



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

**ESTUDIO EN LA ELABORACIÓN DE UN ROBOT INDUSTRIAL
MANIPULADOR SENSORIAL Y LA CONSIDERACIÓN DEL USO DEL
MICROCONTROLADOR COMO ALTERNATIVA A SU DISEÑO**

Luis Arcangel Chacón López

Asesorado por la Inga. Ingrid Rodríguez de Loukota

Guatemala, mayo de 2018

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**ESTUDIO EN LA ELABORACIÓN DE UN ROBOT INDUSTRIAL
MANIPULADOR SENSORIAL Y LA CONSIDERACIÓN DEL USO DEL
MICROCONTROLADOR COMO ALTERNATIVA A SU DISEÑO**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

LUIS ARCANGEL CHACÓN LÓPEZ

ASESORADO POR LA INGA. INGRID RODRÍGUEZ DE LOUKOTA

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN ELECTRÓNICA

GUATEMALA, MAYO DE 2018

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Oscar Humberto Galicia Nuñez
VOCAL V	Br. Carlos Enrique Gómez Donis
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Byron Odilio Arrivillaga Méndez
EXAMINADOR	Ing. Julio Rolando Barrios Archila
EXAMINADOR	Ing. Sergio Leonel Gómez Bravo
SECRETARIA	Inga. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

ESTUDIO EN LA ELABORACIÓN DE UN ROBOT INDUSTRIAL MANIPULADOR SENSORIAL Y LA CONSIDERACIÓN DEL USO DEL MICROCONTROLADOR COMO ALTERNATIVA A SU DISEÑO

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 23 de noviembre de 2010.



Luis Arcangel Chacón López

Guatemala 16 de enero de 2014

Ingeniero
Carlos Eduardo Guzmán Salazar
Coordinador del Área de Electrónica
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Estimado Ingeniero Guzmán.

Me permito dar aprobación al trabajo de graduación titular: "**Estudio en la elaboración de un robot industrial manipulador sensorial y la consideración del uso del microcontrolador como alternativa a su diseño**", del señor Luis Arcangel Chacón López, por considerar que cumple con los requisitos establecidos.

Por tanto, el autor de este trabajo de graduación y, yo, como su asesora, nos hacemos responsables por el contenido y conclusiones del mismo.

Sin otro particular, me es grato saludarle.

Atentamente,



Inga. Ingrid Rodríguez de Loukota
Colegiada 5,356
Asesora

Ingrid Rodríguez de Loukota
Ingeniera en Electrónica
colegiado 5356



Ref. EIME 60. 2016.
Guatemala, 13 de JULIO 2016.

Señor Director
Ing. Francisco Javier González López
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Señor Director:

Me permito dar aprobación al trabajo de Graduación titulado:
**ESTUDIO EN LA ELABORACIÓN DE UN ROBOT INDUSTRIAL
MANIPULADOR SENSORIAL Y LA CONSIDERACIÓN DEL USO
DEL MICROCONTROLADOR COMO ALTERNATIVA A SU
DISEÑO,** del estudiante Luis Arcangel Chacón López, que
cumple con los requisitos establecidos para tal fin.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,
ID Y ENSEÑAD A TODOS

Ing. 
Carlos Eduardo Guzmán Salazar
Coordinador Área Electrónica



SFO



REF. EIME 59. 2016.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; **LUIS ARCANGEL CHACÓN LÓPEZ** titulado: **ESTUDIO EN LA ELABORACIÓN DE UN ROBOT INDUSTRIAL MANIPULADOR SENSORIAL Y LA CONSIDERACIÓN DEL USO DEL MICROCONTROLADOR COMO ALTERNATIVA A SU DISEÑO,** procede a la autorización del mismo.

Ing. Francisco Javier González López



GUATEMALA, 4 DE OCTUBRE 2016.

Universidad de San Carlos
de Guatemala



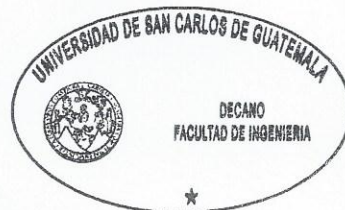
Facultad de Ingeniería
Decanato

DTG. 178 .2018

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **ESTUDIO EN LA ELABORACIÓN DE UN ROBOT INDUSTRIAL MANIPULADOR SENSORIAL Y LA CONSIDERACIÓN DEL USO DEL MICROCONTROLADOR COMO ALTERNATIVA A SU DISEÑO**, presentado por el estudiante universitario: **Luis Arcangel Chacón López**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

Ing. Ángel Roberto Sic García
Decano en Funciones



Guatemala, mayo de 2018

/gdech

ACTO QUE DEDICO A:

Dios Altísimo	Por darme la sabiduría y entendimiento, por bendecirme en mis estudios, en mi familia y en mi vida.
Mis padres	Juan Eduardo Chacón Valdez (q.e.p.d.) y Julia López López, por su amor y apoyo incondicionales.
Mi esposa	María Catalina Escalante de Chacón. Por ser una importante influencia y apoyo en mi vida.
Mis hijos	Juan Javier Moisés y Lourdes Clara Francisca Chacón Escalante, mi hijo que pronto llegará por ser un regalo de Dios y darme felicidad
Mis tíos	Por ser una importante influencia en mi vida.
Mi hermana Pablina	Por su apoyo incondicional y moral.
Mi hermana Alvina	Por motivarme a seguir adelante.
Mi hermana Reyna	Por haberme ayudado en mis estudios de secundaria y por su cariño y amor.
Mi hermana Beatriz	Por el cariño que le tengo.

Mi hermano Juan	Por darme ejemplo de esmero y trabajo.
Mi hermano José	Por el aprecio y respeto que le tengo.
Mis sobrinos	Para ser un ejemplo a seguir.
Mi asesora	Por su amistad y guía profesional, por obsequiarme el tiempo y la ayuda brindada.
Mis cuñados	Por compartir su amistad.
Mis primos	Por compartir su amistad.
Mis amigos	Por compartir su amistad.

AGRADECIMIENTOS A:

Virgen María Santísima	Por su poderosa intercesión y reina de la creación.
Universidad de San Carlos de Guatemala	Por ser fuente de sabiduría y conocimientos, por haberme acogido en esta casa de estudios.
Facultad de Ingeniería	Por mi formación profesional y por la excelencia académica que evidencia.
Mis amigos de la Facultad	Por haber convivido en estos años de logros.
Ing. Ingrid Rodríguez de Loukota	Por ser una importante influencia en mi carrera, por su amistad y apoyo incondicional.
Ing. Guillermo Puente	Por su amistad y conocimientos adquiridos.
Ing. Carlos Guzmán	Por su apoyo y conocimientos adquiridos.
Ing. Francisco López	Por su tiempo y paciencia.
Mis catedráticos	Por haberme enseñado las bases de la ciencia y los conocimientos profesionales invaluable.
Mis padres	Por la vida y los valores que me enseñaron.

Mis hermanos

Por la ayuda incondicional, en mi vida y en mis estudios.

Mi esposa Catalina

Por ser paciente, comprensiva y amorosa.

Vito Moisés Escalante

Por su valioso apoyo.

Marcos y Álvaro Hilario

Por su apoyo en un momento dado.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
GLOSARIO	XI
RESUMEN.....	XXI
OBJETIVOS.....	XXIII
HIPÓTESIS.....	XXV
INTRODUCCIÓN	XXVII
1. FUNDAMENTO DE LA ROBÓTICA	1
1.1. Breve historia del robot.....	1
1.2. Automatización	3
1.2.1. Etapas de funcionamiento de un sistema automatizado	4
1.2.1.1. Periféricos de entrada.....	5
1.2.1.2. Control de datos	5
1.2.1.3. Periféricos de salida	6
1.2.2. Automatización industrial	6
1.2.3. Sistema de control	7
1.2.3.1. Sistema de lazo abierto	7
1.2.3.2. Sistema de lazo cerrado	9
1.3. Anatomía del robot industrial	12
1.3.1. Configuraciones del robot.....	16
1.3.2. Movimientos de articulaciones.....	19
1.3.3. Sistemas de transmisión.....	21
1.4. Espacio de trabajo del robot	28
1.5. Impulsores o fuentes de energía	30

2.	TECNOLOGÍA APLICADA.....	33
2.1.	Actuadores.....	33
2.1.1.	Actuadores eléctricos.....	33
2.1.1.1.	Motores de corriente continua con y sin escobilla.....	34
2.1.1.2.	Motores paso a paso (<i>stepper</i>).....	39
2.1.1.3.	Motores de corriente alterna.....	45
2.1.2.	Actuadores hidráulicos y neumáticos.....	47
2.1.2.1.	Actuadores lineales.....	48
2.1.2.2.	Actuadores rotativos.....	51
2.2.	Válvulas.....	56
2.3.	Sensores.....	60
2.3.1.	Sensor de desplazamiento lineal y rotacional.....	63
2.3.2.	Sensores de proximidad capacitivos.....	65
2.3.3.	Sensores de proximidad inductivos.....	69
2.3.4.	Encóder.....	74
2.3.5.	Sensores ultrasónicos (sensor de posición).....	76
2.3.6.	Sensor de contacto.....	78
2.3.7.	Sensores de velocidad.....	79
2.3.8.	Sensor de fuerza.....	81
2.3.9.	Sensor de aceleración.....	85
2.3.10.	Sensor de inclinación (inclinómetro).....	88
2.4.	Controladores.....	89
3.	FUNCIONES DE TECNOLOGÍA.....	91
3.1.	Análisis de los movimientos.....	91
3.1.1.	Matriz homogénea de Foster, traslación y rotación.....	93
3.1.2.	Asignación del sistema de referencia.....	95

3.1.3.	Cinemática directa e inversa.....	101
3.1.3.1.	Parámetros de Denavit y Hartenberg	103
3.1.3.2.	Matriz de transformación homogénea .	106
3.1.3.3.	Solución del problema de cinemática inversa (solución de Pieper)	109
3.1.3.4.	Matriz Jacobiana y matriz Jacobiana inversa	111
3.1.3.5.	Fuerzas estáticas.....	116
3.2.	Efectores	118
3.2.1.	Pinzas	118
3.2.2.	Herramientas	122
3.3.	Microcontrolador.....	123
3.3.1.	Arquitectura del microcontrolador	126
3.3.2.	Integración del microcontrolador.....	145
4.	DISEÑO E IMPLEMENTEACIÓN CON EL MICROCONTROLADOR..	149
4.1.	Consideraciones de diseño	149
4.2.	Lenguaje MikroBasic	155
4.2.1.	Estructura del programa	157
4.2.2.	Variables, constantes y etiquetas	159
4.2.3.	Funciones y procedimientos	161
4.2.4.	Datos simples, arreglos, cadenas, punteros y estructuras.....	162
4.2.5.	Operadores.....	165
4.2.6.	Declaraciones	168
4.2.6.1.	Declaración de asignación.....	169
4.2.6.2.	Declaraciones condicionales <i>If</i> y <i>Select case</i>	169

4.2.6.3.	Declaraciones de iteración <i>for</i> , <i>while</i> y <i>do</i>	170
4.2.6.4.	Declaraciones de salto	172
4.2.6.4.1.	Declaración <i>break</i>	172
4.2.6.4.2.	Declaración <i>continue</i> ...	173
4.2.6.4.3.	Declaración <i>exit</i>	174
4.2.6.4.4.	Declaración <i>goto</i>	174
4.2.6.4.5.	Declaración <i>gosub</i>	175
4.2.6.5.	Declaración <i>asm</i>	176
4.3.	Programación con MikroBasic.....	176
4.3.1.	Programación de manejo de motores	177
4.3.2.	Programación de lectura de sensores.....	187
4.3.3.	Comunicación entre dos microcontroladores	193
4.4.	Aplicación del microcontrolador al robot.....	195
4.4.1.	Cinemática directa.....	197
4.4.2.	Cinemática inversa.....	208
4.4.3.	Aplicación del microcontrolador.....	221
CONCLUSIONES.....		235
RECOMENDACIONES		237
BIBLIOGRAFÍA.....		239

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Diagrama de bloques de un sistema básico automatizado	5
2.	Sistema de control de lazo abierto	8
3.	Relación de un sistema de control lazo abierto	9
4.	Sistema de control de lazo cerrado	10
5.	Respuesta transitoria de un sistema de lazo cerrado	11
6.	Locomoción lenta y rápida de un robot bípedo	13
7.	Prototipo del primer robot bípedo	14
8.	Aplicación del robot ASIMO	14
9.	Robot zoomorfo.....	15
10.	Configuración polar o esférica.....	17
11.	Configuración cilíndrica.....	17
12.	Configuración cartesiana.....	18
13.	Configuración articulado (imitación del brazo humano).....	19
14.	Diferentes tipos de articulaciones	20
15.	Sistema de transmisión directa	22
16.	Sistema de transmisión indirecta entre etapas.....	24
17.	Mecanismo y ensamblaje del <i>harmonic drive</i>	26
18.	Operación del <i>harmonic drive</i>	27
19.	Mecanismo de movimiento para la muñeca del robot industrial	27
20.	Volúmenes de trajo para distintas configuraciones	28
21.	Circuito equivalente eléctrico del motor de imanes permanentes	35
22.	Gráfica de relación entre el par de salida y la velocidad angular al voltaje aplicado	35

23.	Circuitos equivalentes para las diferentes configuraciones del estator y rotor.....	37
24.	Circuito de control de velocidad de giro	38
25.	Circuitos de motores paso a paso bipolares y unipolares.....	41
26.	Corte transversal de un cilindro de simple acción.....	49
27.	Repulsión y atracción del pistón, debido al fluido o aires comprimido	50
28.	Corte transversal de un cilindro de doble acción	51
29.	Actuadores rotativos para movimiento hasta 90°, 180° y 360°	52
30.	a) Actuador piñón y cremallera, b) actuador piñón y cremallera aspecto físico.....	53
31.	Imagen representativa del yugo escocés.....	54
32.	Imagen física de yugo escocés de doble efecto	55
33.	Actuador de veleta con movimiento en ambos sentidos	56
34.	Válvula de carrete de dos posiciones, a) suministro de aires y b) desfogue de aire	57
35.	Símbolos para distintas formas de activar una válvula	59
36.	Construcción física de la válvula.....	60
37.	Símbolo gráfico de la válvula y modo de accionamiento	60
38.	a) Relación de voltaje a resistencia, b) circuitos equivalentes del potenciómetro	65
39.	a) El condensador dividido por el electrodo de accionamiento, b) el condensador como sensor capacitivo y el efector del elemento accionamiento.....	67
40.	El condensador como sensor de elementos no metales.....	68
41.	Diagrama de bloques del sensor	68
42.	Capacitancia del sensor en función de la distancia	69
43.	Salida del oscilador en presencia del objeto	69

44.	a) El sensor LDVT, b) diagrama de bloques del sensor con voltaje de salida dc y su gráfica de respuesta	71
45.	Diagrama de bloques del sensor.....	73
46.	Forma de operación del sensor.....	74
47.	El encóder básico.....	75
48.	Diferentes tipos de encóder	76
49.	Señal de cuadratura en el encóder incremental.....	76
50.	a) Sensor SRF8 y módulo de conexión, b) rango de operación del sensor RKU8.....	77
51.	Sensor de contacto comercialmente disponible	79
52.	a) Tacogenerador con n devanado y la forma de onda de voltaje de salida, b) tacogenerador dc comercialmente disponible.....	80
53.	a) Gráfica fuerza vs. resistencia, b) gráfica fuerza vs. Conductancia...	82
54.	a) Circuito básico divisor de voltaje y gráfica de respuesta para diferentes valores de R_M , b) circuito convertidor de corriente a voltaje y gráfica de respuesta para diferentes valores de R_G	84
55.	a) Sistema de placas montadas en el chip de silicio, b) movimiento de la superficie conductora "S"	86
56.	a) Efecto de la fuerza sobre el material piezoeléctrico, b) acelerómetro piezoeléctrico	87
57.	a) Posición del origen del sistema O para un objeto respecto al sistema de referencia M, b) orientación del sistema O respecto al M..	92
58.	Rotación alrededor del eje x al z	95
59.	Movimientos con respecto al mismo sistema de referencia	96
60.	Robot con sistema de referencia fijo	98
61.	Representación de la asignación de un sistema de referencia móvil...	99
62.	Rotaciones simples de guiñado, cabeceo y alabeo correspondiente a un sistema de referencia fijo	101
63.	Asignación de los parámetros de las articulaciones.....	102

64.	Parámetros de Denavit y Hartenberg a) para articulación i rotacional, b) para articulación i prismáticas	103
65.	Asignación del sistema de referencia para el eslabón i	106
66.	Transformaciones básicas para pasar del sistema i-1 al sistema i	107
67.	Vectores de posición para la muñeca y el extremo del robot.....	110
68.	Par y fuerza neta en el extremo del robot debido a pares y fuerzas en las articulaciones	117
69.	Pinzas actuadas por articulación	119
70.	a) Actuación de pinza por engranaje y bastidor, b) de pinza por leva, c) de pinza por tornillo y d) de pinza por cable y polea	121
71.	Dispositivos y herramientas sujetos a efectores finales.....	123
72.	Diseño de arquitectura para un microcontrolador, arquitectura von-Neumann (arriba) y arquitectura Harvard (abajo)	127
73.	Diagrama de la memoria de programa para el 16F877	129
74.	Diagrama de conexión del PIC16F877	143
75.	Arquitectura del microcontrolador PIC16F877	144
76.	Reductor de velocidad tonillo sin fin y rueda dentada.....	147
77.	Sistema actuador, reductor y medición para un robot industrial comercial	152
78.	a) Programación guiada pasiva directa y b) programación guiada activa	155
79.	Ventana principal del compilador MikroBasic	157
80.	Puente en H para control de giro de un motor DC	180
81.	Diagrama esquemático del cambio de giro de un motor AC monofásico	182
82.	Simbología para los tipos de relés	183
83.	Diagrama esquemático del cambio de giro de un AC motor trifásico..	184
84.	Sistema neumático y sus componentes (análoga al sistema hidráulico)	186

85.	a) Robot KUKA, determinación de articulaciones (numeración) y b) especificaciones física (dimensiones en mm y ángulos en grados) ...	198
86.	Ejecución de los incisos uno y tres, el inciso dos se representa en la figura 85	200
87.	Asignación del eje z en la articulación i+1 y dirección del eje z en sentido contrario a la asignación positiva de rotación	201
88.	Ejecución de los incisos cinco y seis, asignación del origen del sistema i.....	202
89.	Determinación de los parámetros físicos, forma y tamaño de la articulación, así como tamaño del eslabón	203
90.	a) Especificaciones de eslabones al robot, de los parámetros D-H, b) Asignación del sistema de referencia y ubicación de 0T_5	209
91.	Relaciones geométricas del robot para determinar q_2 y q_3	211
92.	Descripción de movimiento de embalaje realizado por el robot	216
93.	Aplicación del microcontrolador al gobierno del robot.....	222
94.	Ejecución de los movimientos 1, 2 y 3 de embalaje piezas de metal caliente	224

TABLAS

I.	Eventos en la historia del robot	2
II.	Descripción de los sistemas de transmisión.....	22
III.	Secuencia de pulsos para un motor p-p bipolar	41
IV.	Secuencia de pulsos para un motor p-p unipolar en modo normal	42
V.	Secuencia de pulsos para un motor p-p unipolar en modo <i>wave drive</i>	43
VI.	Secuencia de pulsos para un motor p-p unipolar en modo medio paso	44

VII.	Gama de microcontroladores y principales características.....	125
VIII.	Diagrama y registros de la memoria RAM	131
IX.	Configuración de puertos E/S, como analógicos o digitales	140
X.	Tipos de datos simples en MikroBasic.....	163
XI.	Precedencia de los operadores	166
XII.	Operadores aritméticos.....	166
XIII.	Operadores relacionales	167
XIV.	Operadores lógicos.....	168
XV.	Determinación de los parámetros de Denavith-Hartenberg del robot de la figura 89	204
XVI.	Valores de articulaciones para los pasos 1, 2, 3 y 5	221
XVII.	Valores de ángulos en las que se desplazó el sistema de referencia y el valor del encóder de 8 bits	223

GLOSARIO

Actuadores	Son dispositivos que transforman la energía eléctrica, hidráulica o neumática en movimiento mecánico, proporcional a la fuerza de movimiento en el robot.
Actuadores eléctricos	Dispositivos (motores) cuya energía proviene de la energía eléctrica para generar movimiento.
Actuadores hidráulicos	Dispositivos (motores) cuya energía mecánica proviene de fluidos a presión, como por ejemplo, aceites, lubricantes e incluso el agua para generar movimiento.
Actuadores neumáticos	Dispositivos (motores) cuya energía mecánica proviene de aire a presión o elementos gaseosos a presión.
Algoritmos	Conjunto de reglas o procesos para la solución de un problema con un número finito de pasos.
Analógico	Representación de una variable, información o señal que varía en forma continua en el tiempo.
Anatomía del robot	Construcción física del cuerpo, brazos y muñeca de la máquina.

Articulaciones	Unión entre dos piezas rígidas que permite cierto movimiento entre ella.
Autómata	Del griego automatos que significa espontáneo o con movimiento propio, mecanismos que imitan los movimientos y tareas, similares a un ser humano o ser animado.
Automático	Movimiento o actividad propia de un mecanismo autómata.
Automatización	Mecanismos y elementos electromecánico controlados por sistemas de cómputos y electrónicos, sustituyendo al operado humano.
Brazo del robot	Compuesta por los actuadores, articulaciones y la muñeca, soportada por la base.
Chip	Pieza pequeña de material semiconductor de silicio sobre la cual se fabrican circuito electrónico compuesto de transistores, diodos y resistencias.
Cibernética	Estudio comparativo de procesos orgánicos y los realizados por máquinas, con el fin de comprender sus semejanzas y diferencia, y lograr que estos imiten el comportamiento orgánico.
Cinemática	Término relativo al accionamiento del manipulador de la unión de la física entre los mandos del

operador y el elemento terminal. Parte de la mecánica que estudia los movimientos en sus condiciones de espacio y tiempo.

Codificador Dispositivo que transforma un lenguaje con base a un código.

Controlador Dispositivo o software que compara una señal de salida y una orden de entrada que produce una señal de error que controla un periférico o actuador.

Coordenadas Geometría de dirección de cada una de las líneas o plano de referencia que sirven para determinar la posición de un punto. Estas pueden ser angulares, polares y rectangulares.

Dinámica del robot Control de los actuadores en las articulaciones y estas depende de las fuerzas que actúan sobre el manipulador y la inercia reflejada.

Diodo Dispositivo de dos capas de material semiconductor tipo “n” y tipo “p” que se comporta como un interruptor en condición de corte y saturación, sólo conduce corriente en una dirección.

Display Dispositivo electrónico por medio del cual se ofrece una representación visual de la información.

Efecto final	Nombre técnico que se le da a la mano del manipulador o brazo robótico. Componente final especial que permite al robot de uso general realizar una aplicación en particular.
Eje	Dos o más líneas que se utilizan para determinar un punto en el espacio. Cada eje define un grado de libertad.
Electroimán	Dispositivo que adquiere propiedades magnéticas cuando se le hace circular una corriente eléctrica.
Elemento	Cada uno de los componentes de la estructura del manipulador.
Engranaje	Sistema mecánico que transmite el movimiento de rotación desde un eje hasta otro, mediante el contacto sucesivo de pequeñas levas denominadas dientes. Las ruedas dentadas pueden ser cilíndricas o helicoidales.
<i>EEPROM</i>	Erase Programmable Read Only Memory, memoria de solo lectura programable y borrrable.
Fotocélula	Dispositivo electrónico detector de luz que convierte la luz en una señal eléctrica.
Fuente de poder	Unidad que suministra energía eléctrica a otros componentes.

GDL	Grados de libertad, movimientos del robot (contenida en cada articulación) que pueda tener en las n dimensiones.
Giro	Movimiento básico del manipulador.
Hardware	Parte física constituida de componentes electrónicos y componentes físicos que en su totalidad permiten la ejecución del software.
Inteligencia artificial	Capacidad de un artefacto de realizar los mismos tipos de funciones que caracterizan al pensamiento humano.
Interfaz	Conexión física o conector que, junto con un dispositivo o circuito electrónico hace posible la conexión y el entendimiento entre dos componentes de <i>hardware</i> .
Lenguaje de computación	Programa mediante el cual se puede especificar una serie de instrucciones, que el computador efectúe una tarea específica y autónoma.
Librerías del pic	Conjunto de archivos que agregan funciones dedicadas al microcontrolador.
Programación	nuevas capacidades y funciones a un lenguaje de computación específico.

Manipulador	Cualquier componente mecánico capaz de realizar los movimientos del ser humano. Esto se refiere a los elementos mecánicos de un robot para la adecuada operación o posicionamiento.
Microcontrolador	Circuito integrado programable que contiene todos los componentes de un computador, se emplea para realizar una tarea determinada para la cual ha sido programado. Dispone de procesador, memoria para el programa y los datos, líneas de entrada y salida de datos y suele estar asociado a múltiples recursos auxiliares.
Microchips (chip)	Conjunto de circuitos empaquetados para computador (circuito integrado) fabricado de silicón a muy pequeña escala.
Motor	Dispositivo que produce movimientos a expensas de una fuente de energía.
Motor paso a paso	Motor eléctrico que se controla por medio de una secuencia de pulsos discretos, con cada pulso el motor gira cierto ángulo de rotación fijo (por pasos). A diferencia de los motores eléctricos D.C. y A.C. que poseen un rotor de imán permanente y un estator bobinado con su principio de funcionamiento de atracción y repulsión.

PIC	<i>Programmable Interrupt Controller</i> , controlador programable de interrupciones.
Pinzas	Efector final que utiliza dedos mecánicos impulsados por un mecanismo para manipular o agarrar un objeto.
Potenciómetro	Es un dispositivo eléctrico que proporciona una tensión de salida proporcional a la posición del cursor y a la tensión de entrada.
Procedimiento	Secuencia de instrucciones o pasos para resolver un problema determinado.
Programa de control	Secuencia de instrucciones que residen en la computadora de control y que determinan el comportamiento del sistema de robótica.
Protocolo	Conjunto de reglas o normas que establece la correcta comunicación. Puede definirse también como las reglas que dominan las sintaxis, semántica y sincronización de la comunicación.
Puerto	Interfaz de entrada/salida contenida en un computador, microcontrolador, microprocesador, etc. Permite conectar otros dispositivos de un mismo protocolo.

Relé	Se designa a un dispositivo electromagnético que al estimularlo con una pequeña corriente en sus terminales de entrada abre o cierra un circuito, en el cual se disipa una potencia mayor que el circuito estimulador.
Resistividad	Oposición que ofrecen algunos materiales al paso de la corriente (flujos de electrones).
Resolver	Transductor, mide el ángulo de rotación.
Robot	Manipulador multifuncional y reprogramable, diseñado para mover materiales, piezas, herramientas o dispositivos especiales, mediante movimientos programados y variables que permiten llevar a cabo diversas tareas.
Robot Automata (RA)	Robot de sistema completo sin la necesidad de ser guiado o controlado por un operador.
Robot Industrial (RI)	Articulaciones mecánicas y electrónicas destinados a realizar de forma automática y sin necesidad de intervención humana, determinados procesos de fabricación o manipulación. Los RI son pre programados para realizar una tarea específica y no poseen la capacidad para reconfigurarse automáticamente.

Robótica	Rama de la ciencia que se ocupa el estudio, desarrollo y las aplicaciones del robot.
Sensor	Transductor que detecta las variaciones físicas externas y las transforma en una forma de energía eléctrica legible, para los dispositivos electrónicos.
sensor piezoeléctrico	Dispositivo que utiliza el efecto piezoeléctrico para medir presión, aceleración, tensión o fuerza.
Sistema	Conjunto de elementos organizados que se interrelacionan e interactúan para una función u objetivo.
Sistemas guiados	Es cuando un operador conduce al robot para realizar los movimientos requeridos.
Transductor	Dispositivo que transforma una magnitud física o energía de una clase a otra de diferente clase. La magnitud del valor de salida, es directamente proporcional a la magnitud del valor de entrada.
Transistor	Dispositivo semiconductor de tres capas tipo “pnp” o “nnp”. Contracción de las palabras inglesas <i>transfer resistor</i> . Es decir, de resistencia de transferencia.

Zoomorfo (robot)

Robot con forma animal, o que usa ciertas características inspiradas en algún animal, se hace una excepción con los androides.

RESUMEN

El primer capítulo incluye una reseña histórica que especifican a los sistemas automatizados (eléctricos, hidráulicos y neumáticos) y sus diferentes etapas; además, se clasifican y caracterizan los diferentes sistemas de control. Los robots se clasifican por sus características; también se profundiza en las configuraciones para el robot industrial. Se describen los diferentes sistemas de transmisión, sus ecuaciones; asimismo se mencionan las articulaciones y las distintas fuentes de impulsión.

En el segundo capítulo, se analizan las características de los actuadores, su circuito equivalente, gráficas de respuesta y forma de operación; se profundiza en los actuadores eléctricos (motores DC con o sin escobillas, AC y stepper). En los actuadores hidráulicos y neumáticos, se presentan ecuaciones, gráficas de construcción y operación, se especifican los tipos de válvulas y su forma de operación. Se presentan conceptos en régimen permanente y transitorio para los sensores de contacto, proximidad, fuerza, velocidad, aceleración, inclinación y desplazamiento, se muestran circuitos, gráficas y diagramas de operación.

En el tercer capítulo, se presentan conceptos, ecuaciones y gráficas para el análisis del movimiento del robot, se destacan las matriz homogénea de Foster, matriz homogénea, jacobiana, parámetros Denavit y Hartenberg. Se analiza la traslación y rotación, cinemática directa e inversa y se mencionan las fuerzas estáticas, se presentan las clase de pinzas y herramientas. Se hace un estudio detallado y profundo de la arquitectura del microcontrolador.

El cuarto capítulo indica las consideraciones de diseño, se presenta MikroBasic como lenguaje de programación; además, realiza un aprendizaje previo a la programación, también se muestra programas de aplicación. Se resuelve la cinemática directa e inversa para un robot KUKA, se aplica el microcontrolador para un ejemplo que se resolvió en el programa de instrucciones escrita en MikroBasic.

OBJETIVOS

General

Realizar una investigación y un trabajo donde se especifica el proceso de construcción y elaboración de un robot industrial enfatizando en los sensores, efectores y demás componentes; proponer una alternativa en el diseño del robot con fines de estudio para profesionales y estudiantes, utilizando el microcontrolador.

Específicos

1. Tener un concepto general del robot, realizando un estudio de la anatomía, sistemas de transmisión de potencia mecánicas, sistemas de impulsión y un enfoque a la automatización.
2. Estudiar los componentes para el procesamiento de datos que el robot interpretará para su funcionamiento.
3. Analizar la complejidad del movimiento y las tareas asignada del robot, empezando, con ello, el estudio de sistemas de sensores y efectores quienes realizan la manipulación y el trabajo llegando hasta el microcontrolador, su arquitectura e implementación.
4. Considerar criterios de diseño, aplicar MikroBasic para programar el microcontrolador y del microcontrolador al robot.

5. Proporcionar un enfoque eléctrico y electrónico, adquiriendo un propósito pedagógico y una fácil comprensión para quienes quisiesen tener un conocimiento en el diseño de un robot, se hará mención de los sistemas hidráulicos y neumáticos, para lograr un enfoque hacia esta tecnología.

6. Presentar la hipótesis para el diseño y empleo del microcontrolador en robótica industrial.

Hipótesis

Un robot industrial consta de un sistema de impulsión o fuente de energía, actuadores que generan el movimiento, sensores como sentidos y controladores que gobiernan al robot. También es necesaria una estructura liviana y resistente que sea capaz de soportar fuerzas de tensión y compresión. La fuerza generada depende de la estructura y los actuadores, por ello es necesario compensar el torque para la precisión. El microcontrolador es el controlador del robot que efectúa y censa cada componente y gobierna la funcionalidad del robot.

El primer paso para el desarrollo de la robótica es el enriquecimiento teórico y práctico.

Con el microcontrolador se evitaría utilizar un sistema de cómputo, con ello se logra mayor flexibilidad a un menor costo. Se toma en cuenta que el microcontrolador funcionaría de forma modular, para controlar cada parte del robot y en conjunto todo el robot.

Hipótesis nula

La hipótesis nula será escrita de esta forma.

Hipótesis alternativa

Al igual que la hipótesis alternativa.

INTRODUCCIÓN

En Guatemala, no existe desarrollo en robótica industrial, pero al realizar una investigación, diseño y construcción para este robot, se requiere de aportaciones fuera de Guatemala. Se tiene muy poca documentación relacionada con el tema, y la mayor parte del conocimiento que se utiliza para el diseño procede de investigaciones que han realizado personas o entidades de otros países, como España, Estados Unidos, Rusia y Alemania, que es el país con tecnología más avanzado en robótica industrial.

Dadas estas limitaciones, el propósito de este trabajo es realizar un plan de investigación en la arquitectura y el desarrollo para el diseño y construcción del robot. Aplicar la teoría existente para el movimiento y aplicar el microcontrolador, para controlar al robot utilizando un lenguaje de alto nivel, que facilite el desarrollo del programa de instrucciones.

Para el análisis y planteamiento se utilizarán conceptos aprendidos en los cursos libres en MikroBasic, robótica, Electrónica 5, Programación básica e investigaciones de fuentes de internet y de libros existentes en universidades. El contenido del trabajo comprendido en los capítulos primero, segundo y tercero, desarrollan la teoría necesaria para conocer y comprender los dispositivos que generan movimiento, los dispositivos para interactuar en su entorno y conceptos para el movimiento. La parte práctica está comprendida en el capítulo cuarto, que desarrollará paso a paso el *software* que se utiliza para controlar al robot.

1. FUNDAMENTO DE LA ROBÓTICA

1.1. Breve historia del robot

La historia del robot es el punto de partida de interés en esta investigación.

“La palabra robot se utilizó por primera vez en 1920 en una obra llamada *Los Robots Universales* de Rossum, escrita por el dramaturgo checo Karel Capek. Su trama trataba sobre un hombre que fabricó un robot y luego este último mata al hombre. La palabra checa 'Robota' significa servidumbre o trabajado forzado, y cuando se tradujo al inglés se convirtió en el término robot. Luego, Isaac Asimov comenzó en 1939 a contribuir con varias relaciones referidas a robots y a él se le atribuye el acuñamiento del término Robótica y con él surgen las denominadas *Tres Leyes de Robótica* que son las siguientes:

- Un robot no puede actuar contra un ser humano o mediante la inacción, que un ser humano sufra daños.
- Un robot debe de obedecer las órdenes dadas por los seres humanos, salvo que estén en conflictos con la primera ley.
- Un robot debe proteger su propia existencia, a no ser que esté en conflicto con las dos primeras leyes”¹.

George Devol patentó en 1948 manipuladores y en conjunto con la compañía *Unimation (Universal Automation)*; en 1961 fabricó el primer robot.

¹ BELU. *ROBOTICA* <http://robotica.blogspot.com/2007/10/historia-de-la-robotica.html>. Consulta: 15 marzo de 2011.

Tabla I. **Eventos en la historia del robot**

FECHA	DESARROLLO
Siglo XVIII	A mediados del J. de Vaucanson construyó varias muñecas mecánicas de tamaño humano que ejecutaban piezas de música.
1801	J. Jacquard inventó su telar, que era una máquina programable para la urdimbre.
1805	H. Maillardet construyó una muñeca mecánica capaz de hacer dibujos.
1946	El inventor americano G.C Devol desarrolló un dispositivo controlador que podía registrar señales eléctricas, por medios magnéticos y reproducirlas para accionar una máquina mecánica. La patente estadounidense se emitió en 1952.
1951	Trabajo de desarrollo con teleoperadores (manipuladores de control remoto) para manejar materiales radiactivos. Patente de Estados Unidos emitidas para Goertz (1954) y Bergsland (1958).
1952	Una máquina prototipo de control numérico fue objetivo de demostración en el Instituto Tecnológico de Massachusetts después de varios años de desarrollo. Un lenguaje de programación de piezas denominado APT (Automatically Programmed Tooling) se desarrolló posteriormente y se publicó en 1961.
1954	El inventor británico C. W. Kenward solicitó su patente para diseño de robot. Patente británica emitida en 1957.
1954	G.C. Devol desarrolla diseños para Transferencia de artículos programada. Patente emitida en Estados Unidos para el diseño en 1961.
1959	Planet Corporation introdujo el primer robot comercial. Estaba controlado por interruptores de fin de carrera.
1960	Se introdujo el primer robot ‘Unimate’, basada en la transferencia de artic. Programada de Devol. Utilizan los principios de control numérico para el control de manipulador y era un robot de transmisión hidráulica.
1961	Un robot Unimate se instaló en la Ford Motors Company para atender una máquina de fundición de troquel.
1966	Trallfa, una firma noruega, construyó e instaló un robot de pintura por pulverización.
1968	Un robot móvil llamado ‘Shakey’ se desarrolló en SRI (standford Research Institute), estaba provisto de una diversidad de sensores, una cámara de visión y sensores táctiles. Este robot podía desplazarse por el suelo.
1971	El ‘Standford Arm”, un pequeño brazo de robot de accionamiento eléctrico, se desarrolló en la Standford University.
1973	Se desarrolló en SRI el primer lenguaje de programación de robots del tipo de computadora para la investigación con la denominación WAVE. Fue seguido por el lenguaje AL en 1974. Los dos lenguajes se desarrollaron posteriormente en el lenguaje VAL comercial para Unimation por Víctor Scheinman y Bruce Simano.
1974	ASEA introdujo el robot Irb6 de accionamiento completamente eléctrico.
1974	Kawasaki, bajo licencia de Unimation, instaló un robot para soldadura por arco para estructuras de motocicletas.
1974	Cincinnati Milacron introdujo el robot T3 con control por computadora.
1975	El robot ‘Sigma” de Olivetti se utilizó en operaciones de montaje, una de las primitivas aplicaciones de la robótica al montaje.
1976	Un dispositivo de Remote Center Compliance (RCC) para la inserción de piezas en la línea de montaje se desarrolló en los laboratorios Charles Stark Draper Labs en Estados Unidos.

Continuación de la tabla I.

1978	El robot T3 de Cincinnati Milacron se adaptó y programó para realizar operaciones de taladro y circulación de materiales en componentes de aviones, bajo el patrocinio de Air Force ICAM (Integrated Computer- Aided Manufacturing).
1978	Se introdujo el robot PUMA (Programmable Universal Machine for Assably) para tareas de montaje por Unimation, basándose en diseños obtenidos en un estudio de la General Motors.
1979	Desarrollo del robot tipo SCARA (Selective Compliance Arm for Robotic Assably) en la Universidad de Yamanashi en Japón para montaje. Varios robots SCARA comerciales se introdujeron hacia 1981.
1980	Un sistema robótico de captación de recipientes fue objeto de demostración en la Universidad de Rhode Island. Con el empleo de visión de máquina el sistema era capaz de captar piezas en orientaciones aleatorias y posiciones fuera de un recipiente.
1981	Se desarrolló en la Universidad de Carnegie- Mellon un robot de impulsión directa. Utilizaba motores eléctricos situados en las articulaciones del manipulador sin las transmisiones mecánicas habituales empleadas en la mayoría de los robots.
1982	IBM introdujo el robot RS-1 para montaje, basado en varios años de desarrollo interno. Se trata de un robot de estructura de caja que utiliza un brazo constituido por tres dispositivos de deslizamiento ortogonales. El lenguaje del robot AML, desarrollado por IBM, se introdujo también para programar el robot SR-1.
1983	Informe emitido por la investigación en Westinghouse Corp. bajo el patrocinio de National Science Foundation sobre un sistema de montaje programable adaptable (APAS), un proyecto piloto para una línea de montaje automatizada flexible con el empleo de robots.
1984	Robots 8. La operación típica de estos sistemas permitía que se desarrollarán programas de robots utilizando gráficos interactivos en una computadora personal y luego se cargaban en el robot.

Fuente: MOLINA MARICORENA, José L. *Historia de la robótica*.

<http://www.profesormolina.com.ar/tecnologia/robotica/historia.htm>. Consulta: diciembre de 2010.

1.2. Automatización

Es una tecnología relacionada con la tecnología de la robótica. Emplea sistemas mecánicos, electrónicos, eléctricos, neumáticos e hidráulicos, así como sistemas de cómputo para el gobierno y operación de un sistema. En la actualidad, se utilizan los sistemas de microcontrolador como alternativa para el gobierno y operación de los sistemas de automatización. La finalidad de la

automatización es la eliminación de la intervención humana para el funcionamiento de un sistema. Entre los tipos de sistemas están los siguientes:

- Sistemas automatizados eléctricos: funcionan mediante la corriente eléctrica. En este sistema están los más precisos, como la fabricación de dispositivos semiconductores, un dispositivo reproductor de video, los televisores con programación de canales, entre otros.
- Sistemas automatizados hidráulicos: funcionan por transferencia de fluidos, con lo cual logran mayor fuerza, pero pierden precisión, como las grúas, entre otros.
- Sistemas automatizados neumáticos: son los que funcionan por medio de aire comprimido. Entre ellos está un sistema de lavado de automóviles, entre otros.

A pequeña escala se puede mencionar la automatización de procesos que intervienen diferentes tipos de máquina y la automatización programable, que es el grado más elevado de automatización en la que intervienen equipos informáticos y dispositivos robóticos.

1.2.1. Etapas de funcionamiento de un sistema automatizado

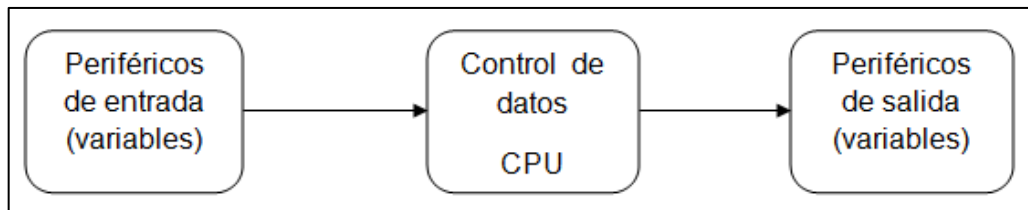
Para el funcionamiento de un sistema automatizado se tiene tres fases:

- Periféricos de entrada.
- Control de datos.
- Periféricos de salida y realización de tareas programadas.

1.2.1.1. Periféricos de entrada

Como se puede apreciar en la figura 1, en la entrada de datos u órdenes; es la información que necesita la unidad de control para ejercer el mando en la realización de las tareas. Estos valores pueden ser información de luz, señales eléctricas, neumáticos y magnéticas. Esta información se obtiene mediante los periféricos de entrada que de los sensores o transductores que transforman una forma de energía legible para el CPU. Las órdenes son una forma de señal que pueden provenir de un sistema de cómputo o comunicación que le dan la orden al CPU, para el procesamiento de los valores y realizar las tareas.

Figura 1. **Diagrama de bloques de un sistema básico automatizado**



Fuente: elaboración propia.

1.2.1.2. Control de datos

El cerebro en la operación de un sistema automatizado es el control y operación de datos (CPU), por medio de los periféricos de salida indica a los actuadores la operación necesaria para llevar a cabo una tarea. Entre los controles se encuentran los manuales, automáticos, programables, etc. Los controles manuales son necesarios cuando varían las condiciones de trabajo. Los controladores automáticos funcionan de manera continua sin importar las variaciones que se producen en un entorno de trabajo y los controles

programables funcionan mediante algoritmo de programas y varían su funcionamiento, según las variaciones en su entorno de trabajo. Se pueden mencionar los microprocesadores, microcontroladores y los controladores lógicos programables (PLC).

1.2.1.3. Periféricos de salida

Se refiere a los actuadores como periféricos de salida. Estas reciben de la unidad de control la información necesaria para ejecutar una tarea concreta. Es importante mencionar que en esta etapa se maneja la potencia necesaria para realizar un proceso y las transformaciones de energía. Entre los actuadores están los actuadores eléctricos, hidráulicos y neumáticos.

1.2.2. Automatización industrial

Es el uso de sistemas o elementos computarizados y electromecánicos para el control y operación de maquinarias y procesos industriales. Abarca no solo el campo de los sistemas de control sino la instrumentación industrial y de la robótica industrial que es el tema central de este trabajo. Es decir, que se abarca una amplia rama de la ingeniería para la operación y mecanización de los procesos. Entre las ventajas están las siguientes:

- El reemplazo del operador humano en tareas repetitivas.
- El reemplazo del ser humano en tareas de levantar y de traslado de cargas pesadas.
- En el manejo de alta precisión.
- Incremento en la producción al mantener una línea de producción automatizada.
- Mayor eficiencia.

Las herramientas utilizadas en la automatización industrial que alude a los diseños asistidos por computadora (CAD), tecnología asistida por computadora (CAX) y la fabricación asistida por ordenador (CAM). Para el monitoreo e implementación de un sistema de control se utiliza la tecnología informática junto con los procesos industriales.

1.2.3. Sistema de control

La palabra sistema alude a un conjunto de componentes que, en sincronía, trabajan de manera ordenada para alcanzar la finalización de una tarea o una meta específica. Un sistema de control es una rama de la automatización que estudia y representa el funcionamiento del sistema físico en un modelo matemático para el control de una unidad o también una cantidad física. La finalidad del control es la eliminación de la intervención humana regulando su propia conducta o la de otro sistema con un funcionamiento predeterminado. De esta manera, se reducen las probabilidades de fallas. Su precisión y exactitud son importantes en el desarrollo de circuitos semiconductores o el sistema de soldadura de automóviles robotizados, entre otras. Un sistema de control se clasifica, según su aplicación de la siguiente manera:

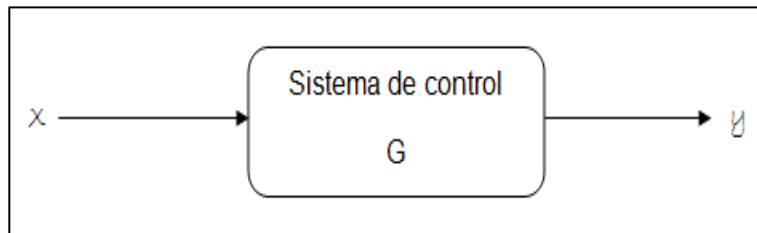
- Sistema de control de lazo abierto.
- Sistema de control de lazo cerrado.

1.2.3.1. Sistema de lazo abierto

En un sistema de lazo abierto. Consta de señal de entrada (x) que actúa sobre un sistema y proporciona una señal de salida (y) para llevar a cabo un proceso o una finalidad. Esto quiere decir que, este sistema consta únicamente

de un controlador que, a su vez, gobierna un sistema con su señal de salida sin ajustar constantemente las acciones tomadas.

Figura 2. **Sistema de control de lazo abierto**



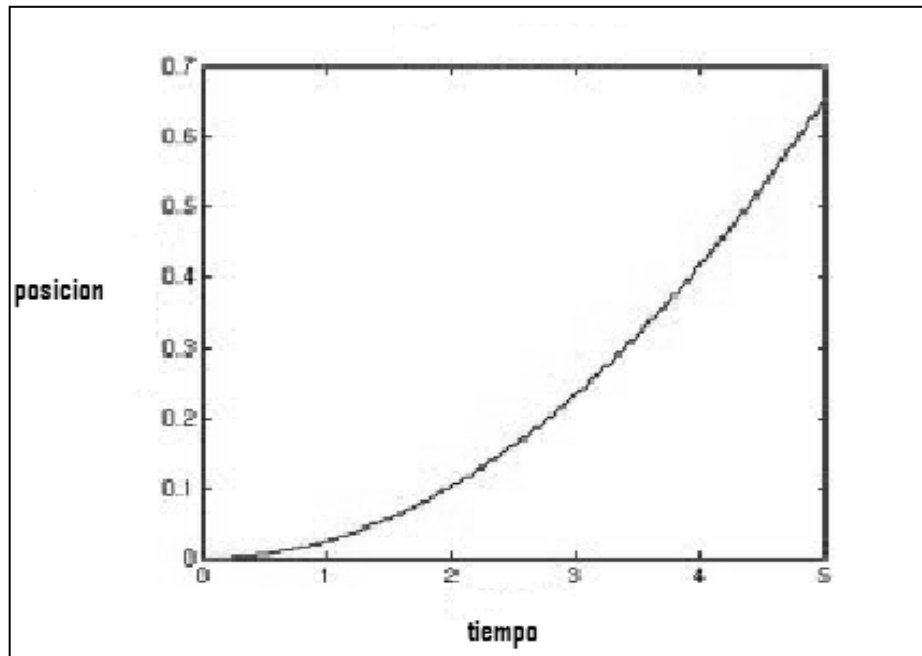
Fuente: elaboración propia.

Como se observa en la figura 2, la acción de control se genera en una sola dirección, la variable x indica cualquier valor de control para el sistema (sistema de control con valor de transferencia G) en la entrada, la variable y indica un valor de salida que se utiliza para el gobierno de algún sistema. Para el sistema de lazo abierto mostrado en la figura anterior, el flujo de control está determinado de izquierda a derecha. Este tipo de sistema se caracteriza por:

- Ser sencillos.
- No son muy estables.
- Económicos.
- Solo depende de la calibración sin asegurar precisión.

En un sistema de lazo abierto se determina que, para una señal de entrada (X) se producirá una señal de salida (y) en función de la entrada, el sistema funcionará de manera que se tendrá una relación cuasi lineal e inestable como se muestra en la figura 3.

Figura 3. **Relación de un sistema de control lazo abierto**

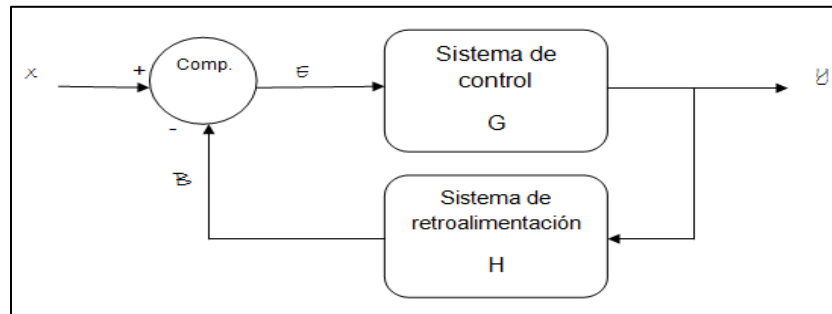


Fuente: elaboración propia.

1.2.3.2. **Sistema de lazo cerrado**

En un sistema de lazo cerrado se compone de un sistema de retroalimentación que, vinculado al sistema de control, forman un circuito cerrado. En estos sistemas utilizan la retroalimentación y la acción de un comparador para ajustar constantemente la acción de control que se encuentra en función de la señal de salida y la entrada.

Figura 4. **Sistema de control de lazo cerrado**



Fuente: elaboración propia.

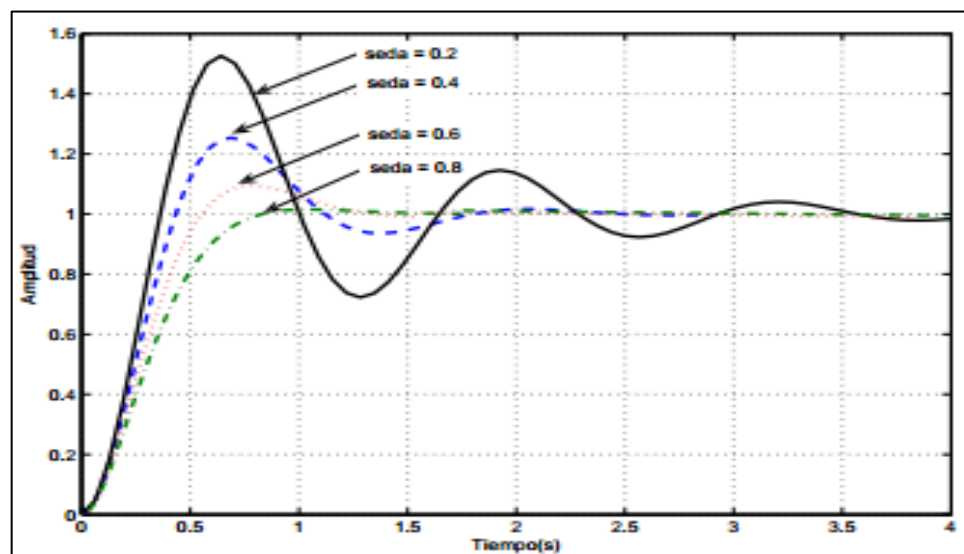
En un sistema de lazo cerrado también hay una señal de entrada (x) y una señal de salida (y), observa que se toma una muestra de la señal de salida y es degradada, a través del bloque H para producir una señal (B) que será comparada con la señal de entrada (x). La señal producida (E) es la resultante de la comparación y se llama señal de error que actúa directamente en el sistema de control para producir una salida. La señal (B) se denomina también señal de retroalimentación. Esta puede ser una fracción o función de la señal de salida o puede ser igual a la señal de salida.

Se puede llamar al sistema de lazo cerrado como sistema retroalimentado inverso y tiene la característica de ajustar constantemente de manera inversa la señal de salida respecto a la entrada. Esto es, si la señal de salida (y) es deficiente la señal de retroalimentación (B) será muy pequeña y la comparación con la señal de entrada ($X-B$) producirá una señal de error (E) bastante grande y el sistema producirá una señal de salida (y) con mayor eficiencia, compensando así, la deficiencia que se tenía. De igual manera si la señal de salida (y) es suficiente, la señal de retroalimentación (B) será mayor y la comparación con la señal de entrada ($X-B$) producirá una señal de error (E)

bastante pequeña y el sistema producirá una señal de salida (y) con menor eficiencia compensando, así la suficiencia que se tenía.

En la figura 5, se muestra la gráfica de respuesta transitoria de un sistema de lazo cerrado. Como se observa, este sistema es muy complejo, debido a esto, se logra una gran estabilidad de mando. El sistema de control estabiliza la señal de salida, debido a la comparación que hace con la señal de entrada hasta lograr una respuesta estable del sistema.

Figura 5. **Respuesta transitoria de un sistema de lazo cerrado**



Fuente: *Análisis de la respuesta temporal*. http://prof.usb.ve/montbrun/Respuesta_Temporal_sin_error_ni_estabilidad.pdf. Consulta: 18 de enero de 2011.

Este sistema se caracteriza por:

- Su complejidad en la implementación.
- Tener trayectoria de retroalimentación.

- La comparación de la entrada con la salida para el control del sistema.
- Ser estable a perturbaciones y variaciones.

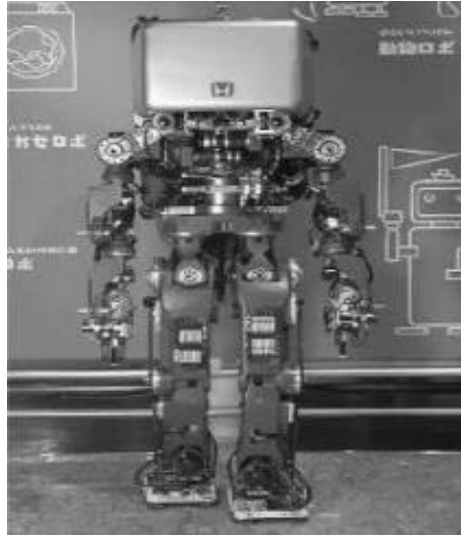
1.3. Anatomía del robot industrial

Este trabajo parte del estudio en el diseño y funcionamiento de un robot industrial que se utiliza para propósitos iniciales en el estudio de la robótica, llegando al nivel más complejo en la aplicación industrial. Un robot de esta naturaleza está diseñado para realizar un trabajo productivo.

La característica del robot humanoide es que se construye, trata que su forma sea igual o muy similar a la humana. Sin embargo, lo más importante es que desempeña funciones de movimientos de desplazamiento (caminar y subir escaleras), manipular objetos, expresiones faciales entre muchas otras propias del ser humano. En la anatomía de un robot humanoide su complejidad aumenta con el incremento en los grados de libertad del robot. Entre ellos, están los movimientos de brazos, piernas, dorso, brazos, hombros, cabeza, entre otras. En los robots bípedos, como su nombre lo indica, utiliza dos patas para su locomoción. Es uno de los principales retos al desarrollo del robot concentrando su centro de gravedad cerca del pie y no sobre el pie como inicialmente lo había desarrollado la empresa pionera Honda.

Otro aspecto importante que se debe considerar es el peso y las dimensiones de cada parte o eslabones en los cuales estará compuesta la estructura mecánica, ya que depende de los actuadores, motores y la energía de consumo en el robot. Con esto, se detalla a grandes rasgos la complejidad de un robot humanoide, y la principal característica de su locomoción bípeda de traslación, manipulación de objetos, con el objetivo de serle útil al ser humano. Se muestra en la figura 6 la gráfica de los pasos de un robot bípedo para su

Figura 7. **Prototipo del primer robot bípedo**



Fuente: BONELL SANCHEZ, Marc. *Construcción de un robot humanoide*. <http://upcommons.upc.edu/pfc/bitstream/2099.1/12840/3/pfc1.pdf>. Consulta: 24 de enero de 2011.

Figura 8. **Aplicación del robot ASIMO**



Fuente: BONELL SANCHEZ, Marc. *Construcción de un robot humanoide*. <http://upcommons.upc.edu/pfc/bitstream/2099.1/12840/3/pfc1.pdf>. Consulta: 24 de enero de 2011.

La característica de un robot zoomorfo es que su sistema de locomoción y forma imita a los diversos seres vivos. Entre la forma que puede adoptar, se encuentra la de un animal acuático, reptil, mamífero de cuatro patas, entre otras.

En la figura 9 se muestra un robot zoomorfo japonés como mascota.

Figura 9. **Robot zoomorfo**



Fuente: *Robótica*. <https://sites.google.com/a/unitecnica.net/dmhenao/articulos/zoomorficos>.

Consulta: 25 de enero de 2011.

La anatomía del robot industrial comprende la construcción física del robot. Un robot industrial, como ya se mencionó, se utiliza para realizar un trabajo productivo y está inspirado en la anatomía del brazo humano capaz de realizar casi cualquier tarea. Comprende el cuerpo, brazo y muñeca. El robot no posee movimiento de traslación y se encuentra montado en una base sujeta al suelo. El cuerpo está unido a la base y el brazo está unido al cuerpo. Al cuerpo, brazo y muñeca; también se le llama manipulador. El efector final (mano, pinzas, herramienta, etc.) que es la encargada de ejecutar el trabajo o aplicación va unido a la muñeca y el efector final no se considera parte de la anatomía del robot.

Es de importancia mencionar que las articulaciones del cuerpo y del brazo se utilizan para situar al efector final y las articulaciones de la muñeca, para orientar al efector final. La estructura mecánica de un robot está formada por los siguientes elementos: estructura mecánica, transmisiones, sistema de accionamiento, sistema sensorial, sistemas de control y elementos terminales. Todo este conjunto forma el robot mediante eslabones unidos a articulaciones que permiten el movimiento relativo.

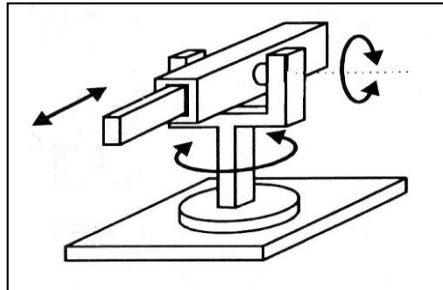
1.3.1. Configuraciones del robot

Existen cuatro configuraciones básicas para el robot industrial y son las siguientes:

- Configuración polar.
- Configuración cilíndrica.
- Configuración cartesiana.
- Configuración de brazo articulado.

La configuración polar o esférica el robot se encuentra montado en una base que le permite girar, contiene un pivote para bajar o subir y un brazo telescópico que faculta el movimiento longitudinal capaz de hacer movimientos en un espacio esférico; de ahí proviene el nombre de robot polar. Contiene tres grados de libertad y los movimientos son: rotacional, angular y lineal, como se muestra en la figura 10.

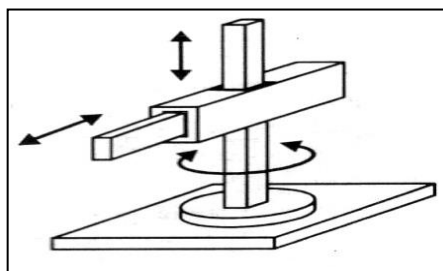
Figura 10. **Configuración polar o esférica**



Fuente: SOTOMAYOR, Nelson. *Robótica industrial*. <http://ciecfie.epn.edu.ec/Material/4toNivel/Robotica/Robotica.pdf>. Consulta: 5 de febrero de 2011.

La configuración cilíndrica tiene un movimiento de rotación y también se encuentra montado en una base sujeta al suelo. Tiene un movimiento longitudinal vertical montado sobre una columna (movimiento prismático) que le permite bajar y subir y el brazo tiene un movimiento longitudinal horizontal que se encuentra en un dispositivo deslizante para un movimiento radial. Este robot es ideal para espacios de trabajo redondos con una configuración cilíndrica. Presenta tres grados de libertad y los movimientos son: rotacional y longitudinal, como se muestra en la figura 11.

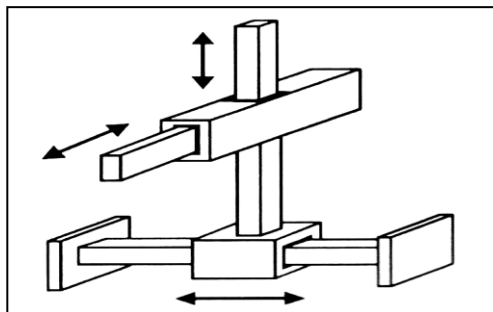
Figura 11. **Configuración cilíndrica**



Fuente: SOTOMAYOR, Nelson. *Robótica industrial*. <http://ciecfie.epn.edu.ec/Material/4toNivel/Robotica/Robotica.pdf>. Consulta: 6 de febrero de 2011.

La configuración cartesiana está inspirada en un sistema de coordenadas cartesianas con cada movimiento prismático en los ejes x, y, z perpendiculares entre sí. Es ideal para espacios de trabajos grandes, donde se requiere precisión. Contiene tres grados de libertad con cada movimiento de forma longitudinal que corresponde a los ejes x, y, z del sistema cartesiano; como se muestra en la figura 12.

Figura 12. **Configuración cartesiana**

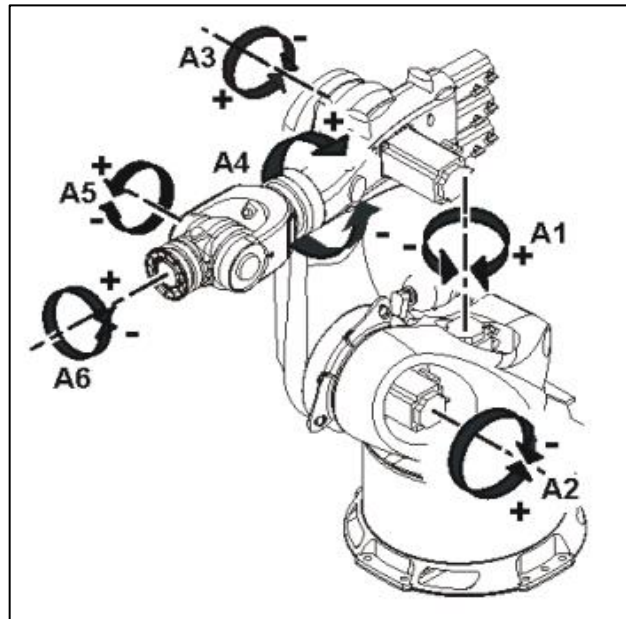


Fuente: SOTOMAYOR, Nelson. *Robótica industrial*. <http://ciecfie.epn.edu.ec/Material/4toNivel/Robotica/Robotica.pdf>. Consulta: 6 de febrero de 2011.

La configuración de brazo articulado es, en sí, el más completo y está inspirado en el brazo humano. Se construye con componentes rectos unidos por articulaciones y montado sobre una base giratoria sujeta al suelo. Este robot es una copia del brazo humano con cintura, hombro, codo y muñeca. Tiene un movimiento rotacional en la base y dos con movimiento angular para el hombro y codo con una muñeca unidad al antebrazo con un movimiento rotacional y uno torsional. El robot utiliza tres juntas o articulaciones para posicionarse y otras tres, para orientar el efector final y es capaz de realizar movimiento por interpolación lineal (este movimiento requiere que dos o más articulaciones se muevan simultáneamente). Su movimiento por naturaleza es de interpolación por articulación, tanto rotacional como angular. Los grados de libertad para este

robot son de seis grado de libertad con movimiento rotacional y torsional, como se muestra en la figura 13.

Figura 13. **Configuración articulado (imitación del brazo humano)**



Fuente: kuka kr 500 forttec. http://www.KUKA-robotics.com/res/sps/e6c77545-9030-49b1-93f5-4d17c92173aa_Spez_KR_500-3_MT_en.pdf. Consulta: 7 de febrero de 2011.

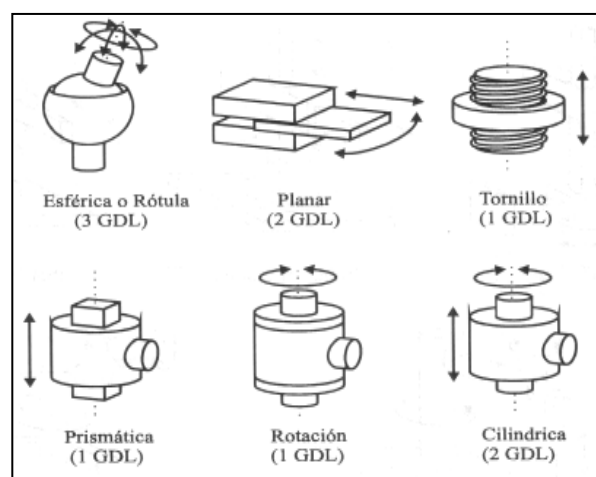
1.3.2. **Movimientos de articulaciones**

Para un robot industrial se tiene dos tipos de movimientos, el movimiento de brazo y cuerpo para posicionar el efector final y el movimiento de la muñeca, para orientar el efector final. A cada movimiento en una articulación se le denomina grado de libertad (DGL). Los eslabones conforman la parte sólida del robot y, para la conexión de las diversas articulaciones se emplean uniones; las juntas o articulaciones son los acoplamientos móviles entre las uniones. En una secuencia de unión-articulación-unión se le llama unión de entrada al

eslabón y la unión de salida es la que se desplaza con respecto a la entrada. A esta secuencia se le llama movimiento de articulación accionada. En el robot industrial los movimientos son relativos de las uniones contiguas y pueden ser movimientos lineales, movimientos rotacionales o una combinación de ambos.

Existen seis diferentes tipos de articulaciones, como se muestra en la figura 14. Para el robot industrial solo se utilizan dos de los seis tipos de articulaciones la prismática o lineal y la rotacional. Cada articulación solo contiene un grado de libertad (DGL) y, para un robot, el número total de grados de libertad es igual al número de articulaciones del que está compuesto. Para una articulación prismática o lineal el eslabón o junta se apoya sobre un deslizador, debido a un tornillo sin fin montado en un motor o, mediante un cilindro hidráulico o neumático. Una articulación rotacional se logra con un motor y un sistema de transmisión o por cilindros hidráulicos, cilindros neumáticos o palancas.

Figura 14. **Diferentes tipos de articulaciones**



Fuente: SOTOMAYOR, Nelson. *Robótica industrial*. <http://ciecfie.epn.edu.ec/Material/4toNivel/Robotica/Robotica.pdf>. Consulta: 10 de febrero de 2011.

1.3.3. Sistemas de transmisión

En el diseño de un robot, se considera la aceleración angular en los extremos de los eslabones, debido a la unión de entrada en cada articulación. También se considera el momento de inercia del peso de cada eslabón y el eslabón consecutivo, así como el de los pares estáticos que deben vencer los actuadores dependiendo de la distancia desde la masa hasta el actuador. Estas consideraciones obligan a tener los actuadores lo más cerca posible de la base, con ello, conlleva a utilizar sistemas de transmisión que trasladen el movimiento desde los actuadores hasta las articulaciones. Estas también se utilizan para convertir movimiento rotacional a lineal y viceversa.

Se buscan los siguientes fines al utilizar un sistema de transmisión: se debe de buscar un tamaño y peso considerable; además, se debe evitar evitar juegos y vibraciones, así como también sistemas de transmisiones de gran rendimiento. En el robot se emplean accionamiento directo como sistemas de transmisión y transformación para accionar las articulaciones del robot. Algunos de los sistemas de transmisión empleados son los siguientes: reductor armónico, reductor piñón-corona cónicos, correa dentada-polea, tornillo sin fin-corona, husillo con tuerca a bolas, correa dentada con guía lineal de bolas, piñón cremallera, *harmonic drive*, etc. En la tabla II, se clasifican los movimientos de entrada y salida lineal o circular.

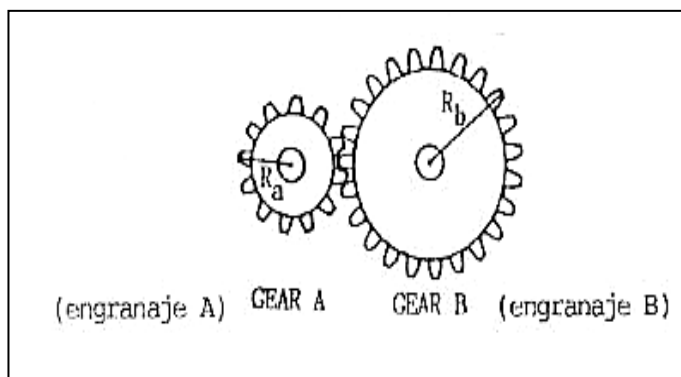
Tabla II. Descripción de los sistemas de transmisión

Sistemas de transmisión para robots			
Entrada-Salida	Denominación	Ventajas	Inconvenientes
Circular-Circular	Engranaje	Pares altos	Holguras
	Correa dentada	Distancia grande	Ruido
	Cadena	Distancia grande	Giro limitado
	Paralelogramo		Deformabilidad
	Cable		
Circular-Lineal	Tornillo sin fin	Poca holgura	Rozamiento
	Cremallera	Holgura media	Rozamiento
Lineal-Circular	Paralelogramo articulado	Holgura media	Control difícil
	Cremallera		Rozamiento

Fuente: SOTOMAYOR, Nelson. *Robótica industrial*. <http://ciecfie.epn.edu.ec/Material/4toNivel/Robotica/Robotica.pdf>. Consulta: 10 de febrero de 2011.

Haré un estudio para un sistemas de engranes como se muestra en la figura 15. Estos sistemas se usan para mover la base del robot con un engrane A montado directamente al eje del motor y un engrane B sujeta a la articulación de la base del robot.

Figura 15. Sistema de transmisión directa



Fuente: *El brazo robot scorbot-er III*. <http://repositorio.bib.upct.es/dspace/bitstream/10317/133/3/Cap%C3%ADtulo2.PDF>. Consulta: 12 de febrero de 2011.

Como puede verse en la figura 15, el engranaje A tiene un radio $R_a = a$ mm y el engranaje B tiene un radio $R_b = b$ mm, el sistema tiene la siguientes secuencia el motor hace girar el engranaje A en sentido de la agujas de reloj a una velocidad ω_A , entonces, el engranaje B girará en sentido contrario a la agujas de reloj a una velocidad ω_B y, ésta a su vez, hace girar a la base del robot. La relación de transmisión, según la figura 15, por ello se determina por la expresión en la ecuación 1.

$$T_{AB} = \frac{R_B}{R_A} \quad [\text{Ec. 1.1}]$$

Donde:

T_{AB} : es la relación de transmisión del engranaje A al engranaje B.

R_B : Radio del engranaje acoplado a la base del robot.

R_A : Radio del engranaje acoplado al eje de salida del motor.

Esto implica que por cada vuelta del engranaje A con radio R_a , el engranaje B con R_b girará un número de vueltas iguala (R_a/R_b) . Otro método de representar la relación de transmisión es como relación entre el número de dientes de la circunferencia de cada engranaje.

$$T_{AB} = \frac{N_B}{N_A} \quad [\text{Ec. 1.2}]$$

La relación entre las velocidades angulares relativas entre los dos engranajes está determinada por la siguiente ecuación:

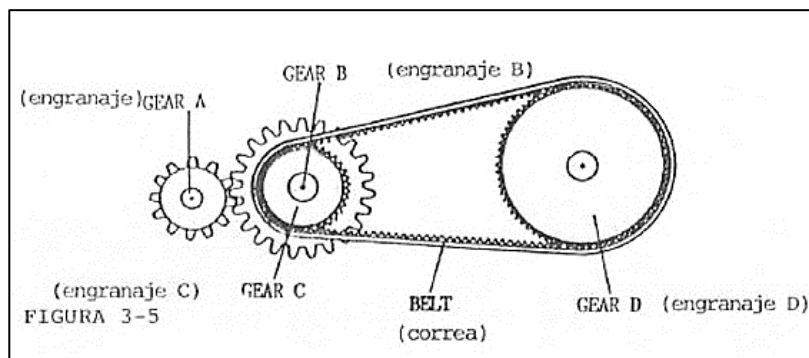
$$\frac{W_B}{W_A} = \frac{1}{T_{AB}} = \frac{N_A}{N_B} \quad [\text{Ec. 1.3}]$$

Donde:

ω_A : es la velocidad angular o de giro del engranaje A.

ω_B : es la velocidad angular o de giro del engranaje B.

Figura 16. **Sistema de transmisión indirecta entre etapas**



Fuente: *El brazo robot scrobot-er III*. <http://repositorio.bib.upct.es/dspace/bitstream/10317/133/3/Cap%C3%ADtulo2.PDF>. Consulta: 12 de febrero de 2011.

Ahora se analizará un sistema de transmisión de correa dentada y un sistema de engranajes A, B, C y D con N_A , N_B , N_C y N_D , que son los números de dientes de cada engranaje. La relación de transmisión total está determinada por el producto de la relación de transmisión en cada etapa y si se añade una etapa más la relación T_{AD} deberá ser multiplicada, por la siguiente relación añadida, para la relación de T_{AD} está dado por la ecuación 1.4.

$$T_{AD} = T_{AB} \cdot T_{AC} \quad [\text{Ec. 1.4}]$$

Y por último, se determinará la relación de velocidad angular del engranaje D al engranaje A, el cual está dado por la ecuación 1.5.

$$\frac{W_D}{W_A} = \frac{W_D}{W_C} \cdot \frac{W_B}{W_A} = \frac{1}{T_{CD}} \cdot \frac{1}{T_{AB}} = \frac{N_C}{N_D} \cdot \frac{N_A}{N_B} \quad [\text{Ec. 1.5}]$$

Ahora me concentro en el sistema de transmisión del *harmonic drive*, el cual es un sistema eficiente y preciso aplicado a la robótica y muchas aplicaciones de ingeniería. *El Harmonic Drive está compuesto por tres elementos distintos, pero con funciones bien definidas, como lo son el Wave Generator (WG), el FlexSpline (FS) y el Circular Spline (CS), el HD tiene tres posibles configuraciones debido a que cualquiera de los tres componentes antes mencionados puede servir de entrada o salida de movimiento de rotación.*

La configuración básica consiste en la fijación del *Circular Spline*, que permite una rotación de alta velocidad en el Wave Generator y una baja velocidad en el *FlexSpline*. De manera similar, con el FS fijo, el par en el WG puede incrementarse y transmitirse, a través del *Circular Spline*. Si tanto el *Circular Spline* como el *FlexSpline* son utilizados como entradas, el Wave Generator puede ser manejado a altas velocidades y bajo par. En general, usando diferentes combinaciones de rotación de los tres componentes del HD, se pueden lograr numerosas relaciones de transmisión y par.

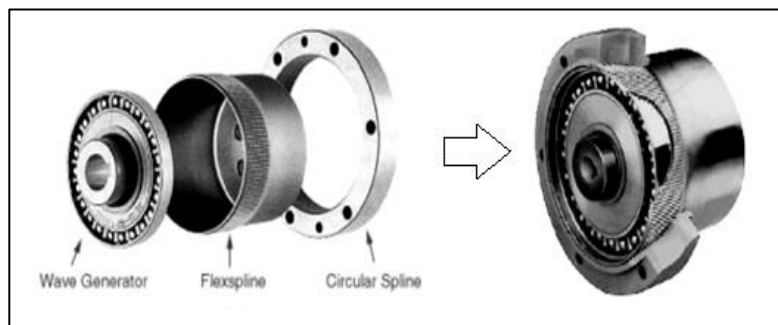
“Cuando se insertan en el FlexSpline y el Wave Generator en el Circular Spline, los dientes externos del FlexSpline engranan con los dientes internos del Circular Spline a lo largo del eje mayor de la elipse del Wave Generator. Los tres componentes de la transmisión pueden rotar a velocidades diferentes, aún estando en el mismo eje de rotación.

El mecanismo de rotación del Harmonic Drive comienza su recorrido en el momento que el Wave Generator tiene la zona de engrane en su eje mayor, y cuando esta zona se desplaza 180° alrededor de la circunferencia del Circular Spline, entonces el FlexSpline, que posee dos dientes menos que el Circular

Spline, tendrá un retraso de un diente relativo al Circular Spline. Por este engrane gradual y continuo de los dientes del engranaje, cada rotación del Wave Generator mueve al FlexSpline dos dientes hacia atrás sobre el Circular Spline.

Las relaciones de transmisión típicas del *harmonic drive* están en el rango de 50:1 a 200:1, aunque pueden obtenerse relaciones de transmisión de hasta 320:1 en una transmisión simple. Las eficiencias del mecanismo normalmente rondan el 85 %, también pueden alcanzar un 90 %; las transmisiones más pequeñas pueden proporcionar un par mínimo de salida de alrededor de 1,3N.m, a altas revoluciones (3 000 rpm); pero con relaciones de transmisión mínimas (50:1); por el contrario, los *harmonic drives* de alta potencia pueden presumir de hasta 10.000 N.m de capacidad de par”.²

Figura 17. **Mecanismo y ensamblaje del *harmonic drive***



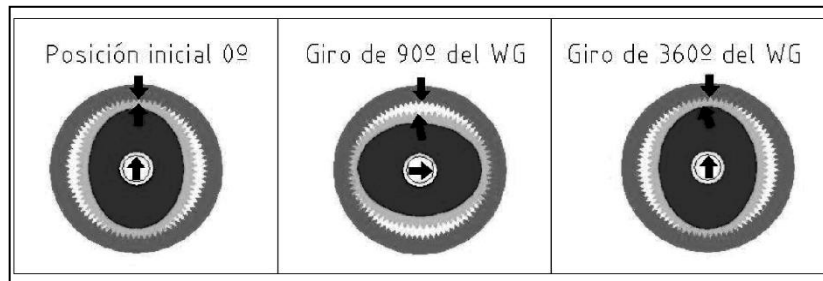
Fuente: CASTEJÓN, Cristina y otros. *Modelado de un sistema de transmisión harmonic drive.*

Análisis del error cinemático. <http://www2.uned.es/ribim/volumenes/Vol13N1Abril2009/>

V13N1A05%202009%20Castejon.pdf. Consulta: 19 de febrero de 2011.

² GONZÁLES, Víctor R. *Origen y desarrollo de la robótica.* [en línea]. http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/historia.htm. Consulta: 26 de marzo de 2016.

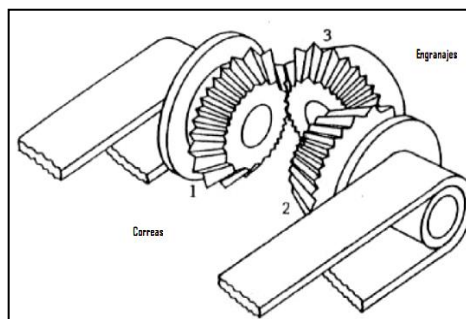
Figura 18. Operación del *harmonic drive*



Fuente: CASTEJÓN, Cristina y otros. *Modelado de un sistema de transmisión harmonic drive. Análisis del error cinemático*. <http://www2.uned.es/ribim/volumenes/Vol13N1Abril2009/V13N1A05%202009%20Castejon.pdf>. Consulta: 19 de febrero de 2011.

En la figura 19, se muestra los ejes de transmisión propuesta para las articulaciones de la muñeca, pero cuando los engranajes uno y dos se mueven en la misma dirección, el engranaje tres tiene un movimiento hacia arriba o hacia abajo, si los engranajes uno y dos se mueven en dirección contraria, el engranaje tres tiene un movimiento giratorio.

Figura 19. Mecanismo de movimiento para la muñeca del robot industrial

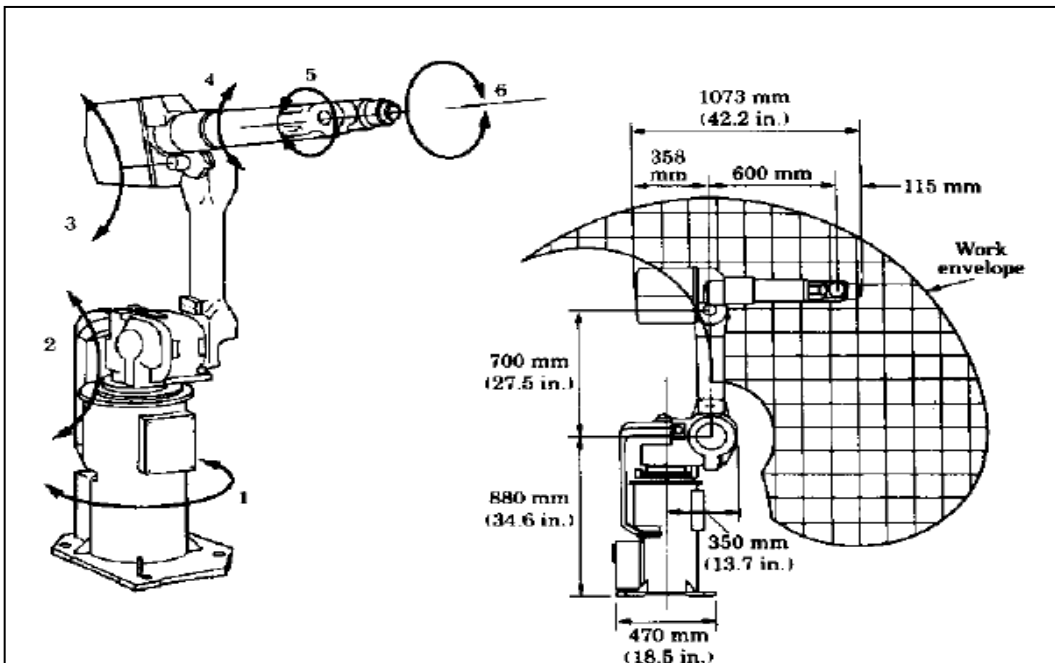


Fuente: *El brazo robot scrobot-er III*. <http://repositorio.bib.upct.es/dspace/bitstream/10317/133/3/Cap%C3%ADtulo2.PDF>. Consulta: 20 de febrero de 2011.

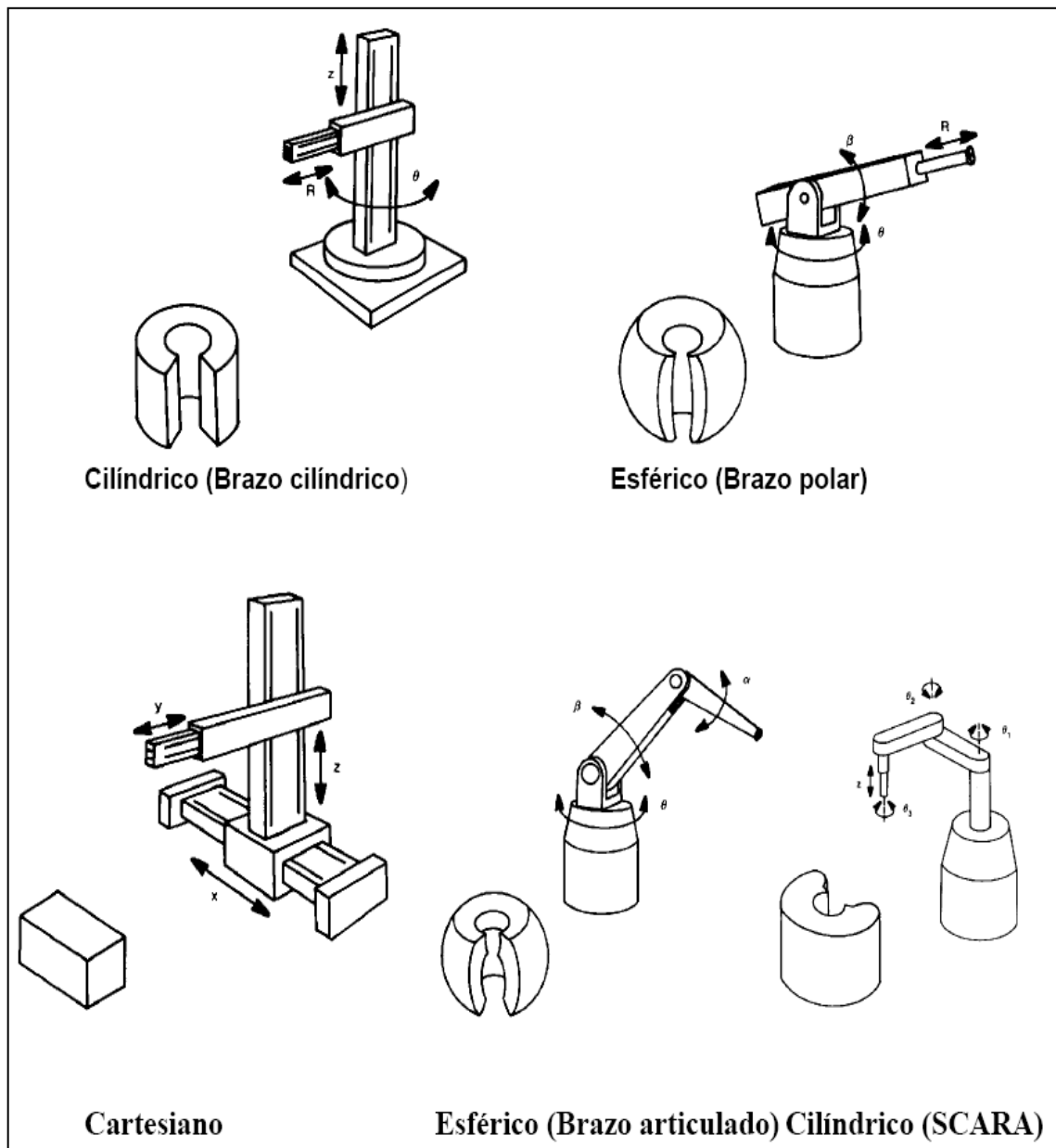
1.4. Espacio de trabajo del robot

Es el espacio físico en el cual el robot puede manipular el extremo de su muñeca. Se utiliza la muñeca como punto máximo o límite de extensión y no el efector final, para distintos efectores finales podrían tener diferentes tamaños, extendiendo o acortado el volumen de trabajo e incluso el efector unido a la muñeca podría no ser capaz de alcanzar algunos puntos dentro del volumen normal de trabajo, debido a la forma de los eslabones o combinación de la articulaciones; no todos los robots tienen el mismo volumen de trabajo, ya que cada robot dependiendo de su configuración, forma y tamaño. Se destacan los robots de brazo articulado y polar por su flexibilidad; comúnmente son usados en la industria automotriz para acabos, pintura, soldadura de arco, etc. En la figura 20, se muestran algunos ejemplos.

Figura 20. Volúmenes de trajo para distintitas configuraciones



Continuación de la figura 20.



Fuente: OCHOA, Oscar. *Robótica industrial*.

http://www.virtual.unal.edu.co/cursos/ingenieria/mecatronica/docs_curso/Anexos/TUTORIALcnc/DOCUMENTOS/TEORIA/ROBOTICA%20INDUSTRIAL.pdf. Consulta: 23 de febrero de 2011.

1.5. Impulsores o fuentes de energía

En esta sección se estudia las fuentes de energía que necesita el robot para moverse y ejercer un trabajo. De las fuentes de energía que se utilizan se mencionan las siguientes: energía eléctrica, energía hidráulica y la energía neumática. El tipo de energía determina el rendimiento, la capacidad de carga y la velocidad del robot. En otras palabras, para una aplicación específica se necesita un tipo específico de energía; en otros casos, surge de la combinación de dos tipos de impulsores. Para los robots más sofisticados se utilizan los impulsores eléctricos e hidráulicos, por su potencia y precisión.

Se hace referencia a la energía eléctrica como la más importante, por ser más limpia, confiable y fácil de implementar. Su capacidad de carga, así como la velocidad no es muy amplia, por lo cual es muy aceptable. Es seguro utilizarla en ambientes inflamables con accesorios de seguridad y la repetitividad de los robots eléctricos suelen ser los mejores. Se utilizan los sistemas compuestos por *Harmonics Drives*, por ser más preciso y eficientes que los sistemas de precisión tradicionales.

Los sistemas de impulsión o energía hidráulica utilizan aceite o líquido a presión como fuente de energía. Se aplican a los robots más grandes capaces de trabajar con elevadas capacidades de carga. Pueden ser utilizadas en ambientes inflamables con seguridad con mayor resistencia mecánica y mayor velocidad: generalmente, necesitan mayor espacio y están propensas a derrames o fugas de aceite. Está diseñada para utilizarse en articulaciones rotacionales o lineales utilizando poleas y pistones.

La impulsión neumática utiliza gas o aire comprimido como fuente de energía; también se utiliza para robots más pequeños con dos a cuatro grados

de libertad. Son utilizados por robots con menor exactitud, y principalmente, se utilizan para agarrar y situar. Esta tiene un menor costo con una potencia neumática adaptada a pistones, se utiliza para movimientos lineales y actuadores giratoria.

2. TECNOLOGÍA APLICADA

2.1. Actuadores

Los actuadores proporcionan el movimiento motriz de las articulaciones. Estas se clasifican por el tipo y forma de energía que utilizan para su operación, entre estos se pueden mencionar: actuadores eléctricos, actuadores hidráulicos y actuadores. Los actuadores no son propios para la aplicación de un robot, también se utilizan en otros dispositivos industriales. Para utilizarlos en un robot se buscan los siguientes criterios de relación potencia-tamaño y potencia-peso, para minimizar el momento de inercia en cada articulación para cada actuador involucrado en la estructura final.

2.1.1. Actuadores eléctricos

Los actuadores eléctricos utilizadas en robótica pueden producir un movimiento con la acción de una corriente o voltaje. Entre los actuadores utilizados se mencionan los motores eléctricos de corriente alterna, los motores de corriente continua y los motores paso a paso (*step*); éstos son los más utilizados por su característica son los motores de corriente continua y los motores paso a paso. Los motores eléctricos se clasifican en motores con escobillas y sin escobillas.

2.1.1.1. Motores de corriente continua con y sin escobilla

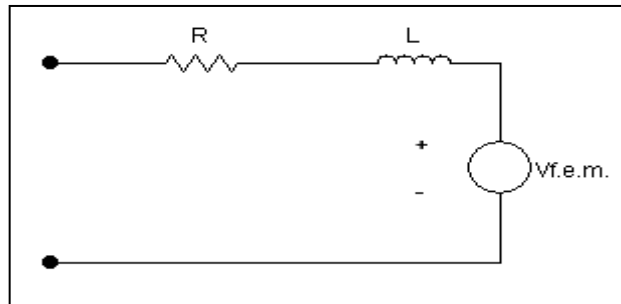
Este tipo de motor se basa en el principio del conductor con una corriente i en dirección del eje “x” que circula a través de ella, sumergida en un campo magnético B en dirección del eje “- z”. Esta producirá una fuerza F en dirección del eje “y” dada por la regla de la mano derecha. Si se introdujera una espira de corriente se producirá una fuerza girando la espira, de modo que esta sea atravesada por la máxima cantidad de flujo del campo magnético y si se invirtiera el sentido de la corriente, en ese momento, la espira continuará girando, debido a la fuerza producida.

En un motor de corriente continua hay varias espiras enrolladas sobre un cilindro de material magnético que gira alrededor de un eje fijo (rotor) formando la bobina del motor. Cada bobina se conecta al conmutador que tiene la función de invertir el sentido de la corriente en el circuito cuando esta ya haya girado, produce una nueva fuerza en la misma dirección, y la bobina continuará girando en la misma dirección. Sobre el conmutador actúan las escobillas, generalmente, de carbón. Para hacer girar el motor en sentido contrario basta con invertir el sentido de la corriente en las escobillas. Otro componente del motor de corriente continua es el estator. Está formado por imanes permanentes para motores pequeños o por electro imanes para motores grandes que producen un campo magnético fijo que, con el campo magnético del rotor, actúan de una forma que se repelen produciendo así el movimiento del rotor.

Es importante representar el circuito equivalente y la gráfica de relación voltaje, velocidad angular y torque (par) del motor para una mejor comprensión sobre la forma de funcionamiento del motor. Se muestra el circuito equivalente

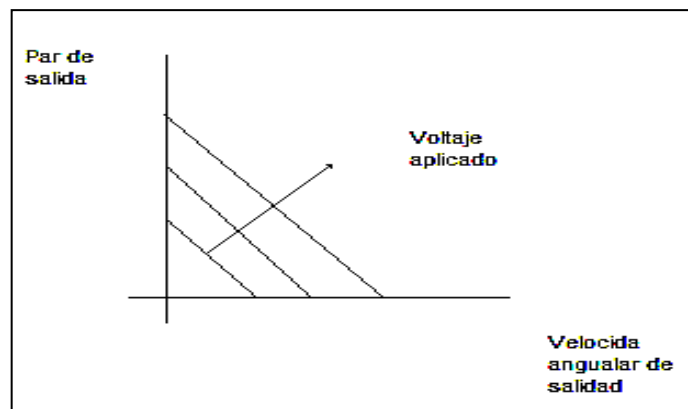
eléctrico para el motor de corriente continua en la figura 21, con una resistencia R , una inductancia L y una f.e.m en serie que se contrapone al voltaje aplicado y en la figura 22 se muestra la gráfica de la relación entre el par de salida y la velocidad angular.

Figura 21. **Circuito equivalente eléctrico del motor de imanes permanentes**



Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. p. 41.

Figura 22. **Gráfica de relación entre el par de salida y la velocidad angular al voltaje aplicado**



Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. p. 41.

Al interpretar la gráfica de relación se observa que para $\omega = 0$ el par obtenido sería igual al voltaje aplicado V . También se observa que para un voltaje aplicado el par decrece al aumentar la velocidad angular y viceversa. De esta manera, se da inicio al estudio del actuador empezando con el estudio del motor D.C. con imanes permanentes. Esto servirá para comprender de mejor manera, el funcionamiento de un motor eléctrico. Para aplicaciones en la realidad, se utilizan motores grandes sin imanes permanentes, como los motores pequeños utilizando electro imanes en el estator para producir el flujo de campo magnético. Con estos imanes se logran diferentes configuraciones. Los circuitos del estator se pueden montar con respecto a las del rotor en serie, en paralelo, serie paralelo e independiente. A continuación se mencionan las características para cada configuración.

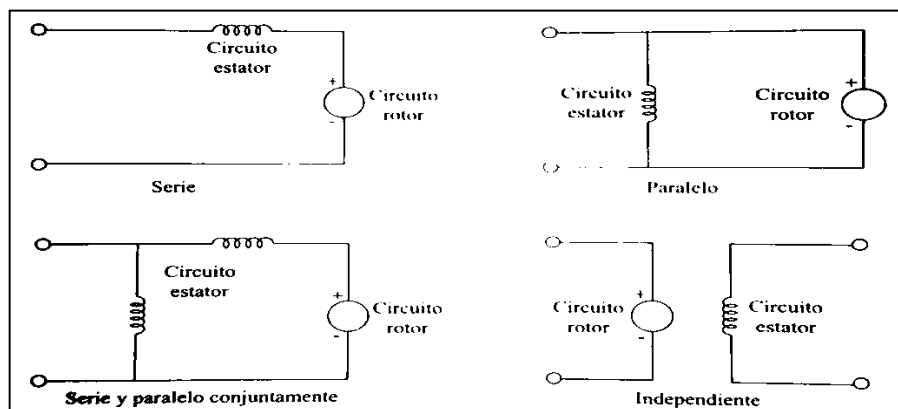
- La configuración en serie. El circuito del estator se encuentra en serie con el circuito del rotor, de esta manera se logra el mayor par en el arranque y con la mayor velocidad angular sin carga. En esta configuración no se puede invertir el sentido de giro del motor, ya que al cambiar la polaridad de voltaje en la entrada también cambia la polaridad en el estator como en el rotor; por lo tanto, no se lograría una inversión de giro, como sucede en el caso para imanes permanentes con un voltaje de entrada fijo al aumentar el par de salida disminuye la velocidad angular de salida y viceversa.
- La configuración en paralelo. Se logra el par más bajo en el arranque; también tiene una buena regulación de velocidad. Una de las características destacables es que para un voltaje dado la velocidad de giro permanece constante, independiente de la carga aplicada que requiere dicho torque. En esta configuración no se puede cambiar el

sentido de giro del motor, ya que ocurre una inversión de polaridad para el estator y rotor como en el caso anterior.

- La configuración combinada. Se obtendrá las características de la configuración en serie con un elevado par de arranque y las características de la configuración en paralelo con una buena regulación de velocidad de giro en el motor.
- La configuración independiente (figura 23). Ambos circuitos son independientes entre sí. En esta configuración se puede controlar por separado el par y la velocidad de giro del motor. Tiene la ventaja que se puede cambiar el sentido de giro solo con invertir la polaridad en uno de los circuitos.

A continuación se muestran los circuitos equivalentes para cada una de las diferentes configuraciones que se mencionaron.

Figura 23. **Circuitos equivalentes para las diferentes configuraciones del estator y rotor**

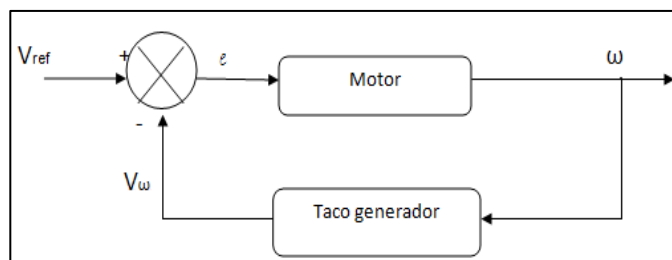


Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. p. 41.

Estos circuitos proporcionan suficiente información acerca del funcionamiento y comportamiento del motor. Como estudio el par y la velocidad de giro dependen del voltaje aplicado para el caso de imanes permanentes, para el caso de motores con electroimanes. En el estator, la velocidad de giro se puede controlar variando la corriente del estator como el rotor. Se observa que dichos circuitos son del tipo de bucle abierto en que las condiciones de funcionamiento permanecen constantes en el tiempo.

Si se requiere variar la velocidad de giro, se utiliza un control de bucle cerrado. Esto se logra colocando un tacogenerador a la salida del motor. Este funciona de forma inversa al motor de imanes permanentes, pero en lugar de alimentarle con un voltaje en la terminales de entrada, para obtener movimiento, se conecta al eje de salida del motor y se obtendrá un voltaje proporcional a la velocidad de giro del motor que, junto a un circuito comparador o controlador es posible controlar la velocidad de giro, así como se muestra en la figura 24.

Figura 24. **Circuito de control de velocidad de giro**



Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. p. 42.

Los motores de corriente continua sin escobillas solucionan el problema de desgaste de las escobillas; además evita el constante mantenimiento del motor, debido a que se eliminan el uso del conmutado y de las escobillas. El motor está construido con el imán permanente en el rotor y el circuito por donde

circula la corriente se encuentra en el estator que creará fuerza, debido al campo magnético producido por el circuito. El circuito del estator experimentará una fuerza de repulsión dada por el campo magnético del rotor con la diferencia de que ahora el circuito de imán permanente se encuentra empotrado y, por la tercera ley de Newton, existirá una fuerza de igual magnitud, pero en dirección contraria. Así, con este principio se obtendrá el movimiento del rotor por la existencia de esta fuerza. El rotor continuará girando gracias a que en el estator existen varios circuitos o bobinados que son conmutados de forma electrónica y con ello siempre existirá una fuerza sobre el rotor.

2.1.1.2. Motores paso a paso (*stepper*)

Algunas de las características de estos motores es su precisión y la forma de operación por medio de pulsos discretos. Cada pulso hace girar al motor en un ángulo fijo que va desde 90° , por pulso hasta $1,8^\circ$, por pulso. Estos motores están formados por un rotor y un estator con diferentes circuitos colocados angularmente que se activan de forma secuencial, según el tren de pulsos con que se alimenta. Existen tres tipos de motores paso a paso y que se describen a continuación:

- Motor pasó a paso de reluctancia variable. Este tipo de motor se construye de tal forma que los polos del estator son superiores al del rotor. El rotor no contiene ningún circuito, sino está formado de acero dulce, cilíndrico y con dos polos menores al del estator. En el estator se encuentran los devanados y cuando la corriente circula en cada devanado opuesto se produce un campo magnético que pasa, a través de los polos del estator del devanado activo; estas pasan a los polos más cercanos del rotor. Las líneas de campo magnético buscan alinearse moviendo el rotor hasta que los polos de rotor y del estator queden

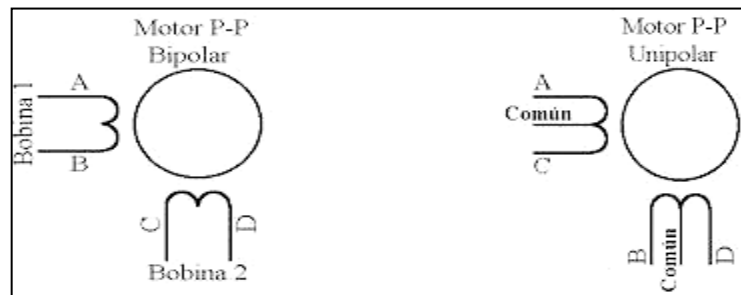
alineados llegando al estado de reluctancia mínima. Este tipo de motor produce avances de 7.5° o 15° .

- Motores paso a paso de imán permanente. Está construido con los circuitos de los devanados en el estator forman dos polos por circuito y el rotor es un imán permanente. Cuando por el circuito circula una corriente, se creará un campo magnético que atraviesa el rotor desplazándolo hasta alinear los polos de este con los polos de estator. Si se alimentara el siguiente circuito de manera que, las polaridades sean similares a la anterior, el rotor se desplazará hasta alinearse nuevamente y, al conmutar la corriente de manera secuencial, se logra el desplazamiento del rotor con cada pulso. En este tipo de motor, los ángulos de avance por cada pulso son de 1.8° , 7.5° , 15° , 30° , 34° y 90° .
- Motor paso a paso híbrido. Este tipo de motor se construyen combinando un imán permanente dentro del estator (tapones) de hierro multi dentado logrando con esto las características de reluctancia mínima y la del imán, cuando se energizan los circuitos de los devanado el rotor se alinea con el campo magnético y se coloca en la posición de reluctancia mínima. Los ángulos de giro para este motor son de $0,9^\circ$ y $1,8^\circ$, como se puede notar, son ideales para aplicaciones de alta precisión.

Los motores que más se utilizan en robótica son los de imán permanente. Los dos tipos que existen son: los motores paso a paso, los bipolares y los unipolares. Los motores paso a paso bipolares se distinguen por tener cuatro cables de conexión. El control de este motor es más complicado por la inversión del flujo de corriente en los devanados. Los motores paso a paso unipolares tienen cinco o seis cables de conexión, dependiendo de cómo estén contruidos internamente y son más simples de controlar. Para controlar un motor paso a

paso, solo se necesita una secuencia de pulsos a cada cable de control, bien sea unipolar o bipolar. A continuación, se muestra las figuras de los circuitos internos de los motores paso a paso.

Figura 25. **Circuitos de motores paso a paso bipolares y unipolares**



Fuente: *Stepper motors*. <http://www.todorobot.com.ar/informacion/tutorial%20stepper/stepper-tutorial.htm>. Consulta: marzo de 2011.

Para controlar un motor paso a paso bipolar se necesita realizar la inversión del flujo de corriente, para polarizar el campo magnético en el circuito del devanado en la secuencia, que se muestra en la tabla III.

Tabla III. **Secuencia de pulsos para un motor p-p bipolar**

Paso	Terminales			
	A	B	C	D
1	+V	-V	+V	-V
2	+V	-V	-V	+V
3	-V	+V	-V	+V
4	-V	+V	+V	-V

Fuente: *Stepper motors*. <http://www.todorobot.com.ar/informacion/tutorial%20stepper/stepper-tutorial.htm>. Consulta: marzo de 2011.

Para un motor de imán permanente unipolar se tienen tres formas posibles de operación, en modo normal, modo *wave drive* y de medio paso. En el modo normal, el motor gira un paso cada vez. En este modo se tienen energizado dos devanado o bobinas al mismo tiempo, con ello un alto torque de paso y retención. En la tabla IV, se muestra la secuencia de pulsos del motor p-p para el funcionamiento en modo normal.

Tabla IV. **Secuencia de pulsos para un motor p-p unipolar en modo normal**

Paso	Bobina A	Bobina B	Bobina C	Bobina D	Secuencia
1	ON	ON	OFF	OFF	
2	OFF	ON	ON	OFF	
3	OFF	OFF	ON	ON	
4	ON	OFF	OFF	ON	

Fuente: *Motores paso a paso.*

http://platea.pntic.mec.es/vgonzale/cyr_0204/cyr_01/robotica/sistema/motores_p-p.htm#funcionamiento. Consulta: marzo de 2011.

En el modo de funcionamiento *wave drive* es la forma más simple y fácil de trabajar. En este modo solo se tiene activa un devanado o bobina a la vez, en una secuencia consecutiva. La desventaja que se tiene al activar solo una bobina o devanado es que se tiene menor torque de paso y retención. En la tabla V, se muestra la secuencia para el modo *wave drive*.

Tabla V. Secuencia de pulsos para un motor p-p unipolar en modo *wave drive*

Paso	Bobina A	Bobina B	Bobina C	Bobina D	Secuencia
1	ON	OFF	OFF	OFF	
2	OFF	ON	OFF	OFF	
3	OFF	OFF	ON	OFF	
4	OFF	OFF	OFF	ON	

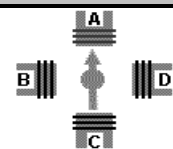
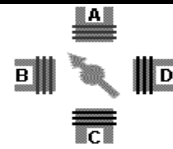
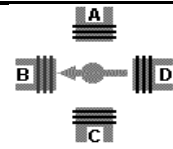
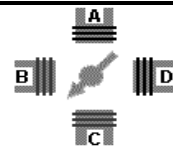
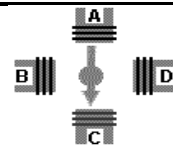
Fuente: *Motores paso a paso*.

http://platea.pntic.mec.es/vgonzale/cyr_0204/cyr_01/robotica/sistema/motores_p-p.htm#funcionamiento. Consulta: marzo de 2011.

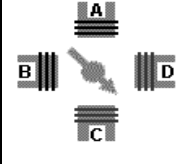
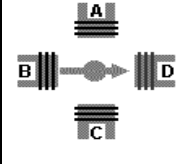
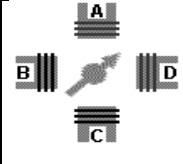
En el último modo de operación, el motor se puede utilizar para que trabaje en forma más precisa, en el modo de operación de medio paso que,

como su nombre lo indica, el motor gira solo medio paso a la vez, activa dos devanados o bobinas en una secuencia. En la secuencia posterior, se activa solo un devanado o bobina. Posteriormente, se repite los mismos procesos. En la tabla VI, se muestra la secuencia de ocho pasos para la operación de medio paso.

Tabla VI. **Secuencia de pulsos para un motor p-p unipolar en modo medio paso**

Paso	Bobina A	Bobina B	Bobina C	Bobina D	secuencia
1	ON	OFF	OFF	OFF	
2	ON	ON	OFF	OFF	
3	OFF	ON	OFF	OFF	
4	OFF	ON	ON	OFF	
5	OFF	OFF	ON	OFF	

Continuación de la tabla VI.

6	OFF	OFF	ON	ON	
7	OFF	OFF	OFF	ON	
8	ON	OFF	OFF	ON	

Fuente: *Motores paso a paso*. http://platea.pntic.mec.es/vgonzale/cyr_0204/cyr_01/robotica/sistema/motores_p-p.htm#funcionamiento. Consulta: marzo de 2011.

Si se requiere que el motor invierta el sentido de giro, basta invertir la secuencia de pasos para cada operación ya se sea bipolar y unipolar.

2.1.1.3. Motores de corriente alterna

Los motores de corriente alterna son los más potentes en cuanto al par de giro. Estos motores pueden ser monofásicos o polifásicos y cada una de estas puede ser de inducción o síncrono. Los motores monofásicos tienen aplicación cuando se requiere baja potencia y si se necesita mayor potencia se utilizan los motores polifásicos.

Los motores de inducción o jaula de ardilla están contruidos por barras de cobre o aluminio y en los extremos están sujetos por aros que forman un circuito eléctrico que, al estar sumergidas en un campo magnético variable, se

crea una f.e.m. en las barras por el que circula corriente. En el motor DC esta corriente junto con el campo magnético creara una fuerza en cada barra del rotor; como se puede entender el rotor no necesita de conexión alguna ya que la energía que casetita la obtiene por medio de la energía inducida que forman fuerzas, debido al campo magnético externo.

El campo magnético externo se crea, debido a varios devanados en el estator que, al ser alimentado con corriente alterna (AC), creara un campo magnético variable. Estos motores necesitan de un circuito para el impulso inicial, ya que el par neto es igual a cero. La velocidad de giro del motor está determinada por la frecuencia de la corriente alterna que alimenta al estator, pero en la realidad la velocidad de rotación es de 1 a 3 % de la frecuencia de la corriente, la velocidad de rotación del campo magnético se denomina velocidad síncrona.

Los motores de inducción polifásica o trifásica que a diferencia del motor de inducción monofásica consta de un estator con tres devanados y cada una de esta es alimentando a una de las tres líneas de alimentación trifásica comercial, los devanados están separadas unas de otras a 120° , ya que cada una alcanza su nivel máximo a 120° de desfase de al anterior. El campo magnético gira en torno a los polos del estator y el rotor completa una rotación en un ciclo exacto de la corriente. Este motor por su construcción no necesita de circuito de excitación y si se requiere cambio de dirección de giro solo necesita intercambiar dos de los circuitos de excitación.

La diferencia entre un motor de inducción y un motor síncrono es que el estator para el motor síncrono está construido por imanes permanentes y no por conductores como el motor de inducción. El estator es muy similar al de inducción en cuanto a monofásico o polifásico. Al igual que en el motor de

inducción, el campo magnético gira y el estator gira de igual manera 360° durante un ciclo completo. En este caso, la frecuencia de rotación es igual a la frecuencia de alimentación.

El inconveniente para el motor síncrono es que se tiene que utilizar un circuito de arranque, así como el uso de fuentes de frecuencia variable o variadores de frecuencia, para mantener el par constante en el motor se tiene que mantener constante la relación entre el voltaje aplicado y la frecuencia. Se puede utilizar el método de conversión de ac-cd-ac. De esta manera, el voltaje se mantiene constante, pero la frecuencia se puede variar utilizando un ciclo convertidor. Este convierte la frecuencia de la corriente alterna de entrada a una corriente alterna de salida con otra frecuencia sin pasar por el proceso de conversión de cd.

2.1.2. Actuadores hidráulicos y neumáticos

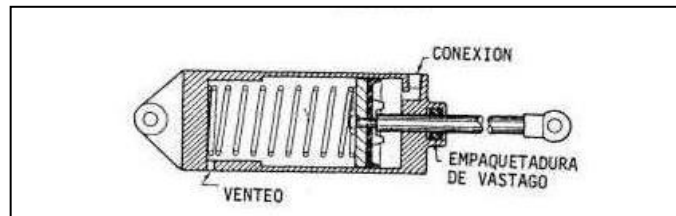
Los actuadores son las encargadas de proporcionar el movimiento y la fuerza necesaria para el accionamiento de algún mecanismo o, en este caso, al efector final utilizando algún fluido a presión o en la mayoría de los casos aceite como fuente de energía. Los actuadores hidráulicos se utilizan en maquinaria pesada y en aplicaciones de carga con útil. El funcionamiento de los sistemas hidráulicos y neumáticos muy similares. La energía es transmitida por medio de tuberías y esta se encuentra en función de la presión del fluido en la fuerza y el caudal para la velocidad. Los actuadores hidráulicos se clasifican en actuadores lineales y rotativos.

2.1.2.1. Actuadores lineales

Se utilizará un actuador para proporcionar movimiento al robot. El cilindro es un tipo de actuador lineal formado por un tubo cilíndrico y un pistón o émbolo en su interior que se desplazará para proporcionar un movimiento mecánico lineal, se encuentran dos tipos de cilindros, los de simple acción y doble acción. En los cilindros de simple acción la presión se aplica en una sola dirección del pistón y se utiliza un resorte para regresar el pistón al estado en reposo. Se puede entender que el estudio y las características de un cilindro hidráulico son similares a la de un cilindro neumático.

Los cilindros hidráulicos y neumáticos varían con respecto al cálculo de la fuerza que pueden proporcionar cada cilindro; para el cilindro neumático, la fuerza que ejerce el aire no es la mínima que ejercería un fluido; la presión que ejerce el aire o el fluido en un cilindro determina la capacidad de fuerza del pistón y de igual manera el caudal de fluido o flujo de aire determina la velocidad del pistón en el cilindro. Se puede determinar la potencia del cilindro que en combinación con la fuerza que ejerce el cilindro y el desplazamiento del pistón se produce un trabajo. Este trabajo que se realiza en un determinado tiempo producirá la potencia que es un factor de interés. En la figura 26, se muestra el corte transversal de un cilindro con su respectiva descripción de partes.

Figura 26. **Corte transversal de un cilindro de simple acción**



Fuente: ESCALONA, Iván. *Actuadores hidráulicos*. http://www.elprisma.com/apuntes/ingenieria_industrial/actuadoreshidraulicos/default.asp. Consulta: 5 de abril de 2011.

En la figura 27, se muestra la manera de determinar la fuerza de empuje de un pistón, dada la presión de aire o de algún fluido, donde el fluido o aire actúa sobre el pistón con una presión dada. Esta presión hace que el pistón se desplace hacia afuera y hacia adentro a lo largo del cilindro. Al desplazamiento del pistón se le llama carrera, como bien se sabe, la presión del aire o fluido es igual en todas partes del recipiente y, en este caso, la presión será igual en cada unidad de superficie del pistón.

Otro dato importante para seleccionar el cilindro es considerar la velocidad de desplazamiento del mismo. En estas circunstancias, se considera el sobredimensionamiento en la carga que debe soportar y, cuanto mayor sea el sobredimensionamiento, mayor será la velocidad bajo carga. Esto significa que el cilindro debe tener mayor empuje que el que se requiere para contrarrestar la carga útil de una aplicación; donde no es de suma importancia la velocidad se sobredimensiona con un 25 % de fuerza de empuje superior a lo necesario para mantener al cilindro en estado en reposo y un 100 % para altas velocidades. Para determinar la fuerza de empuje es igual a presión de trabajo multiplicada por el área transversal del cilindro o mejor dicho el área del pistón y está dada por:

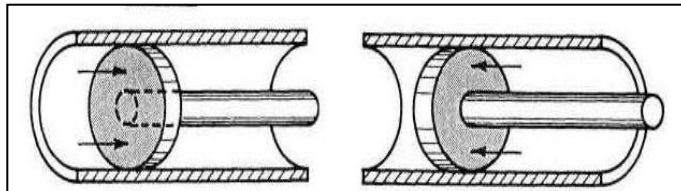
$$F(\text{kg}) = P(\text{kg}/\text{cm}^2) \cdot A(\text{cm}^2) \quad [\text{Ec. 2.1}]$$

Para determinar la velocidad de desplazamiento del cilindro, conociendo el caudal del fluido y el área transversal del cilindro o pistón, se procede de la siguiente manera:

$$\text{Velocidad}(\text{cm}/\text{minutos}) = [\text{Caudal}(\text{lt}/\text{minutos}) \cdot 1000(\text{cm}^3/\text{lt})] / \text{Área}(\text{cm}^2) \quad [\text{Ec. 2.2}]$$

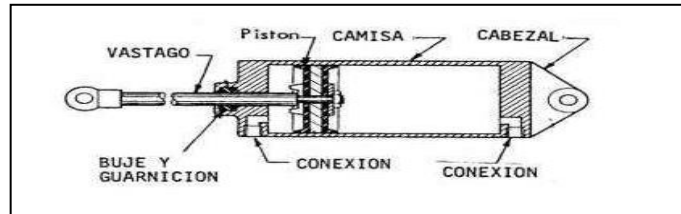
En la figura 27, se muestra la repulsión y atracción del pistón, debido a la presión del fluido o aire. Como se ve en la atracción, el área neta es igual al área del pistón menos el área del vástago. También se muestra en la figura 28 un cilindro de doble acción en el que la dirección de la presión se ejerce en dos direcciones para la repulsión y atracción del vástago.

Figura 27. **Repulsión y atracción del pistón, debido al fluido o aires comprimido**



Fuente: ESCALONA, Iván. *Actuadores hidráulicos*. http://www.elprisma.com/apuntes/ingenieria_industrial/actuadoreshidraulicos/default.asp. Consulta: 7 de abril de 2011.

Figura 28. **Corte transversal de un cilindro de doble acción**



Fuente: ESCALONA, Iván. *Actuadores hidráulicos*. http://www.elprisma.com/apuntes/ingenieria_industrial/actuadoreshidraulicos/default.asp. Consulta: 8 de abril de 2011.

2.1.2.2. Actuadores rotativos

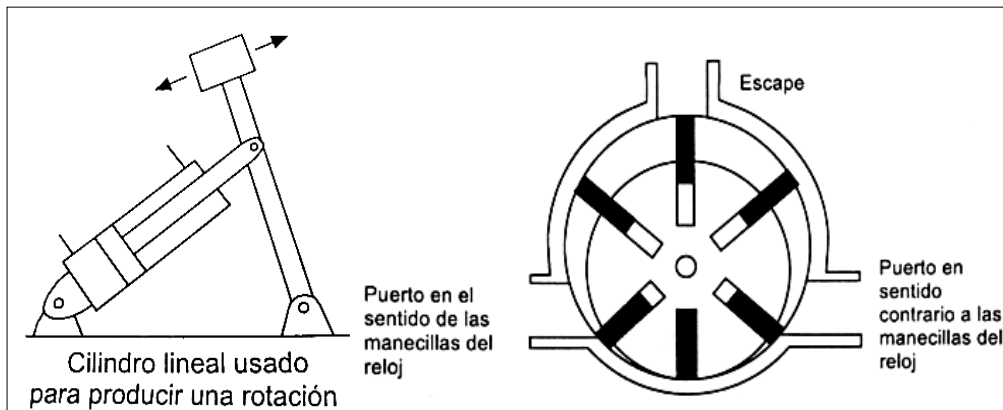
Un actuador rotatorio puede ser de tipo eléctrico, hidráulico y neumático dependiendo de la fuente de energía que se utiliza. Como su nombre lo indica un actuador rotatorio genera un movimiento giratorio del cual el valor de interés sería el torque o par que pueda generar. Los actuadores rotativos compuestos por cilindros lineales con los componentes necesarios pueden dar origen a movimientos con ángulos menores a 360° o puede ser un actuador dedicado también llamado semi giratorio compuesto por alabes que, con la diferencia de presiones, hace girar el alabe que está unido al rotor, se le puede llamar también actuador rotativo de fracción de vuelta.

En el comercio se encuentran actuadores de un cuarto de vuelta con movimientos de 0° a 90° y los actuadores multivuelta con movimientos superiores a los 360° como los motores neumáticos. Este tipo de actuador está construido por varios alabes montados en un rotor excéntrico que tiene ranuras para un movimiento hacia afuera de los alabes. Los alabes dividen el espacio dentro del motor en compartimentos separados y el tamaño de estos aumenta desde el puerto de entrada hasta el puerto de salida. El aire comprimido ingresa

en el puerto de entrada ejerciendo una fuerza en uno de los alabes haciendo girar al rotor.

Si se desea cambiar el sentido de giro se necesita introducir el aire comprimido en el sentido opuesto al émbolo en el puerto de entrada. Otro dato importante es el torque de salida del actuador, el cual es directamente proporcional a la presión de aire o fluido en el actuador. Esto no siempre sucede de esta manera; en la práctica el torque varía dependiendo de la posición del émbolo o paleta y lo que se tiene es una curva de torque en función de la posición de actuador. En la figura 29 se muestran las figuras para los tipos de actuadores rotativos.

Figura 29. **Actuadores rotativos para movimiento hasta 90°, 180° y 360°**



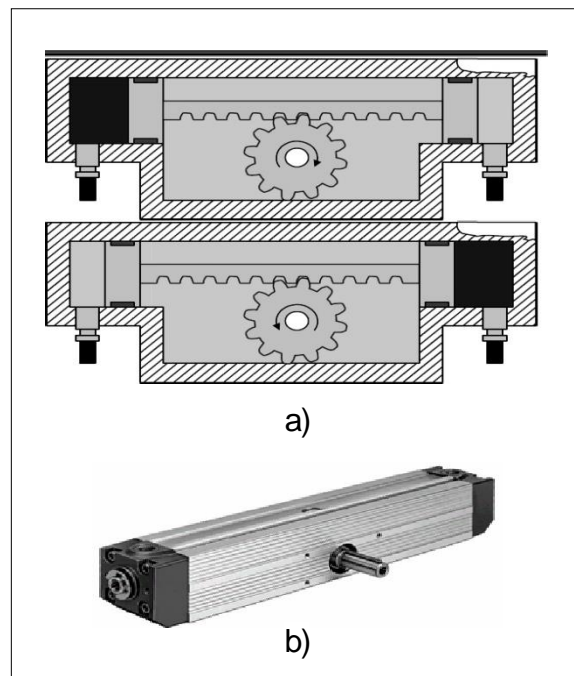
Fuente: BOLTON, William. *Mecatrónica*. p. 139.

Existen tres tipos de actuadores neumáticos:

- Piñón y cremallera.
- Yugo escocés.
- Veleta o paleta.

El actuador neumático, piñón y cremallera es un tipo de actuador rotativo que incorpora la combinación de un cilindro de doble efecto con un vástago dentado o, en otras palabras cremallera y un piñón que, al ser activado transforma el movimiento lineal en un movimiento giratorio de derecha a izquierda. Los ángulos de giro pueden ser de 45°, 90°, 180°, 290° hasta 720°. Como se ve en la figura 30, el par de giro está determinado en función de la presión del aire en la superficie del émbolo. En la práctica, se encuentran actuadores de doble cremallera que proporcionan un mayor par de giro. A continuación se muestra las figuras para un actuador neumático de piñón y cremallera.

Figura 30. **a) Actuador piñón y cremallera, b) actuador piñón y cremallera aspecto físico**

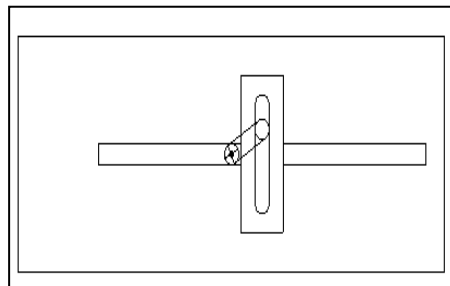


Fuente: SEAS. *Unidad didáctica 3 actuadores neumáticos*. <https://es.scribd.com/doc/2684435/ACTUADORES-NEUMATICOS#>. Consulta: 20 de abril de 2011.

El actuador yugo escocés es un tipo de actuador hidráulico o neumático formado por un vástago o de una guía que permite transformar un movimiento rectilíneo en un movimiento rotativo dado por una manivela y su árbol. También se puede obtener un movimiento en forma viceversa, como se muestra en la figura 31, si la guía o vástago se mueve a una velocidad constante, entonces el movimiento dado por la manivela será de forma rotativo constante.

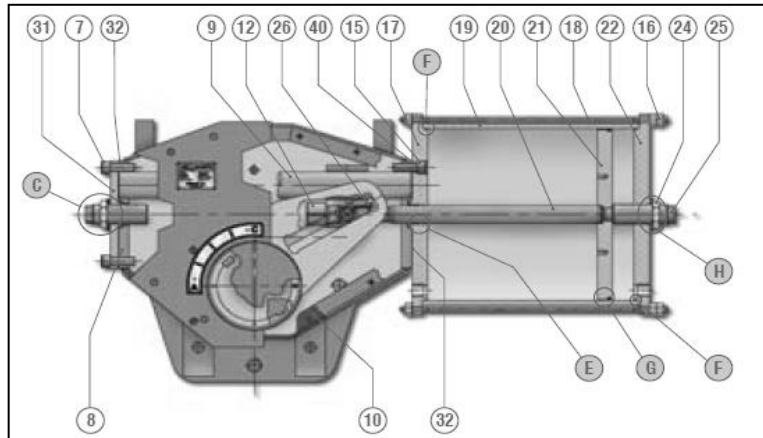
La caja del mecanismo de yugo escocés tiene una brida con agujeros roscados, según norma (ISO 5211) que sujeta el actuador a la válvula en forma directa, o si es requerido, por medio de una brida adaptadora o una caja intermedia (*mounting kit*). La palanca del actuador posee un agujero con dos chaveteros internos, el uso de un buje, adapta el agujero de la palanca a las dimensiones del vástago de la válvula. En la figura 31, se muestra un actuador yugo escocés de doble efector.

Figura 31. **Imagen representativa del yugo escocés**



Fuente: Mecanismo yugo escocés. http://es.wikipedia.org/wiki/Mecanismo_de_yugo_escoc%C3%A9s. Consulta: 14 de mayo de 2011.

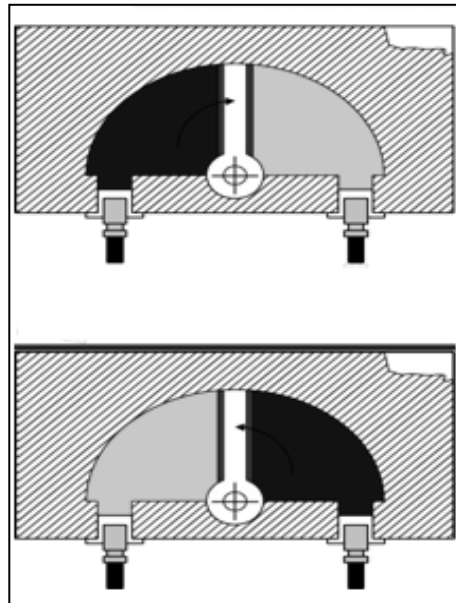
Figura 32. **Imagen física de yugo escocés de doble efecto**



Fuente: *Actuadores neumáticos y sistemas de control*. http://www.wenlen.com/pdfs/actuadores/Wenlen_actuador_neumatico_esp.pdf. Consulta: 14 de mayo de 2011.

El actuador de veleta o paleta es del tipo de actuador con movimiento giratorio limitado con ángulo de giro menor a 270° . Este actuador es neumático o probablemente hidráulico y está construido por una carcasa metálica que, en su interior, se encuentra una paleta o veleta sujeta a un eje que atraviesa la carcasa proporcionando el movimiento giratorio. En la figura 32, se observa que el movimiento angular es proporcionado debido al fluido que ingresa por uno de los puertos del actuador similar a la de un cilindro de doble efecto, el que ingresa en una de sus cámaras hace girar la paleta sobre su eje, debido a que existe una diferencia de presión con respecto a la cámara contraria, generalmente comunicado con la atmósfera y si se requiere invertir el sentido de giro basta con invertir el ingreso de fluido en el puerto posterior. En la figura 32, se presenta la figura del funcionamiento del actuador de paleta o veleta.

Figura 33. **Actuador de veleta con movimiento en ambos sentidos**



Fuente: SEAS. *Unidad didáctica 3 actuadores neumáticos*. <https://es.scribd.com/doc/2684435/ACTUADORES-NEUMATICOS#>. Consulta: 21 de mayo de 2011.

2.2. Válvulas

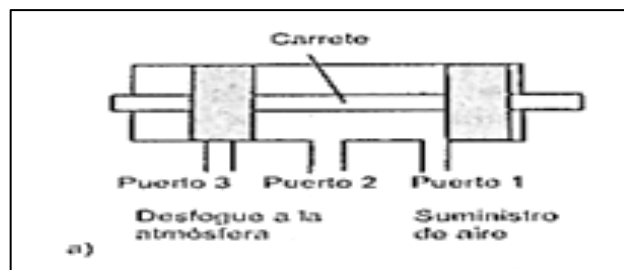
Que un sistema hidráulico funcione se necesita presurizar el aceite o fluido. Para una fuente de energía neumática se necesita someter a presión el aire o gas utilizando un compresor de aire y un enfriador, ya que el compresor aumenta la temperatura del aire por la presión. En cualquier caso, se necesita una fuente de energía que active uno o más actuadores y la utilización de válvulas que permitan el paso de aire o aceite para el control de dirección del flujo, que actúa en el actuador.

La válvula es un dispositivo que permiten o impide el paso del fluido y para cambiar la dirección de los fluidos se realiza, mediante señales eléctricas, mecánicas o fluidos a presión que controlan la dirección del fluido.

La válvula el carrete se desplaza en forma horizontal para controlar la dirección del flujo, como se mostrará en la figura 33, en la que el puerto de entrada está conectado a la fuente de energía y el tercer puerto está cerrado, permitiendo el paso del flujo hacia el puerto dos. Cuando el carrete se desplaza hacia la dirección opuesta en el puerto de entrada es bloqueado junto con el suministro del fluido.

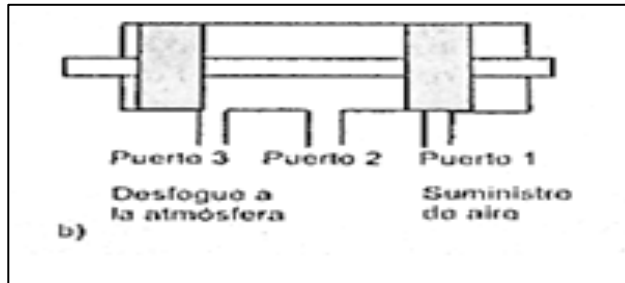
El puerto dos se encuentra activo y el paso del fluido hacia el tercer puerto; el puerto tres es un puerto de desfogue hacia la atmósfera y la presión del aire almacenado en el puerto dos sale por el puerto tres. En la figura 33, se muestra la figura de la válvula de carrete de dos posiciones.

Figura 34. **Válvula de carrete de dos posiciones, a) suministro de aires y b) desfogue de aire**



a)

Continuacion de la figura 34.

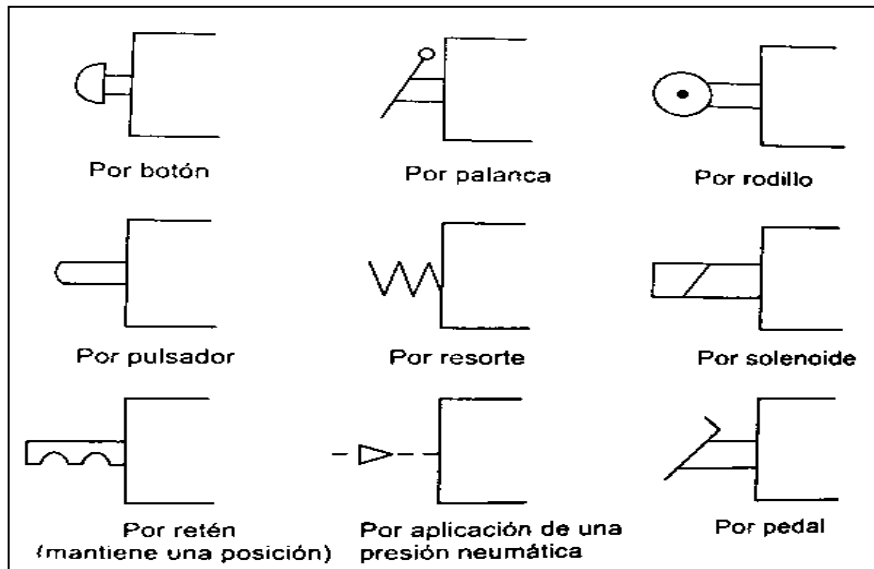


b)

Fuente: BOLTON, William. *Mecatrónica*. p. 124.

La forma de representar la relación de las posiciones y el número de puertos involucrados es de la siguiente forma, para una válvula 3/2, el número 3 representa la cantidad de puertos y el número 2 la cantidad de posiciones. Cada puerto se identificará con un número o una letra de acuerdo con la función que realiza, por ejemplo: el puerto que se identifica como 1 o P corresponde al suministro de aire o aceite, el puerto que se identifica con 3 o T para el regreso hidráulico; para el puerto que se identifica con 3 o 5 (R o S) para el desfogue y el puerto con 2 o 5 (B o A) para el puerto de salida. Una flecha (\rightarrow) indica la dirección del flujo, las líneas con tope (\perp) corresponde al puerto cerrado y no habrá flujo. Para activar una válvula se necesita una señal eléctrica o mecánica, como se muestra en la figura 34, donde se grafican las diferentes formas de accionar una válvula.

Figura 35. **Símbolos para distintas formas de activar una válvula**



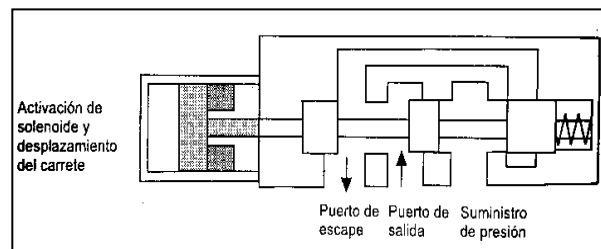
Fuente: BOLTON, William. *Mecatrónica*. p. 124.

En la figura 35, se tiene el símbolo de una válvula que contiene tres puertos y dos posiciones accionado por medio de un solenoide que mediante la circulación de la corriente, a través del devanado, creará un campo magnético que desplazará al núcleo de hierro montado al carrete. En la figura del cuadro de la derecha, se muestra el estado inicial de la válvula con el suministro de aire conectado al puerto 1 y, este a su vez, conectado a puerto 2 con la flecha que va del puerto 1 al puerto 2 indicando el flujo de aire y el puerto 3, se encuentra cerrado.

Al activarse la válvula por medio de la corriente en el solenoide, el carrete se moverá y el estado de la válvula será como se indica en el cuadro de la izquierda con el puerto 2 conectado al puerto 3 con la flecha que representa el flujo de aire y el puerto 1 en este estado se encuentra cerrado he impide el suministro de aire. Cuando la corriente deje de circular por le solenoide el

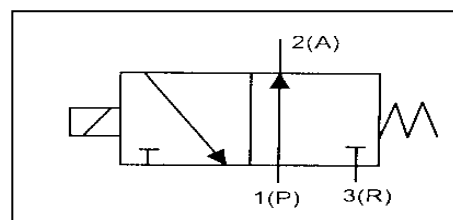
carrete regresa a la posición inicial por medio de la acción de un resorte obligando a este a regresar.

Figura 36. **Construcción física de la válvula**



Fuente: BOLTON, William. *Mecatrónica*. p. 126.

Figura 37. **Símbolo gráfico de la válvula y modo de accionamiento**



Fuente: BOLTON, William. *Mecatrónica*. p. 126.

2.3. Sensores

“Un sensor es un dispositivo electrónico o mecánica que convierte magnitudes físicas (como pueden ser luz, magnetismo, calor, presión, movimiento...) a valores medibles de dicha magnitud”³

³ TORRES, F.; POMARES, J.; GIL, P. y otros. *Robots y sistemas sensoriales*. p. 156.

Los sensores para el robot es lo mismo que los sentidos para el ser humano. Con ella el robot le permite interpretar su entorno interno como externo, por ejemplo la posición, velocidad, dirección, localización, etc. al seleccionar un sensor se tiene que tomar en cuenta el tipo de magnitud que se medirá, así como el rango de medida y las características del mismo. Las consideraciones que se tendrán, serán el tipo de señal de salida del sensor, la precisión, costo y la aplicación de destino.

El régimen permanente es un valor estático que el sensor posee:

- Rango o intervalo. Se refiere a los valores máximos y mínimos que el sensor puede medir como valor de entrada y el valor que puede proporcionar como variable de salida. Para utilizar un sensor en robótica, se necesitará medir una cantidad física y convertir ese valor en una señal eléctrica comprensible para el micro-controlador.
- Exactitud. Al referirse a un valor exacto, se refiere en cuánto un valor de medición (sensor) se desvía del valor real o un valor conocido. Este valor se encontrará muchas veces como un valor porcentual como por ejemplo: +/- X %, este valor se refiere al error máximo que se puede tener en una lectura de medición.
- Error. Se refiere a cuánto un valor medido se aleja del valor real, en otras palabras el error es la diferencia entre las señales de entrada y la señal de salida medido con algún instrumento. Este valor puede ser positivo o negativo.

$$\text{Error} = \text{Valor entrada} - \text{Valor realimentación} \quad [\text{Ec. 2.3}]$$

- Repetividad. Se refiere a la capacidad de un instrumento de proporcionar el mismo valor de salida al aplicar el mismo valor de entrada. Este valor se presenta como un valor porcentual de la capacidad de reproducción de una lectura.
- Estabilidad. Es la capacidad de un sensor de mostrar el mismo valor de salida cuando el valor de entrada permanece constante en un intervalo de tiempo.
- Resolución. Se puede explicar por resolución como la percepción de una variabilidad muy pequeña en el valor de entrada capaz de producir un valor legible y observable en el valor de salida, dicho de otra manera, es el valor de entrada más pequeña que se puede detectar.
- Sensibilidad. Es la razón de cambio en el valor de salida frente a la razón de cambio del valor de entrada, es el valor de salida por unidad de entrada.
- Valor de zona muerta. Para un intervalo de valor de entrada en el cual no hay un valor de salida o no es posible medirla.
- Histéresis. La respuesta de medida de un sensor puede ser distinta para un valor de entrada, según puede ser por un cambio de incremento o decremento formado por dispositivos eléctricos como por ejemplo, un transformador.
- No linealidad. Este valor se produce cuando los valores de respuesta se supone que es una línea recta y se producen variabilidad, este error es el valor máximo de desviación con respecto a la línea recta.

- Saturación. Es el valor de salida máxima estacionaria una vez que el valor de entrada haya superado un determinado valor sin que se pueda incrementar más el valor de salida.

Los descriptores proporcionan información para un valor de entrada dinámico de cómo responderá el sensor en su valor de salida. A continuación se definen algunos parámetros.

- Tiempo de retardo o respuesta. Es el tiempo que transcurre en alcanzar el valor de salida mayor o igual 50 % de su valor máximo después de haber aplicado un valor de entrada.
- Tiempo de ascenso. Es el tiempo en el que el valor de salida alcanza un valor de 95 % o hasta llegar su valor final, con este parámetro proporcionará información acerca qué tan rápido el sensor se establecerá en el valor de régimen estable.
- Tiempo valor pico. Se define como el tiempo en que el valor de salida del sensor alcanza su valor máximo.
- Constante de tiempo. Es un valor igual a 63,3 % del tiempo de retardo.
- Tiempo de régimen estable. Es el tiempo en que el valor de salida del sensor alcanza un valor en +/- 5 % del valor final sin salir de ella.

2.3.1. Sensor de desplazamiento lineal y rotacional

El potenciómetro como un tipo de sensor de desplazamiento lineal o rotacional permite medir la distancia o ángulo (como se verá para el

potenciómetro rotacional) de un objeto con respecto a una posición inicial variando su resistencia entre sus terminales debido al contacto deslizando a lo largo del componente. Esta propiedad de variar su resistencia lo hace ideal para trabajar con ella en articulaciones rotacionales o prismáticas, logrando con ello determinar su desplazamiento. De esta manera se obtiene una lectura de voltaje o corriente aplicando la ley de Ohm.

Como se ve en la figura 36a, para un potenciómetro con tres terminales, el potenciómetro presenta una relación lineal y a mayor resistencia en sus terminales mayor voltaje, se obtendrá en esos terminales, este presenta un divisor de voltaje entre los tres terminales; por triángulos semejantes se obtiene la siguiente relación:

$$\frac{V_{bc}}{V_{ac}} = \frac{R_{bc}}{R_{ac}} \quad [\text{Ec. 2.4}]$$

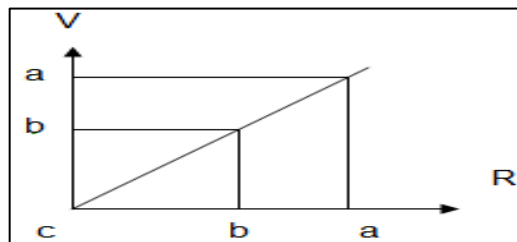
También se observa de la figura 36b, se puede llegar a una ecuación para el voltaje en V_s en función del ángulo de giro del potenciómetro y el voltaje aplicado. Esta relación está dada por el ángulo de giro y el ángulo máximo de giro; la ecuación que se obtiene es la siguiente:

$$V_s = V \cdot \left(\frac{\theta}{\theta_{max}} \right) \quad [\text{Ec. 2.5}]$$

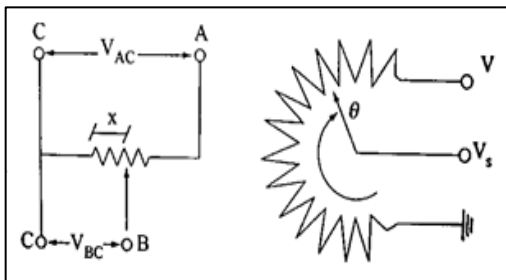
Para una mejor comprensión se menciona que también se pueden encontrar los potenciómetros del tipo logarítmico, anti logarítmico y de la combinación de ambos; pero dada su propiedad, no son útiles para aplicaciones en robótica. Para un estudio más profundo de este tipo de potenciómetros y la clase de potenciómetros que, se pueden encontrar se recomienda visitar las

direcciones electrónicas en las referencias números 16 y 17. En la figura 38, se muestra el circuito equivalente para el potenciómetro y su gráfica característica.

Figura 38. **a) Relación de voltaje a resistencia, b) circuitos equivalentes del potenciómetro**



a)



b)

Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. p. 162.

2.3.2. Sensores de proximidad capacitivos

Los sensores de proximidad capacitivos son un tipo de sensor de proximidad que permite detectar la presencia de un objeto a una distancia determinada como la proximidad de un objeto situado a una distancia de algunos cientos de milímetros, el principio de funcionamiento se base esta determinado por el del capacitor de placas paralelas en la que la capacitancia C

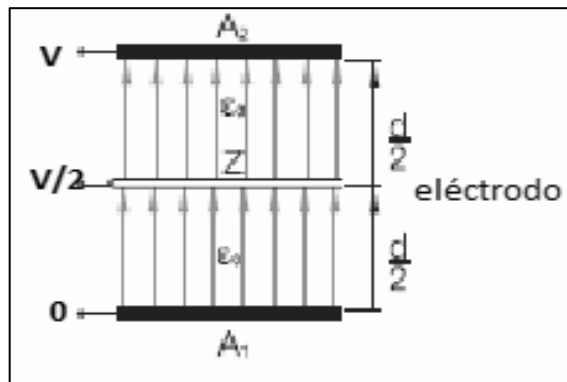
está determinada por la distancia entre las placas, el área de las placas y el dieléctrico entre las placas que posee una permitividad relativa.

Este tipo de sensor está diseñado para detectar objetos aislantes como papel, plástico y madera; también puede detectar materiales metálicos, por medio de variación de capacitancia. Para comprender el funcionamiento se inicia con la característica que se tienen cuando se aplica un voltaje a un par de placas conductoras separadas a una distancia d . Estas placas empezarán a cargarse una carga $q+$ para la placa conectada a la terminal positiva y una carga $q-$ para la terminal negativa. Estas cargas crearán un campo magnético que, a su vez, generará un voltaje entre las terminales. En la ecuación 2,6, se presenta la ecuación de capacitancia para el capacitor de placas paralelas.

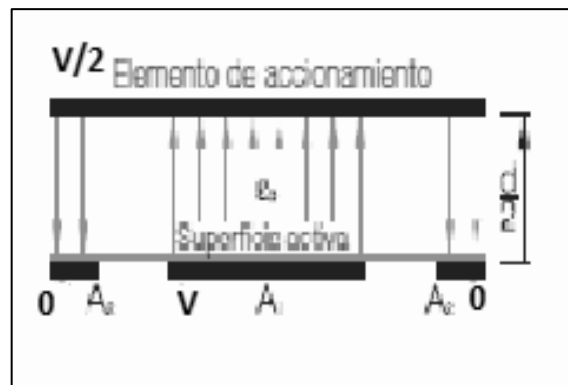
$$C = K_e \cdot \epsilon_0 \cdot \frac{A}{d} \quad [\text{Ec. 2.6}]$$

Donde K_e es la constante dieléctrica del material que se coloca entre las placas, ϵ_0 es la constante dieléctrica del espacio igual a $8,8541878 \times 10^{-12}$, A es el área de traslape de las placas y d la distancia de separación entre las placas. Como puede observarse. Haciendo variar cualquier de los tres componentes del capacitor se obtendrán diferentes capacitancias, para el sensor se podrá medir la proximidad hacia el objeto a censar haciendo variar su capacitancia por medio del cambio de distancia del sensor hacia el objeto o por medio del dieléctrico del objeto a censar.

Figura 39. a) El condensador dividido por el electrodo de accionamiento, b) el condensador como sensor capacitivo y el efector del elemento accionamiento



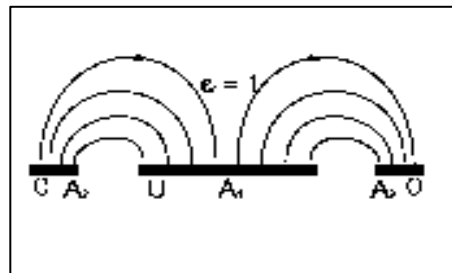
a)



b)

Fuente: *Sensores capacitivos*. http://www.balluff.com/NR/rdonlyres/2DF2E2D2-1935-4788-BD13-5A3C875B9FD0/0/870041_41_0811_ES.pdf. Consulta: 3 de julio de 2011.

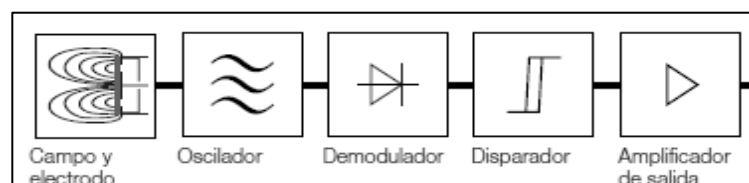
Figura 40. **El condensador como sensor de elementos no metales**



Fuente: *Sensores capacitivos*. http://www.balluff.com/NR/rdonlyres/2DF2E2D2-1935-4788-BD13-5A3C875B9FD0/0/870041_41_0811_ES.pdf. Consulta: 4 de julio de 2011.

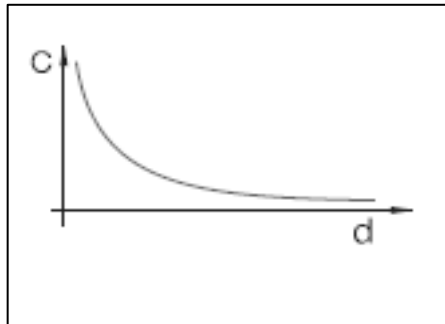
Los componentes del sensor capacitivo son, como se muestra en la figura 38. El sensor está formado por dos placas concéntricas como ya se mencionó y, cuando un objeto se aproxima al área sensible, el sensor capacitivo cambia su capacitancia desde 0 del oscilador. El oscilador empieza a oscilar activando el circuito demodulador eliminando las oscilaciones y este, a su vez, activa el circuito disparador y, cuando el objeto se va alejando, el circuito regresa a un valor de salida de cero. La profundidad a la cual el sensor puede detectar un objeto está determinada como la fracción del cambio de la capacitancia por la capacitancia inicial el cual se llamará área de sensible.

Figura 41. **Diagrama de bloques del sensor**



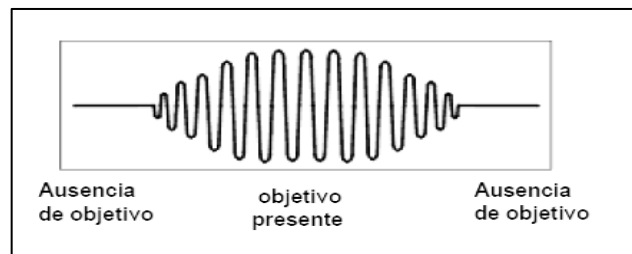
Fuente: *Sensores capacitivos*. http://www.balluff.com/NR/rdonlyres/2DF2E2D2-1935-4788-BD13-5A3C875B9FD0/0/870041_41_0811_ES.pdf. Consulta: 8 de julio de 2011.

Figura 42. **Capacitancia del sensor en función de la distancia**



Fuente: *Sensores capacitivos*. http://www.balluff.com/NR/rdonlyres/2DF2E2D2-1935-4788-BD13-5A3C875B9FD0/0/870041_41_0811_ES.pdf. Consulta: 8 de julio de 2011.

Figura 43. **Salida del oscilador en presencia del objeto**



Fuente: CANTO, Carlos. *Sensores de proximidad capacitivos*. http://galia.fc.uaslp.mx/~cantocar/automatas/PRESENTACIONES_PLC_PDF_S/25_SENORES_CAPACITIVOS.PDF. Consulta: 9 de julio de 2011.

2.3.3. **Sensores de proximidad inductivos**

La inductancia y el campo magnético que produce una bobina se puede utilizar como sensor de desplazamiento, posición o proximidad. El transformador diferencial como sensor de desplazamiento utiliza la inductancia mutua para detectar el movimiento producido por el núcleo ferroso del

transformador, a fin de medir el voltaje inducido en unas bobinas colocadas en serie en los extremos de la bobina primaria; en la bobina primaria se alimenta con una corriente alterna y esta a su vez es transformada por una corriente alterna en las bobinas secundarias de igual característica colocadas de manera que sus salidas se oponen entre sí.

Estas bobinas que se enrollan sobre un núcleo ferromagnético que en condiciones iniciales, el núcleo se encuentra dentro de la bobina primaria y secundaria, de la suposición que la energía que se tiene en la entrada es igual a la salida. El voltaje en las bobinas secundarias sería la misma; por lo tanto, el voltaje en dichas bobinas se cancelará, ya que se encuentran conectadas de manera que sus salidas sean opuestas y el voltaje neto es a cero.

Si se desplaza el núcleo de forma que una de las bobinas secundarias no cubre completamente el núcleo, el campo magnético no atravesará completamente a una de estas bobinas y el voltaje inducido será mayor en una bobina secundaria que en la otra; por lo tanto, el voltaje neto será diferente de cero. El voltaje en la entrada es igual a $v = V \sin(\omega t)$ y el voltaje de salida está dado como $v_1 - v_2$ donde $v_1 = k_1 \sin(\omega t + \phi)$, $v_2 = k_2 \sin(\omega t + \phi)$ donde k_1 , k_2 y ϕ dependen del acoplamiento entre el primario y secundario, utilizando la suma de senos y con la condición de que el desplazamiento del núcleo es mayor en la bobina secundaria 1 que en 2 se tiene $k_1 > k_2$.

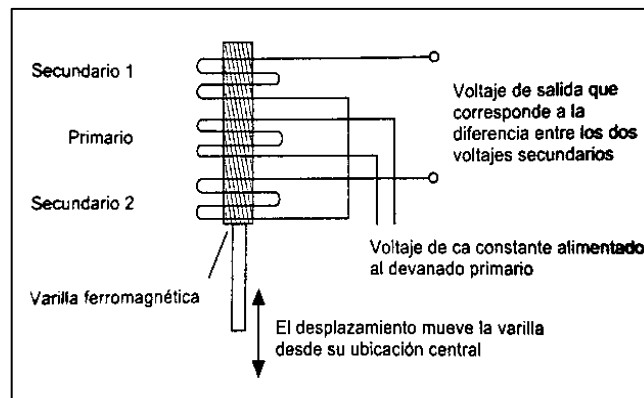
$$V_{salida} = (K_1 - K_2) \cdot \sin(\omega t + \phi) \quad [\text{Ec. 2.7}]$$

Cuando el desplazamiento del núcleo en la bobina secundaria 2 es mayor que en 1 se tiene $k_1 < k_2$ y utilizando las propiedades de los senos se tiene:

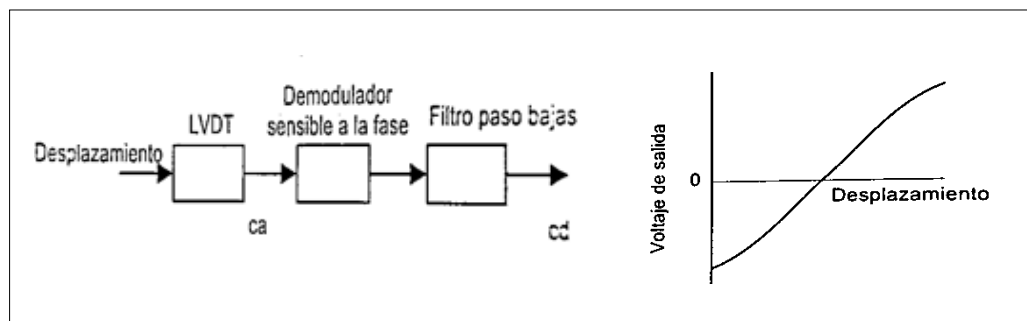
$$V_{salida} = -(K_1 - K_2) \cdot \sin(\omega t + \phi) = (K_1 - K_2) \cdot \sin(\omega t + (\pi + \phi)) \quad [\text{Ec. 2.8}]$$

De la ecuación se observa que, para un desplazamiento del núcleo hacia una de las bobinas secundarias, se tendrá un desfase en el voltaje de salida. En la figura 44, se muestra al sensor, y en la figura 44b se muestra el diagrama de bloques para el transformador diferencial como sensor de desplazamiento LVDT.

Figura 44. **a) El sensor LDVT, b) diagrama de bloques del sensor con voltaje de salida dc y su gráfica de respuesta**



a)



b)

Fuente: BOLTON, William. *Mecatrónica*. p. 28.

El sensor de proximidad inductiva por corrientes de Foucault, funciona debido al efecto que se produce cuando se introduce un conductor o placa metálica dentro de un campo magnético variable en el tiempo. De la ley de Lenz establece que, “el sentido de la corriente inducida sería tal que su flujo se opone a la causa que la produce (en este caso el campo magnético variable) [y del fenómeno eléctrico descubierto por el físico francés León Foucault en 1851. Se produce cuando un conductor atraviesa un campo magnético variable, o viceversa. El movimiento relativo causa una circulación de electrones o corriente inducida dentro del conducto; estas corrientes circulares de Foucault crean electroimanes con campos magnéticos que se oponen al efecto del campo magnético aplicado”.⁴

“Cuanto más fuerte sea el campo magnético aplicado, o mayor la conductividad del conductor, o mayor la velocidad relativa de movimiento, mayores serán las corrientes de Foucault y los campos opositores generados”.⁵ El sensor inductivo consta de una bobina al que se le aplica una corriente alterna, y esta a su vez crea un campo magnético variable en el tiempo, al acercarse un objeto metálico al campo magnético generado, debido al efecto Foucault; mientras ley de Lenz creará un campo magnético que se opondrá al campo magnético que lo origina distorsionándolo y modificando la corriente en la bobina haciendo variar la impedancia de la misma.

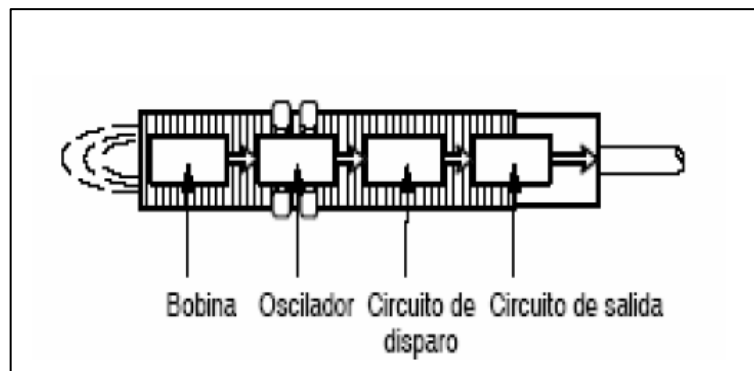
El sensor inductivo que se muestra en la figura 45, está compuesto de una bobina con núcleo de ferrita, un oscilador, un circuito de disparo y un circuito de salida. El núcleo de ferrita dirige el campo magnético oscilante producido por el oscilador asegurando que solo exista el campo en la parte frontal del sensor, las

⁴ MORALES, Gregoria y MARTÍNEZ, Cruz. *Métodos de programación de robots* <https://es.slideshare.net/cruzmartinez9883/mtodos-de-programacin-de-robots>. Consulta: 17 de julio de 2011.

⁵ BELU. *Robótica* <http://robotiica.blogspot.com/2007/10/historia-de-la-robotica.html>. Consulta: 11 de mayo de 2011.

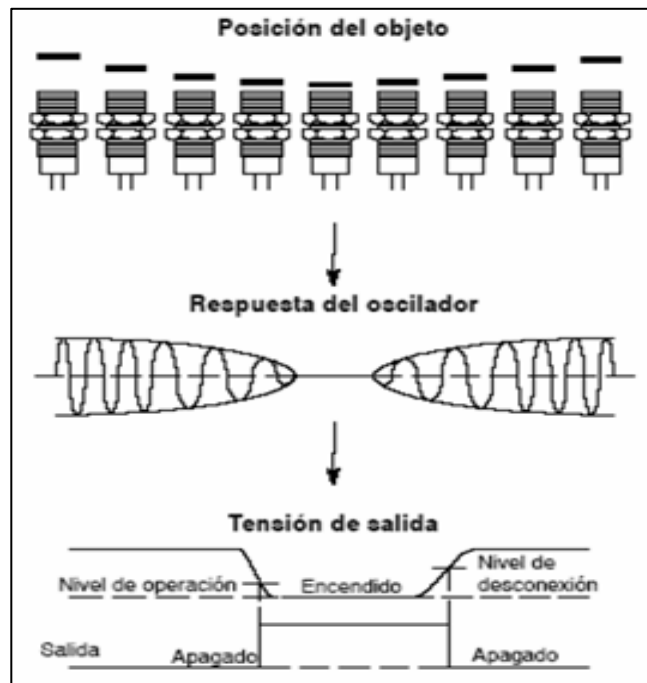
corrientes parásitas causadas por el campo oscilante extraen energía del campo magnético y cuanto más cerca esté el objeto del sensor, más energía extraerá causando un atenuación muy considerable que el circuito disparador no podrá sostener el nivel de disparo y la señal de salida será igual a cero. A continuación en la figura 45, se muestran las figuras del sensor y en la figura 46 la forma de operación del sensor.

Figura 45. **Diagrama de bloques del sensor**



Fuente: FERNÁNDEZ, Gerardo. *Sensores magnéticos e inductivos*. <http://www.uaeh.edu.mx/docencia/Tesis/icbi/licenciatura/documentos/Sensores%20magneticos.pdf>. Consulta: 17 de julio de 2011.

Figura 46. **Forma de operación del sensor**



Fuente: FERNÁNDEZ, Gerardo. *Sensores magnéticos e inductivos*.

<http://www.uaeh.edu.mx/docencia/Tesis/icbi/licenciatura/documentos/Sensores%20magneticos.pdf>. Consulta: 17 de julio de 2011.

2.3.4. **Encóder**

Otro sensor de desplazamiento es el encóder y su funcionamiento es tal que, de un desplazamiento lineal o rotacional se obtiene una señal digital incremental o absoluta de desplazamiento. Los dos tipos de encóders dependen de su forma de operación que pueden ser ópticos o magnéticos, el encóder incremental genera una serie de pulsos ideal para determinar la posición o la velocidad por medio del conteo de pulsos y el encóder absoluto genera varios bits de información por cada desplazamiento.

El encóder óptico está compuesto por un fuente de luz (diodo led infrarrojo), un foto receptor (foto led o foto transistor) infrarrojo y un disco de plástico o metal perforado con una o dos tiras de perforación para el encóder de incremento y con varias tiras de perforación que corresponde a un bit de dato por tira de perforación para el encóder absoluto. Y, por último, se tiene los encóders magnéticos que utilizan la reluctancia variable para detectar los cambios de los campos magnéticos, por la presencian de un objeto ferro magnético.

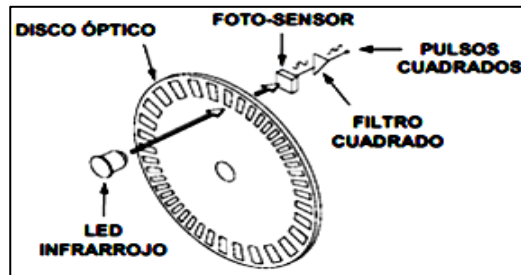
Un encóder incremental utiliza un tren de pulsos para detectar el desplazamiento. Estos pulsos están equitativamente espaciados en todo el disco y tiene cierta cantidad por cada revolución o por centímetro de desplazamiento. Si se necesita solamente el conteo de desplazamiento sin conocer la dirección, el sensor tendrá solamente un canal de salida, pero cuando es necesario conocer la dirección tendrá dos canales de salida desfasadas 90° eléctricos utilizando la salida de cuadratura. El encóder absoluto genera salida de datos digitales proporcionando la posición, velocidad y dirección, los datos de salida pueden estar en código binario o gray.

Figura 47. **El encóder básico**



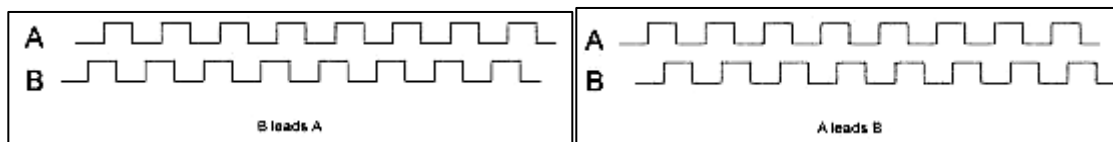
Fuente: *Manual de aplicación de encóder*. <http://www.westmexico.com.mx/pfd/dynapar/catalogos/4.-anual%20de%20Aplicacion%20de%20Encóders.pdf>. Consulta: 24 de julio de 2011.

Figura 48. **Diferentes tipos de encóder**



Fuente: *Diseño básico de un encóder*. <http://sensoresyactuadores2013.blogspot.com/2013/07/disen0-basico-de-un-encoder.html>. Consulta: 24 de julio de 2011.

Figura 49. **Señal de cuadratura en el encóder incremental**



Fuente: *Diseño básico de un encóder*. <http://sensoresyactuadores2013.blogspot.com/2013/07/disen0-basico-de-un-encoder.html>. Consulta: 24 de julio de 2011.

2.3.5. **Sensores ultrasónicos (sensor de posición)**

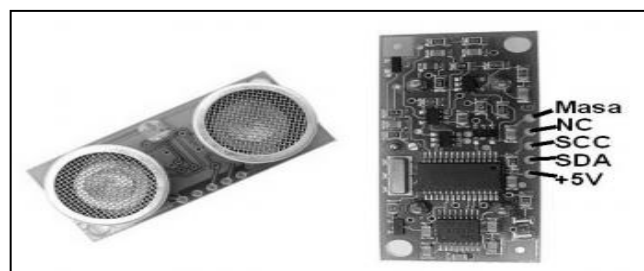
Un sensor ultrasónico utiliza ondas por arriba de la audición humana, evita que el ser humano pueda percibir las. Esta onda viaja por el aire a una velocidad aproximadamente de 340 m/s y sí, en el medio en que viaja encontrase algún objeto, un porcentaje de esta onda o su totalidad es reflejada hacia la fuente y al medir el tiempo que tarda la onda en salir, hasta llegar a la fuente, se determinará la distancia de recorrido de la onda; así con ello la distancia que hay desde la fuente hasta el objeto.

Este sensor presenta algunos problemas de medición, como por ejemplo, a distancias grandes presenta error de medición o medición nula. El ángulo de incidencia produce lecturas erróneas, las superficies lisas producen el mismo problema. Otro inconveniente que se debe que tomar en cuenta es la velocidad del aire en que se utilice el sensor, ya que el sonido utiliza ondas de presión para viajar en el medio y al ser distorsionadas por el aire, éstas se superponen a la onda que produce el aire. La temperatura es otro factor, ya que cambia la densidad del aire y con ella la velocidad de propagación.

La distancia del sensor puede detectar, está determinada por una distancia mínima en que el circuito interno se sincronice y la máxima en que la potencia de la señal reflejada sea percibida por el sensor. La acción de detección del sensor es cónica, esto es, que el sensor no puede detectar objetos a cierto ángulo de aplicación. Fuera de este ángulo, no habrá ninguna onda reflejada o producirá error en la medición, limitándola solamente a detecta objetos situados frente a ella.

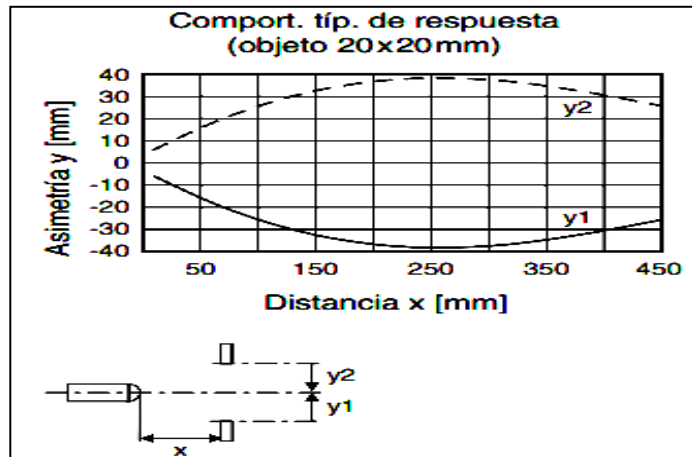
Los sensores encontrados en el mercado son: módulo PING, SRF04, SRF05, SRF08, SRF10, entre otros y los modelos LSU, RKU, HRTU Y VRTU.

Figura 50. **a) Sensor SRF8 y módulo de conexión, b) rango de operación del sensor RKU8**



a)

Continuación figura 50.



b)

Fuente: PÉREZ, Diego. *Sensores de distancia por ultrasonido*.

http://www.leuze.com/downloads/los/03/Pb_BR_USDS_es.pdf. Consulta: 6 de agosto de 2011.

2.3.6. Sensor de contacto

Es un tipo de sensor que tiene dos estados: el de encendido y el apagado, y se necesita contacto físico acompañado de una pequeña fuerza para activarlo. Por su operación tan simple, se puede utilizar como sensor de contacto o proximidad y los modos de activación del interruptor pueden ser: por palanca, por rodillo o por activación por levas. Este sensor se puede dividir en dos categorías: binarios y analógicos; también se puede utilizar para obtener información del efector final y el objeto a manipular.

Los sensores de contacto más simples corresponden a dos estados: el de encendido y apagado. Este es del tipo de sensor binario, por ejemplo, un micro interruptor que determina si un objeto está presente entre el manipulador del

brazo robótico o por deslizamiento sobre el objeto que determina su presencia por activación por rodillo.

El tipo de sensor de contacto analógico proporciona una señal eléctrica analógica proporcional a una fuerza que actúa sobre el sensor. En su forma más simple. Este sensor está constituido por un resorte unido mecánicamente a una varilla sobre un eje giratorio (para fuerzas laterales) y utilizando un potenciómetro o un encóder se obtiene la señal de información. Cuando actúa una fuerza sobre la varilla esta hace girar al resorte de manera proporcional por la fuerza aplicada y crecimiento de la constante del resorte proporciona la fuerza que corresponde a un desplazamiento dado.

Figura 51. **Sensor de contacto comercialmente disponible**



Fuente: MARTÍN, Jorge. *Tecnología de control*. <https://sites.google.com/site/jalumaco/sensores/sensores-de-contacto>. Consulta: 10 de agosto de 2011.

2.3.7. Sensores de velocidad

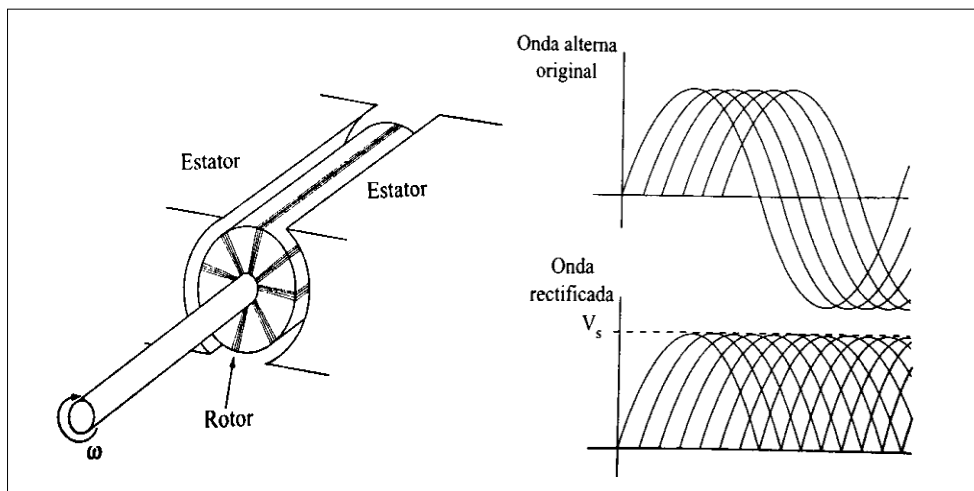
Un sensor de velocidad que se propone en la implementación es el tacogenerador. Este dispositivo permite medir la velocidad angular convirtiendo la rotación a un valor de voltaje proporcional en sus terminales. Este sensor es prácticamente un motor de corriente continua funcionando de manera inversa,

donde se tiene una bobina dentro de un campo magnético que, cuando gira, proporciona un voltaje entre sus terminales. Este valor de voltaje es proporcional a la velocidad de giro del dispositivo, con ello proporciona una señal alterna que es posteriormente rectificadora. La ecuación del voltaje de salida es la siguiente:

$$V = B_f \cdot S \cdot \sin(\omega t) \quad [\text{Ec. 2.9}]$$

Donde V es el voltaje que se obtiene de la salida del dispositivo; B_f es el campo magnético proporcionado por el estator; S es la superficie de la bobina o devanado y $\sin(\omega t)$ es la ecuación que modela el voltaje proporcionado por el sensor. En la figura 52, se muestra un tacogenerador con n devanados y $2n$ contactos, la forma de onda del devanado y la forma de onda de salida del tacogenerador.

Figura 52. **a) Tacogenerador con n devanado y la forma de onda de voltaje de salida, b) tacogenerador dc comercialmente disponible**



a)

Continuacion figura 52.



b)

Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. <http://www.directindustry.es/prod/baumer/tacogeneradores-dc-179-846563.html>. Consulta: 20 de agosto de 2011.

2.3.8. Sensor de fuerza

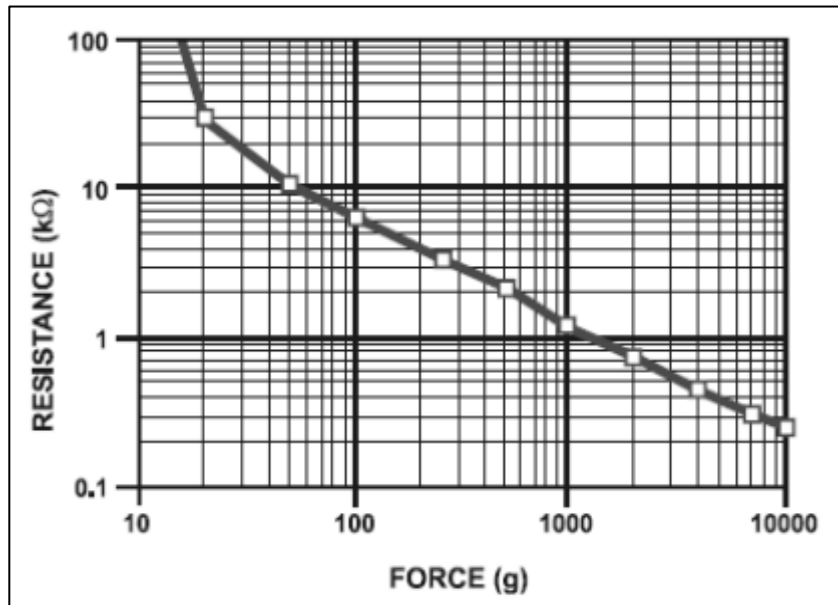
Este tipo de sensor, cuando se utiliza en robótica industrial, se dispone entre las pinzas o en el efector final para determinar la cantidad de fuerza que el efector final ejerce sobre el objeto a manipular. Un tipo de sensor de fuerza resistivo es el *Flexi Force* que varía su resistencia cuando actúa una fuerza sobre el área activa del sensor. Este sensor se encuentra dentro una membrana muy delgada capaz de medir fuerzas estáticas como dinámicas soportando fuerzas de hasta 1000 lbf, este sensor varía su resistencia de área sensorial de forma inversamente proporcional a la fuerza aplicada.

Este tiene una resistencia en reposo o de ruptura de 100 k Ω ; además puede llegar hasta menos de 1 k Ω . La respuesta del sensor, debido a la fuerza aplicada se representará en forma logarítmica (log/log). Como se puede

observar en la figura 46^a, la respuesta del sensor tiende a ser lineal en la gama dinámica y al final la respuesta se desvía a la ley 1/R llegando hasta un punto de saturación, donde el aumento de fuerza ya no se tiene disminución de resistencia, la fuerza de saturación para este sensor es de 10 kg a una presión 100 a 200 psi.

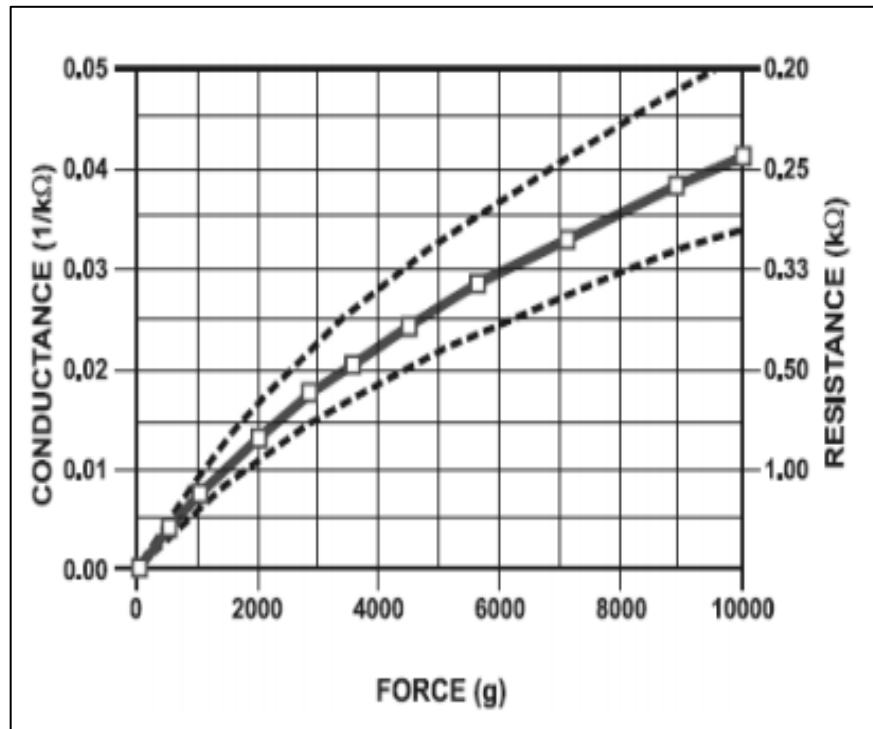
En la figura 46b, se representa la conductividad (1/R) del sensor y, como puede observarse, la respuesta del sensor es casi lineal. Esta respuesta se puede lograr con un circuito convertidor de corriente a voltaje. Este circuito proporciona un voltaje directamente proporcional a la conductividad. A continuación, se presentan las figuras de las respuestas a la resistividad y a la conductancia debido a la fuerza aplicada al sensor.

Figura 53. **a) Gráfica fuerza vs. resistencia, b) gráfica fuerza vs. Conductancia**



a)

Continuación figura 53.



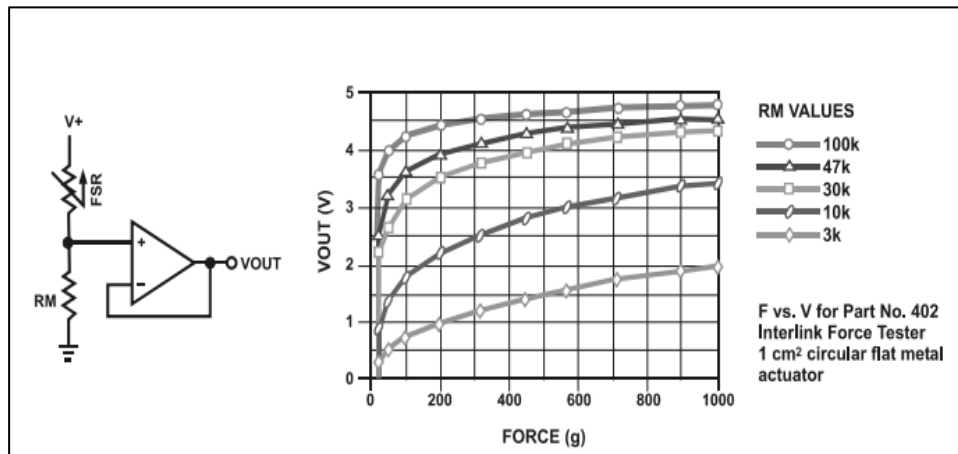
b)

Fuente: CARRIÓN, Luis y otros. *Análisis del funcionamiento del sensor de fuerza resistivo (FSR) con LabView*. http://www.datalights.com.ec/site2/images/stories/robotica/nap/nap_fsr.pdf.

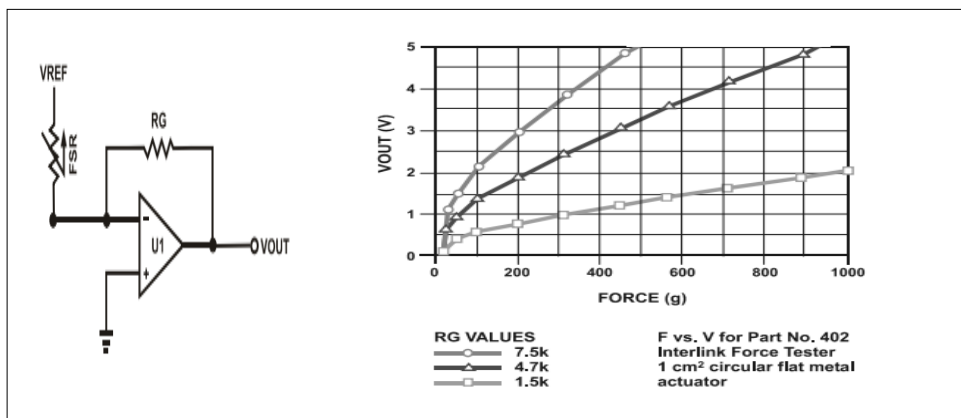
Consulta: 28 de agosto de 2011.

En la figura 54, se presenta algunos circuitos para el FSR.

Figura 54. a) Circuito básico divisor de voltaje y gráfica de respuesta para diferentes valores de RM, b) circuito convertidor de corriente a voltaje y gráfica de respuesta para diferentes valores de RG



a)



b)

Fuente: *interlink electronics fsr force sensing resistors*. http://www.trossenrobotics.com/productdocs/FSR_Integration_Guide.pdf. Consulta: 30 de agosto de 2011.

2.3.9. Sensor de aceleración

En robótica industrial, es interesante conocer la aceleración que se someterá cualquier componente del robot, por los diversos cambios de fuerza que soportará al transportar una carga útil en el extremo del robot. Una manera útil y sencilla de realizar un acelerómetro es a partir de un sensor de velocidad colocándole a la salida un circuito derivador formado por amplificadores operacionales, así se deriva la velocidad respecto al tiempo se obtendrá una señal que indique la aceleración a partir de la velocidad.

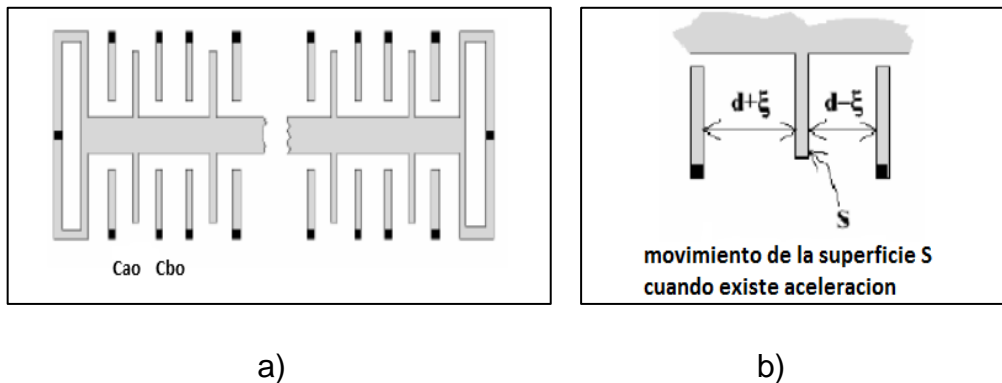
Un sensor capaz de medir directamente la aceleración es el acelerómetro basado en capacitancia. Este sensor está formado por varios micros capacitores montados en un chip de silicio y al ser sometido a una aceleración, la posición relativa de las placas varía y, con ello, varía la capacitancia existiendo entre las placas algún dieléctrico, para obtener la mayor variación de capacidad posible.

La forma de funcionamiento es la siguiente: Internamente se encuentran capacidades fijas formadas por placas separadas a una distancia “d” montadas en el chip y, entre estas placas ancladas, existen un conjunto de placas centrales montadas en un anillo elástico que permite el movimiento de las mismas.

Cuando el sensor está en reposo, las placas centrales se encontrarán en medio y, cuando exista una aceleración, las placas se moverán hacia una de las fijas, con ello provoca que en una aumente y la otra, disminuya. De esta manera, se obtiene una capacidad relativa. En la figura 55, las placas oscuras son las capacidades ancladas al circuito integrado (como C_{ao} y C_{bo}); mientras

que las placas centrales son las que se encuentran en el centro de dos capacidades fijas.

Figura 55. a) Sistema de placas montadas en el chip de silicio, b) movimiento de la superficie conductora "S"

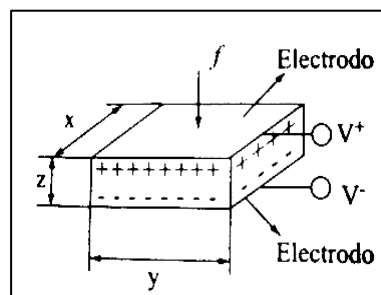


Fuente: BOUZIANE, Hrich. *Diseño e implementación de un acelerómetro, velocímetro móvil digital controlado por un PIC de MICROCHIP*. <http://upcommons.upc.edu/bitstream/handle/2099.1/7998/Mem%c3%b2ria.pdf?sequence=1&isAllowed=y>. Consulta: 18 de septiembre de 2011.

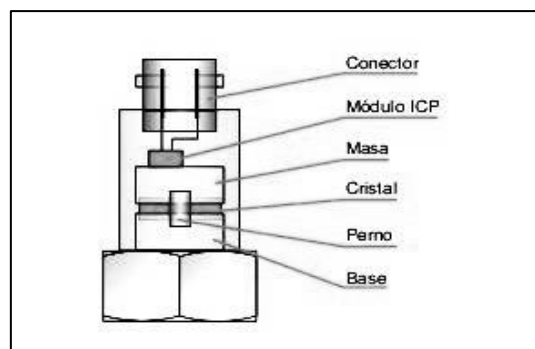
Otro sensor de aceleración es el formado por materiales piezoeléctricos. Estos materiales tienen la propiedad que, cuando sufren alguna deformación física cambia su estructura cristalina y, en consecuencia, sus propiedades eléctricas generando una pequeña cantidad de energía eléctrica. Cuando a una pieza piezoeléctrica se le aplica una fuerza, las cargas positivas se concentrarán en un lado y en el lado opuesto las carga negativas. De esta manera, se produce un voltaje entre sus terminales, con ello también cambia la capacitancia de la pieza. Como se verá en la figura 55, cuando se somete una fuerza f , las cargas se moverán y con ello se producirá una capacitancia c .

Para medir la aceleración, a partir de la fuerza se tiene que $f = m \cdot a$, el material piezoeléctricos se encuentran comprimidos por una masa, sujeta al otro lado, por un muelle dentro de una cubierta metálica. Cuando el material piezoeléctrico se somete a una fuerza proporcional a la aceleración de la masa, debido al efecto piezoeléctrico se creará un voltaje variable que será proporcional a la aceleración. En la figura 56, se muestra el efecto que tiene la fuerza sobre el material piezoeléctrico y un acelerómetro piezoeléctrico.

Figura 56. **a) Efecto de la fuerza sobre el material piezoeléctrico, b) acelerómetro piezoeléctrico**



a)



b)

Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. http://www.sinais.es/sensores/transductores_piezoelectricos.html. Consulta: 2 de octubre de 2011.

2.3.10. Sensor de inclinación (inclinómetro)

Es un sensor que proporcionan un valor de voltaje o corriente para cada valor de ángulo de inclinación, con respecto a la horizontal que va desde $\pm 1^\circ$ hasta $\pm 90^\circ$. Estos sensores poseen una sensibilidad que pueden alcanzar una resolución de salida de 0.0001° que, dependiendo de la tecnología y la gama de ángulo puede llegar a 0.01° . El sensor que comercialmente se puede encontrar es el de la serie AIT435 'L' es una familia de un solo eje, de alta precisión, de circuito cerrado. La gravedad hace referencia inclinómetros del tipo servo y donde el espacio es bien escaso. Disponible en gamas de $\pm 1^\circ$ a $\pm 90^\circ$, con un factor de forma que mide sólo 61 mm (2.41 pulgadas) de altura y un diámetro de menos de 37 mm (1,46 pulgadas); las características se listan a continuación:

- Los márgenes de $\pm 1^\circ$ a $\pm 90^\circ$.
- Resolución de 0.1arc segundo ($0,00003^\circ$).
- Alto nivel de salida de 4 a 20 mA proporcional al seno del ángulo de inclinación.
- Terminaciones eléctricas a través del conector de seis vías (LSOC) o pines de soldadura (LSOP).
- El medio ambiente índice de protección IP65.
- El funcionamiento continuo a temperaturas de -18°C a $+70^\circ\text{C}$.
- Robusto, 1500 g de choque de supervivencia.
- Diseño personalizado y capacidades de fabricación están disponibles para cumplir con determinados requisitos del cliente.

2.4. Controladores

Un controlador en robótica se refiere al circuito electrónico que se encarga de proporcionar las órdenes y regular cada movimiento del robot, así como las acciones y el proceso de la información en la toma de decisiones. Este también tiene la capacidad de almacenar el programa de instrucción (instrucciones que ejecuta el controlador almacenado en una memoria externa, se puede también almacenarse en una memoria interna como para el caso del microcontrolador); asimismo tiene periféricos de entrada/salida para comunicarse con otros dispositivos o maquinaria de trabajo dedicado. Entre los tipos de controladores, dependen de la función a la que se le puede someter o, en otros casos, un solo controlador puede ejercer el control total:

- Posición: el controlador interviene únicamente en el control de la posición del elemento terminal.
- Cinemático: en este caso el control se realiza sobre la posición y la velocidad.
- Dinámico: además de regular la velocidad y la posición, controla las propiedades dinámicas del manipulador y de los elementos asociados al controlador.
- Adaptativo: se ocupa de controlar la variación de las características del manipulador al variar la posición.

Las funciones de un controlador se basan en proporcionar órdenes al sistema actuador para proporcionar movimiento a las articulaciones del robot. Además, se encarga de la toma de decisiones y comparar una fracción de la señal de salida por medio de los distintos sensores instalados en el robot con parámetros establecidos en la entrada o desde el programa de instrucciones. El controlador, los actuadores, los sensores y la señal retroalimentación

proporcionada por los sensores forman un sistema de control de bucle cerrado capaz de ejercer funciones de forma precisa y con el mínimo error posible.

Este trabajo no enfocará su atención en un controlador si no en un microcontrolador que, a diferencia del controlador que necesita de puertos de E/S, y memoria externa para el almacenamiento de programa (memoria ROM) y datos (memoria RAM), el microcontrolador es un dispositivo que dentro de un mismo chip se encuentran los puertos de E/S, para el almacenamiento del programa (memoria EEPROM) y datos (memoria RAM).

3. FUNCIONES DE TECNOLOGÍA

3.1. Análisis de los movimientos

En el capítulo tres se obtienen conceptos de los capítulos tres y cuatro del libro robots y sistemas sensoriales de F. Torres y otros. El análisis se realizará mediante un vector de posición \mathbf{p} , para localizar un objeto en el espacio con componentes x , y , z . El origen del vector estará en el origen del sistema de referencia M y el extremo del vector localizará el origen del sistema O con representación \mathbf{p}_O^M (posición del origen del sistema O respecto al sistema M). Para representar la orientación de un cuerpo en el espacio, se tomará la orientación relativa que existe entre el origen del sistema O que representa la posición del objeto y el sistema de referencia M , para mejor comprensión y análisis de la orientación relativa de ambos sistemas serán coincidentes en el origen.

Para comprender la orientación que tendrá el sistema O respecto del sistema de referencia M , se utilizará una matriz de rotación. La matriz de rotación expresa cada vector unitario \mathbf{x}_O , \mathbf{y}_O , y \mathbf{z}_O del sistema O en una proyección sobre los ejes del sistema M . Con esto se obtiene el producto punto de cada vector unitario del sistema O con cada vector unitario del sistema M llegando a la expresión correspondiente que representa la orientación del sistema O respecto al sistema M .

$${}^M Rot_O = [x_O^M \cdot y_O^M \cdot z_O^M] = \begin{bmatrix} x_O \cdot x_M & y_O \cdot x_M & z_O \cdot x_M \\ x_O \cdot y_M & y_O \cdot y_M & z_O \cdot y_M \\ x_O \cdot z_M & y_O \cdot z_M & z_O \cdot z_M \end{bmatrix} \quad [\text{Ec. 3.1}]$$

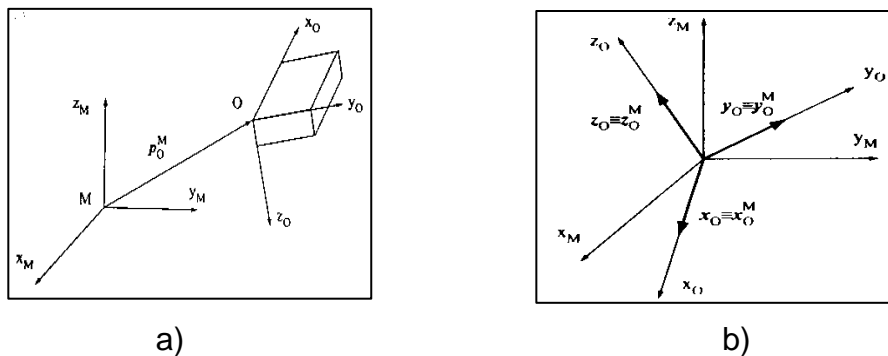
Se puede observar la proyección de cada vector unitario del sistema O con cada vector unitario del sistema M y como cálculo se obtiene el coseno del ángulo formado con los vectores unitarios de ambos sistemas. La matriz anterior corresponde a la orientación del sistema O respecto al M, la orientación del sistema M respecto al O será igual a la matriz inversa de la anterior y para este caso en que los sistemas son ortogonales será igual a la matriz transpuesta de la ecuación 3.1.

$${}^M Rot_O = ({}^O Rot_M)^T = ({}^O Rot_M)^{-1} \quad [\text{Ec. 3.2}]$$

$${}^O Rot_M = [x_M^O \ y_M^O \ z_M^O] = \begin{bmatrix} x_M \cdot x_O & y_M \cdot x_O & z_M \cdot x_O \\ x_M \cdot y_O & y_M \cdot y_O & z_M \cdot y_O \\ x_M \cdot z_O & y_M \cdot z_O & z_M \cdot z_O \end{bmatrix} \quad [\text{Ec. 3.3}]$$

En la figura 57, se muestran las figuras de la ubicación y orientación de un objeto en espacio.

Figura 57. **a) Posición del origen del sistema O para un objeto respecto al sistema de referencia M, b) orientación del sistema O respecto al M**



Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. p. 77.

3.1.1. Matriz homogénea de Foster, traslación y rotación

Para ubicar un objeto en el espacio tridimensional, Forest introdujo en 1969 las coordenadas homogéneas y la matriz de transformación homogénea; esta matriz es de dimensiones 4x4 compuesta por cuatro sub matrices: rotación, traslación, perspectiva y escalado. Las sub matrices de interés para robótica son las de rotación y traslación, las sub matrices perspectiva y escalado toman los valores de 0 para la perspectiva, ya que se trata de un entorno real y no en un entorno gráfico o digital, la sub matriz escalado toma el valor de 1, por encontrarse en un entorno real.

Considerando las componentes de la matriz homogénea, las tres primeras columnas y las tres primeras filas corresponden a la orientación del objeto en el sistema O respecto M y la cuarta columna corresponde a la posición del objeto con respecto al sistema de referencia M. Con la matriz homogénea de Forest es posible orientar y posicionar un objeto asociada a un sistema O respecto a un sistema de referencia M, de la ecuación 3,2, para obtener la orientación y posición del sistema M respecto al sistema O se determina mediante la transformada inversa de la matriz homogénea, se calcula la transpuesta de la matriz de rotación y menos el producto de la transpuesta de la matriz de rotación por el vector de traslación para calcular la sub matriz de traslación. La matriz homogénea de Forest y su inversa se expresan de la siguiente manera:

$$T = \begin{bmatrix} \text{rotación (3x3)} & \text{traslación (3x1)} \\ \mathbf{0 (1x3)} & \mathbf{1 (1x1)} \end{bmatrix} \quad [\text{Ec. 3.4}]$$

$$T^{-1} = \begin{bmatrix} \text{rotación}^T & -\text{rotación}^T \cdot \text{traslación} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \quad [\text{Ec. 3.5}]$$

Para el análisis de la traslación, por ejemplo, mover el efector final de un punto A a un punto B, se ejecuta una traslación sobre cada uno de los ejes del sistema M, formando con ello tres traslaciones básicas. Así se obtiene una traslación compuesta representada por el vector $\mathbf{p}(x,y,z)$. El orden en que se efectúa cada traslación no afecta el resultado final y al no haber rotación del sistema O la matriz homogénea será la siguiente:

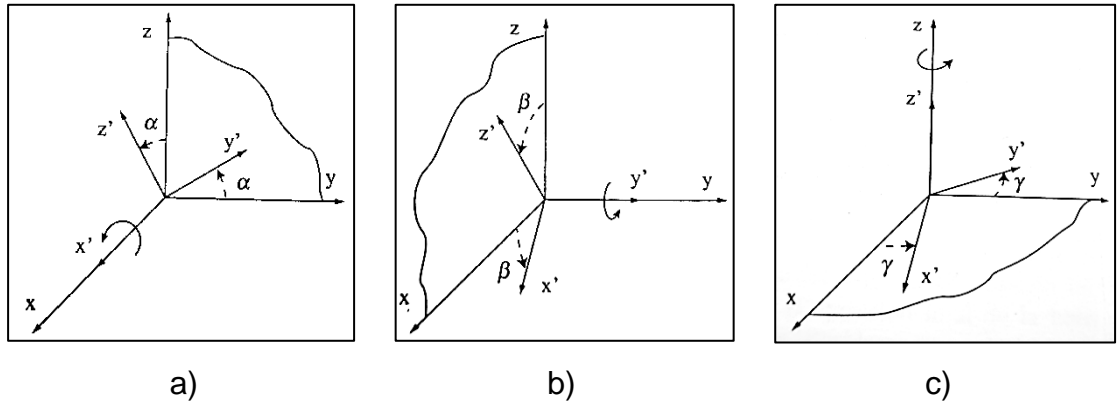
$$Tras(x, y, z) = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [\text{Ec. 3.6}]$$

Ahora se analizará la orientación del sistema O con respecto al sistema M. En este caso, se tienen tres rotaciones básicas alrededor de los ejes x, y, z; por tanto, al no tener ninguna traslación los componentes de la sub matriz de traslación serán cero y al girar un ángulo α alrededor del eje x permaneciendo este eje inalterable y aplicando la ecuación 3.1, donde el producto punto de dos vectores es igual al producto de los módulos de dichos vectores por el coseno del ángulo formado por esos vectores. Considerando que el $\cos(90^\circ)$ es igual a cero, $\cos(0^\circ)$ es igual a uno, $\cos(90^\circ+\alpha)$ es igual a $-\sin(\alpha)$ y $\cos(90^\circ-\alpha)$ es igual a $\sin(\alpha)$ se llegará a la siguiente expresión:

$$Rot(x, \alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & \cos(90 + \alpha) & 0 \\ 0 & \cos(90 - \alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [\text{Ec. 3.7}]$$

En la figura 58, se muestra la rotación alrededor del eje x, y, z.

Figura 58. Rotación alrededor del eje x al z



Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. p. 85.

De la figura 58b y 58c, si se procede a girar un ángulo β alrededor de y, un ángulo γ alrededor del eje z las expresiones simplificadas quedan como sigue:

$$Rot(y, \beta) = \begin{bmatrix} \cos(\beta) & 0 & \cos(90 - \beta) & 0 \\ 0 & 1 & 0 & 0 \\ \cos(\beta + 90) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [\text{Ec. 3.8}]$$

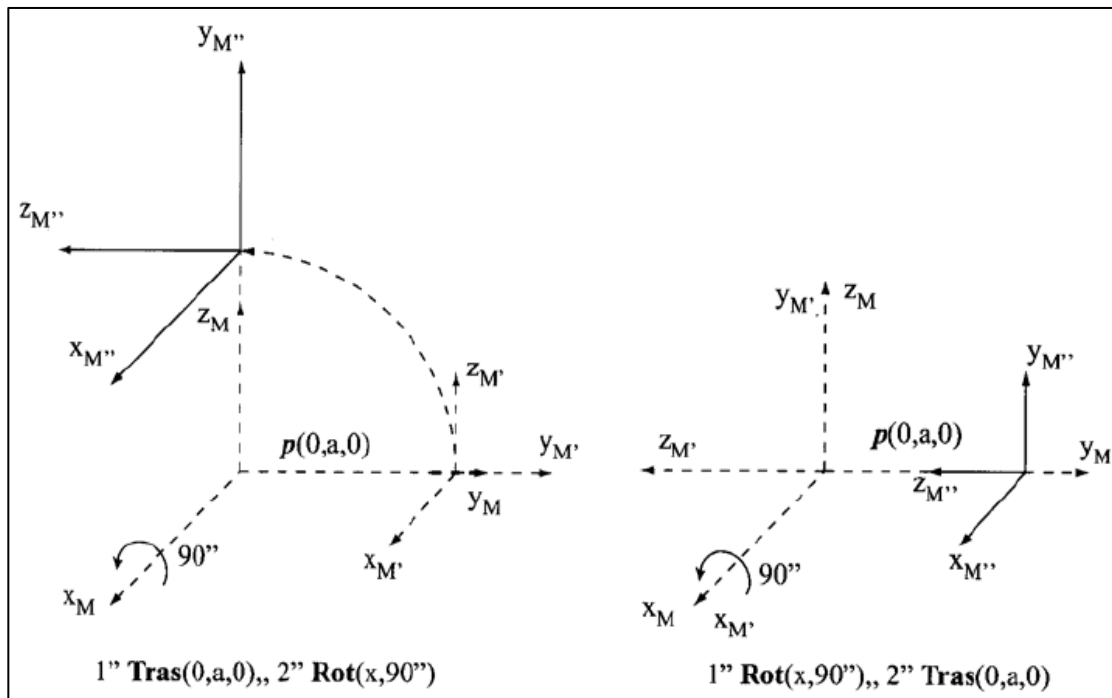
$$Rot(z, \gamma) = \begin{bmatrix} \cos(\gamma) & \cos(90 + \gamma) & 0 & 0 \\ \cos(90 - \gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [\text{Ec. 3.9}]$$

3.1.2. Asignación del sistema de referencia

Como puede observarse en las ecuaciones anteriores, el orden en que se aplican las transformaciones afecta el resultado final. Las transformaciones

están descritas por matrices y, por lo tanto, no son conmutativas. En el mundo real no se obtienen los mismos resultados trasladar y girar que girar y trasladar con respecto al mismo sistema de referencia. En la siguiente figura, se demuestra cómo afecta el orden en que se aplique las transformaciones para el primer caso, donde se tiene una traslación sobre el eje y en una magnitud de “a” y posteriormente una rotación de 90° alrededor del eje x con respecto al sistema M; para el segundo caso, se tiene una rotación de 90° alrededor del eje x con respecto al sistema M, y posteriormente se tiene una traslación sobre el eje y de una magnitud “a” quedando el sistema como se muestra en la figura 59.

Figura 59. **Movimientos con respecto al mismo sistema de referencia**



Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. p. 86.

Para comprender de cómo afecta el movimiento, a partir de dónde se referencia el sistema se tienen los siguientes casos. En el primer caso, el robot dará un giro de -90° alrededor del eje z con referencia en la base del robot y la posición inicial en el extremo del robot con el origen del sistema O, como se muestra en la figura 59. Al girar el robot hasta la posición O' se produce una nueva orientación que se describe con la ecuación 3.10

$$p_{i\ final}^R = {}^R Rot_{O'}(z, -90) \cdot p_{i\ inicial}^R \quad [\text{Ec. 3.10}]$$

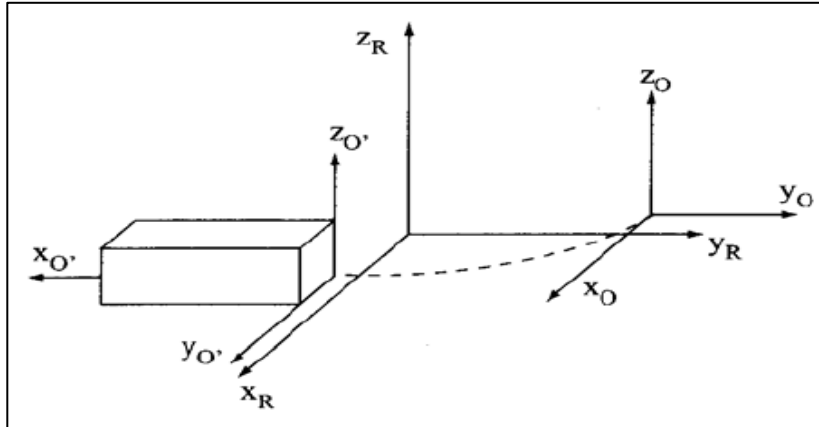
Donde $p_{i\ inicial}^R$ es cualquier punto del objeto y la posición inicial con respecto al sistema R, se observa en la ecuación 3.10 que la posición inicial no se encuentra en la base del robot y se necesitaría una traslación desde el sistema R al O para describir dicha posición, por lo que se necesita una traslación desde la base hasta la posición inicial descrita por la ecuación 3.11 hasta obtener la ecuación 3.12 que describe la nueva posición del objeto.

$$p_{O,i}^R = {}^R Tras_O(a, b, c) \cdot p_i^O \quad [\text{Ec. 3.11}]$$

$$p_{i\ final}^R = {}^R Rot_{O'}(z, -90) \cdot {}^R Tras_O(a, b, c) \cdot p_i^{O'} \quad [\text{Ec. 3.12}]$$

Donde $p_i^{O'}$ es cualquier punto del objeto con respecto al sistema O' que, en general. Es cualquier punto del espacio y que no ha tenido cambio, debido a la orientación del objeto. En el análisis que se obtuvo se puede apreciar que, cualquier transformación se pre multiplicará a la ya efectuada, y que dicha ecuación se aplicará de derecha a izquierda. En la figura 60, se muestra la figura para un sistema de referencia fijo.

Figura 60. Robot con sistema de referencia fijo



Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. p. 88.

En la figura 61, se presenta el siguiente sistema donde se pretende manipular el objeto descrito por el sistema O que representa su posición y orientación. El sistema H describe la posición y orientación de la herramienta, ubicada en el extremo del robot. Que la herramienta cumpla su cometido, la herramienta descrito por H se trasladará hasta la ubicación del objeto de manera que los orígenes de ambos sistemas serán coincidentes. La herramienta, ahora con sistema H' (H trasladado) se orientará con el objeto descrita por O; en primer lugar, se tendrá una traslación que se describe por la ecuación 3.13 y, posteriormente, se tendrá una rotación que coincide con el sistema O dado por H'' descrita por la ecuación 3.14; la rotación de H' ya no se hará con respecto a la posición inicial como en la traslación sino con respecto al sistema intermedio hasta llegar a posición final H''. Esto significa que H' girará con respecto a H'' hasta obtener la ecuación final 3.15.

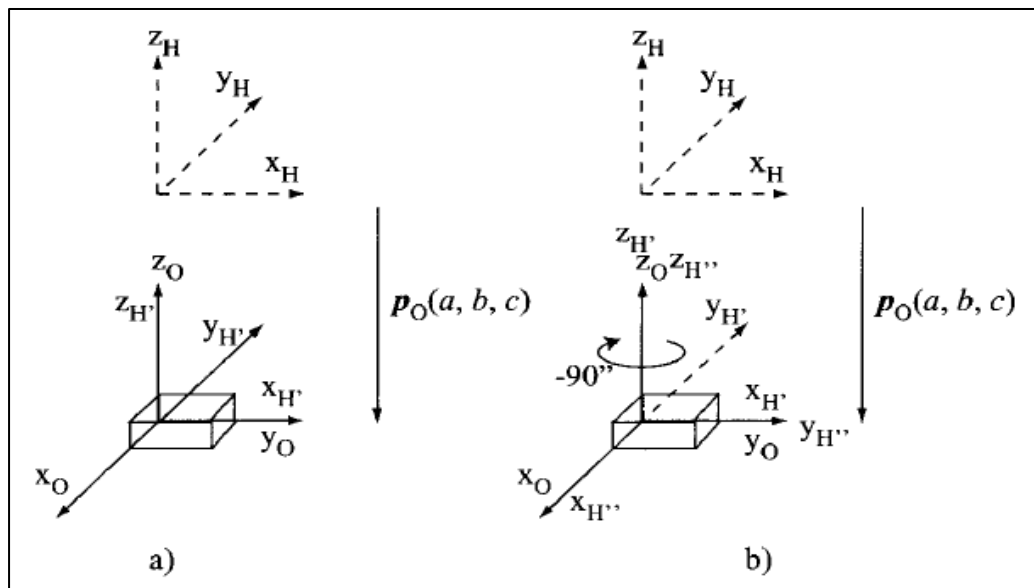
$$p^H = {}^H\text{Tras}_{H'}(a, b, c).p^{H'} \quad [\text{Ec. 3.13}]$$

$$p^{H'} = {}^{H''}Rot_{H'}(z, -90).p^{H''} \quad [\text{Ec. 3.14}]$$

$$p^H = {}^HTras_{H'}(a, b, c).{}^{H''}Rot_{H'}(z, -90).p^{H''} \quad [\text{Ec. 3.15}]$$

Como puede observarse en la ecuación 3.15, primero ocurre una traslación y posteriormente una rotación aplicando la ecuación de izquierda a derecha. Es decir que, cuando se utiliza un sistema de referencia móvil las transformaciones posteriores se pos multiplicarán a la ya efectuada. En la figura 61, se muestra la secuencia de los movimientos que efectúa la herramienta ubicada en el extremo del robot y en la que se asigna un sistema de referencia móvil.

Figura 61. **Representación de la asignación de un sistema de referencia móvil**



Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. p. 89.

En el efector final, también se tendrá una combinación de rotaciones básicas, para obtener una rotación compuesta que oriente de manera adecuada a dicho efector. Estas rotaciones se efectúan en cierto orden de las cuales se pueden obtener en un total de 24 combinaciones: 12 de las cuales son combinaciones de tres rotaciones simples realizadas sobre los ejes de un sistema fijo, y las otras 12 conocidas como ángulos de Euler, que realizan los giros sobre los eje de un sistema móvil.

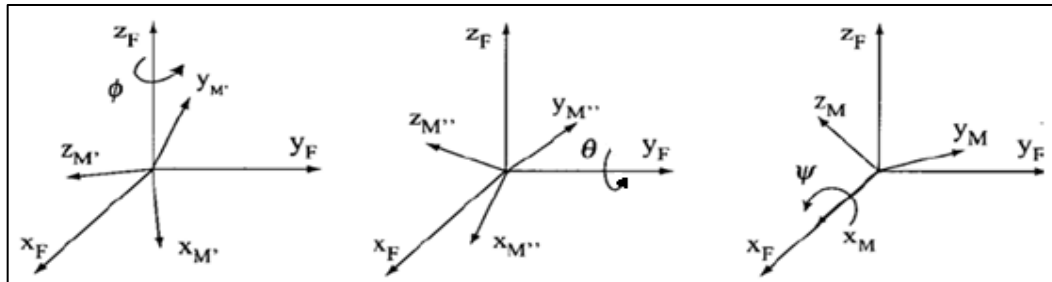
“Las 12 combinaciones, tanto para el sistema fijo como el móvil son las siguientes: XYZ, XZY, YXZ YZX, ZXY, ZYX, XYX, XZX, YXY, YZY, ZXZ, ZYZ (página 92 del libro guía, robots y sistemas sensoriales [27]); la más utilizada para el sistema móvil es la combinación XYZ conocida como guiñado (ψ) alrededor del eje x, cabeceo (θ) alrededor del eje y, alabeo (ϕ) alrededor del eje z”⁶. La combinación de los giros alrededor de los ejes del sistema móvil o ángulos de Euler es como las del sistema fijo, con la diferencia de que se aplica a un sistema móvil y las más utilizadas para este tipo de movimiento corresponden a la combinación ZYZ. La ecuación que representa dicha rotación compuesta para el sistema de referencia fija es la siguiente:

$$Rot(x, y, z_{fijo}) = Rot(z, \phi). Rot(y, \theta). Rot(x, \Psi) \quad [\text{Ec. 3.16}]$$

En la figura 62, se aprecia las rotaciones básicas para alrededor de los ejes del sistema de referencia fijo.

⁶ TORRES, F.; POMARES, J.; GIL, P. y otros. *Robots y sistemas sensoriales*.p. 107.

Figura 62. **Rotaciones simples de guiñado, cabeceo y alabeo correspondiente a un sistema de referencia fijo**



Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. p. 92.

3.1.3. Cinemática directa e inversa

El análisis de la cinemática directa proporcionara la posición y orientación del extremo del robot con respecto a un sistema de referencia fijo en función de los valores de las articulaciones. Si se conociera la posición y orientación del extremo del robot en un punto específico se necesitaría conocer los valores de las articulaciones (se conoce la posición y orientación desconociendo los valores de las articulación). A este tipo de análisis se le conoce como cinemática inversa. En este inciso, al igual que en los anteriores, se desarrolla este tema para entender y determinar de qué manera el robot ejecutará una tarea específica.

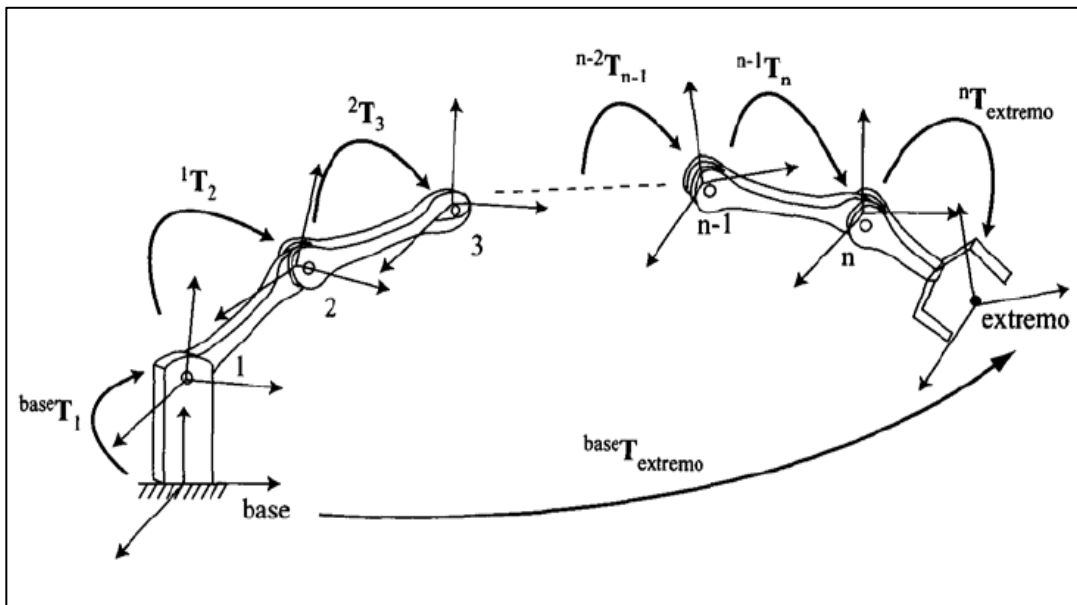
Para el análisis de cinemática utilizar relaciones trigonométricas para resolver la trasformación total resulta muy complicado, que sí se requiere que el robot alcance cualquier posición y orientación en el espacio, para ello se requerirá como mínimo: seis grados de libertad, tres para obtener la posición y tres para la orientación. El robot industrial está compuesto por eslabones y articulación que, a imitación del brazo humano será capaz de ejercer una tarea

específica. Como se estudió en los incisos anteriores, una transformación compleja se puede descomponer en transformaciones básicas de rotación y traslación, con este concepto se tiene que:

$${}^{base}T_{extremo} = {}^{base}T_1 \cdot {}^1T_2 \dots {}^{n-1}T_n \cdot {}^nT_{extremo} \quad [\text{Ec. 3.17}]$$

En donde ${}^{base}T_1$ es la transformación que hay desde la base al eslabón 1. De la misma manera existirá una transformación en los eslabones consecutivos hasta llegar al extremo del robot, donde se asignará un número a cada eslabón tomando la base del robot como el eslabón 0 y el valor de 1 para el siguiente hasta la n-ésimo eslabón. Se asignará el número del eslabón i a la articulación ubicada en la base del eslabón i . En la figura 63, se muestran los parámetros de las articulaciones.

Figura 63. **Asignación de los parámetros de las articulaciones**

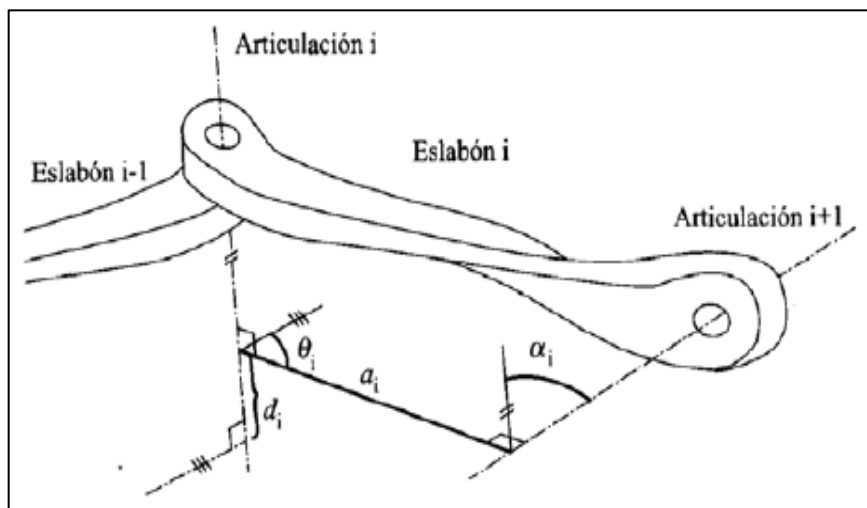


Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. p. 106.

3.1.3.1. Parámetros de Denavit y Hartenberg

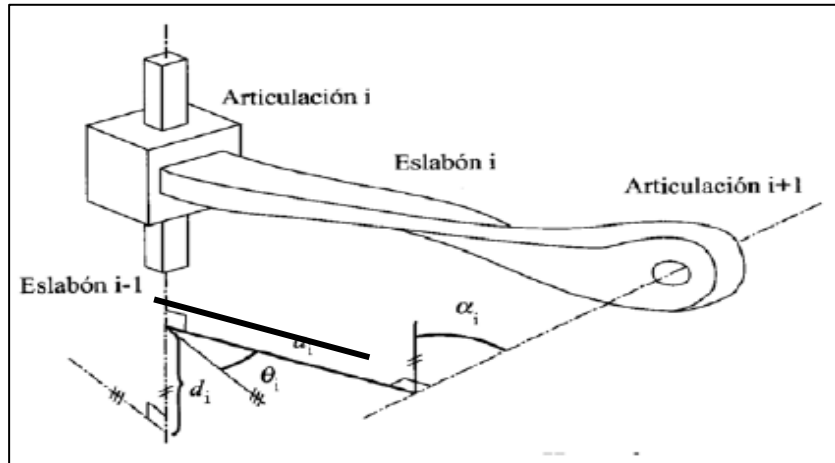
De vital importancia es el análisis matemático, porque se reduce significativamente con la utilización de los parámetros Denavit y Hartenberg. Dos de estos parámetros describen el tamaño y forma del eslabón, y los siguientes dos describen la posición relativa entre dos eslabones consecutivos y la articulación relativa que los enlaza. En la utilización de estos parámetros, también se definirá los ejes de cada articulación con respecto al cual se produce el movimiento de la articulación. En la figura 64, se muestran los parámetros de Denavit y Hartenberg para las articulaciones i rotacional y prismática de dos eslabones consecutivos.

Figura 64. **Parámetros de Denavit y Hartenberg a) para articulación i rotacional, b) para articulación i prismáticas**



a)

Continuación de la figura 64.



b)

Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. p. 108.

Los parámetros que se pueden observar en la figura anterior proporcionan información acerca de la forma, tamaño, posición relativa y articulación relativa que las enlaza. La línea marcada en las figuras representa a la línea normal común que no es más que la línea más corta ortogonal entre los dos ejes de las articulaciones i e $i+1$; a continuación se explica el significado de cada parámetro:

- El parámetro a_i proporciona la distancia entre los dos ejes i e $i+1$ a lo largo de la línea normal común, como se verá más adelante este parámetro representa la el tamaño del eslabón.
- El parámetro α_i proporciona el ángulo que se formaría entre los ejes de las articulaciones i e $i+1$ si se cortasen en el mismo punto, este parámetro específico la forma del eslabón.
- El parámetro d_i proporciona la distancia que existe entre las intersecciones de las normales comunes sobre el eje i , este parámetro

que mide la distancia entre dos eslabones sobre un mismo eje dado por el tamaño y forma del eslabón.

- El parámetro θ_i proporciona el ángulo que se formaría entre las normales comunes si se cortasen en el mismo punto del eje de la articulación, parámetro que proporciona la forma de la articulación dado por el ángulo formado por dos eslabones consecutivos⁷.

Para obtener la transformación en cada eslabón (i) se asignará un sistema de referencia que estará situado sobre el eje de la articulación que enlaza el eslabón i con el eslabón $i+1$ como se muestran en la figura 64. La dirección que tendrá cada eje del sistema será como sigue: el eje z_i del eslabón i se alineará con el eje de la articulación $i+1$, el eje x_i se alinea con la normal común en dirección de i a $i+1$ entre dichos ejes y el eje y_i se establece para que el sistema sea ortogonal dado por la regla de la mano derecha.

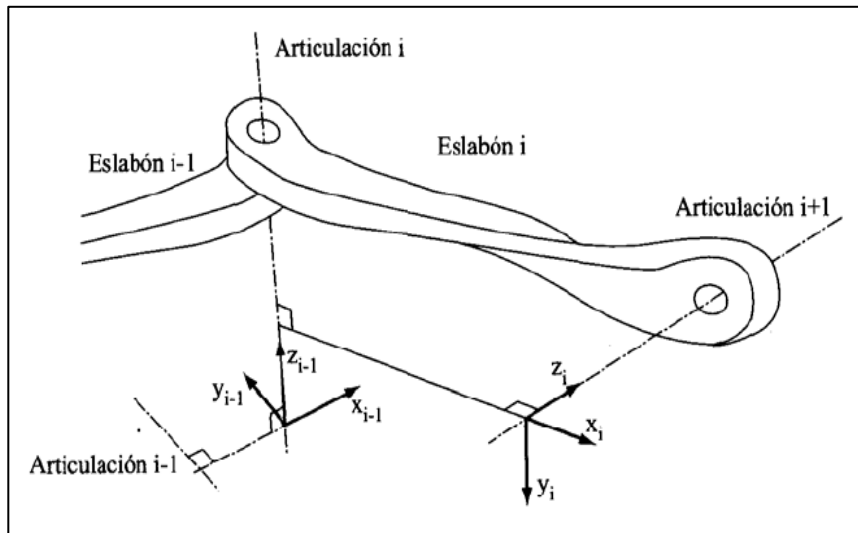
Otras consideraciones que se tendrán serán cuando los ejes son paralelos o cuando se corten en algún punto. Cuando los ejes son paralelos se tendrá varias normales comunes y, por consiguiente, el origen del sistema i es indefinido, por regla se tomará el origen del sistema i en la articulación $i+1$ y la dirección de los ejes como los ya mencionados. Cuando los ejes se cortan en algún punto el origen del sistema i será en el punto de corte con el eje x_i en dirección al plano formado por z_i y z_{i+1} ⁸.

Se procede a determinar cada transformación en todos los eslabones excepto el eslabón 0 (base) y el extremo. La dirección que tendrá el eje z_0 será al de la articulación 1 y al ser el eslabón 0 un sistema fijo no existirá la normal común entre el eslabón 0 al 1, y se tomará los ejes x_0 y y_0 de manera que sean ortogonales entre sí, dada por la regla de la mano derecha. Para el último eslabón el sistema de referencia se localizará en el extremo del eslabón y al no existir articulación $n+1$ la dirección del eje z_n será coincidente al z_{n-1} ; en la figura 65, se muestra la asignación del sistema de referencia para el eslabón i que servirá como ejemplo, para la asignación de los sistemas de referencia para cada eslabón.

⁷ TORRES, F.; POMARES, J.; GIL, P. y otros. *Robots y sistemas sensoriales*. p. 107.

⁸ *Ibíd.* p. 109.

Figura 65. **Asignación del sistema de referencia para el eslabón i**

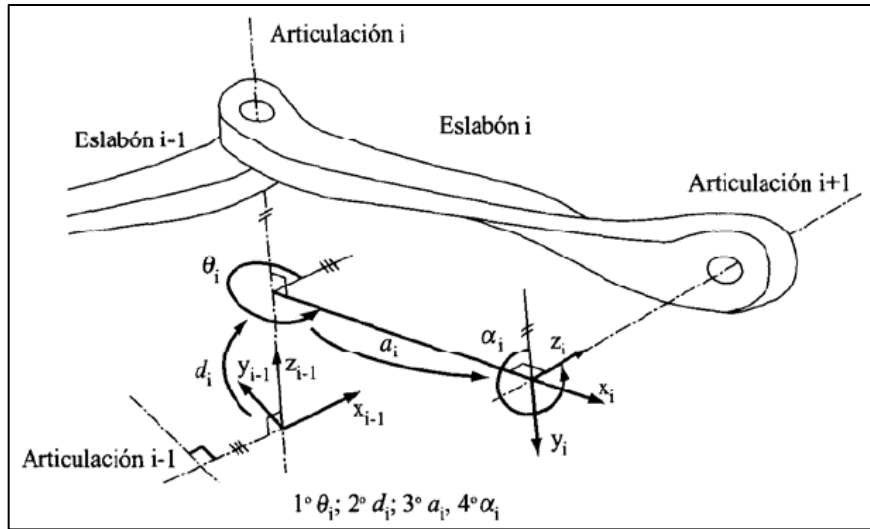


Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. p. 110.

3.1.3.2. **Matriz de transformación homogénea**

Esta matriz determina la transformación que se requiere para pasar del sistema $i-1$ al sistema i utilizando los parámetros Denavit y Hartenberg. Se determina la transformación que existe en una articulación dada por el sistema i , como se puede observar en la figura 66.

Figura 66. Transformaciones básicas para pasar del sistema i-1 al sistema i



Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. p. 113.

Como se puede observar en la figura 66, para pasar del sistema i-1 al sistema i se necesitan solo dos rotaciones y dos traslaciones. Se iniciará con los parámetros que determinan la posición relativa a una rotación y luego a una prismática. Luego, se tienen los parámetros que determinan el tamaño y la forma del eslabón quedando las transformaciones básicas en el orden que siguen: $Rot(z_{i-1}, \theta_i)$, $Tras(z_{i-1}, d_i)$, $Tras(x_i, a_i)$ y $Rot(x_i, \alpha_i)$ obteniendo la siguiente matriz de transformación.

$${}^{i-1}T_i = Rot(z_{i-1}, \theta_i) \cdot Tras(z_{i-1}, d_i) \cdot Tras(x_i, a_i) \cdot Rot(x_i, \alpha_i) =$$

Utilizando las ecuaciones 3.6 a la 3.9 se tiene lo siguiente:

$$\begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
= \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \cdot \sin \theta_i & \sin \alpha_i \cdot \sin \theta_i & a_i \cdot \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cdot \cos \theta_i & -\sin \alpha_i \cdot \cos \theta_i & a_i \cdot \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [\text{Ec. 3.18}]$$

Para utilizar de manera correcta los parámetros de Denavit y Hartenberg y solucionar el problema de la cinemática directa, se proseguirá con lo siguiente:

- Se enumerarán todos los eslabones empezando con la base a 0 hasta el último eslabón.
- Se enumerarán las articulaciones empezando con 1 que es el número del eslabón que conecta la base del robot.
- Se localizarán los ejes de cada articulación.
- Se establecerán los sistemas de referencia en cada articulación como se mostró anteriormente, donde se establece el sistema de referencia i al final del eslabón i.
- Se establecerá el sistema de la base del robo con z_0 en dirección del eje 1, con ejes x_0 y y_0 determinadas por la regla de mano derecha.
- Se establecerá el sistema del último eslabón n, con z_n en la misma dirección que z_{n-1} y x_n e y_n dada por la regla de mano derecha de modo que el sistema sea ortogonal.
- Se creará una tabla de los parámetros de Denavit y Hartenberg para cada articulación.
- Se calculará la matriz de transformación homogénea para cada articulación dada por la ecuación 3.18.
- Por último. se calcula la matriz de transformación homogénea total dada por ecuación 3.17⁹.

⁹ TORRES, F.; POMARES, J.; GIL, P. y otros. *Robots y sistemas sensoriales*.p. 116.

3.1.3.3. Solución del problema de cinemática inversa (solución de Pieper)

En la cinemática directa se determina la posición y orientación del extremo del robot conociendo los valores de las variables articulares. La cinemática inversa determina los valores de las variables articulares para que el extremo del robot se encuentre en una posición y orientación dada. Para solucionar este problema se podría utilizar una solución numérica, como se sabe, resulta tediosa y muy difícil; haciendo uso de una solución geométrica que por trigonometría se descompondrá la cadena cinemática del robot para su respectivo análisis y establecer un sistema de seis incógnitas si se requiere que el robot como mínimo pueda posicionarse y orientarse en cualquier dirección del espacio tridimensional.

El método de Pieper proporciona la solución de la cinemática inversa; empieza con la solución de la posición de la muñeca; es decir, para las tres primeras articulaciones. La orientación de la muñeca no alterará la posición del robot en un punto dado. Con esto se logra cualquier orientación sin afectar la posición, al entender que la posición de la muñeca no es la posición del extremo del robot, ya que físicamente la muñeca tiene una dimensión dada se busca posicionar la muñeca, a partir de la posición deseada del efector final. De la figura 60, se puede observar y determinar que $\mathbf{p}_5^0 = \mathbf{p}_{\text{muñeca}}^0 (x_5^0, y_5^0, z_5^0)$ la posición del extremo del robot $\mathbf{p}_6^0 = \mathbf{p}_{\text{deseada}}^0 (x_{\text{extremo}}^0, y_{\text{extremo}}^0, z_{\text{extremo}}^0)$, en donde se cumple que:

$$p_5^0 = p_6^0 - d_6 z_6 \quad [\text{Ec. 3.19}]$$

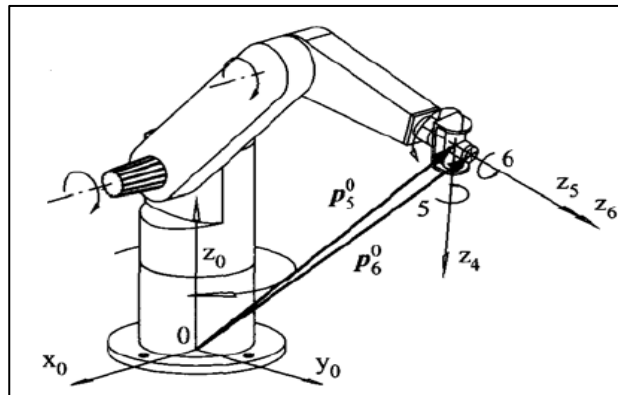
Donde las coordenadas de \mathbf{z}_6 con respecto a la base del robot corresponden a la orientación de la muñeca que es igual a la orientación del

extremo del robot, con los valores para r_{13} , r_{23} y r_{33} de la ecuación 3.4 se tiene lo siguiente:

$$\begin{aligned} x_5^0 &= x_{muñeca}^0 = x_6^0 - d_6 \cdot r_{13} \\ y_5^0 &= y_{muñeca}^0 = y_6^0 - d_6 \cdot r_{23} \\ z_5^0 &= z_{muñeca}^0 = z_6^0 - d_6 \cdot r_{33} \end{aligned} \quad [\text{Ec. 3.20}]$$

En la figura 67, se muestra los vectores para la posición de la muñeca y el extremo del robot.

Figura 67. **Vectores de posición para la muñeca y el extremo del robot**



Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. p. 124.

De la ecuación 3.19 se desconocen los valores de los ángulos para la orientación del efector final de los que corresponden a las tres últimas articulaciones, reescribiendo la ecuación 3.19 de la ecuación general de traslación y rotación del extremo del robot con respecto a la base, se tiene lo siguiente:

$${}^0T_{extremo} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x_{extremo}^0 \\ r_{21} & r_{22} & r_{23} & y_{extremo}^0 \\ r_{31} & r_{32} & r_{33} & z_{extremo}^0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [\text{Ec. 3.21}]$$

$${}^0Rot_{extremo} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad [\text{Ec. 3.22}]$$

$${}^0Tras_{extremo} = p_{extremo}^0(x_{extremo}^0, y_{extremo}^0, z_{extremo}^0) \quad [\text{Ec. 3.22b}]$$

Con la ecuación 3.22 y el problema de cinemático inverso en la que se conoce la posición y orientación del extremo de robot, se obtiene la rotación del extremo del robot con respecto a la base y resolviendo por método geométrico o algebraico para las primeras articulaciones se conoce la matriz de rotación 0Rot_3 . Queda por determinar la matriz del extremo del robot con respecto a la tercera articulación en la que se obtienen las siguientes expresiones y resolviendo el problema de cinemática inversa de manera elegante como sigue:

$${}^0Rot_{extremo} = {}^0Rot_3 \cdot {}^3Rot_{extremo} \quad [\text{Ec. 3.23}]$$

$${}^3Rot_{extremo} = {}^3Rot_6 = ({}^0Rot_3)^{-1} \cdot {}^0Rot_{extremo} = ({}^0Rot_3)^T \cdot {}^0Rot_{extremo} \quad [\text{Ec. 3.24}]$$

3.1.3.4. Matriz Jacobiana y matriz Jacobiana inversa

Para un conjunto m funciones en la que cada función dependen de n variables independientes (ecuación 3.25) y si las variables en que las funciones dependen son dependientes del tiempo y al derivarlas con respecto al tiempo (ecuación 3.26) se obtendrá el siguiente conjunto de funciones:

$$\begin{aligned}
y_1 &= f_1(x_1, x_2, x_3, \dots, x_n) \\
y_2 &= f_2(x_1, x_2, x_3, \dots, x_n) \\
&\vdots \\
&\vdots \\
y_m &= f_m(x_1, x_2, x_3, \dots, x_n)
\end{aligned}
\tag{Ec. 3.25}$$

$$\begin{aligned}
\dot{y}_1 &= \frac{\partial f_1}{\partial x_1} \cdot \dot{x}_1 + \frac{\partial f_1}{\partial x_2} \cdot \dot{x}_2 + \frac{\partial f_1}{\partial x_3} \cdot \dot{x}_3 + \dots + \frac{\partial f_1}{\partial x_n} \cdot \dot{x}_n \\
\dot{y}_2 &= \frac{\partial f_2}{\partial x_1} \cdot \dot{x}_1 + \frac{\partial f_2}{\partial x_2} \cdot \dot{x}_2 + \frac{\partial f_2}{\partial x_3} \cdot \dot{x}_3 + \dots + \frac{\partial f_2}{\partial x_n} \cdot \dot{x}_n \\
&\vdots \\
&\vdots \\
\dot{y}_m &= \frac{\partial f_m}{\partial x_1} \cdot \dot{x}_1 + \frac{\partial f_m}{\partial x_2} \cdot \dot{x}_2 + \frac{\partial f_m}{\partial x_3} \cdot \dot{x}_3 + \dots + \frac{\partial f_m}{\partial x_n} \cdot \dot{x}_n
\end{aligned}
\tag{Ec. 3.26}$$

Al expresar las ecuaciones 3.25 y 3.26 en forma matricial se tiene lo siguiente:

$$\mathbf{y} = \mathbf{F}(\mathbf{x})
\tag{Ec. 3.27}$$

$$\dot{\mathbf{y}} = \mathbf{J}(\mathbf{x}) \cdot \dot{\mathbf{x}}
\tag{Ec. 3.28}$$

La matriz $\mathbf{J}(\mathbf{x})$ se le conoce como matriz jacobiana con la variable \mathbf{x} dependiente del tiempo, por lo que se tendrá un valor diferente de la matriz para cada instante de tiempo. De la ecuación 3.21 (matriz de transformación total), se puede observar que hay doce posibles soluciones que son función de las n variables articulares, de las cuales nueve ecuaciones corresponden a la rotación y tres ecuaciones corresponden a la traslación (ecuación 3.29). Para realizar una rotación en cualquier sentido del espacio articular se necesitan

solamente tres grados de libertad y de las nueve ecuaciones solamente se necesitan tres que resultan linealmente independientes, queda la expresión de la 3.21 como sigue:

$$\begin{aligned}x_{extremo}^0 &= f_1(q_1, q_2, q_3, \dots, q_n) \\y_{extremo}^0 &= f_2(q_1, q_2, q_3, \dots, q_n) \\z_{extremo}^0 &= f_3(q_1, q_2, q_3, \dots, q_n)\end{aligned}$$

$$\begin{aligned}r_{11} &= f_4(q_1, q_2, q_3, \dots, q_n) \\r_{12} &= f_5(q_1, q_2, q_3, \dots, q_n) \\r_{13} &= f_6(q_1, q_2, q_3, \dots, q_n) \\r_{21} &= f_7(q_1, q_2, q_3, \dots, q_n) \\r_{22} &= f_8(q_1, q_2, q_3, \dots, q_n) \\r_{23} &= f_9(q_1, q_2, q_3, \dots, q_n) \\r_{31} &= f_{10}(q_1, q_2, q_3, \dots, q_n) \\r_{32} &= f_{11}(q_1, q_2, q_3, \dots, q_n) \\r_{33} &= f_{12}(q_1, q_2, q_3, \dots, q_n)\end{aligned}$$

[Ec. 3.29]

De lo que resulta de la ecuación 3.29 como sigue:

$$\begin{aligned}x_{extremo}^0 &= f_x(q_1, q_2, q_3, \dots, q_n) \\y_{extremo}^0 &= f_y(q_1, q_2, q_3, \dots, q_n) \\z_{extremo}^0 &= f_z(q_1, q_2, q_3, \dots, q_n)\end{aligned}$$

$$\begin{aligned}\alpha_{extremo}^0 &= f_\alpha(q_1, q_2, q_3, \dots, q_n) \\\beta_{extremo}^0 &= f_\beta(q_1, q_2, q_3, \dots, q_n) \\\gamma_{extremo}^0 &= f_\gamma(q_1, q_2, q_3, \dots, q_n)\end{aligned}$$

[Ec. 3.30]

Es importante la matriz jacobiana en robótica industrial, a partir de ella se obtiene una expresión para la velocidad lineal y velocidad angular del extremo del robot con respecto a la base, se tiene en cuenta que la matriz Jacobiana es una matriz de derivadas y que al derivar la ecuación 3.30 con respecto al tiempo se obtiene la siguiente expresión.

$$\begin{aligned}
 \dot{x}_{extremo}^0 &= \frac{\partial f_x}{\partial q_1} \cdot \dot{q}_1 + \frac{\partial f_x}{\partial q_2} \cdot \dot{q}_2 + \dots + \frac{\partial f_x}{\partial q_n} \cdot \dot{q}_n \\
 \dot{y}_{extremo}^0 &= \frac{\partial f_y}{\partial q_1} \cdot \dot{q}_1 + \frac{\partial f_y}{\partial q_2} \cdot \dot{q}_2 + \dots + \frac{\partial f_y}{\partial q_n} \cdot \dot{q}_n \\
 \dot{z}_{extremo}^0 &= \frac{\partial f_z}{\partial q_1} \cdot \dot{q}_1 + \frac{\partial f_z}{\partial q_2} \cdot \dot{q}_2 + \dots + \frac{\partial f_z}{\partial q_n} \cdot \dot{q}_n \\
 \\
 \dot{\alpha}_{extremo}^0 &= \frac{\partial f_\alpha}{\partial q_1} \cdot \dot{q}_1 + \frac{\partial f_\alpha}{\partial q_2} \cdot \dot{q}_2 + \dots + \frac{\partial f_\alpha}{\partial q_n} \cdot \dot{q}_n \\
 \dot{\beta}_{extremo}^0 &= \frac{\partial f_\beta}{\partial q_1} \cdot \dot{q}_1 + \frac{\partial f_\beta}{\partial q_2} \cdot \dot{q}_2 + \dots + \frac{\partial f_\beta}{\partial q_n} \cdot \dot{q}_n \\
 \dot{\gamma}_{extremo}^0 &= \frac{\partial f_\gamma}{\partial q_1} \cdot \dot{q}_1 + \frac{\partial f_\gamma}{\partial q_2} \cdot \dot{q}_2 + \dots + \frac{\partial f_\gamma}{\partial q_n} \cdot \dot{q}_n \quad \text{[Ec. 3.31]}
 \end{aligned}$$

Donde q_n es la variable articular y \dot{q}_n es la velocidad relativa que existe entre los eslabones i e $i+1$, si se observa la ecuación 3.31 se nota que las velocidades lineales y las velocidades angulares con respecto a la base, se encuentran en función de las velocidades articulares y del tiempo y al expresar la ecuación 3.31 en forma matricial se obtiene lo siguiente:

$$\begin{bmatrix} \dot{x}_{extremo}^0 \\ \dot{y}_{extremo}^0 \\ \dot{z}_{extremo}^0 \\ \dot{\alpha}_{extremo}^0 \\ \dot{\beta}_{extremo}^0 \\ \dot{\gamma}_{extremo}^0 \end{bmatrix} = J(\mathbf{q}) \cdot \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \vdots \\ \dot{q}_n \end{bmatrix} \quad \text{[Ec. 3.32]}$$

$$\begin{bmatrix} v_{extremo}^0 \\ \omega_{extremo}^0 \end{bmatrix} = \begin{bmatrix} J_v(q) \\ J_w(q) \end{bmatrix} \cdot \dot{q} = J(q) \cdot \dot{q} \quad [\text{Ec. 3.33}]$$

En robótica, igual que en cinemática inversa, muchas veces, es necesario conocer qué valores necesita la variable articular para que el extremo del robot tenga una posición específica. En esta caso, se analiza qué velocidades tendrían las variables articulares para que el extremo del robot se mueva a una velocidad lineal o angular determinada.

Para lograr obtener la matriz jacobiana inversa no solo se utiliza necesariamente la ecuación 3.32, ya que al obtener la inversa de forma directa se necesita que la matriz sea cuadrada y que la matriz no resulte nula. Para determinar la jacobiana inversa se determina de manera similar a la forma en que se obtiene la matriz jacobiana directa. A partir de la solución de la cinemática inversa (ecuación 3.34) y derivando esta expresión con respecto al tiempo y expresándola de forma matricial (ecuación 3.35) se obtiene la matriz jacobiana inversa y la solución se determina las velocidades articulares en función de las velocidades lineales y angulares del extremo del robot con respecto a la base:

$$\begin{aligned} q_1 &= f_1(x_{extremo}^0, y_{extremo}^0, z_{extremo}^0, \alpha_{extremo}^0, \beta_{extremo}^0, \gamma_{extremo}^0) \\ q_2 &= f_2(x_{extremo}^0, y_{extremo}^0, z_{extremo}^0, \alpha_{extremo}^0, \beta_{extremo}^0, \gamma_{extremo}^0) \\ &\vdots \\ q_n &= f_n(x_{extremo}^0, y_{extremo}^0, z_{extremo}^0, \alpha_{extremo}^0, \beta_{extremo}^0, \gamma_{extremo}^0) \end{aligned} \quad [\text{Ec. 3.34}]$$

$$\dot{q} = J^{-1}(x_{extremo}^0, y_{extremo}^0, z_{extremo}^0, \alpha_{extremo}^0, \beta_{extremo}^0, \gamma_{extremo}^0) \cdot \begin{bmatrix} v_{extremo}^0 \\ \omega_{extremo}^0 \end{bmatrix} \quad [\text{Ec. 3.35}]$$

3.1.3.5. Fuerzas estáticas

En el estudio de la dinámica es de interés determinar las fuerzas y pares que originan el movimiento. En caso contrario, en el estudio de la cinemática es de interés estudiar las fuerzas y pares que las articulaciones ejercen para que el robot permanezca en equilibrio estático. Las articulaciones de tipo rotacional, como las que ejercen los motores, ejercen algún par entre los eslabones que conecta y las articulaciones prismáticas como los que ejercen los cilindros, ejercen una fuerza entre los eslabones que conecta que aunque sea de tipo primático sigue siendo un eslabón.

La fuerza y el par que el extremo del robot ejerce sobre algún dispositivo, es el resultado de las fuerzas y pares que ejerce cada una de las articulaciones, por lo tanto, se transmiten al extremo del robot, así como se estudia el equilibrio estático del robot; el resultado de las fuerzas y pares serán iguales a cero.

En la figura 68, se observa que en el extremo del robot existirá una fuerza neta \mathbf{f} y un par neto \mathbf{n} que será representado por un vector $\mathbf{F} = (\mathbf{f}, \mathbf{n})^T = (f_x, f_y, f_z, n_x, n_y, n_z)^T$ y si el vector $\boldsymbol{\tau}$ representa las fuerzas y pares de las articulaciones, para el análisis si se tuviera un desplazamiento $\delta \mathbf{x}$ del extremo del robot, debido la fuerza neta \mathbf{F} y $\delta \mathbf{q}$ es el desplazamiento que tendrían las articulaciones, de la ecuación se obtendrá una relación válida, y queda como sigue:

$$\delta \mathbf{x} = \mathbf{J}(\mathbf{q}) \cdot \delta \mathbf{q} \quad [\text{Ec. 3.36}]$$

El trabajo que tendría el sistema sería como la siguiente:

$$\delta w = \mathbf{F}^T \cdot \delta \mathbf{x} - \boldsymbol{\tau}^T \cdot \delta \mathbf{q} \quad [\text{Ec. 3.37}]$$

Al sustituir la ecuación 3.36 en la ecuación 3.37, se obtiene lo siguiente:

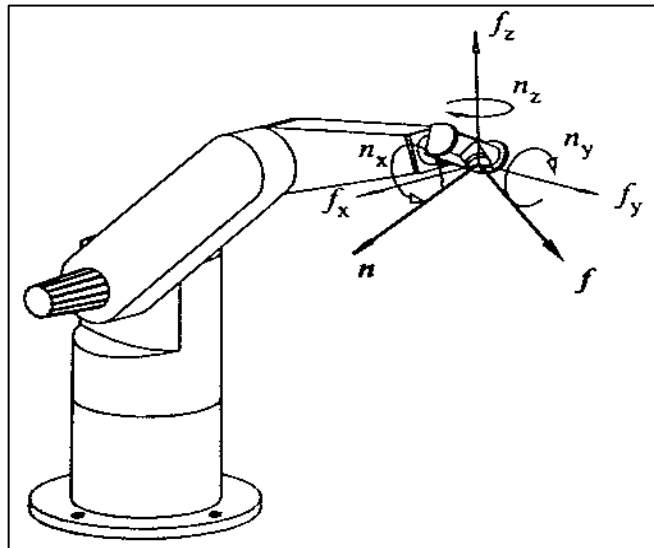
$$\delta w = (\mathbf{F}^T \cdot \mathbf{J}(\mathbf{q}) - \boldsymbol{\tau}^T) \cdot \delta \mathbf{q} \quad [\text{Ec. 3.38}]$$

Como el sistema se encuentra en equilibrio y estático, el trabajo realizado por el sistema será igual a cero, por lo que las fuerzas y pares ejercidas en el extremo del robot están relacionadas con las fuerzas y pares ejercidas por las articulaciones a través de la traspuesta de la Jacobiana:

$$\boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q}) \cdot \mathbf{F} \quad [\text{Ec. 3.39}]$$

En la figura 68, se muestra las fuerzas y pares ejercidas en el extremo del robot.

Figura 68. **Par y fuerza neta en el extremo del robot debido a pares y fuerzas en las articulaciones**



Fuente: TORRES, Fernando y otros. *Robots y sistemas sensoriales*. p. 134.

3.2. Efectores

Un efector final no es más que un dispositivo que se une a la muñeca del robot industrial para realizar una tarea específica. Los efectores finales se pueden agrupar en dos categorías: por un lado, se encontrarán las pinzas; por el otro, las herramientas que junto con estos se incluyen los elementos sensoriales con él le permiten al robot interactuar con su entorno.

3.2.1. Pinzas

Para aplicaciones donde se necesita mover de un lugar un objeto o por su forma, donde se necesita sostener el objeto es necesaria la utilización de pinzas. Un robot con pinzas puede realizar tarea de carga y descargar, recoger objetos, entre otros. Las pinzas permiten agarrar objetos y de cierta manera el robot se asemeja a un brazo humano capaz de realizar alguna aplicación específica, ya que está formado por piezas metálicas en forma de dedos que son impulsados por un mecanismo que permite agarrar y sostener una pieza.

Las pinzas pueden sujetar un objeto de dos formas, una de estas es por medio de compresión de los dedos sobre la pieza, y la segunda, por medio de sujeción. En esta última forma, las pinzas ejercen una fuerza suficiente como para evitar un deslizamiento causado por la gravedad.

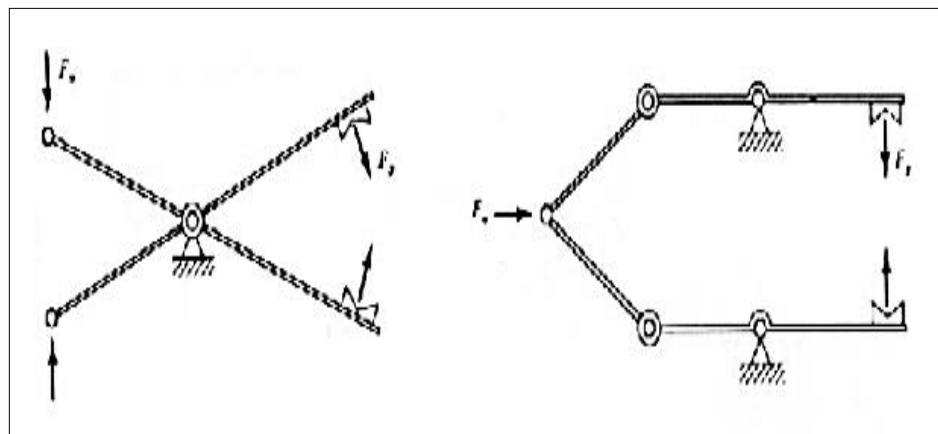
Las pinzas se clasifican de acuerdo con el tipo de movimiento de los dedos de los cuales pueden ser: movimiento por pivote en el que los dedos giran alrededor de los puntos fijos del pivote, para abrir y cerrar las pinzas, y por movimiento lineal o traslación en que los dedos se abren y cierran por medio de carriles guías en el que los dedos están unidos y se deslizan a lo largo durante

el movimiento. Entre los tipos de pinzas que se pueden mencionar de acuerdo con el tipo de componente utilizado como actuador están las siguientes:

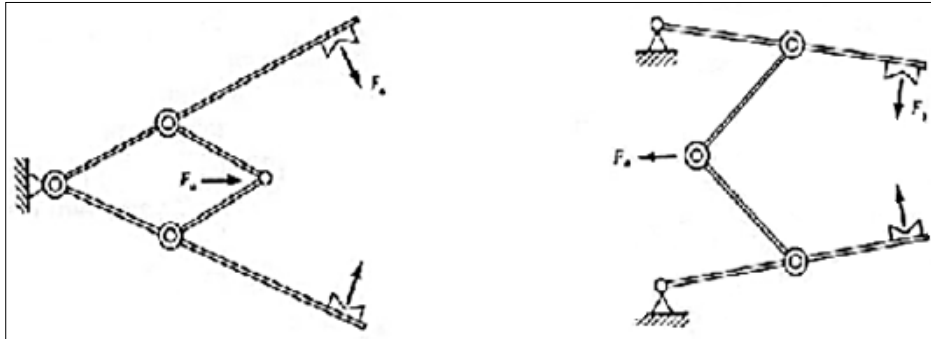
- Actuación por articulación.
- Actuación por engranaje y bastidor.
- Actuación por leva.
- Actuación por tornillo.
- Actuación por cable y polea.

Para el tipo de actuación por articulación se tiene algunos diseños que se muestran en la figura 69. En la figura se muestra la fuerza de entrada a la pinza como F_a y la fuerza de agarre transmitida, a través de la pinza como figura. Al observar las figuras se encuentra que para ciertos diseños se tendrá mayor ángulo de apertura, y para otros, mayor rapidez de acción, así como mayor fuerza de agarre. En la figura 63, se muestran algunos diseños de pinzas de la categoría actuación por articulación.

Figura 69. **Pinzas actuadas por articulación**



Continuación de la figura 69.



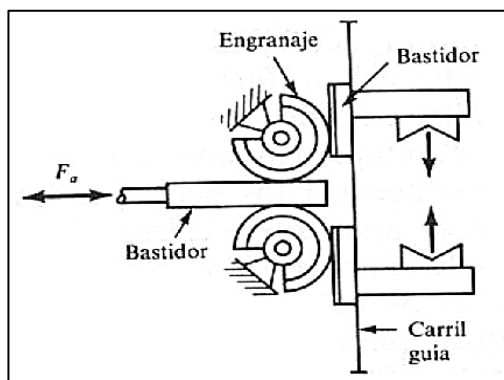
Fuente: GROOVER, Mikell y otros. *Robótica industrial*. p. 137.

En la figura 70^a, se muestra el tipo de pinzas actuadas por engranaje y bastidor, la fuerza de entrada es aplicada por un cilindro u otro actuador que proporciona un movimiento lineal al bastidor, y este, a su vez a impulsan a los engranajes que abrirán y cerrarán las pinzas. En la figura 70b, se muestra al tipo de pinza actuado por leva. Esta pinza está conformada por una leva, rodillos como seguidor y un muelle que regresara a las pinzas a su posición en reposo cuando la leva es accionada en sentido contrario. El muelle se ajustará para las piezas de diferentes tamaños que, junto con la leva, accionarán a las pinzas para su apertura y cierre.

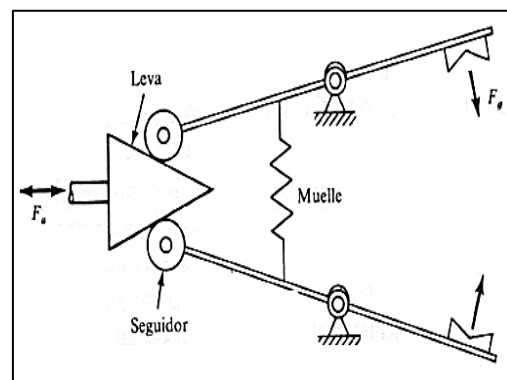
En la figura 70c, se muestra el tipo de pinzas más robusto, ya que utiliza para la apertura y cierre de las pinzas la utilización de un tornillo. Este tornillo es actuado por motor que, junto con un mecanismo de reducción de velocidad hacen girar el tornillo en un sentido, pero contrario para la apertura y cierre de las pinzas. Otro tipo de actuación de pinzas es la que se puede realizar por medio de cables y poleas, para la apertura y cierre de pinzas como se muestra en la figura 70d, donde se tienen resortes o muelles de torsión para el cierre de

las pinzas y los resortes o muelle de tensión que se oponen a los muelles de torsión para la apertura de la misma.

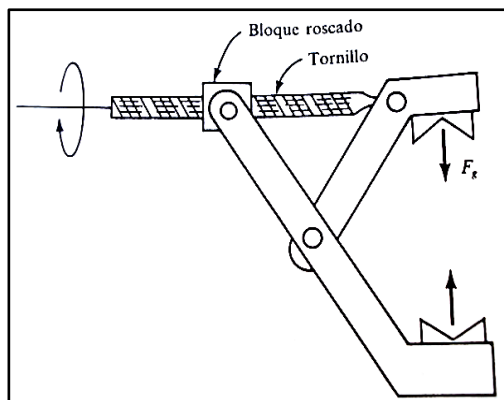
Figura 70. a) Actuación de pinza por engranaje y bastidor, b) de pinza por leva, c) de pinza por tornillo y d) de pinza por cable y polea



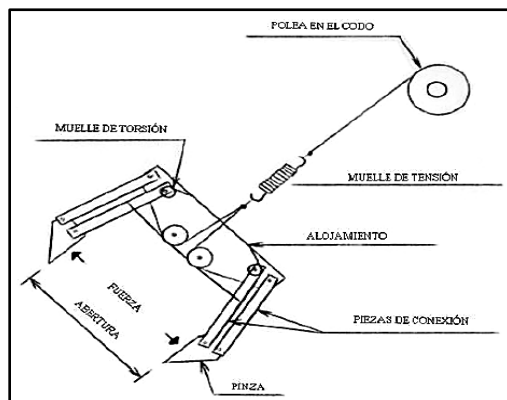
a)



b)



c)



d)

Fuente: GROOVER, Mikell y otros. *Robótica industrial*. p. 137.

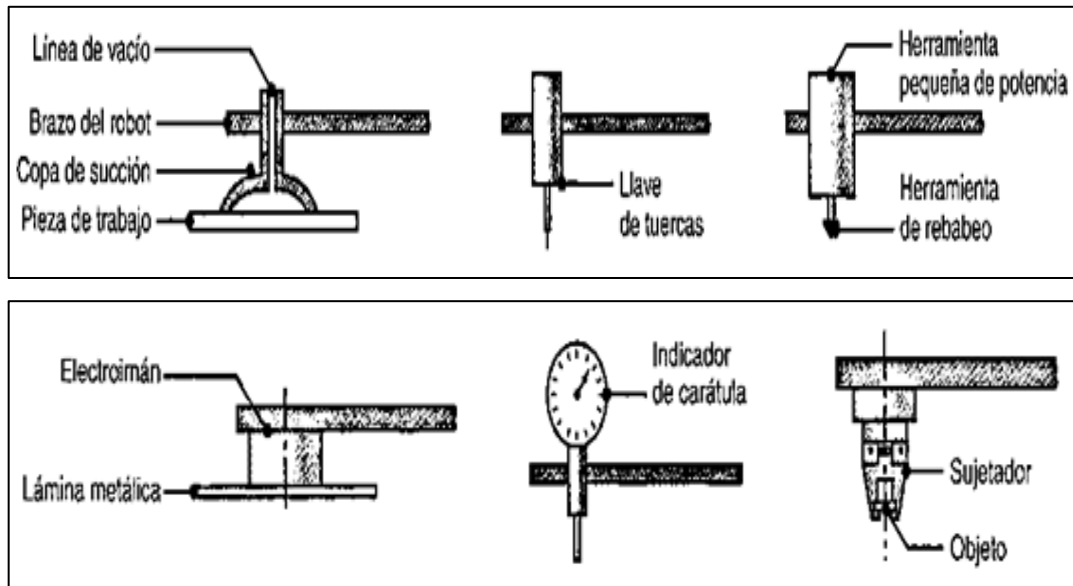
3.2.2. Herramientas

Como herramienta que utilizaría el ser humano, el efector final utilizará herramientas en aplicaciones en donde se exija al robot realizar alguna operación en alguna pieza de trabajo. Se utilizará una pinza diseñada de manera que pueda agarrar y manipular una herramienta, puede haber más de una herramienta que pudiera utilizar el robot en un período de trabajo, y permitiendo de esta manera intercambiar herramientas durante el período de trabajo.

Las herramientas más comunes que se pueden mencionar son las siguientes: las perforadoras, taladros, cepillado, las pinzas de soldar, los sopletes de pintar, las esmeriladoras portátiles, los sopletes de arenar, soplete por punto, soplete por arco, tobera de pintura por pulverización y otros.

La aplicación de los robots son para realizar trabajos donde hay ambientes sucios y peligrosos, un ser humano debería utilizar casco, protectores auditivos, sistema de respiración autónomo o por lo menos una máscara antipolvo, guantes, delantal de cuero, posiblemente con lámina de plomo antirradiaciones electromagnéticas (rayos X), botines de seguridad, etc. Además de un sistema que pueda balancear el peso de las herramientas, por lo que el robot no solo realizará mejor la tarea, sino que además, será más económico. En la figura 71, se muestran algunos dispositivos y herramientas sujetos a efectores finales para realizar algunas variedades de operaciones.

Figura 71. **Dispositivos y herramientas sujetos a efectores finales**



Fuente: *Robots industriales*. <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r69369.PDF>. Consulta: 15 de julio de 2012.

3.3. **Microcontrolador**

Los microcontroladores son dispositivos semiconductores que a diferencia del microprocesador que necesita de dispositivos externos para su funcionamiento, por ejemplo de memoria, dispositivos de entrada/salida, periféricos, etc. Es un circuito integrado que incluye en su interior las tres unidades funcionales de una computadora; CPU, memoria y unidades de E/S (en otras palabras, es una computadora integrada en un chip de silicio). El tamaño de la CPU, la cantidad de memoria y los periféricos incluidos dependerán de la aplicación al cual el microcontrolador se someterá.

Un microcontrolador está compuesto de un generador de reloj integrado, una pequeña cantidad de memoria RAM (512 bytes) y una memoria de solo lectura (*flash* de 8k, 14 bits en cada posición). En resumen, solo se necesita un programa de instrucciones que se almacenará en la memoria *flash* y un cristal de sincronización.

Dispone, generalmente, también de una gran variedad de dispositivos de entrada/salida, convertidores de analógico-digital, temporizadores/contadores, UART's y buses de interfaz serie especializados, como I2C, SPI y CAN. Pueden ser controlados por instrucciones de procesadores especializados y aceptan una de dos tipos de instrucciones como lo son la tecnología RISC (*Reduced Instruction Set Computer*) Computadora con Juego de Instrucciones Reducidas que reconoce y ejecuta solo operaciones básicas y la tecnología CISC (*Complex Instruction Set Computer*) Computadoras con un Juego de Instrucciones Complejo que se caracteriza por ser muy amplio y permitir operaciones complejas entre operandos situados en la memoria o en los registros internos en contraposición a la arquitectura RISC.

El microcontrolador, frecuentemente, incluye un lenguaje de programación integrado de alto nivel como el BASIC, que se utiliza frecuentemente con este propósito para programar al microcontrolador. Este, a su vez, convierte el programa escrito en alto nivel a bajo nivel de manera que el microcontrolador lo entienda. En la tabla VII, se muestra la gama de microcontroladores y sus principales características.

Tabla VII. Gama de microcontroladores y principales características

Familia	ROM [Kbytes]	RAM [bytes]	Pines	Frecuencia de reloj. [MHz]	Entradas A/D	Resolución del convertidor A/D	Comparadores	Temporizadores de 8/16 bits	Comunicación serial	Salidas PWM	Otros
Arquitectura de la gama baja de 8 bits, palabra de instrucción de 12 bits											
PIC10FXXX	0.375 - 0.75	16 - 24	6 - 8	4 - 8	0 - 2	8	0 - 1	1 x 8	-	-	-
PIC12FXXX	0.75 - 1.5	25 - 38	8	4 - 8	0 - 3	8	0 - 1	1 x 8	-	-	EEPROM
PIC16FXXX	0.75 - 3	25 - 134	14 - 44	20	0 - 3	8	0 - 2	1 x 8	-	-	EEPROM
PIC16HVXXX	1.5	25	18 - 20	20	-	-	-	1 x 8	-	-	Vdd = 15V
Arquitectura de la gama media de 8 bits, palabra de instrucción de 14 bits											
PIC12FXXX	1.75 - 3.5	64 - 128	8	20	0 - 4	10	1	1 - 2 x 8 1 x 16	-	0 - 1	EEPROM
PIC12HVXXX	1.75	64	8	20	0 - 4	10	1	1 - 2 x 8 1 x 16	-	0 - 1	-
PIC16FXXX	1.75 - 14	64 - 368	14 - 64	20	0 - 13	8 or 10	0 - 2	1 - 2 x 8 1 x 16	USART I2C SPI	0 - 3	-
PIC16HVXXX	1.75 - 3.5	64 - 128	14 - 20	20	0 - 12	10	2	2 x 8 1 x 16	USART I2C SPI	-	-
Arquitectura de la gama alta de 8 bits, palabra de instrucción de 16 bits											
PIC18FXXX	4 - 128	256 - 3936	18 - 80	32 - 48	4 - 16	10 or 12	0 - 3	0 - 2 x 8 2 - 3 x 16	USB2.0 CAN2.0 USART I2C SPI	0 - 5	-
PIC18FXXJX	8 - 128	1024 - 3936	28 - 100	40 - 48	10 - 16	10	2	0 - 2 x 8 2 - 3 x 16	USB2.0 USART Ethernet I2C SPI	0 - 5	-
PIC18FXXKX	8 - 64	768 - 3936	28 - 44	64	10 - 13	10	2	1 x 8 3 x 16	USART I2C SPI	2	-

Fuente: Mikroelektornika. *Microcontroladores PIC*. <https://learn.mikroe.com/ebooks/microcontroladorespic/chapter/microcontroladores-pic/>. Consulta: 22 de julio de 2012.

3.3.1. Arquitectura del microcontrolador

El microcontrolador es un dispositivo que no necesita de dispositivos externos para su funcionamiento, internamente se encuentran los dispositivos de entrada/salida, memoria, entre otros.

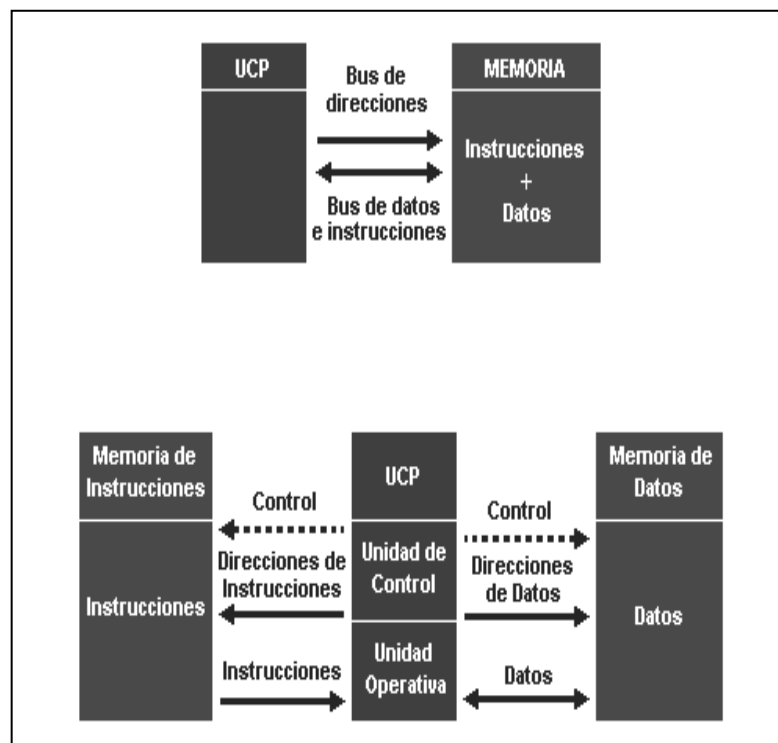
Los microcontroladores que utilizan la arquitectura von-Neumann disponen de un solo bloque de memoria y de un bus de datos de 8 bits. Como todos los datos se intercambian por medio de estas 8 líneas, este bus está sobrecargado haciendo la comunicación de estos dos muy lenta e ineficaz. La CPU puede únicamente leer una instrucción o leer/escribir datos en la memoria, el flujo de datos e instrucciones no pueden ocurrir al mismo tiempo puesto que las instrucciones y los datos utilizan el mismo bus; en otras palabras que, el mismo bus se utiliza para todas las operaciones.

Los microcontroladores que utilizan la arquitectura Harvard disponen de dos buses de datos diferentes. Un bus es de 8 bits de ancho y conecta la CPU con la memoria RAM; el otro consiste en un bus de varias líneas (12, 14 o 16 dependiendo del tipo de microcontrolador) que conecta la CPU y la memoria ROM que contiene el programa de instrucciones. La CPU puede leer las instrucciones y realizar el acceso a la memoria de datos a la vez. Otro dato importante que se tiene es que los datos en los registros de la memoria RAM son de 8 bits de ancho y todos los datos dentro del microcontrolador que se intercambian son de la misma anchura.

Todos los programas escritos para estos microcontroladores serán almacenados en la memoria ROM después de haber sido compilados a código máquina. Las localidades de memoria ROM no tienen 8, sino 12, 14 o 16 bits; de estas 4, 6 u 8 bits adicionales que representan una instrucción que

especifica a la CPU qué hacer con los datos de 8 bits. En la figura 66 se muestra la arquitectura von-Neumann y Harvard.

Figura 72. **Diseño de arquitectura para un microcontrolador, arquitectura von-Neumann (arriba) y arquitectura Harvard (abajo)**



Fuente: RUEDA, Luis. *Microcontroladores pic - capítulo 3*.

http://perso.wanadoo.es/luis_ju/pic/pic03.html. Consulta: 31 de julio de 2012.

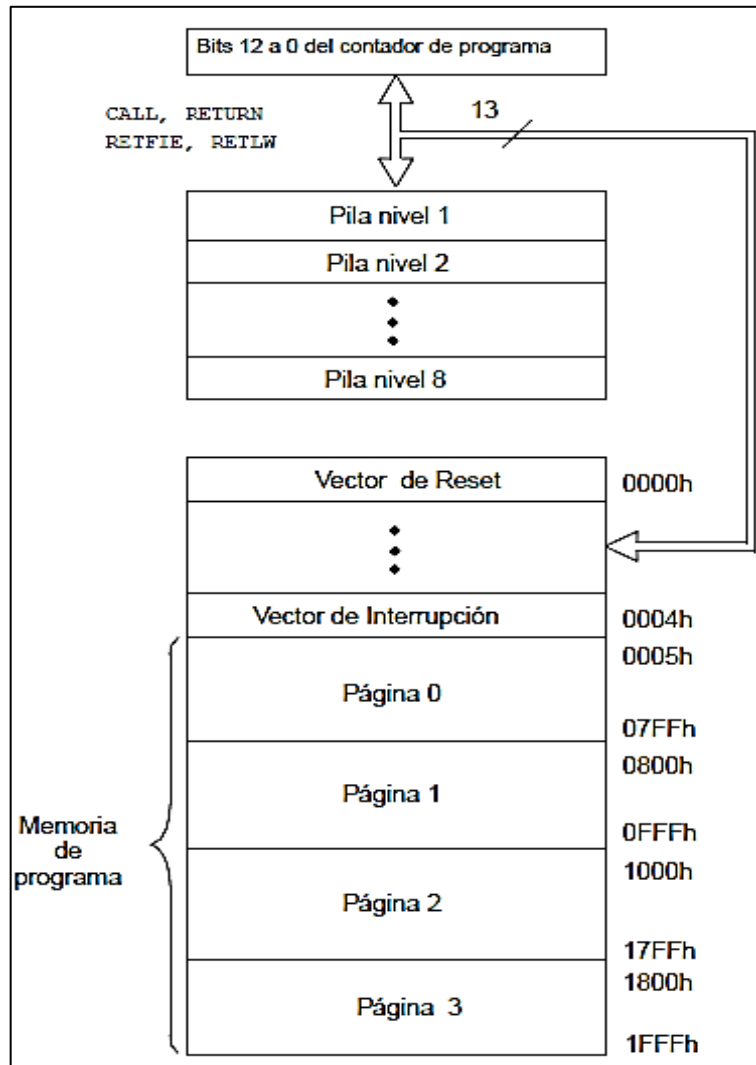
Las unidades de memoria se encuentran dentro del encapsulado del microcontrolador, para el PIC16F877 la memoria de programa ROM es del tipo de memoria *flash*, donde es posible escribir y borrar el contenido de esta memoria, prácticamente, un número ilimitado de veces.

Esta memoria tiene 8 K de longitud y 14 bits de datos en cada localidad de memoria. La memoria de programa está dividida en cuatro bancos de 2 K cada uno. El primero va de la posición de memoria 0000h a la 07FFh; el segundo va de la 0800h a la 0FFFh; el tercero de la 1000h a la 17FFh, y el cuarto de la 1800h a la 1FFFh.

En la figura 67, se muestra al vector de RESET que al aplicarle un *reset* al microcontrolador el contador de programa se coloca en 0000h y en esta dirección se colocará la dirección de inicio de programa. Otro dato importante es el vector de INTERRUPCIÓN que, cuando se recibe un interrupción el contador de programa, apunta a la dirección 04H en esta dirección se localizará el valor que especificará la localización deseada en la memoria de programa.

La PILA (*STACK*) es otra área de la memoria y se utiliza para guardar el valor del contador de programa cuando se hace un llamado a una subrutina o cuando se atiende una interrupción. Luego, cuando el microcontrolador regresa a ejecutar nuevamente el programa, el contador de programa recupera su valor leyéndolo nuevamente desde la pila. La pila de programa, como se puede observar en la figura 73, contiene ocho localidades de memoria capaz de apilar a 8 llamadas a subrutinas, también se muestra el diagrama para la memoria de programa del PIC16F877.

Figura 73. Diagrama de la memoria de programa para el 16F877



Fuente: *El microcontrolador pic16f877 resume de hoja de datos.* <http://www.utp.edu.co/~eduque/arquitect/PIC16F877.pdf>. Consulta: 3 de agosto de 2012.

La memoria RAM del PIC16F877 posee cuatro bancos de memoria de 128 byte cada una, y de estos, los primeros 32 byte son registros de propósitos especiales y que cumplen alguna función en el microcontrolador en la

configuración. Los 96 restantes son registros de propósito general que se utilizan para guardar datos que usa el programa que se está ejecutando.

Para acceder a cada banco de memoria se utilizan dos métodos, direccionamiento directo y direccionamiento indirecto. Para el direccionamiento directo se realiza por medio del registro STATUS y los bits RP0 Y RP1 y los 7 bits de código de operación que indican la posición del dato dentro del banco. El direccionamiento indirecto proporciona la dirección donde se encuentra la dirección del dato utilizando el registro INDF (registro no disponible físicamente). Al usar este registro se accede a los bits, 7 bits menos significativos de los 8 bits, que tiene registro SFR. Estos proporcionan la posición dentro del banco de memoria y el bit más significativo junto con el registro IRP del registro STATUS selecciona el banco. En la figura 68, se muestra el diagrama de la memoria RAM y sus registros correspondientes.

Un registro no es más que una celda de memoria que puede almacenar una serie de bytes de datos, dependiendo de su capacidad. Los registros SFR son un tipo de registro de funciones especiales con una función predeterminada por el fabricante, donde cada bit se encuentra conectado con los circuitos internos del dispositivo, por ejemplo, el convertidor ADC, temporizadores etc. Otro dato importante es que el estado de los bits del registro se fija dentro de programa, y estos registros dirigen los pequeños circuitos dentro del microcontrolador que se conectan por los pines de este a un dispositivo periférico. El PC (contador de programa) especifica la dirección de la instrucción que se ejecutará; el PC contiene 13 bits y con el byte más bajo con el que corresponde al registro de lectura y escritura PCL y al byte más alto que corresponde con el registro PCH que no se pueden leer y escribir directamente si no que son cargados, a través del registro PCLATH.

Tabla VIII. Diagrama y registros de la memoria RAM

INDF	00h	INDF	80h	INDF	100h	INDF	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR TRISA	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD	08h	TRISD	88h		108h		188h
PORTE	09h	TRISE	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reservado	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reservado	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h				
T2CON	12h	PR2	92h				
SSPBUF	13h	SSPADD	93h				
SSPCON	14h	SSPSTAT	94h				
CCPR1L	15h		95h				
CCPR1H	16h		96h				
CCP1CON	17h		97h				
RCSTA	18h	TXSTA	98h				
TXREG	19h	SPBRG	99h	Registros de Propósito General		Registros de Propósito General	
RCREG	1Ah		9Ah	16 Bytes		16 Bytes	
CCPR2L	1Bh		9Bh				
CCPR2H	1Ch		9Ch				
CCP2CON	1Dh		9Dh				
ADRESH	1Eh	ADRESL	9Eh				
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
	20h		A0h	Registros de Propósito General	120h	Registros de Propósito General	1A0h
Registros de Propósito General		Registros de Propósito General		80 Bytes		80 Bytes	
96 Bytes		80 Bytes					
			0EFh		16Fh		1EFh
			0F0h		170h		1F0h
			FFh		17Fh		1FFh
Banco 0		Banco 1		Banco 2		Banco 3	

Fuente: *El microcontrolador pic16f877 resume de hoja de datos*. <http://www.utp.edu.co/~eduque/arquitect/PIC16F877.pdf>. Consulta: 8 de agosto de 2012.

Otros registros de propósitos especiales o FSR son las siguientes:

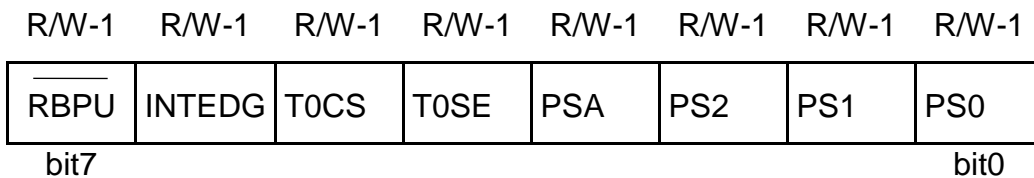
El registro de estado (*STATUS*) es un registro de 8 bits que contiene el estado de la UAL, al *RESET* y a la selección del banco de memoria de datos (RAM), que como se observa en la figura 68, este registro se encuentra en todos los banco y en la misma posición.

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C
bit7							bit0

- bit 7 IRP: Selección de bancos para el direccionamiento indirecto.
 1 = Banco 2,3 (100h – 1FFh).
 0 = Banco 0,1 (00h – FFh).
- bit 6:5 RP1:RP0: Selección del banco de la memoria de datos para a el direccionamiento directo.
 11 = Banco 3 (180h – 1FFh).
 10 = Banco 2 (100h – 17Fh).
 01 = Banco 1 (80h – FFh).
 00 = Banco 0 (00h – 7Fh).
- bit 4 \overline{TO} : *Timer Out*.
 1 = Tras conectar Vdd o ejecutar “CLRWDT” o “SLEEP”.
 0 = Al rebasar el WDT.
- bit 3 \overline{PD} : *Power Down*.
 1 = Tras conectar Vdd o ejecutar “CLRWDT”.
 0 = Al ejecutar la instrucción *SLEEP*.
- bit 2 Z: Bit de cero.
 1 = El resultado de una operación es 0.
 0 = El resultado es distinto de 0.
- bit 1 DC: Acarreo en el 4º bit de menos peso.
 1 = Acarreo en la suma y no en la resta.

- 0 = Acarreo en la suma y no en la resta.
- bit 0 C: Acarreo en el 8º bit.
- 1 = Acarreo en la suma y no en la resta.
- 0 = Acarreo en la suma y no en la resta.

El registro de opciones (*OPCTION_REG*). Este registro contiene varios bits para configurar el pre-escalador o divisor de frecuencia del temporizador TMR0/WDT. El temporizador *Timer 0*, la interrupción externa y la configuración del puerto B con resistencia *pull-up*.



- bit 7 RBPU . Conexión de cargas *Pull-Up* para la Puerta B.
 - 1 = Todas las cargas *Pull-Up* desconectadas.
 - 0 = Todas las cargas *Pull-Up* conectadas.
- bit 6 INTEDG. Tipo de flanco para la interrupción.
 - 1 = RB0/INT sensible a flanco ascendente.
 - 0 = RB0/INT sensible a flanco descendente.
- bit 5 T0CS. Fuente de reloj para el TMR0.
 - 1 = Pulsos introducidos por T0CKI (contador).
 - 0 = Pulsos de reloj interno $F_{osc}/4$ (temporizador).
- bit 4 T0SE. Tipo de flanco activo del T0CKI.
 - 1 = Incremento del TMR0 cada flanco descendente.
 - 0 = Incremento del TMR0 cada flanco ascendente.
- bit 3 PSA. Asignación del divisor de frecuencia.
 - 1 = Se le asigna al WDT.
 - 0 = Se le asigna al TMR0.

bit 2-0 PS2:PS0: Valor del divisor de frecuencia.

Valor	División del TMR0	División del WDT
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

El registro palabra de configuración, se utiliza cuando se programa el microcontrolador, según la aplicación a la que se someterá. Es un registro de 14 bits y se configura durante la fase de programación y no durante la ejecución del programa.

CP1	$\overline{\text{CP0}}$	$\overline{\text{CP1}}$	CP0	CP1	CP0	-	BODEN
-----	-------------------------	-------------------------	-----	-----	-----	---	-------

bit13

CP1	CP0	$\overline{\text{PWRTE}}$	WDTE	FOSC1	FOSC0
-----	-----	---------------------------	------	-------	-------

bit0

bit 13-8 CP<1:0>. Bits de protección de código.

bit 5-4	512	1k	2k
11	Off	Off	Off
10	Off	Off	½ alta On
01	Off	½ alta On	¾ alta On
00	On	On	On

bit 7 No implementado. Su lectura es 1.

- bit 6 BODEN. Detección del *Brown-Out* (Fallo de alimentación).
 1 = detección activada.
 0 = Detección desactivada.
- bit 3 PWRTE. Activación del temporizador *Power-Up*.
 1 = Desactivado.
 0 = Activado.
- bit 2 WDTE. Activación del “*Watchdog*” (perro guardián).
 1 = Activado.
 0 = Desactivado.
- bit 1-0 FOSC1-FOSC0: Selección del tipo de oscilador.
 11 = Oscilador RC.
 10 = Oscilador HS.
 01 = Oscilador XT.
 00 = Oscilador LP.

Los registros del sistema de interrupción necesitan ser habilitados por el usuario en el programa con los bits IE (*interrupt enable*) y un bit de bandera IF (*interrupt flag*). Hay dos formas de activar la interrupción, la E/S controlada por programa hace una prueba continua en un bucle el estado de los elementos externos o periféricos y la E/S controlado por interrupciones en donde los elementos externos o periféricos le indican al microcontrolador cuando está preparado para realizar la interrupción, el microcontrolador continúa realizando la tarea asignada, pero después atiende la interrupción que se le ha solicitado, para luego continuar con la ejecución del programa. Las fuentes posibles de interrupción son las siguientes:

- Interrupción por flanco en el pin RB0/INT (interrupción externa).
- Interrupción por rebose del TMR0.
- Interrupción por cambio de nivel en PORTB(RB7:RB4).

- Interrupción por cambio en el comparador.
- Interrupción del puerto paralelo esclavo.
- Interrupción de la USART (tanto recepción como transmisión).
- Interrupción por finalización de conversión A/D.
- Interrupción de LCD.
- Interrupción por finalización de escritura en memoria de datos EEPROM.
- Interrupción por rebose del TMR1.
- Interrupción por rebose del TMR2.
- Interrupción del módulo CCPx.
- Interrupción del módulo SSP (comunicación serie).

Dentro de los registros que se utilizan para el sistema de interrupción se encuentra el registro INTCON que contiene las habilitaciones, señalizadores de interrupciones más comunes y las habilitaciones globales de las interrupciones.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GIE	PEIE ⁽³⁾	TOIE	INTE ⁽²⁾	RBIE ^(1,2)	TOIF	INTF ⁽²⁾	RBIF ^(1,2)
bit7							bit0

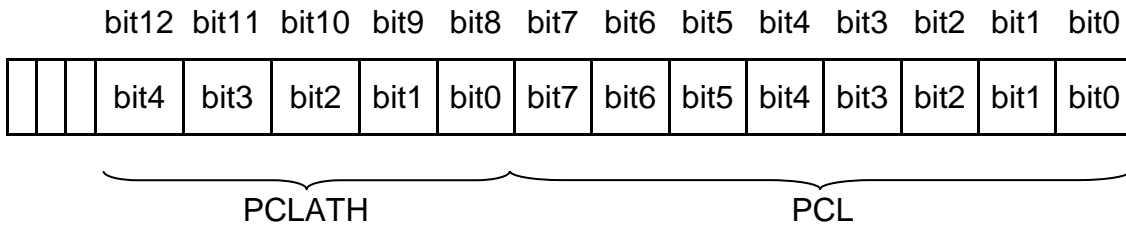
- bit 7 GIE. Bit de habilitación global de interrupciones.
 1 = Interrupciones habilitadas.
 0 = Interrupciones deshabilitadas.
- bit 6 PEIE. Bit de habilitación de interrupciones de los periféricos.
 1 = Habilitadas interrupciones de los periféricos.
 0 = Deshabilitadas interrupciones por los periféricos.
- bit 5 TOIE. Habilitación de interrupción por rebose del TMR0.
 1 = Habilitada interrupción por rebose del TMR0.
 0 = Deshabilitada interrupción por rebose del TMR0.
- bit 4 INTE. Habilitación de interrupción por flanco en RB0/INT.

- 1 = Habilitada.
 - 0 = Deshabilitada.
- bit 3 RBIE(1). Habilitación de interrupción por cambio de estado en PORTB<7:4>.
 - 1 = Habilitada.
 - 0 = Deshabilitada.
- bit 2 T0IF. Señalizador de rebose del TMR0.
 - 1 = TMR0 ha rebosado. Se borra por *software*.
 - 0 = No se ha producido rebose.
- bit 1 INTF. Señalizador de interrupción por flanco en RB0/INT.
 - 1 = La interrupción se ha producido. Se borra por *software*.
 - 0 = No se ha producido interrupción.
- bit 0 RBIF(1). Señalizador de cambio de estado en PORTB<7:4>.
 - 1 = Ha cambiado de estado alguna de las líneas, RB<7:4>.
 - 0 = No ha cambiado ninguna entrada.

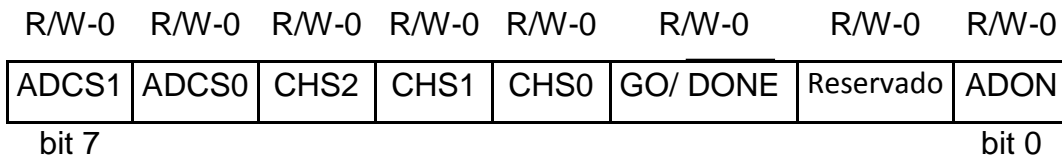
Otros registros involucrados para la interrupción del dispositivo son los registros PIE1 y PIE2. Estas interrupciones son producidas por dispositivos periféricos y, que la configuración de estos registros sean válidos, el bit PEIE del registro INTCON debe estar en 1. Los registros PIR1 y PIR2 contienen los bits de bandera de interrupción.

El registro PC (contador de programa), debe de ser de 13 bits de anchura (2¹³ = 8192 para localidades) y para acceder a una localidad de memoria se utilizará el registro FSR. Los registros SFR son de 8 bits de anchura y el registro PC se dividirá en dos registros virtuales que son el registro PCLATH y el registro PCL. Este registro se utiliza principalmente, cuando se programa en lenguaje ensamblador con las instrucciones GOTO, CALL, RETURN, RETLW o RETFIE, para saltos y subrutina. El registro PCL es de

lectura/escritura y mientras que el PCLATH es de solo escritura. Con los bits 4 y 3 del registro PCLATH se accede a los bancos de memoria (00 para banco 0 y 11 para el banco 3) y se borra únicamente con cada reinicio.



Otros registros de interés son el ADCON0, ADCON1 Y ADRES (ADRESL Y ADRESH) para la configuración y aplicación en la utilización del módulo de conversión ADC. Cuando se realiza una conversión de analógica a digital, el resultado obtenido después de la conversión es un número de 10 bits. Este valor se almacena en el registro ADRES ya sea justificación a la derecha o justificación a la izquierda, dado por el bit ADFM del registro ADCON1. La descripción del registro ADCON0, se explica a continuación:



- bits 7:6 ADCS1:ADCS0: Bits de selección del reloj para la conversión A/D.
 - 00 = FOSC/2.
 - 01 = FOSC/8.
 - 10 = FOSC/32.
 - 11 = FRC (Reloj del oscilador interno RC).
- bits 5:3 CHS2:CHS0: Bits de selección del canal.
 - 000 = Canal 0, (AN0).
 - 001 = Canal 1, (AN1).

000 = Canal 2, (AN2).

000 = Canal 3, (AN3).

000 = Canal 4, (AN4).

000 = Canal 5, (AN5).

000 = Canal 6, (AN6).

000 = Canal 7, (AN7).

- bit 2 $\overline{\text{GO/DONE}}$: Bits de estado de la conversión
Cuando $\text{ADCON} = 1$.
1 = Conversión en proceso (poner a '1' para iniciar una conversión).
0 = La conversión ha finalizado.
- bit 1 Reservado: Siempre se mantiene a cero.
- bit 0 ADON: Activación del conversor A/D.
1 = El comparador está activado.
0 = El comparador no está activado.

Se presenta la descripción del registro ADCON1 .

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
--	--	--	--	--	PCFG2	PCFG1	PCF0
bit 7					bit 0		

- bits 7:3 Bits no implementados: Se leen como '0'.
- bit 2:0 PCFG2:PCFG0: Bit de configuración de los pines del puerto.

Tabla IX. **Configuración de puertos E/S, como analógicos o digitales**

PCFG2: PCFG0	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
0 0 0	A	A	A	A	A	A	A	A
0 0 1	A	A	A	A	VREF	A	A	A
0 1 0	D	D	D	A	A	A	A	A
0 1 1	D	D	A	A	VREF	A	A	A
1 0 0	D	D	D	D	A	D	A	A
1 0 1	D	D	D	D	VREF	D	A	A
1 1 x	D	D	D	D	D	D	D	D

Fuente: SALAMANCA MIÑO, Santiago y ARROYO MUÑOZ, David. *Los pics de gama media arquitectura y técnicas de programación*. <http://wiki.webdearde.com/images/1/1b/Pic-gama-media.pdf>. Consulta: 29 de agosto de 2012.

Existen varios módulos con que se puede trabajar con el microcontrolador entre los que se pueden mencionar: el ADC, contador/temporizador, USB (dependiendo de microcontrolador a utilizar), el módulo CCP (*Captura/Comparación/PWM*), módulo EUSART, módulo de memoria micro SD, etc. El microcontrolador PIC16F887 dispone de dos módulos CCP-CCP1 y CCP2 en que el modo de funcionamiento de interés será el PWM. El módulo CCP1 consta de un registro CPR1 de 16 bits (divido en CPR1H y CPR1L) capaz de generar señales de frecuencia y ciclos de trabajo variado para el control de potencia que se podría utilizar para activar a los motores DC.

El modo de funcionamiento del PWM determinar, la frecuencia con la utilización de los registros PR2 del temporizador timer2 y el ciclo de trabajo con los registros CPR1L (8 bits) y el registro CCP1CON (los dos bits menos significativo DC1B1 y DC1B0) al utilizar la siguientes ecuación y registro CCPCON.

$$\text{Periodo PWM} = (PR2 + 1) \cdot 4T_{osc} \cdot \text{Valor de pre escala del Timer2}$$

[Ec. 3.40]

Ancho de pulso

$$= (CCPR1L, DC1B1, DC1B0) \cdot T_{osc} \cdot \text{Valor de pre escala del Timer2}$$

[Ec. 3.41]

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
--	--	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7						bit 0	

bit 7:6 Bits no implementados: Se leen como '0'.

bit 5:4 DCxB1:DCxB0: Bit1 y bit0 del Duty Cycle del PWM.

Modo captura:

No se usa.

Modo comparación:

No se usa.

Modo PWM:

Son los dos bits menos significativos de los 10 bits que se utilizan para el *Duty Cycle* del PWM. Los ocho bits superiores.

(DCx9:DCx2) del *Duty Cycle* se encuentran en CCPxL.

bit 3:0 CCPxM3:CCPxM0. Bits de selección del modo CCPx.

0000 = Captura/Comparación/PWM apagado (*reset* del módulo CCPx).

0100 = Modo de captura, cada flanco de bajada.

0101 = Modo de captura, cada flanco de subida.

0110 = Modo de captura, cada 4 flancos de subida.

0111 = Modo de captura, cada 16 flancos de subida.

1000 = Modo de comparación, inicializar el pin CCP en bajo y forzarlo a alto cuando se produzca la igualación.

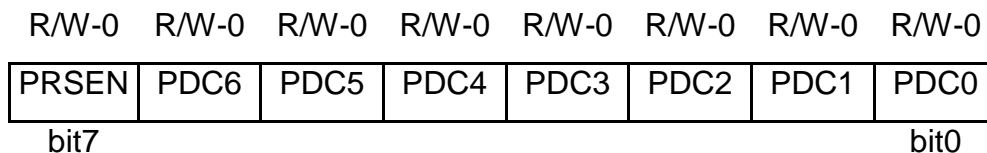
1001 = Modo de comparación, inicializar el pin CCP en alto, y forzarlo a bajo cuando se produzca la igualación.

1010 = Modo de comparación, el pin CCP no se ve afectado.

1011 = Modo de comparación.

11xx = Modo PWM.

Otro de los registros que se utiliza para el funcionamiento del módulo CCP, para el modo PWM es el PWM1CON.



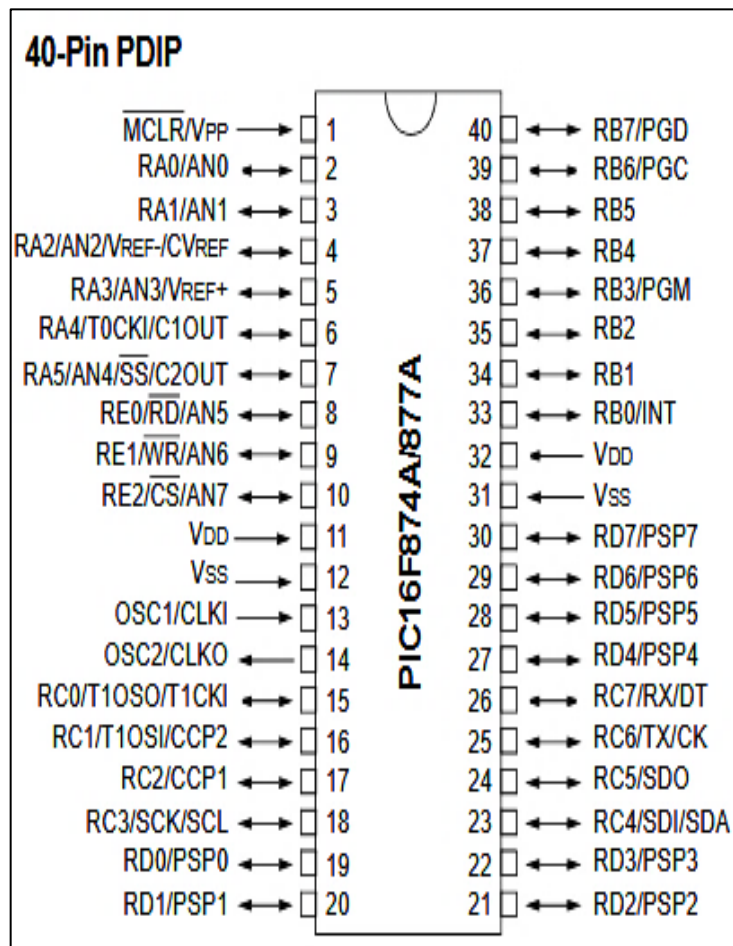
Algunos registros de propósitos especiales actúan directamente sobre el funcionamiento del microcontrolador, a través de los circuitos internos del mismo, así como el tipo de memoria que utiliza.

Existen tres registros importantes TRIS, PORT y ANSEL.

- El registro PORT es un registro bidireccional que se comportan como entrada/salida necesarias para que el microcontrolador se comunique con los dispositivos externos.
- El registro TRIS determina cuáles pines son configuradas como entradas o salidas.
- El registro ANSEL similar al registro TRIS, determina si los pines serán configurados como entradas analógicas o entradas/salidas digitales.

En la figura 74, se muestra al microcontrolador PIC16F877 y su configuración.

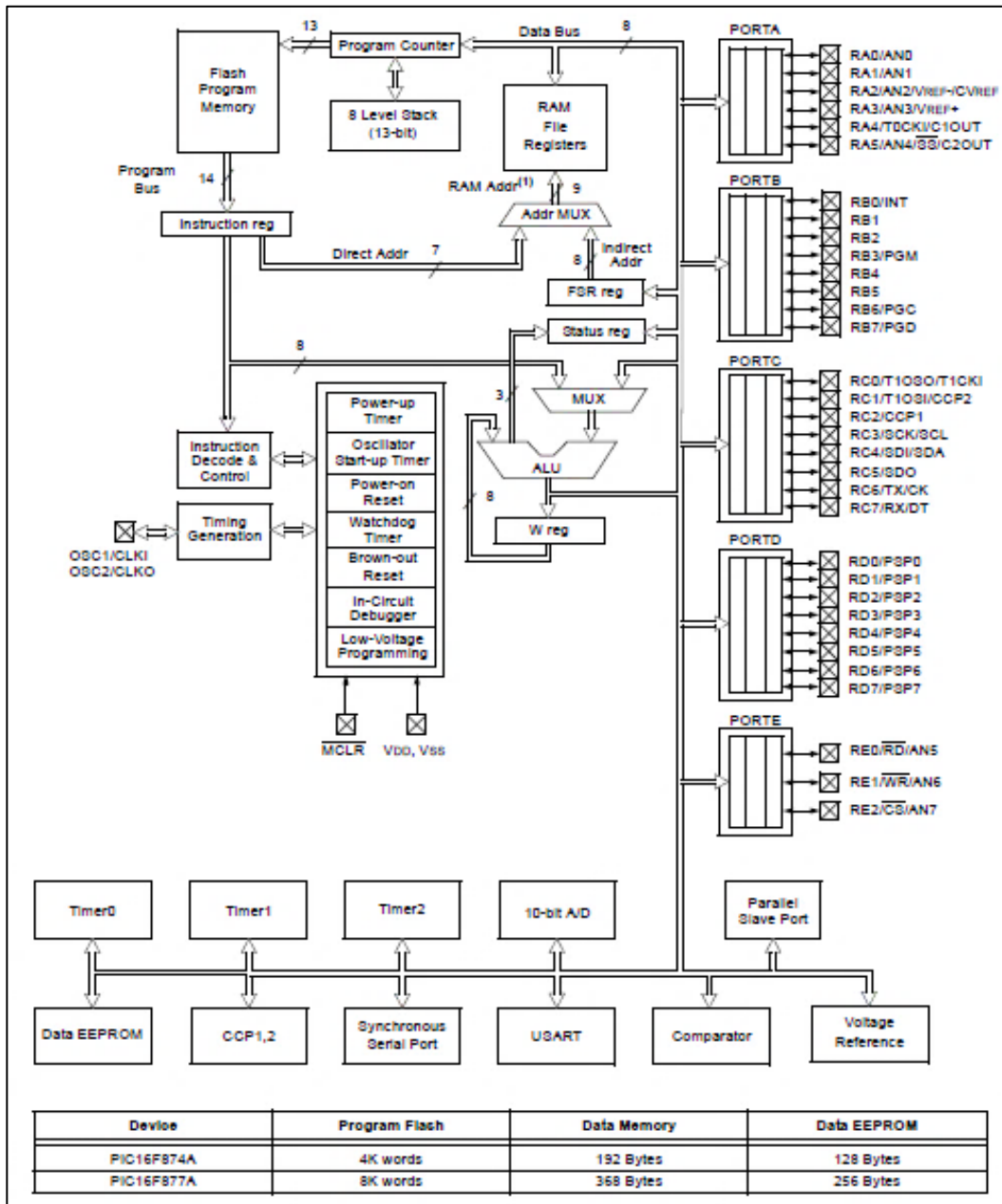
Figura 74. Diagrama de conexión del PIC16F877



Fuente: *Microchip*. <http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>. Consulta: 8 de septiembre de 2012.

En la figura 75, se muestra la arquitectura del microcontrolador.

Figura 75. Arquitectura del microcontrolador PIC16F877



Fuente: *Microchip*. <http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>. Consulta: 9 de septiembre de 2012.

3.3.2. Integración del microcontrolador

Los diferentes módulos que contiene el microcontrolador permiten tener muchas aplicaciones para el control y manejo de dispositivos como el sensor de temperatura, motores, dispositivos de comunicación, entre otros. Los pines del microcontrolador son de cierta manera multifuncionales (referencia 26 y 27), para los actuadores se podrán usar impulsores o fuentes de energía eléctrica, hidráulica o neumática. Para los actuadores eléctricos, se realizará con motores de corriente alterna, motores de corriente continua con escobillas o sin ellas y los motores *stepper*.

Para manejar el motor de corriente alterna se podría utilizar algún puerto del microcontrolador funcionando como entrada/salida digital y utilizando un semiconductor como el Triac, Diac o relé para entregar la potencia necesaria al motor. Cuando al motor se le aplica el voltaje el motor empezará a girar y, para detener la marcha, será necesaria eliminar la alimentación de voltaje al motor. Para este caso, es necesario utilizar un reductor de velocidad compuesto por un tornillo sin fin y un engranaje o algún otro sistema reductor para conservar la posición cuando el motor deje de girar, debido a la ausencia de voltaje. Existen muchos sistemas reductores comercialmente; sin embargo, el autor de este estudio opina que el sistema tornillo sin fin y engranaje conforma un buen sistema que conserve la posición deseada cuando el motor no este girando,

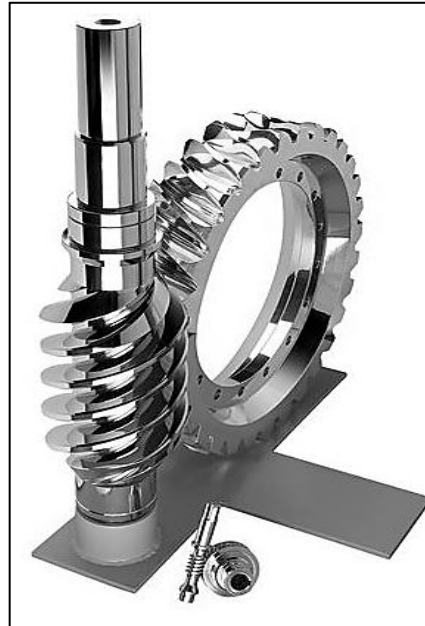
Para el manejo de motores de corriente continua, se puede utilizar nuevamente cualquiera de los puertos del microcontrolador. Estos puertos o pines mantienen su estado hasta un nuevo cambio y con estos valores se controlará a los dispositivos que proporcionará la potencia necesaria a los motores. Para proporcionar esta potencia se utilizarán transistores como switch (en sus regiones corte y saturación) y relés.

Los motores DC y AC necesitan de reductores para proporcionar un mayor torque de salida y mantener su posición cuando al motor no se le aplique voltaje, que es una de las formas más simples para detener la marcha del motor. Para un proyecto pequeño se puede utilizar los motores *stepper* o motores paso a paso que, como se sabe, estos motores mantienen su posición cuando se le aplica voltaje y produce un giro en un grado determinado cuando se le cambia la configuración de voltajes en su entrada.

Para el accionamiento hidráulico o neumático, se utilizará electroválvulas con voltaje de operación: 24VCD, 24VCA, 110VCA y 220VCA con el empleo de *buffers* de salida a colector abierto, los cuales pueden activarse con bajos voltajes y bajas corrientes; además, se puede controlar a la salida con cargas inductivas a mayores voltajes o amplificar la señal de control con una etapa amplificadora transistorizada. Estas válvulas entregan la potencia necesaria para activar al cilindro de simple o doble acción o a un motor hidráulico de paletas, para realizar el movimiento en las articulaciones del robot, el cual es controlado por el microcontrolador con algunos componentes electrónicos externos como los transistores, contactores, entre otros. El microcontrolador opera y controla los actuadores con la selección apropiada de fuente de energía.

En la figura 76, se muestra la figura del reductor de velocidad tornillo sin fin y rueda dentada en donde el tornillo sin fin, que es el elemento motriz y el elemento conducido, es la rueda dentada (nunca a la inversa) se podría aplicar a los actuadores eléctricos.

Figura 76. **Reductor de velocidad tornillo sin fin y rueda dentada**



Fuente: *Direct industry*. <http://www.directindustry.es/prod/gfc-antriebssysteme-gmbh/reductores-ortogonales-de-tornillo-sin-fin-37980-377728.html>. Consulta: 16 de septiembre de 2012.

El robot industrial, además de necesitar dispositivo que le proporcionen movimiento, necesita de algún otro dispositivo para posicionarse y orientarse. Los dispositivos o sensores que pueden medir el desplazamiento circular son los potenciómetros, sensores de inclinación o encoders. Para la utilización de los potenciómetros y de los sensores de inclinación (si así se requiere) se aconseja utilizar el convertidor de análogo a digital del microcontrolador, es decir, el PIC16F877 que cuenta con 8 entradas ADC de 10 bits marcadas como AN, para el potenciómetro se necesitará la programación para interpretar los valores de voltaje en una variación de posición circular (como un cuantificador).

Para el inclinómetro se utilizaría un convertidor de corriente a voltaje con amplificadores operacionales, ya que el valor de la salida del dispositivo es un rango de corriente que se encuentra entre 4 a 20 mA proporcional al seno del ángulo de inclinación para el sensor AIT435 "L" y la utilización del ADC del microcontrolador. El dispositivo con menos complejidad de programación y sin necesidad de utilizar circuitos adicionales es el encóder absoluto con salida binaria. Estos dispositivos manejan un rango de operación de 360° con 8, 10, 12,16 o 24 bits de resolución y solo necesita la programación suficiente para posicionar y orientar al robot.

Los sensores de fuerza, proximidad, de contacto y táctil se utilizan, así el efector final interactúa con su entorno. Estos sensores convierten una magnitud física en una señal eléctrica, estos medirán magnitudes de voltaje continuo, voltaje alterno a una determinada frecuencia o magnitudes de corriente y todo dependerá de la aplicación que al robot se le someterá.

El principal interés es posicionar y orientar al robot, así como proporcionar el mayor torque en las articulaciones (con algunos componentes adicionales, y algunas estructuras adicionales, se logrará mayor rendimiento de la misma) y por esta razón, en este trabajo se dará importancia principal a la programación de la misma.

4. DISEÑO E IMPLEMENTACIÓN CON EL MICROCONTROLADOR

4.1. Consideraciones de diseño

Los elementos que componen el robot y la aplicación que tendría influyen para considerar el tipo de sistema de impulsión, así como los actuadores. Otros componentes indispensables que permiten al robot interactuar con su entorno son los dispositivos sensoriales. Sin ello, no se logrará la comunicación con el espacio físico. Cada aplicación específica existe un tipo de sensor que será capaz de lograr la comunicación del robot para que este ejerce la tarea asignada.

Se tiene que considerar el peso de la estructura, la forma del robot y sus dimensiones, para lograr un mayor torque en las articulaciones es necesario la utilización de un sistema reductor, un sistema servomotor o un sistema tornillo sin fin (elemento de entrada), así como la rueda dentada (elemento de salida) en el que en ausencia de voltaje al motor la estructura mantendrá su posición. Este sistema permite el movimiento del tornillo a la rueda y nunca de la rueda al tornillo.

Es importante que el microcontrolador tenga suficiente capacidad de memoria interna, también si la aplicación así lo requiere se utilizará algún dispositivo de memoria externa como un dispositivo USB o microSD, donde se almacenen los datos de aplicación. Asimismo se debe considerar en la selección, la memoria de programa, cantidad de puertos de entrada/salida y los diferentes módulos que contiene el microcontrolador.

Características del PIC16F877A:

- CPU RISC de alto desempeño.
- 35 instrucciones por aprender.
- Hasta 20 MHz de velocidad (200 nS por instrucción).
- *Flash* 8Kb x 14 bit.
- RAM 368 x 8 (datos).
- EEPROM 256 x 8 (datos).
- Hasta 14 fuentes de interrupción.
- Stack de 8 niveles.
- Power on reset (POR).
- Power on timer (PWRT).
- Oscillator start Up (OST).
- Watchdog timer (WDT).
- Sleep mode.
- CMOS/FLASH EEPROM de bajo consumo, alta velocidad.
- Amplio voltaje de operación 2.0 a 5.5 V.
- Corriente de pines de hasta 25 mA.
- Módulos PWM, capture y comparador 10 y 16 bit.
- Convertidor A-D 8 entradas, 10 bit.
- I2C.
- USART/SCI.

La correcta selección del sistema de impulsión y del tipo de actuador permitirá determinar la capacidad de carga del robot. Los sistemas hidráulicos tienen más capacidades de carga, que el sistema neumático que soportan menor carga hasta 10 kg. Para los sistemas, donde la precisión es de mayor importancia, se emplearán los actuadores eléctricos con precisiones inferiores a

1 mm si se eligen accionamientos eléctricos. Los más rápidos son los accionamientos neumáticos (donde no hay retardos desde que llega la presión hasta que se mueve el eje) y los más lentos los sistemas eléctricos responden de mejor forma a los cambios de sentido y los que mantienen la velocidad constante frente a variaciones de la carga.

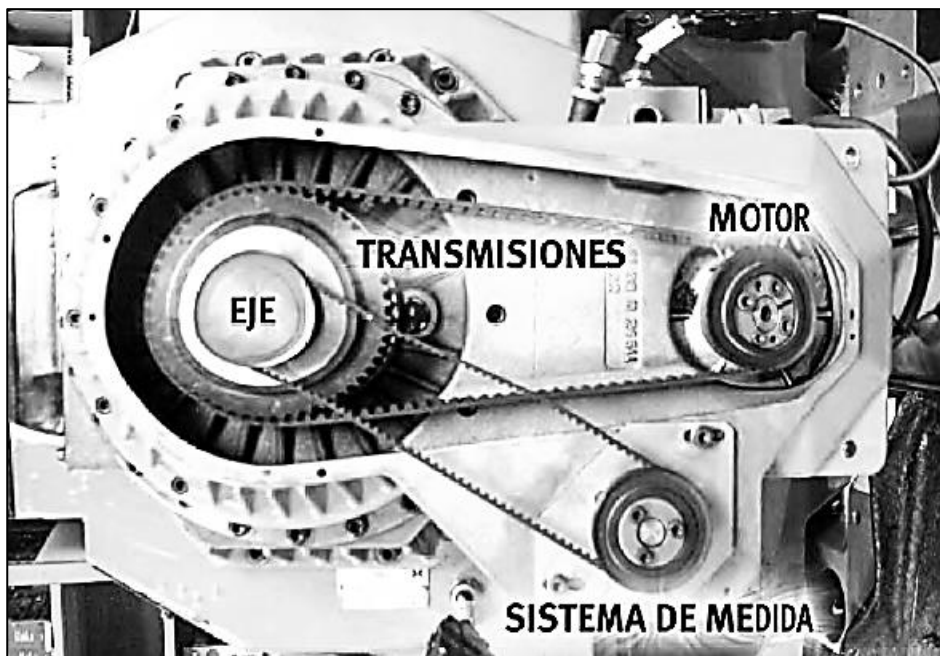
Se consideran las características del robot KUKA con seis grados de libertad. En este robot, el cuerpo base está fabricado con fundición de metal liviano, con alta resistencia a las vibraciones. Las articulaciones y los reductores se mueven prácticamente libres de juegos y todas las piezas en movimiento están cubiertas, los motores de accionamiento son servomotores de CA sin escobillas de baja inercia de mando transistorizado, libres de mantenimiento asegurados contra sobrecargas. Los ejes principales son del tipo de lubricación permanente, por ello es necesario un cambio de aceite solo después de 20 000 horas de servicio industrial; la medición de los trayectos para los ejes se realiza, a través de un sistema de medición absoluto con un resolver para cada eje y en las unidades motrices van integrados el resolver y el freno.

Si se presta atención a las características del robot KUKA, las articulación en las que los actuadores (motores AC sin escobillas) son utilizados conjuntamente con un sistema reductor y, para mantener al brazo en una posición determinada, se utiliza un sistema de frenado, sin la necesidad de utilizar un sistema tornillo sin fin y rueda dentada, que es otra variante de mantener la posición del robot en cualquier punto del espacio, pero con capacidad de carga menor.

El sistema de motor-reductor-freno se utiliza cuando la programación es de forma guiada. Para el sistema tornillo-rueda dentado, se podrá utilizar cuando la programación es forma textual o cualquier otra variante de sistema.

Otra forma de medir el desplazamiento sin la utilización de resolver, es utilizando un encóder absoluto para la facilidad de manejo o algún sistema de resistencia variable (menos efectivo, ya que produce desgaste) como el potenciómetro. En la figura 77, se muestra un sistema mecánico que compone un sistema de articulación.

Figura 77. **Sistema actuador, reductor y medición para un robot industrial comercial**



Fuente: GARCÍA, Salvador, MARTÍNEZ, José y POZANCO, Juan. *Robótica general*.
http://www.academia.edu/10560493/ROB%C3%93TICA_GENERAL_Autores. Consulta: 11 de octubre de 2012.

Para posicionar el extremo de la estructura en cualquier punto del espacio, se logra con el control de las primeras tres articulaciones (articulaciones primarias). Luego de haber posicionado al robot es necesario orientarlo por medio de las tres últimas articulaciones (articulaciones secundarias)

correspondientes a la muñeca. La posición y orientación del robot se realiza por medio de la secuencia pasos que previamente se estableció en la programación, los dos métodos de programación son las siguientes:

- Programación guiada.
- Programación textual.

En la programación guiada se tiene una maqueta del robot o al robot en la que se va realizando una tarea y programando (guardando en memoria) los puntos o movimientos del robot para su posterior repetición en forma automática, dentro de la programación pasiva se encuentran:

- Guiado pasivo directo.
- Guiado pasivo por maqueta.
- Guiado activo básico.
- Guiado activo extendido.

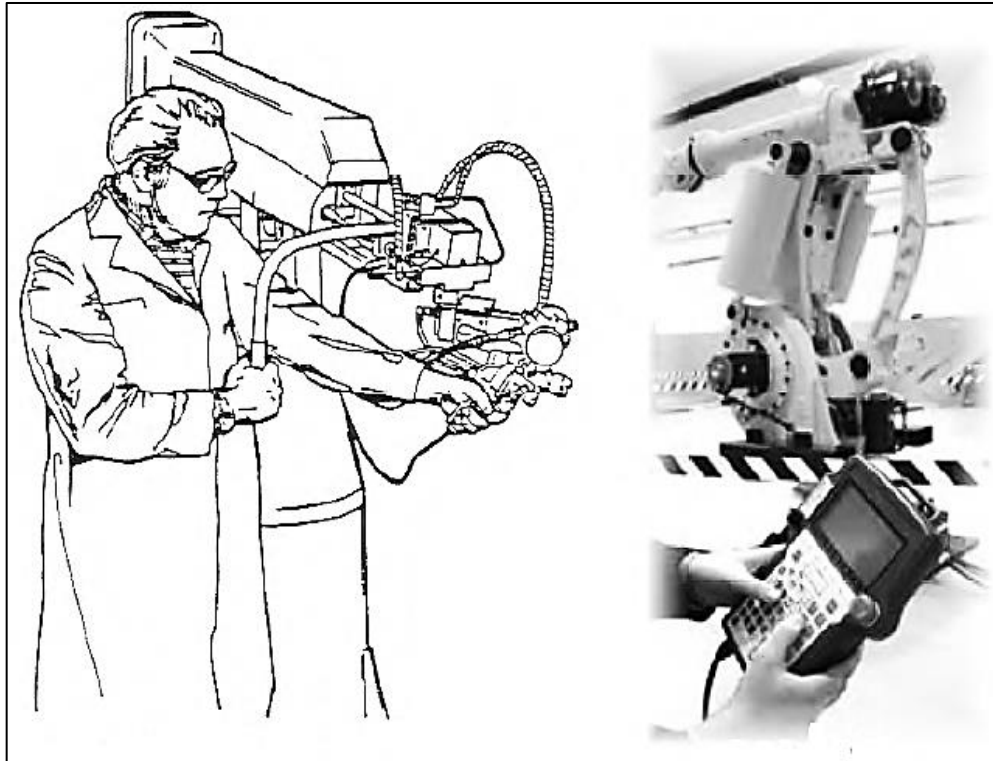
En la programación pasiva directa los actuadores se encuentran desconectados y el programador proporciona la energía de forma directa para mover al robot, tomar el extremo y llevarlo a los puntos deseados por la mejor trayectoria. La unidad de control del robot registra de manera automática la señal de los sensores de posición de las articulaciones en todos los puntos recorridos. La programación pasiva por maqueta, se da cuando el robot es muy pesado y difícil para que el programador la pueda mover. Esta maqueta es muy similar al robot que se pretende programar, pero más ligera y fácil de mover. La programación de esta es igual que la guiada directa con la excepción de utilizar un doble con configuraciones idénticas al robot que se pretende programar.

La programación guiada activo permite utilizar el propio sistema de impulsión de robot para proporcionar el movimiento necesario controlado, por un sistema de botones o por una palanca de mando (joystick) y llevar al robot por la trayectoria deseada para su posterior repetición. En la programación activo básico, el robot es guiado por los puntos en los cuales se desea que pase durante la fase de ejecución automática del programa, el sistema de control haga recorrer los puntos en forma secuencialmente en el mismo orden en que se programaron.

La programación guiada activo extendido es en cierta manera, más compleja, pero con mayor control, que permite especificar los puntos; por ello deberá pasar el robot, datos relativos a la velocidad, tipo de trayectoria, precisión, con esto se quiere alcanzar los puntos, control del flujo del programa y atención a entradas/salidas, entre otros.

En la programación textual se le indica la tarea al robot, por medio de una secuencia de instrucciones, a través de un lenguaje de programación. Actualmente se utiliza la programación textual del robot, donde las órdenes se refieren a los movimientos a realizar por el robot. Se debe especificar cada uno de los movimientos que ha de realizar el robot, como velocidad, direcciones de aproximación y salida, apertura y cierre de la pinza, etc. También es necesario descomponer la tarea global en varias sub tareas, ya que en este tipo de programación se debe de realizar en forma secuencial. En la figura 78, se muestran las figuras para la programación guiada pasiva y la programación guiada activa.

Figura 78. **a) Programación guiada pasiva directa y b) programación guiada activa**



Fuente: GARCÍA, Salvador, MARTÍNEZ, José y POZANCO, Juan. *Robótica general*.
http://www.academia.edu/10560493/ROB%C3%93TICA_GENERAL_Autores. Consulta: 23 de octubre de 2012.

4.2. Lenguaje MikroBasic

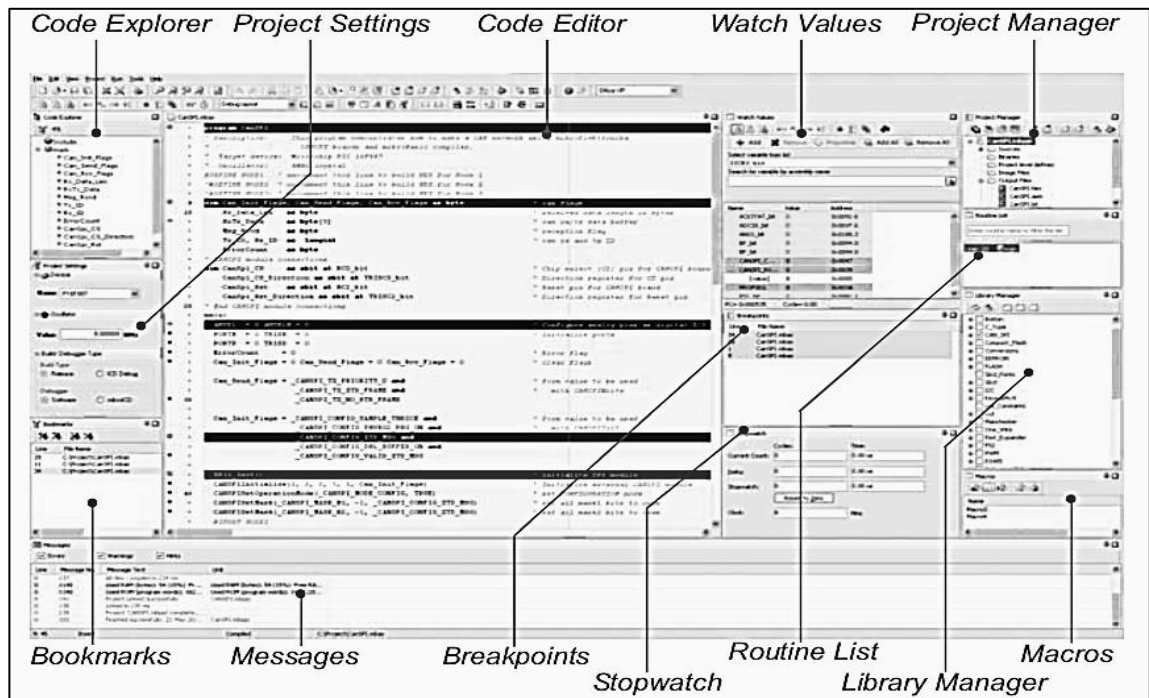
Todos los microcontroladores se programan a nivel de máquina (unos y ceros), un programa de nivel bajo como el ensamblador (*assembler* o *assembly*), que luego se compila para producir el lenguajes máquina, que es interpretado por el microcontrolador como instrucciones. Hay lenguajes de programación de alto nivel que al compilarlos traducen las instrucciones legibles

y escritas, por el programador a lenguajes máquina. Algunos de los lenguajes de alto nivel son: MikroBasic, MikroPascal y MikroC.

MikroBasic es muy similar a lenguaje de programación Visual Basic. En la figura 79, se muestra la pantalla principal del compilador en la que se detallan algunas de las ventanas principales. El explorador de código permite ir en cualquier parte del código del programa escrito, la ventana editora de código permite editar el programa y se accede a ella con F12 o desde el menú *tool*, la ventana de error muestra, los errores que hay posterior al compilado donde muestra el número de línea y el error de sintaxis; el código auxiliar se accede a ella por medio de Ctrl + espacio y permite completar alguna sentencia, basta con solo escribir las primeras letras.

La ventana de puntos de interrupción proporciona información sobre las interrupciones, lo que hace es justamente interrumpir la ejecución del programa, que se pueda analizar lo que el programa está haciendo. Otra de las ventanas importantes es la ventana de guardia (*watch widow*); esta ventana se utiliza para depurar el programa y se utiliza como simulador, que se puede monitorear el programa comportándose de la misma forma como lo haría en el microcontrolador. Los espacios en blanco se utilizan para indicar dónde empieza y termina un token (cadena de caracteres que tiene un significado en un programa) más de un espacio en blanco es descartado, los comentarios que son una cadena de texto para indicar anotación en el programa y son solo para el programador usando el apóstrofo “ ’ ”. Los programas que se mencionaron anteriormente utilizan valores hexadecimales y no decimales ni binarios.

Figura 79. **Ventana principal del compilador MikroBasic**



Fuente: Mikroelektronika. *Mikrobasic user manual*. [http://www.fireflylabs.com/disted/courses/m286\(2012\)/mikroe/MikroBasic_manual.pdf](http://www.fireflylabs.com/disted/courses/m286(2012)/mikroe/MikroBasic_manual.pdf). Consulta: 28 de octubre de 2012.

4.2.1. **Estructura del programa**

MikroBasic consta de dos secciones: la sección de declaración y la sección del cuerpo del programa principal. Al iniciar un programa en MikroBasic se inicia con el nombre del programa, y luego se declara los diferentes módulos que en algunas aplicaciones se deberán de incluir, posteriormente se declaran los símbolos, las constantes, las variables, las funciones y los procedimientos. El cuerpo del programa formado por la función principal “main” como se describe a continuación, se deberá organizar de forma ordenada; de otro forma el compilador no será capaz de comprender el programa correctamente.

```

Program <Nombre del programa>
include <Incluir otros módulos>
*****

* Declaraciones globales:
*****

'Declaraciones simbólicas ← 'se utiliza apostrofo para comentarios.
symbol ...

'Declaración de constantes.
const ...

'Declaración de Variables.
dim ...

'Declaración de procedimientos.
sub procedure Nombre del procedimiento (...)
<Declaraciones locales>

...
end sub

'Declaración de funciones.
sub function Nombre de la función (...)
<Declaraciones locales>

...
end sub
*****

* Cuerpo del Programa.
*****

main:
'Aquí se escribe el código, la función principal es el módulo principal e indica al
'compilador donde debe de empezar el programa.
end.

```

Los módulos que se deseen agregar que no sean las del módulo principal y las implementación se declaran con las palabras reservadas *module* e *implements* y van antes de las declaraciones globales como se muestra.

```
module <Nombre del módulo>
include <Incluir otros módulos>
*****
!* Interfaces (Globales)
*****
```

4.2.2. Variables, constantes y etiquetas

Una variable puede ser global o local. Las variable globales tienen un alcance en todo el programa (se puede utilizar en cualquier parte del programa) se declaran fuera de una función, procedimiento o de la función principal *main* y es declarada debajo de la sentencia *include*. Una variable local tienen un alcance únicamente en un bloque del programa (solo se utiliza dentro de la función, procediendo o en alguna parte del programa) se declara dentro de una función o procedimiento e incluso en la función principal *main*, la sintaxis para la declaración de variables es la siguiente:

```
dim lista_de_identificadores as type
```

Por ejemplo:

```
dim i, j, k as byte
dim counter, temp as word
```

Las constantes no consumen ningún espacio de memoria, a diferencia de las variables que sí consumen memoria RAM. Las constantes son valores globales y se declara en la sección de declaración global de un programa. La sintaxis de una constante necesita un nombre, un valor y es opcional el tipo de dato a usar con la palabra reservada *const*.

```
const constant_name [as type] = value
```

Por ejemplo:

```
const MAX as longint = 10000
```

```
const MIN = 1000 ' El compilador asume el tipo de dato como word.
```

```
const SWITCH = "n" ' El compilador asume el tipo de dato como char.
```

```
const MSG = "Hello" ' El compilador asume el tipo de dato como cadena.
```

```
const MONTHS as byte[12] = (31,28,31,30,31,30,31,31,30,31,30,31)
```

Las etiquetas se utilizan para las sentencias *goto* y *gosub*, una etiqueta se le asigna un nombre y dos puntos. Para declarar una etiqueta en MikroBasic necesita estar dentro del mismo bloque de programa que *goto* y *gosub*. No es posible saltarse de la función principal hacia una función o un procedimiento ni tampoco se debe establecer una misma etiqueta para dos identificadores válidos.

Nombre_de_etiqueta: declaraciones

Ejemplo:

```
loop: Beep
```

```
goto loop
```


4.2.3. Funciones y procedimientos

Son rutinas o subprogramas que realizan una tarea específica de acuerdo al número de entradas y parámetros. Una función regresa un parámetro cuando finaliza y los parámetros dentro del paréntesis son parámetros formales similares a la declaración de variables. En MikroBasic los parámetros son siempre pasados a la función por valor y para pasar argumentos por dirección se añade la palabra reservada *byref* adelante del identificador. Un procedimiento no retorna ningún valor si no que realiza un procedimiento y para declarar una función o un procedimiento en MikroBasic se realiza de la siguiente manera:

```
sub function Nombre de la función (lista de parámetros) as tipo que regresa  
[Declaraciones locales]  
'Cuerpo de la función.  
end sub
```

```
sub procedure Nombre_del_procedimiento (lista de parámetros)  
[Declaraciones locales]  
'Cuerpo del procedimiento.  
end sub
```

Para llamar a una función o un procedimiento se utiliza su propio nombre, con los argumentos necesarios ingresados de la misma manera que los parámetros de la función o procedimiento. Se muestran algunos ejemplos de llamadas.

Ejemplo:

tmp = potencia (3, 12) 'Esta es una llamada a una función.

sec = ((sec and \$F0) >> 4)*10 + (sec and \$0F) 'La llamada es a una declaración auto-contenida.

4.2.4. Datos simples, arreglos, cadenas, punteros y estructuras

MikroBasic es un lenguaje tipificado y es de mucha importancia conocer los tipos de datos que el compilador acepta como válidos, las cuales son:

- Tipo simple.
- Arreglos.
- Cadenas.
- Punteros.
- Estructuras.

Los tipos de datos simples son elementos que, según su tipo, se pueden utilizar para una operación o una declaración, en la tabla IX, se presentan los diferentes tipos de datos en MikroBasic.

Los arreglos son una colección de elementos indexados (cada elemento tiene un índice) del mismo tipo (llamado tipo base). Los arreglos o llamados también vectores se declaran en MikroBasic como se muestra posterior a la tabla.

Tabla X. **Tipos de datos simples en MikroBasic**

Tipo	Tamaño	Rango
Byte	8-bit	0 – 255
char*	8-bit	0 – 255
Word	16-bit	0 – 65535
Short	8-bit	128– 127
Integer	16-bit	32768– 32767
Longint	32-bit	2147483648– 2147483647
Float	32-bit	$\pm 1,17549435082 * 10^{-38}$
		$\pm 6,80564774407 * 10^{38}$

Fuentes: Navaismo. *MikroBasic-español*. <http://www.unrobotica.com/programas/MikroBasic-Espa%C3%B1ol.pdf>. Consulta: 17 de noviembre de 2012.

dim Nombre as tipo[longitud del arreglo]

Ejemplo:

dim días_de_semana as byte[7]

dim muestras as word[50]

const meses as byte[12] = (31,28,31,30,31,30,31,31,30,31,30,31)

Los arreglos multidimensionados como los de dos dimensiones que también se llaman matrices son un arreglo de arreglos de una dimensión y son alojados en el espacio de memoria en filas. A continuación se presenta algunos ejemplos de arreglos multidimensionales.

Ejemplo:

dim m as byte[50][20] 'Arreglo de 2 dimensiones 50x20.

dim n as byte[4][2][7] 'Arreglo de 3 dimensiones 4x2x7.

sub procedure ejemplo (dim byref m as byte[50][20]) 'Podemos omitir la primera 'dimensión, arreglo pasados por dirección.

inc (m[1][1])

end sub

Una cadena representa una secuencia de caracteres y es equivalente a un arreglo construido con tipo de dato *Char*, si se desea añadir cadenas, basta con utilizar el signo “+” y las cadenas a añadir se forma en una; por tanto, la declaración de una cadena es de la siguiente manera:

dim nombre as string[longitud de cadena]

Un puntero es un tipo de dato que almacena una dirección de memoria, que, a diferencia de la variable que accede a esa dirección de memoria, directamente; para declarar un puntero se añade el prefijo (^) antes del tipo, como sigue a continuación:

dim nombre as ^tipo

Las estructuras representan una serie de elementos heterogéneos. Cada elemento se denomina un miembro y la declaración de la estructura define el tipo de cada miembro, donde el nombre de la estructura es un identificador válido. Cada tipo denota un *tipo* y cada miembro es un identificador válido para cada tipo. Las estructuras pueden estar anidadas y para su utilización las variables se declaran como un tipo estructural. Se muestra a continuación: cómo se declaran las estructuras y cómo se declaran las variables tipo estructura.

Structure Nombre de estructura

dim miembro 1 *as* tipo 1

dim miembro n *as* tipo n

end structure

dim Lista_de_identificadores as nombre de estructura

Se puede acceder a cada miembro de la estructura utilizando el “.” con la variable tipo estructura, así como se describe a continuación:

Nombre_variable.miembro1 = valor1

Nombre_variable.miembro2 = valor2

...

Nombre_variable.miembron = valorn

4.2.5. Operadores

Los operadores son aplicados a una variable o a algún objeto en una expresión para obtener algún resultado de alguna operación lógica, aritmética o relacional. MikroBasic soporta los operadores que a continuación se describen, empezando con la precedencia que tienen los operadores tomando en cuenta la asociación si es de derecha a izquierda o viceversa.

Tabla XI. **Precedencia de los operadores**

Precedencia	Operando	Operadores	Asociación
4	1	@ not + -	←
3	2	* / div mod and << >>	→
2	2	+ - or xor	→
1	2	= <> < > <= >=	→

Fuentes: Navaismo. *MikroBasic-español*. <http://www.unrobotica.com/programas/MikroBasic-Espa%C3%B1ol.pdf>. Consulta: 24 de noviembre de 2012.

En la tabla XII, se presentan los valores que MikroBasic acepta para cada operando y el valor del resultado de la operación.

Tabla XII. **Operadores aritméticos**

Operador	Operación	Operando	Resultado
+	suma	byte, short, integer, word, longint, float	byte, short, integer, word, longint, float
-	resta	byte, short, integer, word, longint, float	byte, short, integer, word, longint, float
*	multiplicación	byte, short, integer, word, float	integer, word, longint, float
/	división, punto flotante	byte, short, integer, word, float	byte, short, integer, word, float
div	división, redondeo al entero más cercano	byte, short, integer, word, longint, float	byte, short, integer, word, longint
mod	módulo, regresa el resto de la división de un entero (no se usa con flotantes)	byte, short, integer, word, longint	byte, short, integer, word, longint

Fuentes: Navaismo. *MikroBasic-español*. <http://www.unrobotica.com/programas/MikroBasic-Espa%C3%B1ol.pdf>. Consulta: 25 de noviembre de 2012.

Otros operadores relacionales son las igualdades y desigualdades, que regresan un valor verdadero o falso. En la tabla XI, se observa la precedencia de los operadores que va en dirección de la flecha dada en la columna de la asociatividad y se aplica de izquierda a derecha de arriba hacia abajo. El signo “=” también puede ser un operador de asignación; en la tabla XIII, se presentan los operadores relacionales que MikroBasic acepta como válidos.

Tabla XIII. **Operadores relacionales**

Operador	Operación
=	Igual
<>	Diferente
>	Mayor que
<	Menor que
>=	Mayor igual
<=	Menor igual

Fuentes: Navaismo. *MikroBasic-español*. <http://www.unrobotica.com/programas/MikroBasic-Espa%C3%B1ol.pdf>. Consulta: 27 de noviembre de 2012.

Los operadores lógicos comparan cada bit de los operandos numéricos y devuelven un valor de lógico 1 o 0 dependiendo del operador utilizado. Estos necesitan ser operandos con signo, sin signo o ambas. En la tabla XIV, se muestra la siguiente tabla con los operandos lógicos.

Tabla XIV. **Operadores lógicos**

Operador	Operación
and	AND inteligente; compara pares de bits y genera 1 si ambos son uno de otro modo genera un 0.
or	OR (inclusiva) inteligente; compara pares de bits y genera un 1 si el resultado es 1, de otra forma genera 0.
xor	OR exclusiva (XOR); compara pares de bits y genera un 1 si el resultado de ambos es complementario, de otra forma regresa 0.
not	Complemento (unario); invierte cada bit
<<	Corrimiento a la izquierda; mueve los bits a la izquierda, descarta el bit izquierda más lejano y asigna 0 al bit de la derecha.
>>	Corrimiento a la derecha; mueve los bits hacia la derecha, descarta el bit de la derecha más lejano y si no tiene signo lo asigna con 0 al bit de la izquierda, de otro modo el signo se extiende.

Fuentes: Navaismo. *MikroBasic-español*. <http://www.unrobotica.com/programas/MikroBasic-Espa%C3%B1ol.pdf>. Consulta: 1 de diciembre de 2012.

4.2.6. **Declaraciones**

En las declaraciones se incluyen asignaciones llamadas de rutinas y saltos a otras declaraciones. Se usan para crear ciclos, saltos y declaraciones estructuradas. Si no se utilizan saltos y selección de declaraciones, estas serán ejecutadas de manera secuencial, como se encuentra escrito en el código fuente. A continuación de presentan los tipos de declaraciones:

- Declaración de asignación.
- Declaración condicional.
- Declaración de iteración.
- Declaración de saltos.
- Declaración de lenguaje ensamblador (asm).

4.2.6.1. Declaración de asignación

Las declaraciones por asignación en MikroBasic se utilizan para guardar el resultado en una variable al evaluar una expresión. Las reglas de conversión y las de precedencia se aplican de igual manera en la expresión. La variable en la expresión puede ser declarada como un arreglo o una variable, y la expresión puede ser cualquier expresión, no se debe confundir la asignación con el operador relacional igual (=), ya que MikroBasic interpretará el signo igual (=) dentro de su contexto como a continuación se declara.

Variable = expresión

4.2.6.2. Declaraciones condicionales *If* y *Select case*

Las declaraciones condicionales *if* o *select case*, seleccionan una acción o diferentes acciones dependiendo de los valores o expresiones evaluadas. Se utilizará la declaración *if* para implementar una operación condicional.

if Expresión *then*

Declaraciones

[else

Otras declaraciones]

end if

La expresión necesita ser del tipo *boolean* para que la expresión sea válida. En una declaración condicional la condición *else* es opcional, pero se utilizará cuando se utilicen las *if*'s anidadas. La declaración de selección *select case* se utiliza para pasar el control o saltar a una rutina de programa, basado

en una condición certera, ésta selecciona una expresión de una lista de posibles valores, la sintaxis es la siguiente:

```
select case Selector  
case valor_1  
Declaraciones_1  
...  
case valor_n  
Declaración_n  
[case else  
Declaraciones _default]  
end select
```

Sin ningún valor es encontrado saltará a la declaración *default*, las declaraciones de selección *select case* pueden ser anidadas y si se requiere se puede agrupar valores para un caso simplemente separe los valores por comas.

4.2.6.3. Declaraciones de iteración *for*, *while* y *do*

Las declaraciones de iteración permiten crear ciclos de un número de declaraciones o instrucciones, en MikroBasic se pueden utilizar las declaraciones *for*, *while* y *do* para crear iteraciones. Se puede controlar el flujo de instrucciones en una iteración utilizando las palabras reservadas *break* y *continue*. La declaración *break* termina las declaraciones cuando este se presenta; la declaración *continue* comienza el siguiente ciclo de iteración de la secuencia.

La iteración *for* requiere que se especifique un número de iteraciones con un valor inicial, un valor final y un valor de paso (incremento) que incremente el

contador con cada iteración. El valor de contador es puesto al valor inicial y se incrementa, según el valor de paso hasta llegar al valor final. Si no se especifica el valor de paso, el incremento es por defecto a 1, la sintaxis para la declaración *for* en MikroBasic es como sigue:

for Contador = valor inicial to valor final [step valor de paso]

Declaraciones

next contador

La iteración *while* se ejecuta repetidas veces, mientras la expresión (condición) sea verdadera. En esta iteración, la expresión es evaluada la primera vez si ésta es falsa no se ejecutarán las declaraciones. A continuación se muestran la sintaxis para la declaración *while*:

while Expresión valida

Declaraciones

wend

La iteración *do* ejecuta un ciclo, mientras la expresión (condición) sea verdadera. La expresión es evaluada después de cada iteración y las declaraciones son ejecutadas, mientras la expresión sea verdadera; la sintaxis para la declaración *do*:

do

Declaraciones

loop until expresión

4.2.6.4. Declaraciones de salto

Las declaraciones de salto se utilizan para transferir el control del programa sin la necesidad de una condición o expresión, las declaraciones de salto en MikroBasic son:

- Break.
- Continue.
- Exit.
- Goto.
- Gosub.

4.2.6.4.1. Declaración *break*

Para detener el bucle desde el interior del cuerpo del bucle, se puede utilizar la sentencia *break* desde el interior de los bucles para pasar el control a la primera sentencia, seguido del ciclo más profundo (*for*, *while* o *do*). La sintaxis es como sigue:

```
while true
```

```
  Lcd_Out (1, 1, "tarjeta no insertada")
```

```
    if Cf_Detect ( ) = 1 then
```

```
      break
```

```
    end if
```

```
  Delay_ms (1000)
```

```
Wend
```

```
Lcd_Out (1, 1, "tarjeta detectada ")
```

La sentencia *while true* hasta el final *wend* es un ciclo infinito y, al utilizar la sentencia *break*, pasa el control a la primera sentencia después del ciclo, es decir, la sentencia *Lcd_Out*.

4.2.6.4.2. Declaración *continue*

La declaración *continue* se puede utilizar dentro de los ciclos para saltarse el ciclo, la sentencia *continue* en ciclos *for*, mueve el contador de programa a la línea con la palabra clave *for*, esto no cambia el contador del ciclo.

‘*Continue salta aquí.*

```
for ( i = ... )  
    continue  
next i
```

La sentencia *continue* en ciclos *while* mueve el contador del programa, es decir, a la primer línea de la condición (parte superior del bucle).

‘*Continue salta aquí.*

```
while condición  
    continue  
wend
```

La sentencia *continue* en ciclos *do*, mueve el contador de programa a la línea de la condición (parte inferior del bucle).

```
do  
    continue
```

*'Continue salta aquí.
loop until Condición*

4.2.6.4.3. Declaración *exit*

La declaración *exit* se utiliza para romper y salir de una rutina (función o procedimiento). La declaración *exit* pasa el control a la primera declaración después de la llamada de rutina. Si rompe o sale de una función, el valor regresado será el valor de la variable local *result* al momento de la salir.

```
sub procedure Proc1 ( )  
dim error as byte  
    'Aquí se hace algo.  
    if error = TRUE then  
        exit  
    end if  
    'Algún código no será ejecutado si se encuentra un error.  
end sub
```

4.2.6.4.4. Declaración *goto*

La declaración *goto* utiliza una etiqueta para pasar el control de programa a esa etiqueta. La sentencia *goto* se utiliza para un salto sin condiciones a cualquier etiqueta local, y la sintaxis es como a continuación se describe:

```
goto nombre de etiqueta
```

Algunos aspectos que se deben considerar es que la palabra reservada (*goto*) puede estar antes de la etiqueta o después de ella. No es posible saltar

hacia o desde una rutina. Se puede utilizar para salir dentro de una estructura anidada. Nunca se salta a una estructura de un ciclo o expresiones estructuradas; a continuación, se muestra la sintaxis.

```
for i = 0 to n
    for j = 0 to m
        if condición
            goto Error
        end if
    next j
next i
```

Error: ' Manejo del código, salta hasta etiqueta error.

4.2.6.4.5. Declaración *gosub*

La sentencia *gosub* se utiliza para un salto sin condiciones a cualquier etiqueta, la sintaxis de un *gosub* es como a continuación se muestra:

```
gosub nombre de la etiqueta
...
Nombre de la etiqueta:
...
return
```

La declaración *gosub* es similar a la declaración *goto*, excepto que la palabra reservada *return* retorna al contador de programa, al punto de llamada, el programa regresará y ejecuta la siguiente línea, después del *gosub*. Un *gosub* puede ir antes o después de una etiqueta y no es posible dar un salto dentro de una función o un procedimiento.

4.2.6.5. Declaración *asm*

Para programar en código *assembler* o ensamblador, acceder directamente a registros de propósitos específicos o registros de propósitos generales del PIC, MikroBasic permite la incrustación del código ensamblador con la palabra reservada *asm*. En código ensamblador no se permite el uso de números, así como direcciones, en cambio, permite el uso de nombres simbólicos. Los comentarios en código ensamblador se podrán realizar utilizando el punto y coma (;) y si se requiere utilizar una variable de MikroBasic en código ensamblador tendrá que tener un valor inicial, ya que si no se realiza esta operación el compilador mostrará un error. A continuación se muestra la sintaxis para declarar un bloque en código ensamblador.

asm

Bloque de declaraciones en ensamblador.

end asm

4.3. Programación con MikroBasic

Se realizará ejemplos de manejo de motores, lectura de sensores, uso de puertos, aplicación de módulos como por el ejemplo el *ADC* o el módulo *PWM*, etc. Se mostrará cómo se leen datos desde el puerto B y visualizarlo en el puerto C. Se muestra, a continuación, la sintaxis o código fuente para la lectura de datos del puerto B, al puerto C con la descripción iniciando con un apóstrofo.

Program prueba_1

Dim valor as byte 'Sección de declaración.

Main: 'Función principal main.

trisb = \$ff 'Puerto b como entrada (\$ff = 11111111) \$→hexadecimal.

<code>trisc = \$00</code>	'Puerto c como salida (\$00 = 00000000).
<code>Inicio:</code>	'Etiqueta inicio, el pc (contador de programa) regresa 'después de haber leído el puerto b y desplegado el valor 'en el puerto c.
<code>Valor = 0</code>	'Valor inicial de la variable.
<code>while valor = 0</code>	'Se repite el bucle mientras la condición es 'verdadera (valor = 0).
<code> valor = portb</code>	'Espera a que se ingrese un valor distinto de 'cero desde el puerto b y se guarda en la 'variable valor.
<code> portc = valor</code>	'Despliega el valor de la variable en puerto c.
<code>wend</code>	
<code>goto inicio</code>	'Envía el pc a la etiqueta Inicio para empezar 'de nuevo.
<code>end.</code>	'Fin de main.

4.3.1. Programación de manejo de motores

Para la programación del giro de un motor paso a paso (*stepper*) se debe programar la secuencia de pulsos que necesita para girar ya sea en sentido de las manecillas de reloj o viceversa. De la tabla III, se observa la secuencia de pulsos para un motor paso a paso bipolar.

Program motores_paso_a_paso

dim cnt_m1 as *byte*

sub procedure giro ()

select case cnt_m1

```

case 0
    cnt_m1 = 5      'Valida cuenta regresiva si cnt_m1 < 1.
case 1
    portc = 10     'Despliega en el puerto c 1010.
case 2
    portc = 9      'Despliega en el puerto c 1001.
case 3
    portc = 5      'Despliega en el puerto c 0101.
case 4
    portc = 6      'Despliega en el puerto c 0110.
case 5
    cnt_m1 = 0     'Valida cuenta si cnt_m1 > 4.
end select
end sub

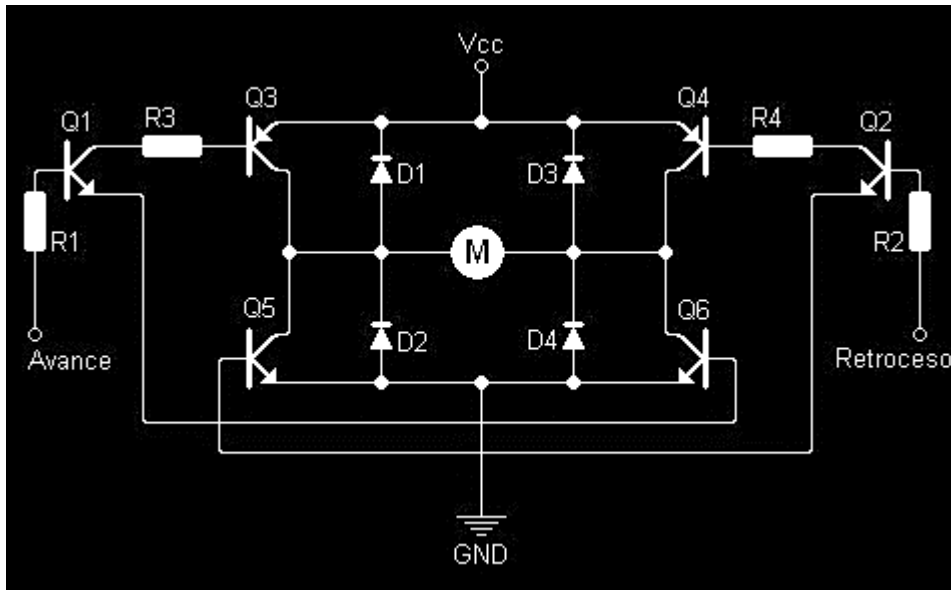
main:
trisc = $00       'Puerto c como salidas.
trisa = $FF       'Puerto a como entrada.
Adcon1 = $87     'Establece puerto a como digitales y no como analógicas.
cnt_m1 = 1
inicio:
    if porta = 1 then      'Valida puerto a para incremento.
        giro ()           'Llamada a sub procedimiento giro.
        inc(cnt_m1)       'Función incremento.
    else
        if porta = 2 then  'Valida puerto a para decremento.
            giro()         'Llamada a sub procedimiento giro.
            dec(cnt_m1)    'Función decremento.
        end if
    end if

```

```
    end if  
  end if  
goto inicio  
end
```

Para programar los motores paso a paso unipolares, se programa de forma similar a la mostrada anteriormente. De las tablas IV, V y VI se muestra la secuencia en modo normal, *wave drive* y de paso medio. La programación para el control de un motor DC utilizando la modulación por ancho de pulsos PWM, se utiliza para variar la velocidad en función del ancho del pulso (valor promedio "DC" de voltaje) y mantener constante el par motor manteniendo constante el nivel de voltaje, se utilizará los puertos del microcontrolador para controlar la dirección de giro, por medio de un puente en H (H-bridge), la dirección de la corriente proporcionará la dirección de giro del motor. Para hacer el control de giro se utiliza un puente en H (H-bridge) como el que se muestra en la figura 80, la señal PWM se ingresa como se indica en la figura.

Figura 80. Puente en H para control de giro de un motor DC



Fuente: *PICs, electrónica y robótica*. <http://picrobot.blogspot.com/2009/05/puente-h.html>.

Consulta: 6 de enero de 2013.

La programación del microcontrolador y la explicación de la misma se muestran a continuación.

Program Manejo_motor_DC

dim giro as byte

main: 'Función principal main.

trisb = \$00 'Puerto b = 00000000 como salida.

trise = \$07 'Puerto e como entradas.

trisc 2 = %1 'Puerto rc2 como salida % corresponde a valor binario.

ADCON1 = \$82 'Se configura el puerto e como entradas digitales.

PWM1_Init (10000) 'Inicializa módulo PWM a una frecuencia a 10khz.

'PWM1_Start() 'Inicializa el módulo PWM.

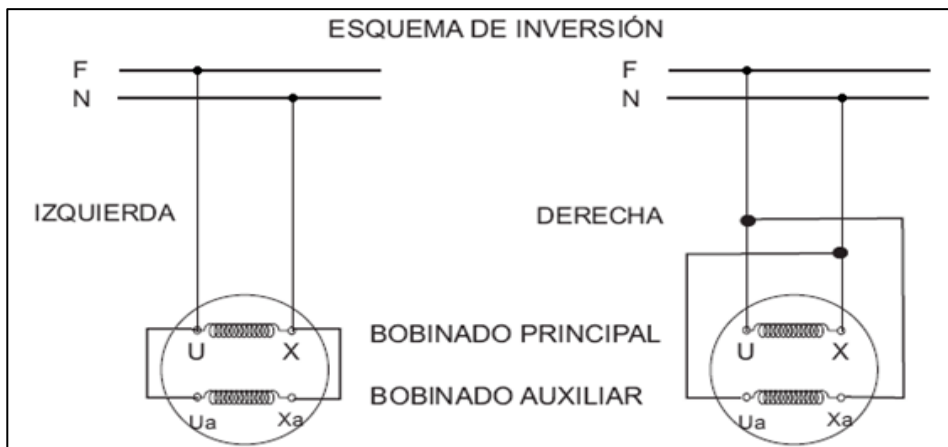
```

PWM1_Set_Duty (153) '153 establece el ciclo de trabajo a 60 % de la siguiente
                    'fórmula (Percent*255)/100.
inicio:              'Etiqueta inicio.
giro = 5
while giro > 4      'Ciclo para capturar valor del puerto e.
    giro = porte
wend
    if giro = 1 then
        PWM1_Start() 'Inicializa el módulo PWM.
        portb = 2     'Bit de selección de giro hacia la derecha = avance.
    else
        if giro = 2 then
            PWM1_Start() 'Inicializa el módulo PWM .
            portb = 8     'Bit de selección de giro hacia la izquierda = retroceso.
        else
            if giro = 0 then
                'No hay giro si no hay selección desde del puerto e.
                portb = 0 'No selecciona giro y el motor no se mueve.
                PWM1_Stop() 'Detiene las señal PWM si no hay giro.
            end if
        end if
    end if
    if giro = 4 then
        'valor puerto e = 100 binario.
        goto salida    'Salta etiqueta salida para finalizar el programa.
    end if
goto inicio
salida:              'Etiqueta salida y finaliza el programa.
end.

```

Para operar un motor AC monofásico se utiliza la alimentación de voltaje AC para hacer girar al motor, para lograr el giro en sentido contrario se necesita solamente invertir la conexión de la bobina de arranque, dejando intacto la conexión de la bobina de trabajo. Si el motor tiene capacitor de arranque ésta se encuentra entre la bobina U y Ua como se muestra en la figura 81.

Figura 81. **Diagrama esquemático del cambio de giro de un motor AC monofásico**

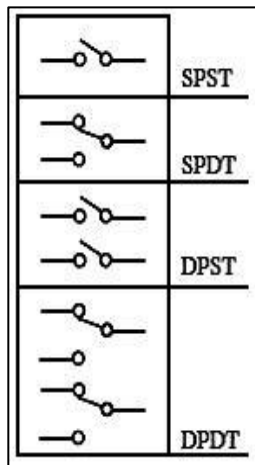


Fuente: *Mecatrónica*. <http://www.mecatronica.co/2013/03/inversion-de-giro-de-un-motor-monofasico.html>. Consulta: 9 de enero de 2013.

Para lograr el cambio de las conexiones en el bobinado secundario, se puede utilizar un relé o un contactor (para el manejo de mayor potencia), y con el uso de un algún puerto del microcontrolador para controlar la dirección. Los códigos para los relés son como siguen: NC normalmente cerrados, NA normalmente abiertos, SPST un Solo Polo un Solo Tiro tienen dos terminales que pueden conectarse o desconectarse, SPDT Un Solo Polo Doble Tiro un terminal común se conecta a cualquiera de los otros dos, DPST Doble Polo un Solo Tiro tienen dos pares de terminales equivale a un relé o dos interruptores

SPST y DPDT Doble Polo Doble Tiro equivale a un relé o dos interruptores DPST como se muestra en la figura 82.

Figura 82. **Simbología para los tipos de relés**

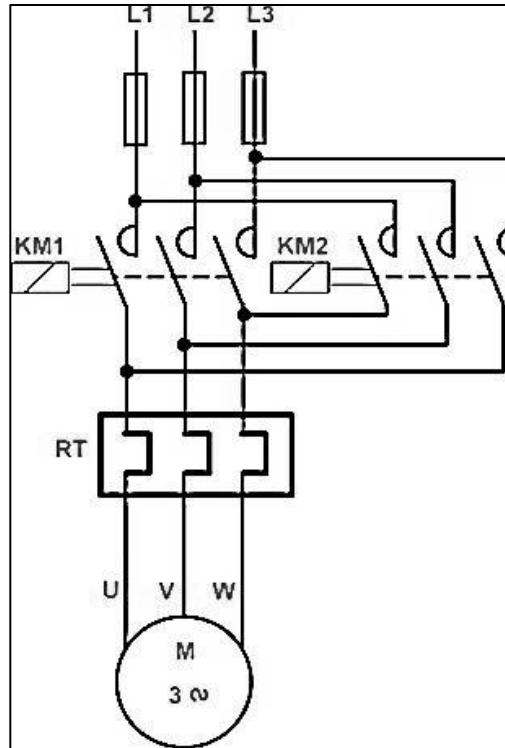


Fuente: Conbotassucias. *Relés*. <http://conbotassucias.wordpress.com/2013/01/14/reles/>.

Consulta: 15 de enero de 2013.

Para un motor trifásico y lograr la inversión de giro, se puede invertir el sentido de giro de un motor trifásico con invertir dos de cualquiera de las tres líneas de alimentación del motor. Como se muestra en la figura 83, el circuito está formado por fusibles en cada fase, dos contactores que efectúan el cambio de giro, y un relé. Las fases que se están intercambiando son las L1 y L3 o se puede intercambiar cualquier par de fases.

Figura 83. Diagrama esquemático del cambio de giro de un AC motor trifásico



Fuente: *Electricidad y automatismo*. <http://www.nichese.com/circuito2.html>. Consulta: 16 de enero de 2013.

Para activar los relés o contactores y suministrar energía a los motores AC, es como se describe a continuación:

Program giro_motor_AC

dim giro_ac as byte

main:

triseb = \$00 'Puerto b = 00000000 ceros como salidas.

trise = \$7 'Puerto e = 111 como entradas.

ADCON1 = \$82 'Configura puertos a y e como digitales y no como analógicas.


```

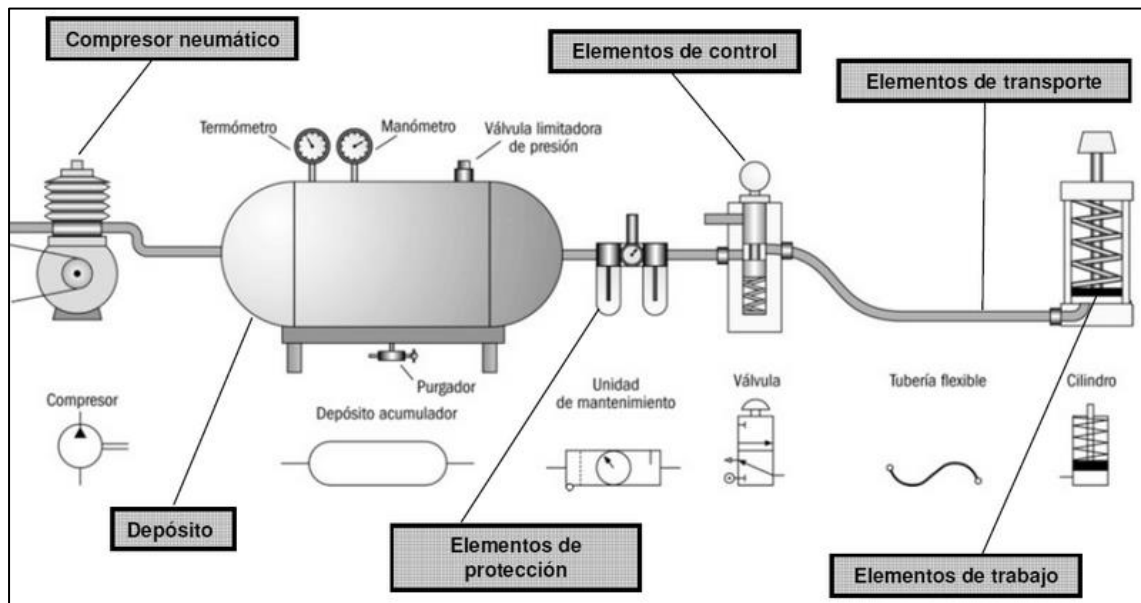
inicio:
giro_ac = 5          'Válida condición while.
  while giro_ac > 4
    giro_ac = porte  'Se carga el valor del puerto e para la sección del giro.
  wend
select case giro_ac  'Variable giro_ac para selección, del puerto e.
  case 0
    portb = 0        'Si puerto e = 0 no hay selección de giro.
  case 1
    portb = 16       'Puerto b = 00010000 giro hacia la derecha.
  case 2
    portb = 64       'Puerto b = 01000000 giro hacia la izquierda.
  case 4
    portb = 0        'Si puerto e = 4 el pc se manda a la etiqueta final.
    goto final      'Manda el pc a la etiqueta final.
  end select
goto inicio
final:              'Etiqueta final para finalizar el programa.
end.

```

Para operar los actuadores neumáticos o hidráulicos, se utilizarán válvulas para abrir y cerrar el paso del fluido o aire comprimido, para proporcionar energía a dichos actuadores. Las válvulas en neumática e hidráulica son los elementos que permiten direccionar el sentido del flujo que regular el comportamiento del circuito. Los componentes básicos para un circuito neumático consta de un compresor, acumulador, elemento de protección, válvulas (elemento de control), elemento de transporte (tuberías) y elementos de trabajos (cilindros). Los componentes en un circuito hidráulico constan de

tuberías, bombas, válvulas, depósitos, filtros, cilindros y/o motores. En la figura 84, se muestra un circuito neumático.

Figura 84. **Sistema neumático y sus componentes (análoga al sistema hidráulico)**



Fuente: *Tecnología (neumática)*, <http://www.areatecnologia.com/NEUMATICA.htm>. Consulta: 19 de enero de 2013.

La necesidad de comprender un sistema hidráulico o neumático como impulsor o fuentes de energía, ayuda a relacionar este sistema con un sistema eléctrico, el microcontrolador puede ser capaz de gobernar un conjunto de válvulas que, a su vez, controlarían la dirección del fluido para el avance o retroceso del vástago en los pistones. Se pueden programar los puertos del microcontrolador, para activar o desactivar las válvulas de acción eléctrica que necesitan un valor de voltaje desde 12, 24, 48 VDC o 120 a 240 VAC con la ayuda de algunos relés o contactores.

4.3.2. Programación de lectura de sensores

Para el uso de sensores, se utiliza el módulo ADC del microcontrolador. El ADC se podrá utilizar para convertir una señal analógica a una señal digital legible y manejable para el microcontrolador. Del inciso 2.3, se pueden observar los distintos dispositivos, su respectivo funcionamiento y su valor de voltaje de salida. Se puede utilizar directamente el ADC para el manejo del potenciómetro como sensores de posición y desplazamiento, como se puede observar en la figura 36, en el sensor de proximidad capacitivo e inductivo como se observan en las figuras 37 y 40.

También es aplicable este módulo con algunos circuitos adicionales si así lo requiere, por ejemplo, el empleo de un convertidor de corriente a voltaje los sensores ultrasónicos, los sensores de fuerza, los acelerómetros, los inclinómetros, los sensores de temperatura. Los sensores de contacto se pueden utilizar con algún puerto del microcontrolador para alguna aplicación.

MikroBasic maneja distintas librerías trigonométricas, operaciones matemáticas, operadores booleanos, operadores relaciones, etc. Las librerías son funciones especiales que pueden o no retornar un valor y son un conjunto de instrucciones que gobierna los periféricos del microcontrolador. También controlan el manejo de los registros de propósitos específicos y generales, así como a los periféricos internos del microcontrolador.

Se utilizará la librería del ADC para controlar el *hardware* del microcontrolador que, al igual al módulo PWM, están formados por una o varias funciones. A continuación se presenta la programación para la lectura de un valor de voltaje en cuatro puertos distintos del PIC (puerto A0, A1, A2 y A3) y

despliega los 8 bits más bajos en el puerto B y los 2 bits más altos en los puertos C0 y C1.

Program Sensor_ADC

dim muestra as word 'Variable muestra de 16 bits.
dim sel as byte 'Variable sel de 8 bits para selección de canal.

main:

'Adcon0 no se configura, porque se usa el ADC_init y ADC_read.

Adcon1=\$80 'De la pág. 140 y tabla VI justificado hacia la derecha
 'y todos los puertos se conf. Como entradas analógicas.
trisa = \$F 'Puerto a como entradas.
trisb = \$00 'Puerto b como salida.
trisc = \$F0 'Puerto c = 11110000, 1 como entrada y 0 como salida.

ADC_init() 'Inicializa módulo ADC interno del PIC para trabajar con
 'el reloj RC.

while portc.4 = 0 'Si porto c4 = 1 termina el ciclo while y termina el programa.

for sel = 0 to 3

 muestra = *ADC_Read(sel)* 'Lee el valor análogo del canal 0, 1, 2, 3 y
 'proporciona 10-bits de resultado de la
 'conversión.

 PORTB = muestra 'Envía los 8 bits bajos al PORTB.

 PORTC = word (muestra>> 8) 'Envía los 2 bits más significativos al RC1 y

next sel 'RC0.

wend

end.

Para la medición de un desplazamiento angular se puede realizar con un potenciómetro, un encóder absoluto de código decimal binario (BCD), un encóder absoluto de código binario o código grey. Entre los encóders absolutos, se pueden mencionar: el EA40 A/B encóder absoluto mono vuelta de 8 bits configurables para código binario o código grey, el EA50 A encóder absoluto mono vuelta hasta 10 bits configurable para código binario o grey y múltiples tipos de salida; el EA58/63 de hasta 13 bits configurable para código binario o grey y múltiples tipos de salida (paralelo, serial SSI y tipo incremental ICO), el TKC/TKTE con código grey o binario mono vuelta y el TKE45 con código binario, grey y BCD todas con funcionamiento de 5 voltios.

Se utilizará un encóder absoluto con salida de forma paralela y no con uno en forma serial o algún encóder incremental, porque el microcontrolador posee 4 puertos paralelos y solamente un puerto para funcionar en forma serial.

De la página 104 y al desarrollar la ecuación 3.16 para obtener la composición de transformaciones correspondiente a la orientación, tomando en cuenta que corresponde al sistema de referencia fijo y que la más utilizada en robótica para la orientación de la muñeca corresponde a las rotaciones básica X-Y-Z (guiñado, cabeceo y alabeo) se obtiene la expresión de la ecuación 4.1 que al desarrollarla y simplificarla se obtiene la ecuación 4.2.

$$\begin{aligned}
 & Rot(x - y - z_{fijo}) \\
 = & \begin{bmatrix} \cos \emptyset & -\sin \emptyset & 0 & 0 \\ \sin \emptyset & \cos \emptyset & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \Psi & -\sin \Psi & 0 \\ 0 & \sin \Psi & \cos \Psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

[Ec. 4.1]

$$\begin{aligned}
 & Rot(x - y - z_{fijo}) \\
 & = \begin{bmatrix} \cos \theta \cos \emptyset & \cos \emptyset \sin \theta \sin \Psi - \cos \Psi \sin \emptyset & \sin \emptyset \sin \Psi + \cos \emptyset \cos \Psi \sin \theta & 0 \\ \cos \theta \sin \emptyset & \cos \emptyset \cos \Psi + \sin \theta \sin \emptyset \sin \Psi & \cos \Psi \sin \theta \sin \emptyset - \cos \emptyset \sin \Psi & 0 \\ -\sin \theta & \cos \theta \sin \Psi & \cos \theta \cos \Psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 & \hspace{20em} [Ec. 4.2]
 \end{aligned}$$

De la matriz resultante solamente tres de los nueve componentes correspondientes son independientes, éstos son necesarios para resolver el problema de cinemática inversa. Al resolver el problema de cinemática inversa, se obtendrá los valores de las articulaciones conociendo los valores extremos respecto a la base.

Se presenta la programación que compara los valores de tres ángulos establecidos con los valores obtenidos de tres encóders, hasta lograr la condición deseada. Como ejemplo, se toman tres valores arbitrarios que se asumen son los ángulos iniciales que las articulaciones deberán tener para la posición inicial del robot. Si la posición inicial no son las deseadas, el microcontrolador evalúa por medio de los encóders los valores obtenidos si son mayores o menores al valor deseado para cada articulación, si es mayor el microcontrolador activará con un pulso la dirección de giro del motor, a través de un pin del puerto A, hasta el decremento del ángulo, para alcanzar el valor deseado.

Si el valor es menor, entonces el microcontrolador activará con un pulso, la dirección de giro contrario, a través de otro pin del puerto A, hasta incrementar el ángulo y alcanzar el valor deseado. Cuando ya se haya alcanzado el valor deseado el microcontrolador deja de emitir el pulso en sus pines para activar el motor, así no haya movimiento de incremento o

decremento de ángulo. El siguiente código posiciona los motores del brazo a un valor inicial.

```
Program Lectura_de_encóder_absoluto
```

```
dim sen1,sen2,sen3 as byte
```

```
sub procedure lectura_sensores() 'Procedimiento lee valores de los sensores.
```

```
    sen1 = portb      'Valor actual del puerto b.
```

```
    sen2 = portc      'Valor actual del puerto c.
```

```
    sen3 = portd      'Valor actual del puerto d.
```

```
end sub
```

```
sub procedure posicion_inicial_robot( ) 'Procedimiento posición inicial.
```

```
'El bucle while termina cuando las variables sen1, sen2 y sen3 son iguales a  
'10 y 126 que para este ejemplo son valores arbitrarios.
```

```
while (sen1 <> 10) or (sen2 <> 126) or (sen3 <>126)
```

```
    lectura_sensores ( ) 'Llamada procedimiento de lectura de sensores.
```

```
    if sen1 > 10 then      'Si sen1 es mayor que 10 activa el motor para que
```

```
        porta.1 = 1      'gire hacia un valor de decremento dado por el pin RA1.
```

```
        porta.0 = 0
```

```
    else
```

```
        if sen1 < 10 then 'Si sen1 en menor que 10 activa el motor para que
```

```
            porta.0 = 1      'gire hacia un valor de incremento dado por el pin RA0.
```

```
            porta.1 = 0
```

```
        else
```

```
            porta.0 = 0      'Si sen1 = 10 el PIC no emite pulsos de inc. O dec.
```

```

    porta.1 = 0
  end if
end if
if sen2 > 126 then 'De igual manera al caso anterior con el pin RA2 y RA3.
  porta.3 = 1
  porta.2 = 0
else
  if sen2 < 126 then
    porta.2 = 1
    porta.3 = 0
  else
    porta.2 = 0      'Si sen2 =126 el PIC no emite pulsos de inc. O dec.
    porta.3 = 0
  end if
end if

if sen3 > 126 then 'De igual manera al caso anterior con el pin RA4 y RA5.
  porta.5 = 1
  porta.4 = 0
else
  if sen3 < 126 then
    porta.4 = 1
    porta.5 = 0
  else
    porta.4 = 0      'Si sen3 = 126 el PIC no emite pulsos de inc. O dec.
    porta.5 = 0
  end if
end if
wend

```



```

'Sen1_A + sen1_B = 111100000000 + 11111111 (12 bits).
dim sen1_B, sen1_A as byte      'Sen1_B byte bajo, sen1_A byte alto.

sub procedure lectura_sensor()
    sen1_B = portb
    sen1_A = portd
end sub

main:
    trisb = $FF                'Puertos B y D como entradas.
    trisd = $FF
    Usart_Init(9600)           'Inicializa módulo USART a 9600 bps.

    sen1_A = 1                 'Condiciona inicio ciclo while.

while sen1_A < 16             'si el valor es menor o igual a 12 bits (porta=10000).
    lectura_sensor()           'Lee encóder por medio del puerto B y D.
    Usart_Write(sen1_B)        'Envía dato bajo vía USART.
    Usart_Write(sen1_A)        'Envía dato alto vía USART.
wend
end.

```

El código de programa para el microcontrolador que recibe los datos utilizando el módulo USART, donde los datos son transmitidos por el microcontrolador en longitud de un byte cada uno, por lo que el microcontrolador receptor los recibe y los guarda en dos bytes, así como en una variable tipo Word (16 bits) que luego son desplegados por el puerto B para los bits menos significativos y por el puerto D los bits más significativos.

Program Comunicacion_entre_pics2

dim valor_A, valor_B as byte

dim dato as byte

main:

trisb = \$00 'Puertos B y D como salidas.

trisd = \$00

usart_init(9600) 'Inicio módulo USART a 9600 bps.

valor_A = 0

dato = 0

while valor_A < 16 'Valida condición a 12 bit recibidos.

if *Usart_Data_Ready* = 1 *then* 'Si dato es recibido.

 valor_B = *Usart_Read* 'Lee dato recibido.

 valor_A = *Usart_Read* 'Lee dato recibido.

end if

dato = (valor_A << 8) + valor_B 'Dato = XXXX00000000 + xxxxxxxx.

portb = dato 'Envía los 8 bits bajos al puerto B.

portd = dato >> 8 'Envía los 4 bits altos al puerto D.

wend

end.

4.4. Aplicación del microcontrolador al robot

Como primer paso, es necesario escoger qué tipo de robot se quiere controlar, la tarea que ejecutará, los grados de libertad de movimiento, el tipo de impulsor que se utilizará, los actuadores, etc. Para este análisis se utiliza un

robot KUKA como ejemplo y se determinan los datos que se requieren para resolver el problema de cinemática directa e inversa, para resolver los problemas de cinemática directa se termina lo siguiente:

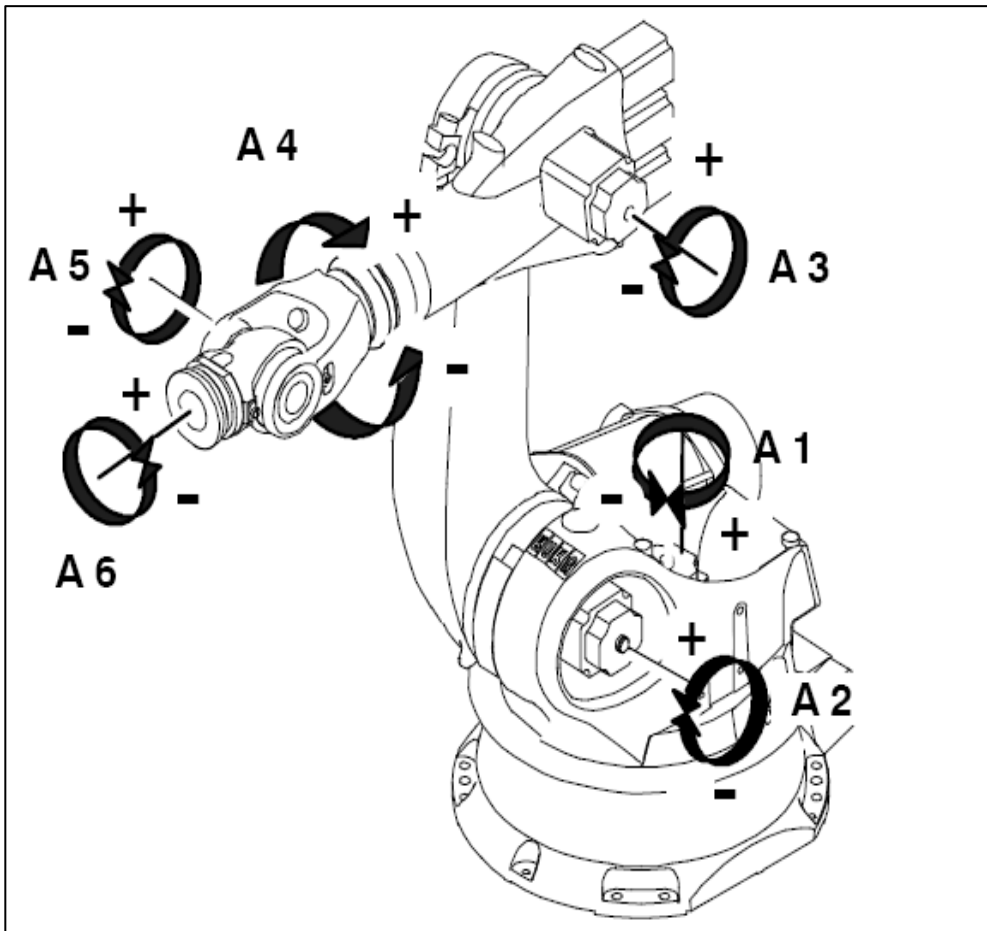
- Se numeran los eslabones, comenzó por 0 en la base y n para el último eslabón.
- Se numera las articulaciones, comenzando por 1 la que enlaza el eslabón de la base con el eslabón 1.
- Se localizan los ejes de las articulaciones.
- Localización del sistema de referencia i:
 - El eje z del sistemas i de se sitúa en el eje de la articulación i+1 con dirección dada por la regla de la mano derecha, según sean los valores positivos de movimiento rotacional o puede ser de forma inversa, según se convenga, se busca tener la dirección de z_n saliendo del último eslabón.
 - Se localiza el origen del sistema i en la intersección de la línea normal común (la normal común entre z_{i-1} y z_i) con el eje z_i y x_i se alinea con la normal común tomando la dirección que apunta de la articulación i e i+1.
 - Si los ejes z_{i-1} y z_i de cortan, el origen se coloca en la intersección de corte de dichos ejes y la dirección de x_i es perpendicular al plano que forman z_i y z_{i-1} .

- Si los ejes z_{i-1} y z_i son paralelos se toma el origen en la articulación $i+1$ y x_i toma la dirección que apunta de la articulación i e $i+1$.
- y_i de toma de forma que el sistema sea dextrógiro.
- Se establece el sistema de la base del robot, alineando el eje z_0 con el eje 1, con los ejes x_0 y y_0 de manera que resulte un sistema dextrógiro.
- Se establece el sistema n del último eslabón con el eje z_n , en la dirección de z_{n-1} y con los ejes x_n y y_n de manera que resulte un sistema dextrógiro.
- Se determina los parámetro Denavit-Hartenberg (θ_i , d_i , α_i y a_i).
- Obtener la transformación homogénea.
- Obtener la transformación homogénea total.

4.4.1. Cinemática directa

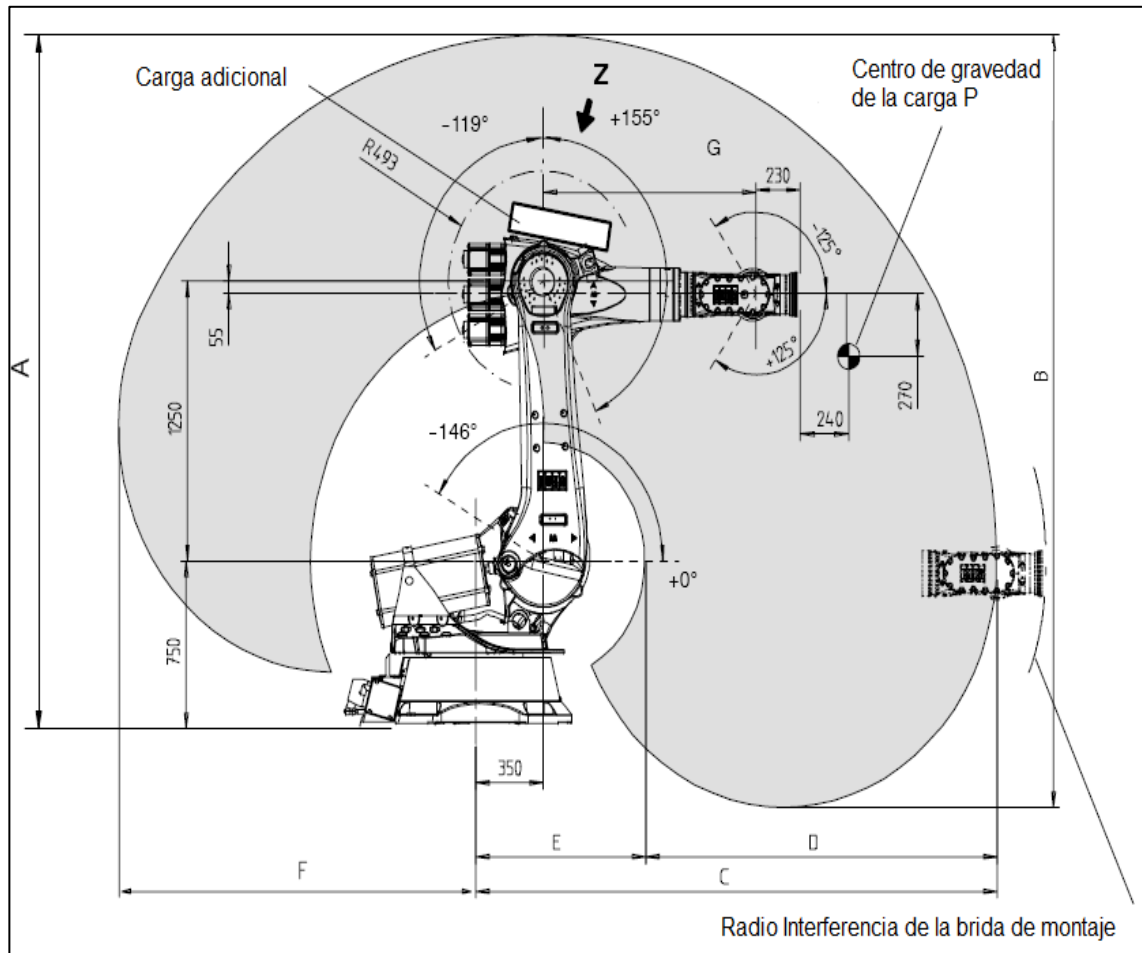
Se utiliza como modelo al robot KUKA que tiene 6 grados de libertad, que es el mínimo número de movimientos para alcanzar cualquier punto en el espacio que, como se muestra en la figura 85, proporciona información acerca de las dimensiones del robot, los ejes, el espacio de trabajo, el sentido de movimiento en cada eje y otros. No se realiza en análisis de cálculo por ser muy tedioso y difícil, se utiliza el programa matemático Scientific Notebook para resolver las operaciones matriciales.

Figura 85. a) Robot KUKA, determinación de articulaciones (numeración) y b) especificaciones física (dimensiones en mm y ángulos en grados)



a)

Continuacion de la figura 85.

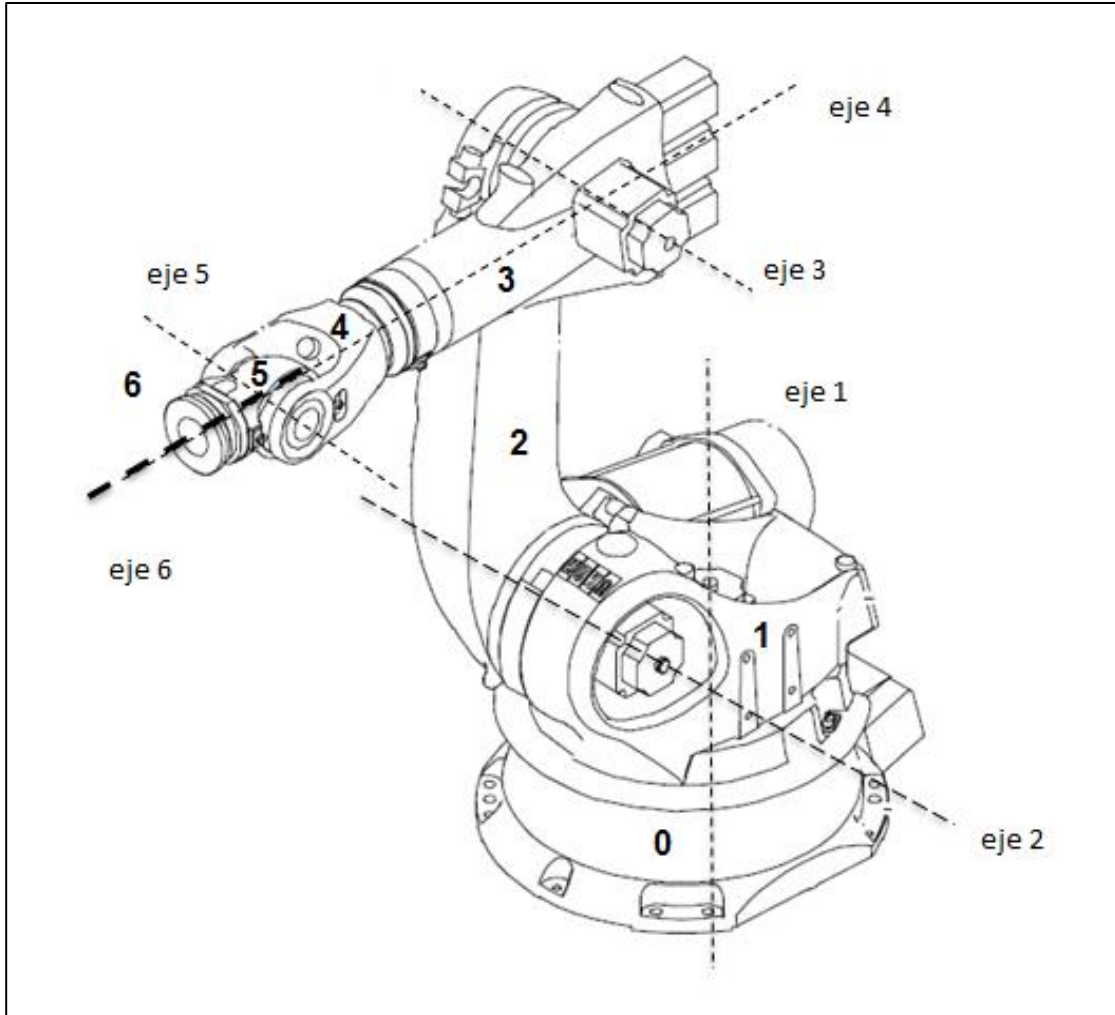


b)

Fuente: *KUKA Specifications*. http://www.KUKA-Robotics.com/es/products/industrial_robots/special/palletizer_robots/kr1000_i950_titan_pa/start.html. Consulta: 17 de marzo de 2013.

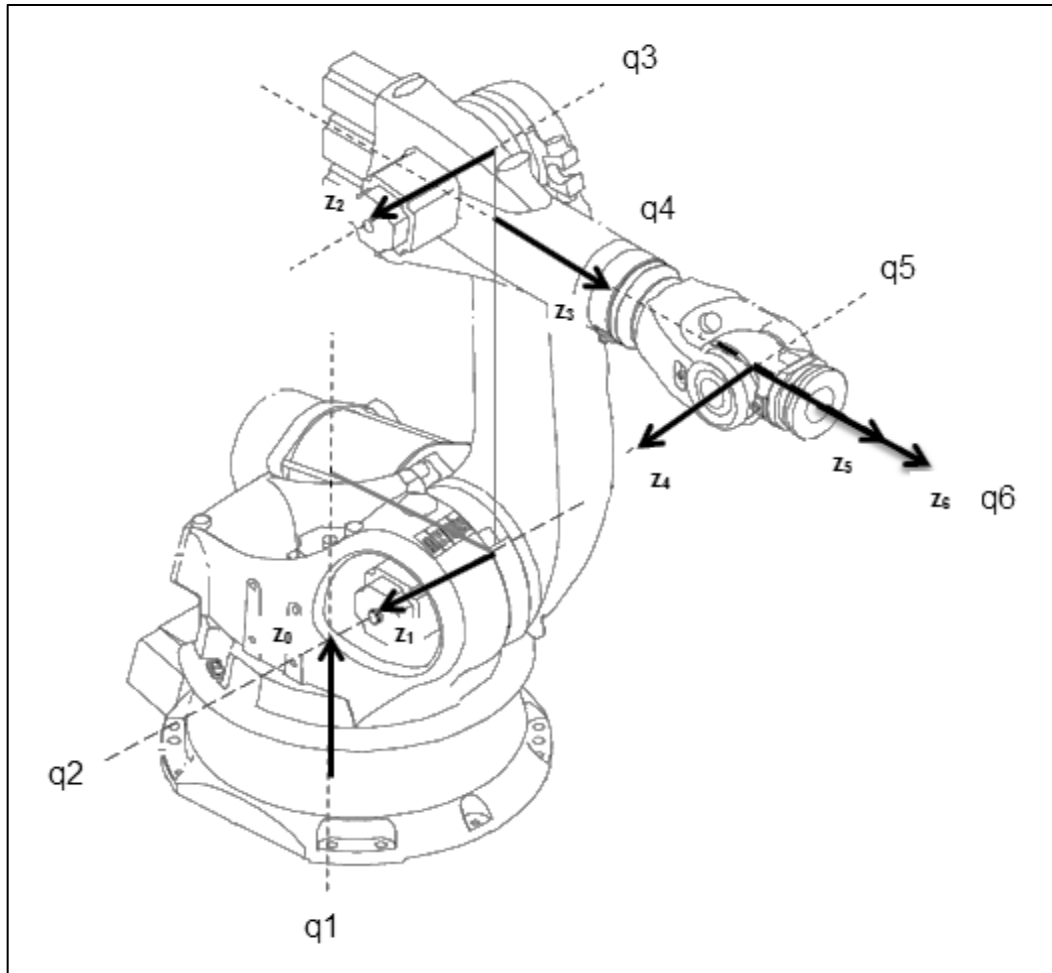
El análisis que se realiza a continuación corresponde en una secuencia de figuras y explicaciones breves, para obtener los parámetros Denavith-Hartenberg que son necesarios para resolver el problema de cinemática directa.

Figura 86. Ejecución de los incisos uno y tres, el inciso dos se representa en la figura 85



Fuente: elaboración propia.

Figura 87. **Asignación del eje z en la articulación i+1 y dirección del eje z en sentido contrario a la asignación positiva de rotación**

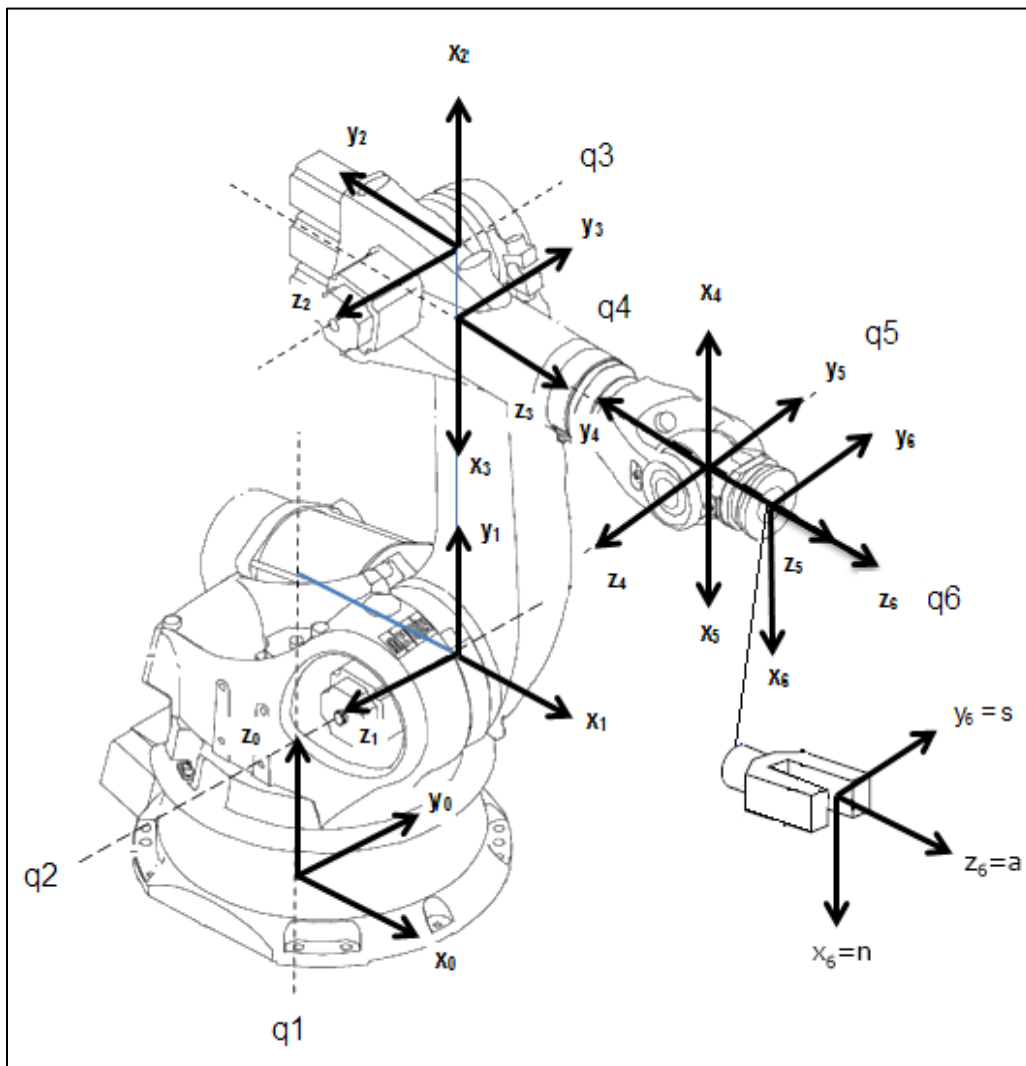


Fuente: elaboración propia.

En la figura 88, se muestra la determinación del origen del sistema i, el eje x se determina como se explica en el inciso cuatro. El eje y se asigna de forma que el sistema sea dextrógiro en donde ya se ha determinado el eje z y el eje x. También se muestra al sistema seis donde al robot se le ha colocado una herramienta, la diferencia es únicamente la distancia d_6 que es el tamaño de la

herramienta y el sistema se traslada al extremo únicamente la distancia d_6 , así todo el sistema queda sin cambios.

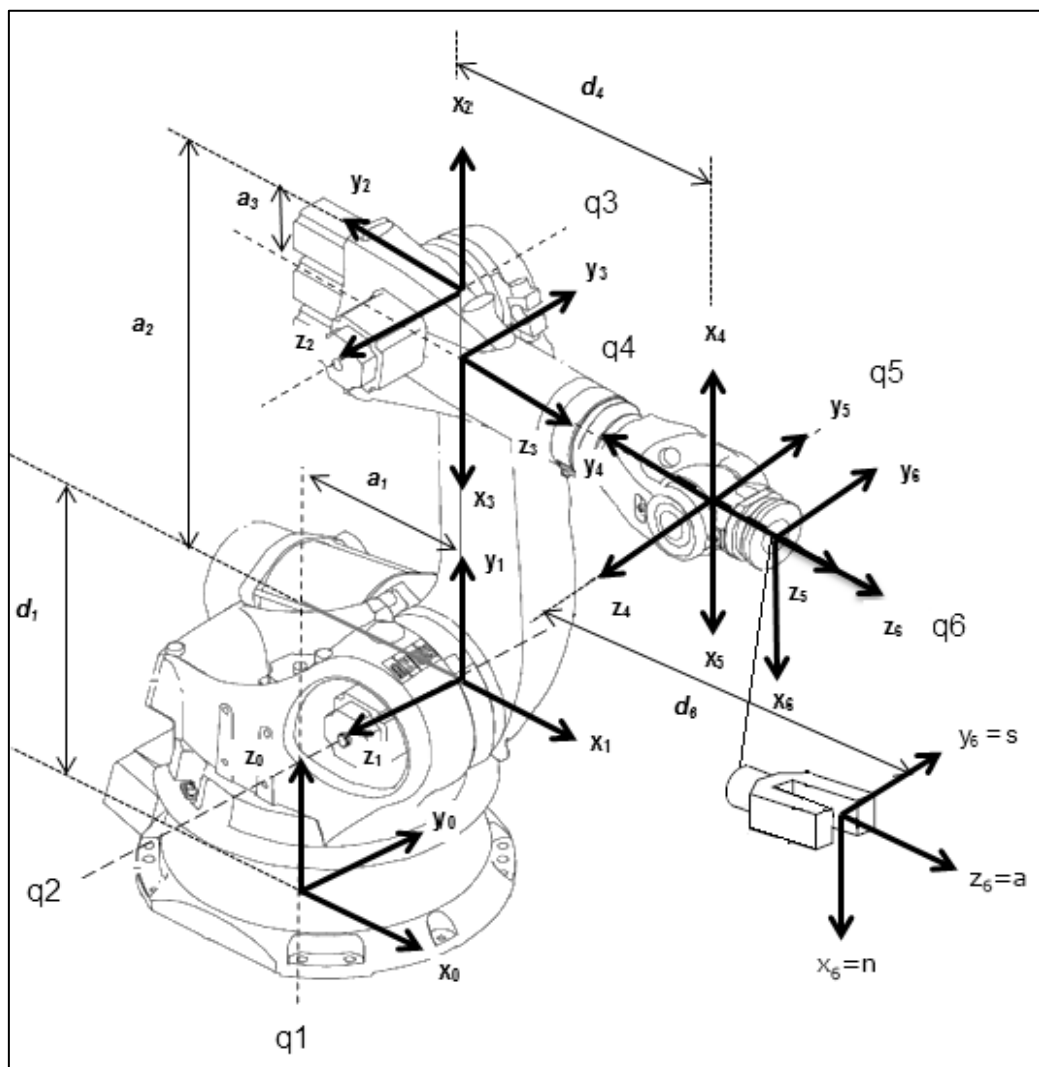
Figura 88. **Ejecución de los incisos cinco y seis, asignación del origen del sistema i**



Fuente: elaboración propia.

En la figura 89, se determinan los parámetros físicos (dimensiones), parámetros que definen el tamaño del eslabón, el tamaño y forma de la articulación.

Figura 89. **Determinación de los parámetros físicos, forma y tamaño de la articulación, así como tamaño del eslabón**



Fuente: elaboración propia.

Para determinar los parámetros θ_i , d_i , α_i y a_i de Denavith-Hartenberg, se calculan con los siguientes criterios en la tabla catorce.

- θ_i es el ángulo formado entre las normales comunes de i a $i+1$ y la normal común anterior, es el ángulo de x_{i-1} a x_i medida sobre q_i (utilizando la regla de la mano derecha). Rotación alrededor del eje z_{i-1} un ángulo θ_i .
- d_i es la distancia entre las normales comunes de i a $i+1$ y la normal común anterior. Es la distancia de x_{i-1} a x_i medida a lo largo de q_i . Traslación a lo largo de z_{i-1} una distancia d_i .
- a_i es la distancia ortogonal más corta entre los ejes z_{i-1} a z_i o, en otras palabras, indica el tamaño del eslabón. Es la distancia de z_{i-1} a z_i medida a lo largo de x_i . Traslación a lo largo x_i de una distancia a_i .
- α_i es el ángulo entre las articulaciones q_i y q_{i+1} si estos se cortasen en los puntos de corte de la normal común, es el ángulo de z_{i-1} a z_i medida sobre x_i (utilizando la regla de la mano derecha). Rotación alrededor del eje x_i un ángulo α_i .

Tabla XV. **Determinación de los parámetros de Denavith-Hartenberg del robot de la figura 89**

Articulación i	θ_i	d_i	a_i	α_i
1	q_1	d_1	a_1	90^0
2	q_2+90^0	0	a_2	0
3	q_3-180^0	0	a_3	-90^0
4	q_4-180^0	d_4	0	-90^0
5	q_5+180^0	0	0	-90^0
6	q_6	d_6	0	0

Fuente: elaboración propia.

Después de haber determinado los parámetros de Denavith-Hartenberg, se ejecuta el inciso ocho donde se determina la transformación homogénea de pasar del sistema i-1 al sistema i; de la página 119, ecuación 3.18 y utilizando un *software* matemático para evaluar y simplificar, se determina la transformación del sistema de la base al sistema uno.

$${}^0T_1 = \begin{bmatrix} \cos q_1 & -\cos 90^\circ \sin q_1 & \sin 90^\circ \sin q_1 & a_1 \cos q_1 \\ \sin q_1 & \cos 90^\circ \cos q_1 & -\sin 90^\circ \cos q_1 & a_1 \sin q_1 \\ 0 & \sin 90^\circ & \cos 90^\circ & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [\text{Ec. 4.3}]$$

Al evaluar y simplificar la matriz se obtiene.

$${}^0T_1 = \begin{bmatrix} \cos q_1 & 0 & \sin q_1 & a_1 \cos q_1 \\ \sin q_1 & 0 & -\cos q_1 & a_1 \sin q_1 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [\text{Ec. 4.4}]$$

Procediendo de la misma manera para determinar las transformaciones hasta el último eslabón se tiene lo siguiente:

$${}^1T_2 = \begin{bmatrix} -\sin q_2 & -\cos q_2 & 0 & -a_2 \sin q_2 \\ \cos q_2 & -\sin q_2 & 0 & a_2 \sin q_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [\text{Ec. 4.5}]$$

$${}^2T_3 = \begin{bmatrix} -\cos q_3 & 0 & \sin q_3 & -a_3 \cos q_3 \\ -\sin q_3 & 0 & -\cos q_3 & -a_3 \sin q_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [\text{Ec. 4.6}]$$

$${}^3T_4 = \begin{bmatrix} -\cos q_4 & 0 & \sin q_4 & 0 \\ -\sin q_4 & 0 & -\cos q_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [\text{Ec. 4.7}]$$

$${}^4T_5 = \begin{bmatrix} -\cos q_5 & 0 & \sin q_5 & 0 \\ -\sin q_5 & 0 & -\cos q_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [\text{Ec. 4.8}]$$

$${}^5T_6 = \begin{bmatrix} \cos q_6 & -\sin q_6 & 0 & 0 \\ \sin q_6 & \cos q_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [\text{Ec. 4.9}]$$

Después de haber determinado las transformaciones del sistema i-1 al sistema i, se ejecuta el paso número nueve en donde se determinará la transformación total. Aplicando la ecuación 3.17 de la página 113 resultará la ecuación 3.21 de la página 122. Para simplificar el análisis se utiliza la solución propuesta por Pieper y desacoplando la cadena cinemática, donde las tres primeras articulaciones proporcionan la ubicación del efector final y las tres últimas articulaciones proporcionan la orientación del efector final, donde se obtiene las siguientes matrices. Para simplificar la escritura se procede a reescribir la matriz resultante con $\sin q_1 = s_1$, $\cos q_1 = c_1$, entre otras.

$${}^0T_1{}^1T_2{}^2T_3 = \begin{bmatrix} c_1c_2s_3 + c_1c_3s_2 & -s_1 & c_1c_2c_3 - c_1s_2s_3 & a_1c_1 - a_2c_1s_2 + a_3c_1c_2s_3 + a_3c_1c_3s_2 \\ c_2s_1s_3 + c_3s_1s_2 & c_1 & c_2c_3s_1 - s_1s_2s_3 & a_1s_1 - a_2s_1s_2 + a_3c_2s_1s_3 + a_3c_3s_1s_2 \\ s_2s_3 - c_2c_3 & 0 & c_2s_3 + c_3s_2 & d_1 + a_2c_2 - a_3c_2c_3 + a_3s_2s_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [\text{Ec. 4.10}]$$

$${}^3T_4{}^4T_5{}^5T_6 = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_6s_4 - c_4c_5s_6 & -c_4s_5 & -d_6c_4s_5 \\ c_4s_6 + c_5c_6s_4 & c_4c_6 - c_5s_4s_6 & -s_4s_5 & -d_6s_4s_5 \\ c_6s_5 & -s_5s_6 & c_5 & d_4 + d_6c_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [\text{Ec. 4.11}]$$

La transformación total se obtiene al multiplicar todas las matrices.

$${}^{base}T_{extremo} = {}^0T_6 = {}^0T_1{}^1T_2{}^2T_3{}^3T_4{}^4T_5{}^5T_6 = {}^0T_3{}^3T_6 \quad [\text{Ec. 4.12}]$$

Reescribiendo la ecuación 3.21 de la página 122 se obtiene:

$${}^{base}T_{extremo} = {}^0T_6 = \begin{bmatrix} n_x & s_x & a_x & p_{x,extremo} \\ n_y & s_y & a_y & p_{y,extremo} \\ n_z & s_z & a_z & p_{z,extremo} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [\text{Ec. 4.13}]$$

Para cada componente de la matriz 4.13 de describe a continuación:

$$n_x = (c_6)((s_5)(c_1c_2c_3 - c_1s_2s_3) + (c_5)((c_4)(c_1c_2s_3 + c_1c_3s_2) - s_1s_4)) - (s_6)((s_4)(c_1c_2s_3 + c_1c_3s_2) + c_4s_1)$$

$$n_y = (c_6)((s_5)(c_2c_3s_1 - s_1s_2s_3) + (c_5)((c_4)(c_2s_1s_3 + c_3s_1s_2) + c_1s_4)) - (s_6)((s_4)(c_2s_1s_3 + c_3s_1s_2) - c_1c_4)$$

$$n_z = (c_6)((s_5)(c_2s_3 + c_3s_2) - (c_4c_5)(c_2c_3 - s_2s_3)) + (s_4s_6)(c_2c_3 - s_2s_3)$$

$$s_x = -(c_6)((s_4)(c_1c_2s_3 + c_1c_3s_2) + c_4s_1) - (s_6)((s_5)(c_1c_2c_3 - c_1s_2s_3) + (c_5)((c_4)(c_1c_2s_3 + c_1c_3s_2) - s_1s_4))$$

$$s_y = -(c_6)((s_4)(c_2s_1s_3 + c_3s_1s_2) - c_1c_4) - (s_6)((s_5)(c_2c_3s_1 - s_1s_2s_3) + (c_5)((c_4)(c_2s_1s_3 + c_3s_1s_2) + c_1s_4))$$

$$s_z = (c_6s_4)(c_2c_3 - s_2s_3) - (s_6)((s_5)(c_2s_3 + c_3s_2) - (c_4c_5)(c_2c_3 - s_2s_3))$$

$$a_x = (c_5)(c_1c_2c_3 - c_1s_2s_3) - (s_5)((c_4)(c_1c_2s_3 + c_1c_3s_2) - s_1s_4)$$

$$a_y = (c_5)(c_2c_3s_1 - s_1s_2s_3) - (s_5)((c_4)(c_2s_1s_3 + c_3s_1s_2) + c_1s_4)$$

$$a_z = (c_5)(c_2s_3 + c_3s_2) + (c_4s_5)(c_2c_3 - s_2s_3)$$

$$p_{x,\text{extremo}} = d_4(c_1c_2c_3 - c_1s_2s_3) + a_1c_1 + d_6((c_5)(c_1c_2c_3 - c_1s_2s_3) - (s_5)((c_4)(c_1c_2s_3 + c_1c_3s_2) - s_1s_4)) - a_2c_1s_2 + a_3c_1c_2s_3 + a_3c_1c_3s_2$$

$$p_{y,\text{extremo}} = d_4(c_2c_3s_1 - s_1s_2s_3) + a_1s_1 + d_6((c_5)(c_2c_3s_1 - s_1s_2s_3) - (s_5)((c_4)(c_2s_1s_3 + c_3s_1s_2) + c_1s_4)) - a_2s_1s_2 + a_3c_2s_1s_3 + a_3c_3s_1s_2$$

$$p_{z,\text{extremo}} = d_1 + a_2c_2 + d_6((c_5)(c_2s_3 + c_3s_2) + (c_4s_5)(c_2c_3 - s_2s_3)) + d_4(c_2s_3 + c_3s_2) - a_3c_2c_3 + a_3s_2s_3$$

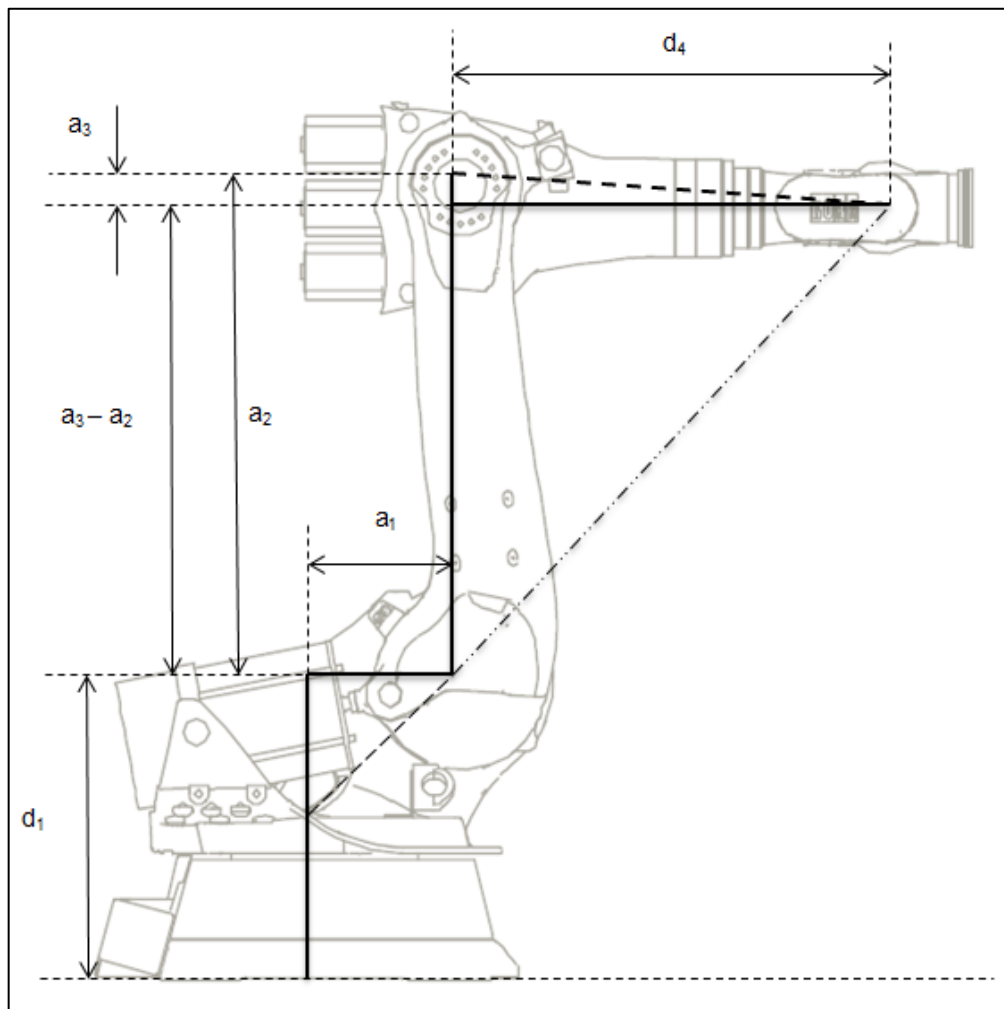
4.4.2. Cinemática inversa

El movimiento de las articulaciones de la muñeca no altera la posición espacial del punto de corte. Las tres primeras articulaciones son las que posicionan el punto de corte de los ejes en el espacio y, cuando se produzca alguna orientación correspondiente, esta no alterará la posición final. Pero la posición del punto de corte de los ejes no es la posición deseada del extremo del robot, ya que la muñeca (si se agrega alguna herramienta la dimensión total sería igual a la muñeca más la herramienta) tiene unas dimensiones físicas. Para determinar la posición del punto de corte de los ejes de la muñeca es necesario determinarla, a partir de la posición deseada para el extremo del robot, como se muestran en las ecuaciones 3.19 y 3.20, así como en las figuras 80b y 84.

Para resolver el problema de cinemática inversa se utiliza el método geométrico, para determinar los tres primeros ángulos que posicionan los ejes de corte de la muñeca, en la figura 90a se muestran unas líneas sobre el robot. Estas líneas representan los eslabones que componen el robot, a partir de los parámetros de Denavit-Hartenberg. En la figura 90b, se muestra al robot

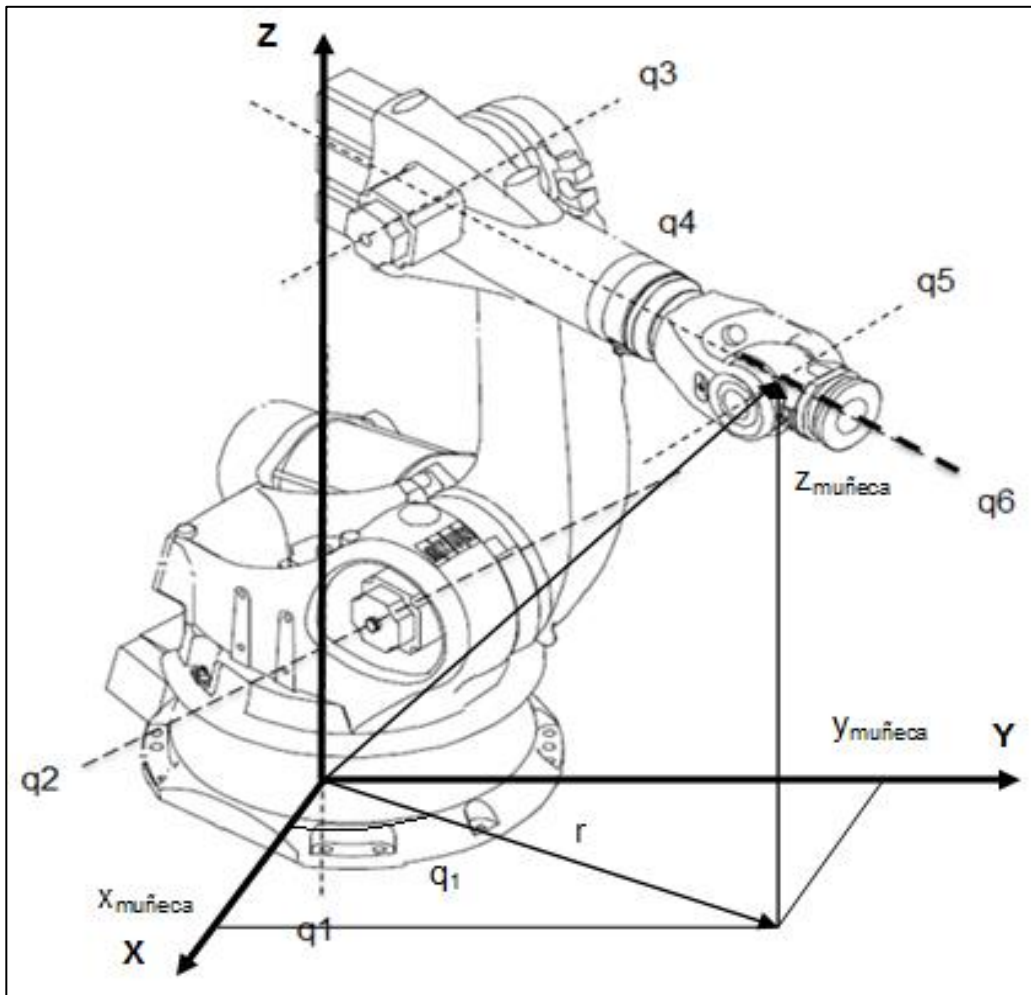
asignado al sistema de referencia M o 0 y el extremo del eslabón cinco o en otras palabras, el punto de corte de los ejes de la muñeca.

Figura 90. a) Especificaciones de eslabones al robot, de los parámetros D-H, b) Asignación del sistema de referencia y ubicación de 0T_5 .



a)

Continuación de la figura 90.



b)

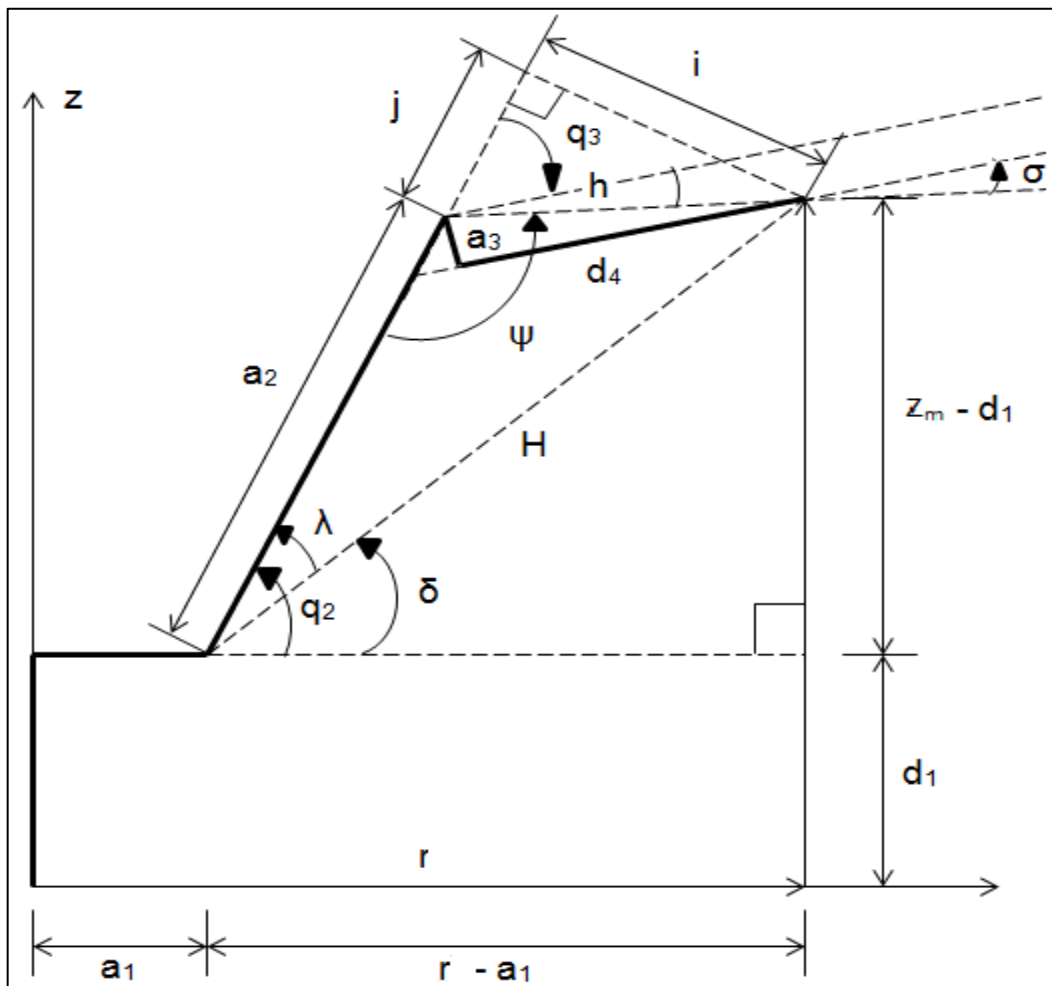
Fuente: elaboración propia.

Se obtiene el valor de q_1 a partir de la figura 82b con la siguiente relación.

$$q_1 = \tan^{-1} \left(\frac{y_m}{x_m} \right) \quad [\text{Ec. 4.14}]$$

Donde y_m se refiere a $y_{\text{muñeca}}$ y x_m se refiere a la $x_{\text{muñeca}}$. Para determinar q_2 y q_3 se puede determinar por relaciones geométricas, a partir de la siguiente figura, donde las líneas representan a los eslabones y las líneas discontinuas las relaciones geométricas.

Figura 91. **Relaciones geométricas del robot para determinar q_2 y q_3 .**



Fuente: elaboración propia.

De la figura 86, se observan las siguientes relaciones:

$$\delta = \tan^{-1} \left(\frac{z_m - d_1}{r - a_1} \right) \quad [\text{Ec. 4.15}]$$

$$r = \sqrt[2]{x_m^2 + y_m^2} \quad [\text{Ec. 4.16}]$$

$$H^2 = (r - a_1)^2 + (z_m - d_1)^2 \quad [\text{Ec. 4.17}]$$

$$\begin{aligned} h^2 &= a_3^2 + d_4^2 \\ \Rightarrow h &= \sqrt[2]{a_3^2 + d_4^2} \end{aligned} \quad [\text{Ec. 4.18}]$$

$$\sigma = \tan^{-1} \left(\frac{a_3}{d_4} \right) \quad [\text{Ec. 4.19}]$$

Por ley de cosenos, se determina el ángulo ψ .

$$\begin{aligned} H^2 &= a_2^2 + h^2 - 2a_2h \cos \psi \\ \Rightarrow 2a_2h \cos \psi &= a_2^2 + h^2 - H^2 \\ \Rightarrow \cos \psi &= \frac{a_2^2 + h^2 - H^2}{2a_2h} \end{aligned} \quad [\text{Ec. 4.20}]$$

$$\begin{aligned} (\sin \psi)^2 + (\cos \psi)^2 &= 1 \\ \Rightarrow \sin \psi &= \sqrt[2]{1 - (\cos \psi)^2} \end{aligned} \quad [\text{Ec. 4.21}]$$

$$\psi = \tan^{-1} \left(\frac{\sqrt[2]{1 - (\cos \psi)^2}}{\frac{a_2^2 + h^2 - H^2}{2a_2h}} \right) \quad [\text{Ec. 4.22}]$$

Para determinar q_3 se suman los ángulos de donde se obtiene.

$$\psi + \sigma = 180^\circ - q_3$$

$$\Rightarrow q_3 = 180^\circ - \psi - \sigma \quad [\text{Ec. 4.23}]$$

$$q_3 = 180^\circ - \tan^{-1} \left(\frac{\frac{2\sqrt{1-(\cos \psi)^2}}{a_2^2+h^2-H^2}}{2a_2h} \right) - \tan^{-1} \left(\frac{a_3}{d_4} \right) \quad [\text{Ec. 4.24}]$$

Se establece q_2 determinando los siguientes valores.

$$i = h \cdot \sin(\sigma - q_3) \quad [\text{Ec. 4.25}]$$

$$j = h \cdot \cos(\sigma - q_3) \quad [\text{Ec. 4.26}]$$

$$\lambda = \tan^{-1} \left(\frac{i}{a_2 + j} \right)$$

$$\Rightarrow \lambda = \tan^{-1} \left(\frac{h \cdot \sin(\sigma - q_3)}{a_2 + h \cdot \cos(\sigma - q_3)} \right) \quad [\text{Ec. 4.27}]$$

$$q_2 = \lambda + \delta$$

$$\Rightarrow q_2 = \tan^{-1} \left(\frac{h \cdot \sin(\sigma - q_3)}{a_2 + h \cdot \cos(\sigma - q_3)} \right) + \tan^{-1} \left(\frac{z_m - d_1}{r - a_1} \right) \quad [\text{Ec. 4.28}]$$

Después de haber establecido los valores de ángulos para posicionar los ejes de corte de la muñeca, se determinan los valores de ángulos para orientar al efector final. Aplicando la ecuación 3.24 de la página 111, se determina que, al dividir las igualdades de las posiciones r_{23} con r_{13} se encuentra el valor de q_4 . Reescribiendo la ecuación 3.24, se obtiene:

$${}^3Rot_6 = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_6s_4 - c_4c_5s_6 & -c_4s_5 \\ c_4s_6 + c_5c_6s_4 & c_4c_6 - c_5s_4s_6 & -s_4s_5 \\ c_6s_5 & -s_5s_6 & c_5 \end{bmatrix} = r_{ij} \quad [\text{Ec. 4.29}]$$

$$\frac{-\sin q_4 \cdot \sin q_5}{-\cos q_4 \cdot \sin q_5} = \frac{r_{23}}{r_{13}} \quad [\text{Ec. 4.30}]$$

Después de haber realizado la operación $({}^0\text{Rot}_3)^T \cdot {}^0\text{Rot}_{\text{extremo}}$ se tiene:

$$\begin{aligned}
 r_{23} &= a_y \cos q_1 - a_x \sin q_1 \\
 r_{13} &= a_x(\cos q_1 \cos q_2 \sin q_3 + \cos q_1 \cos q_3 \sin q_2) \\
 &+ a_y(\cos q_2 \sin q_1 \sin q_3 + \cos q_3 \sin q_1 \sin q_2) - a_z(\cos q_2 \cos q_3 - \sin q_2 \sin q_3) \\
 q_4 &= \tan^{-1}\left(\frac{r_{23}}{r_{13}}\right) \qquad \text{Ec. 4.31]
 \end{aligned}$$

Para determinar el valor del ángulo q_5 se realiza desde la igualdad en la posición r_{33} como se realizó para ángulos anteriores, se busca que la expresión contenga tangentes más ningún seno y coseno para evitar redundancia.

$$\begin{aligned}
 \cos q_5 &= r_{33} \qquad \text{[Ec. 4.32]} \\
 \cos q_5 &= r_{33} = a_x(\cos q_1 \cos q_2 \cos q_3 - \cos q_1 \sin q_2 \sin q_3) \\
 &+ a_y(\cos q_2 \cos q_3 \sin q_1 - \sin q_1 \sin q_2 \sin q_3) + a_z(\cos q_2 \sin q_3 + \cos q_3 \sin q_2)
 \end{aligned}$$

$$q_5 = \cos^{-1}(r_{33}) \qquad \text{[Ec. 4.33]}$$

Por último, para determinar q_6 y el resultado del problema de cinemática inversa para el robot KUKA, se realiza la división de las igualdades de las posiciones r_{32} y r_{31} .

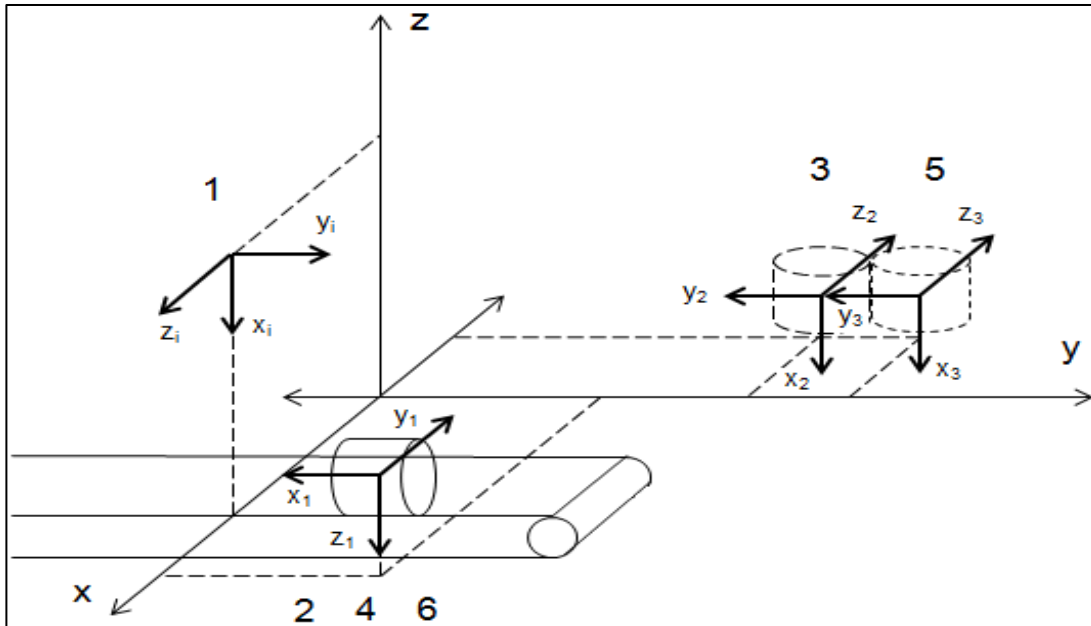
$$\begin{aligned}
 \frac{-\sin q_5 \sin q_6}{\cos q_6 \sin q_5} &= \frac{r_{32}}{r_{31}} \qquad \text{[Ec. 4.34]} \\
 r_{32} &= s_x(\cos q_1 \cos q_2 \cos q_3 - \cos q_1 \sin q_2 \sin q_3) \\
 &+ s_y(\cos q_2 \cos q_3 \sin q_1 - \sin q_1 \sin q_2 \sin q_3) + s_z(\cos q_2 \sin q_3 + \cos q_3 \sin q_2) \\
 r_{31} &= n_x(\cos q_1 \cos q_2 \cos q_3 - \cos q_1 \sin q_2 \sin q_3) \\
 &+ n_y(\cos q_2 \cos q_3 \sin q_1 - \sin q_1 \sin q_2 \sin q_3) + n_z(\cos q_2 \sin q_3 + \cos q_3 \sin q_2)
 \end{aligned}$$

$$q_6 = \tan^{-1} \left(\frac{r_{32}}{r_{31}} \right) \quad [\text{Ec. 4.35}]$$

Para determinar los valores de la ecuación 4.13, se utiliza la ecuación 4.2 de la página 190 para orientar al efector final o herramienta y, para determinar la posición del efector final o extremo del robot, se determina de la figura 87. En la figura se puede apreciar el embalaje de unas piezas de metal que recién hechas salen del horno, por el peso y la temperatura no puede ser realizado por el ser humano, también se aprecia la posición inicial del robot con su respectiva orientación.

A través de una banda transportadora, el robot toma la pieza y la coloca en posición vertical. El proceso se repite nuevamente colocando la siguiente pieza junto a la anterior. En la figura 87, se muestran los números de pasos para embalar las dos piezas. Estos pasos son repetitivos, pero cambian las posiciones al colocar las siguientes piezas. Esto significa que, a mayor número de piezas, mayor será el movimiento que el robot tendrá que realizar para embalar la cantidad de piezas requeridas, con la única condición que las piezas queden dentro del área de trabajo del robot. En la figura 92, se muestra la solución del problema descrito previamente.

Figura 92. Descripción de movimiento de embalaje realizado por el robot



Fuente: elaboración propia.

Los pasos de interés que se necesita determinar para la posición y la orientación del efector final son: el paso 1, paso 2, paso 3 y paso 5; los pasos 4 y 6 son repetitivos. Para posicionar y orientar el efector final, como se muestra en el paso 1, se aplica la ecuación 4.2 de página 190 para la orientación, en la que el efector final rota 90° alrededor del eje y con los datos como se muestran a continuación.

$$L_{herramienta} = 240 \text{ mm}$$

$$d_6 = 230\text{mm} + L_{herramienta} = 230\text{mm} + 240\text{mm} = 470\text{mm}$$

$$z_1 = 1.945\text{m}, x_1 = 1.920\text{m}, y_1 = 0$$

$$\text{Rot}(z, \Phi) = 0, \text{Rot}(y, \theta) = 90^\circ, \text{Rot}(x, \psi) = 0$$

Aplicando la ecuación 4.2 y 4.13.

$$f(\Psi, \theta, \phi) = \begin{bmatrix} \cos \theta \cos \phi & \cos \phi \sin \theta \sin \Psi - \cos \Psi \sin \phi & \sin \phi \sin \Psi + \cos \phi \cos \Psi \sin \theta & 0 \\ \cos \theta \sin \phi & \cos \phi \cos \Psi + \sin \theta \sin \phi \sin \Psi & \cos \Psi \sin \theta \sin \phi - \cos \phi \sin \Psi & 0 \\ -\sin \theta & \cos \theta \sin \Psi & \cos \theta \cos \Psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$f(0^\circ, 90^\circ, 0^\circ) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{base}T_{extremo} = {}^0T_6 = \begin{bmatrix} 0 & 0 & 1 & 1.920 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1.945 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Una vez determinados los valores extremos, se determinan los valores de q_1 , aplicando la ecuación 3.20 y 4.14.

$$x_m = 1.920\text{m} - (0.470\text{m})(1) = 1.450\text{m}$$

$$y_m = 0 - (0 \cdot 0) = 0$$

$$z_m = 1.945\text{m} - (0.470\text{m})(0) = 1.945\text{m}$$

$$q_1 = \tan^{-1}\left(\frac{0}{1.450}\right) = 0^\circ$$

Para determinar q_3 se utilizan las ecuaciones 4.16 al 4.19, 4.20, 4.22 y 4.24.

$$r = \sqrt{x_m^2 + y_m^2} = \sqrt{(1.450)^2 + (0)^2} = 1.45$$

$$H^2 = (r - a_1)^2 + (z_m - d_1)^2 = (1.45 - 0.35)^2 + (1.945 - 0.75)^2 = 2.6380$$

$$h^2 = a_3^2 + d_4^2 = (0.055)^2 + (1.1)^2 = 1.213$$

$$h = \sqrt{(0.055)^2 + (1.1)^2} = 1.101$$

$$\cos \Psi = \frac{a_2^2 + h^2 - H^2}{2a_2h} = \frac{(1.25)^2 + 1.213 - 2.638}{2(1.25)(1.101)} = 4.99 \times 10^{-2}$$

$$q_3 = 180^\circ - \tan^{-1} \left(\frac{\sqrt{1 - (4.99 \times 10^{-2})^2}}{4.99 \times 10^{-2}} \right) - \tan^{-1} \left(\frac{0.055}{1.1} \right)$$

$$q_3 = 180^\circ - 20.01^\circ - 2.86^\circ = 157.12^\circ$$

El valor de q_2 se determina con las ecuaciones 4.25 a la 4.28 con la cual se obtiene.

$$\sigma = \tan^{-1} \left(\frac{0.055}{1.1} \right) = 2.86^\circ$$

$$q_2 = \tan^{-1} \left(\frac{(1.101) \sin(2.86^\circ - 157.12^\circ)}{1.25 + (1.101) \cos(2.86^\circ - 157.12^\circ)} \right) + \tan^{-1} \left(\frac{1.945 - 0.75}{1.45 - 0.35} \right)$$

$$q_2 = \tan^{-1} \left(\frac{-0.478}{0.258} \right) + \tan^{-1} \left(\frac{1.195}{1.1} \right) = -61.64^\circ + 47.37^\circ = -14.27^\circ$$

A continuación se determinan los valores de q_4 , q_5 y q_6 con las ecuaciones 4.31, 4.33 y 4.35.

$$q_4 = \tan^{-1} \left(\frac{r_{23}}{r_{13}} \right) = \tan^{-1} \left(\frac{0}{0} \right) = 0$$

$$q_5 = \tan^{-1}\left(\frac{\sqrt[2]{1 - r_{33}^2}}{r_{33}}\right) = \tan^{-1}\left(\frac{\sqrt[2]{1 - 1^2}}{0}\right) = 0$$

$$q_6 = \tan^{-1}\left(-\frac{r_{32}}{r_{31}}\right) = \tan^{-1}\left(-\frac{0}{0}\right) = 0$$

Al observar el resultado de aplicar la ecuación 4.13, la figura 92 y los valores de q_4 , q_5 y q_6 ; se observa claramente que, para posicionar el efector final en la dirección correspondiente al paso 1, el valor de q_5 es igual a cero. Esta articulación es la encargada de proporcionar la dirección de z_6 . El valor de q_4 es un valor indefinido y para este caso q_4 puede tomar cualquier valor, ya que siempre tendrá la dirección como se muestra para el paso 1. Por conveniencia se asigna el valor de q_4 igual a cero. Esta articulación junto a q_5 es la encargada de proporcionar la dirección de efector final. El valor de q_6 también es un valor indefinido y para esta dirección no afectará el valor que se le asigne. Esta articulación no proporciona ninguna orientación al efector final sino más bien, para el tipo de robot que se analiza, únicamente proporciona la forma en que las pinzas se orienten para agarrar un objeto y por conveniencia se asigna un valor de cero.

Para el paso 1, se obtuvo paso a paso los valores de las articulaciones para orientar el efector final, así como se muestra en la figura 92. En la tabla XVI, se muestran los valores de cada articulación para posicionar y orientar el efector final en los pasos 1, 2, 3 y 5. Antes, es necesario determinar los valores de la ecuación 4.13 para los pasos correspondientes.

Paso 2.

$$f(\Psi, \theta, \phi) = f(0^\circ, 180^\circ, 90^\circ)$$

$${}^{base}T_{extremo} = {}^0T_6 = \begin{bmatrix} 0 & -1 & 0 & 2.2 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0.9 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x_m = 2.2 - (0.47)(0) = 2.2\text{m}$$

$$y_m = 1 - (0.47)(0) = 1\text{m}$$

$$z_m = 0.9 - (0.47)(-1) = 1.37\text{m}$$

Para el paso 3.

$$f(\Psi, \theta, \phi) = f(0^\circ, 90^\circ, 180^\circ)$$

$${}^{base}T_{extremo} = {}^0T_6 = \begin{bmatrix} 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 1.3 \\ -1 & 0 & -1 & 0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x_m = -1 - (0.47)(-1) = -0.53\text{m}$$

$$y_m = 1.3 - (0.47)(0) = 1.3\text{m}$$

$$z_m = 0.3 - (0.47)(0) = 0.3\text{m}$$

Para el paso 5.

$$f(\Psi, \theta, \phi) = f(0^\circ, 90^\circ, 180^\circ)$$

$${}^{base}T_{extremo} = {}^0T_6 = \begin{bmatrix} 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 1.6 \\ -1 & 0 & -1 & 0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x_m = -1 - (0.47)(-1) = -0.53\text{m}$$

$$y_m = 1.6 - (0.47)(0) = 1.6\text{m}$$

$$z_m = 0.3 - (0.47)(0) = 0.3\text{m}$$

Tabla XVI. **Valores de articulaciones para los pasos 1, 2, 3 y 5**

Paso	q ₁	q ₂	q ₃	q ₄	q ₅	q ₆
1	0°	-14.27°	157.12°	0°	0°	0°
2, 4 y 6	24,44°	74,63°	-129,44°	0°	-40,41°	-65,57°
3	11,19°	31,35°	-119,29°	-67,82°	-89,25°	1,92°
5	108,33°	29,83°	-103,82°	-72,34°	-85,02°	15,19°

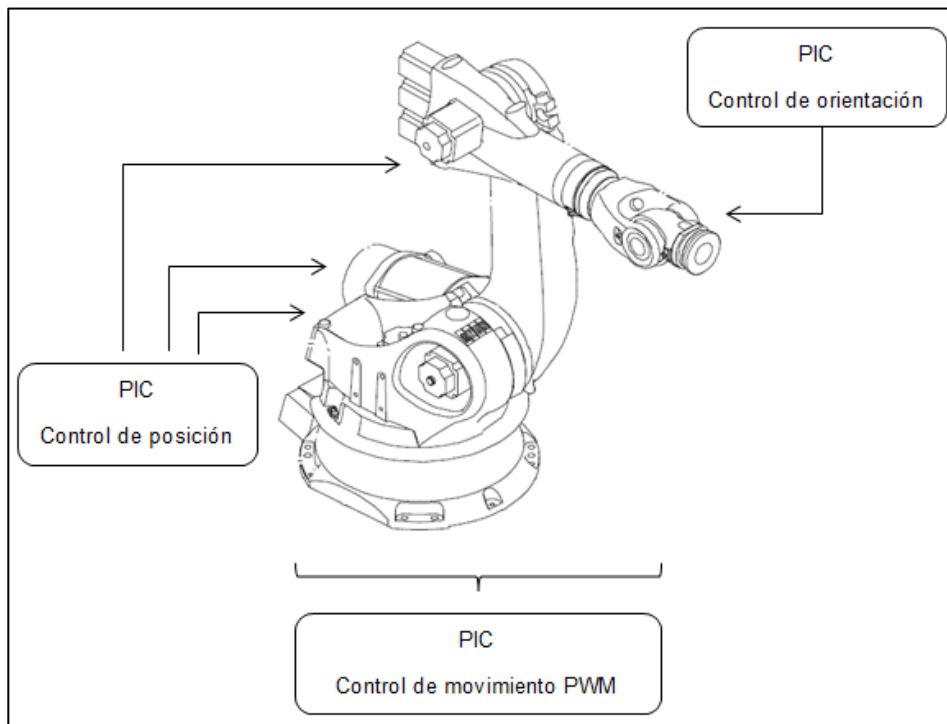
Fuente: elaboración propia.

4.4.3. **Aplicación del microcontrolador**

Como puede observarse en la tabla XVI, se tienen los valores que debe tener cada articulación para posicionar y orientar al efector final en la forma deseada. Para resolver el problema de posicionar y orientar el efector final utilizando el microcontrolador, se utiliza sub procedimiento y sub funciones. También se usan bucles y algunas funciones especiales que permiten controlar los motores, los sensores y algunos protocolos de comunicaciones para comunicar dos o más microcontroladores. La propuesta es utilizar dos microcontroladores para controlar al robot. El primero, se encargará de posicionar al robot; el segundo, orientará al efector final, y un tercero proporcionará la señal PWM para controlar el par motor si se utiliza motores DC, para accionar la articulación del robot, así como la velocidad en los motores.

Se pueden utilizar los siguientes microcontroladores: PIC16F877A, PIC18F458, PIC18F45K22. En la figura 88, se muestra en diagrama de bloques la solución del problema de control para un robot industrial.

Figura 93. **Aplicación del microcontrolador al gobierno del robot**



Fuente: elaboración propia.

De la figura 90b y 91 se pueden observar los valores de inicio de ángulo (posición 0°) para los ángulos q_1 , q_2 y q_3 . En la figura 87, se observa la dirección de giro representado por el vector z_i . También se observa en la figura 85b, que los valores positivos van en sentido de los de la aguja de reloj, para obtener los valores de ángulos de la cinemática inversa, se analizaron como positivos los valores contrarios a los de la aguja de reloj con los valores de 0° correspondientes a los de la figura 86.

En los encóders, los valores obtenidos no pueden ser negativos, ya que tiene valores desde 0 hasta 2^n . Para resolver este problema de ángulo negativo se tomaron los valores finales, por ejemplo, para el valor de q_3 que va desde -155° hasta 119° (valores positivos contrario a las agujas de reloj) desplazando el sistema de referencia con 0° a -155° y 274° correspondiente a 119° , para obtener un desplazamiento angular de 274° , los valores correspondientes al encóder sería de 0 a 196.

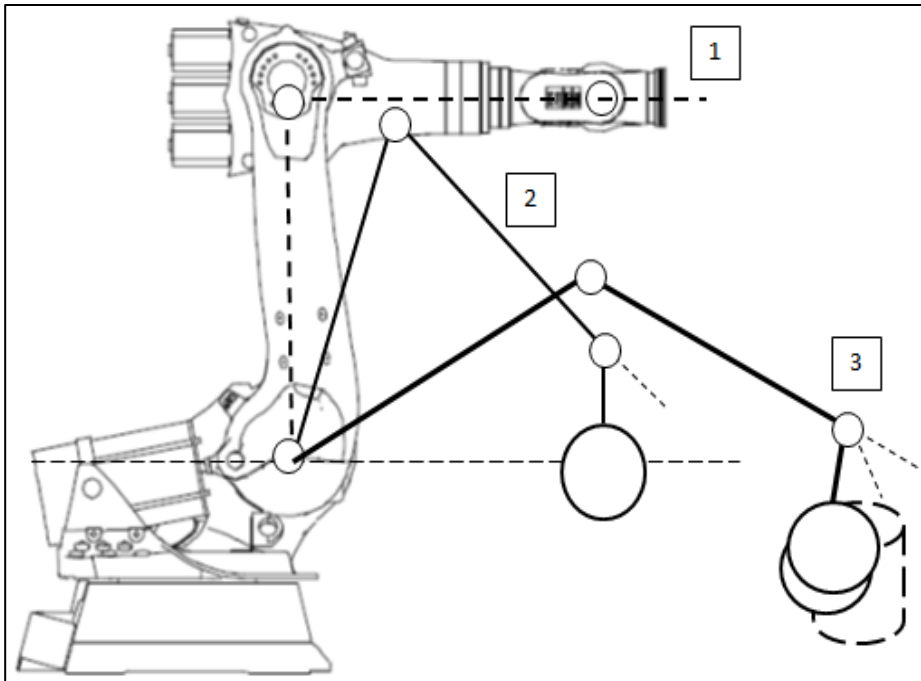
Se observa nuevamente la figura 80b y las características de movimiento del robot KUKA, la articulación q_4 tiene un valor de desplazamiento angular de $\pm 350^\circ$. La articulación q_5 de $\pm 125^\circ$ y q_6 tiene un valor de $\pm 350^\circ$; los valores de referencia 0° están con respecto a la horizontal y contrario al giro de las agujas de reloj. Reescribiendo la tabla XV y utilizando encóders para las articulaciones correspondientes a la orientación, se muestra en la tabla XVI los valores de ángulos en las que se desplazó el sistema de referencia y también se muestran los valores de encóders que leerá el microcontrolador. En la figura 94, se muestra que el desplazamiento angular para los puntos 1, 2 y 3.

Tabla XVII. **Valores de ángulos en las que se desplazó el sistema de referencia y el valor del encóder de 8 bits**

paso	q_1		q_2		q_3		q_4		q_5		q_6	
1	0°	0	90°	64	65°	46	180°	128	125°	89	180°	128
2, 4 y 6	24°	17	75°	54	26°	19	180°	128	85°	61	115°	82
3	112°	80	31°	22	36°	26	112°	80	36°	26	182°	130
5	108°	77	29°	21	52°	37	108°	77	40°	29	195°	139

Fuente: elaboración propia.

Figura 94. **Ejecución de los movimientos 1, 2 y 3 de embalaje piezas de metal caliente**



Fuente: elaboración propia.

A continuación se presenta el código de programa para posicionar la muñeca del robot en MikroBasic.

Program Pro_Final_Posicion

```
dim sen1,sen2,sen3,val1,val2,val3,paso as byte
```

```
sub procedure lectura_sensores() 'Procedimiento lee valores de los sensores.
```

```
sen1 = portb 'Valor actual del puerto b.
```

```
sen2 = portc 'Valor actual del puerto c.
```

```
sen3 = portd 'Valor actual del puerto d.
```


end sub

sub procedure posicion_robot() 'Procedimiento posición.

'El bucle while termina cuando las variables sen1, sen2 y sen3 son iguales a val1, val2 y val3; estos determinan la posición en cada paso.

while (sen1 <> val1) or (sen2 <> val2) or (sen3 <> val2)

if sen1 > val1 *then* 'Si sen1 es mayor que val1 activa el motor para que gire
 porta.1 = 1 'hacia un valor de decremento dado por el pin RA1.
 porta.0 = 0

else

if sen1 < val1 *then* 'Si sen1 es menor que val1 activa el motor para que
 porta.0 = 1 'gire hacia un valor de incremento dado por el pin RA0.
 porta.1 = 0

else

 porta.0 = 0 'Si cumple la igualdad no emite pulsos de inc. O dec.
 porta.1 = 0

end if

end if

if sen2 > val2 *then* 'Igual al caso anterior con el pin RA2 y RA3.

 porta.3 = 1
 porta.2 = 0

else

if sen2 < val2 *then*

 porta.2 = 1
 porta.3 = 0

else

 porta.2 = 0 'Si cumple la igualdad no emite pulsos de inc. O dec.

```

    porta.3 = 0
  end if
end if

if sen3 > val3 then      'Igual al caso anterior con el pin RA4 y RA5.
  porta.5 = 1
  porta.4 = 0
else
  if sen3 < val3 then
    porta.4 = 1
    porta.5 = 0
  else
    porta.4 = 0          'Si cumple la igualdad no emite pulsos de inc. O dec.
    porta.5 = 0
  end if
end if
wend
end sub
main:
trisa = $00      'Puerto a como salida.
trisb = $FF      'Puertos b, c y d como entradas, para los sensores(encóders).
trisc = $FF
trisd = $FF
trise.0 = 1      'Puerto e0 como salida.
trise.1 = 0      'Puerto e1 como entrada.
adcon1 = $87     'Configura el puerto a y e como E/S digitales.

val1 = 0         'Valores para posición 1.
val2 = 64

```

val3 = 46

lectura_sensores() 'Procedimiento lectura de sensores.

posicion_robot() 'Posiciona cada motor a un valor especifico.

inicio:

for paso = 2 to 6

if ((paso = 2) or (paso = 4) or (paso = 6)) *then*

val1 = 17 'Valores para posición 2, 4 y 6.

val2 = 54

val3 = 19

lectura_sensores() 'Procedimiento lectura de sensores.

posicion_robot() 'Posiciona cada motor a un valor especifico.

porte.0 = 1 'Valida que ya se posiciono en el punto 2, 4 y 6.

end if

if (paso = 3) *then*

val1 = 80 'Valores para posición 3.

val2 = 22

val3 = 26

lectura_sensores() 'Procedimiento lectura de sensores.

posicion_robot() 'Posiciona cada motor a un valor especifico.

porte.0 = 1 'Valida que ya se posiciono en el punto 3.

end if

if (paso = 5) *then*

val1 = 77 'Valores para posición 5.

val2 = 21

val3 = 37

```

    lectura_sensores()    'Procedimiento lectura de sensores.
    posicion_robot()     'Posiciona cada motor a un valor específico.
        porte.0 = 1      'Valida que ya se posiciono en el punto 5.
    end if

    while porte.1 = 0
        porte.0 = 0      'Mientras puerto e es 0 permanece en el ciclo y la
                        'muñeca del robot no se ha posicionado.

    wend
    delay_ms(1500)
next paso
    goto inicio          'Manda el pc a la bandera inicio y todo el ciclo empieza
                        'nuevamente hasta completarse.
end.

```

El código de programa para orientar al efector final difiere escasamente al código para posicionar la muñeca del robot y queda como sigue:

Program Pro_Final_Orientacion

```
dim sen1,sen2,sen3,val1,val2,val3,paso as byte
```

```

sub procedure lectura_sensores() 'Procedimiento lee valores de los sensores.
    sen1 = portb    'Valor actual del puerto b.
    sen2 = portc    'Valor actual del puerto c.
    sen3 = portd    'Valor actual del puerto d.
end sub

sub procedure orientacion_robot() 'Procedimiento orientación.

```

'El bucle while termina cuando las variables sen1, sen2 y sen3 son iguales a val1, val2 y val3; estos determinan la orientación en cada paso.

```
while (sen1 <> val1) or (sen2 <> val2) or (sen3 <> val2)
  if sen1 > val1 then   'Si sen1 es mayor que val1 activa el motor para que gire
    porta.1 = 1         'hacia un valor de decremento dado por el pin RA1.
    porta.0 = 0
  else
    if sen1 < val1 then 'Si sen1 es menor que val1 activa el motor para que
      porta.0 = 1       'gire hacia un valor de incremento dado por el pin RA0.
      porta.1 = 0
    else
      porta.0 = 0       'Si cumple la igualdad no emite pulsos de inc. O dec.
      porta.1 = 0
    end if
  end if

  if sen2 > val2 then   'Igual al caso anterior con el pin RA2 y RA3.
    porta.3 = 1
    porta.2 = 0
  else
    if sen2 < val2 then
      porta.2 = 1
      porta.3 = 0
    else
      porta.2 = 0       'Si cumple la igualdad no emite pulsos de inc. O dec.
      porta.3 = 0
    end if
  end if
end if
```

```

if sen3 > val3 then  'Igual al caso anterior con el pin RA4 y RA5.
    porta.5 = 1
    porta.4 = 0
else
    if sen3 < val3 then
        porta.4 = 1
        porta.5 = 0
    else
        porta.4 = 0      'Si cumple la igualdad no emite pulsos de inc. O dec.
        porta.5 = 0
    end if
end if
wend
end sub

```

main:

```

trisa = $00  'Puerto a como salida.
trisb = $FF  'Puertos b, c y d como entradas, para los sensores (encóders).
trisc = $FF
trisd = $FF
trise.0 = 1   'Puerto e0 como salida.
trise.1 = 0   'Puerto e1 como entrada.
adcon1 = $87  'Configura el puerto a y e como E/S digitales.

val1 = 128    'Valores para la orientación 1.
val2 = 89
val3 = 128

lectura_sensores()  'Procedimiento lectura de sensores.
orientacion_robot() 'Posiciona cada motor a un valor específico.

```

```

inicio:                                'Bandera inicio.

for paso = 2 to 6

porte.0 = 0                            'Regresa a cero la validación de orientación.
while porte.1 = 0
    'Mientras puerto e.1 es 0 permanece en el ciclo,
    'el brazo del robot no se ha posicionado.
wend

if ((paso = 2) or (paso = 4) or (paso = 6)) then
    val1 = 128                          'Valores para la posición 2, 4 y 6.
    val2 = 61
    val3 = 82
    lectura_sensores()                  'Procedimiento lectura de sensores.
    orientacion_robot()                 'Posiciona cada motor a un valor específico.
    porte.0 = 1                          'Valida que ya se orientó para el punto 2, 4 y 6.
end if

if (paso = 3) then
    val1 = 80                            'Valores para posición 3
    val2 = 26
    val3 = 130
    lectura_sensores()                  'Procedimiento lectura de sensores.
    orientacion_robot()                 'Posiciona cada motor a un valor específico.
    porte.0 = 1                          'Valida que ya se orientó para el punto 3.
end if

if (paso = 5) then

```

```

val1 = 77          'Valores para posición 5.
val2 = 29
val3 = 139
lectura_sensores() 'Procedimiento lectura de sensores.
orientacion_robot() 'Posiciona cada motor a un valor específico.
    porte.0 = 1     'Valida que ya se orientó para el punto 5.
end if
next paso
    goto inicio     'Manda el pc a la bandera inicio y todo el ciclo empieza
                    'nuevamente hasta que se complete.
end.

```

Por último, se tiene el código de programa del módulo PWM para el control de los motores DC, que se utiliza para proporcionar movimiento a las articulaciones del robot.

Program Pro_Final_PWM_Motor

```

dim salir as byte

main:
PORTC = $0F      'Portc 00001111 unos como salida y ceros como entradas.
Pwm_Init(1000)   'Inicializa el módulo PWM, frecuencia = 1kHz.
Pwm_Change_Duty(153) 'La relación de trabajo para pwm se calcula como
                    '(Porcentaje*255)/100.
                    'Con un porcentaje de 60% = 60*255/100 = 153.
    salir = 0     'Variable que se utiliza para terminar el módulo PWM.

Pwm_Start        'Comienza la modulación PWM.

```



```
while salir < 15
    salir = portc 'si salir '15 sale del ciclo y termina el módulo PWM.
wend
Pwm_Stop
end.
```


CONCLUSIONES

1. La selección del tipo de dispositivo que el robot utiliza para interactuar con su entorno, la fuente de impulsión y la configuración que se utilizará en su construcción es importante, ya que ésta determina la capacidad y complejidad del robot.
2. La mejor configuración para un robot industrial es del tipo articulado, ya que es capaz de alcanzar cualquier punto del espacio de su volumen de trabajo, tiene excelente precisión y control cuando se diseña un adecuado sistema de control de lazo cerrado, es capaz de elevar cargas pesadas utilizando un buen sistema de transmisión como el Harmonic Drive.
3. Los mejores actuadores que se pueden utilizar en el diseño del robot corresponden a los motores de corriente continua sin escobilla. Estos motores son capaces de proporcionar torques elevadas, el mantenimiento es mínimo y puede ser controlado por el módulo PWM del microcontrolador.
4. Los motores de corriente alterna son aptos para cargas pesadas, pero necesitan de la utilización de variadores de frecuencia para su operación. Los actuadores hidráulicos se utilizan para cargas muy pesadas y, los actuadores neumáticos, cuando se necesita rapidez, pero ambos sistemas carecen de precisión y de control.

5. La matriz homogénea de Foster describe de mejor manera el análisis del movimiento del robot. Los parámetros de Denavit-Hartenberg y la matriz homogénea solucionan el problema de cinemática directa tomando en cuenta la forma y tamaño del eslabón. La propuesta de Pieper soluciona de forma fácil y elegante la cinemática inversa.
6. El microcontroladores 16F877A por su versatilidad y accesibilidad en el mercado es el que mejor se adaptó como controlador para el robot.
7. MikroBasic es un lenguaje de alto nivel, debido a esa característica es fácil de programar que en lenguaje ensamblador, al compilar MikroBasic produce un archivo de lenguaje máquina, lo cual es interpretada por el microcontrolador como instrucciones.
8. Al seleccionar como guía la estructura del robot KUKA, se demuestra que se puede resolver la cinemática directa e inversa de cualquier forma estructural. Se requiere menor recurso computacional al resolver la cinemática inversa de forma geométrica.

RECOMENDACIONES

1. Establecer los valores de trayectoria, velocidad y torque adecuado a la aplicación específica del robot, ya que el diseño del mismo depende de éstas y en cada uno de los robots es específico.
2. Para el diseño de la estructura del robot, se deben elegir materiales livianos y resistentes como el aluminio o aleación de materiales con carbono para obtener mayor resistencia.
3. Seleccionar motores con suficiente torque, sistemas de alta eficiencia y los dispositivos electrónicos de última generación.
4. Elegir el microcontrolador dependiendo del tamaño del programa por ejecutar, y se recomienda que sean a partir de los de gama media.
5. Si se desea programar comportamientos complejos en el robot, utilizar lenguajes de programación que utilizan intérprete de Java.

BIBLIOGRAFÍA

1. *Actuadores*. [en línea]. <http://cfievalladolid2.net/tecno/cyr_01/robotica/sistema/actuadores.htm>. [Consulta 10 de diciembre de 2011].
2. *Actuadores hidráulicos*. [en línea]. <[http://www.elprisma.com/apuntes/ingenieria _ industrial / actuadoreshidraulicos / default2.asp](http://www.elprisma.com/apuntes/ingenieria_industrial_actuadoreshidraulicos_default2.asp)>. [Consulta: 7 de noviembre de 2011].
3. ARRIAGA, Erick. *Descripción del funcionamiento de una red de telefonía móvil LTE*. Trabajo de graduación de Ingeniería Eléctrica. Universidad de San Carlos de Guatemala, Facultad de ingeniería, 2015. 271 p.
4. BALLUFF. *Detección de objetos*. [en línea]. <http://assets.balluff.com/WebBinary1/870041_1302_ES.pdf>. [Consulta: 3 de julio de 2011].
5. BELU. *ROBOTICA* [en línea]. <<http://robotiica.blogspot.com/2007/10/historia-de-la-robtica.html>>. [Consulta: 15 marzo de 2011].
6. BOLTON, William. *Mecatrónica*. 3a ed. México: Alfaomega, 2006. 574 p.
7. BOUZIANE, Hrich. *Diseño e implementación de un acelerómetro, velocímetro móvil digital controlado por un PIC de MICROSHIP*.

[en línea]. <<https://es.scribd.com/document/61366981/memoria-PFC>>. [Consulta: 9 de enero de 2012].

8. CANTO Q., Carlos E. *Sensores de proximidad capacitivos*. [en línea]. <http://galia.fc.uaslp.mx/~cantocar/automatas/PRESENTACIONES_PLC_PDF_S/25_SENORES_CAPACITIVOS.PDF>. [Consulta: 2 de julio de 2011].
9. CARLETTI, Eduardo J. *Sensor de fuerza flexiforce*. [en línea]. <http://robots-argentina.com.ar/Sensores_FlexiForce.htm>. [Consulta: 11 de octubre de 2011].
10. CASTELLENOS TORRES, Daniel. *Actuadores*. [en línea]. <http://www.academia.edu/9037702/UNIDAD_2_ACTUADORES>. [Consulta: 11 de octubre de 2011].
11. GARCÍA, Salvador; MARTÍNEZ, José y POZANCO, Juan. *Robótica general*. [en línea]. <<http://www.infoplc.net/documentacion/22-robotica/1518-robotica-general>>. [Consulta: 7 de noviembre de 2011].
12. GARRION, Luis; OCHOA, Darío y VALVERDE, Juan A. *Análisis del funcionamiento del sensor de fuerza resistivo (FSR) con LabView* [en línea]. <http://www.datalights.com.ec/site2/images/stories/robotica/nap/nap_fsr.pdf>. [Consulta: 28 de agosto de 2011].
13. GONZÁLES, Víctor R. *Origen y desarrollo de la robótica*. [en línea]. <http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/historia.htm>. [Consulta: 4 de abril de 2011].

14. GONZALES, Víctor; LOPEZ, Antonio y CABERO, José. *Robots industriales*. [en línea]. <http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/industrial.htm>. [Consulta: 25 de febrero de 2011].
15. GROOVER, Mikell P.; WEISS, Mitchell y otros. *Robotica Industrial*. México: McGraw-Hill, 1990. 600 p.
16. *Manual de aplicación de encoder*. [en línea]. <<http://www.westmexico.com.mx/pfd/dynapar/catalogos/4.anual%20de%20Aplicacion%20de%20Encoders.pdf>>. [Consulta: 20 de julio de 2011].
17. MORALES, Gregoria y MARTÍNEZ, Cruz. *Métodos de programación de robots* [en línea]. <<https://es.slideshare.net/cruzmartinez9883/mtodosdeprogramacin-de-robots>>. [Consulta: 9 de enero de 2012].
18. NAVAISMO. *Especificaciones de mikrobasic* [en línea]. <<http://www.unrobotica.com/programas/MikrobasicEspa%C3%B1ol.pdf>>. [Consulta: 7 de noviembre de 2012].
19. PICs, *Electrónica y robótica*. [en línea]. <<http://picrobot.blogspot.com/2009/05/puente-h.html>>. [Consulta: 28 de agosto de 2012].
20. OCHOA, Oscar. *Robótica industrial*. [en línea]. <http://www.virtual.unal.edu.co/cursos/ingenieria/mecatronica/docs_curso/Anexos/TUTORIALcnc/DOCUMENTOS/TEORIA/ROBOTICA%20INDUSTRIAL.pdf>. [Consulta: 9 de enero 2012].

21. *Robots zoomórficos*. [en línea]. <<https://sites.google.com/a/unitecnica.net/dmhenao/articulos/zoomorficos>>. [Consulta: 25 de enero de 2011].
22. RODRÍGUEZ AVILA, Ernesto Jesús. *Introducción a la ingeniería del control automático*. México: McGraw-Hill, 1998. 374 p.
23. SEAS. *Neumática unidad didáctica 3*. [en línea]. <[http://campusvirtual.edu.uy/archivos/mecanicageneral/MATERIA L%20BIBLIOGRAFICO%20TECNICO%20PARA%20APOYO%20DOCENTE/APORTES%20VARIOS%20PARA%20DOCENTES/CURSO%20DE%20HIDRAULICA/ACTUADORES%20NEUMATICOS.pdf](http://campusvirtual.edu.uy/archivos/mecanicageneral/MATERIA%20BIBLIOGRAFICO%20TECNICO%20PARA%20APOYO%20DOCENTE/APORTES%20VARIOS%20PARA%20DOCENTES/CURSO%20DE%20HIDRAULICA/ACTUADORES%20NEUMATICOS.pdf)>. [Consulta: 4 de abril de 2011].
24. *Sensores capacitivos*. [en línea]. <http://www.balluff.com/NR/rdonlyres/2DF2E2D2-19354788BD135A3C875B9FD0/0/870041_41_0811_ES.pdf>. [Consulta: 3 de julio de 2011].
25. *Sensores de contacto*. [en línea]. <<http://proton.ucting.udg.mx/materias/robotica/r166/r76/r76.htm>>. [Consulta: 28 de agosto de 2011].
26. TODO ROBOT. *Tutorial sobre motores paso a paso (stepper motor)* [en línea]. <<http://www.todorobot.com.ar/informacion/tutorial%20stepper/stepper-tutorial.htm>>. [Consulta: 26 de marzo de 2011].
27. TORRES, F.; POMARES, J.; GIL, P. y otros. *Robots y sistemas sensoriales*. 2a ed. Madrid, España: Pearson Educación S.A., 2002. 480 p.

28. VILDÓSOLA C., Eugenio. *Actuadores*. [en línea]. <<http://www.aie.cl/files/file/comites/ca/abc/actuadores.pdf>>. [Consulta: 9 de enero de 2012].
29. Wikipedia. *Automatización industrial*. [en línea]. <http://es.wikipedia.org/wiki/Automatizaci%C3%B3n_industrial>. [Consulta: 9 de enero de 2012].

