



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE HIDALGO



INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA

INGENIERÍA INDUSTRIAL

“Los Robots en la Industria”

Monografía para obtener el título de:

Ingeniería Industrial

Presenta:

Juan Ruiz Quebrado

Director:

Dr. Omar Arturo Dominguez Ramírez

Mineral de la Reforma, Hidalgo, Octubre del 2007.

DEDICATORIA

A todas aquellas personas que en algún momento fueron parte importante de mi vida, a esos a amigos que se perdieron con el tiempo, en especial a esas personitas que a través de los años siguen siendo parte de mí.

A mi familia que siempre ha estado presente particularmente en los momentos difíciles:
A mis padres: a don Juan que ya se fue, a mi madre Ofelia que siempre esta presente;
a mis hermanos: Luís que es un ejemplo a seguir, Alejandro por su constancia, a Tere con mucho cariño, a Paty la inteligente de la casa, y Armando el pequeño; a mis tíos: Joaquín mi tío consentido, a Silvestre; a mis primos, sobrinos y parientes políticos Agrora, Arheli, Paco, Sarriá, Eder (sobre todo a mis consentidos ustedes saben quienes son).

A mi abuela Carrosi por su excelente carácter.

Y quiero dedicar este trabajo especialmente a las cuatro mujeres de mi vida:
Chave mi esposa, gracias por tu apoyo incondicional, pero sobre todo por tu aguante; a mis hijas: Karla, Ale y Chimal, que son la esencia de mi vida.

A TODOS USTEDES GRACIAS

Los quiere JUAN RUIZ QUEBRADO

AGRADECIMIENTOS

A los catedráticos que en algún momento me inspiraron a imitarlos por el dominio de su materia, pero especialmente por su **calidad humana**: Profa. Lolita, Profa. Angelina Palazuelos, Profa. Filena Ramírez Castillo, Prof. Jesús García, Ing. Benjamín Mohedano Pacheco, Ing. Gerardo Zapata Paz, Ing. Cesar Hugo Villena G., Ing. Roberto Pichardo Cabrera, Dr. Ciro Flores, Ing. Luís Espinoza Farias, Ing. José de Jesús Martínez, Ing. Carlos Manuel Pérez, Profa. Consuelo Altamirano, Química Zamira, Ing. Abelardo, Ing. Samuel López Hernández, Ing. Pedro Carrillo y en especial a mi asesor y amigo Dr. Omar Arturo Domínguez Rodríguez.

Con admiración JUAN RUIZ QUEBRADO

LISTA DE ABREVIATURAS Y ACRÓNIMOS

RIA	Instituto Americano de Robótica
AFNOR	Asociación Francesa de Normalización
AGV	Vehículos Guiados Automáticamente
JIT	Justo a Tiempo
PUMA	Máquina Universal Programable para el Armado
FMS	Sistema Flexible de Manufactura
FMC	Celda de Manufactura Flexible
CN	Control Numérico
PLC	Controlador Lógico Programable
PC	Computadora personal
CIM	Manufactura Integrada por Computadora
FA	Fabrica Automática
GDL	Grados De Libertad
RPL	Lenguaje de Programación de Robots
CAD	Dibujo Asistido por Computadora
OPL	Sistema de Programación Fuera de Línea
TCP	Posición del punto central de la herramienta del robot
ACL	Advanced control lenguaje

ÍNDICE DE FIGURAS

Número de figura	Leyenda de la figura	Página
2.1	AGV Electronics FLD	5
2.2	AGV Electronics FLA	6
2.3	AGV Electronics FLA	7
2.4	AGV Electronics LVQ	7
2.5	Brazo de Stanford	9
2.6	Shakey, primer vehículo controlado por inteligencia artificial	10
2.7	Primer robot industrial	11
2.8	Primer robot accionado eléctricamente IRb6 ASEA	11
2.9	Robot Qrio de Honda	12
2.10	Robot en la exploración especial	12
2.11	Robot empleado en la industria submarina	13
2.12	Robot utilizado en los laboratorios	15
2.13	Aplicaciones de la robótica en la medicina	16
2.14	El robot maestro-esclavo Da Vinci	17
2.15	Robot empleado en la medicina	17
2.16	Robot en la construcción	19
2.17	Sistema de inspección de la vasija del reactor	21

Número de figura	Leyenda de la figura	Página
2.19	Robot en la manufactura	23
2.20	Otra aplicación de robots	24
2.21	Celdas de mecanizado	25
2.22	Otra aplicación industrial	26
2.23	Los Robots en la soldadura	27
3.1	Muestra la similitud de las extremidades de un robot y las del ser humano	28
3.2	Robot industrial	35
4.1	Partes que componen un manipulador	46
4.2	Ilustración del <i>pitch</i> , <i>yaw</i> y <i>roll</i>	47
4.3	Muestra los movimientos de la muñeca del robot	47
4.4	Muestra una herramienta	48
4.5	Muestra otro tipo de herramienta	49
4.6	Muestra otra herramienta	49
4.7	Muestra otra herramienta	50
4.8	Muestra otra herramienta	50
4.9	En este robot el actuador final consiste de una serie de sensores	50
4.10	Muestra el espacio de trabajo de las diferentes configuraciones	54
4.11	Muestra una configuración cartesiana	58
4.12	Muestra una configuración cilíndrica	59

Número de figura	Leyenda de la figura	Página
4.13	Muestra una configuración esférica	59
4.14	Muestra una configuración brazo articulado	60
4.15	Muestra una configuración SCARA	60
5.1	El operador utiliza una interfaz de dispositivo de enseñanza para programar el manipulador	66
5.2	A menudo una célula de trabajo se modela en forma simple, con un conjunto de tramas unidas a objetos relevantes	69
5.3	Una celda de trabajo	71
6.1	Muestra un teach pendant o panel portátil.	110

RESUMEN

Esta monografía tiene como objetivo hacer una breve reseña de los robots empleados en la industria, pasando por los eventos más importantes en la historia de la robótica, desde sus inicios en la década de los cuarenta del siglo pasado hasta las aplicaciones en nuestros días. Se hace referencia a los robots utilizados en la industria de la manufactura, detallando los elementos constitutivos de una celda flexible de manufactura, así como también los diferentes elementos constitutivos de los robots industriales y los lenguajes de programación utilizados en dichos robots, mencionando algunas de las aplicaciones en otras áreas industriales.

ÍNDICE GENERAL

	Página
Lista de abreviaturas y acrónimos	v
Índice de figuras	ix
Índice general	ix
Capítulo 1	“Introducción”
1.1 Antecedentes	1
1.2 Objetivo general	1
1.3 Objetivos específicos	2
1.4 Contribución de la monografía	2
1.5 Organización de la monografía	2
Capítulo 2	“La robótica en la industria”
2.1 Introducción	4
2.2 Antecedentes de la robótica industrial	8
2.3 La robótica en el espacio	12
2.4 La robótica en los vehículos submarinos	13
2.5 La robótica en la educación	14
2.6 La robótica en los laboratorios	14
2.7 La robótica en la medicina	15
2.8 La robótica en la construcción	18
2.9 La robótica en la industria nuclear	19
2.10 La robótica en la manufactura	21
Capítulo 3	“La robótica en la manufactura”
3.1 Introducción	28
3.2 Componentes de una celda de manufactura flexible	31
3.3 Operación de los robots en los procesos de manufactura	40
3.4 Perspectivas de la robótica en la manufactura	41
Capítulo 4	“Los robots industriales”
4.1 Introducción	43
4.2 Los subsistemas de un robot industrial	44
4.3 Elementos que forman un robot	45
4.4 Principales características de los robots	52
4.5 Configuraciones morfológicas de los robots	57

	4.6 Transmisiones y reductores en los robots	61
	4.7 Actuadores en los robots	62
	4.8 Sensores en los robots	63
Capítulo 5	“Lenguajes y sistemas de programación de robots”	
	5.1 Introducción	65
	5.2 Niveles de programación de robots	66
	5.3 Requerimientos de un lenguaje de programación de Robots	70
	5.4 Sistemas de programación fuera de línea	77
	5.5 Controlador lógico programable	79
	5.6 Ejemplos de lenguajes de programación de robots	83
	5.7 ACL (Advanced control lenguaje)	89
	5.8 SCORBASE para Windows	93
	5.9 Perspectivas de los lenguajes de programación	95
Capítulo 6	“Ejemplos de programación de robots”	
	6.1 Descripción del robot Scorbob –ER Vplus	98
	6.2 Descripción del Scorbob – ER IX	101
	6.3 Ejemplos de los lenguajes de programación utilizados en los robots	103
	6.4 Estudio del Robot Scorbob ER-VII	108
	Glosario de términos	121
	Referencias	123

CAPÍTULO 1

INTRODUCCIÓN

1.1 Antecedentes

Con la evolución del hombre a través del tiempo ha surgido la necesidad de inventar diferentes herramientas adecuadas para cada una de las ocupaciones que se han desarrollado desde la aparición de la humanidad hasta la actualidad, para satisfacer estas necesidades se han desarrollado infinidad de herramientas que van de las mas primitivas hasta las mas sofisticadas tecnologías de nuestros días.

Una herramienta importante en la actualidad es sin duda, “el robot” que se utiliza en muy variados campos industriales, es por eso la necesidad de adquirir la mayor cantidad de conocimientos relacionados al robot. Principalmente en la industria de la manufactura que es uno de los campos de trabajo de los ingenieros industriales, y de los diferentes lenguajes utilizados para la programación de dichos robots.

1.2 Objetivo general

Conocer los inicios de la robótica, los acontecimientos mas importantes en el desarrollo de dicha área del conocimiento, saber las diferentes ramas de la industria donde encuentra cabida la utilización de los robots, poniendo especial atención en los robots utilizados en la manufactura, así como los componentes constitutivos de una celda flexible de manufactura. Adquirir conocimientos tanto de los subsistemas que forman un robot industrial y de los elementos de que esta compuesto; de las diferentes configuraciones que existen y los lenguajes utilizados en la programación de diferentes robots industriales.

1.3 Objetivos específicos

Un aspecto importante es conocer los subsistemas y elementos constitutivos de un robot industrial, así como su funcionamiento y las diferentes configuraciones morfológicas existentes.

Especialmente importante es conocer sobre los elementos constitutivos de una celda flexible de manufactura, de un sistema flexible de manufactura.

Otro aspecto importante es el conocimiento de los diferentes lenguajes de programación utilizados en los robots industriales y otros dispositivos utilizados para el mismo fin.

1.4 Contribución de la monografía

Se pretende contar con un documento que proporciona a quien este interesado en conocer de forma rápida y sencilla la información necesaria para adquirir conocimientos de los robots industriales, elementos constitutivos de una celda flexible de manufactura, los elementos que la forman, y los lenguajes de programación utilizados para la programación de los robots industriales.

1.5 Organización de la monografía

El presente trabajo esta organizado de la siguiente manera:

Capítulo 1: el capítulo primero tiene el nombre de “Introducción”, en donde se inicia con los antecedentes, se plantea el objetivo general de la monografía, los objetivos específicos. La contribución de la monografía y finalmente la organización del trabajo.

Capítulo 2: este capítulo lleva por nombre “La robótica en la industria”, e inicia con una introducción a la robótica, donde se mencionan los acontecimientos más importantes en el desarrollo de esta área del conocimiento desde sus inicios hasta nuestros días; posteriormente se describe brevemente la aplicación de la robótica en diferentes campos industriales, en el espacio, en vehículos submarinos, en educación, en laboratorios, en medicina, en la construcción, en la manufactura y en la industria nuclear.

Capítulo 3: este capítulo lleva por nombre “La robótica en la manufactura”, el cual inicia con una introducción, después se detallan los componentes de una celda flexible de manufactura; se describe la operación de los robots en los procesos de manufactura, y por último las perspectivas de la robótica en la manufactura.

Capítulo 4: este capítulo lleva por nombre “Los robots industriales”, se inicia con una breve introducción de lo que es un robot, se continúa con los subsistemas y elementos que integran a un robot industrial, también se enuncian las principales características y configuraciones de los robots, así como la descripción de las transmisiones, reductores, actuadores y sensores de los robots industriales.

Capítulo 5: este capítulo tiene por título “Lenguajes y sistemas de programación de robots”, en el que se escribe una introducción, los diferentes niveles de programación existentes para los robots, las necesidades que debe satisfacer un lenguaje de programación para los robots, también se mencionan los sistemas de programación fuera de línea y se incluye una breve historia y descripción del controlador lógico programable (PLC), se mencionan algunos de los lenguajes existentes para la programación de los robots, se hace una breve descripción del Lenguaje de Control avanzado (ACL) y el SCORBASE para Windows; finalmente se escriben las perspectivas de los lenguajes de programación empleados en los robots.

Capítulo 6: este capítulo lleva el nombre de “Ejemplos de programación de robots”, se inicia con la descripción de los robots Scorbot-ER Vplus, Scorbot-ER IX y Scorbot ER-VII. Y se concluye con algunos ejemplos de la programación de dichos robots.

CAPÍTULO 2

LA ROBÓTICA EN LA INDUSTRIA

2.1 Introducción

La robótica tiene su origen en la ciencia ficción, el término robot fue utilizado por primera vez en una obra escrita en 1920 (Rossum's Robots Universales) por el checoslovaco Karel Capek. La palabra checa robota significa trabajador obligado, cuando se tradujo al inglés, la palabra se redujo a “robot”.

El término Robótica fue empleado como disciplina hasta 1942 por el escritor Americano de origen Ruso Isaac Asimov en su obra "Runaround", Asimov formulo tres leyes fundamentales que en términos sencillos consignan a los robots como elementos que obedecen órdenes humanas, que interactúan con estos y no causan daño alguno.

La robótica es la ciencia que estudia este tipo de dispositivos, y la podemos definir como: diseño, fabricación y utilización de máquinas automáticas programables con el fin de realizar tareas repetitivas como el ensamble de automóviles, aparatos, entre otras actividades.

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

Existen ciertas dificultades a la hora de establecer una definición formal de lo que es un robot industrial. Una de ellas surge de la diferencia conceptual entre el mercado japonés y el euro-americano de lo que es un robot y lo que es un manipulador. Así, mientras que para los japoneses un robot industrial es cualquier dispositivo mecánico dotado de articulaciones móviles destinado a la manipulación, el mercado occidental es más restrictivo, exigiendo una mayor complejidad, sobre todo en lo relativo al control. Otra dificultad, y centrándose ya en el concepto occidental, aunque existe una idea común acerca de lo que es un robot industrial, no es fácil ponerse de acuerdo a la hora de establecer una definición formal. Además, la evolución de la robótica ha ido obligando a diferentes actualizaciones de su definición.

En la actualidad el término robot tiene la siguiente definición según la RIA (Robot Institute of America).

Un robot es un manipulador reprogramable, multifuncional ideado para el transporte de materiales, partes, herramientas o sistemas especializados, con movimientos variados y programados para la realización de tareas diversas.

El cual se ha aplicado a una amplia gama de dispositivos mecánicos, generándose dos variantes, los manipuladores y los móviles.

Una definición establecida por la Asociación Francesa de Normalización (AFNOR), define el manipulador como mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante un dispositivo lógico.

Los robots manipuladores generalmente tienen forma de brazo articulado, en cuyo extremo incorporan elementos de sujeción o herramientas. Realizan tareas repetitivas en industrias de automoción, fabricación mecánica o electrónica, en las que se emplean para montar y mover piezas o componentes, ajustar, soldar, pintar, etc., no tienen cuerpo ni pies, y casi siempre están pegados a una mesa de trabajo. En la industria existe una gran variedad de estos.

Los robots móviles son comunes en las industrias, también conocidos como “Vehículos Guiados Automáticamente” (AGV), pueden definirse como vehículos autopropulsados, capaces de seguir automáticamente una trayectoria variable según un patrón flexible, es decir fácilmente modificable.

Los vehículos guiados automáticamente, son el sistema de transporte flexible más utilizado, en las líneas complejas de fabricación flexible, en donde el recorrido de las piezas entre máquinas y entre éstas y los almacenes es aleatorio y las distancias son considerables.



Fig. 2.1 AGV Electronics FLD [17].

Estos vehículos se extienden desde repartidores con capacidad de carga de unos pocos kilogramos hasta transportadores de más de 100 toneladas, pueden variar su ambiente de trabajo desde oficinas alfombradas hasta los más hostiles sectores fabriles. Éstos factores sumados a la flexibilidad que permiten, han ampliado el campo de aplicación de los AGV de forma tal que en los últimos 10 años, se han extendido a las más diversas áreas como industrias químicas, cerámica, metalmecánica, alimenticias, textiles, depósitos de todo tipo, empresas de servicios como correos, y otros sitios como aeropuertos y hoteles.



Fig. 2.2 AGV Electronics FLA [17]

La gestión en tiempo real de este tipo de transporte con piezas diferentes en trayectos distintos es muy compleja, y solo está al alcance de un sistema de control por ordenador que recoge la múltiple información enviada por los sensores y, en función de la misma, coordina y regula el tráfico. Mientras funcionan en un ambiente de fabricación o de servicio los AGV requieren de dicho ordenador a bordo o centralizado.

Este sistema de transporte puede desglosarse en tres partes:

- El vehículo
- El sistema de tráfico
- El sistema de gestión

Los AGV modernos usualmente se comunican por ondas de radio FM para transmitir datos desde y hacia los vehículos. Las comunicaciones por radio crean por sí mismas muchas libertades para el diseño de sistemas de AGV.



Fig. 2.3 AGV Electronics FLA [17]

Entre las principales aplicaciones no industriales de los robots. Es necesario mencionar su utilización en plantas de energía nuclear, en la exploración submarina, la minería, construcciones, agricultura, medicina, etc.



Fig. 2.4 AGV Electronics LVQ [17]

2.2 Antecedentes de la robótica industrial

Los primeros trabajos que dieron origen a la construcción de los robots que existen en la actualidad se remontan a los últimos años de la década de los cuarenta del siglo próximo pasado, después de la Segunda Guerra Mundial. Donde empiezan los programas de investigación en Oak Ridge y Argonne National Laboratories para desarrollar manipuladores mecánicos controlados de forma remota con el objetivo de manejar materiales radioactivos. Estos sistemas eran de tipo “maestro-esclavo”, diseñados para reproducir fielmente los movimientos de mano y brazo realizados por un operario humano. El manipulador maestro era guiado por el usuario a través de una secuencia de movimientos, mientras que el manipulador esclavo duplicaba el movimiento tan fielmente tal como le era posible. El trabajo sobre manipuladores “maestro-esclavo” fue seguido rápidamente por sistemas más sofisticados capaces de operaciones repetitivas autónomas.

A mediados de los años cincuenta, George C. Devol desarrolló un dispositivo al que llamó “dispositivo de transferencia programada articulada”, un manipulador cuya operación podía ser programada, y por tanto cambiada, además podía seguir una secuencia de movimientos determinados por las instrucciones del programa.

Trabajos posteriores de Devol y Joseph F. Engelberger condujeron al primer robot industrial, introducido por Unimation Inc., en 1959. La clave de este dispositivo era el uso de una computadora en combinación con un manipulador para producir una máquina que podía ser “enseñada” para realizar una variedad de tareas de forma automática. Estos robots se podían reprogramar y cambiar la herramienta a un costo relativamente bajo para efectuar otros trabajos cuando cambiaban los requisitos de fabricación. Aunque los robots programados ofrecían una herramienta de fabricación nueva y potente, en los años sesenta se descubrió que la flexibilidad de estas máquinas se podía mejorar significativamente mediante el uso de una realimentación sensorial.

Al inicio de la década de los sesenta, H. A. Ernest publicó el desarrollo de una mano mecánica controlada por computadora con sensores táctiles. Este dispositivo, llamado el MH-1, podía “sentir” bloques y usar esta información para controlar la mano de manera que apilaba los bloques sin la ayuda de un operario. Este trabajo es uno de los primeros ejemplos de un robot capaz de una conducta adaptativa en un entorno razonable no estructurado. Este programa de investigación posteriormente evolucionó como parte del proyecto MAC, y se le añadió una cámara de televisión para iniciar la investigación sobre la percepción en la máquina.

Durante el mismo periodo, Tomovic y Boni (1962) desarrollaron una mano prototipo provista con un sensor de presión que detectaba el objeto y proporcionaba una señal de alimentación de entrada a un motor para iniciar uno de dos modelos de aprehensión.

En 1963, la American Machine y Foundry Company (AMF) introdujo el robot comercial VERSATRAN. Al inicio de este mismo año, se desarrollaron diversos diseños de brazos para manipuladores, tales como el brazo Roehampton y el de Edinburgh.

A finales de los años sesenta, McCarthy y sus colegas en el Stanford Artificial Intelligence Laboratory publicaron el desarrollo de una computadora con manos, ojos y oídos, es decir, manipuladores, cámaras de TV y 22 micrófonos. Durante este periodo, Pieper (1968) estudió el problema cinemático de un manipulador controlado por computadora. Mientras tanto, en Japón empezaron a entender el potencial de los robots industriales. Ya que en 1968, la compañía japonesa Kawasaki Heavy Industries negoció una licencia con Unimation Inc., para sus robots.

Uno de los desarrollos poco usuales sucedió en 1969 cuando se desarrolló un camión experimental por la General Electric para la Armada Americana. En el mismo año se desarrollo el brazo Boston y al año siguiente el brazo de Stanford, que estaba equipado con una cámara y controlado por computadora. Algunos de los trabajos más serios en robotica comenzaron cuando estos brazos se utilizaron como robots manipuladores.

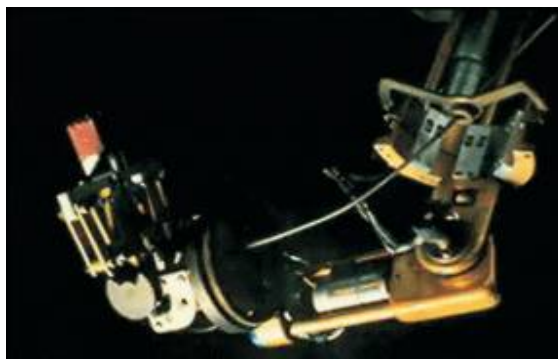


Fig. 2.5 Brazo de Stanford [22].

Las primeras aplicaciones industriales en Europa, de robots industriales en cadenas de fabricación de automóviles, datan de los años 1970 y 1971. En este último año, Kahn y Roth analizan el comportamiento dinámico y el control de un brazo manipulador.

Durante la década de los 70 del siglo pasado, la investigación en robótica se centra en gran parte en el uso de sensores externos para su utilización en tareas de manipulación. Por estos años es cuando se consolida definitivamente la presencia de robots en las cadenas de montaje y plantas industriales en el ámbito mundial. Se lanza también Shakey, el primer vehículo autoguiado controlado por inteligencia artificial.



Fig. 2.6 Shakey, primer vehículo controlado por inteligencia artificial [22].

En 1972 se desarrolló en la universidad de Nottingham, Inglaterra, El SIRCH un robot capaz de reconocer y orientar objetos en dos dimensiones. En este año la empresa japonesa Kawasaki instala su primera cadena de montaje automatizada en Nissan, en Japón usando robots de Unimation.

En 1973, Bolles y Paul utilizan realimentación visual en el de brazo Stanford para el montaje de bombas de agua de automóvil. También este mismo año, la compañía sueca ASEA (después ABB), lanza al mercado su familia de robots IRB 6 e IRB 60, para funciones de perforación de piezas.

En 1978 se introdujo el robot PUMA (Programmable Universal Machine for Assembly) para tareas de montaje por Unimation, basándose en diseños obtenidos en un estudio de la General Motors. En este mismo año el robot T3 se adaptó y programó para realizar operaciones de taladrado y circulación de materiales en componentes de aviones, bajo el patrocinio de Air Force.

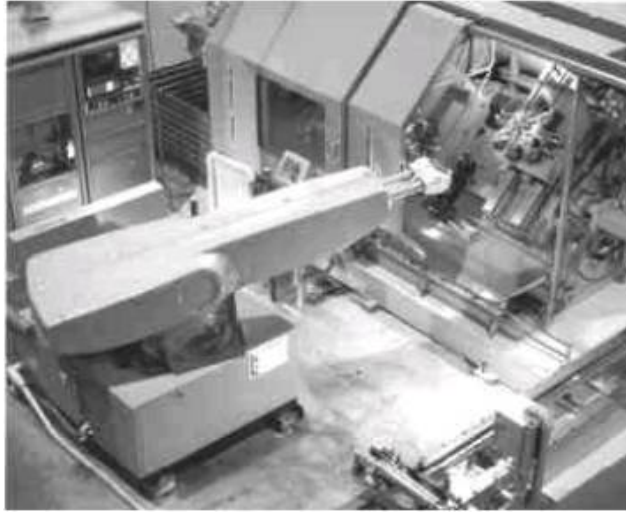


Fig. 2.7 Primer robot industrial [22].

En 1979 Japón introduce el robot SCARA (Selective Compliance Assembly Robot Arm). En la década de los 80 se avanza en las técnicas de reconocimiento de voz, detección de objetos móviles y factores de seguridad. También se desarrollan los primeros robots en el campo de la rehabilitación, la seguridad, con fines militares y para la realización de tareas peligrosas. Así por ejemplo, en 1982, el robot Pedesco, se usa para limpiar un derrame de combustible en una central nuclear. También se pone un gran énfasis en los campos de visión artificial, sensores táctiles y lenguajes de programación. Gracias a los primeros pasos dados por compañías como IBM o Intelledex Corporation, que introdujeron 1984 el modelo ligero de ensamblaje 695, basado en el microprocesador Intel 8087 y con software Robot Basic, una modificación del Microsoft Basic, actualmente se tiende al uso de una interfaz (el ordenador) y diversos lenguajes de programación especialmente diseñados, que evitan el “cuello de botella” que se producía con la programación “clásica”.



Fig. 2.8 Primer robot accionado electricamente IRB6 ASEA [22].

Para 1997 HONDA presenta P3 un enorme robot humanoide. En 1999 la empresa SONY lanza “Aibo” un perro-robot. Para el año 2003, aquel robot humanoide de SONY, Qrio (“*Quest for curiosity*”), se convierte en el primer humanoide comercial completamente autónomo capaz de correr.



Fig. 2.9 Robot Qrio de Honda [18].

2.3 La robótica en el espacio

Como en todas las industrias, la exploración espacial posee problemas inherentes a este campo en la aplicación de los robots. El medio ambiente es hostil para el ser humano, quien requiere un equipo de protección muy costoso para el espacio. Muchos científicos han hecho sugerencia en el uso de Robots para continuar con los avances en la exploración espacial; pero en la actualidad no existe una automatización tan precisa para ésta aplicación, el ser humano aún no ha podido ser reemplazado por robots. Los robots que se utilizan en la exploración espacial son del tipo teleoperador.

En Marzo de 1982 el transbordador Columbia fue el primero en utilizar este tipo de robots, aunque el ser humano participa en la realización del control.



Fig. 2.10 Robot en la exploración espacial [19].

Algunas investigaciones están encaminadas al diseño, construcción y control de vehículos autónomos, los cuales llevarán a bordo complejos laboratorios y cámaras muy sofisticadas para la exploración de otros planetas.

En Noviembre de 1970 los rusos consiguieron el alunizaje del Lunokhod 1, el cual poseía cámaras de televisión, sensores y un pequeño laboratorio, era controlado remotamente desde la tierra.

En Julio de 1976, los norteamericanos aterrizaron en Marte el Viking 1, llevaba a bordo un brazo robotizado, el cual recogía muestras de piedra, tierra y otros elementos las cuales eran analizadas en el laboratorio que fue acondicionado en el interior del robot. Por supuesto también contaba con un equipo muy sofisticado de cámaras de vídeo.

2.4 La robótica en los vehículos submarinos

Básicamente dos eventos provocaron el uso de los vehículos submarinos. Uno fue el accidente de un avión de la Air Indian que se estrelló en el océano atlántico cerca de las costas de Irlanda un vehículo submarino guiado remotamente, normalmente utilizado para el tendido de cable, fue utilizado para encontrar y recobrar la caja negra del avión. Otro fue el descubrimiento del Titanic en el fondo del mar, donde había permanecido después del choque con un iceberg en 1912, cuatro kilómetros debajo de la superficie. Un vehículo submarino fue utilizado para encontrar, explorar y filmar el hallazgo.

En la actualidad muchos de estos vehículos submarinos se utilizan en la inspección y mantenimiento de tuberías que conducen petróleo, gas o aceite en las plataformas oceánicas; en el tendido e inspección del cableado para comunicaciones, para investigaciones geológicas y geofísicas en el suelo marino.

La tendencia hacia el estudio e investigación de este tipo de robots se incrementará a medida que la industria se interese en el uso de dichos dispositivos, a la vez que se obtendrían grandes beneficios si se consigue una tecnología segura para la exploración del suelo marino y la explotación del mismo.



Fig. 2.11 Robot empleado en la industria submarina [19]

2.5 La robótica en la educación

Los robots están apareciendo en los salones de clases de tres distintas formas:

- Los programas educacionales utilizan la simulación de control de robots como un medio de enseñanza. Un ejemplo palpable es la utilización del lenguaje de programación del robot Peleo, el cual es un subconjunto de Pascal; este es utilizado por la introducción a la enseñanza de la programación.
- El uso más común es el robot tortuga en conjunción con el lenguaje LOGO para enseñar ciencias computacionales. LOGO fue creado con la intención de proporcionar al estudiante un medio natural y divertido en el aprendizaje de las matemáticas.
- El uso de los robots en los salones de clases. Una serie de manipuladores de bajo costo, robots móviles, y sistemas completos han sido desarrollados para su utilización en los laboratorios educativos. Debido a su bajo costo muchos de estos sistemas no poseen una fiabilidad en su sistema mecánico, tienen poca exactitud, no están provistos de sensores y en su mayoría carecen de software.

2.6 La robótica en los laboratorios

Los robots están encontrando un gran número de aplicaciones en los laboratorios. Llevan a cabo con efectividad tareas repetitivas como la colocación de tubos de pruebas dentro de los instrumentos de medición. En esta etapa de su desarrollo los robots son utilizados para realizar procedimientos manuales automatizados. Un típico sistema de preparación de muestras consta de un robot y una estación de laboratorio, la cual contiene balanzas, dispensarios, centrifugados, racks de tubos de pruebas.

Las muestras son movidas desde la estación de laboratorios por el robot bajo el control de procedimientos de un programa.

Los fabricantes de estos sistemas mencionan tener tres ventajas sobre la operación manual: incrementan la productividad, mejoran el control de calidad y reducen la exposición del ser humano a sustancias químicas nocivas.

Las aplicaciones subsecuentes incluyen la medición del pH, viscosidad, y el porcentaje de sólidos en polímeros, preparación de plasma humano para muestras para ser

examinadas, calor, flujo, peso y disolución de muestras para presentaciones espectromáticas.



Fig. 2.12 Robot utilizado en los laboratorios [19].

2.7 La robótica en la medicina

De entre las varias aplicaciones de la robótica a la medicina destaca la cirugía. Las primeras aplicaciones de la robótica a la cirugía del cerebro datan del año 1982. En esta fecha se comenzó en Memorial Medical Center de Long Beach (California) un programa cuyo objetivo consistía en utilizar un robot comercial (Puma 260) para realizar determinadas operaciones de neurocirugía. Desde entonces se han puesto a punto varios sistemas que, con la ayuda de un scanner, un ordenador registre toda la información necesaria del cerebro para que el equipo médico decida el punto exacto donde debe ser realizada la incisión, donde penetrara la sonda para obtener una muestra para realizar una biopsia. El robot, que se encuentra perfectamente situado con respecto al paciente, porta en su extremo los instrumentos necesarios para realizar la incisión, tomar la muestra, etc.

La utilización de un robot conectado al ordenador permite que tanto la incisión como la toma de la muestra se realicen con la mayor precisión y en un tiempo notablemente inferior al que se consumiría en caso de emplear el sistema habitual. Además, se descarga al cirujano de la mecánica de ciertas tareas como el correcto posicionamiento de los instrumentos de cirugía con respecto al cráneo del paciente, permitiendo una mayor concentración en el seguimiento y control de la operación.



Fig. 2.13 Aplicaciones de la robótica en la medicina [19]

También, otro posible beneficio de la aplicación de la robótica a la cirugía se encuentra en el tele diagnóstico y la tele cirugía. Esta última consiste en la operación remota de un paciente mediante un tele manipulador.

En julio de 1993 un robot SCARA en Milán (Italia) realizó sobre el hígado de un cerdo una biopsia y una incisión para introducir un laparoscopio. El robot estaba siendo teleoperado en tiempo real desde 14000km de distancia en el Jet Propulsión Laboratory. De California. Las órdenes del cirujano y la información procedente de la sala de operaciones eran intercambiadas a través de 2 satélites de comunicaciones y redes de fibra óptica.

Esta primera experiencia en operación remota permite asegurar que este tipo de intervenciones sobre pacientes humanos podrá ser una realidad en un futuro próximo. Su justificación puede encontrarse en el tratamiento de pacientes situados en localizaciones difícilmente alcanzables (espacio, plataformas submarinas, minería, etc.). No obstante hay que considerar que aun existen importantes dificultades técnicas, como es el retraso en las comunicaciones y el elevado costo.

Otra aplicación de la robótica es en la construcción de prótesis para disminuidos físicos que se adaptan al cuerpo y están dotados de potentes sistemas de mando. Con ellos se logra igualar con precisión los movimientos y funciones de los órganos o extremidades que suplen.



FIG. 2.14 El robot maestro-esclavo Da Vinci [19].

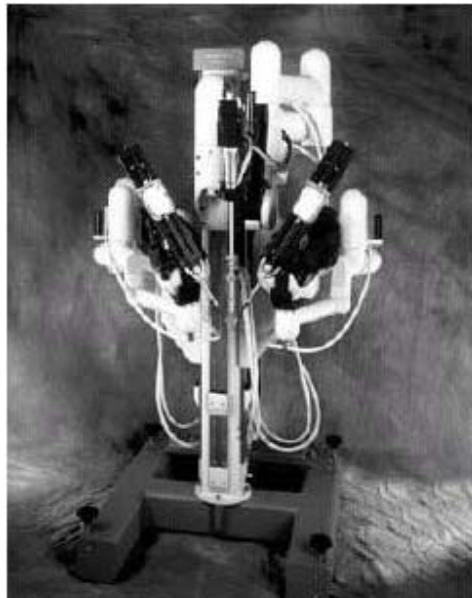


Fig. 2.15 Robot empleado en la medicina [19]

2.8 La robótica en la construcción

El sector de la construcción es, en la mayoría de los países industrializados, uno de los que moviliza mayor número de recursos económicos y humanos. No es de extrañar que desde hace más de una década se estén desarrollando gran número de sistemas robotizados, orientados a automatizar en lo posible algunas de las múltiples labores que entran a formar parte de los procesos constructivos.

En este tipo de aplicaciones de la robótica, es Japón el país que cuenta con mayor número de sistemas en funcionamiento. En algunos casos se trata de robots parcialmente teleoperados, contruidos a partir de maquinaria convencional (grúas, excavadoras). En otros es maquinaria específicamente construida para resolver un proceso concreto.

Al analizar las condiciones existentes para la introducción de robots en la industria la construcción se deben considerar entre otras a las siguientes conclusiones:

- Las condiciones de trabajo son complejas.
- Los robots deben tener capacidad de locomoción y cierto grado de inteligencia.
- Deben manejar piezas pesadas y de grandes dimensiones.
- Las operaciones a realizar son complejas, variadas y poco repetitivas.
- Los robots deben ser fácilmente transportables a la obra.

Con estos condicionantes, las posibles tareas robotizables dentro de la construcción de edificios (comerciales, industriales o residenciales) podrían agruparse en:

Operaciones de colocación de elementos:

- Construcción mediante colocación repetitiva de estructuras básicas (ladrillos, Bloques).
- Posicionamiento de piezas, normalmente grandes y pesadas (vigas, etc.). Unión de diferentes piezas que componen una estructura (soldadura, remaches).
- Sellado de las uniones entre diferentes piezas.

Operaciones de tratamiento de superficies:

- Acabado de superficies (pulido).
- Recubrimiento de superficies con pintura, barniz.
- Extensión de material sobre la superficie (cemento, espuma aislante).

Operaciones de relleno:

- Vertido de cemento.
- Excavación para la preparación de terrenos y movimiento de tierras.
- Rellenado con tierra de volúmenes vacíos.

Otras:

- Inspección
- Control.



Fig. 2.16 Robot en la construcción [19].

2.9 La robótica en la industria nuclear

Por sus características, el sector nuclear es uno de los más propicios para la utilización de robots de diseño específico. Entre las diversas aplicaciones que existen, por su especial relevancia, podemos mencionar las relativas a las operaciones de mantenimiento en zonas contaminadas y de manipulación de residuos.

La inspección de los tubos del generador de vapor en un reactor nuclear: las operaciones de inspección y mantenimiento de las zonas más contaminadas de una central nuclear de producción de energía eléctrica son por su naturaleza largas y costosas. De realizarlas manualmente, el tiempo de exposición de los trabajadores a la radiación es un factor crítico que, junto con el elevado costo que supone una interrupción temporal del funcionamiento del sistema en cuestión, justifica sin lugar a dudas la utilización de sistemas robotizados, total o parcialmente teleoperado, que sustituyan al trabajador humano.

En el generador de vapor se produce el intercambio de calor entre el fluido primario y secundario. Para ello, dentro de la vasija del generador, se encuentran dispuestas en forma matricial los tubos por los que circula el fluido receptor del calor.

El inevitable desgaste de estos tubos obliga a realizar periódicamente labores de inspección, para que en el caso de que alguno se encuentre dañado, inutilizarlo, poniendo en funcionamiento alguno de los tubos de reserva que a tal fin se han dispuesto en el generador.

Para realizar esta labor de manera automática puede utilizarse un robot de desarrollo específico que, introducido en la vasija, posicione una sonda de inspección en la boca de cada tubo. Esta, empujada por el interior del tubo, proporcionara información sobre el estado mismo. Es preciso considerar que el robot se introduce en la vasija mediante un sistema mecánico que, junto con los posibles errores en la disposición matricial de los tubos, obliga a que el robot trabaje, bien con ayuda de tele operación, o bien con sistemas sensoriales externos como visión láser, que proporcionen la posición real relativa entre el extremo del robot y los tubos.

Manipulación de residuos radioactivos: la industria nuclear genera una cantidad considerable de residuos radioactivos de baja contaminación (vestimentas, envases de plástico, papel, etc.) y de alta contaminación (restos de las células del reactor, materiales en contacto directo prolongado con las zonas radioactivas, etc.). La forma tamaño y peso de estos desechos es variable y su manipulación tiene por objeto final su envase en contenedores especiales, que son posteriormente transportados y almacenados (lo que origina una nueva problemática).

Para manipular remotamente estos residuos se hace uso tanto de manipuladores con unión mecánica y seguimiento directo del proceso por parte del operador a través de un cristal (en caso de baja contaminación), y de sistemas con mando remoto por radio o cable en el caso de contaminación elevada. Estos manipuladores permiten la flexibilidad necesaria para manipular elementos de peso variable y forma no definida.

Además, es preciso considerar la importancia que tiene la optimización del espacio

ocupado por los residuos en su almacenamiento, por lo que antes de su envasado en los contenedores puede ser preciso fragmentarlos.



Fig. 2.17 Sistema de inspección de la vasija del reactor [19].

2.10 La robótica en la manufactura

En la actualidad el mercado se caracteriza por una disminución en la tasa de crecimiento, una diversificación del producto para adaptarse a necesidades específicas de los clientes, una mayor exigencia de calidad y una competencia de ámbito mundial. Los clientes son cada día más exigentes y basan sus decisiones de compra en elaborados estudios de rentabilidad sobre las distintas opciones que se les presentan.

Los productos ofrecen cada día mas y mejores prestaciones, una amplia gama de variantes para adaptarse a los gustos y necesidades de los clientes, una gran calidad “cero defectos “ y un ciclo de vida corto debido a las constantes incorporaciones de nuevas y sofisticadas tecnologías.

Según sea el grado de dependencia del proceso respecto del producto pueden establecerse cuatro tipos principales de sistemas de fabricación:

- Independiente: el proceso no depende del diseño del producto. Un taller con máquinas herramientas convencionales (fresadoras, torno, dobladoras).
- Programable: el proceso puede adaptarse a una variedad de productos. (maquinas de control numérico).
- Flexibles: un proceso con una configuración única puede fabricar una amplia gama de productos. (células de fabricación flexible).

- Dedicados: el proceso solo es útil para un único diseño del producto. (máquinas transfer).

En el contexto de los sistemas de manufactura, la flexibilidad puede ser definida como la habilidad del sistema para ajustar o adaptar sus recursos a los cambios en el mismo, debidos normalmente a factores tanto internos como externos, tales como: productos, procesos, demandas, cargas, máquinas, empleados.

Algunos tipos de flexibilidad son:

- Flexibilidad en las máquinas
- Flexibilidad en las herramientas
- Flexibilidad en el sistema de manejo de materiales
- Flexibilidad en el sistema de almacenamiento
- Flexibilidad en las comunicaciones
- Flexibilidad en las rutas



Fig. 2.18 Robot Kawasaki [20].

Un sistema de fabricación flexible esta formado por máquinas e instalaciones técnicas enlazadas entre si por un sistema común de transporte y control, de forma que exista la posibilidad, dentro de un margen determinado, de realizar diversas tareas correspondientes a piezas diferentes sin necesidad de interrumpir el proceso de fabricación para el reequipamiento del conjunto.

La fabricación flexible es un sistema que permite la producción automática y simultanea de una familia de piezas distintas minimizando, y a veces eliminando, los costos

adicionales por cambio de fabricación, y que proporciona por tanto una productividad y unos costos unitarios reservados hasta ahora a la fabricación de grandes series.

Es, por tanto una tecnología muy adecuada para los talleres con gran variedad de piezas en series pequeñas, o con productos de rápida obsolescencia y elevado nivel de cambios y modificaciones.



Fig. 2.19 Robot en la manufactura [21].

La tecnología de fabricación flexible ha evolucionado hacia un concepto de modularidad a partir de las celdas flexibles, de forma que permite a todas las empresas, incluso a las más pequeñas, iniciarse en esta tecnología mediante la instalación de una sola celda.

La celda flexible es una unidad pequeña, con una o varias estaciones de trabajo, dentro de un sistema de manufactura, una estación de trabajo puede contener una o varias máquinas, y cada máquina efectúa una operación diferente en la pieza. Las celdas flexibles generalmente están dotadas de tornos, fresadoras, rectificadoras, electroerosionadoras, dobladoras y centros de maquinado, también puede estar formado por máquinas de propósito especial o máquinas de control numérico, con dispositivos de cambios de herramientas y piezas, con almacenes para garantizar su autonomía durante cierto periodo, y un microordenador que coordina los elementos de mecanizado, manutención y transporte entre las máquinas. Son capaces de mecanizar totalmente o casi totalmente una cierta categoría de piezas, incluyendo fases de control de calidad.

Dicho de otra forma La Celda de Manufactura Flexible, es un arreglo estratégico de estaciones de trabajo con equipos y maquinaria para manufactura. Este diseño le facilita el desarrollo e implementación del proceso de diseño, fabricación y control de una familia de productos.

Una celda puede ser separada del sistema debido a las características específicas de su proceso de manufactura (ruido, requerimientos químicos, requerimientos de materias primas o tiempo del ciclo de manufactura).

Un sistema flexible de manufactura puede contener múltiples celdas, las cuales pueden realizar diferentes y variadas funciones en cada celda o en una máquina o centro en particular.

El aspecto flexible de una celda flexible de manufactura indica que la celda no está restringida a sólo un tipo de parte o proceso, puede acomodarse fácilmente a distintas partes y productos, usualmente dentro de familias de propiedades físicas y características dimensionales similares.

La manufactura celular tiene cierto grado de control automático para las siguientes operaciones:

- Carga y descarga de materias primas y piezas en las estaciones de trabajo.
- Cambio de herramientas en las estaciones de trabajo
- Transferencia de piezas y herramientas entre las estaciones de trabajo
- Calendarización y control de la operación total en la célula



Fig. 2. 20 Otra aplicación de robots [21].

Las características importantes de la manufactura celular son la economía de tener poco trabajo en proceso y detectar en forma inmediata los problemas de calidad, aunado al menor tiempo y mano de obra de preparación, debido a que las piezas elaboradas en la celda tienen cierta similitud. Además, moviendo varios procesos a una celda, se logra que muchas órdenes de producción se consoliden en una sola orden. De esta manera se

programa mejor la producción, así como se disminuyen los movimientos de material, si se usa en conjunto los principios JIT.

Dada la gran diversidad de celdas existentes resulta prácticamente imposible estructurar una clasificación que abarque los campos de acción de todas y cada una de ellas.

Entre las células comúnmente utilizadas se encuentran las siguientes:

- Celdas flexibles de mecanizado

El mecanizado se realiza con máquinas-herramientas dotadas de control numérico o con robots que portan una herramienta en su extremidad, para la carga y descarga de las maquinas generalmente se utilizan robots manipuladores.



Fig. 2.21 Celdas de mecanizado [21].

- Celdas flexibles de ensamblado o montaje

La característica mas destacada es la utilización de robots de gran rapidez, capacidad de carga y precisión dotados de sensores (ópticos, de esfuerzos táctiles, de proximidad, etc.) con el objetivo de incrementar su independencia.

El proceso de ensamble es difícil y delicado a causa de las pequeñas tolerancias entre componentes. En este tipo de celda normalmente trabajan varios robots en paralelo, para minimizar el tiempo del proceso.



Fig. 2.22 Otra aplicación industrial [21].

- Celdas flexibles de soldadura

En estas celdas el robot puede portar el dispositivo de soldadura o bien puede alimentar varias máquinas de soldar, efectuando las tareas de seguimiento del dispositivo de soldadura, carga y descarga, transporte, etc. El robot está diseñado para trabajar en ambientes nocivos que significarían un grave riesgo para la mano de obra humana.

- Celdas flexibles de pintura

Se basa en la utilización de uno o varios robots, cuya programación, habitualmente, se realiza por aprendizaje del movimiento del brazo de un operario especializado, mientras este efectúa la operación de pintado los movimientos del operario quedan memorizados y pueden ser repetidos indefinidamente en forma automática asegurando la correcta ejecución del proceso.

En la planeación de la instalación de una celda flexible de manufactura, varias áreas deben ser tomadas en consideración:

Área de trabajo directo: selección de máquinas que funcionarán sin operador, minimización de tiempos de preparación y tiempo perdido

Área de trabajo indirecto: inspección, manejo y envíos

Área de máquinas: herramientas, enfriadores y lubricantes

Área de manejo de materiales y papeleo: movimientos de partes, programación de trabajos, tiempos perdidos



Fig. 2. 23 Los Robots en la soldadura [21].

CAPÍTULO 3

LA ROBÓTICA EN LA MANUFACTURA

3.1 Introducción

Una gran parte de la manufactura implica la manipulación de materiales, partes y herramientas. Esto se hace en las líneas de transferencia y otras por manipuladores de propósito especial, mecanismos de transferencia, transportadores y posicionadores, cada uno adaptado a un propósito particular.

En los últimos años se ha incrementado el uso de robots a la carga y descarga de las piezas de trabajo y para manipular herramientas.

Un robot industrial es una máquina programable de propósito general que posee ciertas características antropomórficas. La característica antropomórfica más evidente de un robot industrial es un brazo mecánico o manipulador, el cual está constituido físicamente de eslabones mecánicos interconectados por medio de articulaciones.

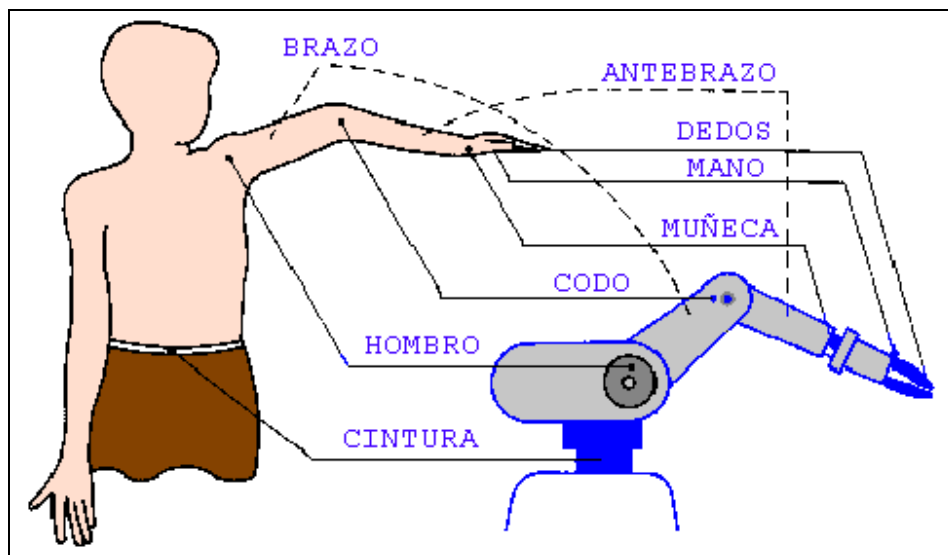


Fig. 3.1 Muestra la similitud de las extremidades de un robot y las del ser humano [22].

Los robots manipuladores son esencialmente, brazos articulados. De forma más precisa, un manipulador industrial convencional es una cadena cinemática abierta formada por un conjunto de eslabones o elementos de la cadena interrelacionados mediante articulaciones o pares cinemáticos. Las articulaciones permiten el movimiento relativo entre los sucesivos eslabones.

Los robots están dirigidos por un procesador de información que sigue las instrucciones programadas en su memoria, y están impulsados por medios neumáticos, hidráulicos, eléctricos o mecánicos, con frecuencia por una combinación de estos en una unidad.

Algunos trabajos industriales se prestan para la implantación de robots. Las características más importantes que tienden a promover la sustitución de un trabajador humano por un robot, en ciertas condiciones de trabajo, son:

- Un ambiente de trabajo peligroso para las personas.
- El ciclo de trabajo es repetitivo.
- El trabajo se realiza en una posición estacionaria.
- El manejo de la parte o la herramienta sería difícil para los humanos.
- Es una operación de cambios múltiples.
- Hay largas líneas de producción y los relevos no son frecuentes.

Las aplicaciones de los robots industriales que tienden a cumplir estas características se dividen en tres categorías básicas.

- Manejo de materiales: tiene como objetivo el movimiento de materiales o partes de una posición a otra. Para realizar la tarea de recolección, el robot está equipado con un sujetador. El sujetador debe diseñarse para asir la parte específica en la aplicación, en esta aplicación se incluye la transferencia de materiales.
- Operaciones de procesamiento: se requiere que el robot manipule una herramienta como el actuador final.
- Ensamble e inspección: frecuentemente requieren el apilamiento de una parte sobre la otra, lo cual es básicamente una tarea de manejo de partes. En otras operaciones de ensamble, se manipula una herramienta. También, las operaciones de inspección en ocasiones necesitan que el robot coloque una parte del trabajo en relación con un dispositivo de inspección o que cargue una parte

en una máquina de inspección, en otros casos implican la manipulación de un sensor para realizar una inspección.

Las nuevas tecnologías, especialmente la microelectrónica, automatización, técnicas de control, los nuevos materiales y la informática, incorporadas al proceso productivo, permiten nuevos enfoques a la resolución de los problemas de producción.

Con esto surgen nuevos conceptos como el de FMS (Flexible Manufacturing Systems). Antes de empezar a tratar de manufactura flexible debemos conocer algunos conceptos de automatización, estos conceptos son automatización fija, programable; para luego abordar el tema de manufactura flexible.

La automatización fija se caracteriza por una secuencia única de operaciones de procesamiento y ensamble. Sus operaciones son simples pero su integración en las diferentes estaciones de trabajos dan lugar a sistemas complejos y costos aplicados a la producción masiva pero cuando se cambia de un producto a otro, es necesario la puesta a punto manual de todo el equipo, implicando otras tareas, como el cambio de herramientas y utillaje.

En la automatización programable la secuencia de operaciones es controlada por un programa y puede cambiar para diferentes configuraciones del producto, la inversión en equipo es alta, y las velocidades son inferiores a las características de la producción fija y el tiempo de preparación de los equipos para cada lote es considerable (control numérico).

La fabricación flexible es la tecnología de fabricación mas adecuada para la pequeña y mediana empresa que trabaja por lotes y con una gran variedad de productos, este concepto es aplicable a cualquier tipo de proceso productivo.

La fabricación flexible se basa en la utilización intensiva de los microordenadores acoplados a las maquinas y elementos de producción en funciones de monitorización control y gestión.

Las primeras experiencias en talleres flexibles, apoyadas en un elevado número de máquinas de control numérico, manipuladores y robots coordinados todos ellos mediante un sofisticado sistema de control por ordenador, hicieron pensar que la aplicación de esta tecnología estaba reservada a las grandes empresas de elevado potencial económico y tecnológico.

Los desarrollos posteriores incorporan el concepto de modularidad, configurando el taller flexible como el conjunto de una serie de elementos, productos autónomos: las células de fabricación flexible. Estas, tanto por su costo como el nivel de conocimiento tecnológico que requieren, están al alcance de la pequeña y mediana empresa, y

permiten iniciar el camino en la implantación de una nueva tecnología de fabricación y seguirlo de acuerdo con el potencial económico y tecnológico de cada una.

El paso de un taller convencional o uno flexible, requiere de una etapa previa de estandarización tanto del producto como medios de fabricación y un conocimiento suficiente de las tecnologías en que se apoya: mecanizado en control numérico, robótica, automática e informática industrial y técnicas de control y gestión de una producción automatizada. Una célula flexible puede ser la instalación piloto que permite adquirir la experiencia suficiente en todos estos campos, para abordar con éxito su aplicación a todo el taller.

Con esto se pretende aumentar la productividad, la calidad y reducir los costos de manufactura.

3.2 Componentes de una celda de manufactura flexible

En un sistema de manufactura flexible existen cuatro componentes principales:

1. Almacenamiento y manejo de partes:

Es una instalación para guardar los materiales, son un conjunto de de plataformas de transporte o porta piezas, los utillajes para la fijación de partes una estación para carga y descarga y un sistema de transporte primario que conecta las estaciones entre si y un sistema de transporte secundario que une las estaciones de trabajo.

2. Almacenamiento y manejo de herramientas:

Es muy similar al anterior y las herramientas pueden instalarse en las máquinas de varias maneras si la maquina posee cabezales que van acopladas allí.

3. Sistemas de control por computadora:

Constituidos por computadoras, los controladores programables, los sensores y la red de comunicaciones correspondiente en las estaciones de trabajo, se encuentran diferente equipos de acuerdo con cada sistema en particular a saber: centro de mecanizado, sistema de cabezales fijos.

Desde un punto de vista funcional y de aplicación, los elementos básicos de automatización son:

- **Líneas transfer**

Las líneas transfer son los elementos clásicos de lo que se ha venido a denominar automatización pesada.

Una línea transfer puede ser de mecanizado, de montaje y consiste básicamente en un conjunto de maquinaria capaz de producir altos volúmenes con muy pocas diferencias.

Una línea transfer está formada por un sistema de transporte (rodillos, cinta) y una serie de estaciones, hornos, estaciones de lavado, estaciones de montaje, estaciones de control de calidad, etc.

El producto pasa secuencialmente por cada estación y durante su estancia se realizan, de forma automática, una serie de operaciones.

Un ejemplo típico es las líneas transfer de mecanizado y premontaje de bloques de un motor en la industria automotriz. En estas máquinas entra el bloque según viene de la fundición y sale preparado para completar el montaje del motor.

Hace algunos años las líneas transfer realizaban solo una referencia, es decir había que fabricar un lote único. El tiempo de cambio de la herramienta, utillaje, programas, etc., para cambiar a otra referencia muy similar, era del orden de semanas por lo que era necesario fabricar lotes enormes.

Actualmente, los avances en la automatización han dado origen a la existencia de líneas transfer flexibles. En estas líneas el cambio de un lote a otro es cuestión de horas, lo que permite trabajar con lotes más pequeños.

Aun así se mantiene una fuerte limitación en cuanto al número de referencias que la línea transfer es capaz de fabricar.

La gran ventaja de las líneas transfer es el reducido costo unitario resultante y los inconvenientes son, además de las altísimas inversiones sólo justificables con altos volúmenes, la falta de flexibilidad referida no sólo al número de referencias, también a la limitada capacidad de poder contar con caminos alternativos en caso de averías. Otra desventaja es el plazo de diseño y puesta en marcha, que son bastante largos.

- **Máquinas de control numérico**

El control numérico se aplica a diferentes tipos de máquinas (soldadura, empaquetado, etc.) la aplicación más conocida es en las máquinas de mecanizado y deformación, desde tornos, fresadoras, taladradoras, plegadoras, punzonadoras, máquinas de corte, electroerosión, etc.

Básicamente existen máquinas de mecanizado de cuatro tipos:

- 2 Ejes. La herramienta, o la pieza, se desplaza continuamente de acuerdo con 2 ejes, como en tornos, recantadoras, ponzonadoras, plegadoras, máquinas de corte, soldadores, etc.
- 2 ½ Ejes. Además de los dos ejes existe la posibilidad de un posicionamiento discreto en un tercer eje, como en fresadoras, plegadoras, etc.
- 3 Ejes. Desplazamiento continuo en los tres ejes. Las principales son fresadoras, centros de mecanizado y máquinas de electroerosión.
- 4 y 5 Ejes. 3 ejes cartesianos y 1 o 2 ejes de giro de la herramienta, todos de forma continua. Con estas características existen fresadoras, centros de mecanizado y máquinas de electroerosión.

La principal ventaja de las máquinas de control numérico es la flexibilidad, mejora la productividad (hasta en 300%) y precisión (mejora la calidad) que proporcionan en entornos medios de volumen y variedad.

Esto siempre que se cuente con la adecuada programación de control numérico, ya que se programa a mano, sobre el control, la flexibilidad queda bastante limitada, pero también los modelos más complejos de control numérico facilitan mucho esta actividad.

En la actualidad existen muchos sistemas adicionales a la máquina en sí, cambiadores automáticos de herramienta, mesas automáticas, sistemas de fijación automáticos, con el fin de aumentar la flexibilidad de la máquina y su tiempo real de producción ayudando a reducir los tiempos de cambio y los de inactividad de la máquina.

Las tendencias apuntan hacia la generalización de los centros de mecanizado o máquinas de multifunción; torno, taladradora, fresadora y al desarrollo de sensores avanzados para facilitar el control adaptativo o autocompensación de desviaciones por desgaste de la herramienta, modificaciones en el posicionamiento de la pieza, deformaciones debidas al peso de la pieza, etc.

Una máquina de control numérico está formada por la máquina en sí y el control numérico, siendo muy importante la elección de este último elemento, de acuerdo con los medios de programación y el entorno de la planta, como de la máquina.

- **Robots**

Definición de robot desde un punto de vista funcional

Máquina automática universal destinada a la manipulación de objetos (herramientas o piezas) dotadas de:

- Capacidad de aprendizaje de un comportamiento tipo.
- Facultad de retener el entorno (percepción).
- Facultad de analizar la información obtenida del mismo.
- Posibilidad de modificar su comportamiento tipo.

A los componentes de un robot atendiendo a la funcionalidad de cada una de sus partes, los podemos agrupar robot según cuatro funciones:

1. Mecánica. Constituida por un conjunto de módulos cuya acción combinada permite desarrollar los movimientos necesarios para la acción del robot.

Los componentes esenciales de la función mecánica son:

- Mesas de rotación
- Codos
- Unidades de translación
- Puños con tres grados de libertad

2. Potencia (eléctrica, hidráulica, etc.). Encargada de accionar los distintos dispositivos para producir movimiento. Los motores, etc., que proporcionan el movimiento de las mesas, codos, puños etc., están en este grupo.

3. Sensores. Encargados de recoger las condiciones del entorno (situación de las piezas a manejar, de los propios elementos mecánicos, etc.) y transmitírselas al equipo de control.

4. Control. Varios procesadores se encargan de recoger la información de los sensores, analizarla comparándola con las informaciones de referencia, decidir los parámetros de movimiento de los elementos mecánicos del robot y

actuar sobre los componentes de potencia para producir el movimiento y la acción de trabajos precisos.



Fig. 3.2 Robot industrial [21].

Un robot utiliza varios procesadores, con las siguientes funciones:

- Gestión de trayectorias
- Accionamiento de elementos de movimientos
- Memorización
- Reconocimiento de formas (sensores)
- Gobierno y control general

Clasificación de los robots:

1. Por su tipo de accionamiento:

- Eléctricos

- Neumáticos
- Hidráulicos

2. Por el tipo de geometría que lo define

- Cartesiano
- Cilíndrico
- Polar
- Scara
- Pórtico
- Brazo articulado

4. Sistemas de transporte automatizado

Los sistemas de manipulación de materiales unen los diversos elementos de la fábrica, su disposición determina en gran parte del sistema y de la planta.

Es muy importante tener control, en tiempo real, sobre el material: del cual se dispone, donde esta y cuanto hay.

Un sistema de manipulación automática de materiales estará formado por todos los elementos que intervienen en el movimiento y manipulación de materiales:

- Cintas transportadoras y otros elementos de transporte fijo.
- Vehículos guiados automáticamente (AGV)
- Almacenes automáticos
- Carretillas elevadoras

Las cintas transportadoras, pistas de rodillos, guías aéreas y otros medios de transporte entre puntos fijos con trayectorias fijas han sido, y son, un elemento común en las fábricas.

La automatización está aportando a estos sistemas “tradicionales” cierta flexibilidad en cuanto a la velocidad y el control de paradas.

La mayor ventaja de los sistemas de manipulación automática es su capacidad como almacenes transitorios, a la vez que desplazan el material, lo que los hace indicados como interfaces entre las estaciones de trabajo o entre los AGV y los almacenes.

Los AGV están dando un importante cambio entre los sistemas de movimiento o sistemas de manipulación de materiales, como se les llama en algunos entornos.

Los AGV se clasifican, de acuerdo con el sistema de guiado del vehículo, en:

- Filoguidados inductivos. El AGV sigue un cable enterrado en el suelo a través del cual circula una señal con una frecuencia fija para cada AGV, el AGV seguirá el tramo de cable con dicha señal.
- Ópticos. El sistema es parecido, con la diferencia de que la detección de la trayectoria se realiza ópticamente, a partir de una cinta adherida al suelo (o una línea pintada) que refleja la luz emitida por una fuente, que incorpora el vehículo, hacia un receptor a bordo.

Los AGV han de estar en contacto con un ordenador de tráfico, a través de radio o por otros métodos, para disponer de máxima flexibilidad, aunque también pueden utilizarse sólo con la electrónica que tienen a bordo, programándolos mediante teclado o código de barras.

La aplicación más importante de los AGV es para el transporte entre secciones o entre plantas. También empiezan a utilizarse como plataformas móviles de montaje.

Una FMC básica esta formada por:

- Una máquina de CN
- Un robot o manipulador destinado a realizar la carga y descarga de piezas en la máquina, así como el intercambio de herramientas y pallets.
- Unas zonas fijas de almacenamiento de entrada y salida para las piezas, herramientas y pallets.
- Un controlador de célula capaz de dirigir la producción, unido al módulo de control.

A partir de esta maquinaria se puede crecer, lo que se quiera según las necesidades como pueden ser:

- El número y tipo de máquinas CN en función de la familia de piezas a fabricar.
- Un subsistema de sensores o entrada de datos interno.
- El número de manipuladores o robots.

Un sistema de transporte interno de piezas, herramientas y pallets.

- La complejidad el sistema de entrada-salida de piezas, herramientas y pallets.
- La potencia y capacidad del controlador de célula y la complejidad del software que utilice.
- La funcionalidad de la red industrial.

Las FMC encuentran su principal aplicación en aquellos procesos de producción con siguientes características:

- Los lotes de producción son pequeños.
- Los productos pertenecen a una misma familia.
- Los tiempos de operación para cada pieza son pequeños, con lo que mejora el rendimiento de las máquinas y el ritmo de producción.

El elemento de control en una FMC puede ser un PLC u ordenador. El PLC es el elemento de control mas usado en las FMC de ensamblado, recibiendo del ordenador o controlador de área, el programa de ensamblaje para los manipuladores o maquinaria especial, de acuerdo con el lote a procesar.

En el caso de las FMC con CN o robots, el controlador de célula puede ser una PC, con capacidad para transmitir a las máquinas y robots los programas necesarios y de recibir las señales provenientes de dicha maquinaria.

Es común que se disponga de un arreglo de varias FMC formando un FMS, en cuyo caso las FMC suelen contar con elementos individuales de control tipo PLC o PC.

Sistemas flexibles de fabricación

Un FMS es básicamente un conjunto de FMC unidas mediante un sistema de manipulación automatizada de materiales, que permite procesar en forma simultánea volúmenes medios de piezas o productos pertenecientes a una misma familia.

Una familia de piezas, es un grupo de piezas cuyas especificaciones de proceso, dimensiones, operaciones, tiempos, etc., están dentro de un cierto margen.

Los FMS son la respuesta a las tendencias, de la producción actual, como son:

- Incremento en la diversidad del producto
- Reducción en el ciclo de vida
- Aumento en la competencia

Y a los problemas relacionados, a este nuevo escenario, de los sistemas tradicionales de los problemas de proceso por lotes:

- Lotes de gran tamaño
- Formación de colas en ciertas áreas
- Infrautilización de recursos en ciertas áreas
- Pocas flexibilidad frente a cambios y especificaciones

Cuando se trabaja en un entorno funcional, generalmente menos del 10% del tiempo que un lote de piezas permanece en el taller corresponde a tiempo de proceso, siendo el resto del tiempo de espera, es decir, tiempo sin valor agregado.

Las principales ventajas de los FMS vienen precisamente por el lado de la reducción del tiempo de espera, lo que representa una reducción de la obra en curso, y, por lo tanto, un beneficio económico.

Adicionalmente, el FMS permite un gran margen de flexibilidad, lo que resulta fundamental para poder corresponder adecuadamente a las cambiantes demandas del mercado.

El ordenador del FMS, normalmente una minicomputadora, que deberá disponer de aplicaciones con la siguiente funcionalidad:

- Programación a corto plazo

- Sincronización de células
- Monitorización
- Seguimiento de piezas
- Seguimiento de ordenes de trabajo
- Gestión de colas de las FMC
- Gestión de manipulación automatizada de materiales
- Descarga de programas e instrucciones a los controladores de la célula o envío directo de datos alarma a las máquinas de la célula.
- Librerías de programa
- Monitorización del FMS
- Monitorización de cada FMC
- Recogida y consolidación de datos
- Gestión de alarmas
- Control estadístico de calidad
- Mantenimiento correctivo

3.3 Operación de los robots en procesos de manufactura

Para que un robot sea útil en una aplicación particular debe equiparse y habilitarse con herramientas diseñadas para una aplicación específica. Un actuador final es la habilitación especial de herramientas que se conecta al extremo de la muñeca del robot para realizar una tarea específica. Existen dos tipos de actuadores finales: herramientas y sujetadores (grippers).

La herramienta se utiliza cuando el robot debe realizar una operación de procesamiento. Las herramientas especiales incluyen pistolas para soldadura de puntos, herramientas para soldadura con arco, boquillas para pintar por aspersión, sopletes para calentamiento y herramientas de ensamble. El robot se programa para manipular la herramienta que se necesita para la parte del trabajo que se procesa.

Los sujetadores están diseñados para asir y mover los objetos durante el ciclo de trabajo. En general, los objetos son parte del trabajo y el actuador final debe diseñarse específicamente para sujetar alguna parte. Los sujetadores se usan para aplicaciones de colocación de partes, carga y descarga de maquinas.

3.4 Perspectivas de la robótica en la manufactura

Con la incorporación de la informática gráfica en las oficinas de ingeniería, tanto en producto como en el proceso, y con la creciente informatización de los equipos de producción dentro de la técnica de fabricación flexible, el ordenador esta presente y operativo en todas y cada una de las funciones empresariales.

Una primera evolución lógica del sistema se orienta hacia la integración de toda la información, recopilada en las distintas áreas de la empresa, una sola base de datos accesible a todos los ordenadores dispuestos entre sí mediante una red local de comunicaciones.

Este concepto es el CIM (Computer Integrated Manufacturing) sobre una red de computadoras y una única base de datos se soporta todo el control y el trabajo empresarial. Producción, comercial, finanzas, vuelcan toda su información al sistema informativo y recuperan del mismo todo dato necesario para desarrollar sus funciones.

La integración de los procesos y operaciones de manufactura y su administración se consideran como un sistema, las funciones generalmente separadas de investigación, desarrollo, diseño, producción, comercialización y finanzas, se relacionan entre sí.

Esto exige un conocimiento amplio de la relación existente de todas las actividades.

Se espera una fábrica sin papeles. El proyectista, en vez de un plano, crea un modelo geométrico de la pieza en la base de datos de su sistema grafico. El ingenio de proceso, sobre este modelo, prepara todos los elementos de producción que sean necesarios, utillajes, herramientas, ciclos, listas de materiales, programas de maquinas de CN, robots, manipuladores, que quedan archivados en la misma base de datos. Los técnicos de planificación introducen sus programas de producción y determinan el plan de acopios y de fabricación fijando cantidades y plazos. Los técnicos de control de

producción, mediante sistemas de JIT, kan-ban, siguen y gobiernan día a día, hora a hora. La evolución del flujo de materiales desde su recepción, mecanización y montaje a su expedición. Todo el movimiento de piezas y herramientas, utilización de maquinas consumo de materiales, mano de obra, incidencia etc.; queda registrado en la base de datos del sistema.

Se pretende que esta base de datos de fabricación sea el núcleo sobre el que el resto de los departamentos financiero, comercial, gestión empresarial, basen sus aplicaciones y al que deben complementar con sus propios datos.

Los sistemas informáticos son cada día mas “inteligentes”. La incorporación de sistemas expertos y de inteligencia artificial, aunado con las posibilidades de interacción con el entorno mediante evolucionados sistemas sensoriales de vista, oído, y tacto, permiten imaginar no solo una fabricación sin atención humana, si no también todo el proceso productivo, desde el diseño del producto, planificación del proceso, planificación de la producción, producción y la entrega del producto final al cliente que sea totalmente automatizado solo con intervención humana en el mantenimiento y supervisión.

El concepto de FA (Full Automation o Fabrica Automática). Si bien en la actualidad no existen Fabricas Automáticas, ya se comercializan sistemas que automatizan funciones complejas que hasta la fecha estaban a cargo de profesionales expertos.

CAPÍTULO 4

LOS ROBOTS INDUSTRIALES

4.1 Introducción

La maquinaria para la automatización rígida dio paso al robot con el desarrollo de controladores rápidos, basados en el microprocesador, así como un empleo de servos en bucle cerrado, que permiten establecer con exactitud la posición real de los elementos del robot y establecer el error con la posición deseada. Esta evolución ha dado origen a una serie de tipos de robots, que se citan a continuación:

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

Manual: cuando el operario controla directamente la tarea del manipulador.

De secuencia fija: cuando se repite, de forma invariable, el proceso de trabajo preparado previamente.

De secuencia variable: se pueden alterar algunas características de los ciclos de trabajo.

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

- **Robots de repetición o aprendizaje:** son manipuladores que se limitan a repetir una secuencia de movimientos, previamente ejecutada por un operador humano, haciendo uso de un controlador manual o un dispositivo auxiliar. En este tipo de robots, el operario en la fase de enseñanza, se vale de una pistola de programación con diversos pulsadores o teclas, o bien, de joystics, o bien utiliza un maniquí, o a veces, desplaza directamente la mano del robot. Los robots de aprendizaje son los más conocidos, en la actualidad, en los ambientes industriales y el tipo de programación que incorporan, recibe el nombre de "gestual".
- **Robots con control por computador:** son manipuladores o sistemas mecánicos multifuncionales, controlados por una computadora, que habitualmente suele ser

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

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

- **Robots inteligentes:** son similares a los del grupo anterior, pero, además, son capaces de relacionarse con el mundo que les rodea a través de sensores y tomar decisiones en tiempo real (auto programable).

De momento, son muy poco conocidos en el mercado y se encuentran en fase experimental, en la que se esfuerzan los grupos investigadores por potenciarles y hacerles más efectivos, al mismo tiempo que más accesibles.

La visión artificial, el sonido de máquina y la inteligencia artificial, son las ciencias que más se están estudiando para su aplicación en los robots inteligentes.

- **Micro-robots:** con fines educativos, de entretenimiento o investigación, existen numerosos robots de formación o micro-robots a un precio muy accesible y, cuya estructura y funcionamiento son similares a los de aplicación industrial.

4.2 Los subsistemas de un robot industrial

Los robots están constituidos de subsistemas, es decir, constan de componentes que forman un todo. Dichos subsistemas realizan una tarea específica y tiene su propia entrada y salida. Aunque los elementos empleados en los robots no son exclusivos de estos Las altas prestaciones que se exigen a los robots han motivado que en ellos se empleen elementos con características específicas. Los robots tienen los siguientes subsistemas principales:

Sub-sistema mecánico: consiste en un manipulador mecánico que actúa sobre los objetos de su entorno a fin de realizar una tarea útil para el usuario. Ejemplo pintar un objeto, soldar dos piezas, trasladar un objeto de un punto a otro, etc.

Sub-sistema de comunicación: constituido por una computadora, que permite al usuario comunicarle al manipulador la tarea a realizar, y conocer la situación de este y de su entorno durante la ejecución de la misma.

Sub-sistema de decisión: es un sistema intermedio, entre la parte de comunicación y la de control, que consiste generalmente en un programa de computadora, cuya función básica es la de interpretar las instrucciones que emite el usuario para definir una tarea, y convertir estas en consignas de movimiento de cada eslabón del manipulador.

Sub-sistema de control: procesa la información recibida de los sub-sistemas de decisión (la consigna original) y de percepción (errores a corregir) y genera las señales físicas necesarias para que, después de ser amplificadas, los actuadores del manipulador ejecuten finalmente los movimientos requeridos por sus eslabones en la realización de la tarea.

Sub-sistema de percepción: conjunto de instrumentos captadores, que a través de las señales físicas que generan, permiten a los sub-sistemas de control y decisión conocer la situación instantánea del manipulador y de los objetos del ambiente durante la ejecución de una tarea.

4.3 Elementos que forman un robot

- **Manipulador:** mecánicamente, es el componente principal. Está formado por una serie de elementos estructurales sólidos o eslabones unidos mediante articulaciones que permiten un movimiento relativo entre dos eslabones consecutivos.

Las partes que conforman el manipulador reciben los nombres de: cuerpo, brazo, muñeca y actuados final (o elemento terminal). A este último se le conoce habitualmente como aprehensor, garra, pinza o *gripper*.

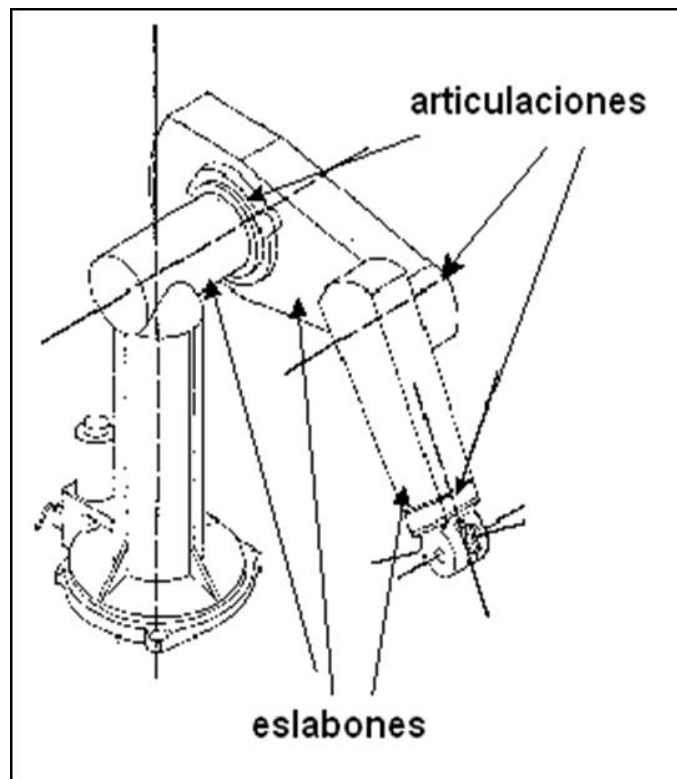


Fig. 4.1 Partes que componen un manipulador [22] .

Cada articulación provee al robot de por lo menos, un grado de libertad. En otras palabras, las articulaciones permiten al manipulador realizar diferentes movimientos:

- **Lineales:** que pueden ser horizontales o verticales.
- **Angulares** (por articulación)

Los tipos de articulación utilizados en las juntas del manipulador son:

- **Prismática /Lineal** - junta en la que el eslabón se apoya en un deslizador lineal. Actúa linealmente mediante los tornillos sin fin de los motores, o los cilindros.
- **Rotacional** - junta giratoria a menudo manejada por los motores eléctricos y las transmisiones, o por los cilindros hidráulicos y palancas.

Básicamente, la orientación de un eslabón del manipulador se determina mediante los elementos *roll*, *pitch* y *yaw*.

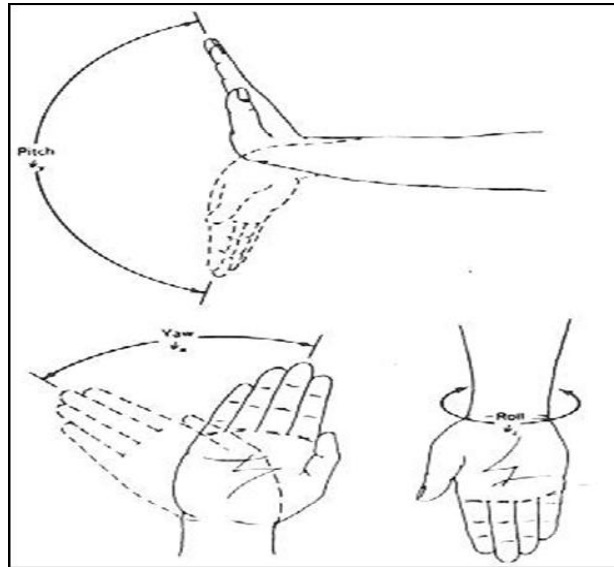


Fig. 4.2. Ilustración del *pitch*, *yaw* y *roll* [22]

A la **muñeca** de un manipulador le corresponden los siguientes movimientos o grados de libertad: **giro** (*hand rotate*), **elevación** (*wrist flex*) y **desviación** (*wrist rotate*) como lo muestra en la fig. 4. 3, aunque cabe hacer notar que existen muñecas que no pueden realizar los tres tipos de movimiento.

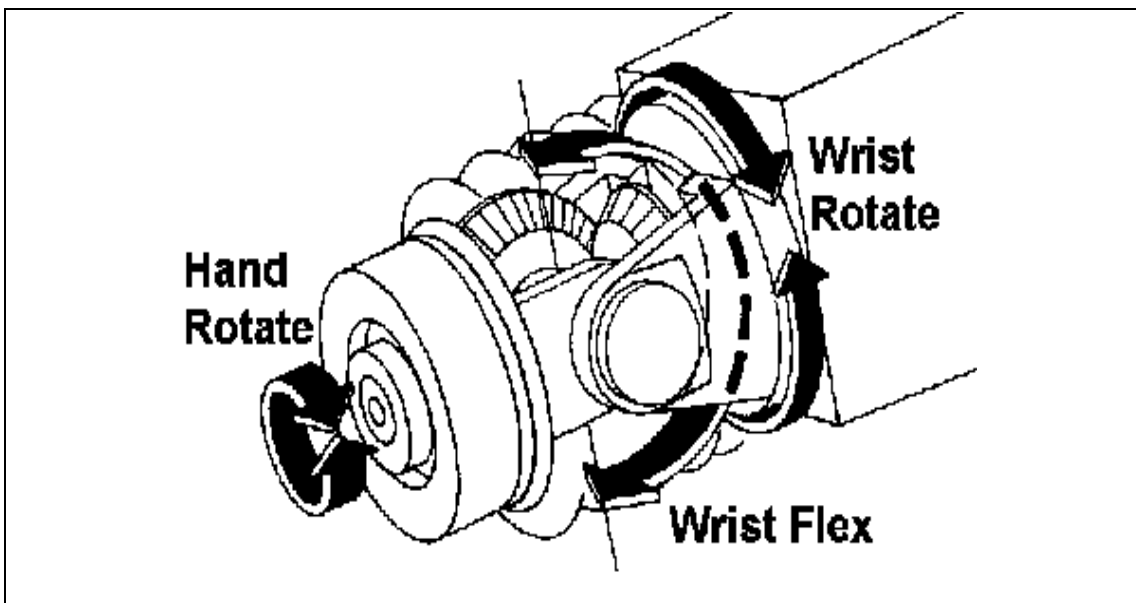


Fig. 4.3 Muestra los movimientos de la muñeca del robot [22].

- **El actuador final (*gripper*):** es un dispositivo que se une a la muñeca del brazo del robot con la finalidad de activarlo para la realización de una tarea específica. La razón por la que existen distintos tipos de elementos terminales es, precisamente, por las funciones que realizan. Para las aplicaciones industriales, las capacidades de un robot básico deben aumentarse por medio de dispositivos adicionales. Podríamos denominar a estos dispositivos como los periféricos del robot. En robótica, el término de actuador final se utiliza para describir la mano o herramienta que está unida a la muñeca. El actuador final representa la herramienta especial que permite al robot de uso general realizar una aplicación particular, y debe diseñarse específicamente para dicha aplicación.

Los actuadores finales pueden dividirse en dos categorías:

- **Pinzas (*gripper*):** las pinzas se utilizan para tomar un objeto, normalmente la pieza de trabajo, y sujetarlo durante el ciclo de trabajo del robot. Hay una diversidad de métodos de sujeción que pueden utilizarse, además de los métodos mecánicos obvios de agarre de la pieza entre dos o más dedos. Estos métodos suplementarios incluyen el empleo de casquetes de sujeción, imanes, ganchos, y cucharas. Se distingue entre las que utilizan dispositivos de agarre mecánico y las que utilizan algún otro tipo de dispositivo (ventosas, pinzas magnéticas, adhesivas, ganchos, etc.). Se pueden clasificar según el sistema de sujeción empleado.
- **Herramientas:** una herramienta se utiliza como actuador final en aplicaciones en donde se exija al robot realizar alguna operación sobre la pieza de trabajo. Estas aplicaciones incluyen la soldadura por puntos, la soldadura por arco, la pintura por pulverización y las operaciones de taladro. En cada caso, la herramienta particular está unida a la muñeca del robot para realizar la operación.



Fig. 4.4 Muestra una herramienta [17].



Fig. 4.5 Muestra otro tipo de herramienta [17].

El accionamiento neumático es el más utilizado por ofrecer mayores ventajas en simplicidad, precio y fiabilidad, aunque presenta dificultades de control de posiciones intermedias. En ocasiones se utilizan accionamiento de tipo eléctrico. En la pinza se suelen situar sensores para detectar el estado de la misma (abierto o cerrado). Se pueden incorporar a la pinza otro tipo de sensores para controlar el estado de la pieza, sistemas de visión que incorporen datos geométricos de los objetos, detectores de proximidad, sensores fuerza par.

Existen ciertos elementos comerciales que sirven de base para la pinza, siendo posible a partir de ellos diseñar actuadores válidos para cada aplicación concreta. Sin embargo, en otras ocasiones el actuador debe ser desarrollado íntegramente, constituyendo un porcentaje importante dentro del costo total de la aplicación.



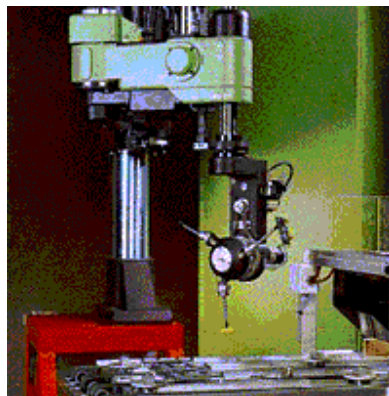
Fig. 4.6 Se muestra otro tipo de herramienta [17].



Fig. 4.7 muestra otra herramienta [17].



Fig. 4.7 muestra otra herramienta su muñeca para aplicaciones de soldadura [17].



4.9 En este robot el actuador final consiste de una serie de sensores [17]

Como se adelantó en los subsistemas de un robot, un robot está formado por los siguientes elementos: estructura mecánica, transmisiones, actuadores, sensores, elementos terminales y controlador.

- **Controlador:** como su nombre indica, es el que regula cada uno de los movimientos del manipulador, las acciones, cálculos y procesamiento de la información. El controlador recibe y envía señales a otras máquinas-herramientas (por medio de señales de entrada/salida) y almacena programas.

Existen varios grados de control que son función del tipo de parámetros que se regulan, lo que da lugar a los siguientes tipos de controladores:

- De 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 a él.
- Adaptativo: engloba todas las regulaciones anteriores y, además, se ocupa de controlar la variación de las características del manipulador al variar la posición.

Otra clasificación del control es la que distingue entre control en bucle abierto y control en bucle cerrado.

El control en bucle abierto da lugar a muchos errores, y aunque es más simple y económico que el control en bucle cerrado, no se admite en aplicaciones industriales en las que la exactitud es una cualidad imprescindible. La inmensa mayoría de los robots que hoy día se utilizan con fines industriales se controlan mediante un proceso en bucle cerrado, es decir, mediante un bucle de **realimentación**. Este control se lleva a cabo con el uso de un sensor de la posición real del elemento terminal del manipulador. La información recibida desde el sensor se compara con el valor inicial deseado y se actúa en función del error obtenido de forma tal que la posición real del brazo coincida con la que se había establecido inicialmente.

- **Dispositivos de entrada y salida:** los más comunes son: teclado, monitor y caja de comandos (*teach pendant*).

Los dispositivos de entrada y salida permiten introducir y, a su vez, ver los datos del controlador. Para mandar instrucciones al controlador y para dar de alta programas de control, comúnmente se utiliza una computadora adicional. Es necesario aclarar que algunos robots únicamente poseen uno de estos componentes. En estos casos, uno de los componentes de entrada y salida permite la realización de todas las funciones.

Las señales de entrada y salida se obtienen mediante tarjetas electrónicas instaladas en el controlador del robot las cuales le permiten tener comunicación con otras máquinas-herramientas

Se pueden utilizar estas tarjetas para comunicar al robot, ejemplo, con las máquinas de control numérico. Estas tarjetas se componen de relevadores, los cuales mandan señales eléctricas que después son interpretadas en un programa de control. Estas señales nos permiten controlar cuándo debe entrar el robot a cargar una pieza a la máquina, cuando debe empezar a funcionar la máquina o el robot, etc.

- **Dispositivos especiales:** entre estos se encuentran los ejes que facilitan el movimiento transversal del manipulador y las estaciones de ensamble, que son utilizadas para sujetar las distintas piezas de trabajo.

El robot cuenta con señales de entrada/salida para poder realizar la integración de su función incorporando estos elementos.

4.4 Principales características de los robots

A continuación se describen las características más relevantes propias de los robots:

- Grados de libertad (GDL): cada uno de los movimientos independientes (giros y desplazamientos) que puede realizar cada articulación con respecto a la anterior. Son los parámetros que se precisan para determinar la posición y la orientación del elemento terminal del manipulador.

La regla de K. Kutzbach permite determinar el número de grados de libertad de un robot con mecanismos especiales en términos del número de eslabones y de articulaciones.

$$G = 6 * (e - 1) - 5 * a \quad (\text{ec. 4.1})$$

donde:

G=Grados de libertad

e=Número de eslabones

a=Número de articulaciones

Puesto que para posicionar y orientar un cuerpo de cualquier manera en el espacio son necesarios seis parámetros, tres para definir la posición y tres para la orientación, si se pretende que un robot posicione y oriente su extremo (y con él la pieza o herramienta

manipulada) de cualquier modo en el espacio, se precisará al menos seis grados de libertad.

Un mayor número de grados de libertad conlleva un aumento de la flexibilidad en el posicionamiento del elemento terminal. Aunque la mayoría de las aplicaciones industriales requieren 6 GDL, como en las de soldadura, mecanizado y paletización, otras más complejas requieren un número mayor, tal es el caso en las labores de montaje. Si se trabaja en un entorno con obstáculos, el dotar al robot de grados de libertad adicionales le permitirá acceder a posiciones y orientaciones de su extremo a las que, como consecuencia de los obstáculos, no hubieran llegado con seis grados de libertad. Otra situación frecuente es dotar al robot de un grado de libertad adicional que le permita desplazarse a lo largo de un carril aumentando así el volumen del espacio al que puede acceder. Tareas más sencillas y con movimientos más limitados, como en las de pintura y paletización, suelen exigir 4 o 5 GDL.

Cuando el número de grados de libertad del robot es mayor que los necesarios para realizar una determinada tarea se dicen que el **robot es redundante**.

Observando los movimientos del brazo y de la muñeca, podemos determinar el número de grados de libertad que presenta un robot. Generalmente, tanto en el brazo como en la muñeca, se encuentra un abanico que va desde uno hasta los tres GDL. Los grados de libertad del brazo de un manipulador están directamente relacionados con su anatomía o configuración.

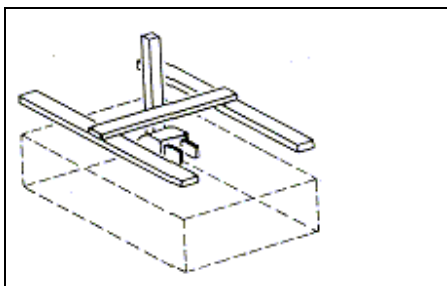
- **Espacio (volumen) de trabajo:** las dimensiones de los elementos del manipulador, junto a los grados de libertad, definen la zona de trabajo del robot, característica fundamental en las fases de selección e implantación del modelo adecuado.

La zona de trabajo se subdivide en áreas diferentes entre sí, por la accesibilidad específica del elemento terminal (aprehensor o herramienta), es diferente a la que permite orientarlo verticalmente o con el determinado ángulo de inclinación.

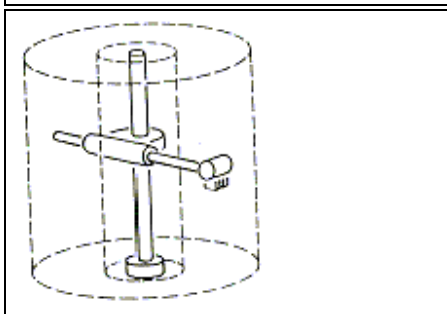
También queda restringida la zona de trabajo por los límites de giro y desplazamiento que existen en las articulaciones.

El volumen de trabajo de un robot se refiere únicamente al espacio dentro del cual puede desplazarse el extremo de su muñeca. Para determinar el volumen de trabajo no se toma en cuenta el actuador final. La razón de ello es que a la muñeca del robot se le pueden adaptar *grippers* de distintos tamaños.

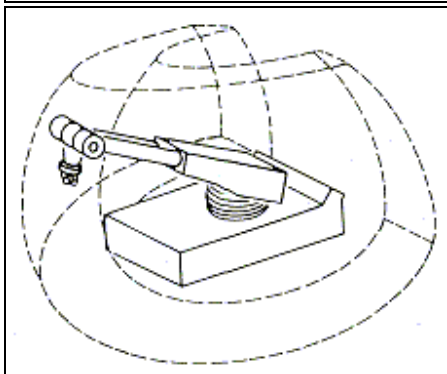
Para ilustrar lo que se conoce como volumen de trabajo regular y volumen de trabajo irregular, tomaremos como modelos varios robots.



El robot cartesiano y el robot cilíndrico presentan volúmenes de trabajo regulares. El robot cartesiano genera una figura cúbica.



El robot de configuración cilíndrica presenta un volumen de trabajo parecido a un cilindro (normalmente este robot no tiene una rotación de 360°)



Por su parte, los robots que poseen una configuración polar, los de brazo articulado y los modelos SCARA presentan un volumen de trabajo irregular.

Fig. 4.10 muestra el espacio de trabajo de de diferentes configuraciones [22].

Para determinar el volumen de trabajo de un robot industrial, el fabricante generalmente indica un plano con los límites de movimiento que tiene cada una de las articulaciones del robot.

- **Precisión de los movimientos**

La precisión del movimiento en un robot industrial depende de tres factores:

La resolución espacial: se define como el incremento más pequeño de movimiento en que el robot puede dividir su volumen de trabajo.

La resolución espacial depende de dos factores:

Los **sistemas que controlan la resolución:** es el medio para controlar todos los incrementos individuales de una articulación. Los controladores dividen el intervalo total de movimiento para una junta particular en incrementos individuales (resolución de control o de mando). La habilidad de dividir el rango de la junta en incrementos depende de la capacidad de almacenamiento en la memoria de mando.

- Las **inexactitudes mecánicas** se encuentran estrechamente relacionadas con la calidad de los componentes que forman las uniones y las articulaciones. Ejemplos de inexactitudes mecánicas pueden citarse la holgura de los engranajes, las tensiones en las poleas, las fugas de fluidos, etc.
- La **exactitud** se refiere a la capacidad de un robot para situar el extremo de su muñeca en un punto señalado dentro del volumen de trabajo. Mide la distancia entre la posición especificada, y la posición real del órgano Terminal.
- La **repetibilidad** se refiere a la capacidad del robot de regresar al punto programado las veces que sean necesarias. Esta magnitud establece el grado de exactitud en la repetición de los movimientos de un manipulador al realizar una tarea programada.
- **Capacidad de carga:** es el peso, que puede transportar la garra del manipulador. Este dato lo proporcionan los fabricantes, incluyendo el peso de la propia garra, es una de las características que más se tienen en cuenta en la selección de un robot, dependiendo la tarea a la que se destine.
- **Velocidad:** se refiere a la velocidad máxima alcanzable por las articulaciones. En muchas ocasiones, una velocidad de trabajo elevada, aumenta extraordinariamente el rendimiento del robot, por lo que esta magnitud se valora considerablemente en la elección del mismo.

En tareas de soldadura y manipulación de piezas es muy aconsejable que la velocidad de trabajo sea alta. En pintura, mecanizado y ensamblaje, la velocidad debe ser media e incluso baja.

- **Actuadores:** son los elementos que generan el movimiento de las articulaciones pueden ser, según la energía que consuman, de tipo oleohidráulico, neumático o eléctrico.

Los actuadores de tipo oleohidráulico se destinan a tareas que requieren una gran potencia y grandes capacidades de carga. Dado el tipo de energía que emplean, se construyen con mecánica de precisión y su coste es elevado. Los robots hidráulicos se diseñan formando

un conjunto compacto que consta de la central hidráulica, la cabina electrónica de control y el brazo del manipulador.

La energía neumática dota a sus actuadores de una gran velocidad de respuesta junto a un bajo costo, pero su empleo está siendo sustituido por elementos eléctricos.

Los motores eléctricos, que cubren la gama de media y baja potencia, acaparan el campo de la Robótica, por su gran precisión en el control de su movimiento y las ventajas inherentes a la energía eléctrica que consumen.

La función del **controlador** es gobernar el trabajo de los **actuadores** (los dispositivos que originan el movimiento) y las **transmisiones** (modificadores del movimiento). La **alimentación** proporciona la energía necesaria para todo el sistema. Además de estos tres subsistemas, los robots de segunda generación incorporan **sensores** que reciben la señal de realimentación procedente de los actuadores pasando la información al controlador, que debe calcular la corrección del error. El entorno proporciona también información a los sensores y se envía de nuevo al controlador para hacer los ajustes necesarios para la realización de la tarea.

- **Programabilidad:** la inclusión del controlador de tipo microelectrónica en los robots industriales, permite la programación del robot de muy diversas formas.

En general, los modernos sistemas de robots admiten la programación manual, mediante un modulo de programación.

Las programaciones gestual y textual, controlan diversos aspectos del funcionamiento del manipulador:

- Control de la velocidad y la aceleración.
- Saltos de programa condicionales.
- Temporizaciones y pausas.
- Edición, modificación, depuración y ampliación de programas.
- Funciones de seguridad.

- Funciones de sincronización con otras máquinas.
- Uso de lenguajes específicos de Robótica.

4.5 Configuraciones morfológicas de los robots

La estructura del manipulador y la relación entre sus elementos proporcionan una configuración mecánica, que da origen al establecimiento de los parámetros que hay que conocer para definir la posición y orientación del elemento terminal. Fundamentalmente, existen cuatro estructuras clásicas en los manipuladores, que se relacionan con los correspondientes modelos de coordenadas en el espacio. Debido a lo anterior el brazo del manipulador puede presentar cuatro configuraciones clásicas:

- Cartesiana
- Cilíndrica
- Esférica
- De brazo articulado

Y una no clásica:

- SCARA (*Selective Compliance Assembly Robot Arm*).

El empleo de diferentes combinaciones de articulaciones en un robot, da lugar a diferentes configuraciones, con características a tener en cuenta tanto en el diseño y construcción del robot como en su aplicación. Las combinaciones más frecuentes son con tres articulaciones, que son las más importantes a la hora de posicionar su extremo en un punto en el espacio. A continuación se presentan las características principales de las configuraciones del brazo manipulador.

- **Cartesiana / Rectilínea:** el posicionando se hace en el espacio de trabajo con las articulaciones prismáticas. Esta configuración se usa bien cuando un espacio de trabajo es grande y debe cubrirse, o cuando la exactitud consiste en la espera del robot. Posee tres movimientos lineales, es decir, tiene tres grados de libertad, los cuales corresponden a los movimientos localizados en los ejes X, Y y Z.

Los movimientos que realiza este robot entre un punto y otro son con base en interpolaciones lineales. Interpolación, en este caso, significa el tipo de trayectoria que realiza el manipulador cuando se desplaza entre un punto y otro. A la trayectoria realizada en línea recta se le conoce como interpolación lineal y a la trayectoria hecha de acuerdo con el tipo de movimientos que tienen sus articulaciones se le llama interpolación por articulación.

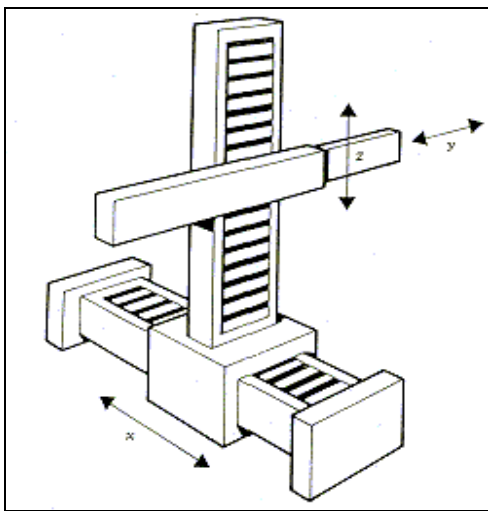


Fig. 4.11 muestra una configuración cartesiana [19].

- **Cilíndrica:** el robot tiene un movimiento de rotación sobre una base, una articulación prismática para la altura, y una prismática para el radio. Este robot se ajusta bien a los espacios de trabajo redondos. Puede realizar dos movimientos lineales y uno rotacional, o sea, que presenta tres grados de libertad

Este robot está diseñado para ejecutar los movimientos conocidos como interpolación lineal e interpolación por articulación. La interpolación por articulación se lleva a cabo por medio de la primera articulación, ya que ésta puede realizar un movimiento rotacional.

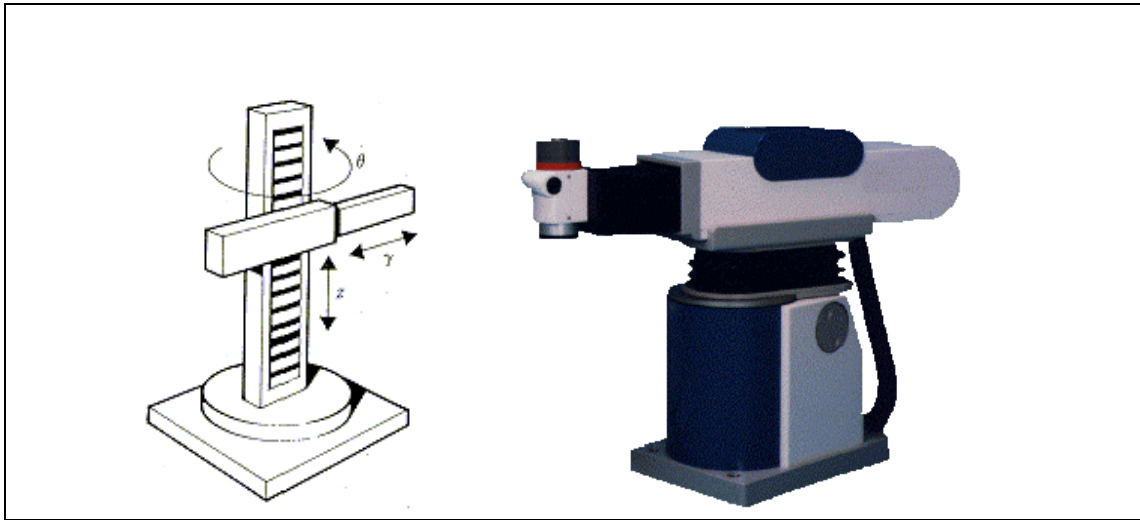


Fig. 4.12 Muestra una configuración cilíndrica [19].

- **Esférica / Polar:** dos juntas de rotación y una prismática permiten al robot apuntar en muchas direcciones, y extender la mano a un poco de distancia radial. Los movimientos son: rotacional, angular y lineal. Este robot utiliza la interpolación por articulación para moverse en sus dos primeras articulaciones y la interpolación lineal para la extensión y retracción.

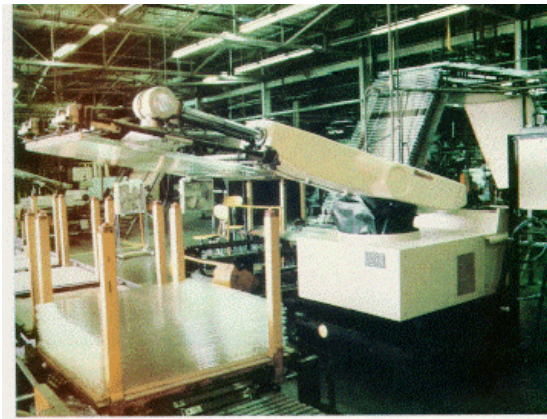
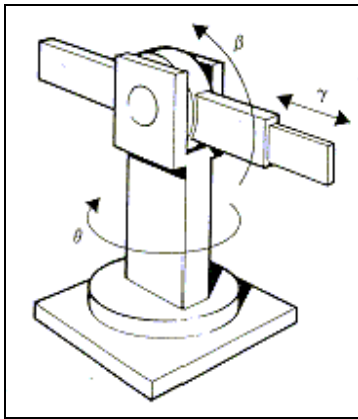


Fig. 4.13 Muestra una configuración esférica [19].

- **De Brazo articulado / Articulación esférica / Articulación coordinada / Rotación / Angular:** el robot usa 3 juntas de rotación para posicionarse. Generalmente, el volumen de trabajo es esférico. Estos tipos de robot se parecen al brazo humano, con una cintura, el hombro, el codo, la muñeca. Presenta una articulación con movimiento rotacional y dos angulares. Aunque el brazo

articulado puede realizar el movimiento llamado interpolación lineal (para lo cual requiere mover simultáneamente dos o tres de sus articulaciones), el movimiento natural es el de interpolación por articulación, tanto rotacional como angular.

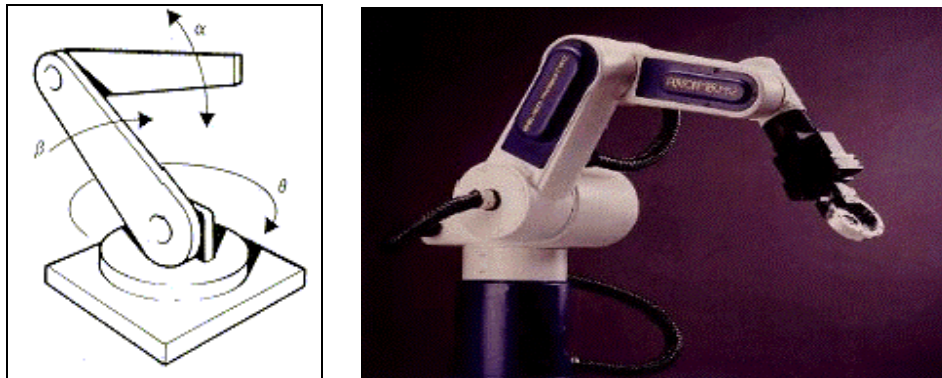


Fig. 4.14 Muestra una articulación brazo articulado [19].

- **SCARA:** similar al de configuración cilíndrica, pero el radio y la rotación se obtiene por uno o dos eslabones. Este brazo puede realizar movimientos horizontales de mayor alcance debido a sus dos articulaciones rotacionales. El robot de configuración **SCARA** también puede hacer un movimiento lineal (mediante su tercera articulación).

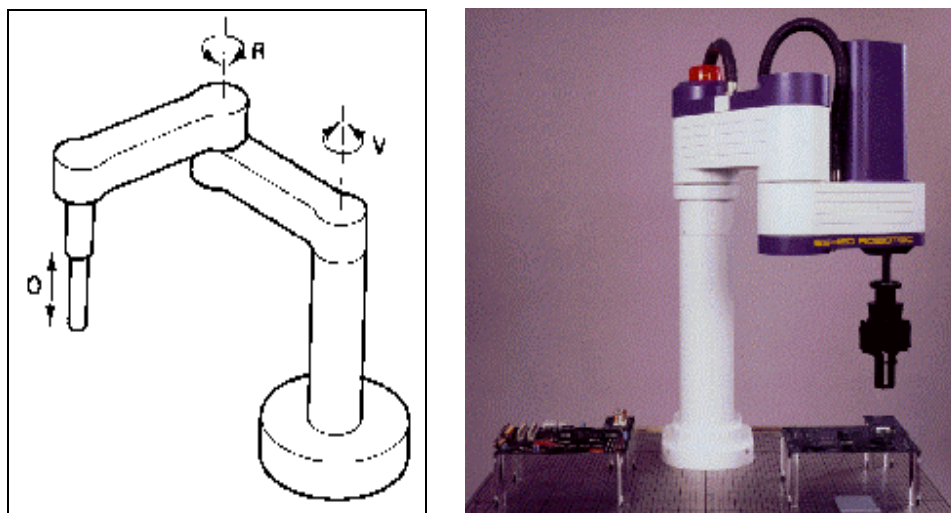


Fig. 4.15 Muestra una configuración SCARA [19].

4. 6 Transmisiones y reductores en los robots

Las **transmisiones** son los elementos encargados de transmitir el movimiento desde los actuadores hasta las articulaciones. Se incluirán junto con las transmisiones a los **reductores**, encargados de adaptar el par y la velocidad de la salida del actuador a los valores adecuados para el movimiento de los elementos del robot.

- **Transmisiones:** dado que un robot mueve su extremo con aceleraciones elevadas, es de gran importancia reducir al máximo su momento de inercia. Del mismo modo, los pares estáticos que deben vencer los actuadores dependen directamente de la distancia de las masas al actuador. Por estos motivos se procura que los actuadores, por lo general pesados, estén lo mas cerca posible de la base del robot. Esta circunstancia obliga a utilizar sistemas de transmisión que trasladen el movimiento hasta las articulaciones, especialmente a las situadas en el extremo del robot. Asimismo, las transmisiones pueden ser utilizadas para convertir movimiento circular en lineal o viceversa, cuando sea necesario.

Existen actualmente en el mercado robots industriales con acoplamiento directo entre accionamiento y articulación. Se trata, sin embargo, de casos particulares dentro de la generalidad que en los robots industriales actuales supone la existencia de sistemas de transmisión junto con reductores para el acoplamiento entre actuadores y articulaciones.

Es de esperar que un buen sistema de transmisión cumpla con una serie de características básicas:

- Debe tener un tamaño y peso reducido;
- Se ha de evitar que presente juegos u holguras considerables;
- Se deben buscar transmisiones con gran rendimiento.

Las transmisiones más habituales son aquellas que cuentan con movimiento circular tanto a la entrada como a la salida. Incluidas en éstas se encuentran los engranajes, las correas dentadas y las cadenas.

Reductores: en cuanto a los reductores, al contrario que con las transmisiones, sí existen determinados sistemas usados de manera preferente en los robots industriales. Esto se debe a que a los reductores utilizados en robótica se les exigen unas condiciones de funcionamiento muy restrictivas. La exigencia de estas características viene motivada por las altas exigencias que se le piden al robot en cuanto a precisión y velocidad de posicionamiento.

Se buscan reductores de bajo peso, reducido tamaño, bajo rozamiento y que al mismo tiempo sean capaces de realizar una reducción elevada de velocidad en un único paso. Se tiende también a minimizar su momento de inercia, de influencia negativa en el funcionamiento del motor, crítico en el caso de motores de baja inercia. Puesto que los robots trabajan en ciclos cortos, que implican continuos arranques y paradas, es de gran importancia que el reductor sea capaz de soportar pares elevados puntuales. También se busca que el juego angular sea lo menor posible. Éste se define como el ángulo que gira al eje de salida cuando se cambia su sentido de giro sin que llegue a girar al eje de entrada. Por último, es importante que los reductores para robótica posean una rigidez torsional, definida como el par que hay que aplicar sobre el eje de salida para que, manteniendo bloqueado el de entrada, aquél gire un ángulo unidad.

4.7 Actuadores en los robots

Los actuadores tienen como misión generar el movimiento de los elementos del robot según las ordenes dadas por la unidad de control. Se clasifican en tres grandes grupos, según la energía que utilizan:

- Neumáticos
- Hidráulicos
- Eléctricos

Los actuadores neumáticos utilizan el aire comprimido como fuente de energía y son indicados en el control de movimientos rápidos, pero de precisión limitada. Los actuadores hidráulicos son recomendables en los manipuladores que tienen una gran capacidad de carga, junto a una precisa regulación de velocidad. Los motores eléctricos son los más utilizados, por su fácil y preciso control, así como por otras propiedades ventajosas que establece su funcionamiento, como consecuencia del empleo de la energía eléctrica.

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

- Potencia
- Controlabilidad

- Peso y volumen
- Precisión
- Velocidad
- Mantenimiento
- Costo

4.8 Sensores en los robots

Para conseguir que un robot realice su tarea con la adecuada precisión, velocidad e inteligencia, será preciso que tenga conocimiento tanto de su propio estado como del estado de su entorno. La información relacionada con su estado (fundamentalmente la posición de sus articulaciones) la consigue con los denominados **sensores internos**, mientras que la que se refiere al estado de su entorno, se adquiere con los **sensores externos**.

Entre los **sensores internos** se encuentran estos diferentes tipos:

- De posición
- De velocidad
- De presencia

Sensores externos: el empleo de mecanismos de detección exteriores permite a un robot interactuar con su ambiente de una manera flexible. Esto contrasta con el funcionamiento preprogramado en el que a un robot se le enseña a realizar tareas repetitivas mediante una serie de funciones preprogramadas. Aunque esto está bastante lejos de la forma más predominante de funcionamiento de los robots industriales actuales, la utilización de la tecnología de detección para proporcionar a las máquinas un mayor grado de inteligencia en relación con su ambiente es, en realidad, un tema activo de investigación y desarrollo en el campo de la robótica.

Un robot que puede ver y sentir es más fácil de entrenar en la ejecución de las tareas complejas mientras que, al mismo tiempo, exige mecanismos de control menos estrictos

que las máquinas preprogramadas. Un sistema sensible y susceptible de entrenamiento es también adaptable a una gama mucho más amplia de tareas, con lo que se consigue un grado de universalidad que se traduce, a la larga, en bajos costos de producción y mantenimiento. La función de los sensores del robot puede dividirse en dos categorías principales:

Estado interno y estado externo.

Los sensores de estado interno operan con la detección de variables, tales como la posición de la articulación del brazo, que se utilizan para el control del robot. Por el contrario, los sensores de estado externo operan con la detección de variables tales como:

- El alcance
- La proximidad
- Al contacto

La detección externa, se utiliza para el guiado del robot, así como para la manipulación e identificación de objetos.

Los sensores de estado externo pueden clasificarse también como sensores de:

- **Contacto:** como su nombre lo indica, la primera clase de sensores responde al contacto físico, tal como el tacto, deslizamiento y torsión.
- **No contacto:** los sensores de no contacto se basan en la respuesta de un detector de las variaciones en la radiación electromagnética o acústica. Los ejemplos más destacados de los sensores de no contacto miden el alcance, la proximidad y las propiedades visuales de un objeto:

Es de interés destacar que la detección de alcance y la visión suelen proporcionar una información de guiado aproximado para un manipulador, mientras que la proximidad y el tacto están asociados con fases terminales de agarre del objeto.

Los sensores de fuerza y torsión se utilizan como dispositivos de retroalimentación para controlar la manipulación de un objeto una vez que haya agarrado.

CAPÍTULO 5

LENGUAJES Y SISTEMAS DE PROGRAMACIÓN DE ROBOTS

5.1 Introducción

En este capítulo estudiaremos la interfaz entre el usuario humano y un robot industrial. Es mediante esta interfaz que el usuario puede aprovechar todos los algoritmos de mecánica y control.

El perfeccionamiento de la interfaz del usuario se ha vuelto más importante a medida que los manipuladores y demás tipos de automatización programable se emplean cada vez más en aplicaciones industriales demandantes. La naturaleza de la interfaz del usuario es una cuestión muy importante; por lo que la mayor parte del reto del diseño y del uso de robots industriales está enfocado a este aspecto del problema.

Los robots manipuladores se diferencian a sí mismos de la automatización fija al ser “flexibles”, lo cual significa que son programables. No solo son programables los movimientos de los manipuladores sino que, a través del uso de sensores de comunicación con otro tipo de automatización de fábrica. Los manipuladores pueden adaptarse a las variaciones a medida que se desarrolle la tarea.

Al considerar la programación de manipuladores es importante recordar que generalmente son sólo una parte mínima de un proceso automatizado. El término **célula de trabajo** se utiliza para describir una colección local de equipo, que puede incluir uno o más manipuladores, sistemas de transportadores, alimentadores de piezas y dispositivos fijos. En el siguiente nivel más alto, las células de trabajo pueden interconectarse en redes distribuidas en toda en toda la fábrica. De manera que una computadora central pueda controlar todo el flujo. Por lo tanto la programación de manipuladores se considera comúnmente dentro del problema más amplio de programar una variedad de máquinas interconectadas a una célula de trabajo de una fábrica automatizada.

5.2 Niveles de programación de robots

A través del tiempo se han desarrollado muchos estilos en la interfaz del usuario para programar robots. Antes de la rápida proliferación de microcomputadoras en la industria, los controladores de robots se asemejaban a los secuenciadores simples utilizados comúnmente para controlar la automatización fija. Los métodos modernos se enfocan a la programación de computadoras, y las cuestiones relacionadas con la programación de robots, incluyen todo lo referente a la programación de computadoras en general.

Enseñar mostrando

Los primeros robots se programaban mediante un método al que llamaremos **enseñar mostrando**, el cual implicaba mover el robot a un punto deseado y registrar su posición en una memoria que el secuenciador leería durante la reproducción. Durante la fase de enseñanza, el usuario guiaba al robot ya sea en forma manual o a través de la interacción con un **dispositivo de enseñanza**. Los dispositivos de enseñanza son cajas de botones portátiles que permiten el control de cada articulación del manipulador o cada grado de libertad cartesiano. Algunos controladores de este tipo permiten las pruebas y las ramificaciones, por lo que pueden introducirse simples programas que involucren la lógica. Algunos dispositivos de enseñanza tienen pantallas alfanuméricas y se aproximan en complejidad a las portátiles.



Fig. 5.1 El operador utiliza una interfaz de dispositivo de enseñanza para programar el manipulador [18]

Lenguajes de programación de robot explícitos

Generalmente estos lenguajes tienen características especiales que se aplican a los problemas de programación de manipuladores y por ello se les llama **lenguajes de programación de robots** (RPL). La mayoría de los sistemas que vienen equipados de un lenguaje de este tipo han retenido además una interfaz estilo dispositivo de enseñanza.

Dado que los lenguajes de programación de robots han tomado muchas formas, los dividiremos en tres categorías.

1. **Lenguajes de manipulación especializados:** se han creado desarrollado un lenguaje completamente nuevo que, aunque se enfoca en áreas específicas de los robots, podría considerarse un lenguaje general de programación de computadoras. Un ejemplo es el lenguaje VAL desarrollado para controlar a los robots industriales de Unimation; Inc. VAL se desarrolló específicamente como un lenguaje de control de manipuladores pues como lenguaje general de computadoras era bastante débil. Ejemplo, no soportaba números de punto flotante o cadenas de caracteres, y las subrutinas no podían pasar argumentos. **V-II**, una versión más reciente, ya contaba con estas características. **V+**, la encarnación actual de este lenguaje, incluye muchas características nuevas. Otro ejemplo de un lenguaje de manipulación especializado es **AL**, desarrollado en la universidad de Stanford y que en la actualidad es una reliquia, de todas formas proporciona buenos ejemplos de algunas características que todavía no se encuentran en la mayoría de lenguajes modernos (control de fuerza, paralelismo). Este lenguaje fue desarrollado en un entorno académico.
2. **Biblioteca de robótica para un lenguaje de computación existente:** estos lenguajes se han desarrollado a partir de un lenguaje popular de computación (por ejemplo pascal) al que se le agrega una biblioteca de subrutinas específicas para robots. el usuario escribe un programa en pascal haciendo uso de llamadas frecuentes al paquete de subrutinas predefinidas para las necesidades específicas del robot. Un ejemplo es **AR_BASIC** de América Cimflex, que esencialmente es una biblioteca de subrutinas para una implementación estándar de **BSIC.JARS**, desarrollado por el laboratorio de propulsión a chorro de la **NASA**, es un ejemplo de lenguaje de programación de robots basado en pascal.
3. **Biblioteca de robótica para un nuevo lenguaje de propósito general:** estos lenguajes de programación se han desarrollado creando primero un nuevo lenguaje de propósito general como base de programación y después suministrando una biblioteca de subrutinas predefinidas, específicas para robots. Ejemplo de estos lenguajes de programación son **RAPID**, desarrollado por **ABB Robotic**. **AML**, desarrollado por **IBM**, y **KAREL**, desarrollado por **GMF Robotics**. Los estudios de programas de aplicación reales para las células de trabajo robóticas han demostrado que un gran porcentaje de las instrucciones del lenguaje no son

específicas para robots, sino que la mayor parte de la programación de robots tiene que ver con la inicialización, prueba lógica y ramificación, etc. Por esta razón podría desarrollarse una tendencia a alejarse del desarrollo de lenguajes especiales para la programación de robots y moverse hacia el desarrollo de extensiones para lenguajes generales, como en las categorías 2 y 3 antes mencionadas.

Lenguajes de programación a nivel de tarea

El tercer nivel de la metodología de programación de robots está relacionado con los **lenguajes de programación a nivel de tarea** estos lenguajes permiten al usuario ordenar correctamente las submetas deseadas de la tarea, en vez de especificar los detalles de cada acción que el robot va a llevar a cabo. En dicho sistema el usuario puede incluir instrucciones en el programa de aplicación, con un nivel considerable más alto que un lenguaje de programación de robots explícito. Un sistema de programación de robots a nivel de tarea debe tener la habilidad de realizar muchas tareas de planeación automáticamente. Ejemplo, si se emite una instrucción para “sujetar el perno” el sistema debe planear una ruta que evite que el manipulador colisione con cualquier obstáculo circundante, y que elija automáticamente una buena posición de sujeción del perno y sujete el perno. En contraste, en un lenguaje de programación de robots explícito el programador debe hacer todas estas acciones.

Hay una buena distinción entre los lenguajes explícitos de programación de robots y los lenguajes de programación a nivel de tarea. Se están haciendo avances en los lenguajes explícitos de programación de robots para facilitar la programación, pero estas mejoras no pueden considerarse como componentes de un sistema de programación a nivel de tarea. Aún no existe la verdadera programación de manipuladores a nivel de tarea, pero desde hace tiempo es un tema de investigación que continúa vigente.

Un ejemplo de aplicación

En la figura siguiente se muestra una célula de trabajo automatizada que completa un pequeño subensamble, en un hipotético proceso de manufactura. La célula de trabajo consiste en un transportador bajo el control de una computadora que entrega una pieza de trabajo; una cámara conectada a un sistema de visión que se utiliza para localizar la pieza de trabajo en el transportador; un robot industrial (se muestra un PUMA 560) equipado con una muñeca detectora de fuerza; un pequeño alimentador ubicado en la superficie de trabajo que suministra otra pieza al manipulador; una prensa controlada por computadora que puede ser cargada y descargada por el robot; y una tarima sobre la que el robot coloca los ensamblajes terminados.



Fig. 5.2 A menudo una célula de trabajo se modela en forma simple, con un conjunto de tramas unidas a objetos relevantes [18].

Todo el proceso está regulado por el controlador del manipulador en una secuencia, como se describe a continuación.

1. Se indica al transportador que inicie y se detiene cuando el sistema de visión reporta que se ha detectado un soporte en el transportador.
2. El sistema de visión juzga la posición y la orientación del soporte en el transportador y lo inspecciona para verificar que no hay defectos, como un número incorrecto de hoyos.
3. Utilizando la salida del sistema de visión, el manipulador sujeta el soporte con una fuerza específica y se verifica la distancia entre las puntas de los dedos para asegurar que el soporte se haya sujetado correctamente. En caso de algún error, el robot se hace a un lado y se repite la tercera visión.
4. El soporte se coloca en el dispositivo fijo de la superficie de trabajo. En este punto se puede indicar al transportador que inicie de nuevo para colocar el siguiente soporte; esto es, los puntos 1 y 2 pueden empezar en paralelo con los siguientes pasos.
5. Se recoge una chaveta del alimentador y se inserta parcialmente en un hoyo cónico en el soporte. Se utiliza el control de fuerza para realizar esta inserción y para realizar comprobaciones simples de acuerdo a la terminación de esta tarea. (si el alimentador de chavetas está vacío lo notifica a un operador y el manipulador espera hasta que el operador le ordene continuar).
6. El robot sujeta el conjunto soporte-chaveta y lo coloca en la prensa.
7. Se ordena a la prensa que actúe, y está presiona la chaveta completamente hacia dentro del soporte. La prensa indica que ha terminado y el soporte se coloca de nuevo en el dispositivo fijo para una inspección final.

8. El conjunto se revisa con un sensor de fuerza para verificar que la chaveta este insertada correctamente. El manipulador detecta la fuerza de reacción cuando presiona hacia los lados de la chaveta y puede hacer las comprobaciones necesarias para saber qué está insertada la chaveta en el soporte.
9. Si el conjunto se encuentra en la posición correcta, el robot coloca la pieza terminada en la siguiente ubicación disponible en la tarima. Si está llena, se notifica al operador. Si el conjunto esta mal ensamblado, se tira en el bote de la basura.
10. Una vez terminado el paso 2 (iniciado anteriormente en paralelo) se vuelve al paso 3.

Este es un ejemplo de una tarea que es posible para los robots industriales de la actualidad, debe quedar claro que tal vez no sea posible definir dicho proceso mediante las técnicas de “enseñar mostrando”. Ejemplo, al tratar con tarimas es laborioso tener que programar todas las ubicaciones de todos los compartimentos; es preferible enseñar toda la ubicación de la esquina y luego calcular las otras a partir de las dimensiones de la tarima, además, por lo general es imposible especificar el señalamiento de interproceso y establecer el paralelismo mediante el uso de un dispositivo de enseñanza típico de una interfaz estilo menú. Este tipo de aplicación necesita un enfoque basado en el uso de un lenguaje de programación de robots para procesar la descripción.

5.3 Requerimientos de un lenguaje de programación de robots

Modelo del entorno

Por definición, los programas de manipulación deben involucrar el movimiento de objetos en el espacio tridimensional, por lo que está claro que cualquier lenguaje de programación de robots necesita un medio para describir dichas acciones. El elemento más común de los lenguajes de programación de robots es la existencia de **tipos geométricos** especiales. Ejemplo, se introducen los tipos para representar conjuntos de ángulos de articulación, posiciones cartesianas, orientadas y tramas.

Todos los movimientos se describen como una trama herramienta relativa a una trama estación, en donde las tramas de destino se crean a partir de expresiones arbitrarias que implican tipos geométricos.

Dado un entorno de programación de robots que soporta tipos geométricos, el robot y las demás máquinas, piezas y dispositivos fijos pueden modelarse si definimos variables con nombres asociadas con cada objeto. En la fig.5.3 muestra parte de nuestro ejemplo de célula de trabajo con tramas unidas en ubicaciones relevantes a la tarea. Cada una de estas tramas se representaran con una variable de tipo “trama” en el programa del robot.



Fig. 5.3 Una celda de trabajo [19].

En muchos lenguajes de programación de robots, la habilidad para definir variables con nombre de distintos tipos geométricos y hacer referencia a ellas en el programa forma la base del modelo del entorno.

Observe que las formas típicas de los objetos no son parte del modelo del entorno, y tampoco lo son las superficies, volúmenes, masas u otras propiedades. La extensión en la que se modelan los objetos en el entorno es una de las decisiones básicas al diseñar un sistema de programación de robots.

Ciertos sistemas de modelos del entorno permiten la noción de **fijaciones** entre objetos con el nombre; esto es, el sistema puede recibir una notificación de que se han “fijado” dos o mas objetos con nombre; a partir de ese momento, si un objeto se mueve explícitamente con una instrucción del lenguaje, se moverán todos los objetos fijados a esté. Así, una vez que se ha insertado la chaveta en el agujero del soporte en nuestra aplicación, el sistema recibirá una notificación (mediante una instrucción del lenguaje) de que se han fijado estos dos objetos. Los movimientos subsiguientes del soporte (es decir los cambios en el valor de la variable de trama “soporte”) harán que el valor almacenado para la variable “chaveta” se actualice al mismo tiempo.

Idealmente, un modelado de entornos incluiría mucho más información sobre los objetos con los que tiene que tratar el manipulador y sobre el manipulador en sí, un ejemplo, considere un sistema en el que los objetos se describen mediante modelos estilo CAD que representan la forma espacial de cualquier objeto mediante las definiciones de sus bordes, sus superficies o su volumen. Con estos datos disponibles

para el sistema, es más factible implementar muchas de las características de un sistema de programación a nivel de tarea.

Especificación del movimiento

Una función muy básica de un lenguaje de programación de robots es permitir la descripción de los movimientos que se espera que se realicen. El usuario interactúa con los planeadores de ruta y los generadores de estilo de movimiento que permiten especificar puntos vía, el punto de destino y si se va a utilizar el movimiento interpolado de las articulaciones o en línea recta cartesiana. Además, el usuario podría tener el control sobre la velocidad o duración de un movimiento.

La mayoría de los lenguajes tiene una sintaxis similar para la introducción de movimientos simples.

Las diferencias en la primitiva del movimiento básico de un lenguaje de programación de robots a otro se hacen mas aparentes si consideramos características como las siguientes.

- La habilidad para realizar matemáticas con tipos estructurados tales como tramas, vectores y matrices de rotación.
- La habilidad de describir entidades geométricas como tramas en varias representaciones convenientes; junto con la habilidad de realizar conversiones entre las representaciones.
- La habilidad de proporcionar restricciones sobre la duración o la velocidad de un movimiento específico; ejemplo, muchos sistemas permiten al usuario establecer la velocidad a una fracción del máximo, pero pocas le permiten especificar directamente una duración deseada o una velocidad de articulación máxima.
- La habilidad de especificar destinos en relación con varias tramas, incluyendo las tramas definidas por el usuario y las tramas en movimiento (en un transportador).

Flujo de ejecución

Al igual que en los lenguajes convencionales de programación de computadoras un sistema de programación de robot permite al usuario especificar el flujo de la ejecución; esto es, los conceptos tales como prueba y ramificación, ciclos, llamadas a subrutinas e incluso interrupciones se encuentran generalmente en los lenguajes de robots.

Otro hecho frecuente es la necesidad de monitorear varios procesos con algún tipo de sensor. Después ya sea mediante la interrupción o el sondeo, el sistema de robots debe

ser capaz de responder a ciertos eventos detectados por los sensores. La habilidad de especificar dichos monitores de eventos es soportada por algunos lenguajes de programación de robots.

Entorno de programación

Al igual que con un lenguaje computacional, un buen entorno de programación fomenta la productividad del programador. La programación de manipuladores es difícil y tiende a ser muy interactiva, con muchas secciones de prueba y error. Si se obligara al usuario a repetir continuamente el ciclo de “editar-compile-ejecutar” de los lenguajes compilados, la productividad sería baja. Por esta razón, la mayoría de los lenguajes de programación ahora son interpretados, de manera que las instrucciones individuales pueden ejecutarse de una en una durante el desarrollo y la depuración del programa.

Muchas de las instrucciones de un lenguaje ocasionan el movimiento de un dispositivo físico, por lo que la pequeña cantidad de tiempo para interpretar las instrucciones del lenguaje es insignificante. También se requiere del soporte típico de programación: editores de texto, depuradores y un sistema de archivos.

Integración de sensores

Una parte extremadamente importante de la programación de robots tiene que ver con la integración de sensores. El sistema debe tener, como mínimo, la capacidad de consultar sensores de tacto y fuerza, y de utilizar la respuesta en instrucciones “if-then-else” (si-entonces). La habilidad de especificar eventos de monitoreo para vigilar las transiciones en dichos sensores en un modo de segundo plano es también muy útil.

La integración de un sistema de visión permite que el sistema de visión envíe al sistema del manipulador las coordenadas del objeto de interés. Ejemplo, un sistema de visión ubica los soportes en la banda transportadora y regresa al controlador del manipulador su posición y orientación relativas a la cámara. Se conoce la trama de la cámara relativa la trama estación, por lo que puede calcularse una trama de destino deseada para el manipulador a partir de esta información.

Algunos sensores pueden ser parte de otro equipo en la célula de trabajo, un ejemplo, pueden ser algunos controladores de robot, que pueden utilizar la entrada proveniente de un sensor unido a una banda transportadora, de manera que el manipulador pueda rastrear el movimiento de la banda y adquirir objetos de esta a medida que se va moviendo.

La interfaz para las capacidades de control de fuerza, se da a través de instrucciones especiales del lenguaje que permiten al usuario especificar estrategias de fuerza. Dichas estrategias son, por necesidad, una parte integral del sistema de control del manipulador; el lenguaje de programación de robots simplemente sirve como interfaz para esas

capacidades. Para programar robots que hagan uso del control de fuerza activo se podrían requerir otras características especiales, tales como la habilidad de mostrar datos de fuerza recolectados durante un movimiento restringido.

En sistemas que soportan el control de fuerza activo, la descripción de la aplicación del movimiento. El lenguaje AL describe el control de fuerza activo en el inicio del movimiento, especificando seis componentes de rigidez (tres de translación y tres de rotación) y una fuerza de desvío. De esta manera, la rigidez aparente del manipulador es programable. Para aplicar una fuerza, por lo general, la rigidez se hace cero en esa dirección y se especifica una fuerza de desvío.

Problemas peculiares para los lenguajes de programación de robots

Los avances en años recientes han ayudado, pero aún es difícil programar robots. La programación de robots comparte todos los problemas de la programación convencional de computadoras, más ciertas dificultades adicionales ocasionadas por el mundo físico.

Comparación entre el modelo del entorno y la realidad externa.

Una característica central de un sistema de programación de robots es el modelo del entorno que se mantiene internamente en la computadora. Aun cuando este modelo es considerablemente simple, hay muchas dificultades para asegurar que concuerde con la realidad física que trata de modelar.

Las diferencias entre el modelo interno y la realidad externa hacen que los objetos se sujeten mal o no se sujeten, generen colisiones y otros problemas más leves.

Esta correspondencia entre el modelo interno y el externo debe establecerse para el estado inicial del programa, y debe mantenerse durante su ejecución. Durante la programación inicial o la depuración es generalmente responsabilidad del usuario, asegurarse que el estado representado en el programa corresponda con el estado físico de la célula de trabajo. A diferencia de la programación convencional, en donde solo se necesitan guardarse variables internas y restaurarse para restablecer una situación previa, en la programación de robots los objetos físicos comúnmente deben reposicionarse.

Además de la incertidumbre inherente en la posición de cada objeto, el manipulador en sí está limitado a un cierto grado de precisión, muy a menudo los pasos en el ensamblaje requieren que el manipulador haga movimientos que requieren mayor precisión de la que es capaz: un ejemplo común es insertar una chaveta en un hoyo en el que el espacio libre es de una magnitud menor que la precisión posicional del manipulador. Para complicar más las cosas, la precisión del manipulador generalmente varía sobre su espacio de trabajo.

Al tratar con estos objetos cuyas ubicaciones no se conocen exactamente, es esencial refinar de alguna manera la información posicional. Esto puede hacerse algunas veces con sensores (de visión o de tacto) o utilizando estrategias de fuerza apropiadas para los movimientos restringidos.

Durante la depuración de los programas del manipulador, es muy útil poder modificar el programa y después respaldar y probar el procedimiento otra vez. Para respaldar hay que restaurar el manipulador y los objetos que se están manipulando a un estado anterior. No obstante, al trabajar con objetos físicos no siempre es fácil, o incluso posible, es decir, deshacer una acción. Tenemos ejemplos en las operaciones de pintura, remachado, perforación o soldadura, que producen una modificación física en los objetos que se están manipulando. Por lo tanto, podría requerirse que el usuario obtenga una nueva copia para reemplazar la anterior que se ha modificado. Además, es probable que algunas de las operaciones realizadas justo antes de la que se va a reintentar también necesiten repetirse para establecer el estado apropiado necesario antes de que pueda reintentarse con éxito la operación deseada.

Sensibilidad al contexto

La programación ascendente (bottom-up programming) es un método estándar para escribir un programa computacional extenso en el que se desarrollan pequeñas piezas de bajo nivel de un programa y luego se juntan en piezas más grandes, formando un programa completo. Para que este método funcione es esencial que las piezas pequeñas sean relativamente insensibles a las instrucciones del lenguaje que las precede y que no haya suposiciones con relación al contexto en el que se ejecutan. Para la programación de manipuladores éste no es comúnmente el caso; el código que trabaja confiablemente al probarse aislado con frecuencia falla cuando se coloca dentro del contexto de un programa más grande. Estos problemas generalmente surgen de las dependencias en la configuración del manipulador y la velocidad de los movimientos.

Los programas de los manipuladores pueden ser altamente sensibles a las condiciones iniciales; ejemplo, la posición inicial del manipulador. En trayectorias de movimiento, la posición inicial podría influenciar la velocidad con la que se moverá el brazo durante cierta parte crítica del movimiento. Ejemplo, estas aseveraciones se cumplen para los manipuladores que siguen las rutas del tipo trazador cúbico en el espacio de articulación. Estos efectos algunas veces pueden tratarse mediante un cuidado apropiado en la programación, pero constantemente estos problemas no surgen sino hasta después que se han depurado en forma aislada las declaraciones iniciales del lenguaje y luego se unen con las instrucciones que las preceden.

Debido a una precisión insuficiente del manipulador, es muy probable que un segmento del programa escrito para realizar una operación en cierta ubicación necesite ajustarse (es decir, volver a enseñar las posiciones, etc.) para hacer que funcione en una ubicación

distinta. Los cambios de ubicación dentro de la célula de trabajo producen cambios en la configuración del manipulador dentro de la célula de trabajo, ponen a prueba la precisión de los comportamientos cinemáticos del manipulador y del sistema del servo, y con frecuencia surgen problemas. Dicha reubicación podría provocar un cambio en la configuración cinemática del manipulador, ejemplo, del hombro izquierdo al hombro derecho, o del codo hacia arriba al codo hacia abajo. Lo que es más, estos cambios en la configuración podrían ocasionar grandes movimientos cortos y simples.

Es muy probable que la naturaleza de la forma espacial de las trayectorias cambie a medida que se ubican las rutas en distintas porciones del espacio de trabajo del manipulador. Esto es especialmente cierto para los métodos de trayectorias en espacio de articulación, pero el uso de los esquemas de ruta cartesiana también puede producir problemas cuando hay singularidades cerca.

Al probar el movimiento de un manipulador por vez primera, a menudo es conveniente hacer que el manipulador se mueva lentamente. Esto da al usuario la oportunidad de detener el movimiento, si parece estar a punto de provocar una colisión; también le permite examinar el movimiento muy de cerca. Una vez que el movimiento ha pasado por cierta depuración inicial a una velocidad menor, es conveniente aumentar las velocidades del movimiento. Al hacer esto algunos aspectos del movimiento podrían cambiar. Las limitaciones de la mayoría de los sistemas de control de manipuladores producen errores de servo mayores, los cuales son de esperarse si se sigue la trayectoria más rápida. Además, en situaciones de control de fuerza que impliquen un contacto con el entorno, los cambios de velocidad pueden modificar completamente las estrategias de fuerzas requeridas.

La configuración del manipulador también afecta la delicadeza y precisión de las fuerzas que pueden aplicarse con él. Está en función de qué tan bien esta acondicionado el jacobiano del manipulador en cierta configuración, algo generalmente difícil de considerar al desarrollar el programa para robots.

Recuperación de errores

Otra consecuencia directa de trabajar con el mundo físico es que los objetos podrían no estar exactamente en donde deberían, y los movimientos que se realizan con esos objetos podrían fallar. Parte de la programación de manipuladores implica el tratar de tomar esto en cuenta y hacer las operaciones de ensamblaje lo más robustas posible pero, aún así, es probable que haya errores, y un aspecto importante de la programación de manipuladores es cómo recuperarse de estos errores.

Casi cualquier instrucción de movimiento en el programa del usuario puede fallar, por una gran variedad de razones. Algunas de las causas más comunes son que los objetos se desplacen o se resbalen de la mano, que un objeto no se encuentre en donde debería estar, que se atasque el objeto durante una inserción y que no se pueda localizar un hoyo.

El primer problema que surge para la recuperación de errores es identificar que efectivamente haya ocurrido un error. Como los robots generalmente tienen capacidades de detección y razonamiento bastante limitadas, a menudo la detección de errores es difícil. Para poder detectar un error, un programa de robots debe contener cierto tipo de prueba explícita. Esta prueba podría implicar el comprobar la posición del manipulador para ver que se encuentre en el rango apropiado; ejemplo, cuando se realiza una inserción, la falta de cambio de posición podría indicar un atascamiento, o demasiado cambio podría indicar que no pudo realizar el hoyo o que el objeto se resbaló de la mano. Si el sistema del manipulador cuenta con cierto tipo de capacidades visuales, entonces podría tomar una fotografía y revisar la presencia o ausencia de un objeto y, si el objeto está presente, reportar su ubicación. Otras comprobaciones podrían implicar el uso de fuerza, como pesar la carga que se lleva para revisar que el objeto aún está allí y no se haya caído, revisar que una fuerza de contacto permanezca dentro de ciertos límites durante un movimiento.

Todas las instrucciones de movimiento en el programa podrían fallar, por lo que estas comprobaciones explícitas pueden ser bastante complicadas y pueden ocupar más espacio que el resto del programa. Es en extremo difícil tratar de lidiar con todos los posibles errores; generalmente solo se comprueban las instrucciones con más probabilidad de fallar; y este proceso requiere de cierta cantidad de interacción y prueba parcial con el robot durante la etapa de desarrollo del programa.

Una vez que se ha detectado un error, puede hacerse un intento por recuperarse del mismo; el manipulador puede hacerlo bajo el control de programa o tal vez el usuario tenga que intervenir, o podría requerirse cierta combinación de las dos cosas. En cualquier caso el intento de recuperación podría producir nuevos errores. Es fácil ver como el código para recuperarse de los errores puede convertirse en la parte principal del programa del manipulador.

El uso del paralelismo en los programas de los manipuladores puede complicar aún más la recuperación después de errores. Cuando varios procesos se ejecutan concurrentemente y uno de ellos provoca un error, podría afectar otros procesos. En muchos casos es posible hacer retroceder el proceso que provoca el error, permitiendo a los otros continuar. Otras veces es necesario reiniciar varios o todos los procesos de ejecución.

5.4 Sistemas de programación fuera de línea

Definiremos un sistema un sistema de programación fuera de línea (OLP, por sus siglas en inglés) como un lenguaje de programación que se ha extendido lo suficiente, generalmente mediante gráficos computacionales, como para que el desarrollo de

programas de robot pueda llevarse a cabo sin necesidad de tener acceso al robot en sí. Los sistemas de programación fuera de línea son importantes como ayuda en la programación en la automatización industrial contemporánea, y también como plataforma de investigación. Hay que considerar muchas cuestiones en el diseño de dichos sistemas. Primero trataremos sobre estas cuestiones y después un sistema de este tipo.

Durante las dos últimas décadas, el crecimiento del mercado de robots industriales no es el que se esperaba; una de las principales causas es que los robots no son tan fáciles de usar. Es necesario gran cantidad de tiempo y experiencia para instalar un robot en una aplicación específica y hacer que el sistema esté listo para la producción.

Por varias razones este problema es más grande en ciertas aplicaciones que en otras, en ciertas áreas (ejemplo, la soldadura por punto y la pintura con pistola de aire) se están implantando robots más rápidamente que otras áreas de aplicación (ejemplo, el ensamblaje). Parece que la falta de implementadores de sistemas de robots con suficiente capacitación está limitando el crecimiento en ciertas áreas de aplicación. En ciertas compañías de fabricación, la gerencia alienta el uso de robots a un nivel mayor del que pueden realizar los ingenieros de aplicación. También, un alto porcentaje de los robots entregados se están utilizando en forma que no aprovechan sus capacidades. Estos síntomas indican que los robots industriales no son muy fáciles de usar como para permitir que se instalen y se programen con éxito en forma oportuna.

Muchos factores hacen que la programación de robots sea una tarea difícil. En primera instancia está relacionada intrínsecamente con la programación de computadoras en general y comparte muchos de los problemas de ese campo; pero la programación de robots, o de cualquier máquina programable, tiene problemas específicos que hacen que el desarrollo de software listo para la producción sea aun más difícil. La mayor parte de estos problemas surgen del hecho de que un robot manipulador interactúa con su entorno físico. Incluso hasta los sistemas de programación simples mantienen un “modelo global” de este entorno físico en la manera de ubicar los objetos, y tienen conocimiento de la presencia y ausencia de algunos objetos codificados en las estrategias del programa. Durante el desarrollo de un programa de robot (y especialmente después, durante su uso en la producción), es necesario que en el sistema de programación contenga el modelo interno en correspondencia con el estado actual del entorno del robot. La depuración interactiva de programa con un manipulador requiere reiniciar frecuentemente el estado del entorno del robot; las piezas, las herramientas y demás objetos necesitan regresar a sus posiciones iniciales.

Este proceso de reiniciar el estado se vuelve especialmente difícil (algunas veces costoso) cuando el robot realiza una operación irreversible en una o más piezas (por ejemplo, perforación o enrutamiento).

5.5 Controlador lógico programable

Los **CLP** o **PLC** (*Programmable Logic Controller* en sus siglas en inglés) son dispositivos electrónicos muy usados en la automatización industrial.

Su historia se remonta a finales de la década de 1960 cuando la industria buscó en las nuevas tecnologías electrónicas una solución para reemplazar los sistemas de control basados en circuitos eléctricos con relés, interruptores y otros componentes comúnmente utilizados para el control de los sistemas de lógica combinatoria

En la actualidad, los **PLC** no sólo controlan la lógica de funcionamiento de máquinas, plantas y procesos industriales, sino que también pueden realizar operaciones aritméticas, manejar señales analógicas para realizar estrategias de control, tales como controladores, los cuales pueden comunicarse con otros controladores y computadoras en redes de área local, y son una parte fundamental de los modernos sistemas de control distribuido.

Existen varios lenguajes de programación utilizados en los **PLC**, los más utilizados son el diagrama de escalera **LADDER**, que es lista de instrucciones y programación por estados, aunque se han incorporado lenguajes más intuitivos que permiten implementar algoritmos complejos mediante simples diagramas de flujo más fáciles de interpretar y mantener. Un lenguaje más reciente, es el **FBD** (en inglés Function Block Diagram) que emplea compuertas lógicas y bloques con distintas funciones conectados entre si.

En la programación se pueden incluir diferentes tipos de operadores, desde los más simples como lógica booleana, contadores, temporizadores, contactos, bobinas y operadores matemáticos, hasta operaciones más complejas como manejo de tablas (recetas), apuntadores, algoritmos PID y funciones de comunicación multiprotocolos que le permitirían interconectarse con otros dispositivos.

Características

La principal diferencia con otros dispositivos son las conexiones especiales de entrada/salida. Estas conexiones conectan el PLC a sensores y actuadores. Los PLC leen interruptores, indicadores de temperatura y las posiciones de complejos sistemas de posicionamiento. Algunos incluso pueden llegar a utilizar visión artificial. En los actuadores, los PLC pueden operar motores eléctricos y neumáticos, cilindros hidráulicos o diafragmas, relés magnéticos y solenoides. Las conexiones de entrada/salida pueden estar integradas en un solo PLC o el PLC puede tener módulos de entrada/salida unidos a una red de ordenadores que se conectan al PLC.

Los PLC fueron ideados para sustituir a los sistemas automáticos que podrían llegar a usar cientos o miles de relés y contadores. A menudo, un solo PLC puede programarse para reemplazar miles de relés. Los controladores programables fueron inicialmente adoptados por la industria automotriz, donde la revisión del software reemplazó a la reescritura o rediseño de los controles cada vez que cambiaban los modelos que se producían.

Los primeros PLC funcionaban con lógica en escalera (LADDER) que tenía mucha similitud con los diagramas eléctricos. Los operarios eran capaces de evitar los errores en los esquemas eléctricos utilizando lógica de escalera. Esta notación fue elegida para reducir las necesidades de aprendizaje de los técnicos existentes.

La funcionalidad de los PLC ha evolucionado a lo largo de los años para incluir un control de relés secuencial, control de movimiento, control de procesos, sistemas de control distribuidos y establecimiento de redes. El tratamiento de datos, el almacenaje, la energía del proceso y las capacidades de la comunicación de algún PLC moderno son aproximadamente equivalentes a las computadoras de escritorio. Una programación como la de los PLC combinada con hardware de E/S remota, permite a una computadora de escritorio igualar en ciertas aplicaciones a un PLC.

Algunos PLC's pueden ser programados usando lenguajes de programación estructurados. Una notación de programación gráfica llamada Sequential Function Charts esta disponible en determinados controladores programables.

PLC en comparación con otros sistemas de control

Los PLC están adaptados para un amplio rango de tareas de automatización. En los típicos procesos industriales de manufactura donde el costo de desarrollo y mantenimiento de un sistema de automatización es relativamente alto, y donde van a existir cambios en el sistema durante toda su vida operacional. Los PLC tienen todo lo necesario para manejar altas cargas de potencia; se requiere poco diseño eléctrico y el problema de diseño se centra en expresar las operaciones y secuencias en la lógica de escalera (o diagramas de funciones). Las aplicaciones del PLC son normalmente hechas a la medida del sistema, por lo que el costo del PLC es bajo comparado con el costo de la contratación del diseñador para un diseño específico que solo se va a usar una sola vez. Por otro lado, en caso de alta producción, los sistemas de control a la medida se amortizan por si solos rápidamente debido al ahorro en los componentes, lo que indica que puede ser una buena elección en vez de una solución "genérica".

Sin embargo, debe ser notado que algunos PLC no tienen un precio alto. Los PLC actuales tienen todas las capacidades por algunos cientos de dólares. Diferentes técnicas son utilizadas para un alto volumen o una simple tarea de automatización.

Un diseño basado en un microcontrolador puede ser apropiado donde cientos o miles de unidades deben ser producidas y entonces el costo de desarrollo (diseño de fuentes de alimentación y equipo de entradas y salidas) puede ser dividido en muchas ventas, donde el usuario final no tiene necesidad de alterar el control. Aplicaciones automotrices son un ejemplo, millones de unidades son vendidas cada año, y pocos usuarios finales alteran la programación de estos controladores.

Algunos procesos de control complejos, como los que son utilizados en la industria química, pueden requerir algoritmos y características más allá de la capacidad de PLC de alto nivel. Controladores de alta velocidad también requieren de soluciones a la medida; ejemplo, controladores para aviones.

Los PLC pueden incluir lógica para implementar bucles analógicos, “proporcional, integral y derivadas” o un controlador PID. Un bucle PID podría ser usado para controlar la temperatura de procesos de fabricación, ejemplo. Históricamente, los PLC’s fueron configurados generalmente con solo unos pocos bucles de control analógico y en donde los procesos requieren cientos o miles de bucles, un Sistema de Control Distribuido (DCS) se encarga de esa tarea. Sin embargo, los PLC se han vuelto más poderosos, y las diferencias entre las aplicaciones de DCS y PLC han quedado menos claras.

Señales Analógicas y digitales

Las señales digitales o discretas como los interruptores, son simplemente una señal de On/Off (1 ó 0, Verdadero o Falso, respectivamente). Los botones e interruptores son ejemplos de dispositivos que proporcionan una señal discreta. Las señales discretas son enviadas usando la tensión o la intensidad, donde un rango específico corresponderá al On y otro rango al Off. Un PLC puede utilizar 24V de corriente continua en la E/S donde valores superiores a 22V representan un On, y valores inferiores a 2V representan Off. Inicialmente los PLC solo tenían E/S discretas.

Las señales analógicas son como controles de volumen, con un rango de valores entre 0 y el tope de escala. Esto normalmente es interpretado con valores enteros por el PLC, con varios rangos de precisión dependiendo del dispositivo o del número de bits disponibles para almacenar los datos. Presión, temperatura, flujo, y peso son normalmente representados por señales analógicas. Las señales analógicas pueden usar tensión o intensidad con una magnitud proporcional al valor de la señal que procesamos. Las entradas de intensidad son menos sensibles al ruido eléctrico (ejemplo, el arranque de un motor eléctrico) que las entradas de tensión.

Capacidades E/S en los PLC modulares

Los PLC modulares tienen un limitado número de conexiones para la entrada y la salida. Normalmente, hay disponibles ampliaciones si el modelo base no tiene suficientes puertos E/S.

Los PLC con forma de rack tienen módulos con procesadores y con módulos de E/S separados y opcionales, que pueden llegar a ocupar varios racks. A menudo hay miles de entradas y salidas, tanto analógicas como digitales. A veces, se usa un puerto serie especial de E/S que se usa para que algunos racks puedan estar colocados a larga distancia del procesador, reduciendo el costo de cables en grandes distancias. Algunos de los PLC actuales pueden comunicarse mediante un amplio tipo de comunicaciones.

Los PLC usados en grandes sistemas de E/S tienen comunicaciones entre los procesadores. Esto permite separar partes de un proceso complejo para tener controles individuales mientras se permita a los subsistemas comunicarse mediante links. Estos links son usados a menudo por dispositivos HMI como keypads o estaciones de trabajo basados en PC.

El número medio de entradas de un PLC es 3 veces el de salidas, tanto en analógico como en digital. Las entradas "extra" vienen de la necesidad de tener métodos redundantes para controlar apropiadamente los dispositivos, y de necesitar siempre más controles de entrada para satisfacer la realimentación de los dispositivos conectados.

Programación

Los primeros PLC, eran programados usando sistemas de programación propietarios o terminales de programación especializados, que a menudo tenían teclas de funciones dedicadas que representaban los elementos lógicos de los programas de PLC. Los programas eran guardados en cintas. Más recientemente, los programas del PLC son escritos en aplicaciones especiales en un ordenador, y luego son descargados directamente mediante un cable o una red al PLC. Los PLC viejos usan una memoria no volátil (magnetic core memory) pero ahora los programas son guardados en una RAM con batería propia o en otros sistemas de memoria no volátil como las memoria flash.

Los primeros PLC fueron diseñados para ser usados por electricistas que podían aprender a programar los PLC en el trabajo. Estos PLC eran programados con "lógica de escalera" ("ladder logic"). Los PLC modernos pueden ser programados de muchas formas, desde la lógica de escalera hasta lenguajes de programación tradicionales como el BASIC o C. Otro método es usar la Lógica de Estados (State Logic), un lenguaje de programación de alto nivel diseñado para programar los PLC basándose en los diagramas de transición de estados.

Mientras que los conceptos fundamentales de la programación del PLC son comunes a todos los fabricantes, las diferencias en el direccionamiento E/S, la organización de la memoria y el conjunto de instrucciones hace que los programas de los PLC nunca se puedan usar entre diversos fabricantes. Incluso dentro de la misma línea de productos de un solo fabricante, diversos modelos pueden no ser directamente compatibles.

Interfaz de usuario

Los PLC necesitan interactuar con la gente para la configuración, las alarmas y el control diario. Para este propósito se emplean los interfaces hombre-maquina HMI.

Un sistema simple puede usar botones y luces para interactuar con el usuario. Las pantallas de texto están disponibles, al igual que las pantallas táctiles. La mayoría de los PLC modernos pueden comunicarse a través de una red con otros sistemas, ejemplo, con un ordenador con SCADA (Supervisory Control And Data Acquisition) o un navegador web.

5.6 Ejemplos de lenguajes de programación de robots

Gestual punto a punto: se aplican con el robot "in situ", recordando a las normas de funcionamiento de un magnetófono doméstico, ya que disponen de unas instrucciones similares: PLAY (reproducir), RECORD (grabar), FF (adelantar), FR (atrasar), PAUSE, STOP, etc. Además, puede disponer de instrucciones auxiliares, como INSERT (insertar un punto o una operación de trabajo) y DELETE (borrar). Este manipulador en línea funciona como un digitalizador de posiciones.

Los lenguajes más conocidos en programación gestual punto a punto son el

FUNKY: creado por IBM para uno de sus robots, y el T3, original de CINCINNATI MILACROM para su robot T3. Los movimientos pueden tener lugar en sistemas de coordenadas cartesianas, cilíndricas o de unión, siendo posible insertar y borrar las instrucciones que se desee. Es posible, también, implementar funciones relacionadas con sensores externos, así como revisar el programa paso a paso, hacia delante y hacia atrás.

En el lenguaje FUNKY se usa un mando del tipo "joystick", que dispone de un comando especial para centrar a la pinza sobre el objeto para el control de los movimientos, mientras que el T3 dispone de un dispositivo de enseñanza ("teach pendant").

El procesador usado en T3 es el AMD 2900 ("bit slice"), mientras que en el FUNKY está constituido por el IBM SYSTEM-7.

Lenguajes a nivel de movimientos elementales: existen lenguajes que expresan los movimientos de punto a punto, los cuales mantienen el énfasis en los movimientos primitivos, ya sea en coordenadas articulares, o cartesianas. En comparación, tienen, como ventajas destacables, los saltos condicionales a una subrutina, además de un aumento de las operaciones con sensores, aunque siguen manteniendo pocas posibilidades de programación "off-line".

Otros datos interesantes de este grupo de lenguajes son los siguientes:

ANORAD: se trata de una transformación de un lenguaje de control numérico de la casa ANORAD CORPORATION, utilizado para robot ANOMATIC. Utiliza, como procesador, al microprocesador 68000 de Motorola de 16/32 bits.

VAL: fue diseñado por UNIMATION INC para sus robots UNIMATE y PUMA. Emplea, como CPU, un LSI-II, que se comunica con procesadores individuales que regulan el servocontrol de cada articulación. Las instrucciones, en idioma inglés, son sencillas e intuitivas.

RPL: dotado con un LSI-II como procesador central, y aplicado a los robots PUMA, ha sido diseñado por SRI INTERNATIONAL.

EMILY: es un lenguaje creado por IBM para el control de uno de sus robots. Usa el procesador IBM 370/145 SYSTEM 7 y está escrito en Ensamblador.

SIGLA: desarrollado por OLIVETTI para su robot SUPER SIGMA, emplea un mini-ordenador con 8 K de memoria. Escrito en Ensamblador, es del tipo intérprete.

MAL: se ha creado en el Politécnico de Milán para el robot SIGMA, con un Mini-multiprocesador. Es un lenguaje del tipo intérprete, escrito en FORTRAN.

RCL: aplicado al robot PACS y desarrollado por RPI, emplea, como CPU, un PDP 11/03. Es del tipo intérprete y está escrito en Ensamblador.

Estos lenguajes son, por lo general, del tipo intérprete, con excepción del **RPL**, que tiene un compilador. La mayoría dispone de comandos de tratamiento a sensores básicos: tacto, fuerza, movimiento, proximidad y presencia. El **RPL** dispone de un sistema complejo de visión, capaz de seleccionar una pintura y reconocer objetos presentes en su base de datos.

Los lenguajes **EMILY** y **SIGLA** son transportables y admiten el proceso en paralelo simple.

Lenguajes estructurados de programación explícita: teniendo en cuenta las importantísimas características que presenta este tipo de programación, merecen destacarse los siguientes lenguajes.

A continuación, se exponen las características más representativas de los lenguajes dedicados a la programación estructurada.

AL: trata de proporcionar definiciones acerca de los movimientos relacionados con los elementos sobre los que el brazo trabaja. Fue diseñado por el laboratorio de Inteligencia Artificial de la Universidad de Stanford, con estructuras de bloques y de control similares al ALGOL, lenguaje en el que se escribió. Está dedicado al manipulador de Stanford, utilizando como procesadores centrales, a un PDP 11/45 y un PDP KL-10.

HELP: creado por GENERAL ELECTRIC para su robot ALLEGRO y escrito en PASCAL/FORTRAN, permite el movimiento simultáneo de varios brazos. Dispone, asimismo, de un conjunto especial de subrutinas para la ejecución de cualquier tarea. Utilizando como CPU, a un PDP 11.

MAPLE: escrito, como intérprete, en lenguaje PL-1, por IBM para el robot de la misma empresa, tiene capacidad para soportar informaciones de sensores externos. Utiliza, como CPU a un IBM 370/145 SYSTEM 7.

PAL: desarrollado por la Universidad de Purdue para el manipulador de Stanford, es un intérprete escrito en FORTRAN y Ensamblador, capaz de aceptar sensores de fuerza y de visión. Cada una de sus instrucciones, para mover el brazo del robot en coordenadas cartesianas, es procesada para que satisfaga la ecuación del procesamiento. Como CPU, usan un PDP 11/70.

MCL: lo creó la compañía MC DONALL DOUGLAS, como ampliación de su lenguaje de control numérico APT. Es un lenguaje compilable que se puede considerar apto para la programación de robots "off-line".

MAL EXTENDIDO: procede del Politécnico de Milán, al igual que el MAL, al que incorpora elementos de programación estructurada que lo potencian notablemente. Se aplica, también, al robot SIGMA.

Con excepción de HELP, todos los lenguajes de este grupo están provistos de estructuras de datos del tipo complejo. Así, el AL utiliza vectores, posiciones y transformaciones; el PAL usa, fundamentalmente, transformaciones y el MAPLE permite la definición de puntos, líneas, planos y posiciones. Sólo el PAL, y el HELP carecen de capacidad de adaptación sensorial. Los lenguajes AL, MAPLE y MCL, tienen comandos para el

control de la sensibilidad del tacto de los dedos (fuerza, movimiento, proximidad, etc.). Además, el MCL posee comandos de visión para identificar e inspeccionar objetos.

Lenguajes especificativos a nivel objeto: en este grupo se encuentran tres lenguajes interesantes:

RAPT: su filosofía se basa en definir una serie de planos, cilindros y esferas, que dan lugar a otros cuerpos derivados. Para modelar a un cuerpo, se confecciona una biblioteca con sus rasgos más representativos. Seguidamente, se define los movimientos que ligan a los cuerpos a ensamblar (alineación de planos, encajar cilindros, etc.).

El lenguaje RAPT fue creado en la Universidad de Edimburgo, departamento de Inteligencia Artificial; está orientado, en especial, al ensamblaje de piezas. Destinado al robot FREDY, utiliza, como procesador central, a un PDP 10. Es un intérprete y está escrito en lenguaje APT.

AUTOPASS: creado por IBM para el ensamblaje de piezas; utiliza instrucciones, muy comunes, en el idioma inglés. Precisa de un ordenador de varios Megabytes de capacidad de memoria y, además de indicar, como el RAPT, puntos específicos, prevé, también, colisiones y genera acciones a partir de las situaciones reales.

Un pequeño ejemplo, que puede proporcionar una idea de la facilidad de relacionar objetos, es el programa siguiente, que coloca la parte inferior del cuerpo C1 alineada con la parte superior del cuerpo C2. Asimismo, alinea los orificios A1 y A2 de C1, con los correspondientes de C2.

El AUTOPASS realiza todos sus cálculos sobre una base de datos, que define a los objetos como poliedros de un máximo de 20,000 caras. Está escrito en PL/1 y es intérprete y compilable.

LAMA: procede del laboratorio de Inteligencia Artificial del MIT, para el robot SILVER, orientándose hacia el ajuste de conjuntos mecánicos. Aporta más inteligencia que el AUTOPASS y permite una buena adaptación al entorno. La operatividad del LAMA se basa en tres funciones principales:

- Creación de la función de trabajo. Operación inteligente.
- Generación de la función de manipulación.
- Interpretación y desarrollo, de una forma interactiva, de una estrategia de realimentación para la adaptación al entorno de trabajo.

Lenguajes en función de los objetivos: la filosofía de estos lenguajes consiste en definir la situación final del producto a fabricar, a partir de la cual se generan los planes de acción tendentes a conseguirla, obteniéndose, finalmente, el programa de trabajo. Estos lenguajes, de tipo natural, suponiendo una potenciación extraordinaria de la Inteligencia Artificial, para descargar al usuario de las labores de programación. Prevén, incluso, la comunicación hombre-máquina a través de la voz.

Los lenguajes más conocidos de este grupo son:

STRIPS: fue diseñado, en la Universidad de Stanford, para el robot móvil SHAKEY. Se basa en un modelo del universo ligado a un conjunto de planteamientos aritmético-lógicos que se encargan de obtener las subrutinas que conforman el programa final. Es intérprete y compilable, utilizando, como procesadores, a un PDP-10 y un PDP-15.

HILAIRE: procedente del laboratorio de Automática Y Análisis de Sistemas (LAAS) de Toulouse, está escrito en lenguaje LISP. Es uno de los lenguajes naturales más interesantes, por sus posibilidades de ampliación e investigación.

5.7 ACL (Advanced control lenguaje)

Se listan las principales instrucciones del ACL, el lenguaje de programación de robots. Con el manejo de este lenguaje se pueden programar diferentes robots.

Comandos utilizados en modo directo (D) y en modo edit (E)

Definición y grabado de posiciones:

Defp: este comando sirve para definir una posición, reservando un espacio en la memoria del controlador; el nombre puede ser numérico o alfanumérico de hasta 5 caracteres.

Defp *pos* define una posición llamada *pos* en el grupo A. (D,E)

Defpb *pos* define una posición llamada *pos* en el grupo B. (D,E)

Defpc *pos* define una posición llamada *pos* en el grupo C (D,E).

Dimp: este comando permite definir un vector de una cantidad determinada de posiciones,

Dimp *pvec* [n] define un vector de n posiciones llamado *pvec* en el grupo A (D,E).

Dimpb *pvec* [n] define un vector de n posiciones llamado *pvec* en el grupo B (D,E).

Dimpc *pvec* [n] define un vector de n posiciones llamado *pvec* en el grupo C Cada posición se llamará *pvec* [1], *pvec* [2], ... *pvec* [n].

Delp: sirve para borrar una posición ya definida, libera el espacio reservado en memoria.

Delp *pos* borra la posición existente llamada *pos* (D, E).

Delp *pvec* borra el vector de posiciones *pvec* (D, E)

Undef: borra los valores de las coordenadas grabadas pero la posición sigue definida. Puede usarse en posiciones individuales o en vectores.

Undef *pos* borra el contenido de la posición existente llamada *pos* (D, E).

Here: graba la posición en la que se encuentra el robot en coordenadas joint.

Here *pos* graba en la posición *pos* definida anteriormente las coordenadas joint (D, E).

Herec: graba la posición en la que se encuentra el robot en el momento de ejecutar la instrucción, en coordenadas cartesianas.

Herec *pos* graba en la posición *pos* definida anteriormente, las coordenadas cartesianas (D, E).

Herer: graba las coordenadas en las que se encuentra el robot relativas a otra posición.

Herer *pos* graba en la posición *pos* las coordenadas joint referidas a la posición actual. Se deberán ingresar los valores de corrimiento luego de ejecutar la instrucción (D).

Herer *pos2 pos1* graba en la posición *pos2* definida anteriormente las coordenadas joint de corrimiento relativas a la posición *pos1* grabada previamente (D, E).

Teach graba coordenadas cartesianas de una posición del robot de acuerdo a valores ingresados por el usuario.

Teach *pos* graba en la posición *pos* las coordenadas cartesianas absolutas. Se deberán ingresar los valores de las coordenadas para definir la posición deseada (D).

>teach *pos*

X -- [.] >
Y -- [.] >
Z -- [.] >
P -- [.] >
R -- [.] >

Teachr: graba coordenadas cartesianas de una posición con respecto a otra ya grabada.

Teachr *pos* graba en la posición *pos* las coordenadas cartesianas relativas a la posición actual. Se deberán ingresar los valores de corrimiento para definir la posición deseada (D).

Teachr *pos2 pos1* graba en la posición *pos2* las coordenadas cartesianas relativas a la posición *pos1* grabada con anterioridad. Se deberán ingresar los valores de corrimiento para definir la posición deseada (D).

Attach *pvec* asigna las posiciones contenidas en el vector *pvec* a las posiciones del teach pendant (D).

Setpv *pos* graba la posición en *pos* en coordenadas joint (D).

Setpv *pos axis var* cambia una de las coordenadas joint de la posición *pos* Predeterminada (D, E).

Setpvc *pos coord var* cambia una de las coordenadas cartesianas de la posición *pos* predeterminada (D, E).

Movimiento a posiciones grabadas

Move *pos* el robot se mueve a la posición grabada en *pos* (D, E).

Move1 *pos* la pinza se mueve desde la posición actual hasta la posición *pos* en línea recta, siempre que esto sea posible (D, E).

Movec *pos1 pos2* mueve la pinza desde la posición en que se encuentra hasta la posición *pos1* siguiendo una trayectoria circular a través de la posición *pos2* (D, E).

Moved *pos* el robot se mueve a la posición grabada en *pos* y hasta que no llega hasta dicha posición no continúa con el programa (E).

Move1d *pos* la pinza se mueve desde la posición actual hasta la posición *pos*. En línea recta, siempre que esto sea posible y no continúa con el programa mientras no haya completado el (E).

Movecd *pos1 pos2* mueve la pinza desde la posición en que se encuentra hasta la posición *pos1* siguiendo una trayectoria circular a través de la posición *pos2* y no pasa a la siguiente línea de programa hasta que no se haya completado el movimiento (E).

Comandos de control de ejes

Open abre la pinza (D, E).

Close cierra la pinza (D, E).

Con habilita el servo control de todos los ejes, o de algún grupo en particular (D).

Con *axis* habilita el servo control de un eje especificado en *axis* (D).

Coff deshabilita el servo control de todos los ejes, o de algún grupo en particular (D).

Coff *axis* deshabilita el servo control de un eje especificado en *axis* (D).

Set anout[n] = DAC deshabilita el control del eje n y setea el valor de la tensión (D, E)

Speed *var* setea la velocidad de los ejes del grupo A para las instrucciones MOVE y MOVES, la variable *var* será un porcentaje del valor de velocidad máxima.

Speedl *var* setea la velocidad de los ejes del grupo A para las instrucciones MOVE y MOVES, la variable *var* será un porcentaje del valor de velocidad máxima. (D, E)

Comandos de control de programas

Run: permite ejecutar programas ya editados.

Run *prog* ejecuta el programa *proa* (D, E).

A: aborta la ejecución de programas.

A *prog* aborta la ejecución del programa *proa* (D).

<ctrl> + A aborta la ejecución de todos los programas que estén corriendo (D).

Stop: aborta la ejecución de programas.

Stop *prog* aborta la ejecución del programa *prog* (E).

Stop aborta la ejecución de todos los programas que estén corriendo (E).

Suspend y Continue: sirven para interrumpir y continuar ejecutando un programa.

Suspend prog interrumpe la ejecución del programa (D, E).

Continue prog continua la ejecución del programa interrumpido (D, E).

Delay var suspende la ejecución del programa durante el tiempo (en centésimas de segundo) especificado en *var* (E).

Wait var1 oper var2 suspende la ejecución del programa hasta que la condición se satisfaga (E).

Trigger prog by ejecuta un programa condicional al cambio en el estado de la {in/out} *n* {0/1} entrada o salida *n* (E).

Comandos de manipulación de programas

Copy prog1 prog2 copia el programa *prog1* a un nuevo programa llamado *prog2* (D).

Rename prg1 prg2 cambia el nombre del programa *prg1* a *prg2* (D).

Remove prog borra el programa *prog* y libera la memoria ocupada por él (D).

Empty pro borra las líneas del programa *prog* manteniéndolo existente y valido (D).

Edit prog activa el modo edit y llama al programa *prog* y permite modificar su contenido (D).

Exit sirve para salir del modo edit (E).

Comandos de reporte

Dir proporciona la lista de programas guardados en ram (D).

List prog muestra el listado de instrucciones que componen el programa *prog* (D9).

Listp muestra una lista de todas las posiciones definidas (D).

Listpv pos muestra en pantalla las coordenadas correspondientes a la posición *pos* (D).

Comandos de control I/O

Disable {in/out} n deshabilita la entrada o salida *n* (D).

Enable {in/out} n habilita la entrada o salida *n* (D).

Force {in/out} n {1/0} fuerza la entrada o salida *n* a un estado 1 o 0 (D)

Show din muestra el estado de las 16 entradas (D).

Show dout muestra el estado de las 16 salidas/entradas (D).

Set out[n]= {1/0} setea el estado de una salida (D).

Comandos de definición y manipulación de variables

Define var1 ... var12 define hasta 12 variables locales (E).

Global var1 ... var12 define hasta 12 variables globales (D,E).

Dim var[n] define un vector de n variables locales (E).

Dimg var[n] define un vector de n variables globales (D, E).

Delvar var elimina la variable *var* (D, E).

Purgue elimina todas las variables no utilizadas (D).

Comandos de programación

If var1 oper var2 inicia una subrutina si la relación entre dos variables es verdadera, el operador puede ser <, >, =, <=, >=, <> (E).

Andif var1 oper var2 combinación lógica con otro comando if (E).

Orif var1 oper var2 combinación lógica con otro comando if (E).

Else se ubica entre el if y el endif, comienza una subrutina cuando la condición if es falsa (E).

Endif fin de la subrutina if (E).

For var1=var2 to var3 ejecuta una subrutina para todos los valores de la Variable (E).

Endfor fin de la subrutina for (E)

Label n marca una subrutina de programa para ser ejecutada con goto (E).

Goto n continua la ejecución del programa en la línea especificada después de label (E).

Gosub prog transfiere el control al programa *prog* el programa actual se suspende mientras se ejecuta la subrutina (E).

5.8 SCORBASE para Windows

Descripción: SCORBASE para Windows es un software de control de robots que brinda un entorno amigable para la operación y la programación del robot.

El SCORBASE tiene las principales funciones del ACL con la comodidad del entorno Windows. El scorbace, entre otras cosas permite:

- control y la visualización en tiempo real de 5 ejes robóticos, la pinza y 2 ejes.
- soporte completo y visualización del estado, en tiempo real de las 16 entradas y salidas.
- definición de posición y visualización de coordenadas.
- movimientos con 10 velocidades seleccionables.
- más de 460 posiciones y 1000 líneas de programas activas.
- interrupciones de programa.
- programación de variables.
- grabación y lectura de programas.

El SCORBASE presenta una pantalla al estilo windows con los siguientes menús:

File: el menú file contiene las usuales funciones de windows las cuales permiten cargar, grabar e imprimir archivos (los cuales contienen programas y posiciones). También se puede salir del software.

Edit: el menú edit contiene las funciones usuales de windows para editar archivos que contengan programas de control de robots.

Run: el menú run contiene los comandos de SCORBASE para efectuar el homing de los ejes del robot y periféricos y los comandos para la ejecución de programas.

Options: el menú options permite definir las preferencias a la hora de operar el Software.

Controller: el menú controller permite configurar el controlador y ajustar la performance de las operaciones del controlador.

View: las opciones de pantalla en el menú view permiten mostrar un conjunto predefinido de ventanas de dialogo y menús del SCORBASE.

Programs: SCORBASE graba cada programa en ventanas separadas, esto permite editar programas mientras se usa otra ventana de SCORBASE.

Help: este menú brinda ayuda para la operación del software.

Edición de programas: Para editar un programa se necesitarán dos ventanas:

- Program window
- Command list.

Para empezar a editar un programa se debe ir al menú **file** y se debe elegir **open** o **new** si se desea editar un programa grabado o editar un programa nuevo respectivamente.

Las instrucciones a utilizar se encuentran en el comand list, seleccionando la ventana program window e ingresando el código correspondiente a la instrucción deseada que se encuentra en la lista de comandos, si fuera necesario fijar algún parámetro, automáticamente se abrirá una ventana de dialogo. Las instrucciones más comunes se pueden ingresar mediante los botones situados en la parte inferior de la pantalla.

Definición y grabado de posiciones: para grabar posiciones se utilizaran las siguientes ventanas:

- Manual movement
- Robot movement
- Teach positions (simple)
- Teach positions (expanded)
- List positions
- Encoders

- XYZ

Con el SCORBASE se podrán grabar posiciones absolutas o relativas en coordenadas joint o cartesianas. Además se podrán listar las posiciones guardadas.

Ejecución de programas

Las siguientes ventanas del SCORBASE se usan para activar y monitorear la ejecución de un programa

- Program window
- Digital I/O
- XYZ
- Encoders
- Log File
- Message Screen

Para empezar la ejecución de un programa colocar el cursor sobre alguna línea del programa en la pantalla activa e iniciar la ejecución mediante los iconos de RUN, mediante las opciones del menú de Run o mediante las teclas de función.

La ejecución puede ser de una sola línea, de un solo ciclo o continúa.

5.9 Perspectivas de los lenguajes de programación

Es mucho lo que se ha hecho en el área de la programación para la robótica; sin embargo aún no existe un lenguaje ideal para la programación de los robots. Son muchos los lenguajes creados hasta ahora, en parte las causas principales de esta amplia gama de lenguajes inadecuados o poco efectivos son:

- Cada lenguaje se ha diseñado tomando como base un robot en específico del mercado, lo que anula su universalidad y la posibilidad de emplearlo en modelos diferentes.
- Los lenguajes, en muchos casos, se dirigen hacia aplicaciones diferentes, lo que limita grandemente su utilización para la programación de otras tareas.

Hoy en el mundo existe un interés general para lograr un sistema de percepción del entorno cada vez más avanzado. Para esto se hace necesaria la ampliación de la Inteligencia Artificial, que interviene en la valoración del espacio exterior o entorno y determina los planes de acción alternativos o lo que es lo mismo la respuesta a la interacción con ese medio.

Características de un lenguaje ideal para la robótica

Algunas características básicas de un lenguaje ideal, serian:

- Claridad y sencillez.
- Claridad de la estructura del programa.
- Sencillez de aplicación.
- Facilidad de ampliación.
- Facilidad de corrección y mantenimiento.
- Eficacia.
- Transportabilidad sobre cualquier equipo mecánico o informático.
- Adaptabilidad a sensores (tacto, visión, etc.).
- Posibilidad de descripción de todo tipo de herramientas acoplables al manipulador.
- Interacción con otros sistemas.

En el aspecto de claridad y sencillez, la programación gestual es la más eficaz, pero impide la confección de programas propiamente dichos. Los lenguajes a nivel de movimientos elementales, como el VAL, disponen de bastantes comandos para definir acciones muy parecidas que fueron surgiendo según las necesidades y que, en gran medida, oscurecen su comprensión y conocimiento. Aunque, inicialmente, las técnicas de programación estructurada son más difíciles de dominar, facilitan, extraordinariamente, la comprensión y corrección de los programas.

Respecto a la sencillez de aplicación, hay algunos lenguajes (como el MCL) dedicados a las máquinas herramienta (APT), que pueden ser valorados, positivamente, por los usuarios conocedores de este campo. El PAL, estructurado sobre la matemática matricial, sólo es adecuado para quienes están familiarizados con el empleo de este tipo de transformaciones.

Uno de los lenguajes más fáciles de utilizar es el AUTOPASS, que posee un juego de comandos con una sintaxis similar a la del inglés corriente.

Es imprescindible que los lenguajes para los robots sean fácilmente ampliables, por lo que se les debe dotar de una estructura modular, con inclusión de subrutinas definidas por el mismo usuario.

La adaptabilidad a sensores externos implica la posibilidad de una toma de decisiones, algo muy interesante en las labores de ensamblaje. Esta facultad precisa de un modelo dinámico del entorno, así como de una buena dosis de Inteligencia Artificial, como es el caso del AUTOPASS.

Aunque los intérpretes son más lentos que los compiladores, a la hora de la ejecución de un programa, resultan más adecuados para las aplicaciones de la robótica. Las razones son las siguientes:

- El intérprete ejecuta el código como lo encuentra, mientras que el compilador recorre el programa varias veces, antes de generar el código ejecutable.
- Los intérpretes permiten una ejecución parcial del programa.
- La modificación de alguna instrucción es más rápida con intérpretes, ya que un cambio en una de ellas no supone la compilación de las demás.

Finalmente, el camino para la superación de los problemas propios de los lenguajes actuales ha de pasar, necesariamente, por la potenciación de los modelos dinámicos del entorno que rodea al robot, acompañado de un aumento sustancial de la Inteligencia Artificial.

CAPÍTULO 6

EJEMPLOS DE PROGRAMACIÓN DE ROBOTS

6.1 Descripción del robot Scorbot –ER Vplus

El **Scorbot – ER Vplus** es un robot vertical articulado con cinco juntas rotativas. Con el agregado de la mordaza, el robot posee seis grados de libertad. El SCORBOT ER Vplus es un elemento didáctico que permite una introducción a la robótica. Este dispositivo es de manejo similar a los utilizados en la industria, por lo que brinda la posibilidad de adquirir un aceptable entrenamiento en el manejo de brazos robóticos.

Numero de eje, nombre de la junta, y acción

- | | | |
|---|-------------|------------------------------------------|
| 1 | Base | Gira el cuerpo (body) |
| 2 | Shoulder | Sube o baja el brazo superior (uper arm) |
| 3 | Elbow | Sube o baja el antebrazo (fore arm) |
| 4 | Wrist pitch | Sube o baja la pinza (gripper) |
| 5 | Wrist roll | Gira la pinza (gripper) |

El controlador cuenta además con **16 entradas** (destinadas a recibir señales de dispositivos externos que interactúen con el robot, como sensores, pulsadores, etc.) y **16 salidas** (destinadas a transmitir señales hacia los dispositivos externos) de las cuales 4 son a relay y las restantes son de transistor.

En el controlador hay un botón de **paro de emergencia**, este una vez presionado desconecta el sistema y se enciende una luz roja. Para volver al funcionamiento normal se deberá presionar dicho botón nuevamente y re inicializar el sistema.

Puesta en marcha:

1- Una vez que se han hecho todas las conexiones de hardware necesarias, usted puede encender el controlador:

- Accione el interruptor de encendido del controlador.
- Accione el interruptor de encendido de motores.

Se encenderán en ambos casos LED's de color verde.

2- encienda su computadora y arranque en modo DOS posiciónese en el directorio ATS y tipee: ats / c2.

Una vez cargado el software, aparecerá la pantalla de ATS en su monitor.

Modos de operación:

Software

ACL (Advanced Control Language) es un avanzado lenguaje de programación de robots, el mismo incluye:

- Control directo de los ejes robóticos
- Programación por el usuario del sistema robótico.
- Control de entradas/salidas.
- Ejecución de programas simultanea, sincronizada e interactiva; soporte multi-tasking completo.
- Fácil gestionamiento de archivos

ATS (Advanced Terminal Software) es la interface para el ACL. Este software es un emulador de terminal para el acceso al ACL desde una PC. El ATS permite:

- Forma corta de configurar el controlador
- Definición de periféricos
- Editor de programas
- Teclas de short-cut para entrar.
- Backup manager

Teach pendant: es un dispositivo opcional, es una terminal de mano, usada para controlar el robot y los periféricos conectados al controlador. Es práctico para el movimiento de ejes, grabado de posiciones, direccionamiento de los ejes a posiciones grabadas y ejecución de programas.

Trabaja tanto en coordenadas joint como en cartesianas.

La posición de los ejes es medida por encoders que registran la variación de movimiento referida a una posición inicial. Para que el robot tenga un buen performance, esta posición deberá ser la misma cada vez que el robot sea utilizado. Para ello el controlador tiene un programa interno llamado **home**, el cual deberá ser ejecutado cada vez que se ponga en marcha el robot.

Para ejecutar el homing desde la PC teclee:

>home <enter>

Para ejecutar el homing desde el teach pendant (TP) oprima:

run – 0 – enter

Para inicializar los ejes (ya sea del robot o de los periféricos) teclee:

>home *nº de eje*

Operación del robot

MANUAL MODE (Modo manual): el modo manual es accionable cuando el sistema esta en modo directo, este modo permite el control directo de los ejes sin la necesidad de usar el teach pendant.

Para activar el modo manual solo basta presionar las teclas: **<alt> + m**

Teclas Acción

Q – 1	Axes 1
W – 2	Axes 2
E – 3	Axes 3
R – 4	Axes 4
T – 5	Axes 5
Y – 6	Axes 6
U – 7	Axes 7
I – 8	Axes 8

DIRECT MODE (Modo directo): cuando el sistema esta en DIRECT MODE, el usuario tiene el control directo de los ejes del robot. El controlador va ejecutando los comandos a medida que van siendo ingresados por el operador. Cuando se esta en modo directo, el prompt aparece en pantalla de la siguiente manera: >_

Grupos de control de ejes

Por defecto el controlador asume a los cinco ejes del robot dentro del grupo A de control, los ejes de los periféricos dentro del grupo B y los ejes independientes dentro del grupo C.

Sistemas de coordenadas

- **Coordenadas joint:** especifican la posición de los ejes robóticos en unidades de encoders, ya que estos proporcionan un número de señales proporcional al movimiento de los ejes.
- **Coordenadas cartesianas:** es un sistema geométrico para especificar la posición del punto central de la herramienta del robot TCP definiendo la distancia en unidades lineales (décimas de milímetros) desde el origen situado en la base.

Disposición de periféricos y sensores

La siguiente es la disposición de sensores y periféricos de la instalación del equipo en los laboratorios de la Universidad Nacional de Quilmas.

Referencia	descripción	conectado a
1	Sensor mecánico	IN 1
2	Sensor mecánico	IN 2
3	Sensor mecánico	IN 3
4	Sensor mecánico	IN 4
L	Luz	OUT 1
B	Bocina	OUT 2
5	Sensor de proximidad	IN 5
6	Sensor de proximidad	IN 6
7	Disco girador	AXES 7
8	Cinta transportadora	AXES 8
9	Controlador neumático	OUT 3

6.2 Descripción del Scrobot – ER IX

El **scrobot – ER IX** es un robot vertical articulado con cinco juntas rotativas. Con el agregado de la mordaza y el riel, el robot posee siete grados de libertad. encontramos en el SCORBOT ER Vplus una herramienta útil, tanto para el aprendizaje como para aplicaciones de laboratorio. El Scrobot ER IX permite adquirir cierto manejo de los

robots. Este dispositivo es de manejo similar a los utilizados en la industria, por lo que brinda la posibilidad de adquirir un aceptable entrenamiento en el manejo de brazos robóticos. Se aconseja a los principiantes utilizar algún robot más pequeño de la misma familia antes que centrarse en el estudio del Scorbot ER IX ya que debido a su robustez no es el más indicado para una introducción a la robótica.

Numero de eje Nombre de la junta Movimiento producido

1	Base	Rota la base
2	Shoulder	Sube y baja el brazo superior
3	Elbow	Sube y baja el antebrazo
4	Wrist pitch	Sube y baja la pinza
5	Wrist roll	Gira la pinza
6	Gripper	Abre o cierra la pinza
7	Riel	Mueve el robot sobre el riel.

Puesta en marcha: una vez que se han hecho todas las conexiones de hardware necesarias, usted puede encender el controlador:

- Accione el interruptor de encendido del controlador.
- Accione el interruptor de encendido de motores.

Se encenderán en ambos casos LED's de color verde.

Encienda su computadora y arranque en modo DOS:

posicionese en el directorio ATS y tipee: ats / c2

Una vez cargado el software, aparecerá la pantalla de ATS en su monitor. Para comenzar a utilizar el robot se deberá efectuar la inicialización del brazo y la del riel (axis 7).

Operación del robot:

MANUAL MODE (Modo manual): el modo manual es accionable cuando el sistema esta en modo directo, este modo permite el control directo de los ejes sin la necesidad de usar el teach pendant.

Para activar el modo manual solo basta presionar las teclas: <alt> + m

Teclas Acción

Q – 1	Axes 1
W – 2	Axes 2

E – 3 Axes 3
R – 4 Axes 4
T – 5 Axes 5
Y – 6 Axes 6
U – 7 Axes 7

DIRECT MODE (Modo directo): cuando el sistema esta en DIRECT MODE, el usuario tiene el control directo de los ejes del robot. El controlador va ejecutando los comandos a medida que van siendo ingresados por el operador.

Cuando se esta en modo directo, el prompt aparece en pantalla de la siguiente manera:
>
_

Teach pendant: es un dispositivo opcional, es una terminal de mano, usado para manejar al robot y los periféricos conectados al controlador. Es práctico para el movimiento de ejes, grabado de posiciones, direccionamiento de los ejes a posiciones grabadas y ejecución de programas.

Trabaja tanto en coordenadas joint como en cartesianas. Tiene una llave selectora para elegir el modo de mando entre *auto* o *manual*. La posición auto permite operar al robot desde la PC, la posición manual deshabilita el mando desde la computadora y habilita el teach pendant.

Posee también un pulsador de parada de emergencia con enclavamiento que desactiva los motores del brazo mientras se encuentra accionado y un botón lateral que también funciona como parada de emergencia cuando el sistema se encuentra en modo manual.

Sistemas de coordenadas:

- **Coordenadas joint:** especifican la posición de los ejes robóticos en unidades de encoders, ya que estos proporcionan un número de señales proporcional al movimiento de los ejes.
- **Coordenadas cartesianas:** es un sistema geométrico para especificar la posición del punto central de la herramienta del robot TCP definiendo la distancia en unidades lineales (décimas de milímetros) desde el origen situado en la base

6.3 Ejemplos de los lenguajes de programación utilizados en los robots

Para ejemplificar las sintaxis para las primitivas del movimiento en un robot manipulador se muestran algunos ejemplos del lenguaje utilizado en AVL:

- Moverse a la posición “destino 1”
- Moverse en línea recta hasta la posición “destino 2”
- Moverse, sin detenerse , a través de “vía 1”
- Detenerse completamente en “destino 3”

Se parte de la premisa que todos esos puntos de ruta ya se han enseñado o se han descrito textualmente, este pequeño programa se escribiría así en AVL II (Advanced Control Language):

MOVE DESTINO1

MOVE DESTINO2

MOVE VIA1

MOVE DESTINO3

En AL (aquí se controla el manipulador “garm”):

MOVE GARM TO DESTINO1;

MOVE GARM TO DESTINO2 LINEARLY;

MOVE GARM TO DESTINO3 VIA VAI1;

En sistemas que soportan el control de fuerza activo, la descripción la descripción de fuerza deseada podría convertirse en parte de la especificación del movimiento. El lenguaje AL describe el control de fuerza activo en los inicios del movimiento especificando seis componentes de rigidez (tres de traslación y tres de rotación) y una de fuerza de desvío.- De esta manera, la rigidez aparente del manipulador es programable. Para aplicar una fuerza, por lo general la rigidez se hace cero en esa dirección y se especifica una fuerza de desvío; ejemplo,

MOVE GARM TO DESTINO

WITH STIFFNESS= (80, 80, 0, 100, 100, 100)

WITH FORCE =20*OUNCES ALONG ZHAT;

La mayoría de los sistemas comerciales para simular robots u otros mecanismos no tratan directamente con este problema. En su lugar, “permiten” al usuario utilizar comandos específicos dentro del programa escrito para controlar el dispositivo simulado, un ejemplo sencillo sería la siguiente secuencia de código:

```
MOVE TO recoger_pieza  
CLOSE pinza  
AFFIX (pinza, pieza[i]);  
MOVE TO colocar_pieza  
OPEN pinza  
UNAFFIX (pinza, pieza[I];
```

Aquí se obliga al usuario a insertar comandos de “fijación” (affix) y “liberación” (unaffix) los que hacen que la pieza se mueva con la pinza de sujeción cuando se recoge y que deje de moverse cuando se libera, Si el simulador permite que el robot sea programado en su lenguaje nativo, por lo general ese lenguaje no es lo suficientemente completo como para soportar estos comandos, por lo tanto requiere un segundo grupo de comandos, incluso hasta podría tener una sintaxis distinta para lidiar con las interacciones del mundo real

Otro ejemplo de programación en ACL, describiendo la función de cada línea.

El programa lleva a cabo un ciclo de trabajo tomando una pieza de la posición del sensor 3, la deposita en la cinta transportadora, cuando la pieza llega a destino el brazo la traslada hasta la mesa giratoria, allí la deja y espera a que se complete una vuelta para luego tomarla y depositarla sobre el sensor 3. En este caso particular las posiciones están definidas para utilizar el SCORBOT ER-Vplus con la pieza cilíndrica intermedia.

PROGRAM DEMO

MOVED 0 (lleva la pinza a la posición 0)

SPEED 20 (fija la velocidad a un 20% de la velocidad máxima)

LABEL 1 (establece una marca en el programa)

IF IN[3] = 0 (ejecuta la subrutina siempre que el estado de la entrada 3 sea 0)

GOTO 1 (el programa continua en la línea marcada con label 1)

ENDIF (fin de la subrutina if)

OPEN (abre la pinza)

MOVE DEMO [10] (mueve la pinza a la posición demo[10])

MOVED DEMO[1] (mueve la pinza a la posición demo[1])

CLOSE (cierra la pinza y toma el objeto)

MOVECD DEMO[3] DEMO[2] (va con movimiento circular a la posición demo[2] pasando por demo[3])

OPEN (abre la pinza)

MOVED DEMO[4] (lleva la pinza a la posición demo[4])

SET ANOUT[8] = -1000 (fija una tensión de -1 volt en el eje 8)

WAIT IN[6] = 1 (detiene el programa hasta que se active el sensor 6)

DELAY 160 (detiene el programa 1,6 seg.)

SET ANOUT[8] = 0 (fija una tensión de 0 volt en el eje 8)

MOVED DEMO[5] (lleva la pinza a la posición demo[5])

CLOSE (cierra la pinza)

MOVED DEMO[6] (lleva la pinza a la posición demo[6])

MOVED DEMO[7] (lleva la pinza a la posición demo[5])

OPEN (abre la pinza)

MOVED DEMO[8] (lleva la pinza a la posición demo[8])

SET ANOUT[7] = 1500 (fija una tensión de 1,5 volt en el eje 7)

DELAY 1000 (detiene la ejecución del programa durante 10 segundos)

WAIT IN[5] = 1 (detiene la ejecución del programa hasta que no se accione el sensor 5)

SET ANOUT[7] = 0 (fija una tensión de 0 volt en el eje 7)

MOVED DEMO[7] (lleva la pinza a la posición demo[7])

CLOSE (cierra la pinza)

MOVED DEMO[8] (lleva la pinza a la posición demo[8])

MOVE DEMO[10] (lleva la pinza a la posición demo[10])

MOVED DEMO[9] (lleva la pinza a la posición demo[9])

OPEN (abre la pinza)

MOVE DEMO[10] (lleva la pinza a la posición demo[8])

MOVE 0 (lleva la pinza a la posición 0)

END (fin del programa)

Posiciones:

X Y Z P R

DEMO[1] 125 3623 918 -868 56

DEMO[2] 1755 3270 2260 -900 204

DEMO[3] 3334 2815 1010 -895 514

DEMO[4] 2786 279 3428 -895 825

DEMO[5] 3556 -2680 972 -894 1305

DEMO[6] 2127 -3835 2535 -894 1305

DEMO[7] 100 -3986 2239 -894 1792

DEMO[8] 76 -3032 4130 -874 1791

DEMO[9] 697 4319 944 -869 184

DEMO[10] 836 3758 2259 -850 48

Precauciones:

- Desconecte los motores mediante el switch del controlador antes de entrar en el área de trabajo del robot.
- No sobrecargue el brazo de robot, el peso entre la pinza y la carga de trabajo no debe exceder lo recomendado (1kg. Para el vplus y 2kg para el ix).
- No use la fuerza física para mover o detener alguna parte del brazo robótico.
- No conduzca al robot a través de obstáculos.
- No deje el brazo extendido por más de algunos segundos.
- No deje ningún eje del robot bajo esfuerzos mecánicos durante mucho tiempo, especialmente la pinza.
- No entre al área de trabajo del robot cuando este está operando.
- Asegúrese de no tener ropa o cabello suelto cuando trabaja con el robot.
- No ponga el dedo entre las mordazas de la pinza.
- El equipo deberá operarse solo en presencia de algún responsable
- se recomienda cubrir el robot una vez finalizada la operación del mismo
- tenga en cuenta que el scorbot er-ix es mas robusto que el vplus, por lo tanto es mas peligroso

6.4 Estudio del Robot Scorbot ER-VII

Resumen: en el presente artículo se muestra la importancia del uso adecuado de la programación de un robot manipulador ya que se deben tomar en cuenta los diferentes factores que pudiesen afectar directa o indirectamente el desempeño industrial de dicho robot, además se presenta la validación experimental de algunos programas que ejemplifican los posibles movimientos del Robot Scorbot ER-VII.

Introducción

Hasta nuestros días no existe una norma que indique cual es el procedimiento para la programación de robots; como consecuencia cada fabricante desarrolla su propio método, sin embargo, en todos ellos se dan una serie de características comunes y el criterio mas utilizado para la programación de robots hace referencia al sistema empleado para indicar la secuencia de acciones a seguir, como cuando se mueve físicamente al robot para registrar esa configuración y el otro es utilizar un lenguaje de programación. No obstante, en la actualidad los sistemas de programación tienden a combinar ambos métodos. Para la robótica, un robot industrial es básicamente un manipulador multifuncional reprogramable, permitiendo así, que sea capaz de adaptarse rápida y económicamente a las diferentes aplicaciones que existen dentro de la industria manufacturera. Es entonces, que la programación de un robot se vuelve indispensable, puesto que es el proceso mediante el cual se indica la secuencia de acciones que deberá llevar a cabo durante la ejecución de alguna tarea. Estas acciones consisten por lo regular en moverse a puntos predefinidos y manipular objetos.

Es este trabajo se hace un enfoque al lenguaje de programación ACL (Advanced Control Language)

Aspectos a considerar antes de programar un robot

La programación de cualquier robot esta sujeta a ciertas restricciones que es recomendable tomar en cuenta, antes de pensar en siquiera moverlo de una posición a otra. Algunos de los aspectos primordiales que se cubrieron durante el aprendizaje de la programación del Scorbot fue estudiar a fondo las características principales del robot, tentativas a influir en la ejecución de los programas, como son:

- Grados de libertad y/o articulaciones que posee.
- Sistema de coordenadas de operación.
- Espacio de trabajo.
- Limites operacionales de cada eje.
- Velocidad óptima de trabajo.
- Perfil de movimiento.

Una vez que se han estudiado los aspectos anteriores ahora es preciso profundizar sobre la sintaxis y comandos permitidos en el lenguaje de programación del robot, tendiendo

el debido cuidado ya que un programa mal elaborado pudiera afectar el estado electromecánico.

Modos de programación del Scorbot ER-VII

Como se mencionó anteriormente ACL es el nombre del lenguaje encargado de la programación del manipulador Scorbot ER-VII, el cual cuenta con una serie de características que para este caso han sido englobadas en los modos de programación.

ACL permite dos modos de programación los cuales son: programación por guiado que a su vez se divide en pasivo y activo y la programación textual.

Programación por guiado básicamente permite al usuario trabajar con el robot de tal manera que no sea necesario el conocimiento total de su lenguaje de programación.

Pasivo: es cuando el robot está trabajando en modo directo, esto quiere decir que el controlador funciona a modo de un shell. El usuario introduce las órdenes por el teclado seguido de un enter y el controlador las ejecuta tan pronto como los recibe. Al trabajar de esta forma se puede activar el control manual por teclado esto permite el control inmediato de los ejes del robot desde el teclado.

Este modo es útil cuando no se dispone de un panel portátil mejor conocido como teach pendant



Fig. 6.1 Muestra un teach pendant o panel portátil [20].

Cuando se encuentra en modo manual se dice que está en modo de ejes es decir se puede mover independientemente cada una de las articulaciones. Dentro del modo manual también existe un modo cartesiano mediante el cual el robot puede moverse en las direcciones XYZ. [2][1]

Activo: es en el cual se utiliza el sistema de accionamiento del robot a través de una botonera o teach pendant. El teach pendant es una terminal de mano que permite al operador el control directo del robot y de los ejes.

Además de controlar el movimiento de los ejes se utiliza también para el almacenamiento de posiciones, así como el envío de los ejes a determinadas posiciones.

Programación textual

Se refiere a realizar movimientos con el robot mediante un programa. Cuando ACL se encuentra en este modo se dice que está en modo editor, el cual se utiliza para crear y editar programas.

Mientras este modo se encuentra activo, en la pantalla se muestra la línea de programa actual.[2]

Para que el modo editor se active solo se tecléa EDIT y el nombre del programa que se desea editar.

Ejemplo: > EDIT PROG1

El sistema muestra la siguiente leyenda:

```
PROG1 NEW PROGRAM  
DO YOU WANT TO CREATE THAT NEW PROGRAM (Y/N)>
```

Después de que aparece esa leyenda se tecléa <Y> para poder crear el programa, posteriormente se desplegará en la pantalla lo siguiente:

```
PROGRAM PROG1
```

```
143:?
```

Ahora solo se edita el programa que se desea y al analizar el programa únicamente se tecléa el comando exit y el controlador validará dicho programa.

Programación en ACL

Cuando se trabajó con el lenguaje de programación del Scrobot ER-VII relativamente se utilizaron los dos modos de programación que éste ofrece, ya que al combinar ambos modos se pueden obtener mejores resultados al optimizar los tiempos de grabación de

las posiciones que serán utilizadas durante la ejecución de alguna tarea o trayectoria que el robot ejecutará. Por ejemplo si el robot debe iniciar desde una posición predefinida al ejecutarse las primeras líneas de algún programa.

Al igual que en la mayoría de los lenguajes de programación, ACL cuenta con comandos, variables de sistema y usuario, parámetros entre otros. Cabe destacar que el robot posee comandos para controlar cada eje del robot así como comandos para manipular y definir posiciones que se desean grabar para que el manipulador las ejecute cuando se requiera.

Validación Experimental

Los programas que se muestran enseguida son los experimentos realizados en el laboratorio, en lo cuales se hace uso de los comandos que posee éste lenguaje de programación. Los programas consisten en calcular los puntos que recorrerá el Scorbot ER-VII al ejecutar la trayectoria, así como recorrer la trayectoria. En éste caso se realizaron cuatro programas para dos trayectorias distintas como son parabólica y sinusoidal.

Programas de la trayectoria parabólica

El programa CALC1 está encargado de calcular los puntos de la trayectoria que recorrería el robot, dada una ecuación de una parábola (ecuación 1). El programa PARAB1 se utiliza para que el robot realice la trayectoria interpolada recorriendo los puntos que fueron calculados con el programa CALC1, es decir con este programas se mueve al robot. La trayectoria a seguir por el robot es:

$$Z = Y^2 = 5000 + 1000 \quad (1)$$

Donde:

PARAB1 (30 posiciones)

-135 mm <= Y <= +135 mm

X=533 mm

El primer paso a realizar es definir variables a utilizar en el programa y el vector de posiciones, para ello se utilizan los comandos siguientes:

DEFINE VY I VZ (define las variables locales).

DIMP VV[30] (define un vector de 30 posiciones, el vector %también es local)

PROGRAM CALC1

SET VY = 1250 (se asigna el valor 1250 a la variable VY, 1250 esta en milímetros)

FOR I = 1 TO 30 (ejecuta una subrutina para todos los valores de I empezando de 1 hasta terminar en 30)

SET VY=VY + 100

SET VZ=VY * VY

SET VZ=VZ / 5000

SET VZ=VZ + 1000

HERE VV[I] (define la posición VV[I] como una posición absoluta, VV[I] es el vector a calcular)

SETPVC VV[I] X 5333 (la coordenada X del vector VV[I] toma el valor de 5333)

SETPVC VV[I] Y VY (la coordenada Y del vector VV[I] es previamente calculada por el valor de VY)

SETPVC VV[I] Z VZ (la coordenada Z del vector VV[I] es previamente calculada por el valor de VZ)

SETPVC VV[I] P -44.5 (al ángulo del roll del vector VV[I] se le asigna el valor de 44.5 grados)

SETPVC VV[I] R 97.6 (al ángulo del roll del vector VV[I] se le asigna el valor de 97.6 grados)

DELAY 1 (demora la ejecución del programa por 10 milisegundos)

PRINT I (imprime la variable I)

ENDFOR (termina la subrutina)

PRINTLN " Vector VV creado" (imprime la leyenda que se encuentra entre " ")

(END) (fin del programa)

PROGRAM PARAB1

MOVED VV[1] (mueve al robot a la posici_on 1 del vector VV[I])

MPROFILE TRAPEZE A (cambia el perfil de movimiento a trapecoide)

LABEL 1 (inicia una subrutina de programa y se ejecuta al leer el comando GOTO)

MOVESD VV 1 30 (mueve al robot secuencialmente a lo largo de las posiciones 1,2,3,...,29,30 del vector VV[I])

MOVESD VV 30 1

GOTO 1 (salta inmediatamente a leer el comando LABEL)

(END) (fin del programa)

Programas de la trayectoria sinusoidal

El criterio de programación para el seguimiento de una trayectoria sinusoidal es el mismo que el descrito anteriormente y cuyos programas se presentan a continuación.

$$Z = 1500\sin Y + 2000 \quad (2)$$

Donde:

SINE 1 (34 posiciones)

X=670 mm

-178 mm <= Y <= +178 mm

50 mm <= Z <= 178 mm

El programa SINE1 está encargado de calcular los puntos de la trayectoria que recorrerá el robot, dada una ecuación de una parábola (ecuación 2).El programa RSSIN se utiliza para mover al robot a través de la trayectoria de la función sinusoidal.

DEFINE YVY I ZVZ (define las variables locales)

DIMP SS1[34] (define un vector de 34 posiciones, el vector también es local)

PROGRAM SINE1

SET YVY = 1731 (se asigna el valor 1731 a la variable YVY 1731 está en centésimas de milímetro)

FOR I = 1 TO 34 (ejecuta un subrutina para todos los valores de I empezando de 1 hasta terminar en 34)

SET YVY=YVY + 50

SET ZVZ=YVY * 360 (la función seno es calculada entre sin (-360o) y sin (360o))

SET ZVZ=ZVZ / 3000

SET ZVZ=1500 SIN ZVZ

SET ZVZ=ZVZ + 2000

HERE SS1[I] (define la posición SS1[I] como una posición absoluta, SS1[I] es el vector a calcular)

SETPVC SS1[I] X 6709 (la coordenada X del vector SS1[I] toma el valor de 6709)

SETPVC SS1[I] Y YVY (la coordenada Y del vector SS1[I] es previamente calculada por el valor de YVY)

SETPVC SS1[I] Z ZVZ (la coordenada Z del vector SS1[I] es previamente calculada por el valor de ZVZ)

SETPVC SS1[I] P -83 (al ángulo del roll del vector SS1[I] se le asigna el valor de 8.3 grados⁹)

SETPVC SS1[I] R 976 (al ángulo del roll del vector SS1[I] se le asigna el valor de 97.6 grados)

DELAY 1 (demora la ejecución del programa por 10 milisegundos)

PRINT I (imprime la variable I)

ENDFOR (termina la subrutina)

PRINTLN "vector ss1 creado" (imprime la leyenda que se encuentra entre " ")

(END) (fin del programa)

PROGRAM RSSIN

MOVED SS1[1] (mueve al robot a la posición 1 del vector SS1[1])

MPROFILE TRAPEZE A (cambia el perfil de movimiento a trapezoide inicia una)

LABEL 1 (subrutina de programa y se ejecuta al leer el comando GOTO)

MOVESD SS1 1 34 (mueve al robot secuencialmente a lo largo de las posiciones 1,2,3,...,29,30 del vector SS1[I])

MOVESD SS1 34 1

GOTO 1 (salta inmediatamente a leer el comando label)

(END) (fin del programa)

Después de ejecutarse estos programas generan valores en pulsos de encoger los cuales son de gran utilidad para visualizar digitalmente la trayectoria que realizó el robot físicamente.

Conclusiones del capítulo

Conocer las diferentes formas en que se puede programar al robot manipulador Scorbot ER-VII, es de vital importancia ya que la manipulación a nivel programación influye de gran manera para la realización de programas que contengan rutinas o tareas que al ejecutarse sean óptimas y válidas para el robot.

Así en este capítulo se presentó el lenguaje de programación ACL el cual proporciona comandos que pueden ser ejecutados en modo directo o editor, además se presentaron dos métodos de programación con los que puede ser programado el robot Scorbot ER-VII, en este caso como pudo observarse el método mas utilizado en este trabajo es la programación textual, es decir que se realizaron programas los cuales contienen instrucciones que permiten calcular posiciones y mover al robot, en este caso se lograron realizar satisfactoriamente cuatro programas que consistieron en ejecutar dos trayectorias: una parabólica y otra sinusoidal, los puntos que integran estas trayectorias son alcanzables y admisibles para el robot manipulador Scorbot ER-VII.

Conclusiones generales

Después leer el presente trabajo, la persona tendrá los conocimientos acerca de los acontecimientos mas sobresalientes en el desarrollo de la robótica, así como los elementos constitutivos de los robots industriales, los componentes de las celdas de manufactura, y de los distintos lenguajes de programación utilizados en los robots industriales, también las instrucciones básicas para la elaboración de un programa para la realización de una tarea de un robot industrial en lenguaje ACL.

GLOSARIO DE TÉRMINOS

Articulación prismática (Lineal): las articulaciones prismáticas implican un movimiento deslizando o de traslación de las uniones de conexión.

Articulación de revolución: tipo de articulación giratoria en la que la unión de entrada es paralela al eje de rotación y la de salida es perpendicular a dicho eje. Esencialmente, la unión de salida gira alrededor de la de entrada como si estuviera en órbital.

Articulación de rotación: suministra un grado de libertad consistente en una rotación alrededor el eje de la articulación. Está articulación es, con diferencia, la mas empleada.

Autómata: palabra que procede del griego (automatos = actuar por sí mismo). Es un mecanismo artificial que imita comportamientos de la vida real.

Brazo (Robot): una de las partes del manipulador. Soportado en la base de éste, sostiene y maneja la muñeca.

Cadena cinemática: es un ensamble de eslabones y juntas interconectados de modo que proporcionen un movimiento de salida controlado en respuesta a un movimiento de entrada proporcionado.

Control: selección de las entradas de un sistema de manera que los estados o salidas cambien de acuerdo a una manera deseada.

Dos tipos de control pueden ser usados en diferentes partes del robot: Lazo cerrado: se monitorea continuamente la posición del brazo del robot mediante un sensor de posición, y se modifica la energía que se manda al actuador de tal forma que el movimiento del brazo recorre el camino deseado, tanto en dirección como en velocidad. Éste control se puede usar cuando la tarea que se ha de llevar a cabo está dirigida mediante un camino definido por la misma pieza, tal como sería soldar, revestir y ensamblar.

En un sistema de lazo abierto: el controlador no conoce la posición de la herramienta mientras el brazo se mueve de un punto a otro. Éste tipo de control es muy usado cuando el movimiento que debe seguir el brazo se encuentra determinado previamente, al ser grabado con anterioridad y reproducido sin cambio alguno, lo cual es útil cuando todas las piezas a ser tratadas son exactamente iguales.

Entorno: estado de la naturaleza o estado del mundo, concepto útil en la programación. Conjunto de todos los factores externos o fuerzas que están mas allá de la influencia de la persona que toma una decisión, pero que no obstante afectan las consecuencias de sus acciones.

Espacio singular: se denominan configuraciones singulares de un robot a aquellas en el que el determinante de su matriz Jacobiana se anula. Por esta circunstancia, en las configuraciones singulares no existe jacobiana inversa. Al anularse el Jacobiano, un incremento infinitesimal de las coordenadas cartesianas supondría un incremento infinito de las coordenadas articulares, lo que en la práctica se traduce en que las inmediateces de las configuraciones singulares, el pretender que el extremo del robot se mueva a velocidad constante, obligaría a movimientos de las articulaciones a velocidades inabordable por sus actuadores.

Espacio de trabajo: las dimensiones de los elementos del manipulador, junto a los grados de libertad, definen el espacio de trabajo del robot. Se subdivide en áreas diferenciadas entre sí, por la accesibilidad específica del elemento Terminal (aprehensor o herramienta), es diferente a la que permite orientarlo verticalmente o con el determinado ángulo de inclinación. También queda restringida el espacio de trabajo por los límites de giro y desplazamiento que existen en las articulaciones. El espacio de trabajo puede ser explicado, como un espacio de configuraciones y consiste en el conjunto de todas las posiciones y orientaciones que el robot puede tomar. Obtener el espacio de configuraciones es equivalente a que el robot recorra y se posicione en todos los puntos posibles del espacio de trabajo.

Interfaz: conexión mecánica o eléctrica que permite el intercambio de información entre dos dispositivos o sistemas. Habitualmente se refiere al “software” y “hardware” necesarios para unir dos elementos de proceso en un sistema o bien para describir los estándares recomendados para realizar dichas interconexiones. También se conoce por su denominación inglesa: “interface”.

Kan- Ban: se define como “Un sistema de producción altamente efectivo y eficiente”, significa en japonés: “etiqueta de instrucción”. Su principal función es ser una orden de trabajo, es decir, un dispositivo de dirección automático que nos da información de lo que se va a producir, en que cantidad, mediante que medios y como transportarlo. Tiene como funciones principales el control de la producción y la mejora de procesos y el desarrollo de un sistema JIT:

Manipulador: en general, cualquier dispositivo mecánico capaz de reproducir los movimientos humanos para el manejo de objetos. En particular, suele referirse a los elementos mecánicos de un robot que producen su adecuado posicionamiento y operación.

Órgano Terminal: éste término se utiliza para describir la mano o herramienta que está unida a la muñeca. Representa el herramental especial que permite al robot de uso general realizar una aplicación particular.

Pitch: Elevación del órgano terminal.

Robot: es un sistema mecánico cuyo control debe estar encaminado a efectuar una tarea de utilidad.

Robot industrial: es un manipulador multifuncional reprogramable diseñado para desplazar materiales, piezas, herramientas o dispositivos especiales mediante movimientos programados variables para la ejecución de una diversidad de tareas.

Robot manipulador: Los manipuladores son brazos articulados con un número de grados de libertad, que oscila entre dos y cinco, cuyos movimientos son de tipo secuencial, se programan mecánicamente o a través de una memoria o de un controlador lógico programable. No permiten la combinación simultánea de movimientos ni el posicionamiento de su elemento terminal.

Robusto: en control, poco sensible a errores o incertidumbres de modelado. En la actividad computacional, tanto una máquina como un algoritmo se consideran robustos si son capaces de actuar y trascender a pesar de dificultades que accidentarían a máquinas o algoritmos más simples o menos flexibles. La tolerancia a las fallas o fallos se considera sinónimo de robustez.

Roll: Giro del órgano terminal.

Sensor: es un transductor que se utiliza para medir una variable física de interés.

Servomecanismo: sistema automático de control de movimiento que usa la realimentación para controlar una salida deseada como posición, velocidad o aceleración

Sistema de control: sistema diseñado para lograr el comportamiento deseado de un sistema dinámico. Puede ser digital (microprocesador) o analógico (por ejemplo, circuitos eléctricos, sistemas hidráulicos, sistemas mecánicos, etc.), manual o automático.

Sistema Dinámico: sistema con un número finito de grados de libertad y que puede ser modelado matemáticamente por medio de ecuaciones diferenciales que dependen del tiempo.

Transmisión: las transmisiones son los elementos encargados de transmitir el movimiento desde los actuadores hasta las articulaciones.

Transmisión mecánica: incrementa la fuerza y el par de salida. Estas transmisiones pueden ser engranes, tornillos sin fin, impulsores armónicos, bandas, cadenas, etc.

Trayectoria de un robot: para realizar una tarea determinada el robot debe moverse desde un punto inicial a la final. Este movimiento puede ser realizado según trayectorias infinitas espaciales.

Variable articular: valor que puede adquirir el ángulo de la articulación.

Yaw: Desviación del órgano terminal.

REFERENCIAS

- [1] Baumgartner, Knischewski, weiding, “CIM Consideraciones Básicas”, Marcombo.
- [2] Barry Render, Jay Heizer “Principios de Administración de Operaciones”, Prentice Hall.
- [3] Chiles, Black, Lissaman, Martin, “Principios de ingeniería de manufactura” editorial CECSA.
- [4] Craig. John J., “Robótica”, Prentice Hall
- [6] Daniel Audi Piera, “Cómo y cuándo aplicar un Robot Industrial”, Marcombo Boixareu Editores
- [7] Dilep R. Sule, “Instalaciones de Manufactura Ubicación, Planeación y Thomson Learning.
- [8] Ferraté, G., AMAT, J., AYSA, J., Y OTROS. “*Robótica Industrial.*” Barcelona: Marcombo, S.A., 1988. 382p.
- [9] Ferré R. “La fabrica flexible”, Marcombo, Barcelona, España, 1988
- [10] Gary Cornell, “Manual de Visual Basic 4 para Windows 95”, McGraw-Hill, 1996, México.
- [11] Groover M. “Fundamentos de Manufactura Moderna”, Prentice Hall, México.
- [12] Hal Mather, “Manufactura Competitiva”, Editorial Ventura
- [13] Ian Sommerville, “Ingeniería de Software”, Addison Wesley, 1998, E.U.A.
- [14] Leo Alting “Procesos para la Ingeniería de manufactura”.
- [15] *James A. Senn, “Análisis y Diseño de Sistemas de Información”, McGraw 1998, México.*
- [16] Roger G. Schroeder , “Toma de Decisiones en Función de las Operacones”. Mc. Graw Hill

REFERENCIAS ELECTRONICAS

- [17]* www.androidewolld
- [18]* [www.chi.itesm.mx/ cim/robind/robotica.html](http://www.chi.itesm.mx/cim/robind/robotica.html)
- [19]* www.gda.itesm.mx/csim/archivos/celda.swf
- [20]* www.kawasiki
- [21]* www.roboticspot.com/spot/asifue/his2004b.html
- [22]* www.stafordt.edu/