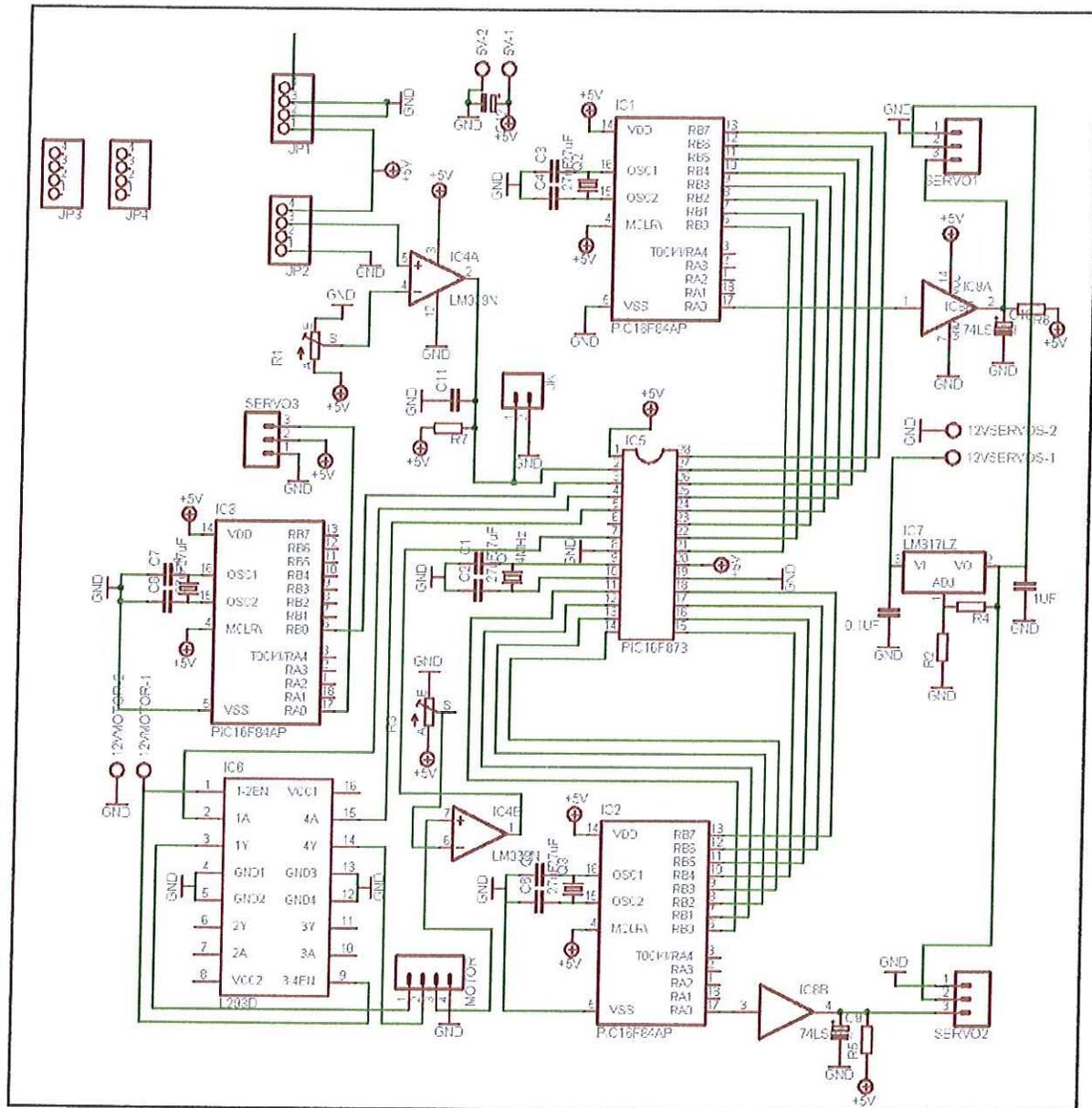


Del Autor

En la tarjeta receptora se requirieron dos fuentes de alimentación, una para alimentar los servomotores y el motoreductor, y la otra para alimentar el circuito de control, ya que debían estar separadas para evitar ruidos generados por los motores.

Dentro de esta tarjeta encontramos una pequeña etapa de potencia la cual se requería para controlar el motoreductor.

Figura 41. Circuito eléctrico Tarjeta Receptora



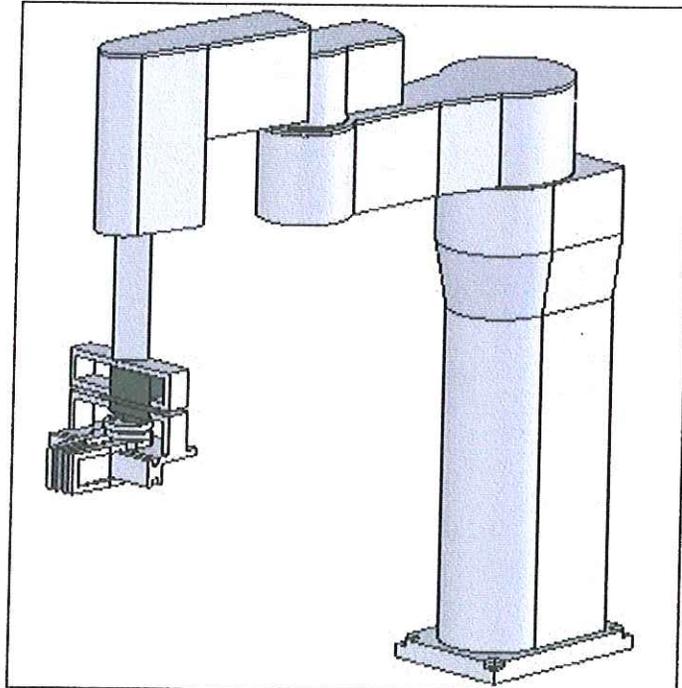
Del Autor

## 2.2.6 DISEÑO DE LA ESTRUCTURA

El diseño de la estructura del manipulador y la selección del tipo de material a utilizar para la misma, es quizás uno de los puntos claves de este proyecto, debido a esto se hicieron estudios preliminares del diseño conceptual del manipulador y con ayuda de una herramienta computacional, el software SolidWorks, se diseño un prototipo del manipulador y se realizo el debido análisis estructural apoyados con el software Ansys Worbench 8.1, con el fin de poner a prueba todos sus componentes y así asegurar un correcto funcionamiento de los mismos.

Para el diseño del manipulador se tuvieron en cuenta dos variables, el área de trabajo y la carga a manipular. El primer diseño propuesto, que fue muy robusto y versátil se encontraron problemas de acople entre cada una de las articulaciones.

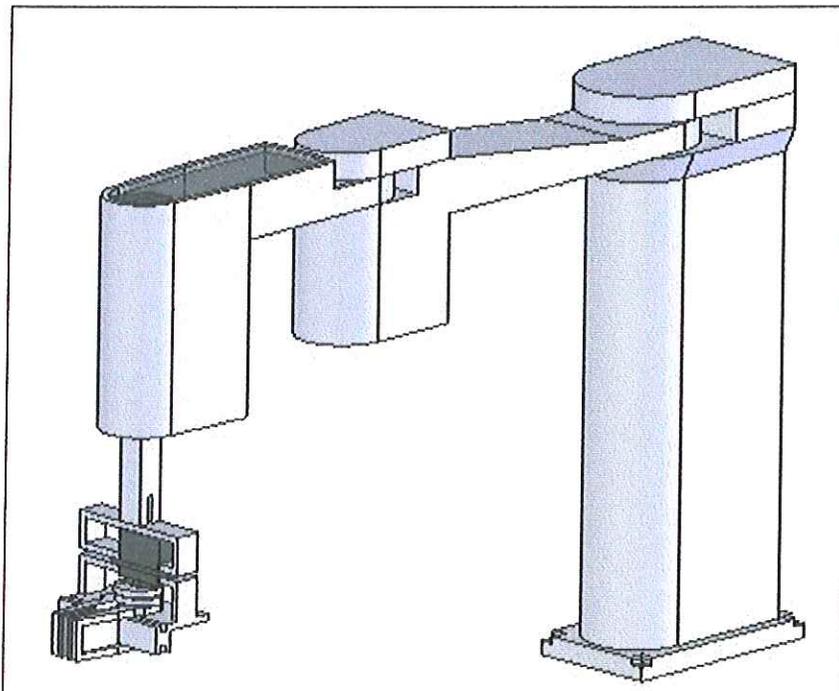
**Figura 42. Primer diseño del manipulador**



Del Autor.

En un segundo diseño se logro una muy buena geometría y estética, pero al realizar los análisis estructurales se encontraron concentraciones de esfuerzos en la segunda articulación o antebrazo, específicamente en los puntos donde había cambio de geometría, además sus articulaciones eran poco robustas y por consiguiente se prosiguió a mejorar este primer diseño.

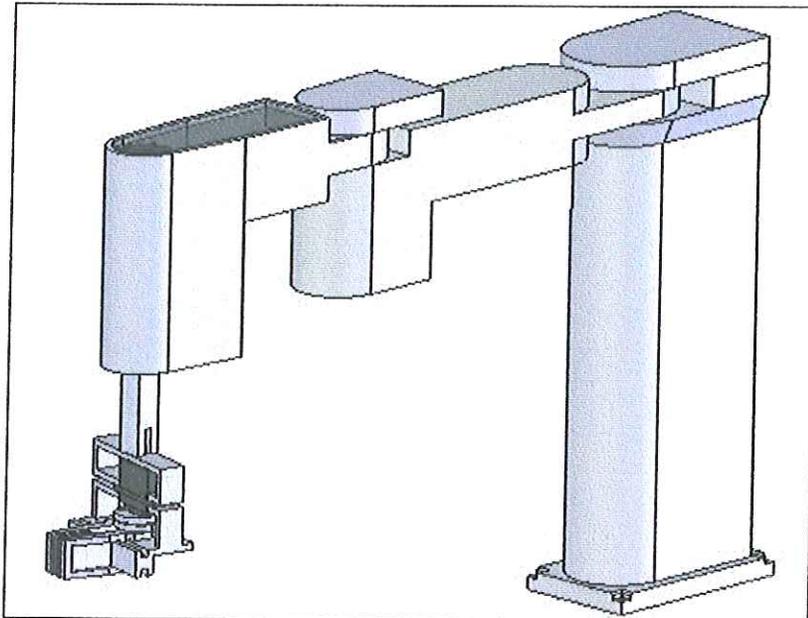
**Figura 43. Segundo diseño del manipulador**



Del Autor

Basados en el primer diseño, se continuo a tratar de hacer sus articulaciones robustas y no tener cambios bruscos de geometría, de acuerdo a esto se realizo un tercer diseño que cumplió con los requerimientos y sus análisis estructurales fueron satisfactorios pero se presento un inconveniente debido a la geometría de la segunda articulación o brazo para su fabricación, ya que el modelo a realizar era complicado.

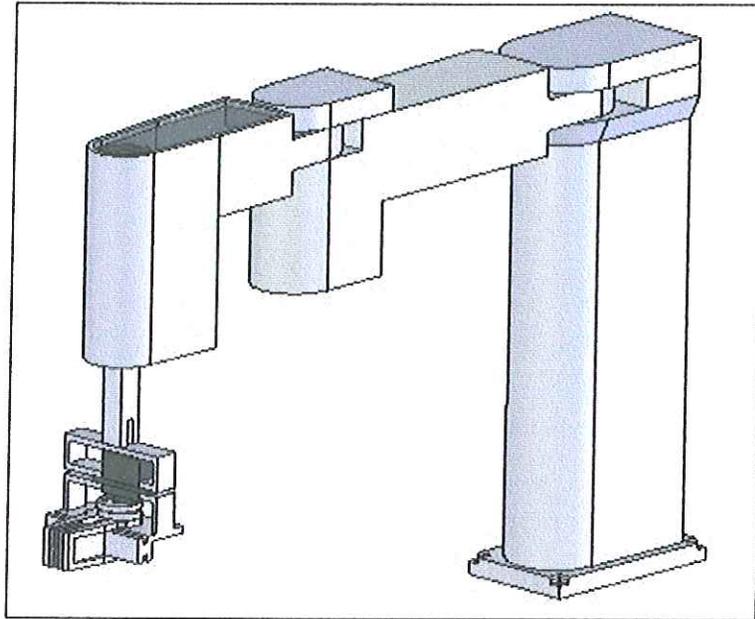
**Figura 44. Tercer diseño del manipulador**



Del Autor.

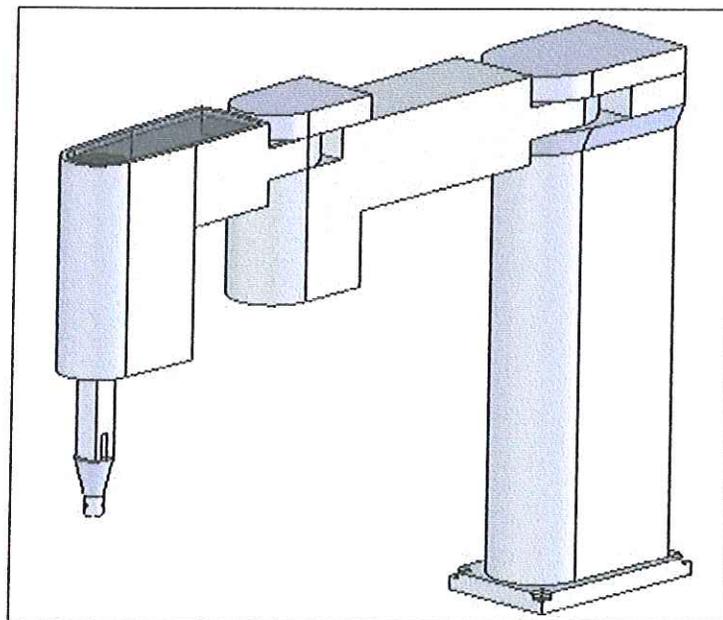
Por consiguiente se realizo un pequeño cambio en dicha articulación y se obtuvieron muy buenos resultados, una vez finalizado todo el diseño y con la correcta aprobación de nuestro director se procedió a seleccionar el tipo de material a utilizar realizando los respectivos análisis estructurales.

**Figura 45. Diseño final del manipulador con pinza**



Del Autor

**Figura 46. Diseño final del manipulador con porta-herramienta**



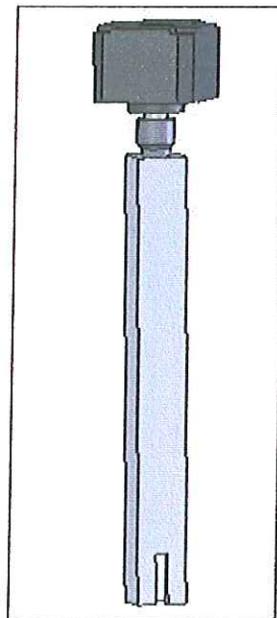
Del Autor

## Diseño del elemento Terminal

Como ya se enunció con anterioridad, la configuración Scara está especialmente diseñada para realizar tareas de montaje en un plano; por esto no es necesario ningún grado de libertad adicional ya que siempre se trabaja en dirección perpendicular al plano de montaje.

El sistema de desplazamiento utilizado para generar el movimiento traslacional (vertical) del elemento Terminal, consiste en un simple mecanismo que consta de una barra cuadrada de aluminio internamente roscada, cuya finalidad es la de una tuerca móvil que permita el desplazamiento vertical con ayuda de un eje roscado, el cual realiza la labor de un tornillo sinfín acoplado al eje del actuador, desplazando la barra a través de la guía. En el momento de accionar el actuador este a su vez mueve el eje roscado que conllevará a que la barra se desplace de manera vertical ya que la guía no permite que se genere movimientos circulares.

**Figura 47. Diseño del sistema de desplazamiento**

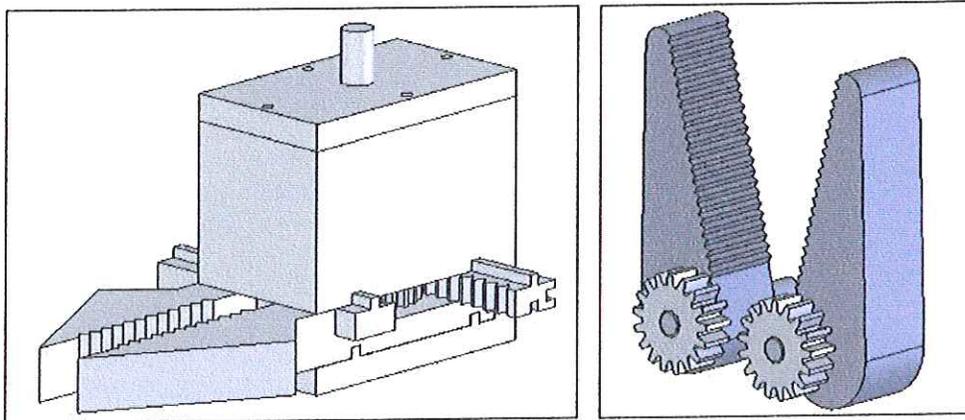


Del Autor

Para el diseño del elemento terminal se tuvieron en cuenta las tareas y trayectorias que debía realizar el manipulador, de este modo se opto por construir una pinza para la manipulación de objetos y otro elemento terminal para acoplar fácilmente cualquier herramienta al momento de generar una trayectoria.

Se realizo un primer diseño mecánico para la pinza, el cual consistía en una pinza paralela accionada a través de un engranaje acoplado al eje del actuador, pero se encontraron problemas de accionamiento, además era muy robusta y poco atractiva.

**Figura 48. Primer y segundo diseño de la pinza del manipulador**

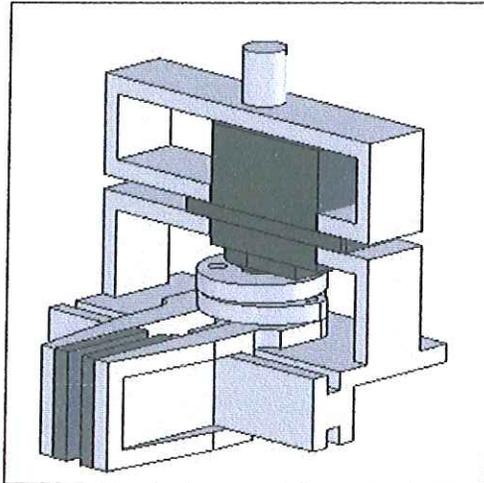


Del Autor

Para el segundo diseño se tuvo en cuenta una pinza angular en la cual el actuador se acoplaba directamente a uno de los dedos y a través de engranajes generaba el movimiento de esta.

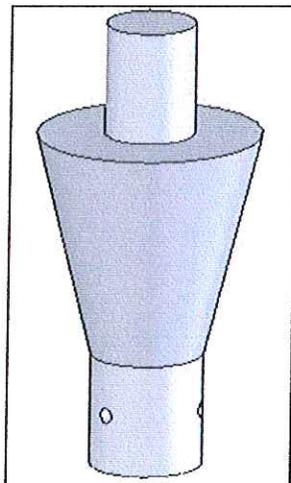
Se opto por realizar un tercer diseño en busca de compararlo con el anterior y decidir cual era más beneficioso al momento de realizar las tareas requeridas por el manipulador. Teniendo como resultado la elección de este ultimo ya que contaba con las características necesarias.

**Figura 49. Diseño final de la pinza del manipulador**



Del Autor

**Figura 50. Diseño del elemento Terminal porta-herramienta**



Del Autor

## 2.2.7 CINEMÁTICA DIRECTA

El problema de la cinemática directa consiste en determinar la posición y orientación de la garra o herramienta del robot desde el punto de vista de un sistema de coordenadas que se toma como referencia. Generalmente el sistema de referencia es el de la base. La cinemática directa es una función de las variables de las articulaciones y parámetros geométricos de la estructura mecánica. Cada articulación contribuye a la estructura mecánica con un grado de movilidad propio de esa articulación.

### Parámetros de Denavit – Hartenberg

Los cuatro parámetros de d-h dependen únicamente de las características geométricas de cada eslabón y de las articulaciones que le unen con el anterior y siguiente. Estos se hallan mediante el desarrollo del algoritmo de D-H, presentado anteriormente.

- $\theta_i$  Es el Angulo que forman los ejes  $X_{i-1}$  y  $X_i$  medido en un plano perpendicular al eje  $Z_i$ , utilizando la regla de la mano derecha. Se trata de un parámetro variable en articulaciones giratorias o rotativas.
- $d_i$  Es la distancia a lo largo del eje  $Z_{i-1}$  desde el origen del sistema coordinado (i-1)-ésimo hasta la intersección del eje  $Z_{i-1}$  con el eje  $X_i$ . Se trata de un parámetro variable en articulaciones prismáticas.
- $A_i$  Es la distancia a lo largo del eje  $X_i$  que va desde la intersección del eje  $Z_{i-1}$ , con el eje  $X_i$  hasta el origen del sistema i-ésimo, en el caso de articulaciones giratorias. En el caso de articulaciones prismáticas, se calcula como la distancia más corta entre los ejes  $Z_{i-1}$  y  $Z_i$ .

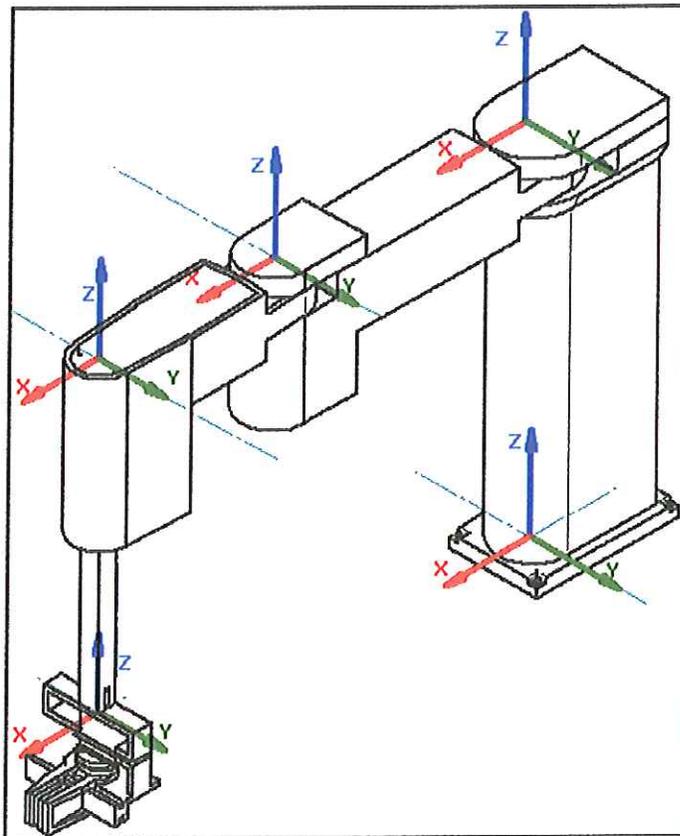
- $\alpha_i$  Es el ángulo de separación del eje  $Z_{i-1}$  y el eje  $Z_i$ , medido en un plano perpendicular al eje  $X_i$ , utilizando la regla de la mano derecha.

**Tabla 3. Parámetros de Denavit Hartenberg del Manipulador Scara**

Articulación	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	$\theta_1$	$l_1$	0	0
2	$\theta_2$	0	$a_2$	0
3	0	0	$a_3$	0
4	0	$-d_1$	0	0

Del Autor

**Figura 51. Sistema de Ejes Coordinados Robot Scara**



Del Autor

Después de haber hallado la tabla de parámetros de Denavit Hartenberg se hallan las matrices de transformación homogéneas:

**Figura 52. Matrices de Transformación Homogénea**

$A_1^0(\theta_1) = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$A_2^1(\theta_2) = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & a_2c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & a_2s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$A_3^2(\theta_3) = \begin{bmatrix} 1 & 0 & 0 & a_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$A_4^3(d_4) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Del Autor

c=coseno ( )

s=seno ( )

El modelo de matemático integrado de la cinemática directa puede escribirse de la siguiente manera:

**Figura 53. Modelo Cinemático integrado**

$T_4^0(\theta) = A_1^0(\theta_1) \times A_2^1(\theta_2) \times A_3^2(\theta_3) \times A_4^3(d_4)$
$T_4^0(\theta) = \begin{bmatrix} n_x(\theta) & s_x(\theta) & a_x(\theta) & p_x \\ n_y(\theta) & s_y(\theta) & a_y(\theta) & p_y \\ n_z(\theta) & s_z(\theta) & a_z(\theta) & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Del Autor

Figura 54. Orientación y posición del Robot

$n_x = \text{Cos}(\theta_1 + \theta_2)$ $n_y = \text{Sin}(\theta_1 + \theta_2)$ $n_z = 0$
$s_x = -\text{Sin}(\theta_1 + \theta_2)$ $s_y = \text{Cos}(\theta_1 + \theta_2)$ $s_z = 0$
$a_x = 0$ $a_y = 0$ $a_z = 1$
$p_x = \text{Cos}(\theta_1 + \theta_2) \times a_3 + \text{Cos}(\theta_1) \times a_2$ $p_y = \text{Sin}(\theta_1 + \theta_2) \times a_3 + \text{Sin}(\theta_1) \times a_2$ $p_z = -d_1 + l_1$

Del Autor

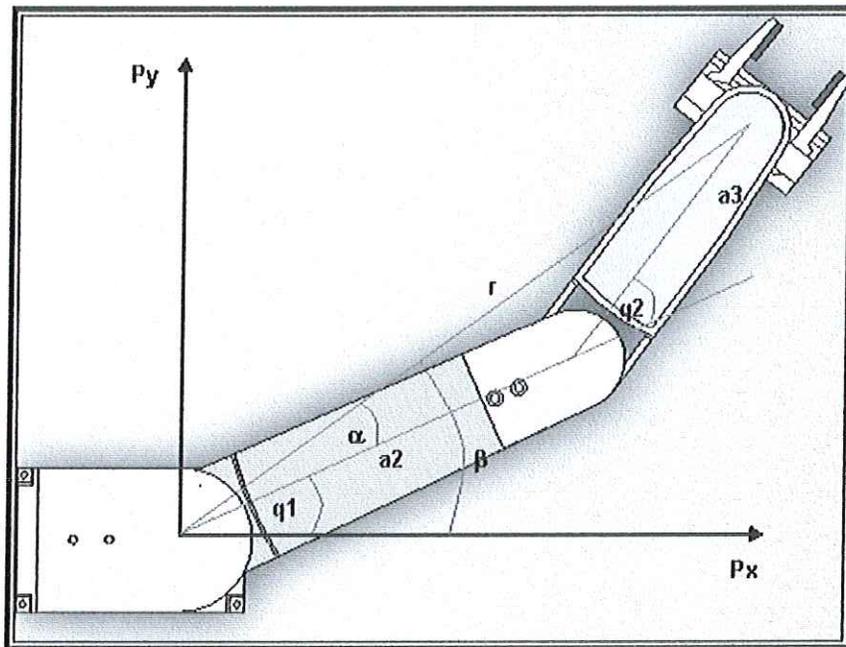
Si la posición y orientación del sistema de coordenadas de la garra o herramienta del robot  $(X, Y, Z)$  calculadas no es correcta, entonces, se crea una MTH de ajuste, porque el sistema mecánico del robot puede generar errores y el sistema electrónico puede generar errores en la etapa de montaje, en este caso, el robot necesita calibración para ajustar sus parámetros antes de entrar en uso.

## 2.2.8 CINEMÁTICA INVERSA

Para esta solución utilizamos varios recursos los cuales nos permitieron una precisión aceptable.

Primero analizamos la solución del problema de la cinemática inversa por métodos geométricos, lo cual nos ayudo a encontrar dos variables la magnitud del ángulo  $\theta_2$ , pero no su signo y la altura del gripper  $d_1$ .

Figura 55. Figura para el análisis del método geométrico



Del autor

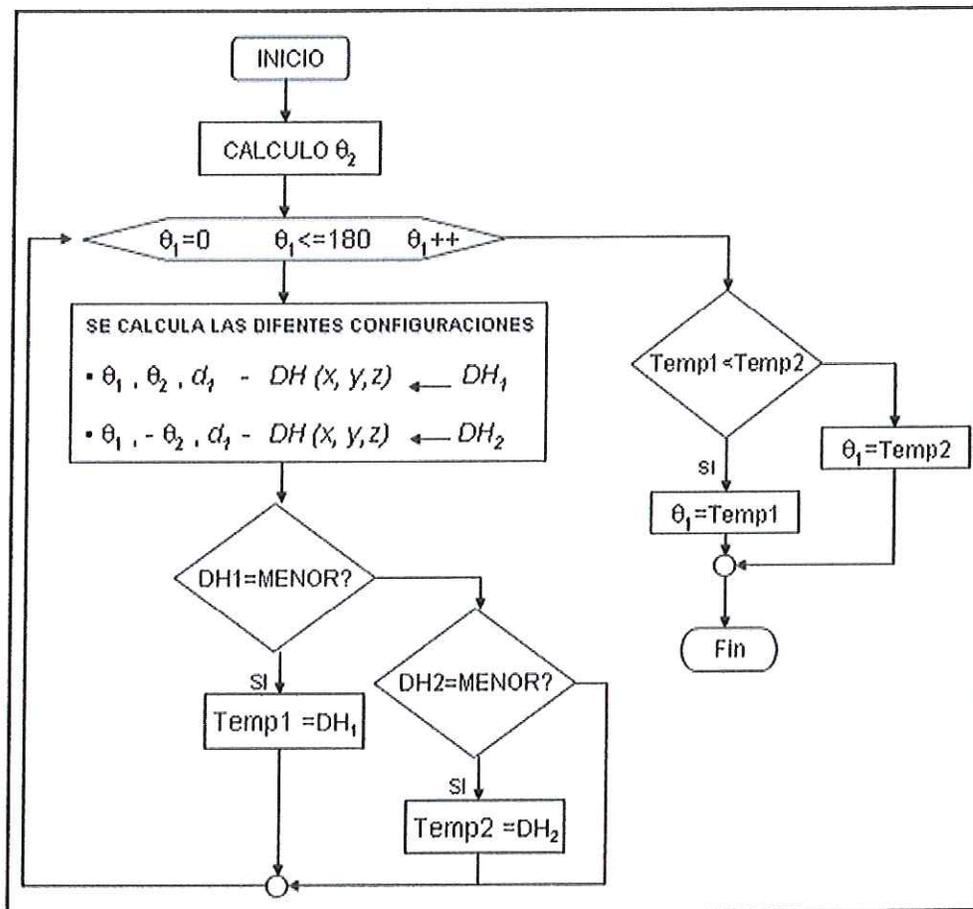
$$r^2 = px^2 + py^2$$

$$\cos \theta_2 = \frac{r^2 - a_2^2 - a_3^2}{2a_2a_3} \quad d_1 = l_1 - p_z$$

$$\theta_2 = \arctg\left(\frac{\sqrt{1 - \cos^2 \theta_2}}{\cos \theta_2}\right)$$

Luego de haber hallado el ángulo  $\theta_2$  realizamos un algoritmo para encontrar el ángulo  $\theta_1$  y de las posibles soluciones escoger la de menor distancia con respecto del punto X, Y Z dado.

**Figura 56. Algoritmo de solución de la Cinemática Inversa**



Del autor

Como se puede observar primero hallamos el ángulo  $\theta_2$ , luego calculamos el Dh para la posición X, Y,Z dada, y comenzamos a comparar variando el ángulo  $\theta_1$ , con las diferentes configuraciones ósea con  $\theta_2$ , positivo y negativo, hasta encontrar el ángulo  $\theta_1$  mas cercano al punto dado.

## 2.2.9 MATRIZ JACOBIANA DIRECTA

Para encontrar la matriz jacobiana se deriva parcialmente con respecto a cada una de las variables a  $\theta_1, \theta_2, d_1$ , quedando de la siguiente manera.

Figura 57. Variables de la Matriz Jacobiana

$$J = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial d_1} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial d_1} \\ \frac{\partial z}{\partial \theta_1} & \frac{\partial z}{\partial \theta_2} & \frac{\partial z}{\partial d_1} \end{bmatrix}$$

Del Autor

Figura 58. Componentes de la Matriz jacobina

$$\begin{array}{l} \frac{\partial x}{\partial \theta_1} = -a_3 \sin(\theta_1 + \theta_2) - a_2 \sin(\theta_1) \\ \frac{\partial x}{\partial \theta_2} = -a_3 \sin(\theta_1 + \theta_2) \\ \frac{\partial x}{\partial d_1} = 0 \\ \hline \frac{\partial y}{\partial \theta_1} = a_3 \cos(\theta_1 + \theta_2) + a_2 \cos(\theta_1) \\ \frac{\partial y}{\partial \theta_2} = a_3 \cos(\theta_1 + \theta_2) \\ \frac{\partial y}{\partial d_1} = 0 \\ \hline \frac{\partial z}{\partial \theta_1} = 0 \quad \frac{\partial z}{\partial \theta_2} = 0 \quad \frac{\partial z}{\partial d_1} = -1 \end{array}$$

Del Autor

## 2.2.10 MATRIZ JACOBIANA INVERSA

Para realizar el cálculo de la matriz jacobiana inversa, primero efectuamos el de la matriz jacobiana directa, después de obtenida la directa efectuamos la inversa para encontrar las velocidades articulares del manipulador.

$$I = J^{-1}$$

Figura 59. Matriz jacobiana Inversa

$$J^{-1} = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial y}{\partial \theta_1} & \frac{\partial z}{\partial \theta_1} \\ \frac{\partial x}{\partial \theta_2} & \frac{\partial y}{\partial \theta_2} & \frac{\partial z}{\partial \theta_2} \\ \frac{\partial x}{\partial d_1} & \frac{\partial y}{\partial d_1} & \frac{\partial z}{\partial d_1} \end{bmatrix}$$

Del Autor

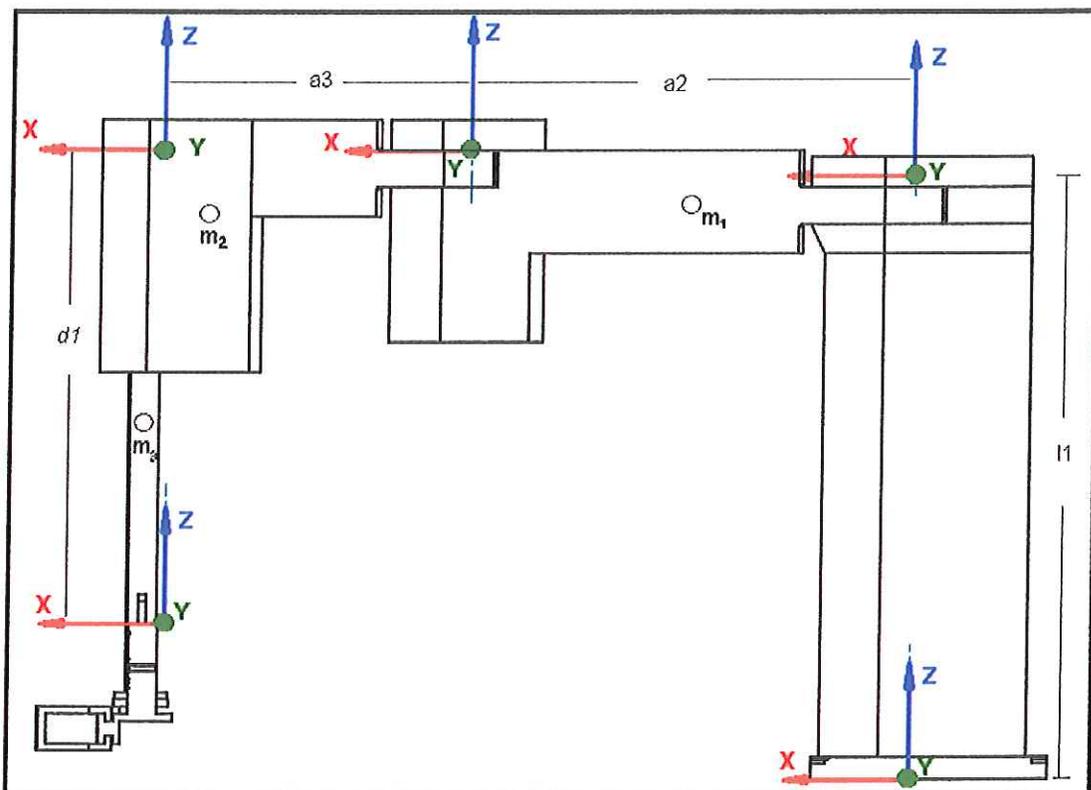
Como se puede observar en la figura, se muestra a que equivale cada componente de la matriz jacobiana.

## 2.2.11 DINÁMICA

Existen dos métodos de realizar este modelado matemático, el más común y más utilizado es el método de Newton – Euler, ya que conduce a un costo computacional mucho menor que el método de Lagrange – Euler. Es decir que el método de Newton- Euler produce menos operaciones que el de Lagrange – Euler, aclarando que la solución por el método de Lagrange - Euler tiene una serie de supuestos que hacen más fácil la solución.

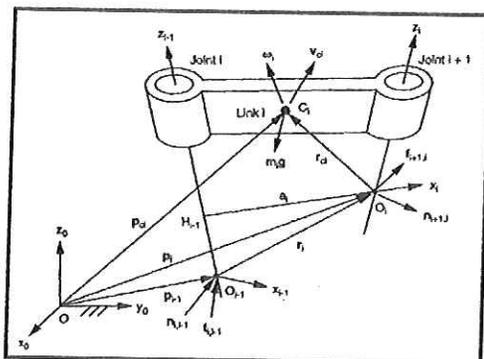
### 2.2.11.1 Método de Solución por Newton – Euler

Figura 60. Diagrama esquemático del robot Scara



Del Autor

Figura 61. Fuerzas y Momentos ejercidos en el eslabón i



Lung-Wen Tsai. Robot Analysis

Tabla 4. Tabla de Variables de Newton Euler

Variable	Descripción
$f_{i,i-1}$	Fuerza ejercida en el eslabón i por i-1 en el punto $O_{i-1}$
$f_i$	Fuerza de inercia ejercida en el centro de masa del eslabón i
$I_i$	Matriz de Inercia del eslabón i respecto de su centro de masa expresado en $S_i$
$n_{i,i-1}$	Momento ejercido en el eslabón i por i-1 en el punto $O_{i-1}$
$n_i$	Momento de inercia ejercido en el centro de masa del eslabón i
$p_i$	Vector de posición del origen del eslabon $S_i$ con respecto del eslabon base, $p_i = O_0 O_i$
$p_{ci}$	Vector de posición del centro de masa del eslabon $S_i$ con respecto del eslabon base, $p_{ci} = O_0 C_i$
$r_i$	Vector de posición del origen del eslabon $S_i$ con respecto del eslabon $S(i-1)$ , $r_i = O_{i-1} O_i$
$r_{ci}$	Vector de posición del centro de masa del eslabon i con respecto del eslabon $S_i$ , $r_{ci} = O_i C_i$
$v_i$	Velocidad lineal absoluta del Origen $O_i$
$v_{ci}$	Velocidad lineal absoluta del centro de masa del eslabón i
$\dot{v}_i$	Aceleración lineal absoluta del Origen $O_i$
$\dot{v}_{ci}$	Aceleración lineal absoluta del centro de masa del eslabón i
$z_i$	Vector de la unidad que señala a lo largo de $Z_i$
$w_i$	Velocidad angular absoluta del eslabón i
$\dot{w}_i$	Aceleración angular absoluta del eslabón i

Lung-Wen Tsai. Robot Analysis

## Algoritmo Dinámico por el método de Newton –Euler

- Obtener La velocidad angular del sistema  $S_i$ .

$${}^i\omega_i = \begin{cases} {}^iR_{i-1}({}^{i-1}\omega_{i-1} + Z_o\dot{\theta}_i) & \text{Si el eslabón } i \text{ es de rotación.} \\ {}^iR_{i-1}{}^{i-1}\omega_{i-1} & \text{Si el eslabón } i \text{ es de traslación.} \end{cases}$$

Donde:

$${}^iR_{i-1} = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\cos\alpha_i\sin\theta_i & \cos\alpha_i\cos\theta_i & \sin\theta_i \\ \sin\alpha_i\sin\theta_i & -\sin\alpha_i\cos\theta_i & \cos\theta_i \end{bmatrix}$$

- Obtener la aceleración angular del sistema  $S_i$ .

$${}^i\dot{\omega}_i = \begin{cases} {}^iR_{i-1}({}^{i-1}\dot{\omega}_{i-1} + Z_o\ddot{\theta}_i) + {}^{i-1}\omega_{i-1} \times Z_o\dot{\theta}_i & \text{Si el eslabón } i \text{ es de rotación.} \\ {}^iR_{i-1}{}^{i-1}\dot{\omega}_{i-1} & \text{Si el eslabón } i \text{ es de traslación.} \end{cases}$$

- Obtener la Velocidad Lineal

$${}^i v_i = \begin{cases} {}^iR_{i-1}{}^{i-1}v_{i-1} + {}^i p_i \times {}^i\omega_i & \text{Si el eslabón } i \text{ es de rotación.} \\ {}^iR_{i-1}({}^{i-1}v_{i-1} + Z_o\dot{\theta}_i) + {}^i p_i \times {}^i\omega_i & \text{Si el eslabón } i \text{ es de traslación} \end{cases}$$

Donde:

$${}^i p_i = \begin{bmatrix} a_i \\ d_i \sin\alpha_i \\ d_i \cos\alpha_i \end{bmatrix}$$

- Obtener la aceleración lineal del sistema i.

$${}^i\dot{v}_i = \begin{cases} {}^i\dot{\omega}_i \times {}^i p_i + {}^i\omega_i \times ({}^i p_i \times {}^i\omega_i) + {}^i R_{i-1} {}^{i-1}\dot{v}_{i-1} & \text{Si } i \text{ es de rotación} \\ {}^i R_{i-1} (Z_0 \dot{c}_i + {}^{i-1}\dot{v}_{i-1}) + {}^i\dot{\omega}_i \times {}^i p_i + 2 {}^i\omega_i \times {}^i R_{i-1} Z_0 \dot{c}_i + {}^i\omega_i \times ({}^i p_i \times {}^i\omega_i) & \text{Si } i \text{ es de traslación.} \end{cases}$$

- Obtener la aceleración lineal del centro de gravedad del eslabón i.

$${}^i a_i = {}^i\dot{\omega}_i \times {}^i S_i + {}^i\omega_i \times ({}^i\omega_i \times {}^i S_i) + {}^i\dot{v}_i$$

- Obtener la fuerza ejercida sobre el eslabón i.

$${}^i f_i = {}^i R_{i+1} {}^{i+1} f_{i+1} + m_i {}^i a_i$$

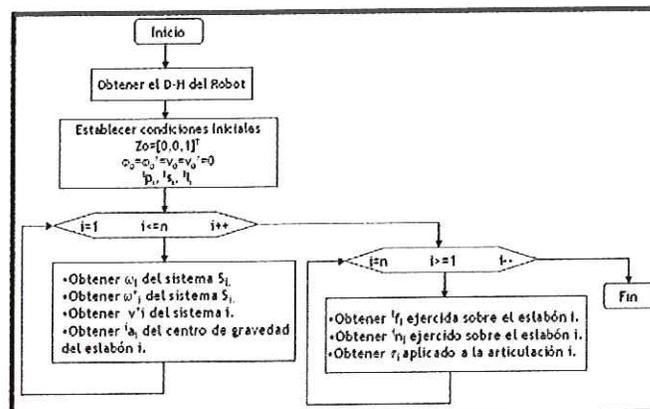
- Obtener el par ejercido sobre el eslabón i.

$${}^i n_i = {}^i R_{i+1} [{}^{i+1} n_i + ({}^{i+1} R_i {}^i p_i) \times {}^{i+1} f_{i+1}] + ({}^i p_i + {}^i S_i) \times m_i {}^i a_i + {}^i L_i {}^i \dot{\omega}_i + {}^i\omega_i \times ({}^i L_i {}^i\omega_i)$$

- Obtener la fuerza o par aplicado a la articulación i.

$$\tau_i = \begin{cases} {}^i n_i^T {}^i R_{i-1} Z_0 & \text{Si el eslabón } i \text{ es de rotación.} \\ {}^i f_i^T {}^i R_{i-1} Z_0 & \text{Si el eslabón } i \text{ es de traslación.} \end{cases}$$

Figura 62. Algoritmo Dinámico e Newton – Euler



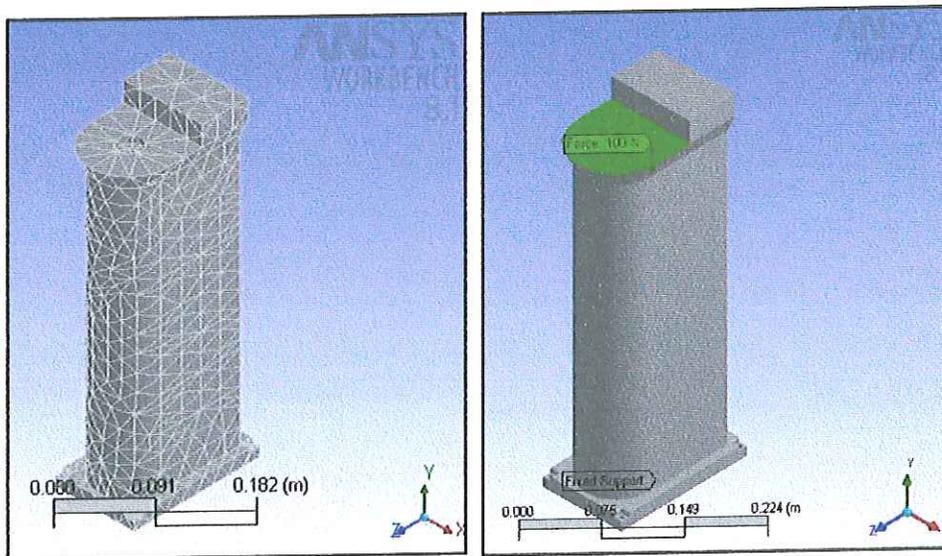
Del Autor

## 2.2.12 ANÁLISIS DEL DISEÑO DEL MANIPULADOR Y SELECCIÓN DE MATERIALES

El análisis estructural se realizó con ayuda del software Ansys Workbench 8.1, el cual es una herramienta para simulación por elementos finitos que permite simular cargas y esfuerzos en estructuras modeladas en CAD.

Para el diseño final del manipulador se obtuvieron los siguientes resultados:

**Figura 63. Enmallado y Fuerza aplicada a la Base**

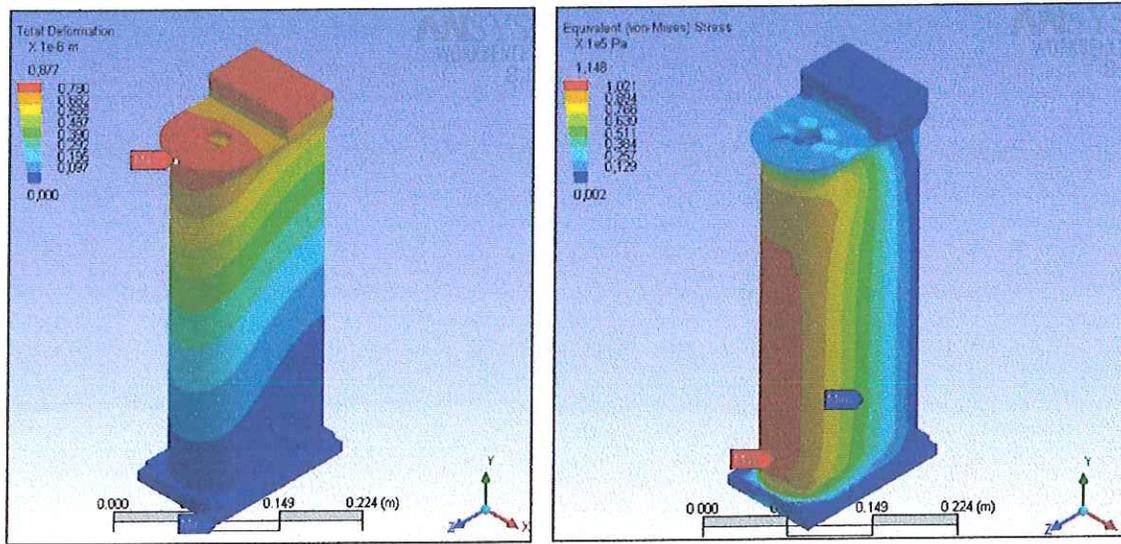


Del Autor

Geometría de la pieza

Name	Material	Bounding Box (m)	Mass (kg)	Volume (m <sup>3</sup> )	Nodes	Elements
"base"	"Aluminum Alloy"	0.1, 0.4, 0.16	3.21	1.16×10 <sup>-3</sup>	6537	3227

**Figura 64. Deformación y esfuerzo equivalente de la Base**

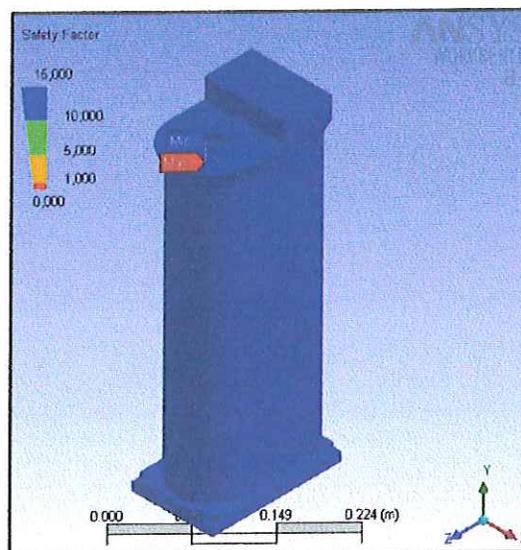


Del Autor

Resultados estructurales

Name	Scope	Minimum	Maximum	Alert Criteria
"Total Deformation"	"Model"	0.0 m	$8.77 \times 10^{-7}$ m	None
"Equivalent Stress"	"Model"	181.45 Pa	114,849.72 Pa	None

**Figura 65. Factor de seguridad de la Base**

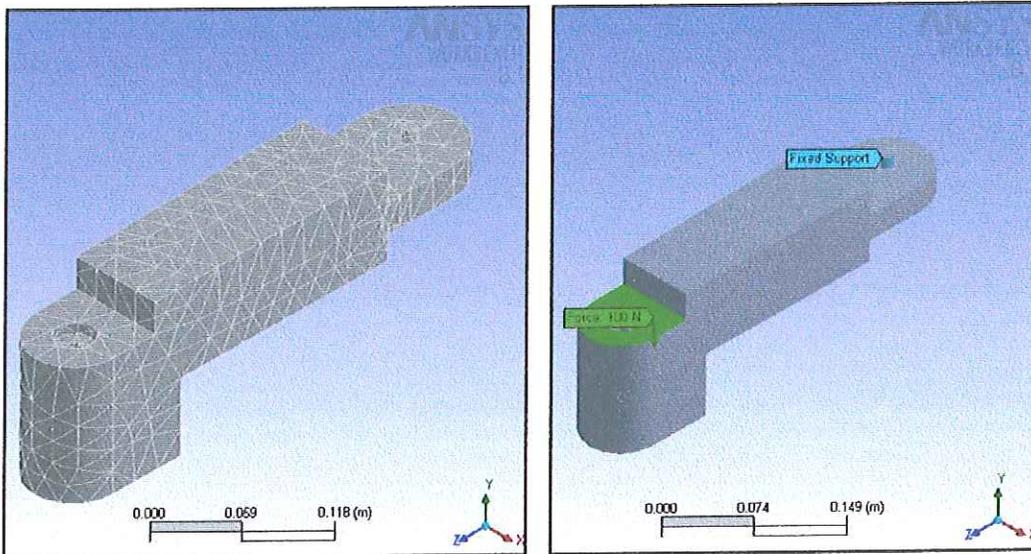


Del Autor

Factor de seguridad

Name	Scope	Type	Minimum	Alert Criteria
"Stress Tool"	"Model"	Safety Factor	15.0	None

Figura 66. Enmallado y fuerza aplicada al Antebrazo

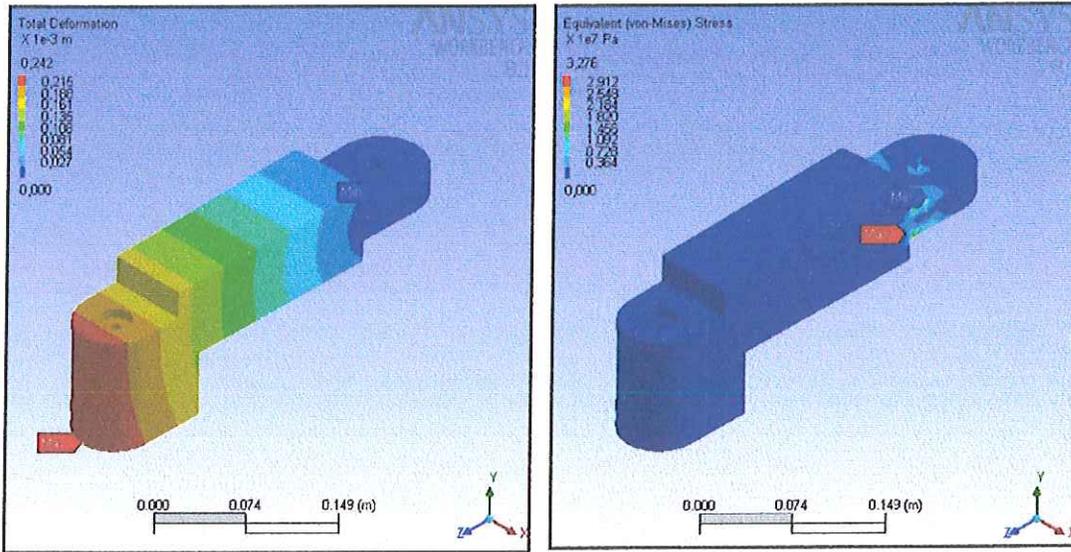


Del Autor

Geometría de la pieza

Name	Material	Bounding Box (m)	Mass (kg)	Volume (m <sup>3</sup> )	Nodes	Elements
"antebrazo"	"Aluminum Alloy"	$8.0 \times 10^{-2}$ , 0.13, 0.38	1.7	$6.14 \times 10^{-4}$	4617	2344

Figura 67. Deformación y esfuerzo equivalente del Antebrazo

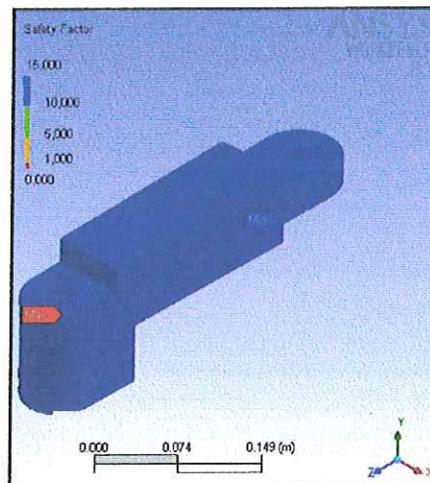


Del Autor

Resultados estructurales

Name	Scope	Minimum	Maximum	Alert Criteria
"Total Deformation"	"Model"	0.0 m	$2.42 \times 10^{-4}$ m	None
"Equivalent Stress"	"Model"	27.04 Pa	$3.28 \times 10^7$ Pa	None

Figura 68. Factor de seguridad del Antebrazo

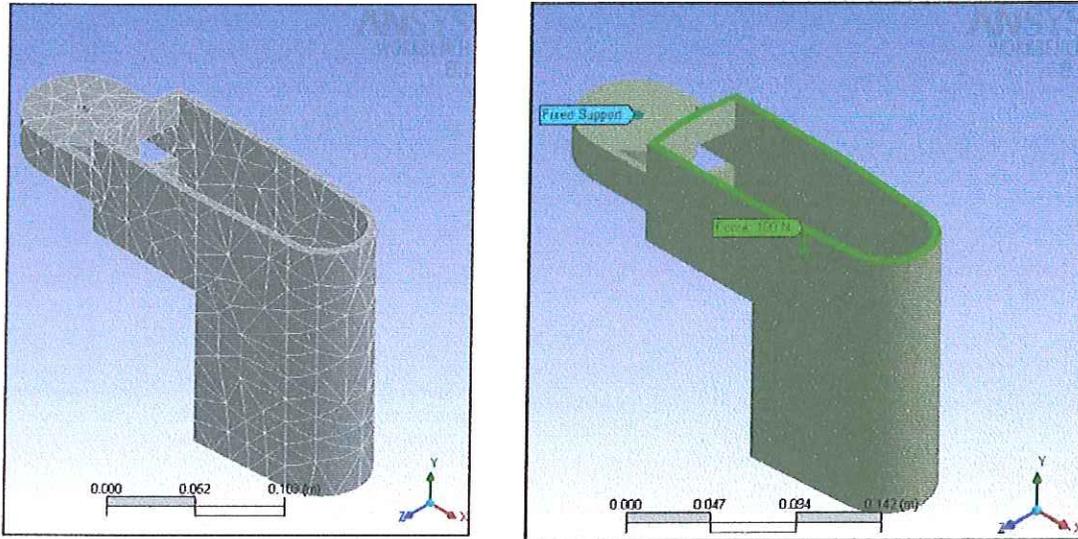


Del Autor

## Factor de seguridad

Name	Scope	Type	Minimum	Alert Criteria
"Stress Tool"	"Model"	Safety Factor	8.55	None

Figura 69. Enmallado y fuerza aplicada al Brazo

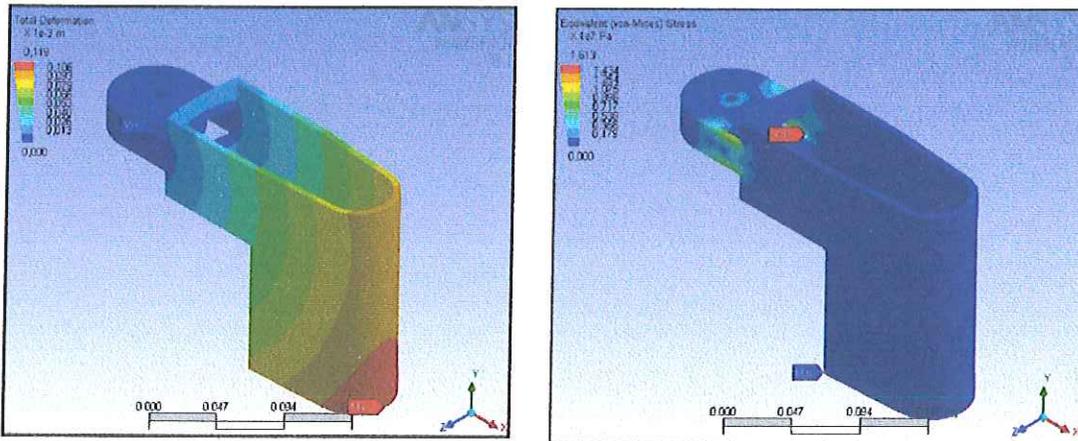


Del Autor

## Geometría de la pieza

Name	Material	Bounding Box (m)	Mass (kg)	Volume (m <sup>3</sup> )	Nodes	Elements
"brazo"	"Aluminum Alloy"	0.26, 0.17, 7.0×10 <sup>-2</sup>	1.35	4.86×10 <sup>-4</sup>	3672	1799

**Figura 70. Deformación y esfuerzo equivalente del Brazo**

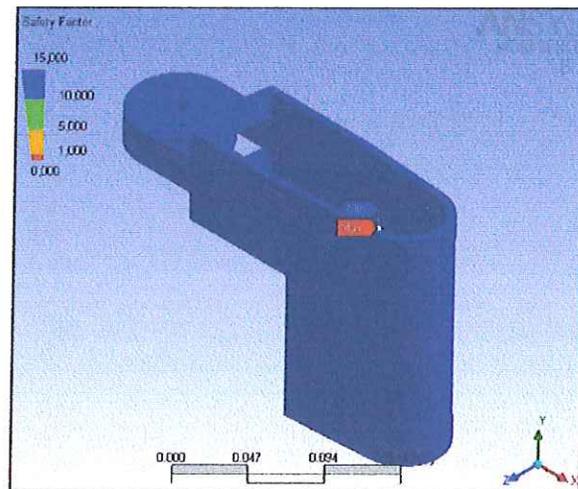


Del Autor

**Resultados estructurales**

Name	Scope	Minimum	Maximum	Alert Criteria
"Total Deformation"	"Model"	0.0 m	$1.19 \times 10^{-4}$ m	None
"Equivalent Stress"	"Model"	186.39 Pa	$1.61 \times 10^7$ Pa	None

**Figura 71. Factor de seguridad del brazo**

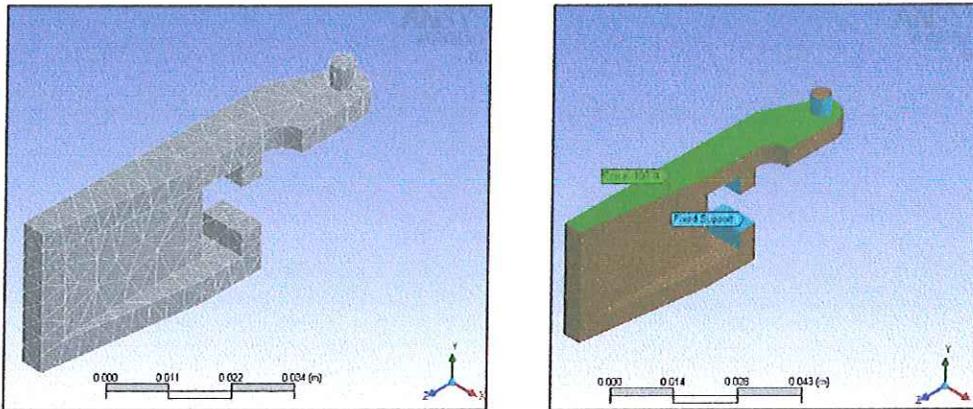


Del Autor

Factor de seguridad

Name	Scope	Type	Minimum	Alert Criteria
"Stress Tool"	"Model"	Safety Factor	15.0	None

Figura 72. Enmallado y fuerza aplicada a la pinza

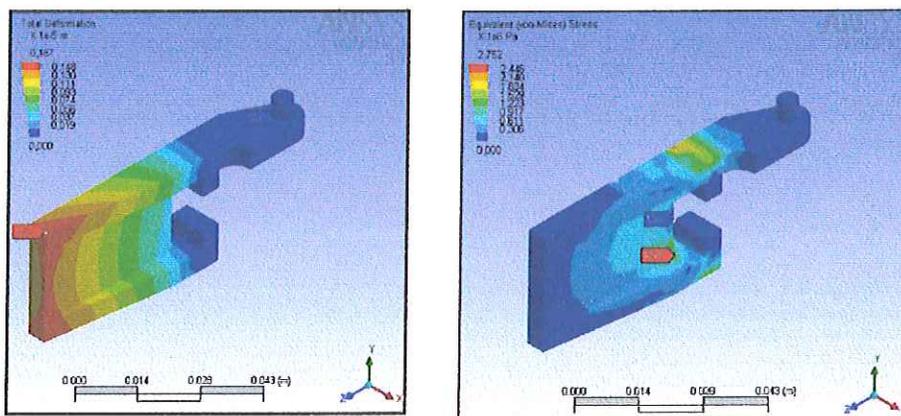


Del Autor

Geometría de la pieza

Name	Material	Bounding Box (m)	Mass (kg)	Volume (m <sup>3</sup> )	Nodes	Elements
"pinza1"	"Aluminum Alloy"	$1.5 \times 10^{-2}$ , $3.5 \times 10^{-2}$ , $7.49 \times 10^{-2}$	$2.83 \times 10^{-2}$	$1.02 \times 10^{-5}$	1672	783

Figura 73. Deformación y esfuerzo equivalente de la pinza Del Autor

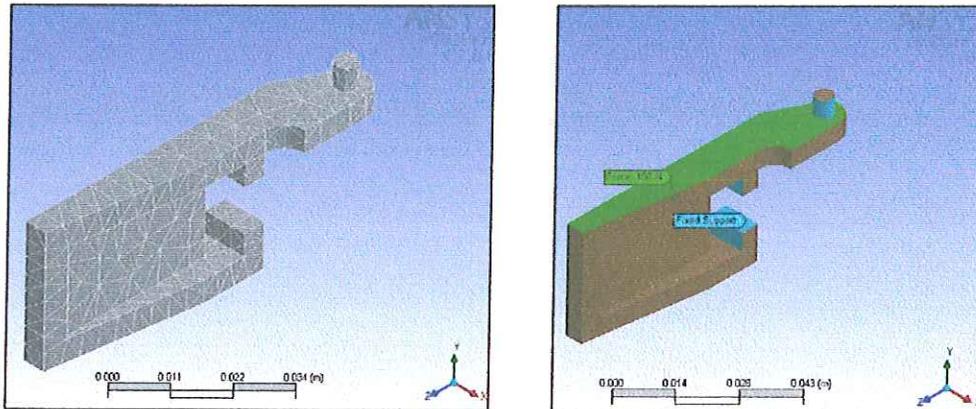


Del Autor

Factor de seguridad

Name	Scope	Type	Minimum	Alert Criteria
"Stress Tool"	"Model"	Safety Factor	15.0	None

Figura 72. Enmallado y fuerza aplicada a la pinza

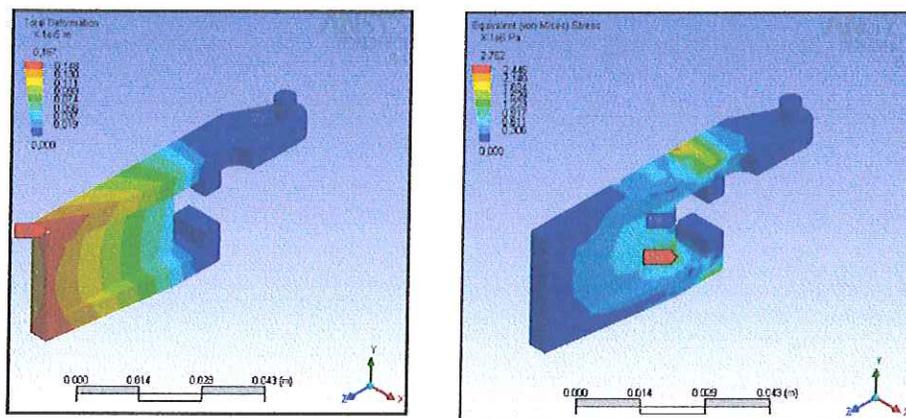


Del Autor

Geometría de la pieza

Name	Material	Bounding Box (m)	Mass (kg)	Volume (m <sup>3</sup> )	Nodes	Elements
"pinza1"	"Aluminum Alloy"	$1.5 \times 10^{-2}$ , $3.5 \times 10^{-2}$ , $7.49 \times 10^{-2}$	$2.83 \times 10^{-2}$	$1.02 \times 10^{-5}$	1672	783

Figura 73. Deformación y esfuerzo equivalente de la pinza Del Autor

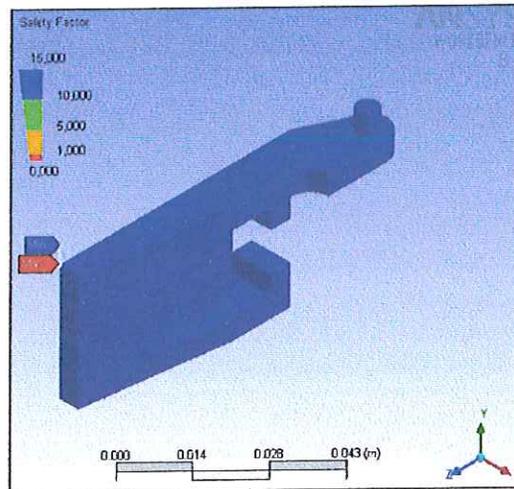


Del Autor

## Resultados estructurales

Name	Scope	Minimum	Maximum	Alert Criteria
"Total Deformation"	"Model"	0.0 m	$8.77 \times 10^{-7}$ m	None
"Equivalent Stress"	"Model"	181.45 Pa	114,849.72 Pa	None

Figura 74. Factor de seguridad de la pinza



Del Autor

Factor de seguridad

Name	Scope	Type	Minimum	Alert Criteria
"Stress Tool"	"Model"	Safety Factor	15.0	None

## A. Selección de Materiales

Para la selección de los materiales utilizados en la fabricación del manipulador, igualmente se realizaron simulaciones del diseño con diferentes tipos de materiales.

Después de varias pruebas se selecciono el aluminio para la fabricación de las piezas del manipulador, por que después del acero, es el metal mas ampliamente usado y sus aplicaciones son muy variadas, es ligero, posee una buena resistencia a la corrosión, ofrece facilidad para darle forma y maquinarlo, su densidad es aproximadamente la tercera parte que la del acero.

**Tabla 5. Usos de Aleaciones de Aluminio**

<i>Aleación</i>	<i>Aplicaciones</i>	<i>Formas</i>
1060	Equipo químico y tanques	Lámina, placa, tubo
1350	Conductores eléctricos	Lámina, placa, tubo, varilla, barra, alambre, tubería, formas
2014	Estructuras para aeronaves y bastidores para vehiculos	Lámina, placa, varilla, barra, alambre, formas, piezas forjadas (forjas)
2024	Estructuras para aeronaves, ruedas, partes de máquinas	Lámina, placa, tubo, varilla, barra, alambre, formas, remaches
2219	Partes sujetas a altas temperaturas (hasta 600°F)	Lámina, placa, tubo, varilla, barra, formas, forjas
3003	Equipo químico, tanques, utensilios de cocina, partes para arquitectura	Lámina, placa, tubo, varilla, barra, alambre, formas, tubería, remaches, forjas
5052	Tubería hidráulica, aparatos para el hogar (línea blanca), fabricación de láminas de metal	Lámina, placa, tubo, varilla, barra, alambre, remaches
6061	Estructuras, bastidores y partes para vehiculos, usos marinos	Todas las formas
6063	Muebles, accesorios para arquitectura	Tubos, tubería, formas extruidas
7001	Estructuras de alta resistencia	Tubo, formas extruidas
7075	Estructuras para aeronaves y para trabajo pesado	Todas las formas, excepto tubería

Robert L. Mott, P.E., "Diseño de Elementos de Maquinas 2ª Edición". Tabla correspondiente a las aleaciones de aluminio comunes y sus usos. Pagina 44.

**Tabla 6. Propiedades del aluminio**

Con aleación y templado	Resistencia a la tracción		Resistencia a punto cedente		Ductibilidad (elongación porcentual en 2 pulg)	Resistencia al corte		Resistencia por durabilidad	
	Ksi	MPa	Ksi	MPa		Ksi	MPa	Ksi	MPa
1060-O	10	69	4	28	43	7	48	3	21
1060-H14	14	97	11	76	12	9	62	5	34
1060-H18	19	131	18	124	6	11	121	6	41
1350-O	12	83	4	28	28	8	55	—	—
1350-H14	16	110	14	97	—	10	69	—	—
1350-H19	27	186	24	165	—	15	103	7	48
2014-O	27	186	14	97	18	18	124	13	90
2014-T4	62	427	42	290	20	38	262	20	138
2014-T6	70	483	60	414	13	42	290	18	124
2024-O	27	186	11	76	22	18	124	13	90
2024-T4	68	469	47	324	19	41	283	20	138
2024-T361	72	496	57	393	12	42	290	18	124
2219-O	25	172	11	76	18	—	—	—	—
2219-T6z	60	414	42	290	10	—	—	15	103
2219-T87	69	476	57	393	10	—	—	15	103
3003-O	16	110	6	41	40	11	121	7	48
3003-H14	22	152	21	145	16	14	97	9	62
3003-H18	29	200	27	186	10	16	110	10	69
5052-O	28	193	13	90	30	18	124	16	110
5052-H34	38	262	31	214	14	21	145	18	124
5052-H38	42	290	37	255	8	24	165	20	138
6061-O	18	124	8	55	30	12	83	9	62
6061-T4	35	241	21	145	25	24	165	14	97
6061-T6	45	310	40	276	17	30	207	14	97
6063-O	13	90	7	48	—	10	69	8	55
6063-T4	25	172	13	90	22	—	—	—	—
6063-T6	35	241	31	214	12	22	152	10	69
7001-O	37	255	22	152	14	—	—	—	—
7001-T6	98	676	91	627	9	—	—	22	152
7075-O	33	228	15	103	16	22	152	—	—
7075-T6	83	572	73	503	11	48	331	23	159

Robert L. Mott, P.E., "Diseño de Elementos de Maquinas 2ª Edición". Tabla correspondiente a las propiedades típicas del aluminio. Apéndice A-19.

El eje roscado que conforma el sistema de desplazamiento es fabricado en acero inoxidable, ya que es el material que más se utiliza en elementos de maquinaria debido a sus propiedades de alta resistencia, extrema rigidez, durabilidad y relativa facilidad para fabricarlo, además se caracteriza por el alto nivel de resistencia a la corrosión.

**Tabla 7. Usos de Aceros**

Número AISI	Aplicaciones
1015	Partes de metal laminado; partes maquinadas (pueden ser carburizadas)
1030	Partes en forma de barra para uso general, palancas o manijas, eslabones o uniones, cuñas de unión
1045	Flechas o ejes, engranes
1080	Piezas para equipo agrícola (rejas, discos, dientes de rastillos, dientes de podadoras de césped) que se someten a fricción; resortes.
1112	Piezas de tornillos para máquinas.
4140	Engranes, flechas o ejes, piezas forjadas
4340	Engranes o ejes, piezas que requieren de un buen endurecimiento directo.
4640	Engranes, flechas o ejes, levas.
5150	Flechas o ejes para trabajo pesado, resortes, engranes.
52100	Pistas de rodamiento, bolas y baleros (acero para cojinetes)
6150	Engranes, piezas forjadas, flechas o ejes, resortes
8650	Engranes, flechas o ejes
9260	Resortes

Robert L. Mott, P.E., "Diseño de Elementos de Maquinas 2ª Edición". Tabla correspondiente al uso de algunos aceros. Pagina 33.

**Tabla 8. Propiedades del Acero Inoxidable**

Designación del material		Condición	Resistencia a la tracción		Resistencia a punto cedente		Ductibilidad (elongación porcentual en 2 pulg)
Número AISI	UNS		Ksi	MPa	Ksi	MPa	
<b>Aceros austeníticos</b>							
201	S30100	Recocido	115	793	55	379	55
		1/4 duro	125	862	75	517	20
		1/2 duro	150	1030	110	758	10
		3/4 duro	175	1210	135	931	5
		Totalmente duro	185	1280	140	966	4
301	S30100	Recocido	110	758	40	276	60
		1/4 duro	125	862	75	517	25
		1/2 duro	150	1030	110	758	15
		3/4 duro	175	1210	135	931	12
		Totalmente duro	185	1280	140	966	8
304	S30400	Recocido	85	586	35	241	60
310	S31000	Recocido	95	655	45	310	45
316	S31600	Recocido	80	552	30	207	60
<b>Aceros ferríticos</b>							
405	S40500	Recocido	70	483	40	276	30
430	S43000	Recocido	75	517	40	276	30
446	S44600	Recocido	80	552	50	345	25
<b>Aceros martensíticos</b>							
410	S41000	Recocido	75	517	40	276	30
416	S41600	Q&T 600	130	1240	140	966	15
		Q&T 1000	145	1000	115	793	20
		Q&T 1400	90	621	60	414	30
431	S43100	Q&T 600	195	1344	150	1034	15
440A	S44002	Q&T 600	280	1950	270	1860	3
<b>Aceros endurecidos por precipitación</b>							
17-4PH	S17400	H 900	200	1380	185	1280	14
		H 1150	145	1000	125	862	19
17-7PH	S17700	RH 950	200	1350	175	1210	10
		TH 1650	175	1210	155	1070	12

Robert L. Mott, P.E., "Diseño de Elementos de Maquinas 2ª Edición". Tabla correspondiente a las propiedades de los aceros inoxidable. Apéndice A-15.

Otro tipo de material empleado en la construcción del manipulador fue el acrílico, ya que ofrece una buena resistencia mecánica, dureza, fácil pulido y maquinado, superficie brillante, alta claridad óptica, disponibilidad en colores.

**Tabla 9. Propiedades del Acrílico**

Propiedad	Norma ASTM	Unidad	Acrílico
Resistencia a la tracción	D 638	P.S.I	8000 - 11000
Elongación %	D 638	%	2 - 7
Resistencia a la compresión	D 695	P.S.I.	11000 - 19000
Resistencia a la flexión	D 790	P.S.I.	12000 - 17000
Dureza Rockwell	D 785	-	M80 - M100
Módulo de flexión	D 790	PSI X 10E5	3.90 - 4.75
Módulo de tracción	D 638	PSI X 10E5	3.5 - 4.5
Módulo de compresión	D 695	PSI X 10E5	3.90 - 4.75
Peso específico	D 638	g/cm <sup>3</sup>	1.17 - 1.20
Conductividad térmica	C 177	10E-1 cal/seg/cm <sup>2</sup>	4.6
Calor específico	-	Cal/°C/gm	0.35
Resistencia al calor continuo	-	°C	60 - 93
Temperatura de distorsión 264 P.S.I. - 66 P.S.I.	D 648	-	71 - 102 74 - 113
Resistencia específica	D 257	OHM/cm	>10E15
Resistencia dieléctrica de ruptura	D 149	nota	450 / 550
Constante dieléctrica 60 ciclos	D 150	-	3.3 - 4.5
Resistencia arco voltaico	D 495	seg	no surca
Absorción de agua (24 hs. 1/8" esp.)	D 570	%	0.2 - 0.4
Efecto de los rayos solares	-	-	Ninguna
Resistencia a los ácidos leves	D 543	-	Buena
Resistencia a los ácidos fuertes	D 543	-	Buena
Resistencia a las bases débiles	D 543	-	Buena
Resistencia a bases fuertes	D 543	-	Buena
Resistencia a solventes orgánicos	D 543	-	Buena
Pérdida por abrasión	-	Mg.	-

Del Autor.

## **B. Selección de actuadores**

Los actuadores son los encargados de generar el movimiento de los elementos del robot según las ordenes dadas por la unidad de control. En robótica los actuadores utilizados pueden emplear energía neumática, hidráulica o eléctrica. En la actualidad se ha extendido el uso de motores eléctricos, y en especial de motores de corriente continua servo controlados, empleándose en algunos casos motores paso a paso y otros actuadores electromecánicos sin escobillas.

Cada uno de estos sistemas presenta características diferentes, siendo preciso evaluarlas a la hora de seleccionar el tipo de actuador más conveniente. Las características a considerar son entre otras:

- Potencia
- Controlabilidad
- Peso y volumen
- Precisión
- Velocidad
- Mantenimiento

Las características de control, sencillez y precisión de los actuadores eléctricos hacen de estos los más usados en los robots industriales.

Se seleccionaron motores de corriente continua servocontrolados, más comúnmente llamados servomotores, para realizar el movimiento de las articulaciones del robot, ya que además de las características antes mencionadas ofrecen disminución de la inercia, suministran un par elevado, aumentan la precisión, originan menos ruido magnético y son de bajo peso y consumo. Los servomotores que cumplían con las características de los torques necesarios son los HITEC HSR-5995TG.

**Figura 75. Servomotor Hitec**

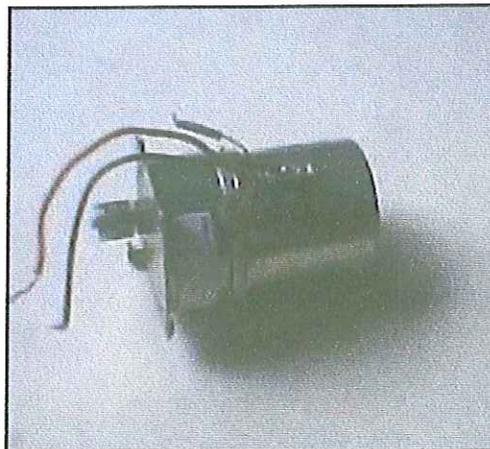


[http://www.servocity.com/html/hsr-5995tg\\_ultra\\_torque.html](http://www.servocity.com/html/hsr-5995tg_ultra_torque.html)

Para realizar el accionamiento del mecanismo que generaría el movimiento traslacional del elemento terminal se selecciono un motoreductor.

Este motor es de corriente continua; y como su nombre lo indica funciona con corriente y voltaje constantes. Su voltaje terminal es de 12V, y en consecuencia la corriente terminal es elevada; la velocidad de transmisión son 60 rpm y el peso que soporta son 2 Kg. aproximadamente.

**Figura 76. Motoreductor**



Del Autor

### 3. CONSTRUCCION Y MONTAJE

Para la construcción del manipulador se realizaron planos de las piezas diseñadas con el fin de elaborar modelos en madera de las mismas para posteriormente utilizarlos como moldes en la fundición del aluminio.

Los modelos debieron construirse en dos mitades, cada pieza fue moldeada con presión y exactitud, ya que de esto dependía la buena calidad de la fundición que garantizara excelente resistencia para que cada una de ellas pudiera ser maquinada después.

**Figura 77. Modelos en madera**



Del Autor

**Figura 78. Piezas en fundición de aluminio**



Del Autor

Una vez entregadas las piezas del manipulador fabricadas en fundición de aluminio, se procedió a realizar el mecanizado de las mismas en el laboratorio de manufactura de la facultad

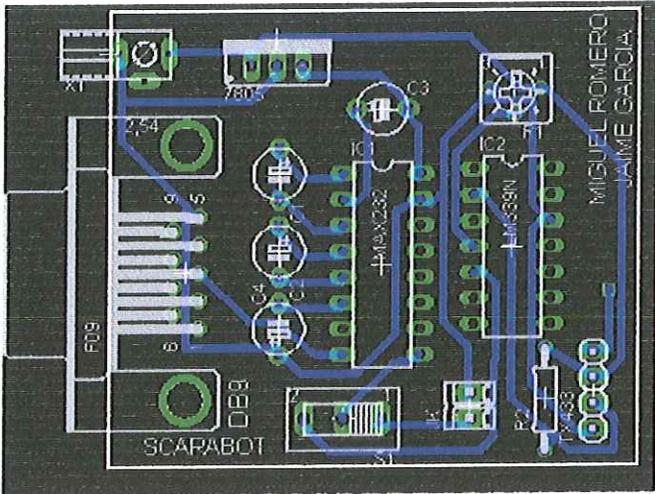
**Figura 79. Piezas mecanizadas**



Del Autor

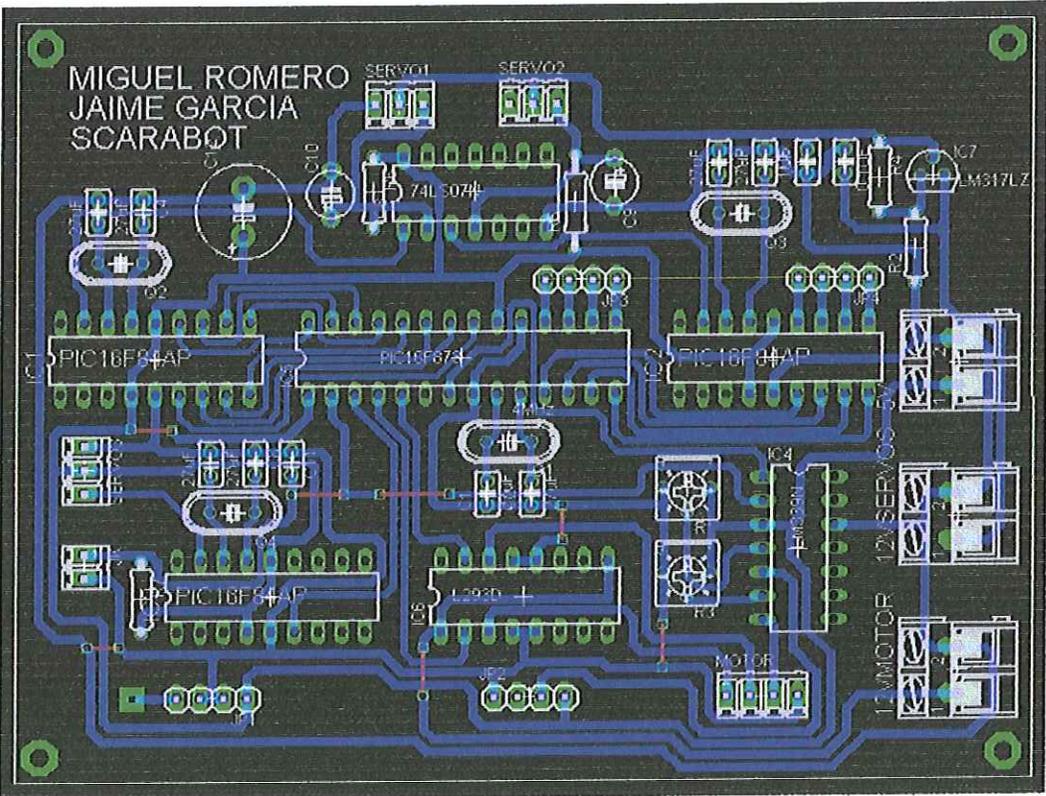
Circuitos Impresos realizados para integración al prototipo.

Figura 80. Circuito tarjeta transmisora



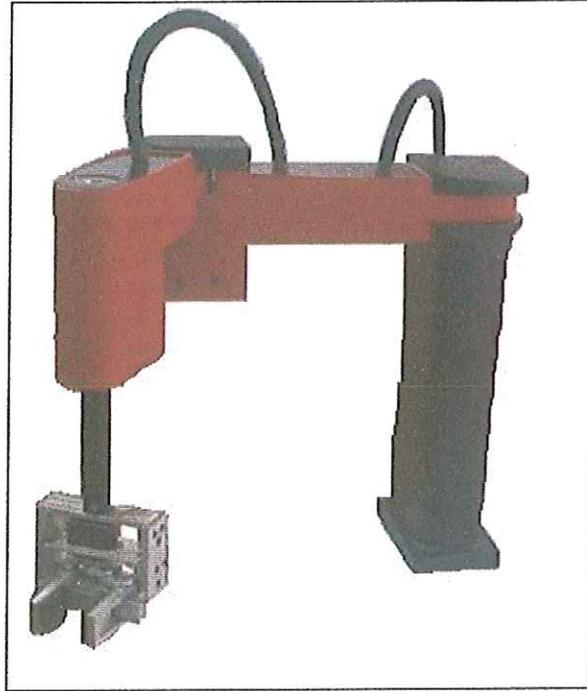
Del Autor

Figura 81. Circuito tarjeta receptora



Del Autor

**Figura 82. Manipulador tipo Scara construido**



Del Autor

#### 4. CONCLUSIONES Y RECOMENDACIONES

En el desarrollo de este proyecto se logro integrar y conjugar valiosos conocimientos obtenidos durante la carrera, además se obtuvo un aporte en las áreas de programación de microcontroladores y comunicación inalámbrica.

- Se revisaron los análisis Cinemático y Dinámico anteriormente hechos en la tesis **“DISEÑO Y CONSTRUCCIÓN DE UN MANIPULADOR TIPO SCARA CON IMPLEMENTACION DE UN SISTEMA DE CONTROL”**.
- Las pruebas realizadas al manipulador anteriormente construido, nos ayudaron a comprender el error que se estaba cometiendo en la programación de los microcontroladores.
- Se realizo un mejoramiento a los algoritmos de control de Inercia y de trayectorias para el robot Scara construido anteriormente para la facultad.
- Se diseño e implemento un programa en Matlab 7.0 que permite simular los diferentes tipos de movimientos del Manipulador tipo Scara.
- La simulación computacional realizada (Análisis Cinemático y Dinámico), nos permitieron encontrar valores importantes para el diseño del manipulador. tales como torques, velocidades de los servomotores y área de trabajo.

- El diseño realizado en SolidWorks 2004 y los análisis en Ansys Workbench y CosmoExpress, nos permitieron encontrar el mejor prototipo de diseño del manipulador.
- Los servomotores escogidos para la construcción del manipulador, fueron acertados y precisos, ya que nos permitieron realizar un control de posición e inercia aceptables.
- La investigación realizada para el control de los servomotores con los microcontroladores, nos permitieron demostrar, que la mejor programación era en lenguaje Basic.
- Se logro obtener una resolución de 1 grado, en los servomotores por medio de la programación en lenguaje Basic de los microcontroladores.
- Las pruebas realizadas a los servomotores nos ayudaron a encontrar que al variar la frecuencia de la señal de control (Pwm) en el servomotor, su velocidad tiene una variación mínima, la cual nos permitió realizar el control.
- Las pruebas realizadas en la simulación computacional nos permitieron lograr un algoritmo preciso para la cinemática inversa y el control de trayectorias.
- La metodología implementada fue un éxito ya que nos ayudo en el desarrollo del proyecto y nos permitió obtener buenos resultados en todas las etapas, ya que por medio de esta se ahorro tiempo y dinero.

## 5. BIBLIOGRAFIA

- [1]. CARRASCAL CORREDOR, Sindy Melisa y GIL AREIZA Leonardo. Diseño y construcción de un manipulador tipo SCARA con implementación de un sistema de control. Bucaramanga, 2004, 155 p. Tesis (Ingeniero Mecatronico). Universidad Autónoma de Bucaramanga. Facultad de Ingeniería Mecatrónica. Área de Robótica.
  
- [2]. BARRIENTOS, Antonio. Fundamentos de Robótica. Editorial McGraw Hill. 1997.
  
- [3]. OLLERO BATURONE, Aníbal. Robótica, Manipuladores y Robots Móviles. Editorial Alfaomega Marcombo. 2001.
  
- [4]. ANGULO USATEGUI, José. Microcontroladores PIC. Diseño práctico de aplicaciones 2<sup>da</sup> Parte. Editorial McGraw Hill.
  
- [5]. LUNG-Wentsai. Robots Analysis. The Mechanics of serial and parallel manipulators. Editorial Wiley-Interscience.
  
- [6]. DEUTSCHMAN, Aaron D. Diseño de maquinas. Teoría y Práctica. Editorial CECSA. 1995.
  
- [7]. IOVINE, Jhon. PIC Robotics. Editorial McGraw Hill. 2004

## Web Sites:

- Instituto Colombiano de Normas Técnicas "Icontec".  
<http://www.icontec.org.com>
- [www.monografias.com/trabajos16/estacion-robotica/estacion-robotica.shtml](http://www.monografias.com/trabajos16/estacion-robotica/estacion-robotica.shtml)
- [www.microchip.com](http://www.microchip.com)
- [www.x-robotics.com](http://www.x-robotics.com)
- [www.learnaboutrobots.com/images/scara.gif](http://www.learnaboutrobots.com/images/scara.gif)
- <http://www.mecc.polimi.it/rovetta/robotica/HTML/studentivaldata/filmati>
- <http://prime.jsc.nasa.gov/ROV/images>
- <http://prt.fernuni-hagen.de/lehre/KURSE/PRT001/EXAMPLES/eqs6/node1>
- <http://www.gimatic.com>
- [http://gemwelding.com/img\\_genr/pinzas/G01n.jpg](http://gemwelding.com/img_genr/pinzas/G01n.jpg)
- [http://www.esab.es/ESABImages/PR2003\\_Robot\\_laser.jpg](http://www.esab.es/ESABImages/PR2003_Robot_laser.jpg)

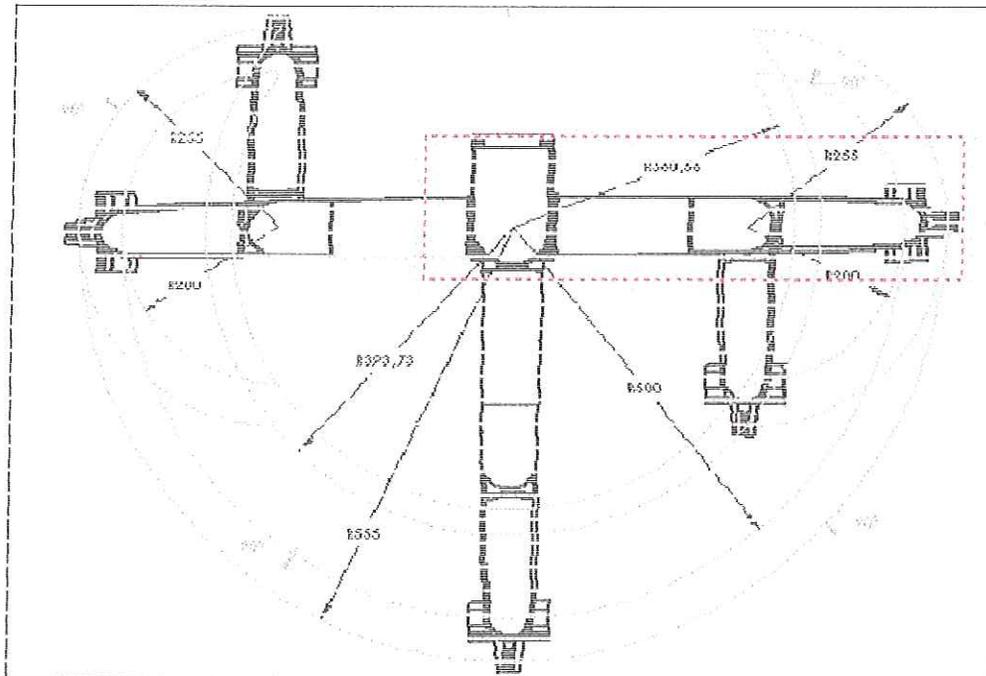
# **ANEXOS**

## ANEXO A

### DATOS TECNICOS DEL MANIPULADOR SCARA

ESPECIFICACIONES DEL MANIPULADOR SCARA	
NUMERO DE ARTICULACIONES	4: 2 rotacionales ( antebrazo y brazo), 1 traslacional (actuador) y Pinza
ELEMENTOS TERMINALES	2: 1 pinza mecánica y 1 porta-herramienta
ROTACION ANTEBRAZO	-90°, 0° y 0°, 90°
ROTACION BRAZO	-90°, 0° y 0°, 90°
TRASLACION ACTUADOR	12 cm.
ACTUADORES	3 servomotores y 1 motoreductor
CAPACIDAD SERVOMOTORES	30 Kg.*cm.
TRANSMISION	Transmisión directa mediante acoples
CARGA	5 Kg.

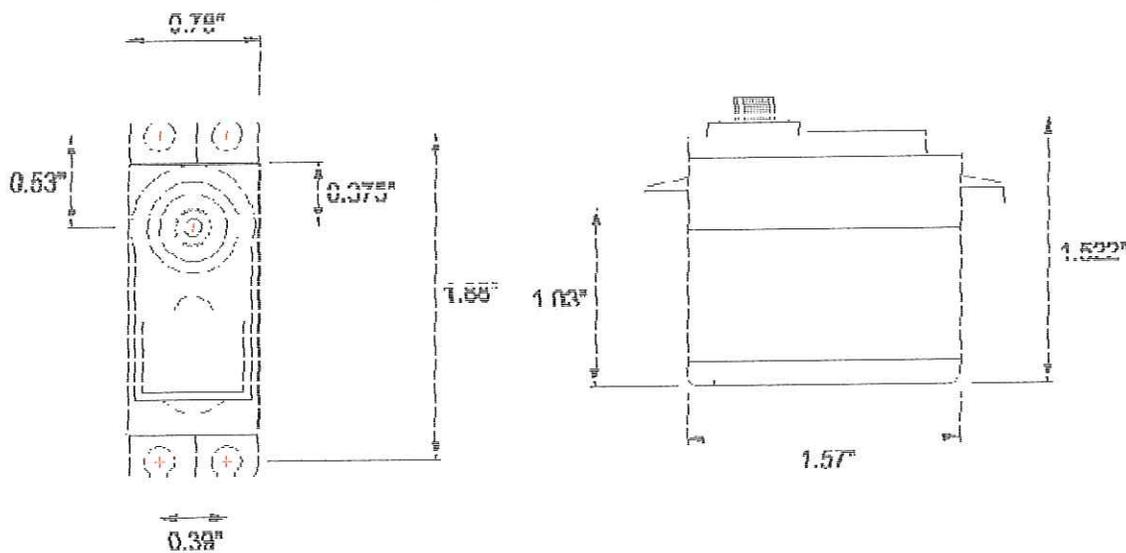
### AREA DE TRABAJO DEL MANIPULADOR



Del Autor

## ESPECIFICACIONES DEL SERVOMOTOR HITEC HS-5995TG

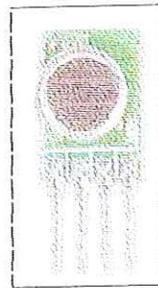
**Control System:** +Pulse Width Control 1500usec Neutral  
**Required Pulse:** 3 3-7 4 Volt Peak to Peak Square Wave  
**Operating Voltage Range:** 4.8-7.4 Volts  
**Operating Temperature Range:** -20 to +60 Degree C (-68F to +140F)  
**Operating Speed (6.0V):** 0.15sec/60 degrees at no load  
**Operating Speed (7.4V):** 0.12sec/60 degrees at no load  
**Stall Torque (6.0V):** 333.29oz/in. (24kg.cm)  
**Stall Torque (7.4V):** 416.61oz/in. (30kg.cm)  
**Standing Torque (6.0V):** 433.27oz/in. (31.2kg.cm) 5 degree deflection  
**Standing Torque (7.4V):** 541.59oz/in. (39kg.cm) 5 degree deflection  
**Operating Angle:** 90 Deg. one side pulse traveling 400usec  
**360 Modifiable:** Yes  
**Direction:** Clockwise/Pulse Traveling 1500 to 1900usec  
**Idle Current Drain (6.0V):** 3mA at stop  
**Idle Current Drain (7.4V):** 3mA at stop  
**Current Drain (6.0V):** 300mA/idle and 4.2 amps at lock/stall  
**Current Drain (7.4V):** 300mA/idle and 5.2 amps at lock/stall  
**Dead Band Width:** 2usec  
**Motor Type:** Coreless Metal Brush  
**Potentiometer Drive:** 6 Slider Indirect Drive  
**Bearing Type:** Dual Ball Bearing MR106  
**Gear Type:** 4 Titanium Gears  
**Connector Wire Length:** 11.81" (300mm)  
**Dimensions:** 1.57" x 0.78" x 1.45" (40 x 20 x 37mm)  
**Weight:** 2.18oz (62g)



## MODULOS TRANSMISOR Y RECEPTOR DE DATOS

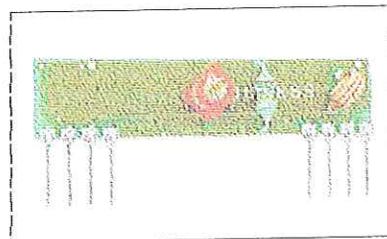
Se trata de una pareja de dispositivos de la marca wemshing que permiten emitir y recibir señales de radiofrecuencia a una frecuencia de 433 MHz, fija en el transmisor y ajustable en el receptor. El TX-433 y el RWS-434 Trabajan en AM, el transmisor TX-433 se muestra en la figura. Allí se aprecia que posee cuatro pines el pin 1 es GND, el pin 2 es data, por allí llega la señal a ser transmitida, el pin 3 es VCC y el pin 4 es ANT, en el pin cuatro se coloca una antena de aproximadamente 15cm.

TRANSMISOR TX-433



El receptor RWS-434 se muestra en la figura. Allí se aprecia que posee ocho pines y que su tamaño es reducido, no mayor al tamaño de un pila AA. La asignación de los pines es como sigue:

RECEPTOR RWS-434



1. Gnd
2. Out digital
3. Out
4. Vcc

5. Vcc
6. Gnd
7. Gnd
8. Ant

La función en conjunto de esta pareja de dispositivos es, en este trabajo, la de reemplazar el cable que une al computador, que emite la señal a transmitir, con el microcontrolador que la recibe. Así un cero lógico aplicado al pin  $\overline{DATA}$  del transmisor será interpretado por el receptor y este tomara un valor de voltaje  $x$  en sus pines de salida, y un uno lógico aplicado en el pin DATA del transmisor será también interpretado por el receptor que fijara un valor de voltaje  $x/2$  en sus terminales de salida, estos dos niveles de voltaje son llevadas mediante un Op – Amp a modo de comparador, LM 339, a niveles lógicos TTL antes de entregarse al microcontrolador y de esta forma cerrar el lazo de comunicación.



**MICROCHIP**

# PIC16F84A

## 18-pin *Enhanced* FLASH/EEPROM 8-Bit Microcontroller

### High Performance RISC CPU Features:

- Only 35 single word instructions to learn
- All instructions single-cycle except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input  
DC - 200 ns instruction cycle
- 1024 words of program memory
- 68 bytes of Data RAM
- 64 bytes of Data EEPROM
- 14-bit wide instruction words
- 8-bit wide data bytes
- 15 Special Function Hardware registers
- Eight-level deep hardware stack
- Direct, indirect and relative addressing modes
- Four interrupt sources:
  - External RB0/INT pin
  - TMR0 timer overflow
  - PORTB<7:4> interrupt-on-change
  - Data EEPROM write complete

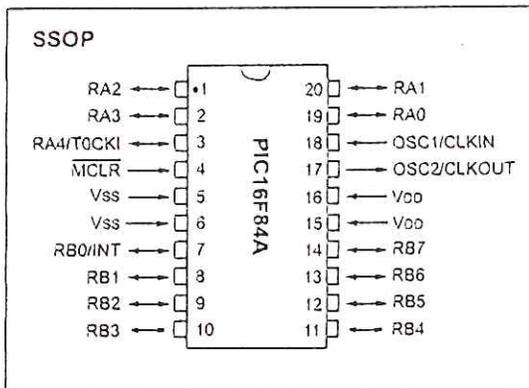
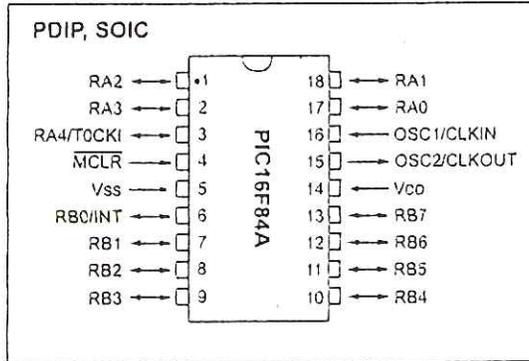
### Peripheral Features:

- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
  - 25 mA sink max. per pin
  - 25 mA source max. per pin
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler

### Special Microcontroller Features:

- 10,000 erase/write cycles *Enhanced* FLASH Program memory typical
- 10,000,000 typical erase/write cycles EEPROM Data memory typical
- EEPROM Data Retention > 40 years
- In-Circuit Serial Programming™ (ICSP™) - via two pins
- Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own On-Chip RC Oscillator for reliable operation
- Code protection
- Power saving SLEEP mode
- Selectable oscillator options

### Pin Diagrams



### CMOS Enhanced FLASH/EEPROM Technology:

- Low power, high speed technology
- Fully static design
- Wide operating voltage range:
  - Commercial: 2.0V to 5.5V
  - Industrial: 2.0V to 5.5V
- Low power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 15 µA typical @ 2V, 32 kHz
  - < 0.5 µA typical standby current @ 2V



MICROCHIP

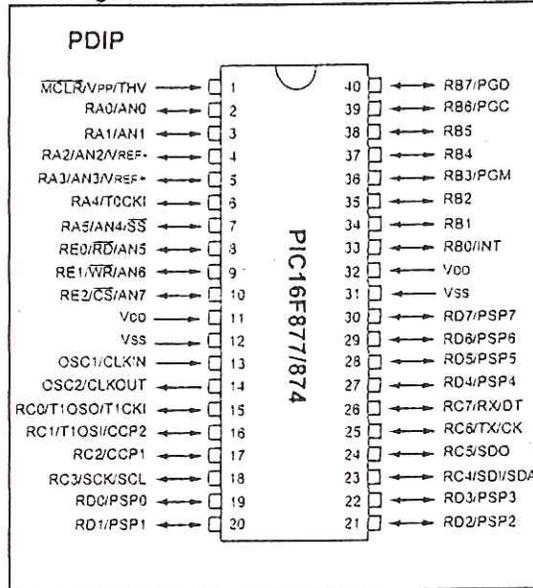
# PIC16F87X

## 28/40-pin 8-Bit CMOS FLASH Microcontrollers

### Microcontroller Core Features:

- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input  
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,  
Up to 368 x 8 bytes of Data Memory (RAM)  
Up to 256 x 8 bytes of EEPROM data memory
- ★ Pinout compatible to the PIC16C73/74/76/77
- Interrupt capability (up to 14 internal/external interrupt sources)
- Eight level deep hardware stack
- Direct, indirect, and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low-power, high-speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ via two pins
- ★ Only single 5V source needed for programming
- ★ In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low-power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

### Pin Diagram



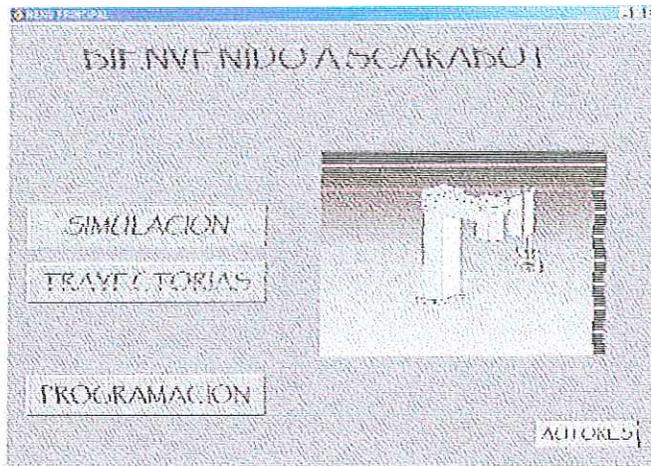
### Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
- Capture is 16-bit, max. resolution is 12.5 ns, Compare is 16-bit, max. resolution is 200 ns, PWM max. resolution is 10-bit
- ★ 10-bit multi-channel Analog-to-Digital converter
- ★ Synchronous Serial Port (SSP) with SPI™ (Master Mode) and I<sup>2</sup>C™ (Master/Slave)
- ★ Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external  $\overline{RD}$ ,  $\overline{WR}$  and  $\overline{CS}$  controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

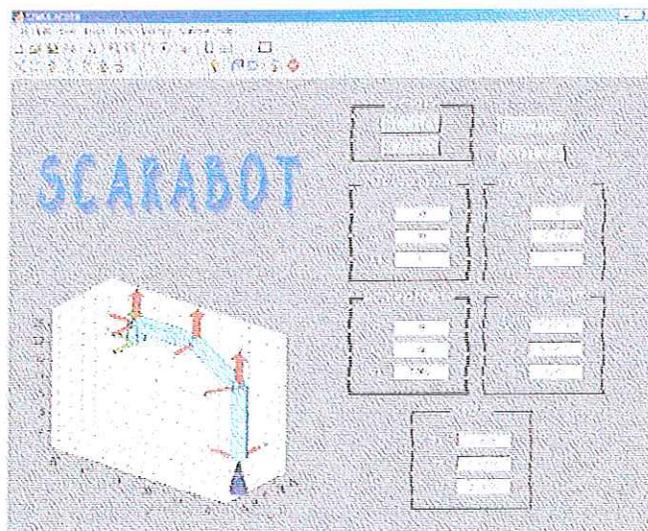
## ANEXO B

### PROGRAMAS DE MATLAB

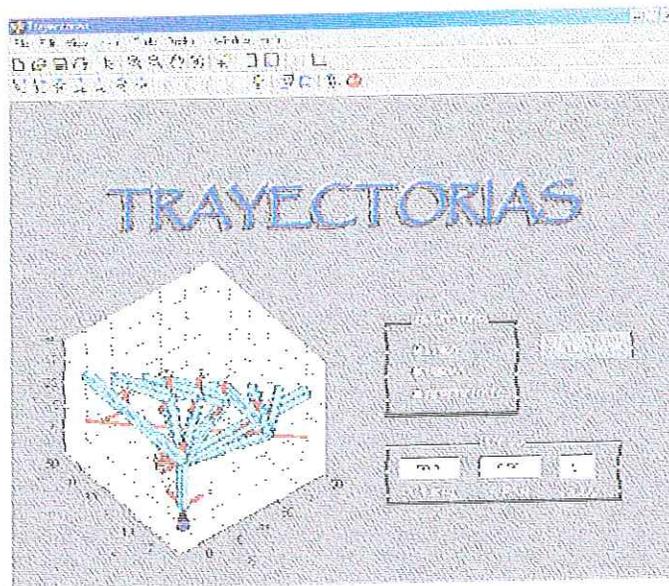
Pantalla de Inicio del Software de Simulación SCARABOT



Pantalla Opción Simulación



## Pantalla Opción Trayectorias



## Pantalla de los Autores



## Programas de Matlab 7.0

Programa del Menú

```
function varargout = Menu(varargin)
% MENU M-file for Menu.fig
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @Menu_OpeningFcn, ...
                  'gui_OutputFcn', @Menu_OutputFcn, ...
                  'gui_LayoutFcn', [], ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function Menu_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = Menu_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function Trayec_Callback(hObject, eventdata, handles)
TRAYEC
close(Menu)
function Write_Callback(hObject, eventdata, handles)
Write
close(Menu)
function Program_Callback(hObject, eventdata, handles)
Credit
close(Menu)
function Simu_Callback(hObject, eventdata, handles)
Simulac
close(Menu)
function autor_Callback(hObject, eventdata, handles)
Credit
```

---

Programa de la Simulacion del Manipulador.

```
function varargout = Simulac(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @Simulac_OpeningFcn, ...
    'gui_OutputFcn', @Simulac_OutputFcn, ...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function Simulac_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
set(handles.Cd, 'BorderWidth', 2);
set(handles.Jd, 'BorderWidth', 2);
set(handles.Ad, 'Visible', 'on', 'Position', [134.4 42.077 18.4 1.846]);
set(handles.Ai, 'Visible', 'off', 'Position', [134.4 42.077 18.4 1.846]);
set(handles.HOME, 'Visible', 'on');
q1=0; q2=0; l1=20; a2=30; a3=20; d1=20;
axes(handles.ROT)
dh=[0 0 0 0 0; 0 0 q1 l1 0; 0 a2 q2 0 0; 0 a3 0 0 0; 0 0 0 -d1 1];
plot3(dh,[0 0 0 0 0], 'kwr')
hold on
plot3(50,50,50);
hold off
function varargout = Simulac_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function Ad_Callback(hObject, eventdata, handles)
%Analisis Directo Cinematica
pifia=0;
n i = str2double(get(handles.Q i, 'String'));
if (h1>90) | (h1<-90)
    uiwait(msgbox('La Articulacion q1 dede estar entre -90 y 90
Grados', 'Error', 'error'))
    pifia=1;
end
```

```

te1=(pi*h1)/180;
h2= str2double(get(handles.Q2,'String'));
if (h2>90) | (h2<=-90)
    uiwait(msgbox('La Articulacion q2 dede estar entre -90 y 90
Grados','Error','error'))
    pifia=i;
end
te2=(pi*h2)/180;
h3= str2double(get(handles.D1,'String'));
if (h3>20)|(h3<0)
    uiwait(msgbox('La Articulacion d1 dede estar entre 20 y 0
Centimetros','Error','error'))
    pifia=1;
end
% Velocidades para los Jacobiano
vx= str2double(get(handles.W1,'String'));
vy= str2double(get(handles.W2,'String'));
vz= str2double(get(handles.V1,'String'));
%Solucion Cinematica
l1=20; a2=30; a3=20;
dh=[0 0 0 0 0;0 0 te1 l1 0; 0 a2 te2 0 0; 0 a3 0 0 0; 0 0 0 -h3 1];
q=[0 te1 te2 0 -h3]; Ts= (fkine(dh,q));
tX=Ts(1,4); tY=Ts(2,4); tZ=Ts(3,4);
if pifia==0
    dh=[0 0 0 0 0;0 0 te1 l1 0; 0 a2 te2 0 0; 0 a3 0 0 0; 0 0 0 -h3 1];
    plotbot(dh,[0 0 0 0 0],'fwr'),pause(1)
    plotbot(dh,[0 te1 te2 0 -h3],'fwi')
end
%Calculo los jacobianos
[Jaco, Mj]=jacob(te1,te2,vx,vy,vz);
%Calculo Torques
[t1,t2,t3]=Torques(te2,vx,vy,vz);
% Imprimo Resultados
set(handles.X,'ForegroundColor',[0.502 0.502 0.502],'String',tX);
set(handles.Y,'ForegroundColor',[0.502 0.502 0.502],'String',tY);
set(handles.Z,'ForegroundColor',[0.502 0.502 0.502],'String',tZ);
set(handles.VX,'ForegroundColor',[0.502 0.502 0.502],'String',Jaco(1));
set(handles.VY,'ForegroundColor',[0.502 0.502 0.502],'String',Jaco(2));
set(handles.VZ,'ForegroundColor',[0.502 0.502 0.502],'String',Jaco(3));
set(handles.T1,'ForegroundColor',[0.502 0.502 0.502],'String',t1);
set(handles.T2,'ForegroundColor',[0.502 0.502 0.502],'String',t2);
set(handles.T3,'ForegroundColor',[0.502 0.502 0.502],'String',t3);

```

```

function Ai_Callback(hObject, eventdata, handles)
%Analisis Inverso, Cinematica

```

```

ero=0;
p1= str2double(get(handles.X,'String'));
p2= str2double(get(handles.Y,'String'));
p3= str2double(get(handles.Z,'String'));
w1= str2double(get(handles.VX,'String'));
w2= str2double(get(handles.VY,'String'));
v1= str2double(get(handles.VZ,'String'));
if (p3>15)
    uiwait(msgbox('La Articulacion d1 dede estar entre 0 Y 15
Centimetros','Error','error'))
    ero=1;
end
l1=30; l2=20; l3=20; qnew=inversa(p1,p2);
g=qnew*pi/180; zn=20-p3; g(3)=zn;
%Jacobiano Inverso
[Jaco, Mj]=jacob(qnew(1),qnew(2),w1,w2,v1);
Jinver = inv (Mj);
Jinv = Jinver*[w1;w2;v1];
%Calculo los Torques
[t1,t2,t3]=Torques(qnew(2),Jinv(1),Jinv(2),Jinv(3));
%Imprimo Variables
set(handles.Q1,'String',qnew(1)); set(handles.Q2,'String',qnew(2));
set(handles.D1,'String',zn);
set(handles.W1,'String',Jinv(1)); set(handles.W2,'String',Jinv(2));
set(handles.V1,'String',Jinv(3));
set(handles.T1,'String',t1); set(handles.T2,'String',t2); set(handles.T3,'String',t3);
axes(handles.ROT)
if ero==0
dhne=[0 0 0 0;0 0 g(1) l3 0;0 l1 g(2) 0 0;0 l2 0 0 0;0 0 0 -g(3) 1];
plotbot(dhne,[0 0 0 0],'fwr')
pause(1)
plotbot(dhne,[0 g(1) g(2) 0 -g(3)],'fwr')
end
function HOME_Callback(hObject, eventdata, handles)
X1_1=0; Y1_1=0; Z1_1=0; X=50; Y=0; Z=0;
q1=0; q2=0; l1=20; a2=30; a3=20; d1=20;
dh=[0 0 0 0;0 0 q1 l1 0; 0 a2 q2 0 0; 0 a3 0 0 0; 0 0 0 -d1 1];
plotbot(dh,[0 0 0 0],'fwr')
set(handles.Q1,'ForegroundColor',[0 0 0],'String',X1_1);
set(handles.Q2,'ForegroundColor',[0 0 0],'String',Y1_1);
set(handles.D1,'ForegroundColor',[0 0 0],'String',Z1_1);
set(handles.X,'ForegroundColor',[0.502 0.502 0.502],'String',X);
set(handles.Y,'ForegroundColor',[0.502 0.502 0.502],'String',Y);
set(handles.Z,'ForegroundColor',[0.502 0.502 0.502],'String',Z);
function C_Callback(hObject, eventdata, handles)

```

```

set(handles.Ci,'BorderWidth',1);
set(handles.Ji,'BorderWidth',1);
set(handles.Cd,'BorderWidth',2);
set(handles.Jd,'BorderWidth',2);
set(handles.Ad,'Visible','on','Position',[134.4 42.077 18.4 1.846]);
set(handles.Ai,'Visible','off');
set(handles.HOME,'Visible','on');
set(handles.Q1,'ForegroundColor',[0 0 0]);
set(handles.Q2,'ForegroundColor',[0 0 0]);
set(handles.D1,'ForegroundColor',[0 0 0]);
set(handles.X,'ForegroundColor',[0.502 0.502 0.502]);
set(handles.Y,'ForegroundColor',[0.502 0.502 0.502]);
set(handles.Z,'ForegroundColor',[0.502 0.502 0.502]);
set(handles.W1,'ForegroundColor',[0 0 0]);
set(handles.W2,'ForegroundColor',[0 0 0]);
set(handles.V1,'ForegroundColor',[0 0 0]);
set(handles.VX,'ForegroundColor',[0.502 0.502 0.502]);
set(handles.VY,'ForegroundColor',[0.502 0.502 0.502]);
set(handles.VZ,'ForegroundColor',[0.502 0.502 0.502]);
set(handles.Cd,'BorderWidth',1);
set(handles.Jd,'BorderWidth',1);
set(handles.Ci,'BorderWidth',2);
set(handles.Ji,'BorderWidth',2);
set(handles.X,'ForegroundColor',[0 0 0]);
set(handles.Y,'ForegroundColor',[0 0 0]);
set(handles.Z,'ForegroundColor',[0 0 0]);
set(handles.Q1,'ForegroundColor',[0.502 0.502 0.502]);
set(handles.Q2,'ForegroundColor',[0.502 0.502 0.502]);
set(handles.D1,'ForegroundColor',[0.502 0.502 0.502]);
set(handles.VX,'ForegroundColor',[0 0 0]);
set(handles.VY,'ForegroundColor',[0 0 0]);
set(handles.VZ,'ForegroundColor',[0 0 0]);
set(handles.W1,'ForegroundColor',[0.502 0.502 0.502]);
set(handles.W2,'ForegroundColor',[0.502 0.502 0.502]);
set(handles.V1,'ForegroundColor',[0.502 0.502 0.502]);
set(handles.Ad,'Visible','off');
set(handles.Ai,'Visible','on','Position',[134.4 42.077 18.4 1.846]);
set(handles.HOME,'Visible','on');

```

### Programas de Trayectorias

```
function varargout = TRAYEC(varargin)
```

```
gui_Singleton = 1;
```

```
gui_State = struct('gui_Name', mfilename,'gui_Singleton', gui_Singleton,...
```

```

        'gui_OpeningFcn', @TRAYEC_OpeningFcn,'gui_OutputFcn',
        @TRAYEC_OutputFcn, ...
        'gui_LayoutFcn', [] , 'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function TRAYEC_OpeningFcn(hObject, eventdata, handles, varargin)
set(handles.points,'Visible','on');
pinicial=[0 0]; pfinal=[0 0]; pasos=0; Repet=0;
handles.output = hObject;
l1=30; l2=20; l3=20;
axes(handles.D3)
dh_1= [0 0 0 0 0;0 0 0 l3 0;0 l1 0 0 0;0 l2 0 0 1];
q_1=[0 0 0 0];
plotbot(dh_1,q_1,'fwr')
hold on, plot3(0,40,40);
guidata(hObject, handles);
function varargout = TRAYEC_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function A_Callback(hObject, eventdata, handles)
t1=0; t2=0; t3=0;
% Estos valores de t1, t2 y t3 son irrelevantes
l1=30; l2=20; l3=20;
% Se crea la matriz dh con los parámetros del manipulador
dh = [0 0 0 0 0;0 0 t1 l3 0;0 l1 t2 0 0;0 l2 t3 0 1];
plotbot(dh,[0 0 0 0],'fwr')
cla
pinicial=str2num(get(handles.ini,'String'));
pfinal=str2num(get(handles.fin,'String'));
pasos=str2double(get(handles.pas,'String'));
Repetir=str2double(get(handles.R,'String'));
xi=pinicial(1); yi=pinicial(2); xf=pfinal(1); yf=pfinal(2); pnew=pasos+1;
if Repetir>1
    pasos=1;
end
if (xi>xf)
    tempx=xi-xf; d=(xi-xf)/pasos; x=[xi:-d:xf];
else
    tempx=xf-xi; c=(xf-xi)/pasos; x=[xi:c:xf];

```

```

end
if (yi>yf)
    tempy=yi-yf; d1=(yi-yf)/pasos; y=[yi:-d1:yf];
else
    tempy=yf-yi; c1=(yf-yi)/pasos; y=[yi:c1:yf];
end
if(xi==xf)
    x=xi*(ones(1,pnew));
end
if(yi==yf)
    y=yi*(ones(1,pnew));
end
ph=zeros(1,length(x));
for k=1:length(x)
    z(k)=0;
    TG1(:,k)=[cos(ph(k)) -sin(ph(k)) 0 x(k); sin(ph(k)) cos(ph(k)) 0 y(k); 0 0 1 z(k); 0
0 0 1];
end
for g=1:length(x);
    h=1:4;
    q(g,h)=0;
end
t=trayectorias(x,y)
for k=1:length(x)
    for j=2:3
        q(k,j)=t(k,j-1)*pi/180;
    end
end
axes(handles.D3)
plot3(40,40,40)
if Repetir==1
    plotbot(dh,q,'fdr')
    hold on
    plot3(x,y,z,'r','LineWidth',2.5); grid on
end
axes(handles.D3)
grid on
if Repetir>1
% for g=1:length(x);
    for j=1:Repetir
        plotbot(dh,q,'fwr')
        hold on,plot3(40,40,40)
        pause(1.2),grid on
    end
end
end

```

```

function L_Callback(hObject, eventdata, handles)
axes(handles.D3)
cla
set(handles.points, 'Visible', 'on');
set(handles.pas, 'Visible', 'on');
set(handles.tp, 'Visible', 'on');
set(handles.R, 'Visible', 'off');
set(handles.tr, 'Visible', 'off');
set(handles.R, 'String', '1');
function C_Callback(hObject, eventdata, handles)
axes(handles.D3)
cla
t1=0; t2=0; t3=0; l1=30; l2=20; l3=20;
dh = [0 0 0 0 0; 0 0 t1 l3 0; 0 l1 t2 0 0; 0 l2 t3 0 1];
plotbot(dh,[0 0 0 0], 'fwr')
hold on, plot3(50,50,50);
set(handles.points, 'Visible', 'off');
function RE_Callback(hObject, eventdata, handles)
axes(handles.D3)
cla
t1=0; t2=0; t3=0; l1=30; l2=20; l3=20;
dh = [0 0 0 0 0; 0 0 t1 l3 0; 0 l1 t2 0 0; 0 l2 t3 0 1];
plotbot(dh,[0 0 0 0], 'fwr');
hold on
plot3(40,40,40);
set(handles.points, 'Visible', 'on');
set(handles.pas, 'Visible', 'off');
set(handles.tp, 'Visible', 'off');
set(handles.R, 'Visible', 'on', 'Position', [39 2.077 6.0 2.077]);
set(handles.tr, 'Visible', 'on', 'Position', [38 0.385 11.2 1.231]);

```

### Programa que Calcula Jacobiano

```

function [J, Matriz]=jacob(an1,an2,ve1,ve2,ve3)
% Unidades sistema internacional m
a2=0.30; a3=0.20; ve1=ve1*pi/180; ve2=ve2*pi/180;
dxteta1=-a3*sin(an1+an2)-a2*sin(an1); dxteta2=-a3*sin(an1+an2);
dyteta1=a3*cos(an1+an2)+a2*cos(an1); dyteta2=a3*cos(an1+an2);
dztras=-1;
Matriz=[dxteta1 dxteta2 0; dyteta1 dyteta2 0; 0 0 dztras];
J=Matriz*[ve1; ve2; ve3];

```

## PROGRAMA COMUNICACION SERIAL

```
function ComuRs232(Dato1,Dato2, Dato3,Dato4)
s = serial('COM1');
set(s,'BaudRate',1200,'DataBits',8);
fopen(s);
s.Timeout=0.7; fwrite(s,Dato1,'async')
s.ReadAsyncMode = 'continuous';
leido=fscanf(s); fwrite(s,Dato2,'async'); leido=fscanf(s)
fwrite(s,Dato3,'async'); leido=fscanf(s); fwrite(s,Dato4,'async');leido=fscanf(s);
stopasync(s)
fclose(s)
```

---

Programa que Calcula la Inversa de una Posición Dada

```
function Q = inversa(X,Y)
ang1=-90; error=0;
temp=0; temp1=0; win=0; d1menor1=0.6; d1menor2=0.6; d2menor1=0.6;
d2menor2=0.6;
if X==20 & Y==30
    Q=[90 -90];error=1;
end
if X==30 & Y==20
    Q=[0 90];error=1;
end
if (X== 50 & Y==0) | (X== 0 & Y==0)
    Q=[0 0]; error=1;
end
if error==0
    l1=20; a2=30; a3=20; q1=0; q2=0;
    dhn=[0 0 q1 l1 0; 0 a2 q2 0 0; 0 a3 0 0 0];
    %Calculos Angulo 2 para los datos recibidos
    TW=[1 0 0 X; 0 1 0 Y; 0 0 1 0; 0 0 0 1];
    r2=(X^2)+(Y^2); r=sqrt(r2);
    beta=atan2(Y,r); beta=(beta*180)/pi;
    teta2=(r2-(a2^2)-(a3^2))/(2*a2*a3);
    ang2=atand(sqrt(1-(teta2^2))/teta2);
    ang2=round(ang2); teta2=ang2;
    ang1=ang1*pi/180; ang2=ang2*pi/180;

    for k=0:180
        Tnew=ikine(dhn,[ang1 ang2 0]); Tnew1=ikine(dhn,[ang1 -ang2 0]);

        d1=TR2DIFF(TW,Tnew); d2=TR2DIFF(TW,Tnew1);
```

```

d1=abs(d1); d2=abs(d2);
if (d1(1) < d1menor1 & d1(2) < d1menor2 )
    d1menor1=d1(1); d1menor2=d1(2); temp=ang1*180/pi;
else
    if (d2(1) < d2menor1 & d2(2) < d2menor2 )
        d2menor1=d2(1); d2menor2=d2(2); temp=ang2*180/pi;
    end
end
ang1=ang1 + pi/180;
end
ang1=temp*pi/180; ang2=temp*pi/180;
Tnew=fkine(dhn,[ang1 ang2 0]); Tnew1=fkine(dhn,[ang2 ang1 0]);
d1=TR2DIFF(TW,Tnew); d2=TR2DIFF(TW,Tnew1);
d1=abs(d1); d2=abs(d2);
degre=[d1(1) d1(2) d2(1) d2(2)]; mayor= d1(1);
for j=2:4
    if ( degre(j) > mayor )
        win=j; mayor=degre(j);
    end
end
if win==1 | win==2
    Q1=temp1; Q2=-1*teta2; Q=[Q1 Q2];
else
    Q1=temp; Q2=teta2; Q=[Q1 Q2];
end
end
end

```

---

Programa que Calcula las trayectorias  
function T = Trayectorias(X,Y)

```

for a=1:length(X)
    winner(a,:)=inversa(X(a),Y(a));
end
T=winner;

```

---

## ANEXO C

### PROGRAMA DE LOS MICRONCONTROLADORES

```
PROGRAMA DEL
MICRONCONTROLADOR
MAESTRO

; 4: Dim dato3 As Byte
; The address of 'dato3' is 0x32
dato3 EQU 0x32
; 5: Dim dato4 As Byte
; The address of 'dato4' is 0x33
dato4 EQU 0x33
; 6:
; 7: TRISA = 0
BSF STATUS,RP0
CLRF 0x05
BCF STATUS,RP0
; 8: TRISB = 0
BSF STATUS,RP0
CLRF 0x06
BCF STATUS,RP0
; 9: TRISC = 0
BSF STATUS,RP0
CLRF 0x07
BCF STATUS,RP0
; 10: ADCON1 = 7
BSF STATUS,RP0
MOVLW 0x07
MOVWF 0x1F
BCF STATUS,RP0
; 11: PORTB = 0
CLRF 0x06
; 12: PORTC = 0
CLRF 0x07
; 13:
; 14: For dato1 = 10 To 8 Step -1 'for-
next loop
MOVLW 0x0A
MOVWF 0x30
L0003:
MOVLW 0x08
SUBWF 0x30,W
BTFS STATUS,C
GOTO L0004

; Begin
R0L EQU 0x20
R0H EQU 0x21
R1L EQU 0x22
R1H EQU 0x23
R2L EQU 0x24
R2H EQU 0x25
R3L EQU 0x26
R3H EQU 0x27
R4L EQU 0x28
R4H EQU 0x29
R5L EQU 0x2A
R5H EQU 0x2B
SO_PORT EQU 0x2C
SO_BIT EQU 0x2D
SO_INTI EQU 0x2F
SO_INTH EQU 0x2F
ORG 0x0000
BCF PCLATH,3
BCF PCLATH,4
GOTO L0002
ORG 0x0004
RETFIE
L0002:
; 1: Define SEROUT_DELAYUS =
1000 'WaitMs 1000 this delay should
be used in a real device
; 2: Dim dato1 As Byte 'declare a
variable
; The address of 'dato1' is 0x30
dato1 EQU 0x30
; 3: Dim dato2 As Byte
; The address of 'dato2' is 0x31
dato2 EQU 0x31
```

; 15: Serout PORTA.1, 9600, "N#: ",  
#dato1, CrLf'send formatted output to  
PORTB.1 (TX line of software UART)

```
BSF STATUS,RP0
BCF TRISA,1
BCF STATUS,RP0
BSF PORTA,1
MOVLW 0x05
MOVWF SO_PORT
MOVLW 0x02
MOVWF SO_BIT
MOVLW 0x5B
MOVWF SO_INTL
MOVLW 0x00
MOVWF SO_INTH
MOVI W 0x4F
CALL SO01
MOVLW 0x23
CALL SO01
MOVLW 0x3A
CALL SO0i
MOVLW 0x20
CALL SO0i
MOVF 0x30,W
MOVWF R2L
CLRF R2H
CALL SO21
MOVLW 0x0D
CALL SO01
MOVLW 0x0A
CALL SO01
```

; 16: 'WaitMs 500 'this delay should  
be used in a real device

; 17: Next dato1

```
MOVLW 0x01
SUBWF 0x30,F
BTFSC STATUS,C
GOTO L0003
```

L0004: MOVLW 0x1F  
ANDWF STATUS,F

; 18:

; 19: dato1 = 0  
CLRF 0x30

; 20: dato2 = 0  
CLRF 0x31

; 21: dato3 = 0  
CLRF 0x32

; 22: dato4 = 0  
CLRF 0x33

; 23:

; 24: ioop:

L0001:

; 25:

; 26: Serin PORTA.2, 9600, dato1  
'wait to receive a byte on PORTB.2  
(RX line of software UART)

```
BSF STATUS,RP0
BSF TRISA,2
BCF STATUS,RP0
MOVLW 0x05
MOVWF SO_PORT
MOVLW 0x04
MOVWF SO_BIT
MOVLW 0x5B
MOVWF SO_INTL
MOVLW 0x00
MOVWF SO_INTH
CALL Si0i
MOVWF 0x30
```

; 27: Serout PORTA.1, 9600, "OK: ",  
#dato1, CrLf

```
BSF STATUS,RP0
BCF TRISA,1
BCF STATUS,RP0
BSF PORTA,1
MOVLW 0x05
MOVWF SO_PORT
MOVI W 0x07
MOVWF SO_BIT
MOVLW 0x5B
MOVWF SO_INTL
MOVLW 0x00
MOVWF SO_INTH
MOVLW 0x4F
CALL SO01
MOVLW 0x4B
CALL SO0i
MOVLW 0x3A
CALL SO0i
MOVLW 0x20
```

```

CALL SO01
MOVWF 0x30,W
MOVWF R2L
CLRF R2H
CALL SO21
MOVLW 0x0D
CALL SO01
MOVLW 0x0A
CALL SO01
; 28: Serin PORTA.2, 9600, dato2
BSF STATUS,RP0
BSF TRISA,2
BCF STATUS,RP0
MOVLW 0x05
MOVWF SO_PORT
MOVI W 0x04
MOVWF SO_BIT
MOVLW 0x5B
MOVWF SO_INTL
MOVLW 0x00
MOVWF SO_INTH
CALL SI01
MOVWF 0x31
; 29: Serout PORTA.1, 9600, "OK: ",
#dato2, CrLf
BSF STATUS,RP0
BCF TRISA,1
BCF STATUS,RP0
BSF PORTA,1
MOVLW 0x05
MOVWF SO_PORT
MOVLW 0x02
MOVWF SO_BIT
MOVLW 0x5B
MOVWF SO_INTL
MOVLW 0x00
MOVWF SO_INTH
MOVLW 0x4F
CALL SO01
MOVLW 0x4B
CALL SO01
MOVLW 0x3A
CALL SO01
MOVLW 0x20
CALL SO01

```

```

MOVWF 0x31,W
MOVWF R2L
CLRF R2H
CALL SO21
MOVLW 0x0D
CALL SO01
MOVLW 0x0A
CALL SO01
; 30: Serin PORTA.2, 9600, dato3
'wait to receive a byte on PORTB.2
(RX line of software UART)
BSF STATUS,RP0
BSF TRISA,2
BCF STATUS,RP0
MOVLW 0x05
MOVWF SO_PORT
MOVLW 0x04
MOVWF SO_BIT
MOVLW 0x5B
MOVWF SO_INTL
MOVLW 0x00
MOVWF SO_INTH
CALL SI01
MOVWF 0x32
; 31: Serout PORTA.1, 9600, "OK: ",
#dato3, CrLf
BSF STATUS,RP0
BCF TRISA,1
BCF STATUS,RP0
BSF PORTA,1
MOVLW 0x05
MOVWF SO_PORT
MOVLW 0x02
MOVWF SO_BIT
MOVI W 0x5B
MOVWF SO_INTL
MOVLW 0x00
MOVWF SO_INTH
MOVLW 0x4F
CALL SO01
MOVLW 0x4B
CALL SO01
MOVLW 0x3A
CALL SO01
MOVLW 0x20

```

```

CALL SO01
MOVWF 0x32,W
MOVWF R2L
CLRF R2H
CALL SO21
MOVLW 0x0D
CALL SO01
MOVLW 0x0A
CALL SO01
; 32: Serin PORTA.2, 9600, dato4
BSF STATUS,RP0
BSF TRISA,2
BCF STATUS,RP0
MOVLW 0x05
MOVWF SO_PORT
MOVI W 0x04
MOVWF SO_BIT
MOVLW 0x5B
MOVWF SO_INTL
MOVLW 0x00
MOVWF SO_INTH
CALL SI01
MOVWF 0x33
; 33: Serout PORTA.1, 9600, "OK: ",
#dato4, CrLf
BSF STATUS,RP0
BCF TRISA,1
BCF STATUS,RP0
BSF PORTA,1
MOVLW 0x05
MOVWF SO_PORT
MOVLW 0x02
MOVWF SO_BIT
MOVLW 0x5B
MOVWF SO_INTL
MOVLW 0x00
MOVWF SO_INTH
MOVLW 0x4F
CALL SO01
MOVLW 0x4B
CALL SO01
MOVLW 0x3A
CALL SO01
MOVLW 0x20
CALL SO01

```

```

MOVWF 0x33,W
MOVWF R2L
CLRF R2H
CALL SO21
MOVLW 0x0D
CALL SO01
MOVLW 0x0A
CALL SO01
; 34: PORTB = dato1
MOVWF 0x30,W
MOVWF 0x06
; 35: PORTC = dato2
MOVWF 0x31,W
MOVWF 0x07
; 36: Goto loop
GOTO L0001
; End of program
L0005: GOTO L0005
; Division Routine
D001: MOVLW 0x10
MOVWF R3L
CLRF R2H
CLRF R2L
D002: RLF R0H,W
RLF R2L,F
RLF R2H,F
MOVWF R1L,W
SUBWF R2L,F
MOVWF R1H,W
BTFSS STATUS,C
INCFSZ R1H,W
SUBWF R2H,F
BTFSC STATUS,C
GOTO D003
MOVWF R1L,W
ADDWF R2L,F
MOVWF R1H,W
BTFSC STATUS,C
INCFSZ R1H,W
ADDWF R2H,F
BCF STATUS,C
D003: RLF R0L,F
RLF R0H,F
DECFSZ R3L,F
GOTO D002

```

```

        MOVF R0L,W
        RETURN
; Waitus Routine - Word Argument
Y001: MOVLW 0x10
        SUBWF R4L,F
        CLRW
        BTFSS STATUS,C
        ADDLW 0x01
        SUBWF R4H,F
        BTFSS STATUS,C
        RETURN
        GOTO Y002
Y002: MOVLW 0x0A
        SUBWF R4L,F
        CLRW
        BTFSS STATUS,C
        ADDLW 0x01
        SUBWF R4H,F
        BTFSS STATUS,C
        RETURN
        GOTO Y002
; Serout Routine
SO01: MOVWF R5L
        MOVLW 0xE8
        MOVWF R4L
        MOVLW 0x03
        MOVWF R4H
        CALL Y001
        MOVF SO_PORT,W
        MOVWF FSR
        MOVLW 0x09
        MOVWF R5H
        BCF STATUS,C
SO02: CALL SO03
        RRF R5L,F
        DECFSZ R5H,F
        GOTO SO02
        BSF STATUS,C
        CALL SO03
        RETURN
SO03: MOVF INDF,W
        IORWF SO_BIT,W
        BTFSS STATUS,C
        XORWF SO_BIT,W
        MOVWF INDF

```

```

        MOVF SO_INTL,W
        MOVWF R4L
        MOVF SO_INTH,W
        MOVWF R4H
        CALL Y001
        RETURN
; Serout Decimal Conversion Routine
SO21: BSF R3H,7
        MOVLW 0x27
        MOVWF R1H
        MOVLW 0x10
        CALL SO22
        MOVLW 0x03
        MOVWF R1H
        MOVLW 0xE8
        CALL SO22
        CLRF R1H
        MOVLW 0x64
        CALL SO22
        CLRF R1H
        MOVLW 0x0A
        CALL SO22
        MOVF R2L,W
        GOTO SO23
SO22: MOVWF R1L
        MOVF R2H,W
        MOVWF R0H
        MOVF R2L,W
        MOVWF R0L
        CALL D001
        MOVF R0L,W
        BTFSS STATUS,Z
        RCF R3H,7
        BTFSC R3H,7
        RETURN
SO23: ADDLW 0x30
        CALL SO01
        RETURN
; Serin Routine
SI01: CALL SI03
        BTFSC STATUS,C
        GOTO SI01
        MOVF SO_INTL,W
        MOVWF R4L
        MOVF SO_INTH,W

```

```

MOVWF R4H
RRF R4H,F
RRF R4L,F
BCF STATUS,C
RRF R4H,F
RRF R4L,F
CALL Y001
MOVLW 0x08
MOVWF R5H
SI02: MOVF SO_INTL,W
MOVWF R4L
MOVF SO_INTH,W
MOVWF R4H
CALL Y001
CALL SI03
RRF R5L,F
DECFSZ R5H,F
GOTO SI02
MOVF SO_INTL,W
MOVWF R4L
MOVF SO_INTH,W
MOVWF R4H
CALL Y001
MOVF R5L,W
RETURN
SI03: MOVF SO_PORT,W
MOVWF FSR
MOVF SO_BIT,W
ANDWF INDF,W
ADDLW 0xFF
RETURN
; End of listing
END

```

PROGRAMA DE  
MICROCONTROLADOR ES  
ESCLAVOS

; Begin

```

R0L EQU 0xC
R0H EQU 0xD
R1L EQU 0xE
R1H EQU 0xF

```

```

R2L EQU 0x10
R2H EQU 0x11
R3L EQU 0x12
R3H EQU 0x13
R4L EQU 0x14
R4H EQU 0x15
R5L EQU 0x16
R5H EQU 0x17
ORG 0x0000
BCF PCLATH,3
BCF PCLATH,4
GOTO L0003
ORG 0x0004
RETFIE

```

L0003:

```

; 1: Dim angnew As Byte 'declare a
variable
;   The address of 'angnew' is 0x18
   angnew EQU 0x18
; 2: Dim angprevious As Byte 'declare
a variable
;   The address of 'angprevious' is
0x19
   angprevious EQU 0x19
; 3: Dim paso As Byte 'declare a
variable
;   The address of 'paso' is 0x1A
   paso EQU 0x1A
; 4: Dim s As Word 'declare a variable
;   The address of 's' is 0x1B
   s EQU 0x1B
; 5: Dim y As Word
;   The address of 'y' is 0x1D
   y EQU 0x1D
; 6: Dim x As Word
;   The address of 'x' is 0x1F
   x EQU 0x1F

```

; 7:

```

; 8: TRISA = 0
   BSF STATUS,RP0
   CLRF 0x05
   BCF STATUS,RP0
; 9: TRISB = 255
   BSF STATUS,RP0
   MOVLW 0xFF

```

```

        MOVWF 0x06
        BCF STATUS,RP0
; 10: PORTB = 0
        CLRF 0x06
; 11: s = 30000
        MOVLW 0x30
        MOVWF 0x1B
        MOVLW 0x75
        MOVWF 0x1C
; 12: angnew = 0
        CLRF 0x18
; 13: angprevious = 0
        CLRF 0x19
; 14: paso = 0
        CLRF 0x1A
; 15:
; 16:
; 17: loop:
L0001:
; 18: angprevious = angnew
        MOVF 0x18,W
        MOVWF 0x19
; 19: angnew = PORTB
        MOVF 0x06,W
        MOVWF 0x18
; 20: If angprevious = PORTB Then
        MOVF 0x19,W
        SUBWF 0x06,W
        BTFSS STATUS,Z
        GOTO L0004
; 21: y = (10 * angprevious) + 800
        MOVLW 0x0A
        MOVWF R3L
        CLRF R3H
        MOVF 0x19,W
        MOVWF R1L
        CLRF R1H
        CALL M001
        MOVWF 0x1D
        MOVF R2H,W
        MOVWF 0x1E
        MOVF 0x1D,W
        ADDLW 0x20
        MOVWF 0x1D
        MOVF 0x1E,W
        BTFSC STATUS,C
        ADDLW 0x01
        ADDLW 0x03
        MOVWF 0x1E
; 22: x = 44000 - y
        MOVF 0x1D,W
        SUBLW 0xE0
        MOVWF 0x1F
        MOVF 0x1E,W
        BTFSS STATUS,C
        ADDLW 0x01
        SUBLW 0xAB
        MOVWF 0x20
; 23: Gosub pwm
        CALL L0002
; 24: Else
        GOTO L0005
L0004:
        MOVLW 0x1F
        ANDWF STATUS,F
; 25: If angnew >= angprevious Then
        MOVF 0x19,W
        SUBWF 0x18,W
        BTFSS STATUS,C
        GOTO L0006
; 26: paso = angprevious + 1
        MOVF 0x19,W
        ADDLW 0x01
        MOVWF 0x1A
; 27: While paso < angnew
L0007:
        MOVF 0x18,W
        SUBWF 0x1A,W
        BTFSC STATUS,C
        GOTO L0008
; 28: y = (10 * paso) + 800
        MOVLW 0x0A
        MOVWF R3L
        CLRF R3H
        MOVF 0x1A,W
        MOVWF R1L
        CLRF R1H
        CALL M001
        MOVWF 0x1D
        MOVF R2H,W
        MOVWF 0x1E

```

```

MOVF 0x1D,W
ADDLW 0x20
MOVWF 0x1D
MOVF 0x1E,W
BTFSC STATUS,C
ADDLW 0x01
ADDLW 0x03
MOVWF 0x1E
; 29: x = s - y
MOVF 0x1D,W
SUBWF 0x1B,W
MOVWF 0x1F
MOVF 0x1E,W
BTFSS STATUS,C
ADDLW 0x01
SUBWF 0x1C,W
MOVWF 0x20
; 30: s = s + 60
MOVF 0x1B,W
ADDLW 0x3C
MOVWF 0x1B
MOVF 0x1C,W
BTFSC STATUS,C
ADDLW 0x01
ADDLW 0x00
MOVWF 0x1C
; 31: paso = paso + 1
MOVF 0x1A,W
ADDLW 0x01
MOVWF 0x1A
; 32: Gosub pwm
CALL L0002
; 33: Wend
GOTO L0007
L0008:    MOVLW 0x1F
ANDWF STATUS,F
; 34: y = (10 * angnew) + 800
MOVLW 0x0A
MOVWF R3L
CLRF R3H
MOVF 0x18,W
MOVWF R1L
CLRF R1H
CALL M001
MOVWF 0x1D
MOVF R2H,W
MOVWF 0x1E
MOVF 0x1D,W
ADDLW 0x20
MOVWF 0x1D
MOVF 0x1E,W
BTFSC STATUS,C
ADDLW 0x01
ADDLW 0x03
MOVWF 0x1E
; 35: x = 44000 - y
MOVF 0x1D,W
SUBLW 0xE0
MOVWF 0x1F
MOVF 0x1E,W
BTFSS STATUS,C
ADDLW 0x01
SUBLW 0xAB
MOVWF 0x20
; 36: Gosub pwm
CALL L0002
; 37: Else
GOTO L0009
L0006:    MOVLW 0x1F
ANDWF STATUS,F
; 38: paso = angprevious - 1
MOVLW 0x01
SUBWF 0x19,W
MOVWF 0x1A
; 39: While paso > angnew
L0010:    MOVF 0x1A,W
SUBWF 0x18,W
BTFSC STATUS,C
GOTO L0011
; 40: y = (10 * paso) + 800
MOVLW 0x0A
MOVWF R3L
CLRF R3H
MOVF 0x1A,W
MOVWF R1L
CLRF R1H
CALL M001
MOVWF 0x1D
MOVF R2H,W

```

```

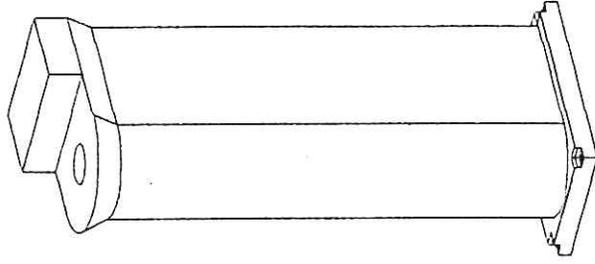
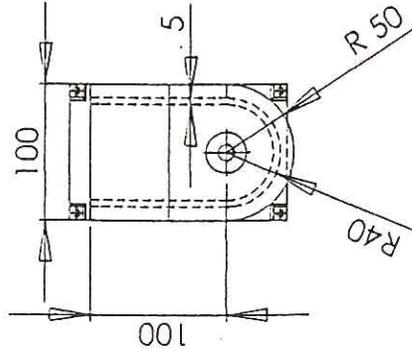
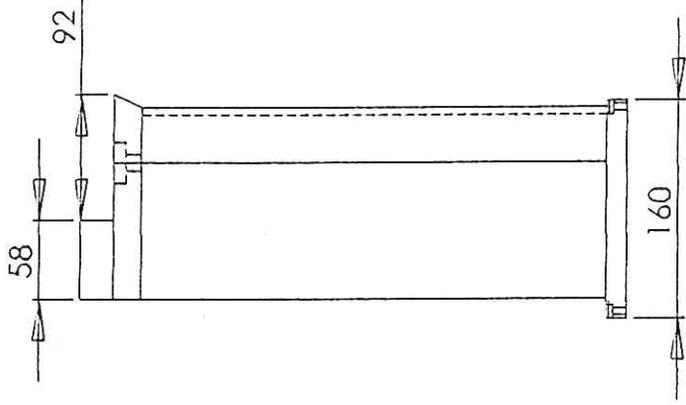
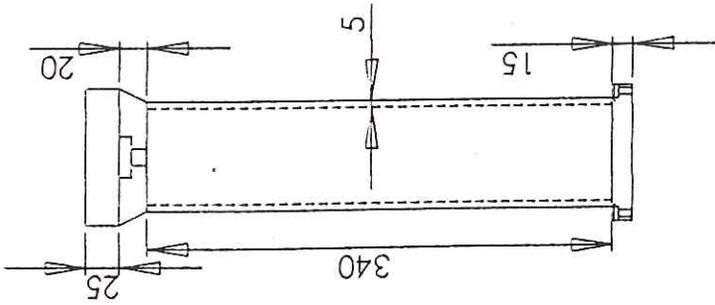
MOVWF 0x1E
MOVF 0x1D,W
ADDLW 0x20
MOVWF 0x1D
MOVF 0x1E,W
BTFSC STATUS,C
ADDLW 0x01
ADDLW 0x03
MOVWF 0x1E
; 41: x = s - y
MOVF 0x1D,W
SUBWF 0x1B,W
MOVWF 0x1F
MOVF 0x1E,W
BTFSS STATUS,C
ADDLW 0x01
SUBWF 0x1C,W
MOVWF 0x20
; 42: s = s + 60
MOVF 0x1B,W
ADDLW 0x3C
MOVWF 0x1B
MOVF 0x1C,W
BTFSC STATUS,C
ADDLW 0x01
ADDLW 0x00
MOVWF 0x1C
; 43: paso = paso - 1
MOVLW 0x01
SUBWF 0x1A,W
MOVWF 0x1A
; 44: Gosub pwm
CALL L0002
; 45: Wend
GOTO L0010
L0011:    MOVLW 0x1F
ANDWF STATUS,F
; 46: y = (10 * angnew) + 800
MOVLW 0x0A
MOVWF R3L
CLRF R3H
MOVF 0x18,W
MOVWF R1L
CLRF R1H
CALL M001
MOVWF 0x1D
MOVF R2H,W
MOVWF 0x1E
MOVF 0x1D,W
ADDLW 0x20
MOVWF 0x1D
MOVF 0x1E,W
BTFSC STATUS,C
ADDLW 0x01
ADDLW 0x03
MOVWF 0x1E
; 47: x = 44000 - y
MOVF 0x1D,W
SUBLW 0xE0
MOVWF 0x1F
MOVF 0x1F,W
BTFSS STATUS,C
ADDLW 0x01
SUBLW 0xAB
MOVWF 0x20
; 48: Gosub pwm
CALL L0002
; 49: Endif
L0009:    MOVLW 0x1F
ANDWF STATUS,F
; 50: Endif
L0005:    MOVLW 0x1F
ANDWF STATUS,F
; 51: Goto loop
GOTO L0001
; 52: End
L0012:    GOTO L0012
; 53: pwm:
L0002:
; 54: PORTA = 5
MOVLW 0x05
MOVWF 0x05
; 55: WaitUs y
MOVF 0x1D,W
MOVWF R4L
MOVF 0x1E,W
MOVWF R4H
CALL Y001
; 56: PORTA = 0
CLRF 0x05

```

```

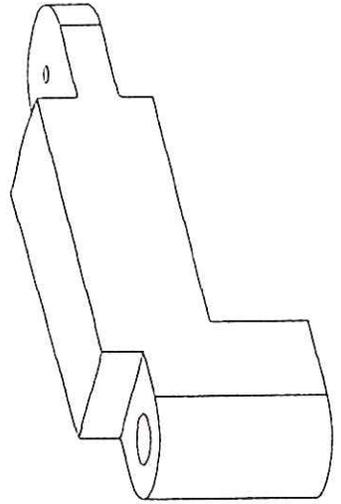
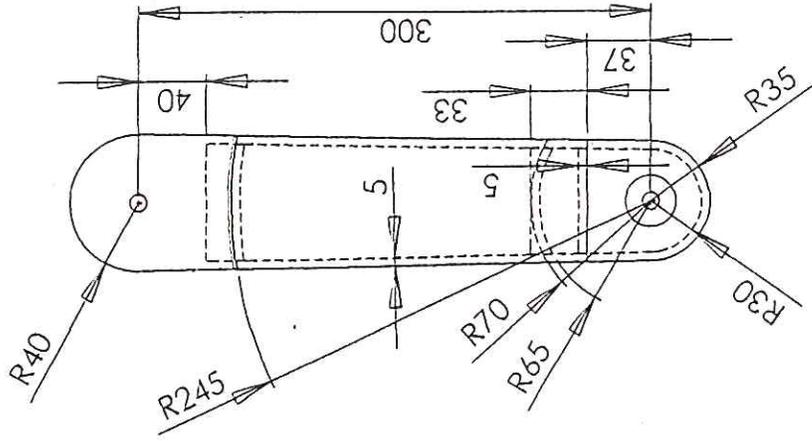
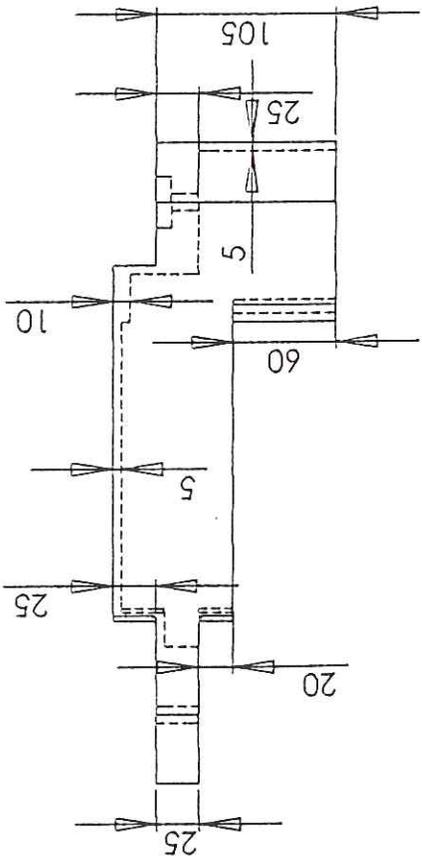
; 57: WaitUs x
    MOVF 0x1F,W
    MOVWF R4L
    MOVF 0x20,W
    MOVWF R4H
    CALL Y001
; 58: Return
    RETURN
; 59:
; End of program
L0013:    GOTO L0013
; Multiplication Routine
M001: MOVLW 0x10
    MOVWF R4L
    CLRF R0H
    CLRF R0L
M002: RRF R3H,F
    RRF R3L,F
    BTFSS STATUS,C
    GOTO M003
    MOVF R1L,W
    ADDWF R0L,F
    MOVF R1H,W
    BTFSC STATUS,C
    INCFSZ R1H,W
    ADDWF R0H,F
M003: RRF R0H,F
    RRF R0L,F
    RRF R2H,F
    RRF R2L,F
    DECFSZ R4L,F
    GOTO M002
    MOVF R2L,W
    RETURN
; Waitus Routine - Word Argument
Y001: MOVLW 0x10
    SUBWF R4L,F
    CLRW
    BTFSS STATUS,C
    ADDLW 0x01
    SUBWF R4H,F
    BTFSS STATUS,C
    RETURN
    GOTO Y002
Y002: MOVLW 0x0A
    SUBWF R4L,F
    CLRW
    BTFSS STATUS,C
    ADDLW 0x01
    SUBWF R4H,F
    BTFSS STATUS,C
    RETURN
    GOTO Y002
; End of listing
END

```

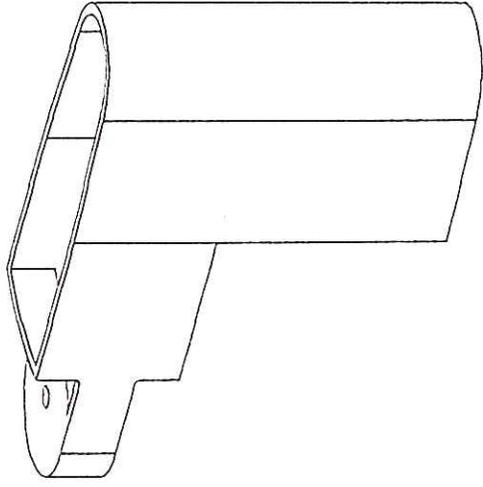
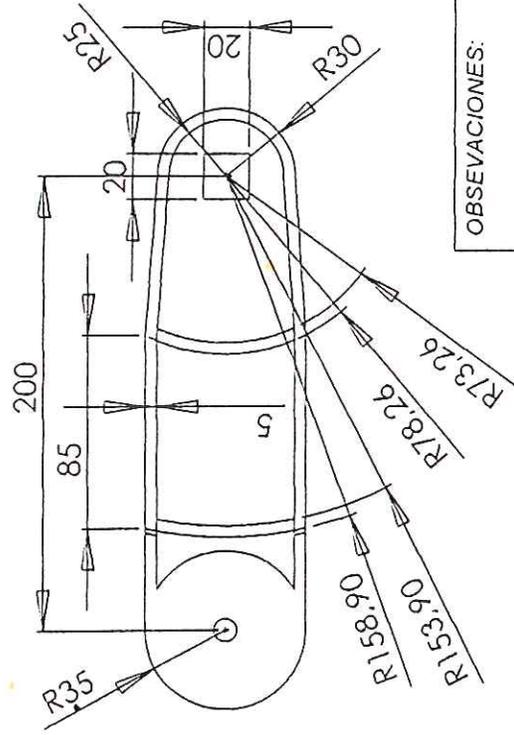
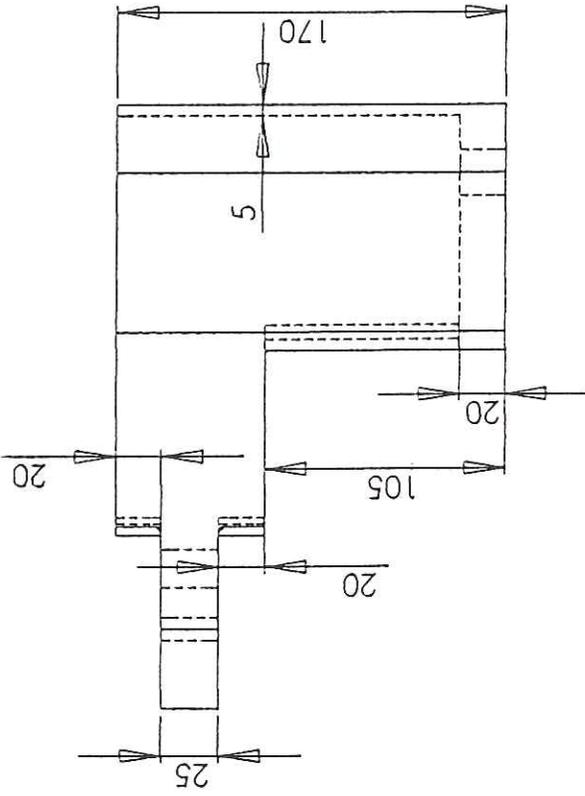


OBSERVACIONES:	DIBUJO:	Miguel A. Romero Jaime A. Garcia
	DISEÑO:	Miguel A. Romero Jaime A. Garcia
NORMA: ISO	APROBO:	Ing. Jhon F. Archila
	FECHA:	15/08/05
	PLANO No. Base	

DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR SCARA		ESC: 1:5
UNIVERSIDAD AUTONOMA DE BUCARAMANGA FACULTAD DE INGENIERIA MECATRONICA		CONVENCIONES



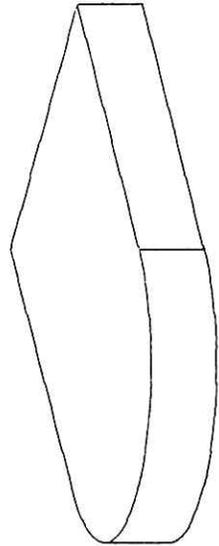
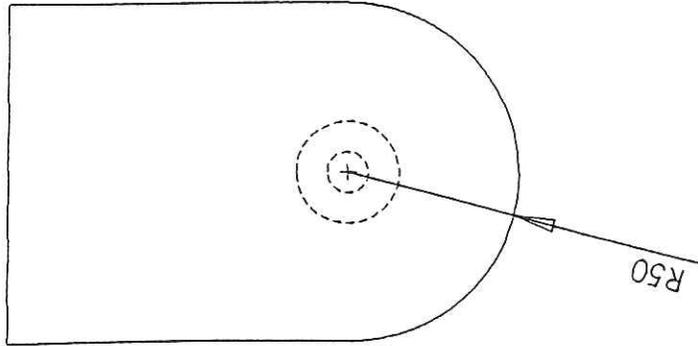
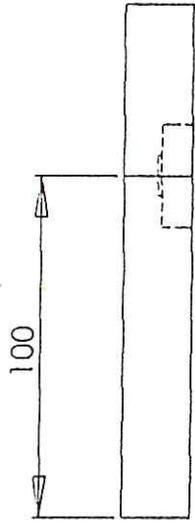
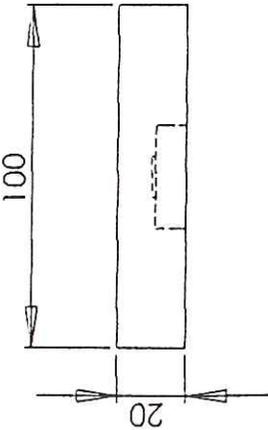
OBSERVACIONES:	DIBUJO:	Miguel A. Romero Jaime A. Garcia	DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR SCARA
	DISEÑO:	Miguel A. Romero Jaime A. Garcia	
NORMA: ISO	APROBO:	Ing. Jhon F. Archila	UNIVERSIDAD AUTONOMA DE BUCARAMANGA FACULTAD DE INGENIERIA MECATRONICA
	FECHA:	15/08/05	
	PLANO No.	Antebrazo	



OBSEVACIONES:	
NORMA:	ISO

DIBUJO:	Miguel A. Romero Jaime A. Garcia
DISEÑO:	Miguel A. Romero Jaime A. Garcia
APROBO:	Ing. Jhon F. Archila
FECHA:	15/08/05
PLANO No.	Brazo

DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR SCARA	
UNIVERSIDAD AUTONOMA DE BUCARAMANGA FACULTAD DE INGENIERIA MECATRONICA	
ESC:	1:3
CONVENCIONES	



OBSEVACIONES:

DIBUJO: Miguel A. Romero  
Jaime A. Garcia

DISEÑO: Miguel A. Romero  
Jaime A. Garcia

APROBO: Ing. Jhon F. Archila

FECHA: 15/08/05

PLANO No. Soporte Antebrazo

NORMA:

ISO

DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN  
MANIPULADOR SCARA

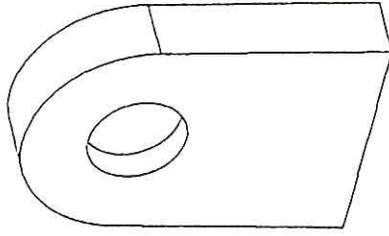
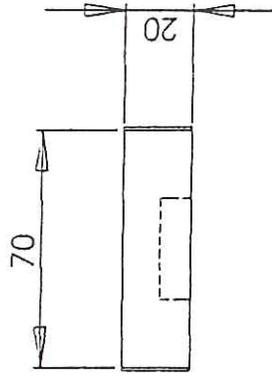
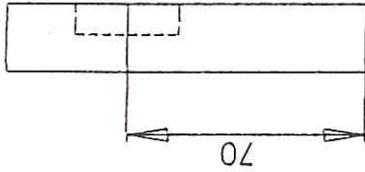
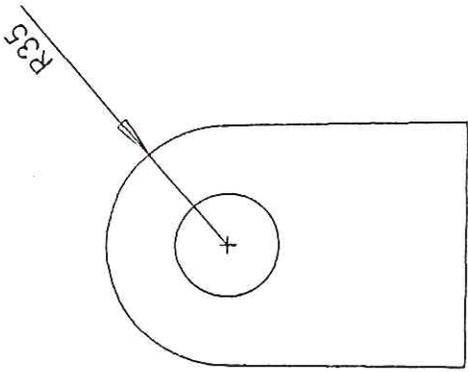
UNIVERSIDAD AUTONOMA DE  
BUCCARAMANGA  
FACULTAD DE INGENIERIA  
MECATRONICA

ESC: 1:2

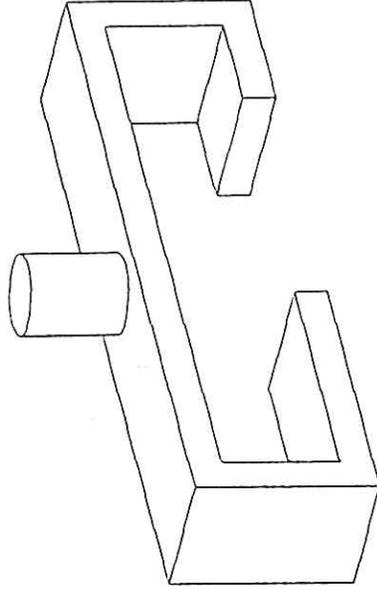
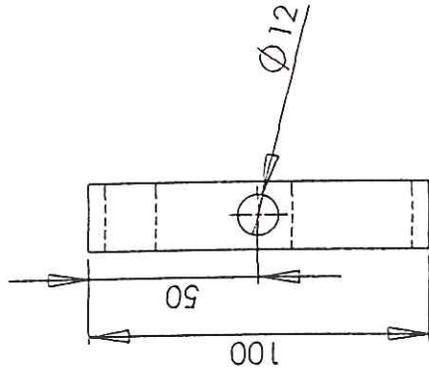
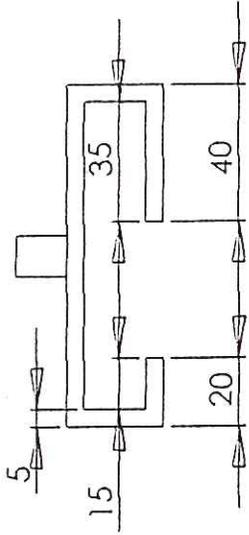
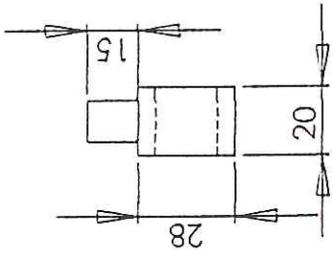
CONVENCIONES

2

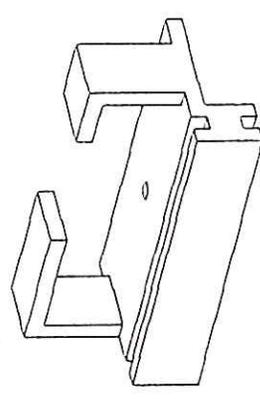
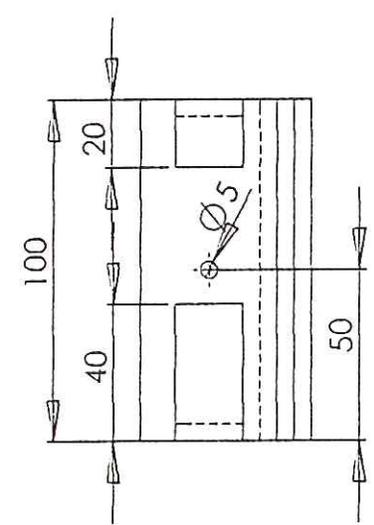
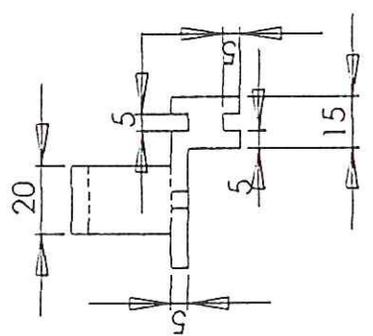
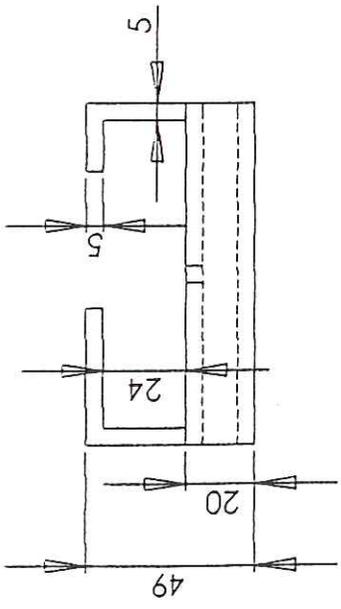
1



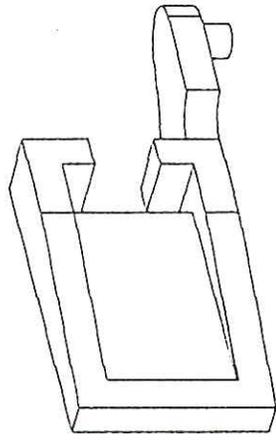
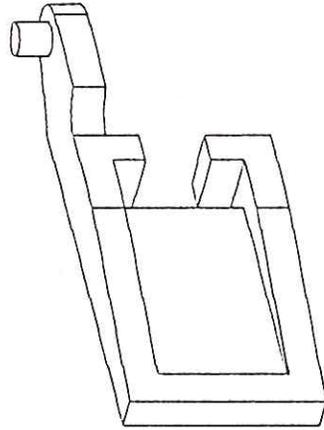
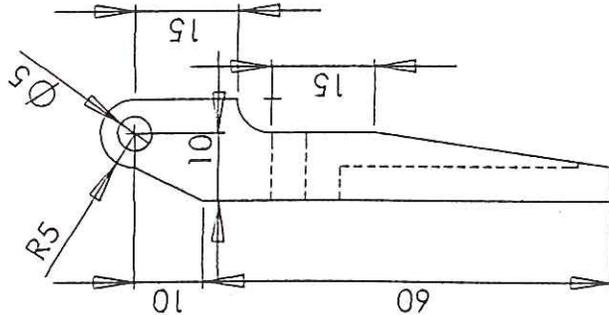
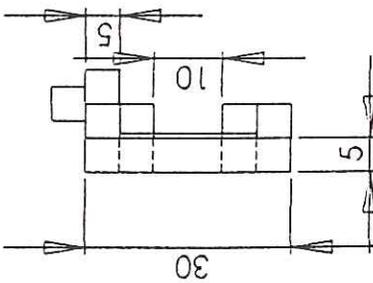
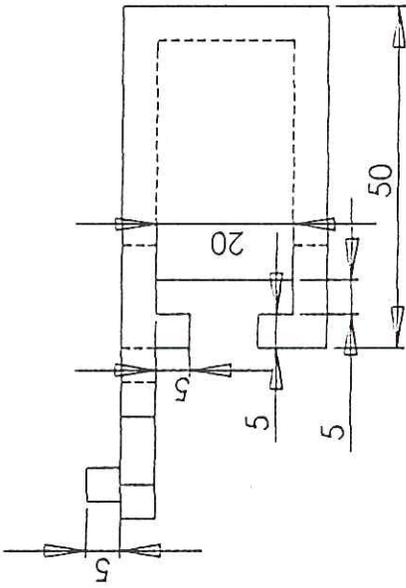
OBSERVACIONES:	DIBUJO:	Miguel A. Romero Jaime A. Garcia	DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR SCARA
	DISEÑO:	Miguel A. Romero Jaime A. Garcia	
NORMA:	APROBO:	Ing. Jhon F. Archila	UNIVERSIDAD AUTONOMA DE BUCARAMANGA FACULTAD DE INGENIERIA MECATRONICA
	FECHA:	15/08/05	
	PLANO No. Soporte Brazo		ESC: 1:2 CONVENCIONES



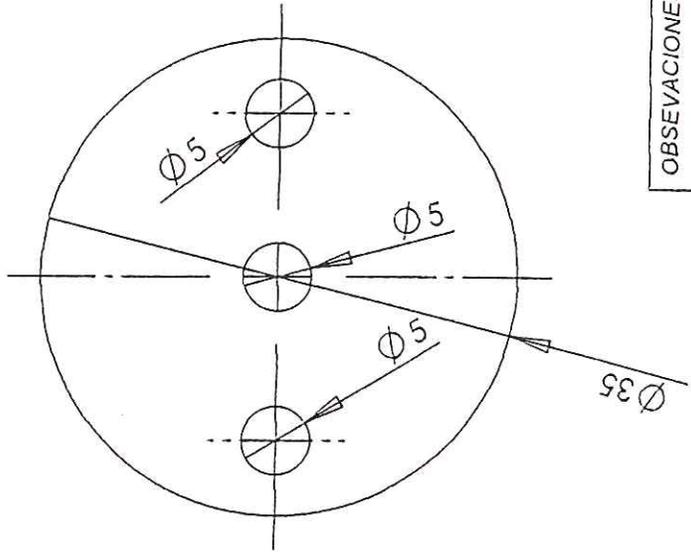
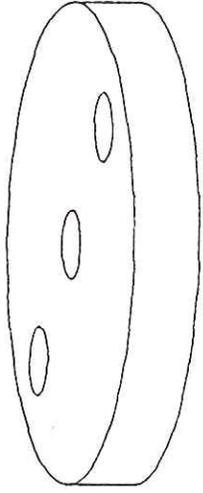
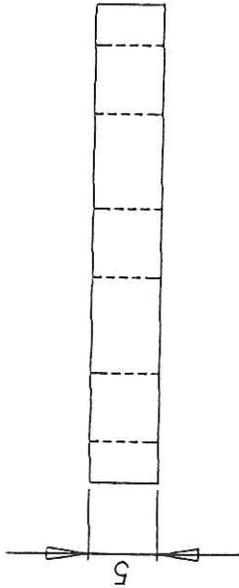
OBSERVACIONES:	DIBUJO:	Miguel A. Romero Jaime A. Garcia	DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR SCARA	ESC: 1:2
	DISEÑO:	Miguel A. Romero Jaime A. Garcia		
NORMA:	APROBO:	Ing. Jhon F. Archila	UNIVERSIDAD AUTONOMA DE BUCARAMANGA FACULTAD DE INGENIERIA MECATRONICA	CONVENCIONES
	FECHA:	15/08/05		
ISO	PLANO No. Soporte Pinza			



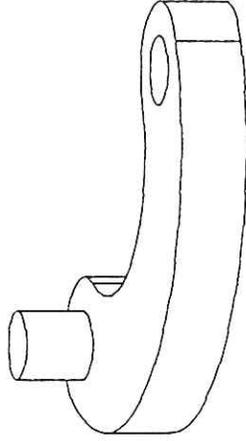
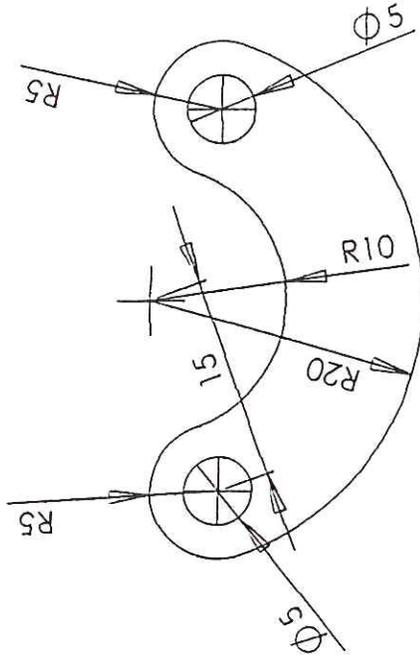
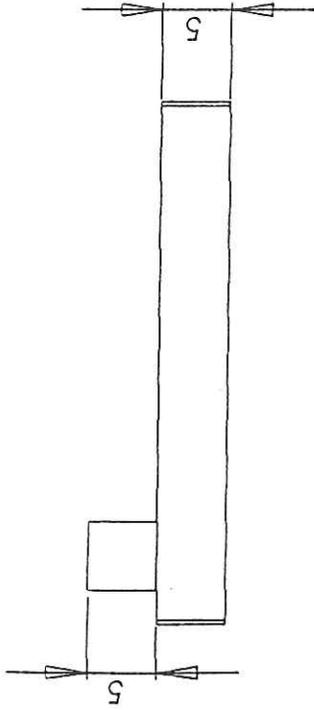
OBSEVACIONES:	DIBUJO:	Miguel A. Romero Jairne A. Garcia	DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR SCARA
	DISEÑO:	Miguel A. Romero Jairne A. Garcia	
NORMA:	APROBO:	Ing. Jhon F. Archila	UNIVERSIDAD AUTONOMA DE BUCARAMANGA FACULTAD DE INGENIERIA MECATRONICA
		FECHA: 15/08/05	
		PLANO No. Base Pinza	ESC: 1:2 CONVENCIONES



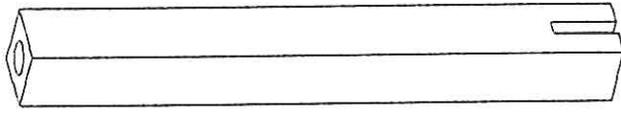
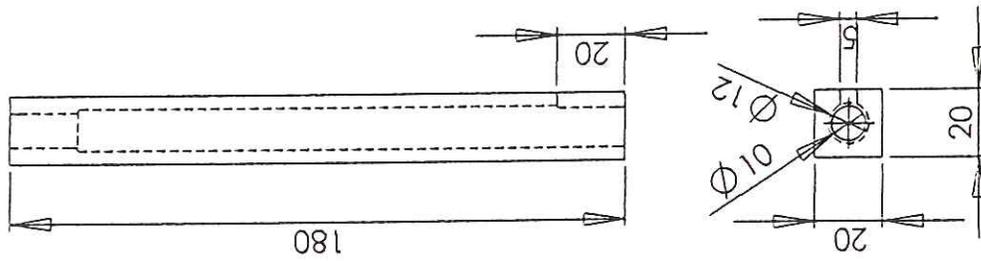
OBSEVACIONES:	DIBUJO:	Miguel A. Romero Jaime A. Garcia	DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR SCARA	
	DISEÑO:	Miguel A. Romero Jaime A. Garcia		
NORMA:	APROBO:	Ing. Jhon F. Archila	UNIVERSIDAD AUTONOMA DE BUCARAMANGA FACULTAD DE INGENIERIA MECATRONICA	
	FECHA:	15/08/05		ESC: 1:1
	PLANO No. Pinzas			CONVENCIONES



OBSERVACIONES:	DIBUJO:	Miguel A. Romero Jaimé A. García	DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR SCARA
	DISEÑO:	Miguel A. Romero Jaimé A. García	
NORMA: ISO	APROBO:	Ing. Jhon F. Archila	UNIVERSIDAD AUTONOMA DE BUCARAMANGA FACULTAD DE INGENIERIA MECATRONICA
	FECHA:	15/08/05	
		PLANO No. Circulo Pinza	ESC: 2:1 CONVENCIONES

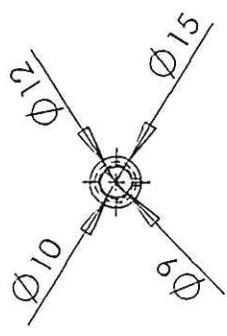
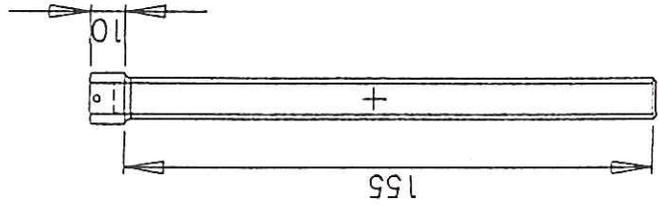
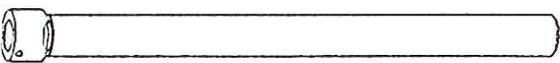


OBSERVACIONES:	DIBUJO:	Miguel A. Romero Jaime A. Garcia	DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR SCARA	ESC: 2:1 CONVENCIONES
	DISEÑO:	Miguel A. Romero Jaime A. Garcia		
NORMA: ISO	APROBO:	Ing. Jhon F. Archila	UNIVERSIDAD AUTONOMA DE BUCARAMANGA FACULTAD DE INGENIERIA MECATRONICA	
	FECHA:	15/08/05		
	PLANO No.Union Pinza			

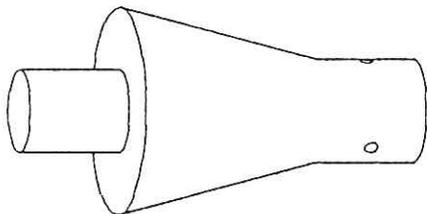
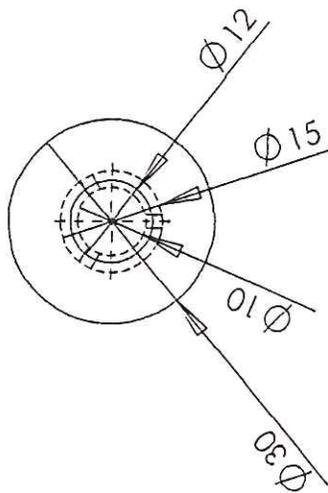
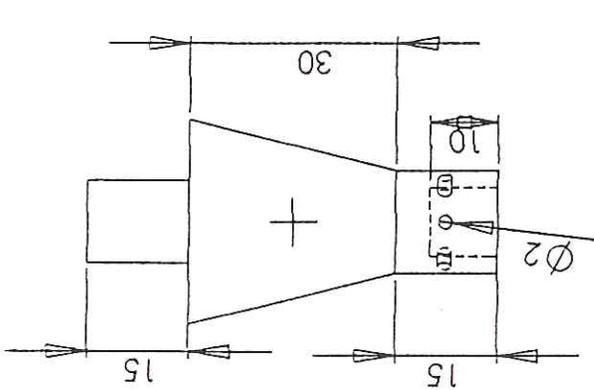


OBSEVACIONES:	DIBUJO: Miguel A. Romero Jaime A. Garcia	DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR SCARA
	DISEÑO: Miguel A. Romero Jaime A. Garcia	
	APROBO: Ing. Jhon F. Archila	UNIVERSIDAD AUTONOMA DE BUCCARAMANGA FACULTAD DE INGENIERIA MECATRONICA
NORMA:	FECHA: 15/08/05	
	PLANO No. Guia	ESC: 1:2 CONVENCIONES

A	B	C
3	4	5
1	2	3
4	5	6
7	8	9

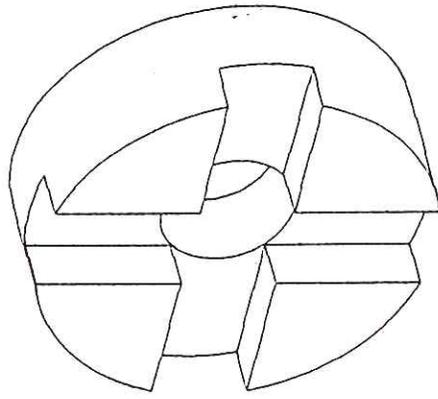
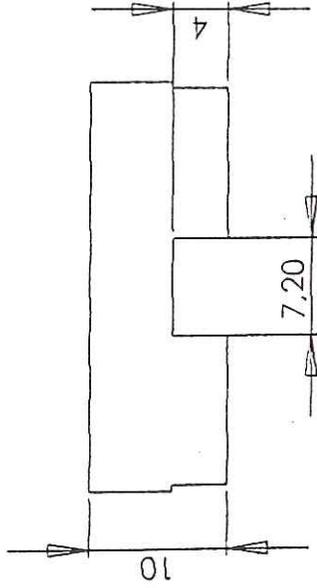
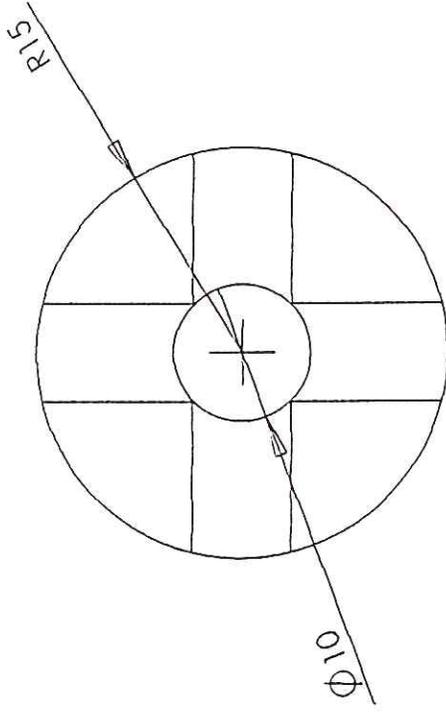


DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR SCARA	UNIVERSIDAD AUTONOMA DE BUCARAMANGA FACULTAD DE INGENIERIA MECATRONICA
ESC: 1:2 CONVENCIONES	
DIBUJO: Miguel A. Romero Jairine A. Garcia	DISEÑO: Miguel A. Romero Jairine A. Garcia
APROBO: Ing. Jhon F. Archila	FECHA: 15/08/05
NORMA: ISO	PLANO No.Eje Roscado

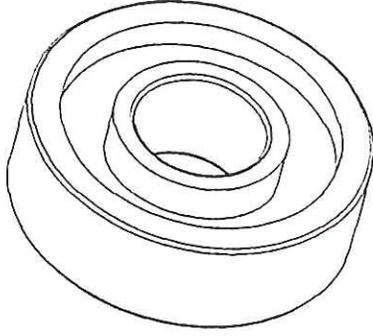
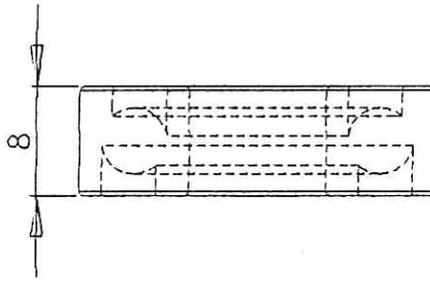
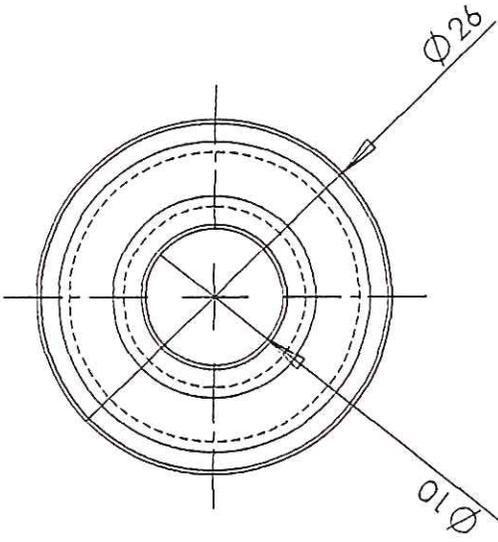


OBSEVACIONES:	DIBUJO: Miguel A. Romero Jaime A. Garcia	DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR SCARA
NORMA: ISO	DISEÑO: Miguel A. Romero Jaime A. Garcia	

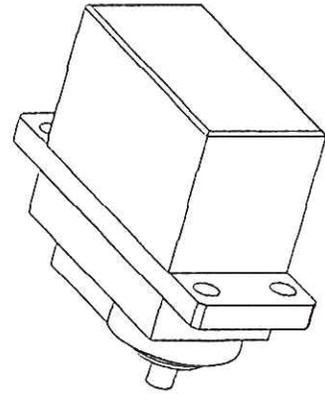
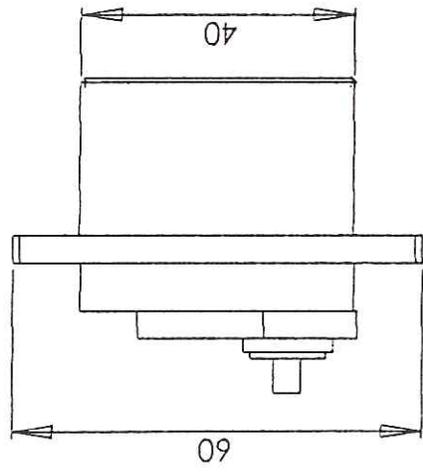
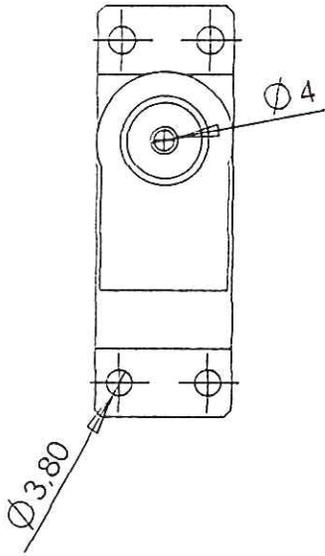
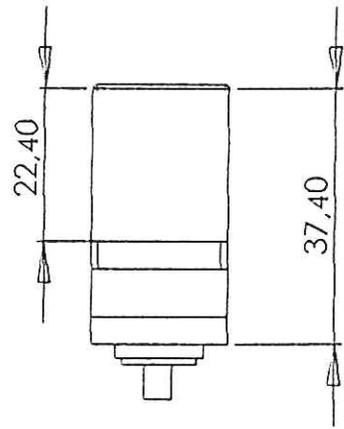
APROBO: Ing. Jhon F. Archila	UNIVERSIDAD AUTONOMA DE BUCARAMANGA FACULTAD DE INGENIERIA MECATRONICA
FECHA: 15/08/05	
PLANO No.Porta Herramienta	ESC: 1:1 CONVENCIONES



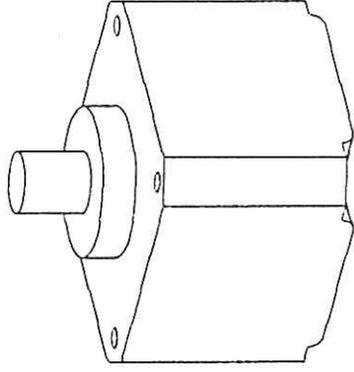
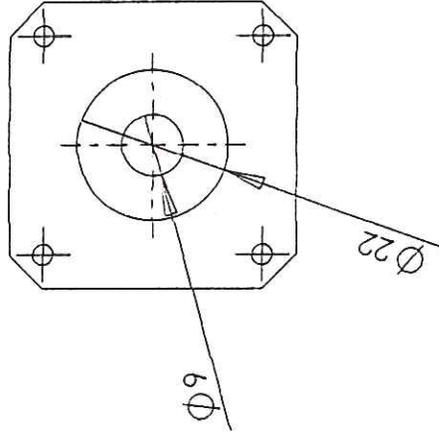
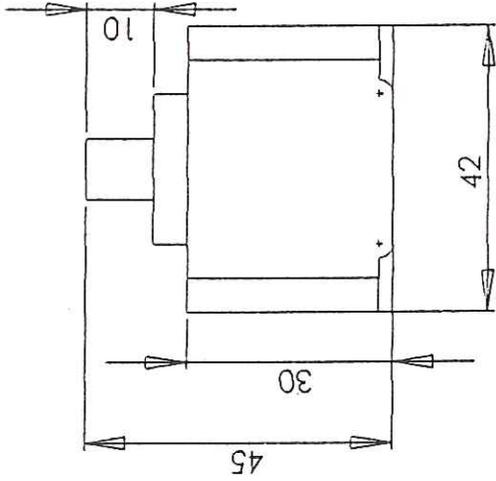
OBSERVACIONES:	DIBUJO:	Miguel A. Romero Jaime A. Garcia	DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR SCARA
	DISEÑO:	Miguel A. Romero Jaime A. Garcia	
NORMA: ISO	APROBO:	Ing. Jhon F. Archila	UNIVERSIDAD AUTONOMA DE BUCARAMANGA FACULTAD DE INGENIERIA MECATRONICA
	FECHA:	15/08/05	
	PLANO No.	Acople Servomotor	
			ESC: 2:1 CONVENCIONES



OBSERVACIONES:	DIBUJO:	Miguel A. Romero Jaime A. Garcia	DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR SCARA	ESC: 2:1
	DISEÑO:	Miguel A. Romero Jaime A. Garcia		
NORMA: ISO	APROBO:	Ing. Jhon F. Archila	UNIVERSIDAD AUTONOMA DE BUCARAMANGA FACULTAD DE INGENIERIA MECATRONICA	CONVENCIONES
	FECHA:	15/08/05		
	PLANO No. Rodamiento			



OBSERVACIONES:	DIBUJO: Miguel A. Romero Jaime A. Garcia	DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR SCARA	
	DISEÑO: Miguel A. Romero Jaime A. Garcia	UNIVERSIDAD AUTONOMA DE BUICARAMANGA FACULTAD DE INGENIERIA MECATRONICA	
NORMA: ISO	APROBO: Ing. Jhon F. Archila	ESC: 1:1 CONVENCIONES	
	FECHA: 15/08/05 PLANO No.Servomotor		



OBSEVACIONES:

DIBUJO: Miguel A. Romero  
Jaime A. Garcia

DISEÑO: Miguel A. Romero  
Jaime A. Garcia

APROBO: Ing. Jhon F. Aichila

FECHA: 15/08/05

PLANO No. Motor Paso a Paso

NORMA:

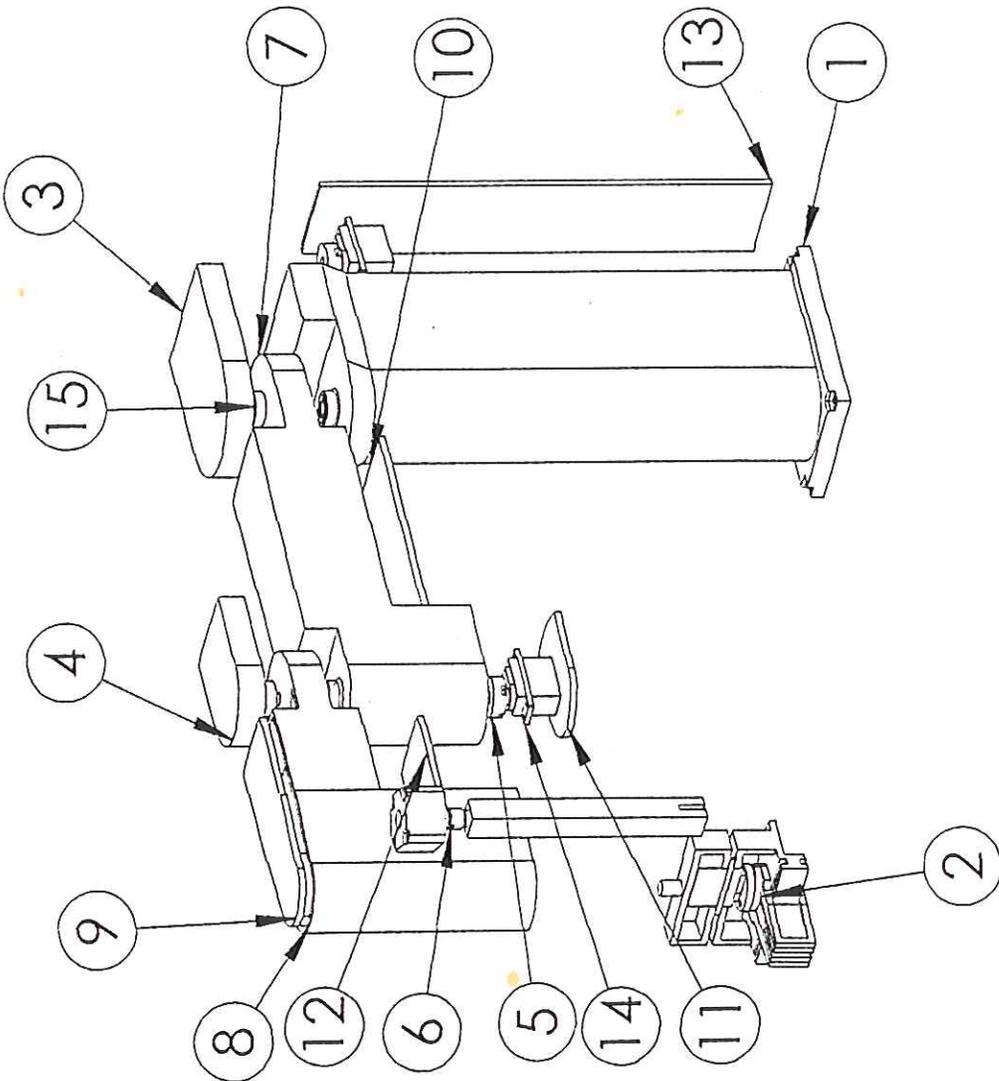
ISO

DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN  
MANIPULADOR SCARA

UNIVERSIDAD AUTONOMA DE  
BUCARAMANGA  
FACULTAD DE INGENIERIA  
MECATRONICA

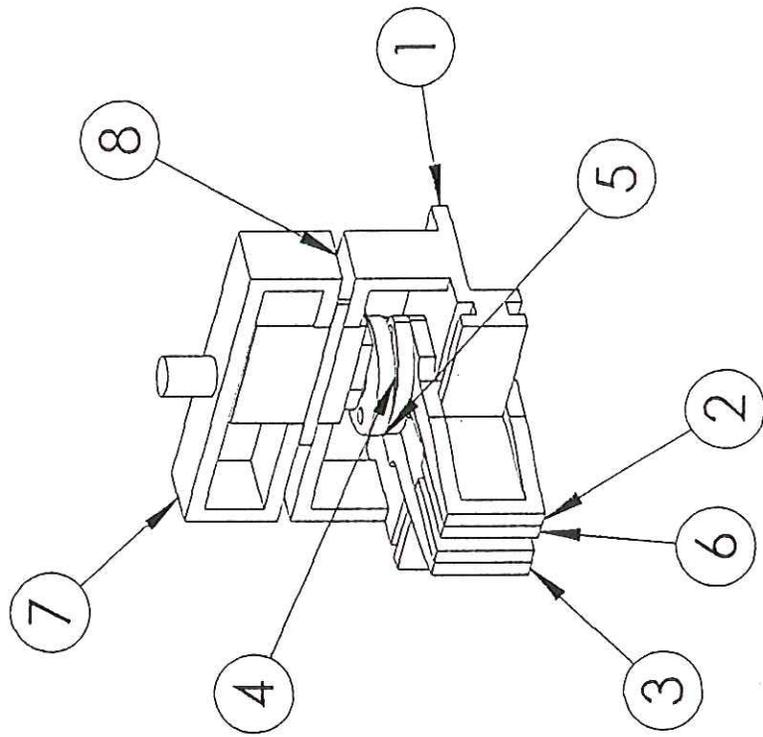
ESC: 1:1

CONVENCIONES



Nº DE ELEMENTO	NÚMERO DE PIEZA	CANTIDAD.
1	Base	1
2	Pinza	1
3	Soporte Antebrazo	1
4	Soporte Brazo	1
5	Acople Servomotor	2
6	Motor Paso a Paso	1
7	Antebrazo	1
8	Brazo	1
9	Tapa Superior Brazo	1
10	Tapa1 Antebrazo	1
11	Tapa2 Antebrazo	1
12	Tapa Inferior Brazo	1
13	Tapa Posterior Base	1
14	Servoapequño	2
15	Rodamiento	4

OBSERVACIONES:  NORMA: ISO	DIBUJO: Miguel A. Romero Jaime A. Garcia	DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR SCARA
	DISEÑO: Miguel A. Romero Jaime A. Garcia	
	APROBO: Ing. Jhon F. Archila	UNIVERSIDAD AUTONOMA DE BUCARAMANGA FACULTAD DE INGENIERIA MECATRONICA
	FECHA: 15/08/05 PLANO No.Scara	



Nº DE ELEMENTO	NÚMERO DE PIEZA	CANTIDAD.
1	Base Pinza	1
2	Pinza1	1
3	Pinza2	1
4	Circulo	1
5	Union	2
6	Caucho	2
7	Soporte Pinza	1
8	Servomotor	1

DIBUJO:	Miguel A. Romero
	Jaime A. Garcia
DISEÑO:	Miguel A. Romero
	Jaime A. Garcia
APROBO:	Ing. Jhon F. Archila
FECHA:	15/08/05
PLANO No. Ensamble Pinza	

DISEÑO, CONSTRUCCION Y PROGRAMACION DE UN MANIPULADOR SCARA	
UNIVERSIDAD AUTONOMA DE BUCARAMANGA FACULTAD DE INGENIERIA MECATRONICA	ESC: 1:2 CONVENCIONES

NORMA: ISO