



Universidad Veracruzana

UNIVERSIDAD VERACRUZANA

FACULTAD DE INSTRUMENTACIÓN ELECTRÓNICA
Y CIENCIAS ATMOSFÉRICAS

DOCUMENTACIÓN Y PROGRAMACIÓN DE PRÁCTICAS CON EL ROBOT INDUSTRIAL RV-2AJ ESTACIÓN 6 DEL LABORATORIO MÓVIL DE AUTOMATIZACIÓN Y ROBÓTICA DE LA UNIVERSIDAD VERACRUZANA

T E S I N A

Que para obtener el Título de:

**INGENIERO EN INSTRUMENTACIÓN
ELECTRÓNICA**

Presentan:

**HERRERA FLORES LAURA GUADALUPE
LÓPEZ CASTELLANOS ADRIANA**

Director de Tesina:

M. I. A. Leticia Cuéllar Hernández

Co-Director de Tesina:

M. C. Jesús Sánchez Orea

Xalapa Ver.

Septiembre 2009

ÍNDICE

<i>Introducción</i>	5
CAPÍTULO I. ROBOT INDUSTRIAL	6
1.1 Definición de un robot industrial.....	6
1.2 Componentes.....	7
1.3 Morfología del robot manipulador	8
1.3.1 Tipos de articulaciones	8
1.3.2 Estructuras básicas.....	9
1.3.3 Efectores finales	12
1.3.4 Grados de libertad.....	12
1.3.5 Capacidad de carga.....	13
1.3.6 Espacio de trabajo.....	13
1.3.7 Exactitud, repetibilidad y resolución espacial	13
1.3.8 Velocidad	14
1.3.9 Sensores	14
1.3.9.1 Sensores internos	14
1.3.9.2 Sensores externos	17
1.4 Lenguajes de programación de robot manipuladores	20
1.4.1 Programación por guiado.....	20
1.4.2 Programación textual.....	21
1.4.3 Programación fuera de línea.....	23
CAPÍTULO II. ROBOT INDUSTRIAL RV-2AJ	24
2.1 Medidas de seguridad.....	24
2.2 Componentes.....	25
2.2.1 Brazo robótico RV-2AJ.....	26
2.2.2 Controlador CR1-571	26
2.2.3 T/B o caja de aprendizaje	28
2.2.4 Pinza	29
2.2.5 Módulos.....	30
2.3 Modos de operación JOG	31
2.3.1 Modo de operación JOINT	31
2.3.2 Modo de operación TOOL.....	32
2.3.3 Modo de operación XYZ.....	33
2.3.4 Modo de operación 3-EJES XYZ.....	35
2.3.5 Modo de operación CYLINDER.....	36

2.4 Posición inicial.....	37
2.5 Sensores	38
2.6 Alimentaciones.....	38
2.7 Limpieza	39

CAPÍTULO III. PROGRAMACIÓN.....40

3.1 COSIMIR ® Industrial 4.1	40
3.1.1 Nuevo proyecto	41
3.1.2 Barra de herramientas.....	43
3.1.3 Toma de posiciones.....	44
3.1.4 Descarga de los archivos .POS y .MB4 al controlador CR1-571.....	46
3.1.5 Ejecución del programa con el brazo robótico RV-2AJ	48
3.1.6 Monitor de entradas y salidas (I/O)	48
3.2 Lenguaje de programación MELFA BASIC IV	49
3.2.1 Instrucciones de movimiento.....	49
3.2.1.1 Interpolación angular (MOV).....	49
3.2.1.2 Interpolación lineal (MVS).....	50
3.2.1.3 Interpolación circular (MVR y MVC).....	52
3.2.1.4 Movimiento continuo (CNT).....	53
3.2.2 Instrucciones de velocidad y aceleración/desaceleración.....	54
3.2.3 Instrucciones de control de la pinza	56
3.2.4 Instrucciones de paletizado.....	58
3.2.5 Estructuras de control	60
3.2.6 Repeticiones	61
3.2.7 Interrupción	62
3.2.8 Subrutina.....	62
3.2.9 Temporizador.....	63
3.2.10 Suspensión.....	63
3.2.11 Entrada y salida de señales externas.....	64
3.2.12 Expresiones y operaciones	64
3.2.13 Líneas de instrucción.....	67
3.3 Pasos para la elaboración de prácticas con el robot industrial RV-2AJ.....	68

CAPÍTULO IV. PRÁCTICAS.....71

Práctica 1: Interpolación angular (MOV).....	72
Práctica 2: Interpolación lineal (MVS).....	74
Práctica 3: Interpolación circular (MVR y MVC).....	76
Práctica 4: Selección de piezas por color negro o no negro	78
Práctica 5: Paletizado tipo1 (PLT1: zig-zag).....	81
Práctica 6: Paletizado tipo2 (PLT2: mismo sentido)	83
Práctica 7: Paletizado tipo3 (PLT3: arco).....	85

Conclusiones	87
APÉNDICE A: CARACTERÍSTICAS GENERALES DEL ROBOT INDUSTRIAL RV-2AJ.....	88
APÉNDICE B: CARACTERÍSTICAS GENERALES DEL CONTROLADOR CR1-571	89
APÉNDICE C: INSTALACIÓN DEL PROGRAMA Y CONFIGURACIÓN DE LA COMUNICACIÓN.....	90
APÉNDICE D: LISTA DE ERRORES	95
APÉNDICE E: CAMBIO DE BATERÍAS	97
APÉNDICE F: CONFIGURACIÓN DEL ORIGEN DE LOS EJES	100
Bibliografía	106

INTRODUCCIÓN

Esta tesina centra su estudio en la necesidad de adquisición de conocimientos para el manejo de un robot manipular industrial, debido a que en la actualidad la tecnología juega un papel muy importante.

Uno de los sectores en donde se emplean con más frecuencia el uso de robots, es en el sector industrial, ya que estos robots tienen como propósito el de liberar al hombre de tareas peligrosas, tediosas o pesadas ejecutando los trabajos con mayor precisión y con una velocidad considerable sin agotamiento o cansancio.

Los tipos de robots que se encargan de realizar tareas vinculadas con la producción de bienes de origen manufacturado o industrias, se le conocen como robots industriales, los más comunes son los robots utilizados para realizar las tareas de soldadura, carga y descarga, ensamblaje, aplicación de pintura e inspección de productos. En este trabajo se estudia el robot industrial RV-2AJ, manipulador industrial que cuenta con cinco grados de libertad.

Algunos de los motivos que han llevado a la elaboración de este trabajo, es por la oportunidad de tener acceso al robot industrial, así como también, la oportunidad de compartir los conocimientos adquiridos de operabilidad del robot a los estudiantes interesados presentando la información de una manera resumida y completa, pues debido a las limitantes de tiempo y espacio, el uso del robot se hace un poco más complicado sin algún tipo de apoyo de esta índole.

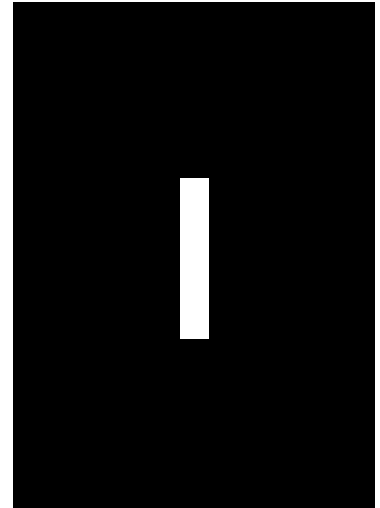
Para facilitar el aprendizaje, este trabajo está dividido en cuatro capítulos:

El primer capítulo tiene como objetivo dar a conocer las características generales de un robot manipulador industrial cualquiera, es decir, nos describe cuales son los componentes que lo forman, a que le llaman grados de libertad, los movimientos más comunes que pueden realizar, las diferentes morfologías que pueden tener, los tipos de sensores que usualmente utilizan, así como también, describe la clase de programación en las que el robot manipulador industrial puede ser programado.

El segundo capítulo comienza haciendo hincapié en el tema de medidas de seguridad, las cuales se deben tomar antes de utilizar el robot industrial RV-2AJ, con el fin de evitar accidentes o pérdidas. También da a conocer las características de este robot industrial como sus componentes, parámetros de inicialización, modos de operación JOG y tipos de módulos.

El tercer capítulo justifica las características del software COSIMIR ®INDUSTRIAL 4.1, que se utiliza para la programación de prácticas, así como, el lenguaje de programación MELFA BASIC IV el cual explica los diferentes comandos que se pueden usar, finalizando con la descripción de pasos para la elaboración de estas.

El cuarto y último capítulo presentan algunas prácticas realizadas con el brazo robótico RV-2AJ, se describe el objetivo, las posiciones y el código del programa, con el fin de guiar al estudiante en sus prácticas.



ROBOT INDUSTRIAL MANIPULADOR

1.1 Definición de un robot industrial

Antes de mencionar el concepto de robot industrial es importante saber que un robot no es simplemente una máquina, pues un robot es capaz de realizar tareas y al mismo tiempo modificarlas por esta razón los robots se convierten en una opción ideal para el ser humano y como consecuencia para la industria.

El Instituto Norteamericano de Robótica realiza la siguiente definición: “Un robot es un manipulador multifuncional reprogramable diseñado para mover materiales, piezas, herramientas o dispositivos especiales mediante movimientos programados y variables que permiten realizar diversas tareas”¹.

Mientras que la IFR (Federación Internacional de Robótica), hace la siguiente distinción entre robot industrial y otros robots: Por robot industrial de manipulación se entiende a una máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que pueden posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento.

¹ Anibal Ollero, “Robótica. Manipuladores”, p. 5.

1.2 Componentes

Los robots industriales manipuladores no son capaces de iniciar ninguna acción por su cuenta, están constituidos por sensores que reciben datos de entrada para realizar secuencias, las cuales son determinadas a través de un programa, permitiéndoles la ejecución de sus movimientos y la memorización de diversas sentencias de trabajo. En consecuencia, estos tienen la opción de adaptación a distintas tareas y diferentes medios de trabajo.

A continuación se describen brevemente las partes que compone a un robot industrial manipulador como la unidad de control, la herramienta, la fuente de poder y el manipulador o brazo mecánico, como se conoce comúnmente.

- › **Unidad de control:** También conocido como controlador se encarga de dirigir los diferentes elementos del robot, la comunicación interna y externa, además de almacenar las trayectorias generadas por el robot manipulador.
- › **Manipulador o brazo mecánico:** Es el responsable de los movimientos necesarios para llevar a cabo una tarea. El brazo está constituido por una serie de eslabones unidos mediante articulaciones. De forma mas precisa, un manipulador industrial convencional es una cadena cinemática abierta formada por un conjunto de eslabones o elementos de la cadena interrelacionados mediante articulaciones. Las articulaciones se enumeran comenzando de la articulación más cercana a la base del manipulador al efector final como se muestra en la figura 1.1.

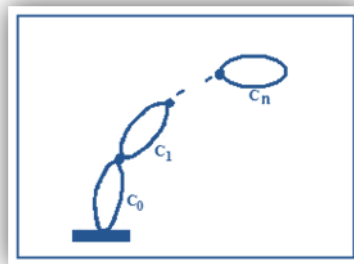


Figura 1.1: Modelado de un manipulador industrial.

- › **Herramienta o efector final:** El brazo mecánico necesita de una herramienta en el último enlace para poder realizar una tarea, se coloca un órgano terminal o efector final como una pinza o un dispositivo especial para realizar operaciones.
- › **Fuente de poder:** La fuente de poder puede ser: eléctrica, neumática o hidráulica. La unidad de control funciona con energía eléctrica, pero el brazo mecánico y la herramienta pueden utilizar energía neumática (gas comprimido) o incluso energía hidráulica (fluidos).
- › **Dispositivos de entrada y salida:** Los dispositivos de entrada y salida permiten mandar instrucciones al controlador, dar de alta programas de control además de ver los datos del controlador, respectivamente. Los dispositivos más comunes son: teclado, monitor y caja de comandos también conocido como teach pendant o caja de aprendizaje, este último dispositivo es el más destacado debido a que enseña las posiciones al robot manipulador.

Es necesario aclarar que algunos robots únicamente poseen uno de estos componentes.

- › **Dispositivos especiales:** Aquí se localizan las estaciones de ensamblaje que son utilizadas para sujetar las diferentes piezas de trabajo, algunos dispositivos especiales son:
 - › Estación de posición sobre el transportador para la carga/descarga de piezas de trabajo.
 - › Eje transversal para aumentar el volumen de trabajo del robot.
 - › Estación de inspección por computadora integrada con el robot.
 - › Estación de ensamble.

1.3 Morfología del robot manipulador

Se describen las características más relevantes de la estructura de los robots industriales manipuladores, los tipos de articulaciones y configuraciones básicas de estos.

1.3.1 Tipos de articulaciones

Los robots industriales manipuladores son esencialmente brazos articulados, siendo una articulación, el medio de contacto que une dos eslabones próximos. Existen diversos tipos de articulaciones (figura 1.2), las más utilizadas son:

- › **Articulación de rotación:** Esta articulación es la más empleada (figura 1.2a), suministra un grado de libertad que consiste en una rotación alrededor del eje de la articulación.
- › **Articulación prismática:** Consta de un grado de libertad el cual consiste en una traslación a lo largo del eje donde se encuentra dicha articulación (figura 1.2b).
- › **Articulación cilíndrica:** Está constituido por dos grados de libertad: una de rotación y otra de traslación (figura 1.2c).
- › **Articulación planar:** Se caracteriza por el movimiento de desplazamiento en un plano, existiendo en él dos grados de libertad (figura 1.2d).
- › **Articulación esférica:** Combina tres giros en tres direcciones perpendiculares en el espacio (figura 1.2e).

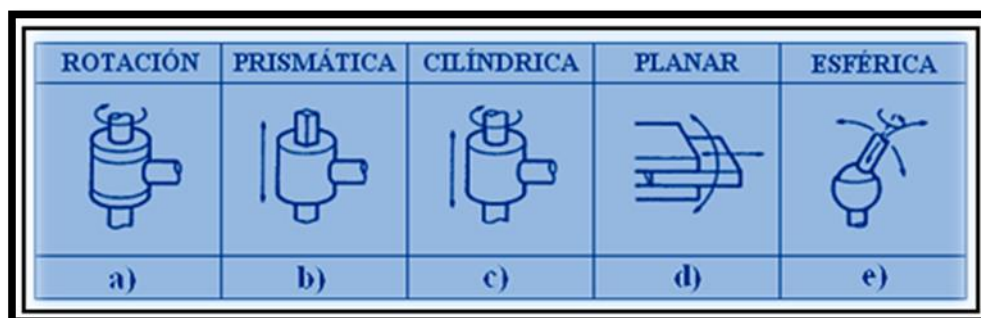


Figura 1.2: *Tipos de articulaciones robóticas.*

1.3.2 Estructuras básicas

La estructura típica de un manipulador, como se mencionó anteriormente consiste en un brazo compuesto por elementos con articulaciones entre ellos. En el último enlace se coloca un dispositivo especial para realizar operaciones².

Estas estructuras tienen diferentes propiedades en cuanto a espacio de trabajo y accesibilidad a posiciones determinadas, es decir, de acuerdo a la estructura del manipulador se proporciona una determinada configuración, que da origen a los parámetros que se tienen que establecer para determinar la posición y orientación del elemento final.

Un robot industrial manipulador puede presentar cualquiera de las 5 estructuras básicas que se presentan en la figura 1.3.



Figura 1.3: *Tipos de configuraciones de robots industriales manipuladores.*

- **Cartesiana:** 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 también conocida como interpolación angular.

En la figura 1.4 se muestra de manera básica un robot con estructura cartesiana, donde: a) representa los tipos de movimientos y articulaciones que posee, b) representa los movimientos posibles para ubicar un punto en el espacio y el c) representa el alcance o espacio de trabajo máximo que puede llegar a tener el robot.

² Aníbal Ollero, “Robótica. Manipuladores”, p. 18.

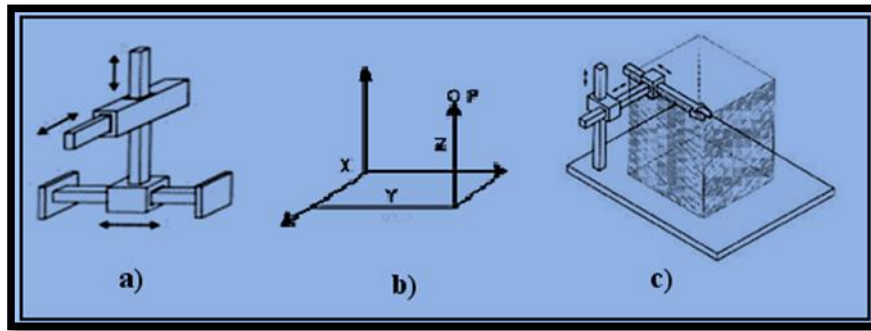


Figura 1.4: Estructura cartesiana.

- **Cilíndrica:** Posee tres grados de libertad: dos movimientos lineales y uno rotacional. El robot de configuración cilíndrica está diseñado para ejecutar los movimientos conocidos como interpolación lineal e interpolación por articulación.

En la figura 1.5 se muestra de manera básica un robot con estructura cilíndrica, donde: a) representa los tipos de movimientos y articulaciones que posee, b) representa los movimientos posibles para ubicar un punto en el espacio y el c) representa el alcance o espacio de trabajo máximo que puede llegar a tener el robot.

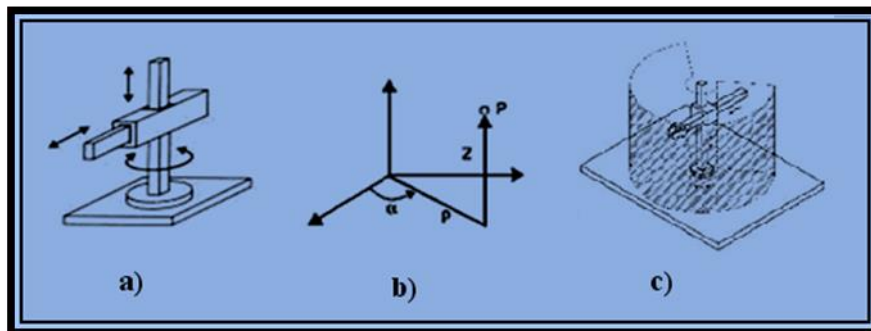


Figura 1.5: Estructura cilíndrica.

- **Esférica:** Tiene tres articulaciones. Cada una de ellas puede realizar un movimiento distinto: 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.

En la figura 1.6 se muestra de manera básica un robot con estructura esférica, donde: a) representa los tipos de movimientos y articulaciones que posee, b) representa los movimientos posibles para ubicar un punto en el espacio y el c) representa el alcance o espacio de trabajo máximo que puede llegar a tener el robot.

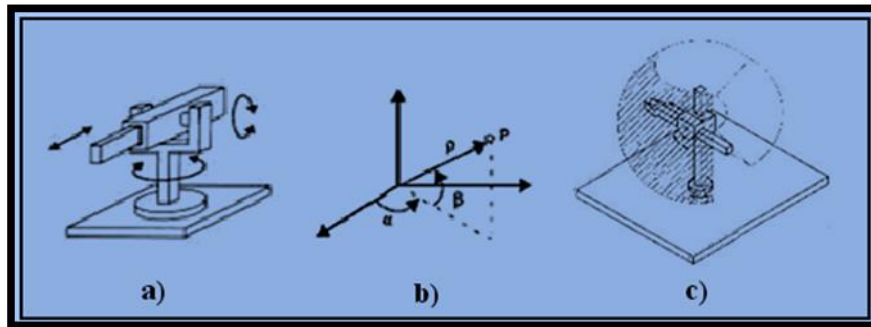


Figura 1.6: Estructura esférica.

- **Angular:** 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. Se le conoce como brazo articulado por su similitud con el brazo humano.

En la figura 1.7 se muestra de manera básica un robot con estructura angular, donde: a) representa los tipos de movimientos y articulaciones que posee, b) representa los movimientos posibles para ubicar un punto en el espacio y el c) representa el alcance o espacio de trabajo máximo que puede llegar a tener el robot.

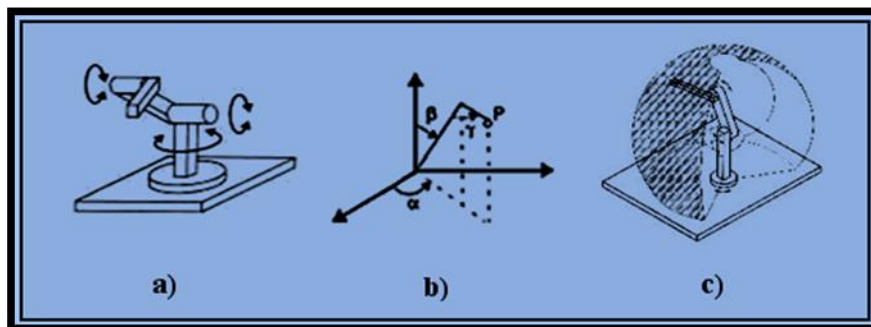


Figura 1.7: Estructura angular.

- **SCARA:** Selective Compliance Assembly Robot Arm, o bien, brazo selectivo del robot de montaje obediente, este brazo puede realizar movimientos horizontales de mayor alcance debido a sus dos articulaciones rotacionales. Este tipo de robot puede hacer un movimiento lineal mediante su tercera articulación.

En la figura 1.8 se muestra de manera básica un robot con estructura SCARA, donde: a) representa los tipos de movimientos y articulaciones que posee, b) representa los movimientos posibles para ubicar un punto en el espacio y el c) representa el alcance o espacio de trabajo máximo que puede llegar a tener el robot.

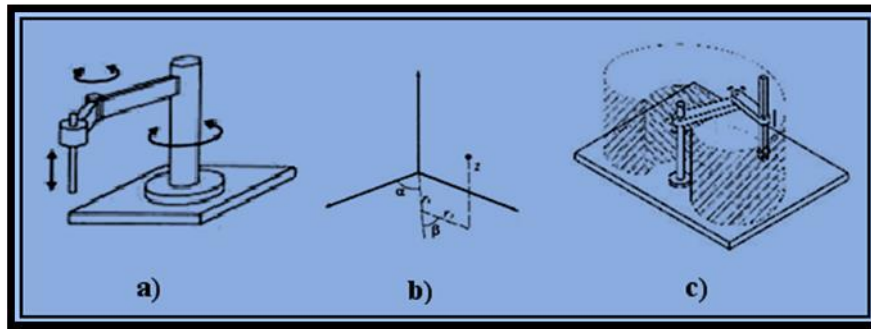


Figura 1.8: Estructura SCARA.

1.3.3 Efectores finales

Es la herramienta que se coloca en el extremo del último enlace del manipulador y que suministra la capacidad de agarre del objeto que se pretende manipular, el tipo de herramienta depende de la tarea que el robot tenga programada.

Las características que hay que tener en cuenta para su diseño son: capacidad de carga, fuerza de agarre, geometría y dimensiones de los objetos que se debe manejar, tipos de movimientos que puede realizar, alimentación (neumática, eléctrica, hidráulica), tiempo de actuación del mecanismo de agarre y características de la superficie de contacto³.

Los efectores finales más simples son pinzas mecánicas, típicamente con dos dedos y accionamiento neumático todo/nada. Se emplean también acondicionamientos eléctricos con control proporcional, existen también dedos con material deformable para evitar que se produzca el corrimiento de la pieza. Asimismo, es posible utilizar sensores táctiles en los dedos empleando un bucle de control del esfuerzo de agarre. Se dispone también de pares de dedos con diferentes cavidades para piezas de distintos tamaños. Otra técnica clásica de agarre es la sujeción por succión. Se emplean ventosas elásticas para manipular piezas rígidas y ventosas rígidas para manipular piezas elásticas.

1.3.4 Grados de libertad

Los grados de libertad definen al robot pues es equivalente a decir el número y tipo de movimientos del manipulador. Viendo los movimientos del brazo y de la muñeca se puede determinar el número de grados de libertad que presenta el robot. Por lo general, tanto en el brazo como en la muñeca, se encuentra una variación que va desde uno hasta los tres grados de libertad.

A la muñeca de un manipulador le corresponden los siguientes movimientos o grados de libertad: giro, elevación y desviación, como lo muestra la figura 1.9.

³ Aníbal Ollero, "Robótica. Manipuladores", p. 24.

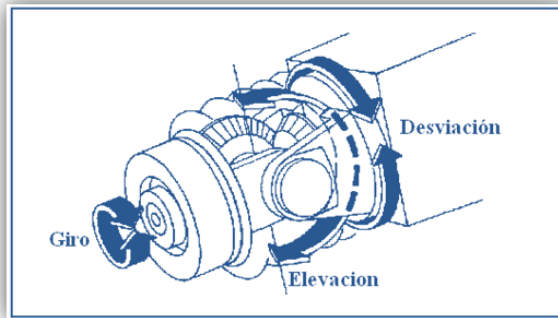


Figura 1.9: Grados de libertad de la muñeca de un robot manipulador.

Cabe mencionar que existen muñecas que no pueden realizar los tres tipos de movimientos citados anteriormente. Los grados de libertad del brazo están directamente relacionados con su configuración.

1.3.5 Capacidad de carga

La capacidad de carga es el peso en kilogramos que puede transportar la pinza del manipulador. Esta varía de acuerdo al tipo de robot manipulador pues viene condicionada por el tamaño, la configuración y el sistema de accionamiento del propio robot por ello el fabricante del robot debe proporcionar estos datos.

1.3.6 Espacio de trabajo

El espacio de trabajo del robot se refiere a la zona dentro de la cual puede desplazarse el extremo de la muñeca, esto depende de las dimensiones de los elementos del manipulador junto con los grados de libertad. Para determinar el espacio de trabajo no se toma en cuenta el efector final ya que a la muñeca del robot se le pueden adaptar efectores finales de diversos tamaños de acuerdo a la tarea que llevará a cabo.

1.3.7 Exactitud, repetibilidad y resolución espacial

La precisión de movimiento en un robot manipulador se relaciona con tres factores importantes: exactitud, repetibilidad y resolución espacial.

- › **Exactitud:** Es la capacidad de un robot para situar el extremo de su muñeca en un punto señalado dentro de la zona de trabajo, mide la distancia entre la posición especificada y la posición real del efector final del robot. Mantiene una relación directa con la resolución espacial.

- › **Resolución espacial:** Es el incremento más pequeño de movimiento en donde el robot puede dividir su volumen de trabajo.
- › **Repetibilidad:** Es 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.

1.3.8 Velocidad

La velocidad del robot depende de la tarea que desempeñe, por ejemplo, en tareas de soldadura y manipulación de piezas, la velocidad de trabajo es alta; mientras que en pintura, mecanizado y ensamblaje, la velocidad debe ser media e incluso baja.

1.3.9 Sensores

En la actualidad los robots tienen la necesidad de interactuar con el medio para llevar a cabo sus tareas, los sensores juegan un papel muy importante porque con ellos se conocen los parámetros como la posición, la velocidad y el color.

Algunas características principales de los sensores que se deben tener en cuenta, son las siguientes:

- › **Exactitud:** Hace referencia a la detección del valor verdadero de la variable sin errores sistemáticos. La media de los errores cometidos debe tender a cero.
- › **Precisión:** Una medida será más precisa que otra si los posibles errores aleatorios en la medición son menores. Se debe procurar la máxima precisión posible.
- › **Rango de funcionamiento:** Es la capacidad de medir de manera exacta y precisa un amplio rango de valores de la magnitud correspondiente.
- › **Velocidad de respuesta:** El sensor debe responder a los cambios de la variable a medir en un tiempo mínimo.
- › **Calibración:** Es el proceso mediante el cual se establece la relación entre la variable medida y la señal de salida que produce el sensor. La calibración debe poder realizarse de manera sencilla y además el sensor no debe precisar una recalibración frecuente.
- › **Fiabilidad:** El sensor no debe estar sujeto a fallos inesperados durante su funcionamiento.
- › **Facilidad de funcionamiento:** Sería ideal que la instalación y uso del sensor no necesitara de un aprendizaje excesivo.

Una clasificación general que engloba a los sensores en dos grandes grupos, según la relación de la variable a medir con el sensor son: internos y externos.

1.3.9.1 Sensores internos

Los sensores internos permiten que el robot tenga información relacionada con su propio

estado. En la tabla 1.1, se muestran las tres categorías posibles de clasificación de los sensores internos.

SENSORES INTERNOS		
Posición	Velocidad	Aceleración
<ul style="list-style-type: none"> • Decodificador incremental • Decodificador absoluto • Potenciómetro • LVDT • Inclínómetro • Sistemas basado en faros • GPS 	<ul style="list-style-type: none"> • Tacogenerador • Sensor de efecto Doppler • LVT 	<ul style="list-style-type: none"> • Servo-acelerómetro • Acelerómetro piezo-resistivo

Tabla 1.1: Clasificación de los sensores internos.

1. Sensores de posición: Determinan la posición del robot en un instante determinado. No existe un método garantizado y universal para calcular la posición pero existen diversas técnicas que intentan resolver el problema, por ejemplo, los siguientes:

- **Decodificadores incrementales:** Se utilizan fundamentalmente para el cálculo de la posición angular. Básicamente constan de un disco transparente, el cual tiene una serie de marcas opacas colocadas radialmente y equidistantes entre si, el disco separa un elemento emisor de luz y un elemento fotosensible que actúa como receptor, provocando la lectura de una serie de pulsos de luz, correspondientes a la luz que atraviesa los huecos entre las marcas. Llevando una cuenta de esos pulsos es posible conocer la posición del eje. Realmente los decodificadores incrementales miden la velocidad de giro, pero podemos extrapolar la posición angular. Como es lógico, la resolución de este tipo de sensores depende directamente del número de marcas que podamos poner físicamente en el disco.
- **Decodificadores absolutos:** Miden la posición angular, en este caso lo que se va a medir no es el incremento de esa posición, sino la posición exacta. La estructura es parecida a la de los decodificadores incrementales, la diferencia estriba en la graduación o codificación del disco. En este caso el disco se divide en un número fijo de sectores (potencia de 2) y se codifica cada uno con un código cíclico (normalmente con código Gray); este código queda representado en el disco por zonas transparentes y opacas colocadas radialmente. La codificación de los distintos sectores angulares es absoluta.
- **Potenciómetros:** Miden la posición angular y pequeños desplazamientos de posición lineal, constan de una resistencia a través de la cual hay una diferencia de potencial, además de un contacto unido a la resistencia, el cual se puede deslizar a su alrededor; este elemento es conocido como wiper. El wiper se conecta físicamente al elemento cuyo movimiento vamos a medir, cuando este elemento se mueva, la tensión de salida irá cambiando. Midiendo esta tensión de salida se determinará el desplazamiento del elemento que se pretende controlar.
- **LVDT (Transformador Diferencial de Variación Lineal):** Sensor electro-magnético. Es un tipo de transformador eléctrico utilizado para medir desplazamientos lineales. El

transformador posee tres bobinas solenoidales situadas extremo con extremo alrededor de un tubo. La bobina central es el devanado primario y las externas son los secundarios. Un centro ferromagnético de forma cilíndrica, sujeto al objeto cuya posición desea ser medida, se desliza con respecto al eje del tubo. A medida que el núcleo se mueve, la inductancia mutua cambia, causando que el voltaje inducido en el secundario cambie. Cuando el núcleo es desplazado en una dirección, el voltaje en una bobina aumenta mientras que en la otra disminuye, causando que el voltaje de salida también aumente desde cero hasta su máximo, por lo tanto, la diferencia de potencial será proporcional al movimiento del núcleo. Este sistema se utiliza ampliamente debido a su gran resolución, alta linealidad y rápida respuesta. Sin embargo, no permite medir grandes desplazamientos.

- **Giroscopios:** El giroscopio es un dispositivo mecánico formado esencialmente por un cuerpo con simetría de rotación que gira alrededor de su eje de simetría. Cuando se somete el giroscopio a un momento de fuerza que tiende a cambiar la orientación del eje de rotación su comportamiento en lugar de cambiar de dirección, cambia de orientación en una dirección perpendicular a la dirección "intuitiva". Basándose en este principio, los giroscopios son capaces de medir el ángulo de giro de un objeto.
- **Inclinómetros:** Estos sensores sirven para medir la inclinación, el ángulo de un objeto con respecto a un eje horizontal. Están formados por un electrolito (líquido conductor) situado en un recipiente en el cual se encuentran dos electrodos de platino enfrentados y ambos con una parte fuera del electrolito. Cuando el sensor se inclina, uno de los electrodos entra más en contacto con el electrolito y el otro menos. Si se miden las corrientes de salida de los electrodos, es posible determinar el ángulo de inclinación.
- **Sistemas basados en faros:** Estos sistemas están orientados a conocer la posición de un robot móvil en un sistema de coordenadas. El principio básico es similar al de los faros usados en navegación marítima, consiste en situar una serie de puntos de referencia que el robot pueda consultar en cualquier momento, y así se pueda calcular su posición. El tipo de señal que emiten esos puntos de referencia o "faros" puede ser de muchos tipos, como láser, ultrasonido o radiofrecuencia, estas suelen emplearse en sistemas reales.
- **GPS (Sistema de Posicionamiento Global):** El sistema se basa en una constelación de 24 satélites geoestacionarios, con una frecuencia de órbita de 12 horas y situados a una altura de 10.900 millas náuticas. Para poder usar este sistema de medida se necesita un elemento receptor. Realmente lo que se calcula es la posición con este procedimiento: el receptor mide el tiempo de vuelo de las señales que le llegan de los distintos satélites y por triangulación es capaz de deducir su posición exacta en términos de longitud, latitud y altitud. Este sistema de medida puede tener una precisión en centímetros, pero la posibilidad de ruido y el tiempo que transcurre en todo el proceso, hace que no sea un método adecuado para su uso en robots móviles que se desenvuelven en entornos más bien reducidos en comparación con la Tierra.

2. Sensores de velocidad: La velocidad es otro de los parámetros internos del robot que se utiliza para el desarrollo de su tarea. Un método se basa en la medida de la posición, conociendo la posición del robot podemos derivar la velocidad aplicando directamente su definición, es decir, incremento de posición dividido entre el tiempo que tarda en realizar el cambio de posición.

- **Tacogeneradores:** Son dispositivos para medir la velocidad angular, también conocidos

como dínamo taquimétrico. Su funcionamiento es sencillo: convertir la energía rotacional en energía eléctrica proporcional a la rotacional. Para generar la corriente a partir del giro se acopla al eje a medir una espira situada dentro de un campo magnético fijo (creado por dos imanes). Al girar el motor, la espira gira en el interior del campo magnético, lo que provoca una corriente eléctrica.

- **Sensores Doppler:** Los sensores basados en el efecto Doppler miden la velocidad lineal de un objeto móvil apoyándose en otra superficie. Se basan en la observación del desplazamiento en frecuencia de una radiación emitida por el sensor y reflejada en una superficie que se está moviendo con respecto al robot. Una vez conocida la velocidad de vuelta de la señal, mediante una relación trigonométrica se puede calcular la velocidad del móvil.
- **LVT (Transductor de Velocidad Lineal):** Se basan en un principio electromagnético similar a los LVDT. Los sensores LVT constan de un núcleo magnético permanente en forma de varilla; este núcleo es conectado al dispositivo cuya velocidad vamos a medir. Arriba y abajo de la varilla se sitúan dos espirales conductoras. Al girar el núcleo de imán se genera un cambio en el campo magnético esto implica en las espiras una diferencia de potencial, obteniendo así la velocidad del dispositivo.

3. Sensores de aceleración: La aceleración es una variable interna del robot cuyo valor es utilizado para aplicaciones bastante concretas; no obstante existen una serie de métodos y sensores para su cálculo.

La primera manera que se puede pensar para conocer la aceleración de un robot es derivarla de la velocidad, de forma análoga a como se puede conocer la velocidad a partir de la posición. Sin embargo, este sistema no suele aportar buenos resultados. Es por esto que también existen sensores especializados en el cálculo de la aceleración.

- **Servo-acelerómetro:** Este es un dispositivo para medir la aceleración angular. El dispositivo cuya aceleración de giro se medirá está conectado a un péndulo. Cuando gira dicho elemento el péndulo lo hace con él. Un sensor de posición capta el movimiento del péndulo y mediante un circuito electrónico se compara la señal del sensor de posición con una señal de referencia. Entonces un motor de rotación aplica una fuerza al péndulo determinada por ese circuito electrónico y que hace girar al péndulo en sentido opuesto al del elemento. La posición en la que se detiene el péndulo es proporcional a la aceleración inicial aplicada.
- **Acelerómetro piezo-resistivo:** Este dispositivo consta de una masa en forma de barra y dos medidores de tensión. La masa está introducida en un recipiente y tiene situados arriba y abajo los dos medidores. Cuando se produce una aceleración en el elemento a medir la pieza en forma de tabla se dobla y con los medidores de tensión se podrá calcular la aceleración que ha provocado esa torsión.

1.3.9.2 Sensores externos

Los sensores externos permiten al robot interactuar con su ambiente de una manera flexible.

Aunque muchos de los robots industriales manipuladores trabajan de una forma preprogramada, el uso de estos sensores como apoyo en la ejecución de tareas es cada día más amplio. Los sensores externos dan al robot mayor independencia del entorno concreto en el que se mueven, lo que se traduce en un mayor grado de "inteligencia". En la tabla 1.2, se muestran las tres categorías posibles de clasificación de los sensores externos.

SENSORES EXTERNOS		
Táctil <ul style="list-style-type: none"> • Contacto • Fuerza <ul style="list-style-type: none"> • Muñeca detectora de fuerza • Detección de articulaciones • Sensores de array táctil 	Presencia <ul style="list-style-type: none"> • Inductivos • Efecto Hall • Capacitivos • Ultrasónicos • Ópticos 	Alcance <ul style="list-style-type: none"> • Triangulación • Iluminación estructural • Tiempo de vuelo

Tabla 1.2: Clasificación de los sensores externos.

1. Sensores táctiles: Los sensores táctiles son dispositivos que indican el contacto de ellos contra un objeto sólido. Comúnmente se emplean en las pinzas de los robots para controlar la manipulación de objetos. Estos pueden dividirse en dos tipos: de contacto y de fuerza.

- **Sensores de contacto:** Estos sensores nos indican si ha habido contacto con algún objeto, sin medir la magnitud de la fuerza de contacto. Suelen ser dispositivos sencillos cuyo uso es muy variado. Se pueden situar en las pinzas de los robots para determinar cuando se ha tomado un objeto o una pieza de trabajo, ayudar a determinar las dimensiones de dichos objetos incluso pueden situarse en el exterior de las pinzas para ir investigando o palpando su entorno.
- Estos sensores suelen ser interruptores de límite o microinterruptores, que son sencillos dispositivos eléctricos que al hacer contacto con ellos cambian de estado.
- **Sensores de fuerza:** Los sensores de fuerza determinan el contacto y la magnitud de la fuerza aplicada, la cual se halla con las siguientes técnicas:
 - › **Muñeca detectora de fuerza:** Consta de una célula de carga que se sitúa entre la muñeca y las pinzas del brazo. Su objetivo es proporcionar información sobre las tres componentes de la fuerza (F_x , F_y , F_z) y sobre sus tres momentos en el extremo del brazo.
 - › **Detección de articulaciones:** Esta técnica se basa en la medida del par de torsión de la articulación. La medida de este par puede resultar sencilla, ya que es proporcional a la corriente que circula por el motor que provoca dicha torsión.
 - › **Sensores de array táctil:** Es un tipo especial de sensores de fuerza, está constituido por una matriz de pequeños sensores de fuerza. Debido a esta característica permiten

reconocer formas en los objetos que se están manipulando. Este tipo de dispositivos suelen montarse en las pinzas de los brazos de robot. Cada uno de los sensores de fuerza que componen la matriz suele ser una almohadilla elastomérica, que cuando se comprime cambia su resistencia eléctrica de manera proporcional a la fuerza aplicada. Midiendo esa resistencia se obtiene la información acerca de la fuerza.

2. Sensores de presencia: Los sensores de presencia tienen como finalidad determinar la presencia de un objeto en un intervalo de distancia especificado. Este tipo de sensor se utiliza para tomar o evitar algún objeto. Se basan en el cambio de alguna característica del sensor debido a la proximidad del objeto. Algunos sensores de presencia son:

- **Sensores Inductivos:** Estos sensores se basan en el cambio de inductancia que provoca un objeto metálico en un campo magnético, constan básicamente de una bobina y de un imán. Cuando un objeto ferromagnético penetra o abandona el campo del imán el cambio que se produce en dicho campo induce una corriente en la bobina.
- **Sensores de efecto Hall:** El efecto Hall relaciona la tensión entre dos puntos de un material conductor o semiconductor con un campo magnético a través del material. Cuando un objeto (ferromagnético) se aproxima al sensor, el campo provocado por el imán en el elemento se debilita, así se puede determinar la proximidad de un objeto.
- **Sensores capacitivos:** Como su nombre lo indica, están basados en la detección de un cambio en la capacitancia del sensor provocado por una superficie próxima a éste. Este tipo de sensores tienen la ventaja de detectar la proximidad de objetos de cualquier naturaleza; sin embargo, hay que destacar que la sensibilidad disminuye bastante cuando la distancia es superior a algunos milímetros. Además, es dependiente del tipo de material.
- **Sensores ultrasónicos:** Su elemento principal es un transductor electroacústico. Este elemento emite ondas ultrasónicas, estas viajan y son reflejadas al chocar con un objeto, cuando las ondas regresan, quiere decir, que hay algún objeto en las proximidades. Dependiendo del tiempo de espera de las ondas ultrasónicas se detectará el grado de proximidad. Este tipo de sensores son más independientes del tipo de material que los anteriores y permiten detección de proximidad a mayores distancias.
- **Sensores ópticos:** Este tipo de sensores son muy parecidos a los anteriores. En estos, las señales que se transmiten y detectan son luminosas. En los sensores ópticos el emisor y el receptor suelen ser elementos separados. El primero suele ser un diodo emisor de luz (LED) y el receptor un fotodiodo.

3. Sensores de alcance: Los sensores de alcance miden la distancia desde un punto de referencia (que suele estar en el propio sensor) hasta objetos que están dentro de un determinado campo de referencia. La detección de alcance se suele usar para la evasión de obstáculos. A continuación examinaremos algunas técnicas de detección de alcance.

- **Triangulación:** Este es uno de los métodos más sencillos para medir el alcance. El sensor dispone de un emisor y un detector de luz. Un objeto se ilumina por un haz estrecho de luz que barre toda la superficie. Cuando el detector descubre luz en la superficie del objeto se puede calcular la distancia de la parte iluminada del objeto al detector con una sencilla

relación trigonométrica (suponiendo que conocemos la distancia del emisor al detector y el ángulo con el que la luz incide en el objeto).

- **Iluminación estructural:** Este método se basa en la proyección de una configuración de luz sobre un conjunto de objetos, y en la utilización de la distorsión de la luz para determinar el alcance a los diferentes objetos. La configuración de luz que suele transmitirse es de forma cilíndrica. El inconveniente principal de este método es que precisa de un tratamiento más o menos complejo de información (el de las imágenes) que ha de ser realizado por un ordenador.
- **Tiempo de Vuelo:** En este tipo de sensores, la estimación de la distancia a un objeto se basa en el tiempo transcurrido entre la emisión y recepción de un impulso sónico o luminoso análogamente al sistema usado por los murciélagos.

1.4 Lenguajes de programación de robot manipuladores

Programar un robot consiste en indicar paso a paso las diferentes acciones que el manipulador debe realizar durante la ejecución de la tarea. La utilidad del robot y la flexibilidad en la aplicación van a depender en gran parte de su programación.

Según el sistema utilizado para indicar la secuencia de acciones a realizar, un robot puede ser programado por:

- › Programación por guiado
- › Programación textual
- › Programación fuera de línea

1.4.1 Programación por guiado

La programación por guiado consiste en desplazar un sistema de referencia asociado al robot, de forma que se alcancen configuraciones deseadas al mismo tiempo que se registran sus valores. A este tipo de programación también se le conoce como programación no textual, programación por aprendizaje, programación en línea o programación en modo enseñanza. En los primeros tiempos de la robótica se distinguía entre guiado pasivo y guiado activo, estos son:

- › **Guiado pasivo:** El programador posicionaba el robot con sus propias manos y solo se registraban las coordenadas articulares que posteriormente se convertían en tensiones de mando de los servomotores que controlan el robot⁴.
- › **Guiado activo:** El robot se posiciona desde el puesto de mando, empleando para ello distintos dispositivos, registrándose durante la enseñanza las posiciones. Entre los dispositivos de guiado están: el teclado, la caja de aprendizaje o los dispositivos especiales

⁴ Anibal Ollero, "Robótica. Manipuladores", p. 343.

de guiado⁵.

Existe también una técnica de guiado dentro del guiado pasivo, que es el movimiento de una réplica del robot, de idéntica configuración pero más ligero y, por tanto, más fácil de mover. Obsérvese también que, en vez de un dispositivo físico, es posible mover un robot virtual que se visualiza en la pantalla de una PC.

La programación por guiado es relativamente simple y no requiere el estudio de lenguajes de programación. Sin embargo, si no se utilizan réplicas o robots virtuales, es necesario ocupar el robot durante el tiempo de programación, lo que obliga a sacar al robot de la línea de producción e interrumpirla.

Por otra parte, en la programación por guiado, existen problemas para incorporar la percepción del entorno y las estructuras de decisión. Así mismo, en algunas aplicaciones, es difícil que un operador posicione el robot con suficiente precisión, para evitar esos problemas se han creado sistemas por guiado avanzado o guiado extendido:

- › **Guiado extendido:** Permite especificar datos junto a los puntos por los que debe pasar el robot; datos relativos como velocidad, tipo de trayectoria, precisión con la que se quiere alcanzar los puntos, control del flujo del programa, atención a entradas/salidas binarias, etc. Es decir, la programación de guiado extendido permite realizar movimientos precavidos.

1.4.2 Programación textual

La programación textual permite indicarle la tarea al robot a través de un lenguaje de programación específico. Un programa se entiende como una serie de órdenes que son editadas y posteriormente ejecutadas, por lo tanto, existe un texto para el programa.

La ventaja principal de la programación textual es que permite expresar la adaptación del comportamiento del robot ante imprecisiones y variaciones del entorno de trabajo medidas mediante sensores.

La clasificación más frecuente de la programación textual es:

- a. Nivel robot
 - b. Nivel objeto
 - c. Nivel tarea
-
- a. Nivel robot:** Cuando las instrucciones se refieren a los movimientos realizados por parte del robot, por ejemplo, quitar B de A poniendo B sobre C e introducir A en D, etc. La operación que consiste en colocar B sobre C, tendría el siguiente aspecto:

⁵ Anibal Ollero, "Robótica. Manipuladores", p. 343.

Mover_a P1 vía P2	; <i>Situarse en un punto sobre la pieza B</i>
Vel = 0.2 * VELMAX	; <i>Reducir la velocidad</i>
Pinza = ABRIR	; <i>Abrir la pinza</i>
Prec = ALTA	; <i>Aumentar la precisión</i>
Mover_recta_a P3	; <i>Descender verticalmente en línea recta</i>
Pinza = CERRAR	; <i>Cerrar la pinza para tomar la pieza B</i>
Espera= 0.5	; <i>Esperar para garantizar cierre de pinza</i>
Mover_recta_a P1	; <i>Ascender verticalmente en línea recta</i>
Prec = MEDIA	; <i>Decrementar la precisión</i>
Vel = VELMAX	; <i>Aumentar la velocidad</i>
Mover_a P4 via P2	; <i>Situarse sobre la pieza C</i>
Prec = ALTA	; <i>Aumentar la precisión</i>
Vel = 0.2 * VELMAX	; <i>Reducir velocidad</i>
Mover_recta_a P5	; <i>Descender verticalmente en línea recta</i>
Pinza = ABRIR	; <i>Abrir pinza</i>

- b. Nivel objeto:** Cuando las instrucciones se refieren al estado en el que deben ir quedando los objetos, por ejemplo: situar B sobre C haciéndolo coincidir y situar A dentro D haciéndolo coincidir:

LADO_B1 con LADO_C1 y LADO_B2 con LADO_C2	; <i>B sobre C haciéndolo coincidir</i>
EJE_A con EJE_HUECO_ y BASE_A con BASE_D	; <i>A dentro D haciéndolo coincidir</i>

- c. Nivel tarea:** Cuando las instrucciones se refieren al objetivo a conseguir. El programa se reduce a una única sentencia, ya que, se especifica qué es lo que debe hacer el robot en lugar de cómo debe hacerlo, por ejemplo:

Ensamblar A con D

La mayor parte de los lenguajes son de nivel robot, a los que también se denominan lenguajes de programación textual explícita, haciendo alusión a la necesidad de definir explícitamente las diferentes acciones.

La técnica de programación habitual consiste en que el programador establece la secuencia de operaciones preparando un texto y dejando para un posterior ajuste ciertos parámetros que dependen de información sensorial, el programador puede hacer uso de esta información para iniciar o terminar movimientos, involucrando sincronización con los elementos del entorno, escoger entre acciones alternativas, identificar objetos y sus posiciones, y acomodarse a condiciones externas de forma activa.

Existen diferentes clasificaciones de lenguajes de programación de nivel robot tal como lenguajes de movimientos primitivos y lenguajes de programación estructurada:

- › **Lenguajes de movimientos primitivos:** tienen un nivel de potencia próxima a la de programación por guiado avanzado, de la que se distingue por tener características tales como: bifurcaciones simples, subprogramas sin transferencia de parámetros, capacidad sensorial limitada y paralelismo elemental. Estos lenguajes suelen disponer de bastantes órdenes, algunas demasiado similares, pero pocas estructuras de control⁶.
- › **Lenguajes de programación estructurada:** se diferencian de los primitivos en que fueron desarrollados sobre una base de definición formal que les proporciona características tales como primitivas estructuradas, uso extensivo de transformaciones de coordenadas, definición de estructuras complejas de datos, definición por el usuario de subprogramas con parámetros y proceso paralelo estructurado y sincronizado. Sin embargo, el empleo generalizado de transformaciones hace más difícil la interpretación⁷.

1.4.3 Programación fuera de línea

La programación fuera de línea también conocida como OLP (Off-Line Programming), es un entorno de programación generalmente mediante gráficos computacionales, para llevar a cabo el desarrollo de programas sin la necesidad de acceder al robot. Un argumento común que ha surgido a favor a estos sistemas es que no es necesario detener el robot para reprogramarlo, por lo tanto, las fábricas automatizadas pueden permanecer en modo de producción una mayor porción del tiempo⁸, es decir, consiste en la manipulación del robot a partir de un lenguaje de programación con gráficos computacionales. Estos disponen generalmente de modelos 3D de los dispositivos que se utilizan en la realidad, así como piezas de trabajo y objetos del entorno.

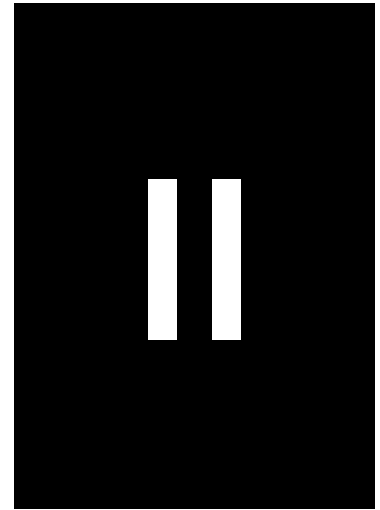
Funciona como un lenguaje interactivo el cual obliga al usuario a pasar por el ciclo “editar-compile-ejecutar” en cada modificación del programa.

Una ventaja del uso de la OLP es la facilidad de simular la detección automática de colisiones; esto es cuando ocurre un choque entre los objetos en un entorno simulado, el sistema OLP deberá advertir automáticamente al usuario e indicarle exactamente donde ocurrió tal colisión.

⁶ Anibal Ollero, “Robótica. Manipuladores”, p. 346.

⁷ Anibal Ollero, “Robótica. Manipuladores”, p. 346.

⁸ Jhon J., “ROBÓTICA”, p. 15.



ROBOT INDUSTRIAL RV-2AJ

Antes de iniciar el estudio del robot industrial RV-2AJ⁹, es de suma importancia saber las medidas de seguridad, las cuales no deben ser ignoradas, de lo contrario pueden suceder accidentes o daños a la unidad. Se recomienda que los alumnos sean pacientes pues las prácticas que realicen se deben probar a una velocidad mínima que se menciona más adelante.

2.1 Medidas de seguridad

En este tema se dan a conocer las condiciones de seguridad¹⁰ para el manejo del brazo robótico RV-2AJ, es imprescindible y obligatorio leerlas para evitar daños físicos a la unidad o lo que es más importante, lesiones¹¹ al usuario o terceros.

⁹ Ver apéndice E: “Cambio de baterías”, p. 97.

¹⁰ Ver manual “Detailed explanation of functions and operations”, p. 1-3.

¹¹ La gravedad de las lesiones pueden llegar a ser considerables.



Recomendaciones:

- Cualquier persona que trabaje con el robot industrial RV-2AJ, debe observar con especial atención las recomendaciones de seguridad.
- Deben respetarse las normas y regulaciones sobre prevención de accidentes.



Precauciones:

- Todo trabajo de enseñanza debe ser dada por una persona instruida.
- Realizar un plan de trabajo de métodos y procesos de operaciones del robot.
- El usuario nunca debe entrar en el área de trabajo del robot mientras este esté activo, es decir, mientras el servomotor esté encendido.
- Antes de comenzar a manipular el robot, el usuario debe asegurarse que no hay personas en el área de trabajo.
- Dar siempre una señal acústica antes de ejecutar un programa.
- Mantener siempre una mano en algún botón de paro de emergencia y observar el robot durante el ciclo operativo.
- Nunca ausentarse mientras el robot este encendido.
- Nunca apagar el controlador mientras esté encendido el servomotor.
- Evitar las colisiones. Si cree que el robot va a causar un choque o daño, oprima el botón de paro de emergencia, inmediatamente.
- Mantenga la velocidad tan baja como sea posible, y siempre mire el robot.
- Asegurarse que el peso de la pieza de trabajo (incluyendo la mano), no exceda el valor máximo de carga.
- Después de editar el programa, confirmar la operación.
- Nunca llevar a cabo modificaciones basadas en juicios personales.
- No tocar ninguna parte móvil del robot durante el funcionamiento.
- Al realizar alguna prueba, la velocidad del robot (OVRD) debe ser de 5, 10 o 20 como máximo.



Peligro:

- Instalación de cerca de seguridad.

2.2 Componentes

El robot industrial RV-2AJ también conocido como robot manipulador, ofrece una entrada óptima al complicado campo de la tecnología de robots, pues se trata de un verdadero robot industrial, el cual por pequeño que se vea, no se diferencia en calidad, posibilidades de aplicación y manipulación de los grandes robots.

Los componentes más importantes del robot industrial RV-2AJ¹², son:

- Brazo robótico RV-2AJ
- Controlador del robot
- T/B o caja de aprendizaje
- Pinza
- Módulos

2.2.1 Brazo robótico RV-2AJ

El brazo robótico RV-2AJ que se muestra en la figura 2.1, es el encargado de realizar los movimientos necesarios para ejecutar la tarea. Es un robot manipulador que cuenta con 5 grados de libertad, con una estructura de tipo angular y ofrece una capacidad máxima de carga de 2kg. Este robot tiene un alcance o área de trabajo de 410mm y combina una velocidad máxima de 2200 mm/s con una repetibilidad de posicionamiento de ± 0.02 mm, siendo ideal para entornos pequeños y especialmente para prácticas académicas y de investigación.



Figura 2.1: Brazo robótico RV-2AJ.

2.2.2 Controlador CR1-571

El controlador CR1-571¹³ es el encargado del sistema de control del robot; es decir, se encarga de los cálculos de la cinemática y dinámica de movimiento del robot. Guarda un máximo de 88 programas y permite la ejecución de hasta 32 programas en paralelo en modo multitarea, cuenta con un puerto serie RS-232C y 16 entradas/salidas digitales. A continuación se describen las partes del panel frontal de controlador CR1-571¹⁴ (figura 2.2):

¹² Ver apéndice A: “Características generales del robot industrial RV-2AJ”, p. 88.

¹³ Ver apéndice B: “Características generales del controlador CR1-571”, p. 89.

¹⁴ Ver manual “Detailed explanation of functions and operations”, p. 2-5.

1. **POWER:** Botón de encendido y apagado del controlador CR1-571.
2. **REMOVE T/B:** Se utiliza para conectar/desconectar el T/B sin desconectar la potencia.
3. **EMEG. STOP:** Paro de emergencia. El servomotor no se apaga. Girar el botón en sentido de las manecillas de reloj para cancelar este estado. Este funciona aunque el controlador este apagado.
4. **UP/DOWN:** Desliza arriba y abajo los detalles visualizados.
5. **RESET:** Restablece una situación de error. El led rojo se ilumina mientras se produce un error. También se restablece el programa en estado interrumpido.
6. **END:** Detiene el programa que se está ejecutando.
7. **CHANGE DISP.:** Cambia detalles visualizados en el LCD.
8. **START:** Ejecuta el programa seleccionado y opera el brazo robótico. El programa se ejecuta de forma continua. El led verde se ilumina durante la operación.
9. **STOP:** Detiene el brazo robótico inmediatamente. El servomotor no se apaga. El led rojo se enciende mientras está detenido, es decir, cuando se interrumpe un programa.
10. **SVO. ON:** Se encienden los servomotores.
11. **SVO. OFF:** Se apagan los servomotores.
12. **STATUS NUMBER:** Muestra el No. de programa, el No. de error y la velocidad.
13. **MODE:** Selección del modo de operación:
 - › **MODE TEACH:** Solo son válidas operaciones desde el T/B.
 - › **MODE AUTO (Ext.):** Solo son válidas operaciones desde el dispositivo exterior.
 - › **MODE AUTO PC:** Solo son válidas operaciones desde el controlador.
14. **CONECTOR T/B:** Base para conectar el T/B.
15. **CONECTOR RS-232C:** Base para establecer comunicación con la PC.



Figura 2.2: Panel frontal de operación del controlador CR1-571.

2.2.3 T/B o caja de aprendizaje

El T/B también conocido como caja de aprendizaje se utiliza para mover el brazo robótico y así determinar las posiciones requeridas para la tarea, estos movimientos se realizan con ayuda de la operación JOG. A continuación se describen las partes del panel frontal de la caja de aprendizaje del robot RV-2AJ¹⁵ (figura 2.3):

1. **[ENABLE/DISABLE]:** Habilita o deshabilita el T/B.
2. **[EMG. STOP]:** Seguro de paro de emergencia, el servomotor se apagará y el brazo robótico se detendrá inmediatamente aunque el T/B esté habilitado o no. Para cancelar este estado girar el botón en sentido de las manecillas del reloj.
3. **[ADD/↑]:** Mueve el cursor hacia arriba.
4. **[RPL/↓]:** Mueve el cursor hacia abajo.
5. **[DEL/←]:** Mueve el cursor a la izquierda.
6. **[HAND/→]:** Cuando es presionada junto con [+C(J6)] o [-C(J6)], la mano se abre o cierra. También mueve el cursor hacia la derecha.
7. **[INP/EXE]:** Ingresa al programa y lleva a cabo el regreso de paso.
8. **[JOG OPERATION]:** Son 12 teclas llamadas teclas de operación JOG y van desde [-X(J1)] hasta [+C(J6)], son usadas para ingresar valores alfanuméricos o la selección de algún menú.
9. **[ERROR RESET]:** Reinicia un estado de error ocurrido.
10. **[POS CHAR]:** Cambia entre caracteres numéricos y alfabéticos.
11. **[COND]:** Se utiliza para editar un programa.
12. **[-/BACKWD]:** Cuando esta tecla es presionada conjuntamente con la tecla [STEP/MOVE] la velocidad disminuye.
13. **[+/FORWD]:** Cuando esta tecla es presionada conjuntamente con la tecla [STEP/MOVE] la velocidad aumenta.
14. **[STEP/MOVE]:** Los servomotores se encienden; la selección de la operación JOG es permitida; el brazo robótico se puede mover. Para poder realizar tal movimiento se deben oprimir al mismo tiempo las teclas [STEP/MOVE]+[JOG OPERATION]+SWITCH DEADMAN¹⁶. Una vez que se deja de oprimir [STEP/MOVE] el servo se apaga.
15. Selecciona el modo de operación JOG:
 - › **[TOOL]:** Selecciona el modo TOOL.
 - › **[JOINT]:** Selecciona el modo JOINT.
 - › **[XYZ]:** Selecciona el modo XYZ, 3-EJES XYZ o CYLINDER.
16. **SWITCH DEADMAN:** Cuando [ENABLE/DISABLE] está en habilitado y esta tecla es liberada o presionada con fuerza el servomotor se apaga. Este botón se debe presionar suavemente¹⁷ cuando se está llevando a cabo una función con el servomotor encendido. Si el paro de emergencia es aplicado, el servomotor se apaga y no se encenderá incluso cuando el SWITCH DEADMAN esté ligeramente presionado. En este caso, se llevará a

¹⁵ Ver manual “Detailed explanation of functions and operations”, p. 2-8.

¹⁶ El SWITCH DEADMAN debe ser ligeramente presionado, es decir, antes de llegar a su límite.

¹⁷ Antes de llegar a su límite mecánico.

cabo la operación de encendido otra vez quitando el paro de emergencia. Este switch se ubica en la parte trasera del T/B.

17. LCD: Se despliega el contenido del programa, el estado del robot, estado de las operaciones JOG, entrada de datos desde el T/B.

18. SWITCH DE OPCIONES DE CONTRASTE: Control de brillo del LCD.

19. [MENU]: Despliega el menú en el LCD.

20. [STOP]: Detiene el programa y desacelera el robot hasta detenerse.

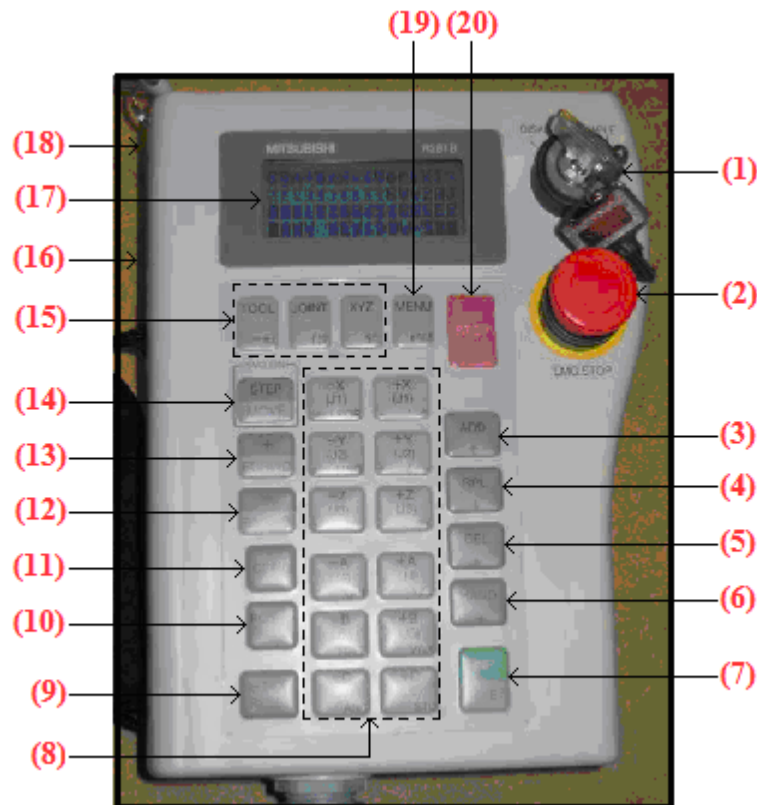


Figura 2.3: T/B o caja de aprendizaje.

2.2.4 Pinza

El efector final del brazo robótico RV-2AJ se muestra en la figura 2.4, se trata de una pinza neumática que consta de tres secciones cilíndricas de sujeción y un sensor óptico de reflexión directa (con cable de fibra óptica), integrado en la parte externa de la pinza. Tal sensor sirve para la identificación del color de piezas, ya sea en negras o no negras. No se dispone de detección de pinza abierta o cerrada. La pinza debe ser alimentada con una presión de 5 bar.



Figura 2.4: Pinza neumática de sujeción.

2.2.5 Módulos

Los módulos son elementos que ayudan a la realización de prácticas se emplean para transportar, almacenar, retener o incluso montar piezas de trabajo. En la figura 2.5, se observan los módulos del robot industrial RV-2AJ.

- › **Módulo de rampa 1:** Se utiliza para transporte o almacenaje de las piezas y así alimentar al módulo retenedor.
- › **Módulo de rampa 2:** Este módulo de rampa transporta y almacenaje las piezas.
- › **Módulo retenedor:** Se utiliza para inmovilizar las piezas que llegan a través del modulo de rampa 1 y son detectadas por medio de un sensor óptico que se encuentra montado en este módulo.
- › **Módulo de montaje:** Sirve para realizar el ensamblaje de las piezas, pero también puede servir como base para colocar alguna otra pieza. Este módulo cuenta con un sensor óptico que identifica si la pieza es negra o no negra.
- › **Módulo de almacén:** El módulo almacén se utiliza para almacenar piezas redondas. Se pueden depositar hasta 8 piezas en cualquier orden en él.



Figura 2.4: Módulos.

2.3 Modos de operación JOG

Los modos de operación JOG también llamados modos de desplazamientos, son los posibles movimientos que nos ayudan a mover el brazo robótico a una posición deseada de manera manual a través del T/B. Estos modos de operación son:

- › **JOINT**
- › **TOOL**
- › **XYZ**
- › **3-EJES XYZ**
- › **CYLINDER**

2.3.1 Modo de operación JOINT

El modo de operación JOINT¹⁸, realiza el movimiento de cada eje de manera independiente. Para seleccionar este modo de operación, se realizan los siguientes pasos:

1. Seleccionar MODE TEACH en el controlador.
2. Habilitar el T/B.
3. Oprimir ligeramente¹⁹ el SWITCH DEADMAN.
4. Pulsar la tecla [STEP/MOVE]. El servo se enciende.
5. Pulsar la tecla [JOINT].
6. Pulsar tecla deseada del [OPERATION JOINT].

En las figuras 2.5a, b, y 2.6c y d se muestran las combinaciones de teclas, la forma y dirección del movimiento de las articulaciones J1, J2; J3, J5 y J6, respectivamente. En la tabla 2.1 se visualizan en forma de lista las combinaciones de teclas y se describen las acciones de cada una de ellas.

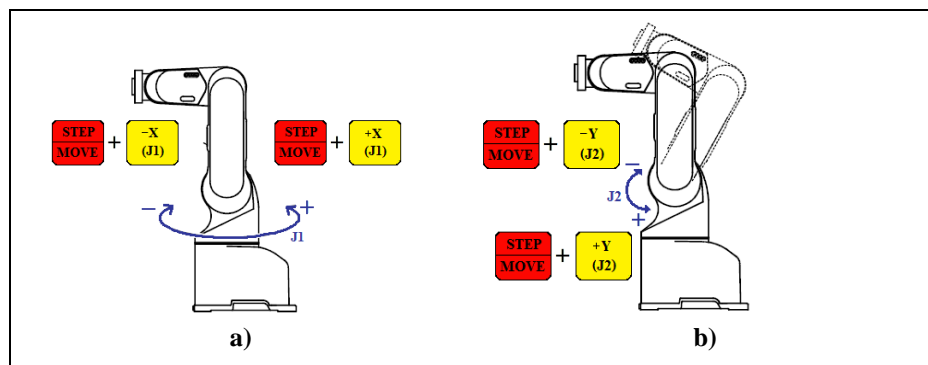


Figura 2.5: Modo de operación JOINT.

¹⁸ Ver manual “Robot arm & maintenance”, p. 2-17.

¹⁹ Antes de llegar a su límite mecánico.

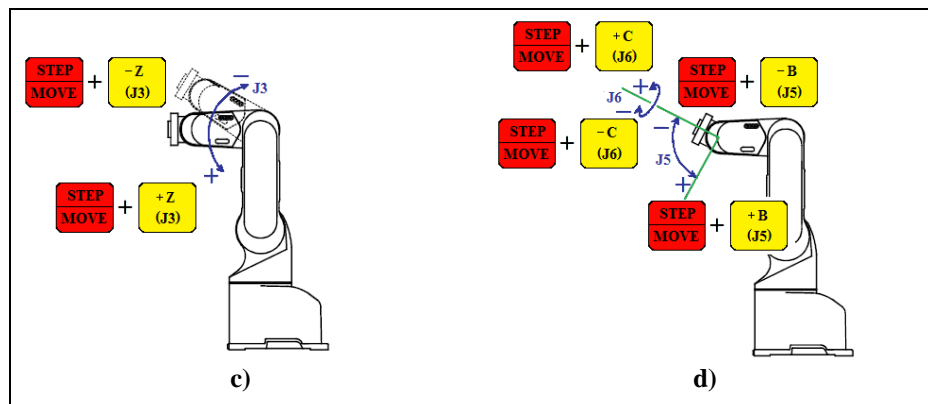


Figura 2.6: Continuación de modo de operación JOINT.

Figura 2.5	Combinación de teclas	Acción
a)	[MOVE] + [+X (J1)]	El eje J1 rotará en la dirección positiva
	[MOVE] + [- X (J1)]	El eje J1 rotará en la dirección negativa
b)	[MOVE] + [+Y (J2)]	El eje J2 rotará en la dirección positiva
	[MOVE] + [- Y (J2)]	El eje J2 rotará en la dirección negativa
c)	[MOVE] + [+Z (J3)]	El eje J3 rotará en la dirección positiva
	[MOVE] + [- Z (J3)]	El eje J3 rotará en la dirección negativa
d)	[MOVE] + [+B (J5)]	El eje J5 rotará en la dirección positiva
	[MOVE] + [- B (J5)]	El eje J5 rotará en la dirección negativa
	[MOVE] + [+C (J6)]	El eje J6 rotará en la dirección positiva
	[MOVE] + [- C (J6)]	El eje J6 rotará en la dirección negativa

Tabla 2.1: Combinación de teclas del modo de operación JOINT y la acción correspondiente.

2.3.2 Modo de operación TOOL

Con el modo de operación TOOL²⁰, el brazo se mueve respecto al sistema coordinado de la pinza mientras que ésta mantiene su postura. Para seleccionar este modo de operación, se realizan los siguientes pasos:

1. Seleccionar MODE TEACH en el controlador.
2. Habilitar el T/B.
3. Oprimir ligeramente²¹ el SWITCH DEADMAN.
4. Pulsar la tecla [STEP/MOVE]. El servo se enciende.
5. Pulsar la tecla [TOOL].
6. Pulsar tecla deseada del [OPERATION JOINT].

²⁰ Ver manual “Robot arm & maintenance”, p. 2-18.

²¹ Antes de llegar a su límite mecánico.

En la figura 2.7a y b se muestran las combinaciones de teclas, la forma y dirección del movimiento. En la tabla 2.2 se visualizan en forma de lista las combinaciones de teclas y se describen las acciones de cada una de ellas.

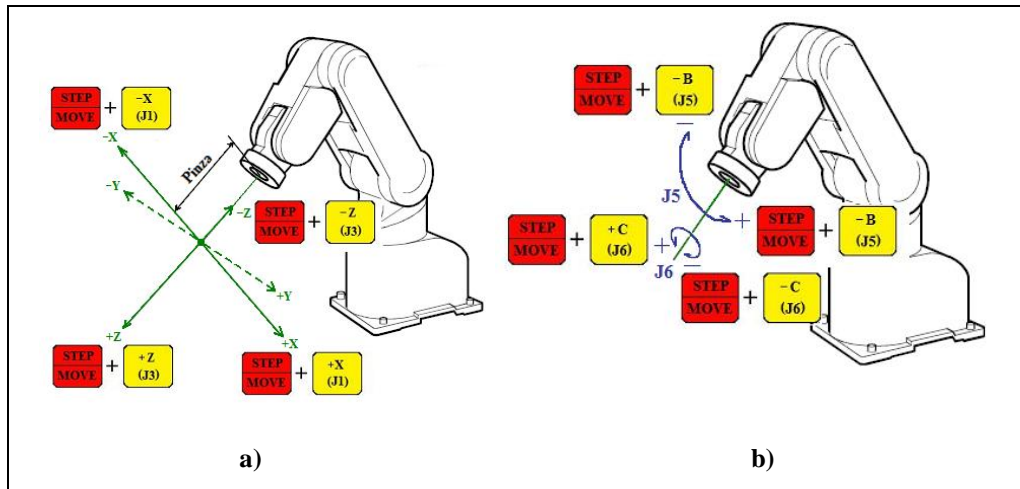


Figura 2.7: Movimiento TOOL.

Figura 2.6	Combinación de teclas	Acción
a)	[MOVE] + [+X (J1)]	El robot se moverá a lo largo del eje X en la dirección positiva del sistema coordenado de la pinza
	[MOVE] + [- X (J1)]	El robot se moverá a lo largo del eje X en la dirección negativa del sistema coordenado de la pinza
	[MOVE] + [+Y (J2)]	No hay ninguna operación
	[MOVE] + [- Y (J2)]	No hay ninguna operación
	[MOVE] + [+Z (J3)]	El robot se moverá a lo largo del eje Z en la dirección positiva del sistema coordenado de la pinza
	[MOVE] + [- Z (J3)]	El robot se moverá a lo largo del eje Z en la dirección negativa del sistema coordenado de la pinza
b)	[MOVE] + [+A (J4)]	No hay ninguna operación
	[MOVE] + [- A (J4)]	No hay ninguna operación
	[MOVE] + [+B (J5)]	El eje J5 rota en dirección positiva
	[MOVE] + [- B (J5)]	El eje J5 rota en dirección negativa
	[MOVE] + [+C (J6)]	El eje J6 rota en dirección positiva
	[MOVE] + [- C (J6)]	El eje J6 rota en dirección negativa

Tabla 2.2: Combinación de teclas del modo de operación TOOL y la acción correspondiente.

2.3.3 Modo de operación XYZ

Con el modo de operación XYZ²², el brazo se mueve respecto al sistema coordenado del robot mientras que la pinza mantiene su postura. Para seleccionar este modo de operación, se realizan los siguientes pasos:

²² Ver manual “Robot arm & maintenance”, p. 2-18.

1. Seleccionar MODE TEACH en el controlador.
2. Habilitar el T/B.
3. Oprimir ligeramente²³ el SWITCH DEADMAN.
4. Pulsar la tecla [STEP/MOVE]. (El servo se enciende).
5. Pulsar la tecla [XYZ] tantas veces hasta que seleccione el modo XYZ.
6. Pulsar tecla deseada del [OPERATION JOINT].

En la figura 2.8a y b se muestran las combinaciones de teclas, la forma y dirección del movimiento. En la tabla 2.3 se visualizan en forma de lista las combinaciones de teclas y se describen las acciones de cada una de ellas.

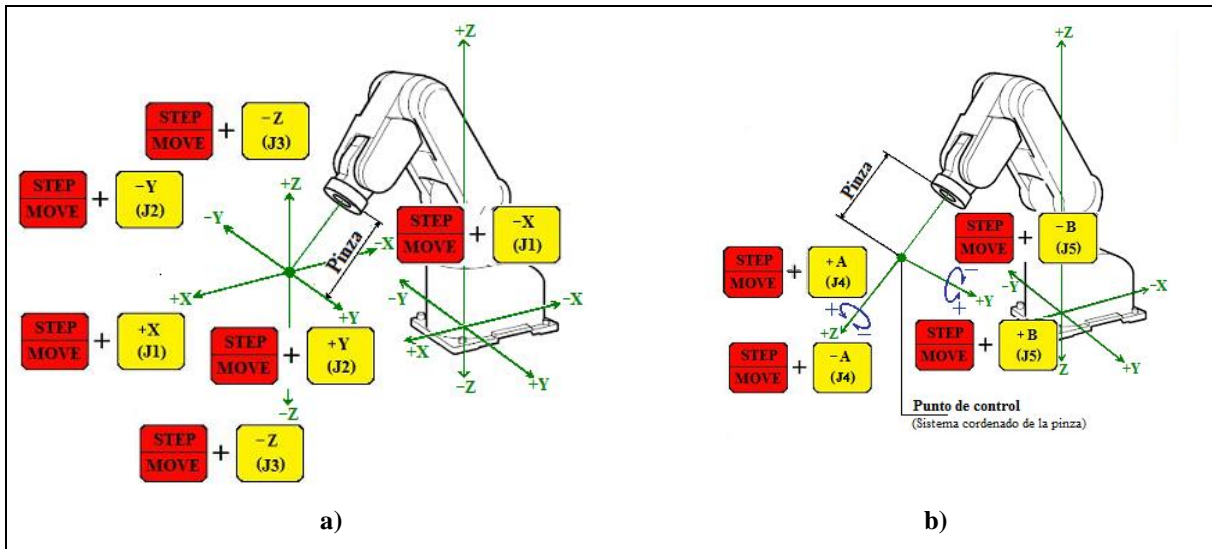


Figura 2.8: Movimiento XYZ.

Figura 2.7	Combinación de teclas	Acción
a)	[MOVE] + [+X (J1)]	El robot se moverá a lo largo del eje X en dirección positiva
	[MOVE] + [- X (J1)]	El robot se moverá a lo largo del eje X en dirección negativa
	[MOVE] + [+Y (J2)]	El robot se moverá a lo largo del eje Y en dirección positiva
	[MOVE] + [- Y (J2)]	El robot se moverá a lo largo del eje Y en dirección negativa
	[MOVE] + [+Z (J3)]	El robot se moverá a lo largo del eje Z en dirección positiva
	[MOVE] + [- Z (J3)]	El robot se moverá a lo largo del eje Z en dirección negativa
b)	[MOVE] + [+A (J4)]	El eje Z rota en dirección positiva del sistema de coordenadas de la pinza
	[MOVE] + [- A (J4)]	El eje Z rota en dirección negativa del sistema de coordenadas de la pinza
	[MOVE] + [+B (J5)]	El eje Y rota en dirección positiva del sistema de coordenadas de la pinza
	[MOVE] + [- B (J5)]	El eje Y rota en dirección negativa del sistema de coordenadas de la pinza
	[MOVE] + [+C (J6)]	No hay ninguna operación
	[MOVE] + [- C (J6)]	No hay ninguna operación

Tabla 2.3: Combinación de teclas del modo de operación XYZ y la acción correspondiente.

²³ Antes de llegar a su límite mecánico.

2.3.4 Modo de operación 3-EJES XYZ

Con el modo de operación 3-EJES XYZ²⁴, el brazo se mueve respecto al sistema coordinado del robot, la pinza no mantiene la postura. Para seleccionar este modo de operación, se realizan los siguientes pasos:

1. Seleccionar MODE TEACH en el controlador.
2. Habilitar el T/B.
3. Oprimir ligeramente²⁵ el SWITCH DEADMAN.
4. Pulsar la tecla [STEP/MOVE]. El servo se enciende.
5. Pulsar la tecla [XYZ] tantas veces hasta que seleccione el modo 3-EJES XYZ.
6. Pulsar tecla deseada del [OPERATION JOINT].

En la figura 2.9a y b se muestran las combinaciones de teclas, la forma y dirección del movimiento. En la tabla 2.5 se visualizan en forma de lista las combinaciones de teclas y se describen las acciones de cada una de ellas.

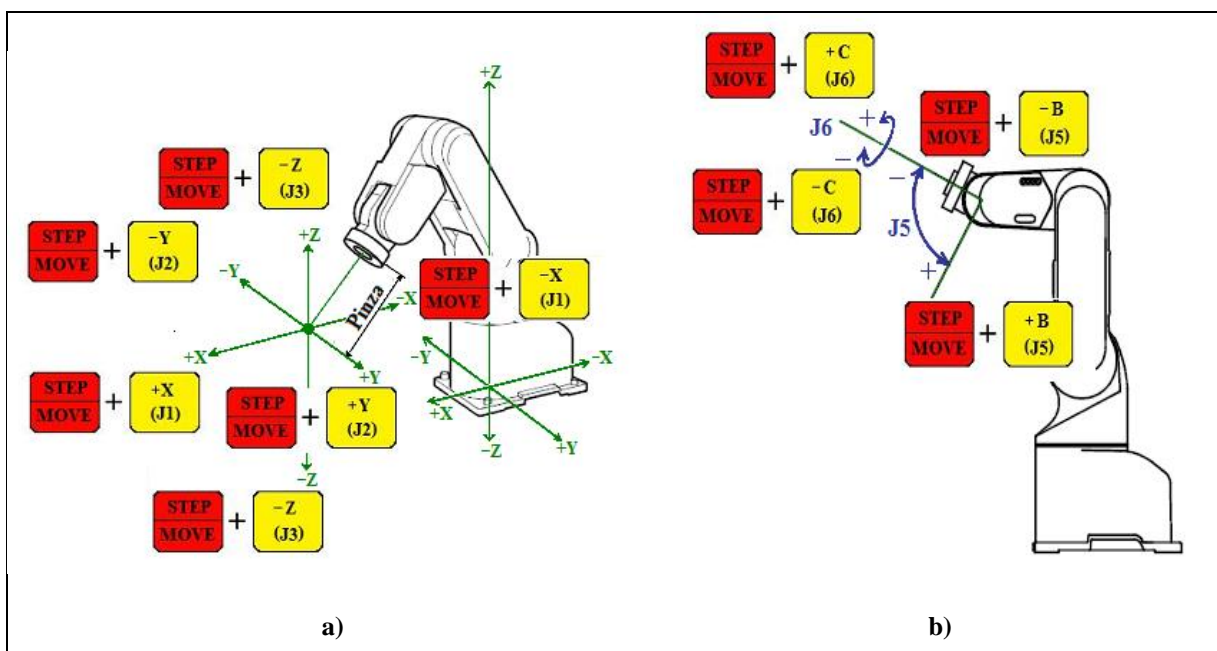


Figura 2.9: Movimiento 3-EJES XYZ.

²⁴ Ver manual "Robot arm & maintenance", p. 2-19.

²⁵ Antes de llegar a su límite mecánico.

Figura 2.8	Combinación de teclas	Acción
<i>a)</i>	[MOVE] + [+X (J1)]	El robot se moverá a lo largo del eje X en dirección positiva
	[MOVE] + [- X (J1)]	El robot se moverá a lo largo del eje X en dirección negativa
	[MOVE] + [+Y (J2)]	El robot se moverá a lo largo del eje Y en dirección positiva
	[MOVE] + [- Y (J2)]	El robot se moverá a lo largo del eje Y en dirección negativa
	[MOVE] + [+Z (J3)]	El robot se moverá a lo largo del eje Z en dirección positiva
	[MOVE] + [- Z (J3)]	El robot se moverá a lo largo del eje Z en dirección negativa
<i>b)</i>	[MOVE] + [+A (J4)]	No hay ninguna operación
	[MOVE] + [- A (J4)]	No hay ninguna operación
	[MOVE] + [+B (J5)]	El eje J5 rota en dirección positiva
	[MOVE] + [- B (J5)]	El eje J5 rota en dirección negativa
	[MOVE] + [+C (J6)]	El eje J6 rota en dirección positiva
	[MOVE] + [- C (J6)]	El eje J6 rota en dirección negativa

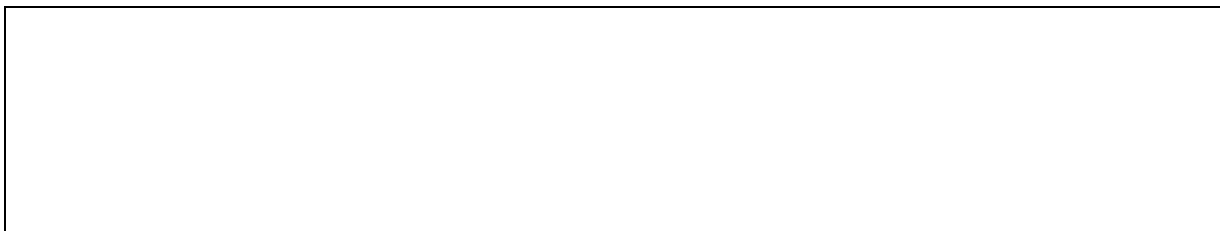
Tabla 2.4: Combinación de teclas del modo de operación 3-EJES XYZ y la acción correspondiente.

2.3.5 Modo de operación CYLINDER

Con el modo de operación CYLINDER²⁶, se establece como un arco con centro en el eje Z, el brazo robótico se mueve a lo largo de ese arco, se expande o contrae en la dirección de radio, y se mueve verticalmente. Para seleccionar este modo de operación, se realizan los siguientes pasos:

1. Seleccionar MODE TEACH en el controlador.
2. Habilitar el T/B.
3. Oprimir ligeramente²⁷ el SWITCH DEADMAN.
4. Pulsar la tecla [STEP/MOVE]. El servo se enciende.
5. Pulsar la tecla [XYZ] tantas veces hasta que seleccione el modo CYLINDER.
6. Pulsar tecla deseada del [OPERATION JOINT].

En la figura 2.9a y b se muestran las combinaciones de teclas, la forma y dirección del movimiento. En la tabla 2.6 se visualizan en forma de lista las combinaciones de teclas y se describen las acciones de cada una de ellas.



²⁶ Ver manual “Robot arm & maintenance”, p. 2-19.

²⁷ Antes de llegar a su límite mecánico.

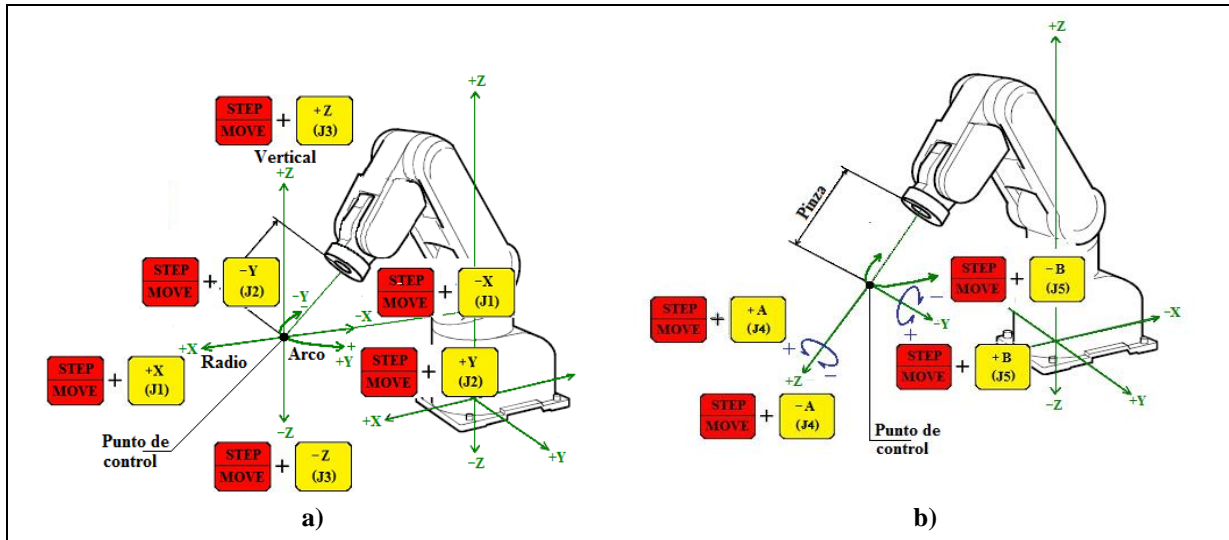


Figura 2.9: Movimiento CYLINDER.

Figura 2.9	Combinación de teclas	Acción
a)	[MOVE] + [+X (J1)]	El radio del robot se expandirá
	[MOVE] + [- X (J1)]	El radio del robot se contraerá
	[MOVE] + [+Y (J2)]	El robot se moverá a lo largo del arco en dirección positiva
	[MOVE] + [- Y (J2)]	El robot se moverá a lo largo del arco en dirección negativa
	[MOVE] + [+Z (J3)]	El robot se moverá a lo largo del eje Z en dirección positiva
	[MOVE] + [- Z (J3)]	El robot se moverá a lo largo del eje Z en dirección negativa
b)	[MOVE] + [+A (J4)]	El eje Z rota en dirección positiva del sistema de coordenadas de la pinza
	[MOVE] + [- A (J4)]	El eje Z rota en dirección negativa del sistema de coordenadas de la pinza
	[MOVE] + [+B (J5)]	El eje Y rota en dirección positiva del sistema de coordenadas de la pinza
	[MOVE] + [- B (J5)]	El eje Y rota en dirección negativa del sistema de coordenadas de la pinza
	[MOVE] + [+C (J6)]	No hay ninguna operación
	[MOVE] + [- C (J6)]	No hay ninguna operación

Tabla 2.5: Combinación de teclas del modo de operación CYLINDER y la acción correspondiente.

2.4 Posición inicial

La posición inicial²⁸ es la posición de partida que debe adoptar el brazo robótico al comienzo de cada programa, esta debe cumplir con lo siguientes:

²⁸ La posición inicial ocupada en este trabajo tiene una forma similar a la letra griega gamma "Γ"

- › La pinza se encuentra en una posición en la que no son posibles las colisiones con otros componentes de la estación.
- › Ningún eje del brazo del robot está cerca de una posición final.

2.5 Sensores

La Estación 6 donde se encuentra el robot industrial RV-2AJ, cuenta con sensores de reflexión directa, algunos²⁹ están ubicados en:

- › **Sensor 1:** En el módulo retenedor, da información sobre la presencia de alguna pieza.
- › **Sensor 2:** En la pinza, comúnmente se utiliza para saber si un objeto es negro o no.
- › **Sensor 3:** En el módulo de montaje, para determinar la orientación de la pieza. En este trabajo, se utiliza para sensar si hay pieza o no.

Estos sensores se basan en la detección de la luz reflejada por el objeto a detectar. El ajuste se realiza mediante un tornillo situado en el cuerpo de la fotocélula, ver figura 2.10. Para realizar el ajuste de esta fotocélula se utiliza un desarmador plano miniatura, cuando la detección es correcta, se enciende el indicador luminoso.

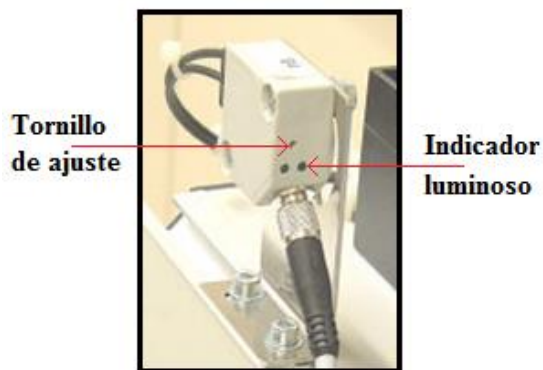


Figura 2.10: Módulos.

2.6 Alimentaciones

La estación 6 necesita de tres alimentaciones para poder operar de manera eficiente:

- › Alimentación eléctrica (220V AC)
- › Alimentación neumática (5 bar)
- › Para los sensores (24 VDC, máx. 5A).

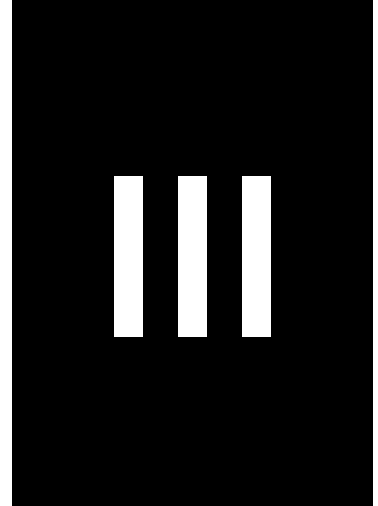
²⁹ Solo se mencionan los que se utilizaron en este trabajo.

2.7 Limpieza

La estación 6 del robot industrial RV-2AJ, casi no necesita mantenimiento. Aún así, se recomienda limpiarla de forma periódica, pues de no hacerlo, las partículas de suciedad pueden originar falsas detecciones. Se debe limpiar:

- › Las lentes de los sensores ópticos
- › Los reflectores catadióptricos
- › Toda la estación

Para esta acción, utilizar un pañuelo limpio y seco que no genere residuos. No utilizar productos de limpieza.



PROGRAMACIÓN

Para programar una tarea que se quiera ejecutar en el brazo robótico RV-2AJ, se utiliza el entorno de simulación COSIMIR ® Industrial 4.1 y el lenguaje de programación MELFA BASIC IV, los cuales se explican de forma independiente y detallada. Finalizando con el tema de “Pasos para la elaboración de prácticas con el robot industrial RV-2AJ”.

3.1 COSIMIR ® Industrial 4.1

El software COSIMIR ® Industrial 4.1 es un entorno de simulación 3D para el robot industrial RV-2AJ, el cual permite comprobar la accesibilidad de todas las posiciones y realizar la programación de las tareas que el brazo robótico debe desempeñar. Además, simula las secuencias de movimientos para evitar colisiones reales.

3.1.1 Nuevo proyecto

Para abrir un nuevo proyecto en COSIMIR ® Industrial 4.1, se deben realizar los siguientes pasos:

1. Colocar la llave física³⁰ en un puerto USB de la PC.
2. Abrir el programa COSIMIR ® Industrial:

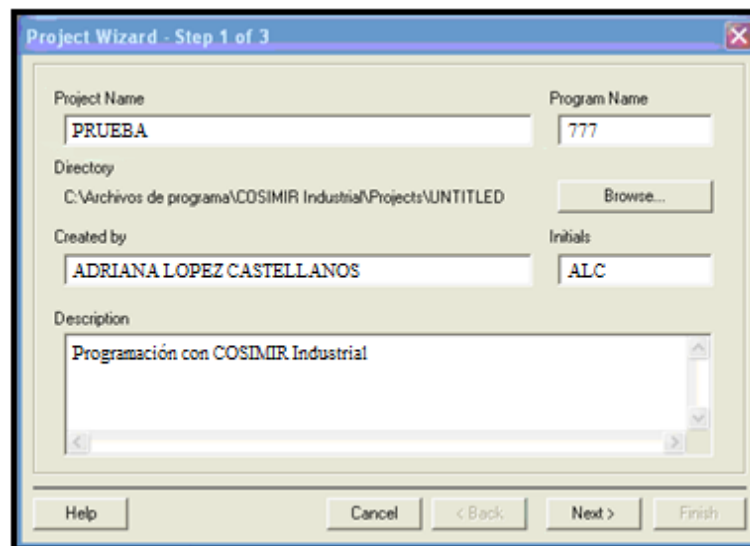
Menú Inicio → Programas → COSIMIR Industrial → COSIMIR Industrial

3. Dentro del programa de COSIMIR Industrial, generar un proyecto:

Menú File → Project Wizard

Aparece la ventana que se muestra en la figura 3.1. Se rellenan los siguientes campos:

- › **Project Name:** Nombre del proyecto.
- › **Program Name:** Nombre del programa, este debe ser con caracteres numéricos.
- › **Created by:** Nombre del autor.
- › **Initials:** Iniciales del autor.
- › **Description:** Descripción del programa.



The image shows a screenshot of the 'Project Wizard - Step 1 of 3' dialog box. It has a title bar with a close button. The dialog contains several input fields and buttons. The 'Project Name' field contains 'PRUEBA'. The 'Program Name' field contains '777'. The 'Directory' field contains 'C:\Archivos de programa\COSIMIR Industrial\Projects\UNTITLED' and has a 'Browse...' button next to it. The 'Created by' field contains 'ADRIANA LOPEZ CASTELLANOS'. The 'Initials' field contains 'ALC'. The 'Description' field contains 'Programación con COSIMIR Industrial'. At the bottom, there are buttons for 'Help', 'Cancel', '< Back', 'Next >', and 'Finish'.

Figura 3.1: Datos de entrada.

4. Seleccionar RV-2AJ en “Robot Type” y compruebe que la opción seleccionada en “Programming Lenguaje” sea MELFA-BASIC IV, como se muestra en la figura 3.2, hacer click en “Finísh”.

³⁰ La llave física se encuentra en el Laboratorio Móvil de Automatización y Robótica de la Universidad Veracruzana.

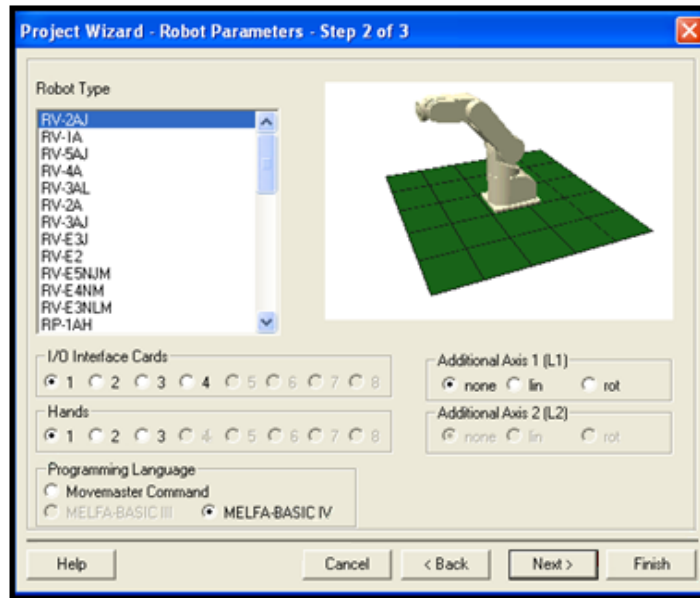


Figura 3.2: Selección del tipo de robot y del lenguaje de programación.

5. Finalmente, se abre una ventana similar a la figura 3.3. ¡El nuevo proyecto ha sido creado!

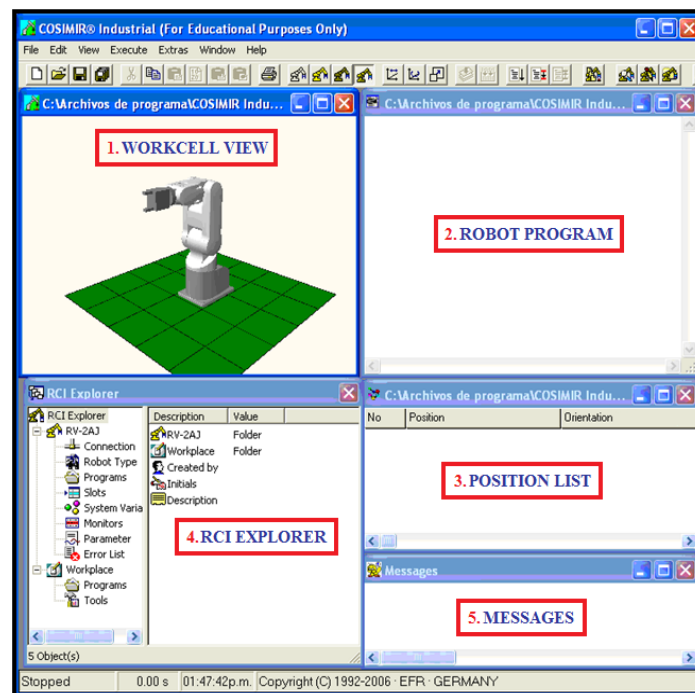


Figura 3.3: Proyecto creado.

Una vez abierto el programa de simulación COSIMIR ® Industrial 4.1, de manera predeterminada, se abren consigo 5 ventanas como se muestra en la figura 3.3. Estas ventanas son:

1. **Workcell View:** Ventana de simulación. Consiste en la simulación de la tarea a ejecutar por el brazo robótico RV-2AJ.
2. **Robot Program:** Ventana de programación. Aquí se realiza el programa de la tarea del robot. Este programa se realiza con el lenguaje de programación MELFA BASIC IV. Cuando el nuevo proyecto es guardado, este archivo tiene la extensión .MB4.
3. **Position List:** Ventana de posiciones. Contiene la lista de posiciones que se ocupan en la programación de la tarea deseada. Estas posiciones³¹ se obtienen de manera manual con ayuda del T/B³². Cuando el nuevo proyecto es guardado, este archivo tiene la extensión .POS.
4. **RCI Explorer:** Ventana de interfaz controladora del robot. Contiene las propiedades del controlador en la PC.
5. **Messages:** Ventana de mensajes. Despliega las tareas realizadas dentro del COSIMIR ® Industrial 4.1, así como los errores³³ y advertencias durante la compilación.

Los archivos .POS y .MB4, que corresponden a la lista de posiciones y al programa respectivamente, deben tener el mismo nombre de archivo.

3.1.2 Barra de herramientas

Dentro de la ventana principal de COSIMIR ® Industrial 4.1 se encuentra una barra de herramientas. Los iconos que la conforman llaman directamente a las opciones. Los iconos que se ocupan en las prácticas, se muestran en las tablas 3.1 y 3.2:








Icono	Nombre	Descripción
	New	Abre un nuevo programa en el proyecto actual.
	Open	Abre un archivo existente, es decir, el archivo es cargado y sus contenidos son mostrados.
	Save	Almacena los datos perteneciendo a la ventana activa o seleccionada. Esta orden es sólo disponible si la ventana activa contiene un programa de robot, una lista de posición o muestra una celda de trabajo.
	Save all	Almacena todo los archivos comprendidos en cualquiera ventana abierta.
	Renumber	Renombra las líneas de código del programa.
	Compile	Se usa para compilar el programa del robot cuando la ventana de programación está activa.
	Compile+Link	Verifica el proyecto y se cargar en el control del robot simulado. Este se habilita cuando la ventana de programación está activa.

Tabla 3.1: Iconos de barras de herramientas.

³¹ Ver en este trabajo el tema “Modos de operación JOG”, p. 31.

³² Ver en este trabajo el tema “T/B o caja de aprendizaje”, p. 28.

³³ Ver apéndice D: “Lista de errores”, p. 94.








Icono	Nombre	Descripción
	Start	Ejecuta el programa de manera continua.
	Stop	Detiene la ejecución del programa.
	Next Step	Ejecuta el próximo paso del programa.
	Previous Step	Ejecuta el paso anterior del programa.
	Init Connection	Inicia la conexión entre el PC-Robot.
	Jog Operation	Entra en la opción de toma de posiciones automática.
	I/O Monitor	Monitoriza el estado de las entradas y salidas.

Tabla 3.2: Continuación de iconos de barras de herramientas.

3.1.3 Toma de posiciones

Para realizar un programa, se deben tener las posiciones reales, en las cuales, el brazo robótico realizará su trayectoria.

La toma de posiciones se realiza por medio del T/B, para esto se deben de llevar a cabo los siguientes pasos:


1. Poner el controlador CR1-571 en MODE TEACH.
2. Habilitar el T/B.
3. Mover el brazo robótico a la posición deseada, utilizando los modos de operación JOG³⁴ del T/B.
4. En la PC, abrir Jog Operation haciendo click en el icono .
5. Seleccionar el número de la posición deseada en “Pos.-No.” y registrar la posición actual del robot haciendo click en “Current Position → Pos. List”, como se muestra en la figura 3.4.



Figura 3.4: Toma de posiciones.

³⁴ Ver en este trabajo el tema “Modos de operación JOG”, p. 31.

Notas:

- De manera automática, el “Pos.-No.” se incrementa y las posiciones se registran en la ventana Position List.
- La ventana de Jog Operation seguirá abierta hasta dar click en cerrar.
- La comunicación entre PC-Robot se habilita automáticamente al hacer click por primera vez en el icono de Jog Operation.
- En la lista de posiciones no deben existir dos posiciones con el mismo nombre.
- Cada vez que se registre una posición, es importante anotar algún comentario de referencia, para facilitar la programación.

6. En la ventana Position List, seleccionar la posición dando un click izquierdo dentro de la fila correspondiente, inmediatamente dar click derecho y seleccionar “Properties” dentro del menú contextual que aparece, como se muestra en la figura 3.5.

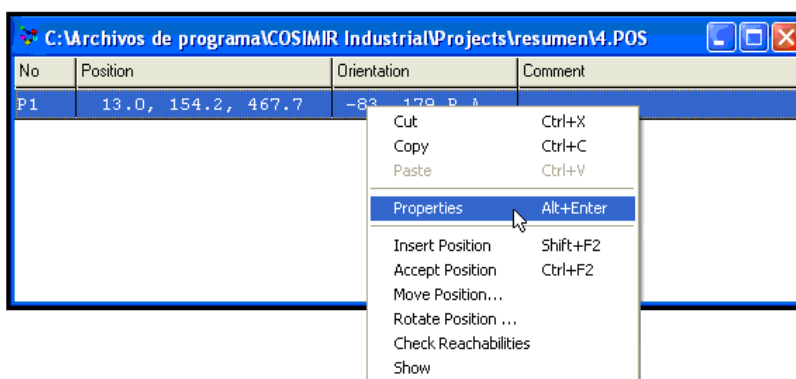


Figura 3.5: Selección de las propiedades de posición.

A continuación, emerge la ventana que se ilustra en la figura 3.6 se escribe la referencia en la caja de texto marcada con un asterisco, en este caso, la referencia es “posición inicial”.

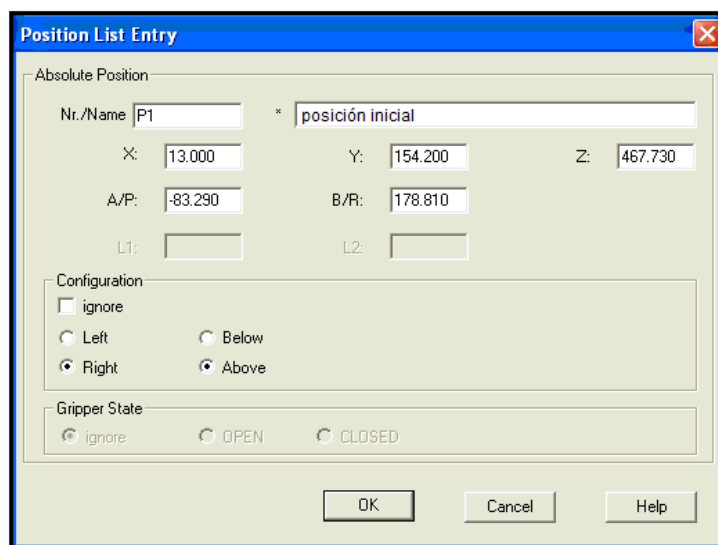


Figura 3.6: Propiedades de posición.

Cabe mencionar, que también es posible cambiar los valores de las coordenadas X, Y y Z, además de la inclinación y giro de la muñeca; esto se lleva a cabo ingresando a las cajas de texto marcadas con X, Y, Z, A/P y B/R, respectivamente.

3.1.4 Descarga de los archivos .POS y .MB4 al controlador CR1-571

Cuando se tenga el programa compilado y simulado previamente, sin generar error o un movimiento no deseado, se prosigue a descargar el programa al controlador CR1-571.

En la ventana de RCI Explorer, seleccionar “Programs” que se encuentra dentro de la división Workplace. En el cuadro de la derecha aparecerán dos archivos, uno con extensión .MB4 y el otro con extensión .POS (ambos con el mismo nombre). Estos dos archivos se deben descargar al controlador, arrastrando cada uno de ellos a “Programs” dentro de la división RV-2AJ.

En figura 3.7, se muestra un ejemplo de lo mencionado anteriormente, el archivo que se pretende cargar es el archivo de nombre 1.MB4.

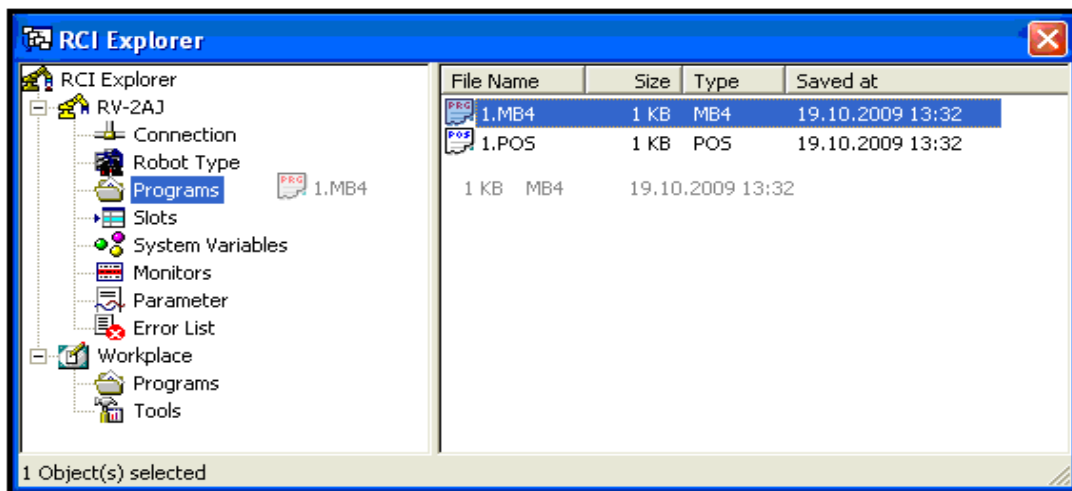


Figura 3.7: Solicitud de carga del archivo 1.MB4 al controlador CR1-571.

Notas:

- En la ventana RCI Explorer, en la división RV-2AJ se encuentra la información del controlador, así como los programas almacenados dentro del mismo. Siempre y cuando se hayan actualizado los datos, lo cual se mencionará más adelante.
- En la ventana RCI Explorer, en la división Workplace se encuentra la información y programas del proyecto actual.

Continuando con lo anterior, una vez soltado el archivo en Programs de la división de RV-2AJ, aparecerá la ventana Up. and Download que se muestra en la figura 3.8, en esta ventana haciendo click en OK, se da la orden de descargar tal archivo al controlador.

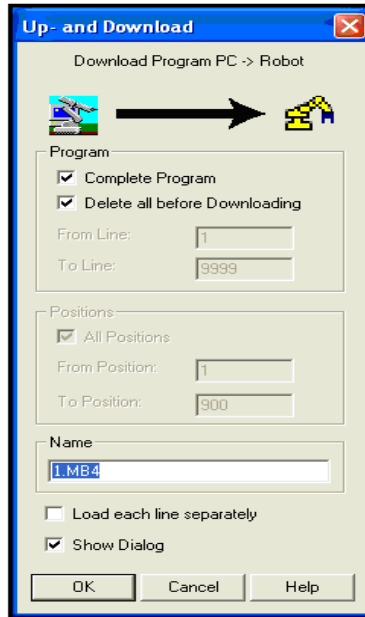


Figura 3.8: Cargando el archivo 1.MB4 al controlador CR1-571.

El porcentaje de descarga se puede visualizar en la ventana Robot Communication que se abre de manera automática, esta ventana se ilustra en la figura 3.9.

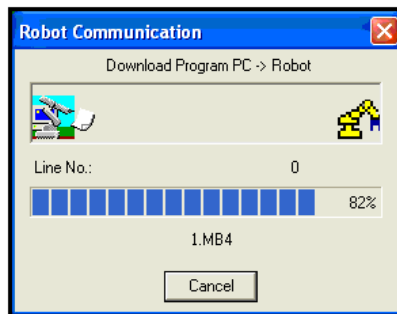


Figura 3.9: Porcentaje de descarga.

Recordar que se deben descargar al controlador ambos archivos con extensión .MB4 y .POS por separado. La descarga al controlador del archivo con extensión .POS se realiza de manera similar a la descarga del archivo .MB4³⁵.

Notas:

- Si se han cargado los dos archivos al controlador y se realiza alguna modificación al programa sin modificar la lista de posiciones, solo se necesita descargar el nuevo archivo .MB4.
- Si se han cargado los dos archivos al controlador y se realiza alguna modificación a la lista de posiciones sin alterar el programa, solo se necesita descargar el nuevo archivo .POS.
- Si ambos archivos son modificados, se deberán descargar nuevamente.

³⁵ Si no se realizan ambas descargas, el robot no realiza la tarea y es recibido un mensaje de error.

3.1.5 Ejecución del programa con el brazo robótico RV-2AJ


Para que el brazo ejecute una tarea o programa previamente cargado en el controlador, se debe realizar lo siguiente:

1. Seleccionar MODE AUTO (Op.) en el controlador.
2. Seleccionar la velocidad³⁶ y el número de programa en el STATUS NUMBER del panel frontal del controlador, utilizando las teclas CHANGE DISP. Y UP/DOWN del controlador.
3. Habilitar los servomotores presionando el botón SVO.ON del controlador.
4. Ejecutar el programa oprimiendo la tecla START del controlador.

¡El brazo robótico comenzará a ejecutar el programa!

En caso que el robot tenga riesgo de alguna colisión, se debe oprimir inmediatamente cualquier botón de paro de emergencia.

3.1.6 Monitor de entradas y salidas (I/O)

Una manera de visualizar que entrada o salida está siendo accionada es dando click en el icono de I/O Monitor , aparecerá la ventana que se muestra en la figura 3.10. Del lado izquierdo con los cuadros verdes se representan las entradas y del lado derecho con los cuadros rojos se representan las salidas.

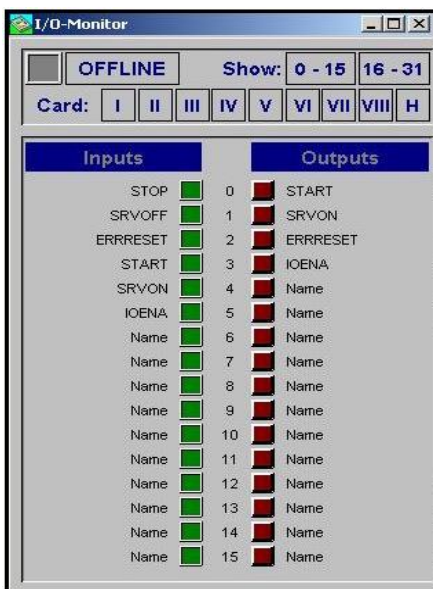


Figura 3.10: I/O Monitor.

³⁶ La velocidad máxima para realizar pruebas es de OVRD 20.

3.2 Lenguaje de programación MELFA BASIC IV

Para la programación del robot industrial RV-2AJ se utiliza el lenguaje de programación MELFA-BASIC IV, el cual se basa en el lenguaje BASIC estándar, ampliado con datos específicos de robots, entre ellos el robot RV-2AJ. Dicho lenguaje contiene instrucciones para el control del movimiento, control de la pinza y comandos de entrada y salida, además cuenta con una extensa biblioteca de funciones, entre las cuales se encuentran las funciones trigonométricas.

De este modo, el lenguaje de programación MELFA-BASIC IV resulta ser una excelente introducción para todos aquellos que quieran iniciarse en el mundo de la programación de robots permitiendo la creación de programas sencillos hasta programas altamente complejos.

3.2.1 Instrucciones de movimiento

Se describen las instrucciones de los tipos de movimientos que puede realizar el brazo robótico RV-2AJ los cuales son la base de la mayoría de las secuencias de una aplicación.

3.2.1.1 Interpolación angular (MOV)

En la interpolación angular³⁷ el brazo robótico genera la trayectoria interpolando los puntos que están entre el punto origen y el punto de destino. El robot se mueve a la posición indicada con una interpolación angular de cada eje, es decir, el movimiento no es lineal, por lo tanto, no es 100% predecible. Este movimiento, se ejecuta con mayor velocidad que los otros. Algunos ejemplos de declaraciones son:

MOV P1	' Se mueve con interpolación angular a P1
MOV P1+P2	' Mueve con interpolación angular a la posición resultante de la suma P1 y P2.
MOV P1*P2	' Mueve con interpolación angular a la posición relativa convertida de P1 a P2.
MOV P1,-50	' Se mueve con interpolación angular a la posición 50mm retraída de P1, en dirección de la pinza.
MOV P1 WTH M_OUT(17)=1	' Comienza el movimiento hacia P1 y a la vez, habilita la salida 17.
MOV P1 WTHIF M_IN(20)=1	' Si durante el movimiento hacia P1 la entrada 20 se activa, se detiene el movimiento y el programa sigue.

A continuación, se muestra un ejemplo empleando el uso de la instrucción MOV, en la figura 3.11 se visualiza de manera gráfica los movimientos de una tarea a realizar y en la tabla 3.3 se describe el código de esta.

³⁷ Ver manual "Detailed explanation of functions and operations", p. 4-61.

Ejemplo:

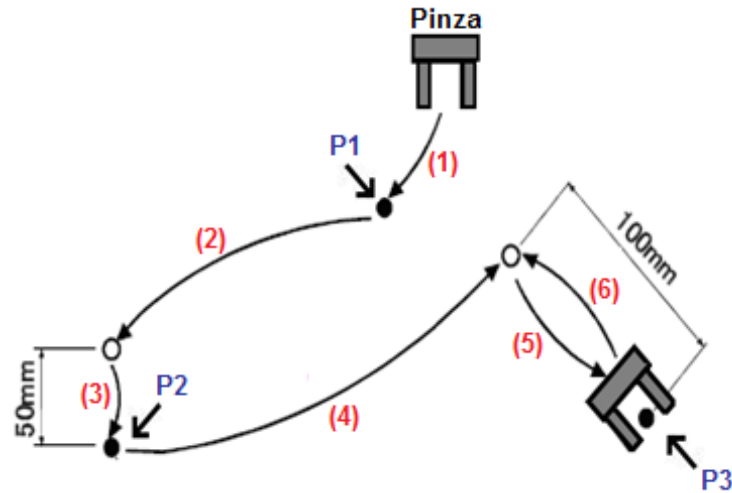


Figura 3.11: Interpolación angular.

Programa	Descripción
10 MOV P1	(1) Se mueve con interpolación angular a P1.
20 MOV P2,-50	(2) Se mueve con interpolación angular a la posición 50mm retráida de P2, en dirección de la pinza.
30 MOV P2	(3) Se mueve con interpolación angular a P2.
40 MOV P3,-100 WTH_MOUT(17)=1	(4) Se mueve con interpolación angular a la posición 100mm retráida de P3, en dirección de la pinza, y a la vez, habilita la salida 17.
50 MOV P3	(5) Se mueve con interpolación angular a P3.
60 MOV P3,-100	(6) Se mueve con interpolación angular a la posición 100mm retráida de P3, en dirección de la pinza.
70 END	' Fin del programa.

Tabla 3.3: Código de ejemplo de interpolación angular.

3.2.1.2 Interpolación lineal (MVS)

En la interpolación lineal³⁸ la trayectoria describe una línea recta en el espacio. Es un movimiento lento y solo se emplea en movimientos críticos cuando el robot se encuentra cerca de completar su trabajo como dejar o recoger una pieza. Algunos ejemplos de declaraciones son:

- MVS P1 ' Se mueve en línea recta a P1.
- MVS P1+P2 ' Se mueve en línea recta a la posición resultante de la suma P1 y P2.
- MVS P1*P2 ' Se mueve en línea recta a la posición relativa convertida de P1 a P2.

³⁸ Ver manual "Detailed explanation of functions and operations", p. 4-62.

- MVS P1,-50 ' Se mueve en línea recta a la posición 50mm retraída de P1, en dirección de la pinza.
- MVS ,-50 ' Se mueve en línea recta a la posición 50mm retraída de la posición actual, en dirección de la pinza.
- MVS P1 WTH M_OUT(17)=1 ' Comienza el movimiento hacia P1 y a la vez, habilita la salida 17.
- MVS P1 WTHIF M_IN(20)=1 ' Si durante el movimiento hacia P1 la entrada 20 se activa, se detiene el movimiento y el programa sigue.

A continuación, se muestra un ejemplo empleando el uso de la instrucción MVS; en la figura 3.12 se visualiza de manera gráfica los movimientos de una tarea a realizar mientras que en la tabla 3.4 se describe el código de esta.

Ejemplo:

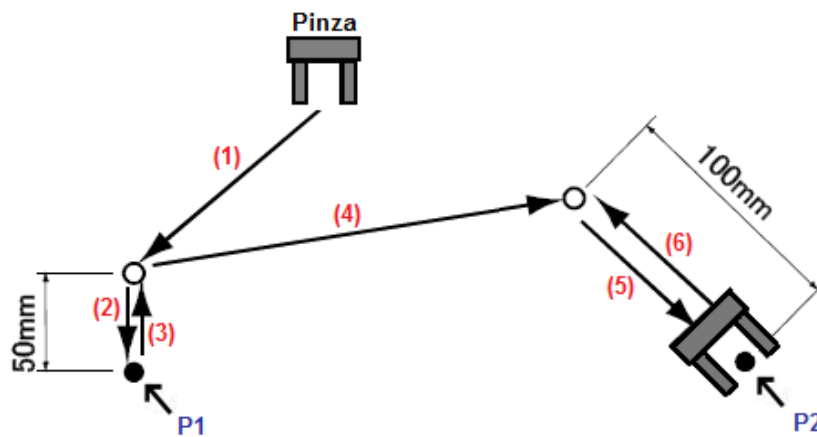


Figura 3.12: Interpolación lineal.

Programa	Descripción
10 MVS P1, -50	(1) Se mueve en línea recta a la posición 50mm retraída de P1, en dirección de la pinza.
20 MVS P1	(2) Se mueve en línea recta a P1.
30 MVS, -50	(3) Se retrae linealmente 50mm de P1 en dirección de la pinza.
40 MVS P2,-100 WTH M_OUT(17)=1	(4) Se mueve en línea recta a la posición 50mm retraída de P2 en dirección de la pinza, y a la vez, habilita la salida 17.
50 MVS P2	(5) Se mueve en línea recta a P2.
60 MVS, -50	(6) Se retrae linealmente 50mm de P2 en dirección de la pinza.
70 END	' Fin del programa.

Tabla 3.4: Código de ejemplo de interpolación lineal.

3.2.1.3 Interpolación circular (MVR y MVC)

En la interpolación circular³⁹ el robot se mueve por un círculo definido por tres puntos. Si al empezar el movimiento no se encuentra en el punto de inicio, el brazo robótico se moverá linealmente hasta él antes de empezar. Hay 4 tipos de instrucciones de interpolación circular, estos son:

- **MVR:** Se define un punto inicial, un punto de tránsito y un punto final, el robot se mueve en este orden: punto inicial → punto de tránsito → punto final.
- **MVR2:** Se define un punto inicial, un punto final y un punto de referencia, el robot se mueve en este orden: punto inicial → punto final, sin pasar por el punto de referencia.
- **MVR3:** Se define un punto inicial, un punto central y un punto final, el robot se mueve en este orden: punto inicial → punto final, por el camino más corto de la circunferencia siendo el centro el punto central.
- **MVC:** Se define un punto inicial/final, un punto de tránsito 1 y un punto de tránsito 2, el robot se mueve en este orden: punto inicial → punto de tránsito 1 → punto de tránsito 2 → punto final.

Algunos ejemplos de declaraciones son:

MVR P1, P2, P3	' Se mueve con interpolación circular: P1→P2→P3.
MVR P1, P2, P3 WTH M_OUT (17) = 1	' Se mueve con interpolación circular: P1→P2→P3, y a la vez, habilita la salida 17.
MVR P1, P2, P3 WTHIF M_IN (20) = 1, SKIP	' Si la entrada 20 se activa durante el movimiento, se para y sigue en el siguiente paso.
MVR2 P1, P2, P3	' Se mueve con interpolación circular: P1→P2, sin pasar por P3.
MVR3 P1, P2, P3	' Se mueve con interpolación circular: P1→P2, por el camino más corto con P3 como centro.
MVC P1, P2, P3	' Se mueve con interpolación circular: P1→P2→P3→P1

A continuación, se muestra un ejemplo empleando el uso de las instrucciones MVR y MVC; en la figura 3.13 se visualiza de manera gráfica los movimientos de una tarea a realizar mientras que en la tabla 3.5 se describe el código de esta.

³⁹ Ver manual "Detailed explanation of functions and operations", p. 4-63.

Ejemplo:

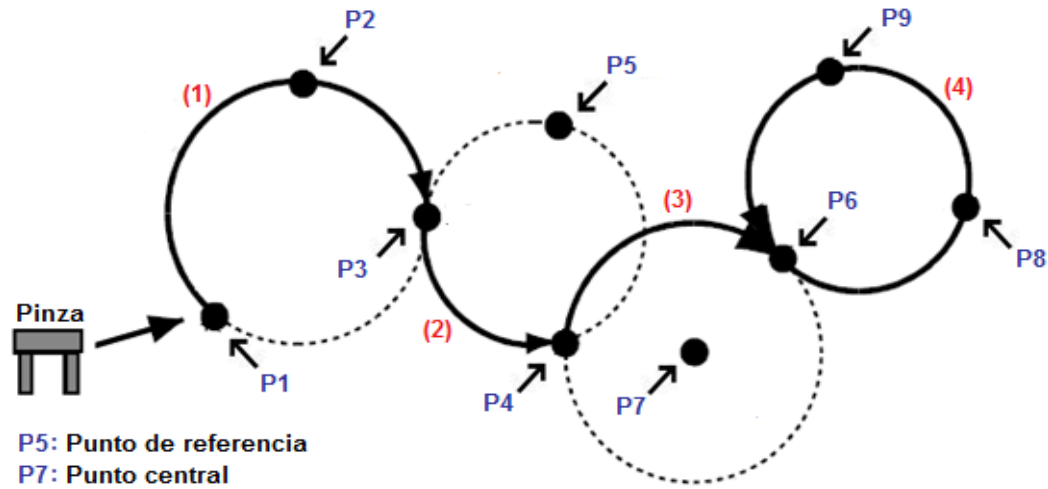


Figura 3.13: Interpolación circular.

Programa	Descripción
10 MVR P1, P2, P3	(1) Se mueve de P1→P2→P3 con interpolación circular formando un arco.
20 MVR2 P3,P4,P5	(2) Se mueve de P3→P4 con interpolación circular formando un arco sin pasar por P5.
30 MVR3 P4,P6,P7	(3) Se mueve de P4→P6 con interpolación circular formando un arco por el camino más corto tomando a P7 como centro.
40 MVC P6,P8,P9	(4) Se mueve de P6→P8→P9→P6 con interpolación circular formando un círculo.
50 END	' Fin del programa.

Tabla 3.5: Código de ejemplo de interpolación circular.

3.2.1.4 Movimiento continuo (CNT)

Esta instrucción permite realizar un movimiento continuo⁴⁰ entre puntos definidos por las instrucciones MOV, MVS, MVR y MVC, es decir, sin aceleraciones ni deceleraciones. Algunos ejemplos de declaraciones son:

```

CNT 1           ' Habilita movimiento continuo.
CNT 1, 100, 200 ' Define el inicio del movimiento continuo, la distancia de separación con el
                  punto inicial (100mm), y la distancia de separación con el punto final (200mm).
CNT 0           ' Deshabilita movimiento continuo.
  
```

A continuación, se muestra un ejemplo empleando el uso de la instrucción MVS; en la figura 3.14 se visualiza de manera gráfica los movimientos de una tarea a realizar mientras que en la

⁴⁰ Ver manual "Detailed explanation of functions and operations", p. 4-65.

tabla 3.6 se describe el código de esta.

Ejemplo:

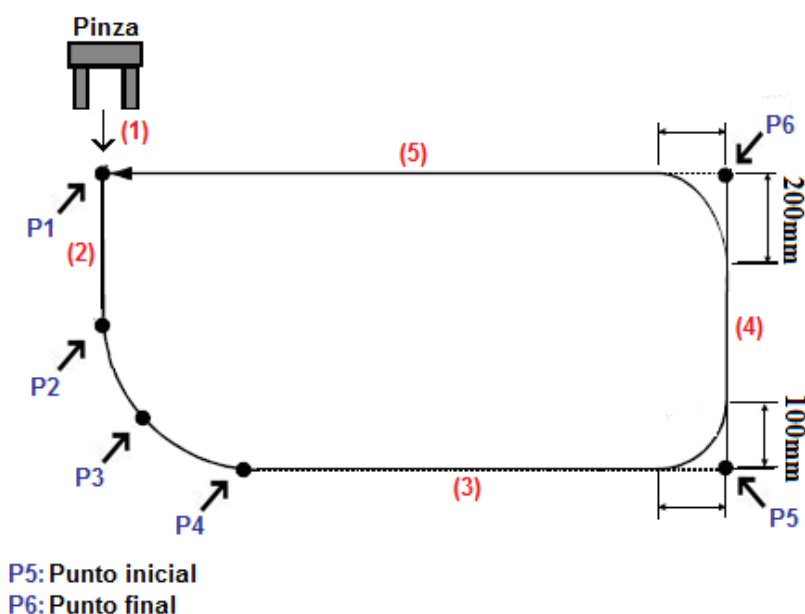


Figura 3.14: Movimiento continuo.

Programa	Descripción
10 MOV P1	(1) Se mueve con interpolación angular a P1.
20 CNT 1	' Habilita movimiento continuo.
30 MVR P2,P3,P4	(2) Se mueve de P2→P3→P4 con interpolación circular formando un arco.
40 MVS P5	(3) Se mueve en línea recta a P5.
50 CNT 1,100,200	' Define el inicio del movimiento continuo, la distancia de separación con P5 es de 100mm, y la distancia de separación con el P6 es de 200mm.
60 MVS P6	(4) Se mueve en línea recta a P6.
70 MVS P1	(5) Se mueve en línea recta a P1.
80 CNT 0	' Deshabilita movimiento continuo.
90 END	' Fin del programa.

Tabla 3.6: Código de ejemplo de movimiento continuo.

3.2.2 Instrucciones de velocidad y aceleración/desaceleración

La velocidad y la aceleración/desaceleración⁴¹ de los servomotores se pueden modificar con las siguientes instrucciones:

- **ACCEL:** Designa el porcentaje (%) de la aceleración y desaceleración respecto a las máximas permitidas.

⁴¹ Ver manual "Detailed explanation of functions and operations", p. 4-66.

- **OVRD:** Designa la velocidad de trabajo del robot en porcentaje (%) respecto a la velocidad máxima.
- **JOVRD:** Designa la velocidad de interpolación de ejes en porcentaje (%) respecto a la máxima permitida.
- **SPD:** Designa la velocidad en mm/s de la interpolación lineal y circular de la pinza o efecto final.

Algunos ejemplos de declaraciones son:

ACCEL	' Establece la aceleración y la desaceleración al 100%.
ACCEL 60, 80	' Establece la aceleración al 60% y la desaceleración al 80%. (Para máxima aceleración/desaceleración es de 0,2seg.; aceleración $0.2/0.6 = 0,33\text{seg.}$, desaceleración $0.2/0.8 = 0,25\text{seg.}$).
OVRD 50	' Establece las interpolaciones de ejes, lineal y circular al 50% de la velocidad máxima.
JOVRD 70	' Establecer la interpolación de ejes al 70% de la velocidad máxima.
SPD 30	' Establece la velocidad de la interpolación lineal y la interpolación circular a 30mm/s.

A continuación, se muestra un ejemplo empleando el uso de las instrucciones ACCEL, OVRD, JOVRD y SPD, en la figura 3.15 se visualiza de manera gráfica los movimientos de una tarea a realizar mientras que en la tabla 3.7 se describe el código de esta.

Ejemplo:

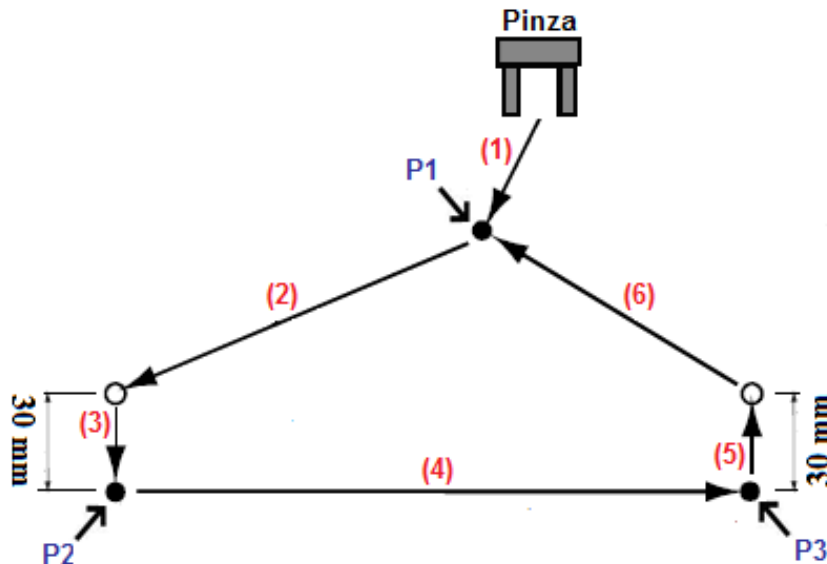


Figura 3.15: Control de velocidad.

Programa	Descripción
10 OVRD 100	' Fija la velocidad de trabajo al 100%.
20 MOV P1	(1) Se mueve con interpolación angular a P1 con velocidad máxima.
30 MOV P2,-50	(2) Se mueve con interpolación angular a velocidad máxima y en línea recta a la posición 50mm retraída de P2 en dirección de la pinza.
40 OVRD 50	' Establece la velocidad de trabajo a la mitad de la velocidad máxima.
50 MVS P2	(3) Se mueve linealmente a P2 con la mitad de la velocidad previamente establecida.
60 SPD 120	' Establece la velocidad de la pinza de 120mm /s. (Puesto que el reemplazo es del 50%, lo que realmente se mueve a 60mm /s.)
70 OVRD 100	' Fija la velocidad de trabajo al 100%.
80 ACCEL 70,70	' Aceleración y desaceleración al 70% de la velocidad máxima.
90 MVS P3	(4) Se mueve en línea recta a P3 con velocidad final en pinza de 120mm/s.
100 SPD M_NSPD	' Devuelve la velocidad final con el valor predeterminado.
110 JOVRD 70	' Velocidad de interpolación angular al 70%
120 ACCEL	' Fija la aceleración y deceleración al 100%
130 MVS, -50	(5) Se retrae linealmente 50mm de P3 en dirección de la pinza, con la velocidad predeterminada.
140 MOV P1	(6) Se traslada con interpolación angular a P1 en el 70% de la velocidad máxima.
150 END	' Fin del programa.

Tabla 3.7: Código de ejemplo de control de velocidad.

3.2.3 Instrucciones de control de la pinza

Abre o cierra la pinza⁴² designada. Las declaraciones que se ocupan en el presente trabajo son:

```
HOPEN 1      ' Abre la pinza 1
HCLOSE 1     ' Cierra pinza 1
```

A continuación, se muestra un ejemplo empleando el uso de la instrucción de control de la pinza, en la figura 3.16 se visualiza de manera gráfica los movimientos de una tarea a realizar mientras que en la tabla 3.7 se describe el código de esta.

⁴² Ver manual "Detailed explanation of functions and operations", p. 4-70.

Ejemplo:

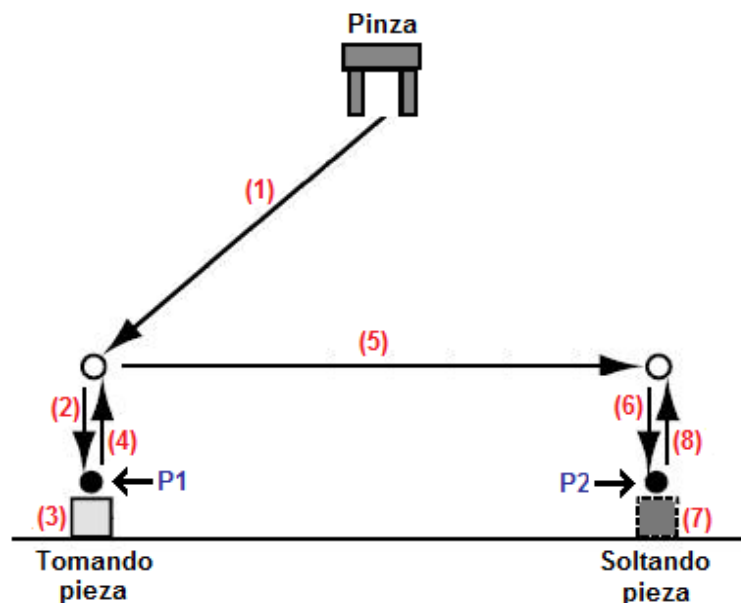


Figura 3.16: Control de pinza.

Programa	Descripción
10 MVS P1,-50	(1) Se mueve en línea recta a la posición 50mm retraída de P1 en dirección de la pinza.
20 OVRD 50	' Fija la velocidad de trabajo a la mitad de la velocidad máxima.
30 MVS P1	(2) Se mueve en línea recta a P1 (preparado para tomar la pieza).
40 DLY 0.5	' Espera 0.5 segundos.
50 HCLOSE 1	(3) Cierra la pinza (toma la pieza).
60 DLY 0.5	' Espera 0.5 segundos.
70 OVRD 100	' Fija la velocidad de trabajo al 100%.
80 MVS ,-50	(4) Se mueve en línea recta a la posición 50mm retraída de P1 en dirección de la pinza.
90 MVS P2,-50	(5) Se mueve en línea recta a la posición 50mm retraída de P2 en dirección de la pinza.
100 OVRD 50	' Fija la velocidad de trabajo al 100%.
110 MVS P2	(6) Se mueve linealmente a P2 (preparado para colocación de pieza).
120 DLY 0.5	' Espera 0.5 segundos.
130 HOPEN 1	(7) Abre la pinza (suelta la pieza),
140 DLY 0.5	' Espera 0.5 segundos.
150 OVRD 100	' Fija la velocidad de trabajo al 100%.
160 MVS ,-50	(8) Se mueve en línea recta a la posición 50mm retraída de P2 en dirección de la pinza.
170 END	' Fin del programa.

Tabla 3.7: Código de ejemplo de control de pinza.

3.2.4 Instrucciones de paletizado

Esta función desarrolla un movimiento programado en filas y columnas, lleva a cabo la colocación o la extracción de piezas semejantes en forma ordenada, a esto se le conoce como paletizado o despaletizado⁴³, respectivamente. Además esta función se puede utilizar para obtención de nuevas posiciones.

- **DEF PLT:** Define la operación de paletizado a utilizar.
- **PLT:** Obtiene una posición deseada de una operación de paletizado.

Algunos ejemplos de declaraciones son:

- DEF PLT 1, P1, P2, P3, P4, 4, 3, 1 ' Define operación de paletizado tipo 1 (zig-zag), con posición de inicio=P1, el punto final A=P2, el punto final B=P3 y punto diagonal=P4, el paletizado será de 4x3, con un total de 12 posiciones de trabajo. Ver figura 3.17a.
- DEF PLT 2, P1, P2, P3, , 8, 5, 2 ' Define operación de paletizado tipo 2 (misma dirección), con posición de inicio=P1, el punto final A=P2, y el punto final B=P3, el paletizado será de 8x5, con un total de 40 posiciones de trabajo. Ver figura 3.17b.
- DEF PLT 3, P1, P2, P3, , 5, 1, 3 ' Define operación de paletizado tipo 3 (arco), con posición de inicio=P1, punto de tránsito=P2 y el punto final=P3, el paletizado será en forma de arco con un total de 5 posiciones de trabajo. Ver figura 3.17c.
- (PLT1, 5) ' Quinta posición de paletizado tipo 1.
- (PLT1, M1) ' Posición M1 del paletizado tipo 1, donde M1 es una variable numérica.

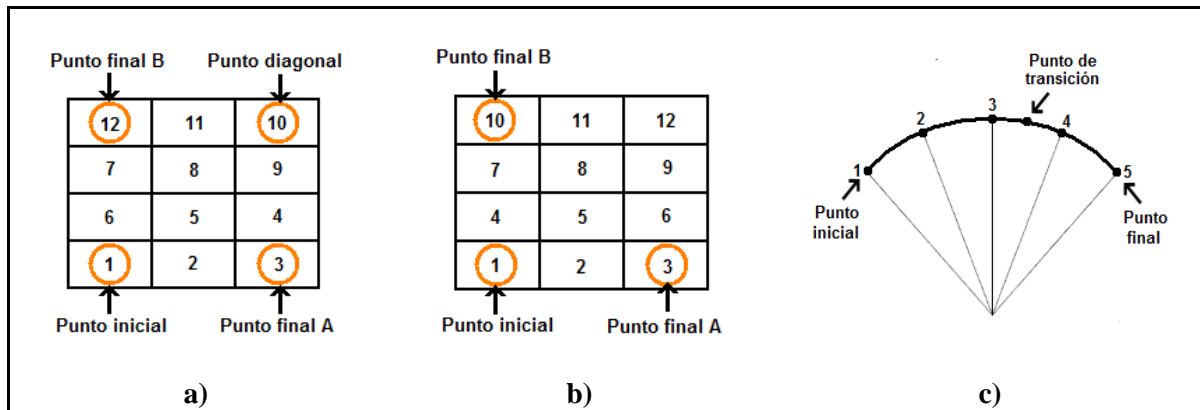


Figura 3.17: Tipos de paletizado: a) Paletizado tipo 1 (zig-zag), b) Paletizado tipo 2 (misma dirección), c) Paletizado tipo 3 (arco).

A continuación se muestra un ejemplo empleando el uso de la instrucción de control de la pinza, en la figura 3.18 se visualiza de manera gráfica los movimientos de una tarea a realizar mientras que en la tabla 3.8 se describe el código de esta.

⁴³ Ver manual "Detailed explanation of functions and operations", p. 4-71.

Ejemplo:

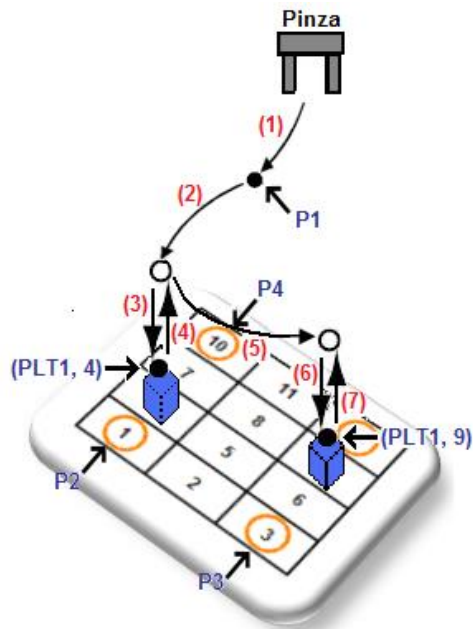


Figura 3.18: Operación de paletizado.

Programa	Descripción
10 DEF PLT 1, P2, P3, P4, , 3, 4, 2	' Define operación de paletizado tipo 2 (misma dirección), con posición de inicio=P2, el punto final A=P3, y el punto final B=P4, el paletizado será de 3x4, con un total de 12 posiciones de trabajo.
20 MOV P1	(1) Se mueve con interpolación angular a P1 con interpolación de ejes.
30 HCLOSE 1	' Cierra la pinza (toma la pieza).
40 P10=(PLT 1, 4)	' Define la posición de la celda 4 del paletizado como la posición P10.
50 MOV P10, -50	(2) Se mueve con interpolación angular a la posición 50mm retraída de P10 en dirección de la pinza.
60 MVS P10	(3) Se mueve a la posición P10 previamente definida.
70 HOPEN 1	' Abre pinza (suelta pieza).
80 MVS P10, -50	(4) Se regresa en línea recta a la posición 50mm retraída de la posición P10 en dirección de la pinza.
90 P10=(PLT 1, 9)	' Define la posición de la celda 9 del paletizado como la nueva posición P10.
100 MOV P10, -50	(5) Se mueve con interpolación angular a la posición 50mm retraída de P10 en dirección de la pinza.
110 MVS P10	(6) Se mueve a la posición P10 previamente definida.
120 HCLOSE 1	' Cierra la pinza (toma la pieza).
130 MVS P10, -50	(7) Se regresa en línea recta a la posición 50mm retraída de P10 en dirección de la pinza.
140 END	' Fin del programa.

Tabla 3.8: Código de ejemplo de operaciones de paletizado.

3.2.5 Estructuras de control

La secuencia del programa se puede controlar mediante saltos, interrupciones, llamadas a subrutinas o paros⁴⁴.

- **GOTO:** Salta incondicionalmente a la línea designada.
- **ON GOTO:** Salta de acuerdo al valor de la variable designada. El valor de las condiciones de la variable siguen el siguiente orden 0, 1, 2, 3,...
- **IF THEN ELSE:** (Instrucción escrita en una sola línea). Ejecuta el comando correspondiente a las condiciones designadas. El valor de las condiciones pueden ser designadas aleatoriamente. Si la condición se cumple se ejecuta la instrucción que se encuentra después del THEN y antes del ELSE. Si la condición no se cumple se ejecuta la instrucción después del ELSE. Ambas son escritas en una sola línea.
- **IF THEN/ELSE/END IF:** (Instrucciones escritas en varias líneas). Varias líneas pueden ser procesadas de acuerdo a las variables especificadas. Es posible especificar cualquier condición de variables. Si se cumplen las condiciones, las instrucciones que están después del THEN y antes del ELSE se ejecutan. Si las condiciones no se cumplen, las instrucciones que se encuentran después del ELSE y antes del END IF son ejecutadas.
- **SELECT/CASE/END SELECT:** Salta de acuerdo a la variable especificada y a las condiciones designadas de dicho valor. El valor de las condiciones pueden ser asignados aleatoriamente. Múltiples tipos de condiciones pueden ser propuestos por declaración. Para finalizar los CASE, se puede poner BREAK, en el último se puede poner DEFAULT, que englobará el resto de condiciones que no se hayan especificado antes de los CASE anteriores. El programa debería salir del SELECT CASE pasando por la línea END SELECT, ya que si sólo sale con GOTO o GOSUB, acaba parándose por anidamiento excesivo.
- **WAIT:** Espera a que una variable se cumpla para ejecutar la siguiente línea de código.

A continuación se muestran algunos ejemplos de declaraciones:

GOTO 200	' Salta incondicionalmente a la línea 200.
GOTO * FIN	' Salta incondicionalmente a la línea con etiqueta FIN.
ON M1 GOTO 100, 200, 300	' Si la variable M1 es 1 salta a la línea 100; si es 2 salta a la línea 200; y si es 3 salta a la línea 300. Si el valor no corresponde, procede a la siguiente línea de código.
IF M1=1 THEN 100	' Si la variable M1 es 1 salta a la línea 100. Si no, pasa al siguiente paso.
IF M1=1 THEN 100 ELSE 200	' Si la variable M1 es 1 salta a la línea 100. Si no, salta a la línea 200.
IF M1=1 THEN M2 = 1 M3 = 2	' Si la variable M1 es 1 entonces M2=1 y M3=2. Si M1 es diferente de 1, entonces M2=-1 y M3=-2.

⁴⁴ Ver manual "Detailed explanation of functions and operations", p. 4-73.

ELSE	
M2 =- 1	
M3 =- 2	
ENDIF	
SELECT M1	' Salta a la declaración correspondiente de CASE con el valor numérico de la variable M1.
CASE 10	' Si el valor es de 10, solo se ejecutan las instrucciones que estén entre CASE 10 y CASE 11.
:	
BRAKE	
CASE IS 11	' Si el valor es de 11, solo se ejecutan las instrucciones que estén entre CASE 11 y CASE IS <5.
:	
BRAKE	
CASE IS <5	' Si el valor es menor que 5, solo se ejecutan las instrucciones que estén entre CASE IS <5 y CASE 6 TO 9.
:	
BRAKE	
CASE 6 TO 9	' Si el valor es de entre 6 y 9, solo se ejecutan las instrucciones que estén entre CASE 6 TO 9 y DEFAULT.
:	
BRAKE	
DEFAULT	' Si el valor no corresponde a ninguna de las anteriores, solo se ejecutan las instrucciones que estén entre DEFAULT y END SELECT.
:	
BRAKE	
END SELECT	' Finaliza la instrucción SELECT CASE.
WAIT M_IN (1)=1	' Espera a que el bit 1 de la señal de entrada se ponga en alto.

3.2.6 Repeticiones

Las instrucciones de repetición⁴⁵ pueden ser ejecutadas en diversas ocasiones de acuerdo a las condiciones designadas.

- **FOR/NEXT:** Repite las instrucciones que se encuentran entre FOR y NEXT hasta satisfacer las condiciones especificadas.
- **WHILE/WEND:** Repite las instrucciones que se encuentran entre WHILE y WEND mientras se cumplan las condiciones designadas.

A continuación se muestran algunos ejemplos de declaraciones:

FOR M1=1 TO 10	' Repite 10 veces las instrucciones que se encuentran entre FOR y NEXT. El valor inicial de M1 es 1,
:	incrementándose M1 en uno en cada repetición.
NEXT	
FOR M1=0 TO 10 STEP 2	' Repite 6 veces las instrucciones que se encuentran entre FOR y NEXT. El valor inicial de M1 es 0 y este se incrementa en dos por cada repetición.
:	
NEXT	
WHILE (M1 >= 1) AND (M1 <= 10)	' Repite las instrucciones que se encuentran entre WHILE y WEND mientras el valor de la variable M1 pertenezca al rango [1..10].
:	
WEND	

⁴⁵ Ver manual "Detailed explanation of functions and operations", p. 4-75.

3.2.7 Interrupción

Las interrupciones⁴⁶ son instrucciones que como su nombre lo indica interrumpen al programa durante su ejecución al recibir o cumplirse una cierta condición. Cuando esto sucede se realiza una tarea especificada y una vez finalizada retorna al punto del programa donde se generó la interrupción.

- **DEF ACT:** Define las condiciones de interrupción y el proceso para generarla.
- **ACT:** Habilita o deshabilita la interrupción.
- **RETURN:** Si una subrutina es llamada por el proceso de interrupción, retorna a la línea de código donde se generó la interrupción.

A continuación se muestran algunos ejemplos de declaraciones:

DEF ACT 1, M_IN(10)=1 GOSUB 100	' Se define la interrupción 1, con la siguiente condición: si el bit 10 de la señal de entrada se pone en 1, será llamada la subrutina de la línea 100. Esta es llamada después de que el robot desacelere y se detenga.
DEF ACT 2, M_IN(11)=1 GOSUB 200, L	' Se define la interrupción 2, con la siguiente condición: si el bit 11 de la señal de entrada se pone en 1, será llamada la subrutina de la línea 200. Esta es llamada una vez finalizada la declaración que se está ejecutando.
DEF ACT 3, M_IN(12)=1 GOSUB 300, S	' Se define la interrupción 3, con la siguiente condición: si el bit 12 de la señal de entrada se pone en 1, será llamada la subrutina de la línea 300. Esta es llamada después de que el robot se desacelera y se detiene en el menor tiempo y distancia posible.
ACT 1=1	' Habilita interrupción con prioridad 1.
ACT 2=0	' Deshabilita interrupción con prioridad 2.
RETURN 0	' Retorna a la línea donde se generó la interrupción.
RETURN 1	' Retorna a la línea siguiente de la línea donde se generó la interrupción.

3.2.8 Subrutina

En la programación de la tarea se pueden usar subrutinas⁴⁷ y subprogramas, el uso de estas funciones reduce el número de pasos y pueden ser creadas con una estructura jerárquica para hacer más entendible el programa.

- **GOSUB:** Llama a la subrutina del número de línea o etiqueta de la línea designada.

⁴⁶ Ver manual "Detailed explanation of functions and operations", p. 4-76.

⁴⁷ Ver manual "Detailed explanation of functions and operations", p. 4-77.

- **ON GOSUB:** Llama a la subrutina de acuerdo al número de la variable designada. El valor de las condiciones de la variable siguen el siguiente orden 0, 1, 2, 3,...
- **RETURN:** Retorna a la siguiente línea donde se generó la llamada con el comando GOSUB.

A continuación se muestran algunos ejemplos de declaraciones:

```
GOSUB 100           ' Llama a la subrutina con número de línea 100.
ON GOSUB *GET      ' Llama a la subrutina con etiqueta *GET.
ON M1 GOSUB 100, 200, 300 ' Si el valor de la variable M1 es 1 salta a la línea 100, si es 2
                        ' salta a la línea 200 y si es 3 salta a la línea 300. Si el valor no
                        ' corresponde, procede a la siguiente línea de código.
RETURN             ' Retorna a la siguiente línea donde se generó la llamada con el
                  ' comando GOSUB.
```

3.2.9 Temporizador

El temporizador⁴⁸ sirve para que el programa pueda ser retardado por la asignación de un tiempo, al cumplir ese tiempo el programa continúa con la siguiente línea de código.

- **DLY:** Funciona como un tiempo designado por un temporizador.

```
DLY 0.05           ' Espera sólo 0.05 segundos.
M_OUT(10)=1 DLY 0.5 ' Pone en alto el bit 10 de la señal de salida, únicamente por 0.5
                    ' segundos.
```

3.2.10 Suspensión

La ejecución del programa puede ser suspendida⁴⁹. El movimiento del robot desacelerará hasta detenerse.

- **HLT:** Esta instrucción detiene el robot y detiene la ejecución del programa. Cuando el programa es iniciado se ejecuta nuevamente a partir de la siguiente línea donde se suspendió.
- **END:** Esta instrucción define el fin de un ciclo de un programa. En operación continua el programa se ejecuta nuevamente desde la línea de inicio. En la operación ciclo el programa termina con la ejecución de la instrucción END cuando se detiene el ciclo.

```
HLT               ' Interrumpe la ejecución del programa.
IF M_IN(20)=1 THEN HLT ' Detiene el programa si el bit 20 de la señal de entrada está
                       ' en alto.
MOV P1 WTHIF M_IN(18)=1, HLT ' Detiene el programa si el bit 18 de la señal de entrada se
```

⁴⁸ Ver manual "Detailed explanation of functions and operations", p. 4-78.

⁴⁹ *Ibíd.*, p. 4-79.

Clase	Signo	Significado	Ejemplo	Descripción
Sustitución	=	El lado derecho es sustituido por el lado izquierdo.	P1=P2 M1=1	Sustituye P2 en la variable de posición P1. Sustituye el valor 1 en la variable numérica M1.
Operaciones de valores numéricos	+	Suma	P10=P1+P2 MOV P8+P9 M1=M1+1	Sustituye el resultado de la operación sumatoria de las coordenadas P1 y P2 en la variable de posición P10. Mueve la posición obtenida de la sumatoria de los elementos de las coordenadas de P8 y P9. Suma 1 a la variable numérica M1.
	-	Resta	P10=P1-P2 MOV P8-P9 M1=M1-1	Sustituye el resultado obtenido por la resta de las coordenadas P2 en P1 y lo guarda en la variable de posición P10. Mueve la posición obtenida de la resta de P9 en P8. Resta 1 a la variable numérica M1.
	*	Multiplicación	P1=P10*P3 M1=M1*5	Sustituye el resultado del producto cruz entre P10 y P3, en la variable P1. Realiza el producto punto de la variable numérica M1 por 5.
	/	División	P1=P10/P3 M1=M1/2	Sustituye el resultado de la división (inversa del producto cruz), entre P10 y P3 en la variable P1. Divide la variable numérica M1 entre 2.
	^	Exponencial	M1=M1^2	Eleva al cuadrado la variable numérica M1.
	\	División entera	M1=M1\3	Divide el valor de la variable numérica M1 entre 3 y el resultado lo vuelve entero (redondeo hacia abajo).
	MOD	Módulo	M1=M1 MOD 3	Divide el valor de la variable numérica M1 entre 3, sin redondear
	-	Signo inverso	P1=-P1 M1=-M1	Invierte el signo de cada elemento de la coordenada en la variable de la posición P1. Invierte el signo del valor de la variable numérica M1.

Tabla 3.9: Expresiones y operaciones.

Clase	Signo	Significado	Ejemplo	Descripción
Operaciones de comparación	=	Igual a	IF M1=1 THEN 200	' Salta a la línea 200 si el valor de la variable numérica M1 es 1.
	<> or ><	Diferente a	IF M1<>2 THEN 300	' Salta a la línea 300 si el valor de la variable numérica M1 es diferente de 2.
	<	Menor que	IF M1< 10 THEN 300	' Salta a la línea 300 si el valor de la variable numérica M1 es menor que 10.
	>	Mayor que	IF M1>9 THEN 200	' Salta a la línea 200 si el valor de la variable numérica M1 es mayor que 9.
	=< or <=	Igual o menor que	IF M1<=10 THEN 200	' Salta a la línea 200 si el valor de la variable numérica M1 es menor o igual que 9.
	=> or >=	Igual o mayor que	IF M1=>11 THEN 200	' Salta a la línea 200 si el valor de la variable numérica M1 es mayor o igual que 11.
Operaciones lógicas	AND	y	M1=M_INB(1) AND &H0F	' Realiza la operación lógica AND entre los bits 1 al 8 de la señal de entrada con el valor hexadecimal 0F.
	OR	o	M_OUTB(20)=M1 OR &H80	' Saca el valor de la variable numérica M1 por los bits del 20 al 27 de la variable de salida, donde el bit 27 siempre está en 1 por la operación OR con el valor hexadecimal 80.
	NOT	Negación	M1=NOT M_INW(1)	' Invierte el estado de los bits del 1 al 16 de la señal de entrada y lo sustituye en la variable numérica M1.
	XOR	OR exclusiva	N2=M1 XOR M_INW(1)	' Obtiene la OR exclusiva de los estados de M1 y los bits 1 al 16 de la señal de entrada, lo convierte en un valor y lo sustituye en la variable numérica M2.
	<<	Desplazamiento a la izquierda	M1=M1<<2	' Desplaza la variable numérica M1 dos bits hacia la izquierda.
	>>	Desplazamiento a la derecha	M1=M1>>1	' Desplaza la variable numérica M1 un bit hacia la derecha.

Tabla 3.10: Continuación de expresiones y operaciones.

3.2.13 Líneas de instrucción

El programa se construye en base a líneas de instrucción que se ejecutarán de manera secuencial excepto cuando haya instrucciones de salto, bucles o llamadas a subrutina. Los elementos que las forman se muestran a continuación en la figura 3.19.

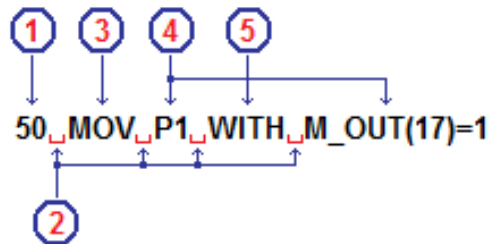


Figura 3.19: Elementos de líneas de instrucción.

- 1. Número de línea:** Es el número usado para determinar el orden de ejecución del programa. El programa se ejecuta en orden ascendente.
- 2. Espacio:** El caracter de espacio es requerido como un delimitador inmediatamente después de un número de línea o una palabra de comando y entre objetos de datos.
- 3. Mnemotécnico de la instrucción:** Es el comando que designa la operación a realizar por el robot.
- 4. Dato:** El dato puede ser un valor o una variable del tipo requerido por la instrucción.
- 5. Declaración añadida:** Solo se usa si es necesario.


Notas:

- Los caracteres en minúsculas se leerán como minúsculas cuando sean usados en comentarios o en datos de cadenas de caracteres. En todos los otros casos, estos serán convertidos a letras mayúsculas cuando el programa sea leído.
- El guión bajo (_) es usado para segundos caracteres de un identificador (nombre de variable) para identificar la variable como una variable externa entre programas.
Ejemplo M_01, M_ABC.
- El apóstrofe (') es utilizado para iniciar las líneas de comentarios.
Ejemplo: 100 MOV P1 'Pieza
donde "Pieza" es el comentario.
- El asterisco se coloca delante de los nombres de etiqueta utilizada como destino de salto. Ejemplo 200 *CHECK
- La coma es usada para delimitar cuando hay muchos parámetros o sufijos.
Ejemplo: P1 = (200, 150,)

Dependiendo el tipo de variable a utilizar, estos pueden ser:

- **Posición:** Corresponde a las coordenadas ortogonales de la posición. El nombre de la variable debe comenzar con la letra P.
- **Ángulos de eje:** Corresponde a los ángulos de las articulaciones del robot. El nombre de la variable debe comenzar con la letra J.
- **Aritmética:** Corresponde al valor numérico real. El nombre debe comenzar con la letra M.
- **Caracter:** Corresponde a las cadenas de letras. El nombre de la variable debe terminar con el caracter \$.

3.3 Pasos para la elaboración de prácticas con el robot industrial RV-2AJ

1. Conectar el cable de comunicación serial entre el controlador CR1-571 y la PC⁵².
2. Poner la llave física en un puerto USB previamente configurado⁵³.
3. Crear un nuevo proyecto⁵⁴.
4. Tomar posiciones⁵⁵.
5. Crear el programa en la ventana Robot Program. Este se puede hacer cuando se hayan tomado todas las posiciones, o bien, mientras se van metiendo posiciones en la ventana de posiciones.
6. Renumerar el programa haciendo click en el icono Renumber , esto se hace para evitar errores de sintaxis.
7. Actualizar datos del controlador CR1-571 en la PC haciendo click izquierdo en cada subdivisión de RV-2AJ que se encuentra dentro de la ventana RCI Explorer, ver figura 3.20. No es necesario dar click en Error List a menos que se quiera saber la lista de errores⁵⁶ obtenidos durante la elaboración del programa.

Nota:

- Si este paso es omitido, se generará error y por lo tanto, no se podrá simular el programa en la ventana Workcell View, por consiguiente, no se podrá descargar el programa al controlador.

⁵² Ver apéndice C: “Instalación del programa y configuración de la comunicación”, p. 93.

⁵³ *Ibíd.*, p. 90.

⁵⁴ Ver en este trabajo el tema “Nuevo proyecto”, p. 41.

⁵⁵ Ver en este trabajo el tema “Toma de posiciones”, p. 44.

⁵⁶ Ver apéndice D: “Lista de errores”, p. 95.

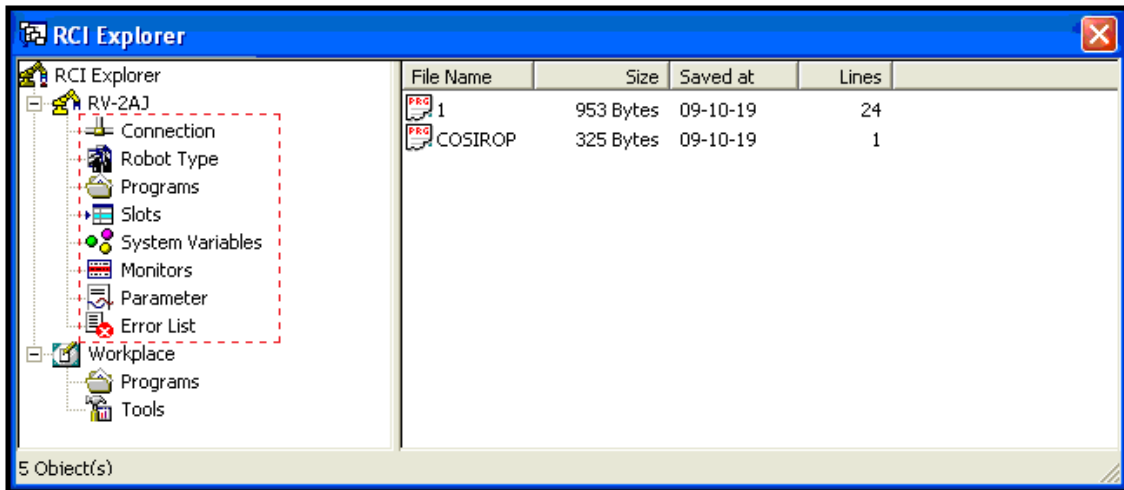







Figura 3.20: Actualizando datos del controlador en la PC.

8. Compilar el programa haciendo click en el icono Compile+Link . Si en la ventana Messages se notifica algún error, se debe volver a revisar la programación. Cada vez que se modifique el programa debe dar click en el icono Renumber y volver a realizar este paso. Si en la ventana Messages no se notifica ningún error, pasar al paso 9.
9. Simular programa, esto se realiza haciendo click en el icono Start . El brazo de simulación que se encuentra en la ventana Workcell View ejecutará la tarea programada. Si se quiere detener la simulación, dar click en el icono Stop . Si se quiere realizar la simulación de programa paso a paso dar click en los iconos New Step  o Previous Step , ya sea para ver el paso siguiente o el paso anterior, respectivamente.
Si la simulación ejecuta el programa como se desea, entonces realizar el paso 10, de lo contrario, se debe modificar el programa inclusive la lista de posiciones si es necesario. Volver a reenumerar y pasar al paso 8.
10. Descargar el programa al controlador.
11. Ejecutar el programa en el brazo robótico RV-2AJ.

Notas:

- Si es la primera vez que se va a ejecutar el programa, la velocidad del controlador debe ser forzosamente a 5, 10 o 20 como máximo.
- Si se ejecuta un programa modificado, la velocidad del controlador debe ser de 5, 10 o 20 como máximo.
- Si se ha movido o añadido alguna parte de la unidad que se encuentre dentro del área de trabajo del brazo robótico, que altere la ejecución del programa o los resultados obtenidos se deberá disminuir la velocidad a 5, 10 o 20, para comprobar que no hay colisiones y que la tarea se realiza de manera adecuada.
- Si el programa ejecutado, no realiza la tarea adecuadamente entonces debe revisar el programa, la lista de posiciones, reenumerar, compilar, simular y descargar el programa, para volver a ejecutarlo en el

brazo robótico.

- Si el programa se ha ejecutado satisfactoriamente a velocidad de 5, 10 o 20, se puede incrementar dicha velocidad.
- Si durante la ejecución del programa se nota una posible colisión, activar inmediatamente el paro de emergencia del T/B o del controlador.
- Llevar a cabo las medidas de seguridad.

IV

PRÁCTICAS

En este capítulo se exponen algunas de las herramientas mencionadas anteriormente, aplicadas en siete prácticas diferentes las cuales se basan en los movimientos de interpolación que utiliza el brazo robótico RV-2AJ.

En cada una de las prácticas se describen las posiciones y el código del programa, como una secuencia recomendada para que el alumno pueda basarse en sus próximas aplicaciones.

Cabe mencionar que las posiciones y los programas de estas prácticas están comentados en su totalidad para facilitar su comprensión, además de ser acompañadas por imágenes que plasman el objetivo de la tarea.

Práctica 1

Interpolación angular (MOV)

OBJETIVO: Realizar un programa para que el brazo industrial RV-2AJ tome un lapicero y dibuje en papel la trayectoria con interpolación angular generada con la instrucción MOV⁵⁷.

Las posiciones que se ocupan para la elaboración de esta, se muestran en la tabla 4.1:

Posición	X	Y	Z	A/P	B/R	Referencia
P1	-0.88	163.19	451.59	-88.49	179.03	Posición inicial
P2	221.43	166.23	231.75	-172.21	178.98	Arriba del lápiz
P3	221.44	166.23	156.86	-172.21	178.98	Toma lápiz
P4	221.44	166.23	173.56	-172.21	178.98	Deja lápiz
P5	-152.59	305.67	357.68	-105.46	178.99	Arriba de libreta
P6	-129.65	311.44	170.15	-105.46	178.99	Arriba de P_1
P7	-129.65	311.44	152.15	-105.46	178.99	P_1
P8	-129.65	279.09	152.15	-105.46	178.99	P_2
P9	-178.46	279.09	152.15	-105.46	178.99	P_3
P10	-178.46	311.44	152.15	-105.46	178.99	P_4

Tabla 4.1: Posiciones del brazo industrial RV-2AJ para la práctica 1.

El código del programa es:

```

10 OVRD 40      ' Designa velocidad del robot
20 MOV P1      ' Mueve el brazo a la posición inicial
30 HCLOSE 1    ' Cierra pinza
40 MOV P2      ' Se posiciona arriba del lápiz
50 HOPEN 1     ' Abre pinza
60 OVRD 20     ' Se reduce la velocidad
70 MVS P3      ' Se acerca para tomar el lápiz
80 HCLOSE 1    ' Cierra pinza (toma lápiz)
90 OVRD 20     ' Aumenta velocidad
100 MVS P2     ' Retira el lápiz de su base (hacia arriba)
110 MOV P1     ' Mueve a posición inicial
120 MOV P5     ' Se posiciona arriba de la libreta
130 OVRD 10    ' Disminuye la velocidad
140 MOV P6     ' Se posiciona arriba del primer punto (P_1)
150 MOV P7     ' -----
160 MOV P8     ' Genera un rectángulo entre estos cuatro puntos
170 MOV P9     ' para visualizar la interpolación joint

```

⁵⁷ Ver en este trabajo el tema “Interpolación angular (MOV)”, p. 49.


```

180 MOV P10      '(MOV)
190 MOV P7      '-----
200 OVRD 40     ' Aumenta la velocidad
210 MOV P5      ' Retira el lápiz de la libreta (hacia arriba)
220 MOV P1      ' Mueve a posición inicial
230 MOV P2      ' Se posiciona la pinza arriba del la base del lápiz (para depositarlo)
240 OVRD 10     ' Reduce la velocidad
250 MVS P4      ' Se prepara para dejar lápiz
260 HOPEN 1     ' Abre pinza (deposita lápiz)
270 OVRD 20     ' Aumenta velocidad
280 MVS P2      ' Se retira la pinza de la base del lápiz (hacia arriba)
290 MOV P1      ' Mueve a posición inicial
300 END         ' Fin del programa

```

En la figura 4.1 se muestra la sujeción del lapicero y el dibujo de la trayectoria con la interpolación angular, respectivamente.

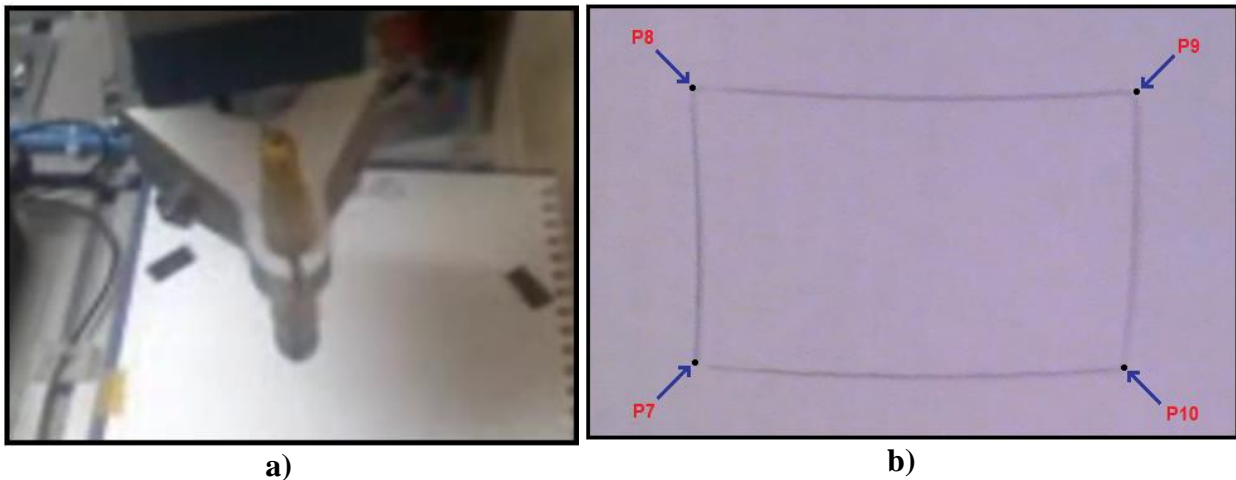


Figura 4.1: *Práctica 1, a) Sujeción del lapicero por pinza del robot RV-2AJ, b) Dibujo de trayectoria con interpolación angular.*

Práctica 2

Interpolación lineal (MVS)

OBJETIVO: Realizar un programa para que el brazo industrial RV-2AJ tome un lapicero y dibuje en papel la trayectoria con interpolación lineal generada con la instrucción MVS⁵⁸.

Las posiciones que se ocupan para la elaboración de esta, se muestran en la tabla 4.2:

Posición	X	Y	Z	A/P	B/R	Referencia
P1	-0.88	163.19	451.59	-88.49	179.03	Posición inicial
P2	221.43	166.23	231.75	-172.21	178.98	Arriba del lápiz
P3	221.44	166.23	156.86	-172.21	178.98	Toma lápiz
P4	221.44	166.23	173.56	-172.21	178.98	Deja lápiz
P5	-152.59	305.67	357.68	-105.46	178.99	Arriba de libreta
P6	-129.65	311.44	170.15	-105.46	178.99	Arriba de P_1
P7	-129.65	311.44	152.15	-105.46	178.99	P_1
P8	-129.65	279.09	152.15	-105.46	178.99	P_2
P9	-178.46	279.09	152.15	-105.46	178.99	P_3
P10	-178.46	311.44	152.15	-105.46	178.99	P_4

Tabla 4.2: Posiciones del brazo industrial RV-2AJ para la práctica 2.

El código del programa es:

```

10 OVRD 40      ' Designa velocidad
20 MOV P1      ' Mueve el brazo a la posición inicial
30 HCLOSE 1    ' Cierra pinza
40 MOV P2      ' Se posiciona arriba del lápiz
50 HOPEN 1     ' Abre pinza
60 OVRD 20     ' Se reduce la velocidad
70 MVS P3      ' Se acerca para tomar el lápiz
80 HCLOSE 1    ' Cierra pinza (toma lápiz)
90 OVRD 20     ' Aumenta velocidad
100 MVS P2     ' Retira el lápiz de su base (hacia arriba)
110 MOV P1     ' Mueve a posición inicial
120 MOV P5     ' Se posiciona arriba de la libreta
130 OVRD 10    ' Disminuye la velocidad
140 MOV P6     ' Se acerca y posiciona arriba del primer punto (P_1)

150 MVS P7     ' -----
160 MVS P8     ' Genera un rectángulo entre estos cuatro puntos

```

⁵⁸ Ver en este trabajo el tema “Interpolación lineal (MVS)”, p. 50.

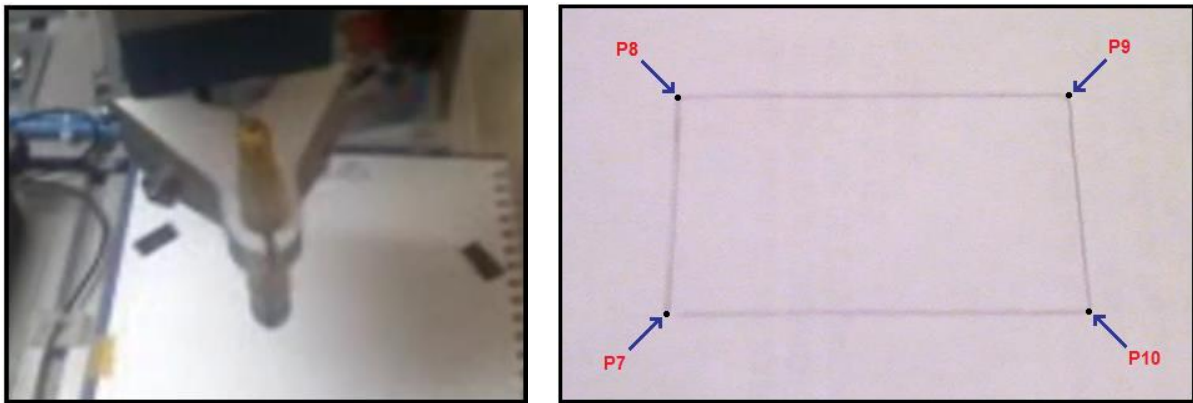
```

170 MVS P9      ' para visualizar la interpolación lineal
180 MVS P10    ' (MVS)
190 MVS P7      ' -----

200 OVRD 40    ' Aumenta la velocidad
210 MOV P5     ' Retira el lápiz de la libreta (hacia arriba)
220 MOV P1     ' Mueve a posición inicial
230 MOV P2     ' Se posiciona la pinza arriba del la base del lápiz (para depositarlo)
240 OVRD 10    ' Reduce velocidad
250 MVS P4     ' Se prepara para dejar lápiz
260 HOPEN 1    ' Abre pinza (deposita lápiz)
270 OVRD 20    ' Aumenta velocidad
280 MVS P2     ' Se retira la pinza de la base del lápiz (hacia arriba)
290 MOV P1     ' Mueve a posición inicial
300 END        ' Fin del programa

```

En la figura 4.2 se muestra la sujeción del lapicero y el dibujo de la trayectoria con la interpolación lineal, respectivamente.



a)
b)
Figura 4.2: Práctica 2, a) Sujeción del lapicero por pinza del robot RV-2AJ,
 b) Dibujo de trayectoria con interpolación lineal.

Práctica 3

Interpolación circular (MVR y MVC)

OBJETIVO: Realizar un programa para que el brazo industrial RV-2AJ tome un lapicero y dibuje en papel la trayectoria con interpolación circular generada con las instrucciones MVR y MVC⁵⁹.

Las posiciones que se ocupan para la elaboración de esta, se muestran en la tabla 4.3:

Posición	X	Y	Z	A/P	B/R	Referencia
P1	-0.88	163.19	451.59	-88.49	179.03	Posición inicial
P2	221.43	166.23	231.75	-172.21	178.98	Arriba del lápiz
P3	221.44	166.23	156.86	-172.21	178.98	Toma lápiz
P4	221.44	166.23	173.56	-172.21	178.98	Deja lápiz
P5	-152.59	305.67	357.68	-105.46	178.99	Arriba de libreta
P6	-129.65	311.44	170.15	-105.46	178.99	Arriba de P_1
P7	-129.65	311.44	152.15	-105.46	178.99	P_1
P8	-129.65	279.09	152.15	-105.46	178.99	P_2
P9	-178.00	279.00	152.15	-105.46	178.99	P_3
P10	-178.46	311.44	152.15	-105.46	178.99	P_4
P20	-158.15	333.27	152.14	-105.46	178.99	Referencia 4
P21	-178.00	295.00	152.16	-105.46	178.99	Centro 3
P22	-151.78	295.99	152.15	-105.46	178.99	Referencia 1
P23	-129.70	295.25	152.15	-105.46	178.99	Referencia 2
P24	-194.00	295.00	152.15	-105.46	178.99	P_3 Referencia 5

Tabla 4.3: Posiciones del brazo industrial RV-2AJ para la práctica 3.

El código del programa es:

```

10 OVRD 50           ' Designa velocidad
20 MOV P1            ' Mueve el brazo a la posición inicial
30 HCLOSE 1         ' Cierra pinza
40 MOV P2            ' Se posiciona arriba del lápiz
50 HOPEN 1          ' Abre pinza
60 OVRD 20          ' Se reduce la velocidad
70 MVS P3            ' Se acerca para tomar el lápiz
80 HCLOSE 1         ' Cierra pinza (toma lápiz)
90 OVRD 20          ' Aumenta velocidad
100 MVS P2           ' Retira el lápiz de su base (hacia arriba)
110 MOV P1           ' Mueve a posición inicial
120 MOV P5           ' Se posiciona arriba de la libreta

```

⁵⁹ Ver en este trabajo el tema “Interpolación circular (MVR y MVC)”, p. 52.

130 OVRD 10	' Disminuye la velocidad
140 MVS P7,-30	' Se posiciona arriba del primer punto
150 MVS P7	' Baja el lápiz a la altura del papel
160 MVR P7,P22,P8	' Genera un semicírculo entre P7 y P8 utilizando P22 como punto de referencia por donde pasará el semicírculo
170 MVR2 P8,P9,P23	' Genera un semicírculo ente P8 y P9, utilizando P23 como punto de referencia sin pasar a través de él
180 MVR3 P9,P24,P21	' Genera un semicírculo entre P9 y P24, con centro en P21
190 MVS P24,-30	' -----
200 MVS P10,-30	' Se levanta el lápiz y se posiciona en P10
210 MVS P10	' -----
220 MVC P10,P7,P20	' Genera un circunferencia iniciando por P10 'utilizando P7 y P20 como puntos de transición
230 MVS P7,-30	' Levanta el lápiz
240 OVRD 50	' Aumenta la velocidad
250 MOV P5	' Retira el lápiz de la libreta (hacia arriba)
260 MOV P1	' Mueve a posición inicial
270 MOV P2	' Se posiciona el pinza arriba del 'la base del lápiz (para depositarlo)
280 OVRD 10	' Reduce velocidad
290 MVS P4	' Se prepara para dejar lápiz
300 HOPEN 1	' Abre pinza (deposita lápiz)
310 OVRD 20	' Aumenta velocidad
320 MVS P2	' Se retira la pinza de la base del lápiz (hacia arriba)
330 MOV P1	' Mueve a posición inicial
340 END	' Fin del programa

En la figura 4.3, se muestra la sujeción del lapicero y el dibujo de la trayectoria con la interpolación circular, respectivamente.

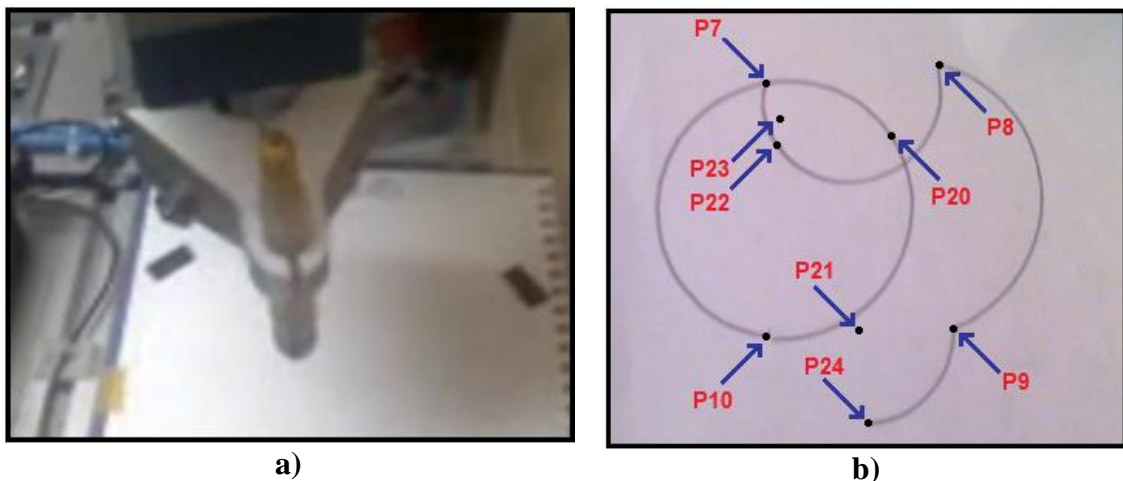


Figura 4.3: Práctica 3, a) Sujeción del lapicero por pinza del robot RV-2AJ, b) Dibujo de trayectoria con interpolación circular.

Práctica 4

Selección de piezas por color

OBJETIVO: Realizar un programa para que el brazo industrial RV-2AJ tome una pieza de trabajo y la seleccione respecto a su color (negro o no negro), consta de una interrupción⁶⁰ desde posición de espera hasta la posición de sensado de color⁶¹. La interrupción es generada, si en el transcurso antes mencionado, no hay pieza en el contenedor de rampa.

Las posiciones que se ocupan para la elaboración de esta, se muestran en la tabla 4.4:

Posición	X	Y	Z	A/P	B/R	Referencia
P1	-0.79	163.14	471.59	-88.60	180.37	Posición Inicial
P2	218.84	243.88	373.33	-27.74	180.18	Posición de espera
P3	218.83	243.88	157.31	-27.74	180.18	Se acerca a pieza para sensado
P4	227.57	243.88	157.31	-27.75	180.18	Sensa color
P5	255.06	192.58	399.67	-230.95	179.99	Levanta pieza
P6	-155.10	170.16	417.14	-157.92	179.22	Arriba depósito 1
P7	-155.10	170.16	335.35	-157.92	179.22	Listo para soltar en depósito 1
P8	20.89	194.30	420.86	-107.59	179.22	Arriba depósito 2
P9	20.89	194.30	329.45	-107.59	179.22	Listo para soltar en depósito 2
P10	218.83	243.88	250.31	-27.74	180.18	Se acerca a la pieza
P11	254.21	192.21	164.42	128.07	179.99	Tomar pieza
P12	254.21	192.22	219.53	128.07	179.99	Preparando para tomar pieza

Tabla 4.4: Posiciones del brazo industrial RV-2AJ para la práctica 4.

El código del programa es:

```

10 OVRD 60           ' Velocidad de movimiento del robot
20 MOV P1           ' Posición inicial
30 DEF ACT 1,M_IN(8)=0 GOSUB *INT1,S ' Define interrupción cuando no hay pieza
40 *MAIN           ' Función principal
50 HCLOSE 1        ' Cierra pinza
60 MOV P2           ' Posición de espera
70 ACT 1=1         ' Habilita interrupción
80 GOSUB *PIEZA    ' Si hay pieza en contenedor va a *PIEZA
90 GOTO *MAIN      ' Regresa a *MAIN para realizar un ciclo
                   ' infinito
100 END            ' Fin de función principal

```

⁶⁰ Ver en este trabajo el tema “Interrupción”, p. 62.

⁶¹ Ver en este trabajo el tema “Entrada y salidas de señales externas”, p. 64.

110	*PIEZA	' Subrutina *PIEZA
120	MVS P3	' Se posiciona el sensor de la pinza cerca
		' de la pieza del contenedor
130	MVS P4	' Se aproxima el sensor a la pieza
140	DLY 0.5	' Espera 0.5 seg
150	M1=M_IN(9)	' Sensa color de pieza, el dato es guardado
		' en M1
160	MVS P3	' Se aleja el sensor de la pieza
170	MVS P10	' Sube el brazo
180	MOV P12	' Preparando para tomar pieza
190	HOPEN 1	' Abre pinza
200	MOV P11	' Pinza posicionada en pieza
210	DLY 0.5	' Espera 0.5 seg
220	ACT 1=0	' Deshabilita interrupción
230	HCLOSE 1	' Cierra pinza
240	MVS P5	' Levanta la pieza
250	MOV P1	' Se lleva a posición inicial
260	IF M1=0 THEN	' Si M1=0, es negra; si M1=1, es no negra
270	MOV P8	' -----
280	MVS P9	' Si la pieza es negra
290	HOPEN 1	' La pieza es colocada en contenedor 2
300	MVS P8	' -----
310	ELSE	
320	MOV P6	' -----
330	MVS P7	' Si la pieza es no negra
340	HOPEN 1	' La pieza es colocada en contenedor 1
350	MVS P6	' -----
360	ENDIF	
370	MOV P1	' Posición inicial
380	RETURN	' Finaliza subrutina y regrese a la
		' instrucción 70
390	*INT1	' Subrutina de interrupción
400	WHILE M_IN(8)=0	' El robot no hace nada hasta que haya
		' pieza
410	WEND	
420	RETURN 0	' Retorna a la línea donde se generó la
		' interrupción

En la figura 4.4 se muestran algunas partes de la práctica 4, estas son: a) Espera de pieza y al mismo tiempo habilitación de la interrupción, b) Sensado de color de la pieza y a la vez,

deshabilitación de interrupción y c) Colocación de la pieza en el contenedor, dependiendo el color previamente sentido.



a) *Espera de pieza y habilitación de la interrupción,*
b) *Sensado de color de la pieza y deshabilitación de interrupción y*
c) *Colocación de pieza en contenedor, dependiendo el color.*

Práctica 5

Paletizado tipo 1 (PLT 1: zig-zag)

OBJETIVO: Realizar un programa para que el brazo industrial RV-2AJ tome un sello y marque en papel las posiciones que genera el paletizado tipo 1⁶² de 3x3.

Las posiciones que se ocupan para la elaboración de esta, se muestran en la tabla 4.5:

Posición	X	Y	Z	A/P	B/R	Referencia
P1	-0.88	163.19	451.59	-88.49	179.03	Posición inicial
P2	269.03	171.88	224.84	-91.82	179.03	Arriba del sello
P3	269.03	171.87	185.50	-91.82	179.03	Toma el sello
P4	-161.50	297.62	382.09	-91.82	179.03	Arriba de libreta
P5	-161.50	297.62	171.60	-91.82	179.03	Se acerca a libreta
P6	-111.21	279.70	148.85	-91.82	179.03	P_1
P7	-111.21	339.15	148.85	-91.82	179.03	P_2
P8	-197.67	279.71	148.85	-91.82	179.03	P_3
P9	-197.67	339.15	148.85	-91.82	179.03	P_4

Tabla 4.5: Posiciones del brazo industrial RV-2AJ para la práctica 5.

El código del programa es:

```

10 OVRD 40           ' Designa velocidad
20 DEF PLT 1,P6,P7,P8,P9,3,3,1 ' Define paletización tipo 1 de 3x3
30 M1=1             ' Se inicia la variable de posiciones de paletización
                    ' en 1
40 MOV P1           ' Mueve el brazo a la posición inicial
50 HCLOSE 1         ' Cierra pinza
60 MOV P2           ' Posiciona brazo arriba del sello
70 HOPEN 1          ' Abre pinza
80 MVS P3           ' Se prepara para tomar sello
90 HCLOSE 1         ' Cierra pinza (toma sello)
100 MVS P2           ' Se levanta el sello
110 MOV P1           ' Se lleva a posición inicial
120 MOV P4           ' Posiciona arriba de libreta
130 MOV P5           ' Se acerca a la libreta
140 OVRD 10         ' Reduce velocidad63
150 WHILE (M1<=9)  ' Realiza tantas repeticiones como posiciones de la

```

⁶² Ver en este trabajo el tema “Instrucciones de paletizado”, p. 81.

⁶³ Se reduce velocidad para evitar forzar al robot por la posición actual.

160 P10=PLT 1,M1	' matriz de paletización ' La variable P10 toma el valor de las coordenadas del punto de paletización actual
170 M1=M1+1	' Incrementa M1
180 MVS P10,-30	' El brazo se mueve a 30mm retraídos de la posición P10, previamente definida
190 MVS P10	' Mueve al punto actual de paletización (P10)
200 MVS P10,-30	' Retrae el brazo 30mm de P10
210 WEND	' Evalúa si la condición del WHILE sigue cumpliéndose sino sale del ciclo
220 MOV P1	' Mueve a posición inicial
230 END	' Fin del programa

En la figura 4.5, se muestra la sujeción del sello y el dibujo de la posiciones del paletizado tipo 1 de magnitud 3x3 además se visualiza el trayectoria de este tipo de paletizado.

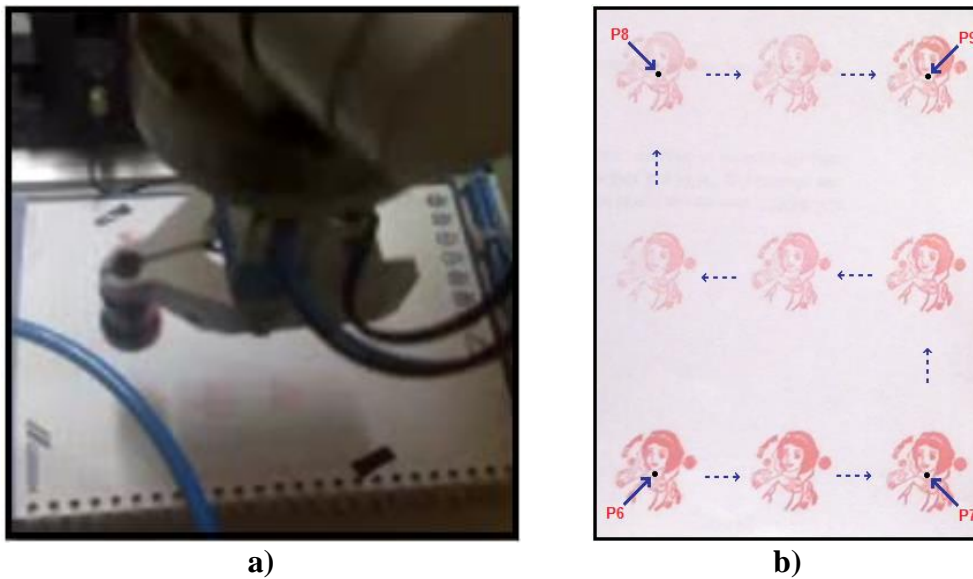


Figura 4.5: Práctica 5, a) Sujeción del sello por pinza del robot RV-2AJ, b) Dibujo de la posiciones del paletizado tipo 1 de magnitud 3x3 y trayectoria generada.

Práctica 6

Paletizado tipo 2 (PLT 2: mismo sentido)

OBJETIVO: Realizar un programa para que el brazo industrial RV-2AJ tome un sello y marque en papel las posiciones que genera el paletizado tipo 2⁶⁴ de 3x3.

Las posiciones que se ocupan para la elaboración de esta, se muestran en la tabla 4.6:

Posición	X	Y	Z	A/P	B/R	Referencia
P1	-0.88	163.19	451.59	-88.49	179.03	Posición inicial
P2	269.03	171.88	224.84	-91.82	179.03	Arriba del sello
P3	269.03	171.87	185.50	-91.82	179.03	Toma el sello
P4	-161.50	297.62	382.09	-91.82	179.03	Arriba de libreta
P5	-161.50	297.62	171.60	-91.82	179.03	Se acerca a libreta
P6	-111.21	279.70	148.85	-91.82	179.03	P_1
P7	-111.21	339.15	148.85	-91.82	179.03	P_2
P8	-197.67	279.71	148.85	-91.82	179.03	P_3

Tabla 4.6: Posiciones del brazo industrial RV-2AJ para la práctica 6.

El código del programa es:

```

10 OVRD 40           ' Designa la velocidad del robot
20 DEF PLT 1,P6,P7,P8, ,3,3,2 ' Define paletización tipo 2 de 3x3
30 M1=1             ' Se inicia la variable de posiciones de paletización
                    ' en 1
40 MOV P1           ' Mueve el brazo a la posición inicial
50 HCLOSE 1        ' Cierra pinza

60 MOV P2           ' Posiciona el brazo arriba del sello
70 HOPEN 1         ' Abre pinza
80 MVS P3           ' Se prepara para tomar sello
90 HCLOSE 1        ' Cierra pinza (toma sello)
100 MVS P2          ' Se levanta el sello
110 MOV P1          ' Se lleva a posición inicial
120 MOV P4          ' Posiciona arriba de libreta
130 MOV P5          ' Se acerca a la libreta
140 OVRD 5         ' Reducción de velocidad65
150 WHILE (M1<=9) ' Realiza tantas repeticiones como posiciones de la

```

⁶⁴ Ver en este trabajo el tema “Instrucciones de paletizado”, p. 83.

⁶⁵ Se reduce velocidad para evitar forzar al robot por la posición actual.

160 P10=PLT 1,M1	' matriz de paletización ' La variable P10 toma el valor de las coordenadas del punto de punto de paletización actual
170 M1=M1+1	' Incrementa M1
180 MVS P10,-30	' El brazo se mueve a 30mm retraídos de la posición P10, previamente definida
190 MVS P10	' Mueve al punto actual de paletización (P10)
200 MVS P10,-30	' Retrae el brazo 30mm de P10
210 WEND	' Evalúa si la condición del WHILE sigue cumpliéndose sino sale del ciclo
220 OVRD 40	' Incrementa velocidad
230 MOV P1	' Mueve a posición inicial
240 END	' Fin del programa

En la figura 4.6, se muestra la sujeción del sello y el dibujo de la posiciones del paletizado tipo 2 de magnitud 3x3, además se visualiza el trayectoria de este tipo de paletizado.

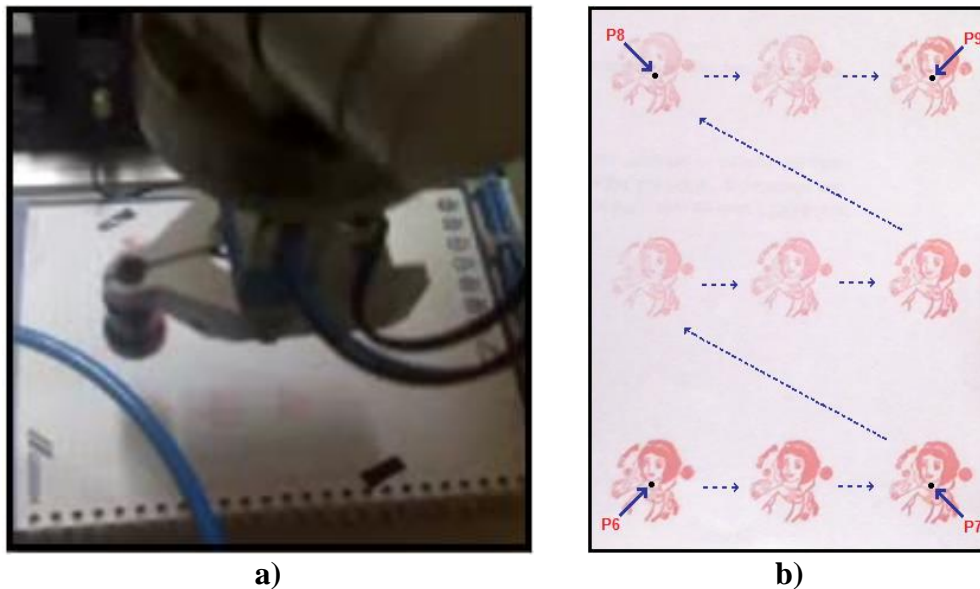


Figura 4.6: Práctica 6, a) Sujeción del sello por pinza del robot RV-2AJ, b) Dibujo de la posiciones del paletizado tipo 1 de magnitud 3x3 y trayectoria generada.

Práctica 7

Paletizado tipo 3 (PLT 3: arco)

OBJETIVO: Realizar un programa para que el brazo industrial RV-2AJ tome un sello y marque en papel las posiciones que genera el paletizado tipo 3⁶⁶ de 4x1.

Las posiciones que se ocupan para la elaboración de esta, se muestran en la tabla 4.7:

Posición	X	Y	Z	A/P	B/R	Referencia
P1	-0.88	163.19	451.59	-88.49	179.03	Posición inicial
P2	269.03	171.88	224.84	-91.82	179.03	Arriba del sello
P3	269.03	171.87	185.50	-91.82	179.03	Toma el sello
P4	-161.50	297.62	382.09	-91.82	179.03	Arriba de libreta
P5	-161.50	297.62	171.60	-91.82	179.03	Se acerca a libreta
P6	-111.21	279.70	148.85	-91.82	179.03	P_1
P7	-111.21	339.15	148.85	-91.82	179.03	P_2
P8	-197.67	279.71	148.85	-91.82	179.03	P_3
P9	-197.67	339.15	148.85	-91.82	179.03	P_4

Tabla 4.7: Posiciones del brazo industrial RV-2AJ para la práctica 7.

El código del programa es:

```

10 OVRD 40           ' Designa la velocidad del robot
20 DEF PLT 1,P6,P7,P8,P9,3,4,1 ' Define paletización tipo 1 de 4x3
30 P11=PLT 1,2      ' Definen los puntos P11, P12 y P13,
40 P12=PLT 1,5      ' que son utilizados para la paletización tipo 3
50 P13=PLT 1,7      ' paletización tipo 3

60 DEF PLT 2,P11,P12,P13,,4,1,3 ' Define paletización tipo 3 de 4 divisiones
70 M1=1             ' Se inicia la variable de posiciones de paletización
                   ' en 1
80 MOV P1           ' Mueve el brazo a la posición inicial
90 HCLOSE 1         ' Cierra pinza

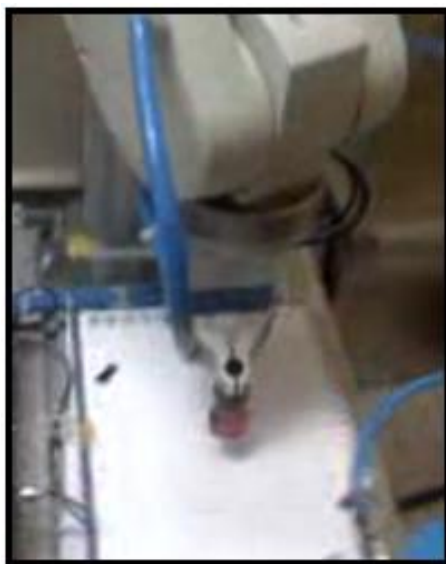
100 MOV P2          ' Posiciona brazo arriba del sello
110 HOPEN 1         ' Abre pinza
120 MVS P3          ' Se prepara para tomar sello
130 HCLOSE 1       ' Cierra pinza (toma sello)
140 MVS P2          ' Se levanta el sello
150 MOV P1          ' Se lleva a posición inicial

```

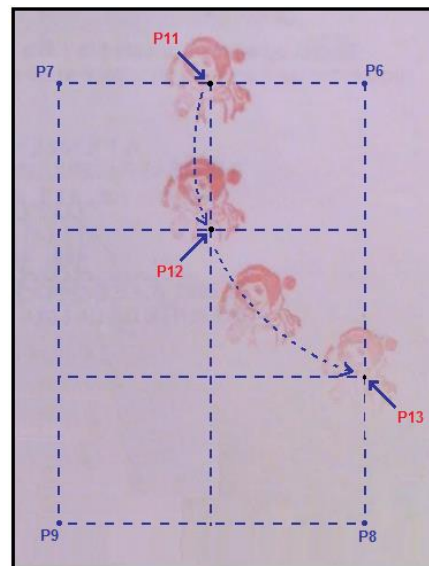
⁶⁶ Ver en este trabajo el tema “Instrucciones de paletizado”, p. 85.

160 MOV P4	' Posiciona arriba de libreta
170 MOV P5	' Se acerca a la libreta
180 OVRD 10	' Reduce velocidad ⁶⁷
190 WHILE (M1<=4)	' Realiza tantas repeticiones como posiciones de la
	' matriz de paletización tipo 3
200 P10=PLT 2,M1	' La variable P10 toma el valor de las coordenadas del
	' punto de punto de paletización actual
210 M1=M1+1	' Incrementa M1
220 MVS P10,-30	' El brazo se mueve a 30mm retraídos de la posición
	' P10, previamente definida
230 MVS P10	' Mueve al punto actual de paletización (P10)
240 MVS P10,-30	' Retrae el brazo 30mm de P10
250 WEND	' Evalúa si la condición del WHILE sigue
	' cumpliéndose sino sale del ciclo
260 OVRD 40	' Incrementa velocidad
270 MOV P1	' Mueve a posición inicial
280 END	' Fin del programa

En la figura 4.7, se muestra la sujeción del sello y el dibujo de la posiciones del paletizado tipo 3 de magnitud 4x1, además se visualiza los puntos (P11, P12 y P13), tomados de una previa paletización tipo 1, así como la trayectoria generada por el paletizado tipo 3.



a)



b)

Figura 4.5: Práctica 6, a) Sujeción del sello por pinza del robot RV-2AJ, b) Dibujo de la posiciones del paletizado tipo 3 de magnitud 4x1 y trayectoria generada.

⁶⁷ Se reduce velocidad para evitar forzar al robot por la posición actual.

CONCLUSIONES

En base a la experiencia que hemos tenido utilizando el robot industrial RV-2AJ Estación 6 del Laboratorio Móvil de Automatización y Robótica de la Universidad Veracruzana, podemos realizar las siguientes observaciones:

- Es de vital importancia que el operador del robot sea paciente y que haya leído con especial atención las medidas de seguridad.
- Así también estar atentos al cambio de baterías, preguntando previamente cuando fue la última vez que se sustituyeron.
- Es muy importante que cuando se lleve a cabo un cambio de baterías se vuelvan a configurar los ejes.
- Una ventaja de los robots industriales manipuladores de una configuración similar al robot RV-2AJ, es la capacidad del modo de producción, pues esta puede ser de manera distribuida, debido a que el robot puede realizar diferentes tareas y no simplemente una como la producción en línea, teniendo como desventaja una producción lenta.
- Las aplicaciones con el robot industrial RV-2AJ no se limita a este trabajo, de lo contrario, este trabajo tiene el fin de ayudar a facilitar en manejo básico del robot.

APÉNDICE A

CARACTERÍSTICAS GENERALES DEL ROBOT INDUSTRIAL RV-2AJ

CARACTERÍSTICAS/FUNCIÓN		ESPECIFICACIONES RV-2AJ	
Grado de libertad (número de ejes)		5	
Montaje		Posible montaje en suelo o en techo	
Construcción		Brazo articulado vertical	
Sistema de accionamiento		Servo AC (eje J1, J2, J3 y J5: con freno; J4 y J6: sin freno)	
Detección de posición		Decodificador en valor absoluto	
Rango de movimientos	Cuerpo (J1)	grados	300 (-150 hasta +150)
	Hombro (J2)		180 (-60 hasta +120)
	Codo (J3)		230 (-110 hasta +120)
	Inclinación muñeca (J5)		180 (-90 hasta +90)
	Giro muñeca (J6)		400 (-200 hasta +200)
Velocidad de movimiento	Cuerpo (J1)	grados/s	180
	Hombro (J2)		90
	Codo (J3)		135
	Inclinación muñeca (J5)		180
	Giro muñeca (J6)		210
Velocidad máxima resultante		mm/s	2200
Fuerza de elevación	Valor nominal	kg	1.5
	Valor máximo		2
Repetibilidad posicionamiento		mm	±0.02
Temperatura ambiente		°C	0 hasta 40
Peso		kg	17
Momento nominal	Inclinación muñeca (J5)	Nm	2.16
	Giro muñeca (J6)		1.10
Momento de inercia nominal	Inclinación muñeca (J5)	kgm ²	3.24x10 ⁻²
	Giro muñeca (J6)		8.43x10 ⁻³
Radio de alcance (hasta el punto de giro del eje J5)		mm	410
Cableado de herramienta		4 líneas de señales de entrada (conexión en la zona de la pinza), 4 líneas de señales de salida (conexión en la zona del zócalo), línea de tensión para pinza motorizada (conexión en la zona de la pinza)	
Tubo neumático para herramienta		04x4 (desde el nivel de la base hasta el área de la pinza)	
Presión de alimentación neumática		MPa (bar)	0.5±10% (5±10%)
Brida de la pinza		ISO 9409-1-31.5	
Clase de protección		IP 30	
Controlador del robot		CR1	

APÉNDICE B

CARACTERÍSTICAS GENERALES DEL CONTROLADOR CR1-571

CARACTERÍSTICAS		UNIDAD	ESPECIFICACIÓN	OBSERVACIONES
Modelo		-	CR1-571	
Número de ejes a controlar		-	6 a la vez (máx.)	
CPU			61 bit RISC, DSP	
Capacidad de memoria	Posiciones de enseñanza y números de pasos	punto de paso	2500 5000	
	Número de programas	-	88	
Lenguaje de programación		-	MELFA- BASIC IV Movemaster Command	
Método de enseñanza de posición			Método MDI	
Entradas y salidas externas	Entradas y salidas	punto	16/16	Max. 240/240
	E/S dedicadas	punto	Asignado con E/S de uso general	“STOP” 1 punto fijado
	E/S de abrir cerrar/mano	punto	Entradas 4 puntos/ Salidas 0 puntos	Se pueden añadir hasta 4 puntos de salida
	Entrada de paro de emergencia	punto	1	Solo una línea de emergencia
	Ent. Interruptor de puerta	punto	1	Solo una línea de interruptor puerta
Interfaz	RS-232C	puerto	1	
	RS-122	puerto	1	Dedicada para el T/B
	Ranura para la mano	slot	1	Dedicada para interfaz de mano neumática
	Ranura de expansión	slot	3	Caja opción extensión usada
	Conexión de E/S del robot	canal	1	240 usados con E/S de uso general
Fuente de alimentación	Rango de voltaje	V	1 fase, AC, 90 a 132V 1 fase, AC, 170 a 253V	
	Potencia máxima	kVA	0.7	No incluye la sobrecorriente de arranque
Dimensión de contorno		mm	212X290X151	Excepto salientes
Peso		kg	Aprox.8	
Construcción		-	Aautomático y abierto	IP20
Rango de temperaturas		°C	De 0 a 40	
Humedad ambiente		%RH	De 45 a 85	Sin condensación
Toma de tierra		Ω	100 o menos	Clase D
Color		-	Gris claro	Munsell 7.65 y 7.64

APÉNDICE C

INSTALACIÓN DEL PROGRAMA Y CONFIGURACION DE LA COMUNICACIÓN

Para la instalación del programa COSIMIR ® Industrial, la PC debe contar con los siguientes requerimientos mínimos:

- ❑ **Procesador:** Pentium III a 800 MHz.
- ❑ **Memoria:** 256 MB RAM.
- ❑ **Disco duro:** 700 MB de espacio libre.
- ❑ **Sistema Operativo:** Windows 98/Windows NT/2000/XP.
- ❑ **Adaptador de gráficos:** Adaptador con aceleración 3D y soporte OpenGL.
- ❑ **Unidad de CD ROM:** Velocidad 16x.
- ❑ **Interfaz:** Un puerto serial libre para la conexión al controlador del robot o una interfaz de red para la conexión TCP/IP. Un puerto paralelo o un Puerto USB para la llave de licencia USB.

> Instrucciones de instalación del software COSIMIR ® Industrial 4.1:

1. Ejecutar el archivo "SETUP. EXE " del disco de instalación.
2. Seleccionar el idioma como se muestra en la figura C.1.

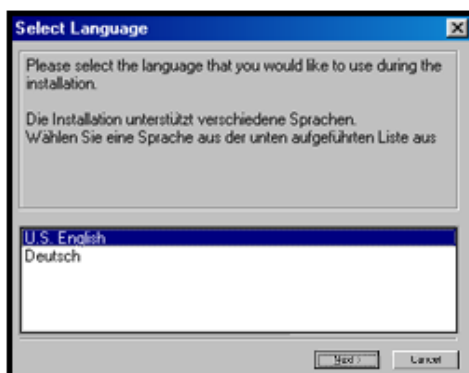


Figura C.1: Selección de idioma.

3. Lo siguiente, son instrucciones y acuerdos de licencia (figura C.2).

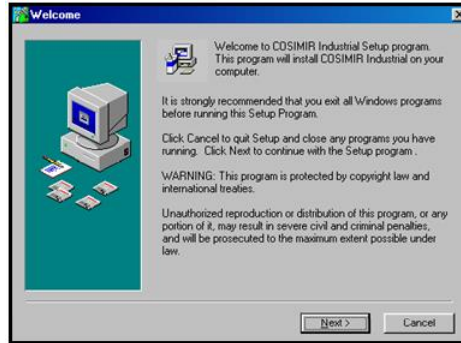


Figura C.2: Instrucciones y acuerdos de licencia.

4. Seleccionar la unidad y la ruta en la cual se instalará COSIMIR ® industrial 4.1 (figura C.3).

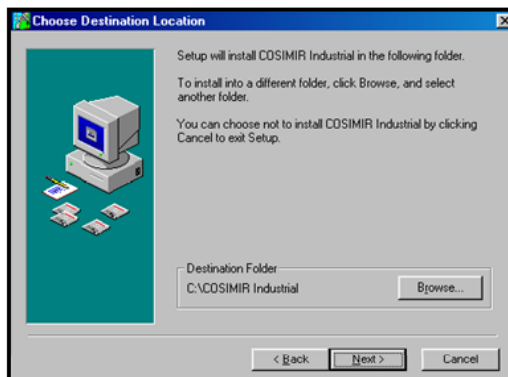


Figura C.3: Selección de ruta.

5. Seleccionar opciones de COSIMIR ® Industrial 4.1 (figura C.4).

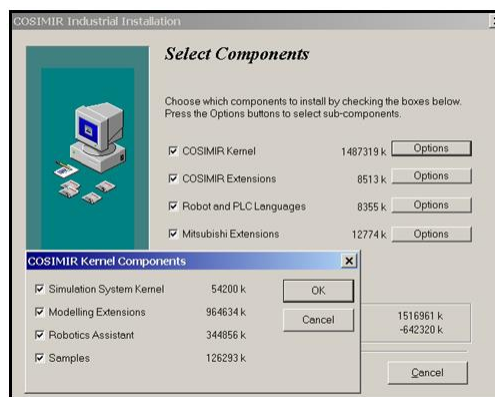


Figura C.4: Selección de componentes.

6. Seleccionar el puerto serial de interfaz y el tipo de robot (figura C.5).

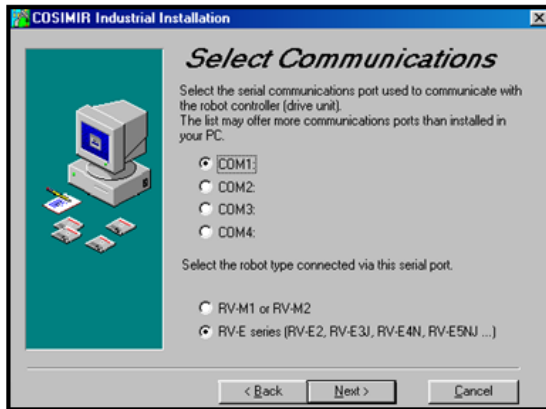


Figura C.5: Selección de puerto para la comunicación serial.

7. Seleccionar el grupo de programas dentro de Windows, en el que se encontrará COSIMIR ® Industrial 4.1 (figura C.6).

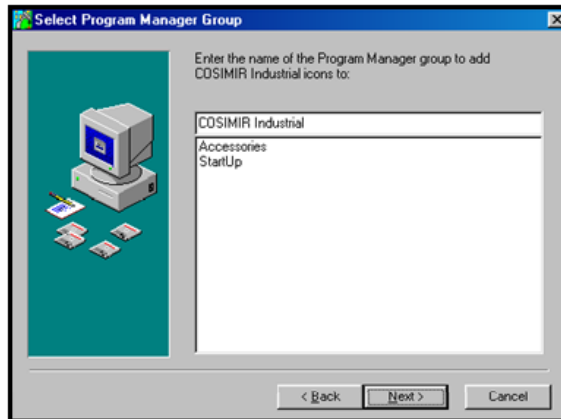


Figura C.6: Selección de directorio del programa.

8. Presionar Next, para iniciar la instalación (figura C.7).

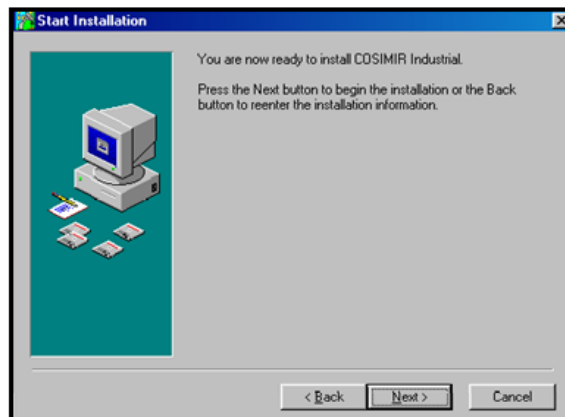


Figura C.7: Inicia instalación.

9. La ayuda en línea de COSIMIR ® Industrial 4.1 se basa en el estándar de formato de ayuda HTML para Windows. Para usarla es necesario utilizar Microsoft Internet Explorer 3.0 o superior.
10. COSIMIR ® Industrial 4.1, utiliza una clave de licencia y los controladores para esta clave de licencia se instalan automáticamente (figura C.8).



Figura C.8: *Instalación de controladores para es uso de la ayuda del software.*

¡La instalación se ha completado!

11. Una vez instalada la llave física, debe reiniciar el sistema (figura C.9).

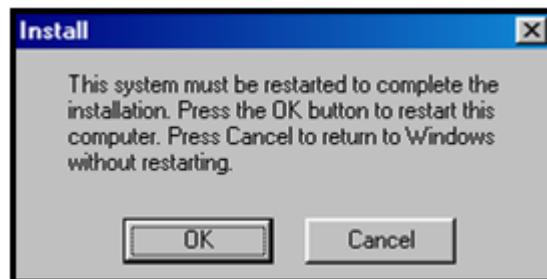


Figura C.9: *Reinicio del sistema.*

› Configuración de la comunicación

Una vez instalado el COSIMIR ® Industrial 4.1, para poder tener una correcta comunicación entre el robot industrial RV-2AJ y este software, se debe configurar el puerto de comunicación serial. Para esto, se debe ingresar a:

Menú Extras → Settings → Communications Port

Aparece una ventana similar a la figura C.10. Seleccionar la pestaña Common, en

Communication Interface, elegir Serial Interface.

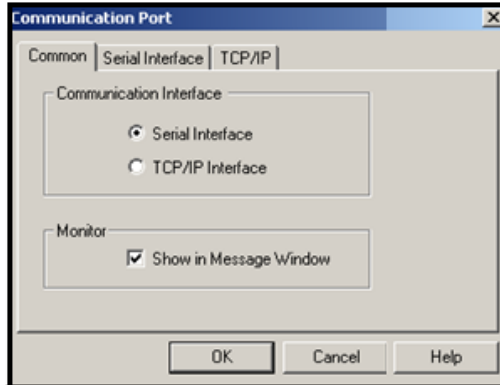


Figura C.10: Elección de interfaz de comunicación.

Después, seleccionar la pestaña Serial Interface (figura C.11); y en Port, escoger el número de puerto que se ha designado durante la instalación del programa para conectar el cable de comunicación (en este caso, es el COM1). Los otros campos permanecen sin modificaciones.

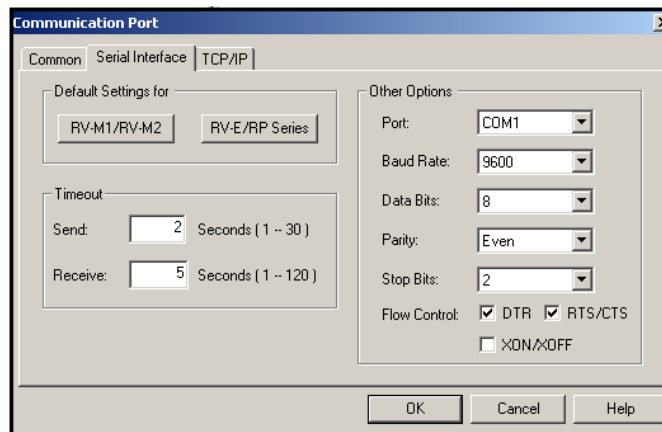


Figura C.11: Elección del puerto de comunicación.

APÉNDICE D

LISTA DE ERRORES

Los posibles errores⁶⁸ que podrían aparecer en un determinado momento al trabajar con el robot industrial RV-2AJ. El formato de error se muestra en la figura D.1

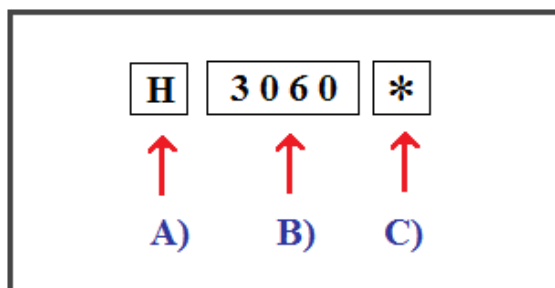


Figura D.1: Formato de error.

Donde:

A) Clase de error:

H: Error de Alto nivel..... El servomotor se apaga.

L: Error de Bajo nivel..... La operación se detiene.

C: Advertencia.....La operación continua.

B) El tipo de error es indicado con un número de 4 dígitos.

C) Un error marcado con un “*” se restablece colocando la alimentación de energía en ON u OFF.

En las tablas D.1 y D.2 se presentan los errores más comunes, así también, se muestra el mensaje de error, la causa que ocasiona y las medidas que se deben llevar a cabo para corregirlo.

ERROR	MENSAJE, CAUSAS Y MEDIDAS	
H0070	MENSAJE DE ERROR	La señal EMG está activada.
	CAUSA	Paro de emergencia del T/B está activado.
	MEDIDA	Cancele el paro de emergencia.
H0072	MENSAJE DE ERROR	Error al remover el switch de operación del T/B.
	CAUSA	La operación de conectar/desconectar el T7B no fue realizada correctamente.
	MEDIDA	Pruebe con el botón “REMOVE T/B”, con el modo de operación “MODE” del controlador o con el switch “ENABLE/DISABLE” en el T/B.

Tabla D.1: Lista de errores más comunes.

⁶⁸ Ver manual “Troubleshooting”, par aver lista completa.

ERROR	MENSAJE, CAUSAS Y MEDIDAS	
C1760	MENSAJE DE ERROR	Origen de dato prohibido.
	CAUSA	El origen del dato no es permitido.
	MEDIDA	Corregir el origen correcto del dato.
L3285	MENSAJE DE ERROR	No se puede ejecutar el programa.
	CAUSA	No se puede efectuar el programa en el estado de paro o ejecución.
	MEDIDA	Resetear el programa.
L4130	MENSAJE DE ERROR	Nombre restringido para un programa.
	CAUSA	Un carácter restringido se uso en el nombre del programa.
	MEDIDA	Solo pueden ser usados números o caracteres del alfabeto.
L4220	MENSAJE DE ERROR	Error de sintaxis.
	CAUSA	Hay un error en la sintaxis de declaración del comando de entrada.
	MEDIDA	Utilizar la sintaxis correcta.
L4320	MENSAJE DE ERROR	La variable escrita está protegida.
	CAUSA	La variable escrita está protegida.
	MEDIDA	1.- Use una variable que se pueda escribir. 2.-Cancela la protección de escritura.
H5000	MENSAJE DE ERROR	La llave del T/B se encuentra habilitada.
	CAUSA	El switch de habilitar/deshabilitar del T/B se encuentra “habilitado”, o cuando el switch “ENABLE/DISABLE” del panel de operación se encuentra en “auto”.
	MEDIDA	Deshabilitar la llave del T/B, o cambiar el modo a TEACH en el panel de operación.
C7520	MENSAJE DE ERROR	Batería agotada.
	CAUSA	Consumo de batería.
	MEDIDA	Reemplazar las baterías del brazo robótico y del controlador ⁶⁹ .

Tabla D.2: *Continuación de lista de errores más comunes.*

⁶⁹ Ver apéndice E: “Cambio de baterías”, p. 96.

APÉNDICE E

CAMBIO DE BATERÍAS

Este juego de baterías de mantenimiento (TN 537000) debe instalarse por primera vez cuando haya caducado el temporizador de la batería (aproximadamente un año tras la compra del robot). Incluye:

- › 5 pilas A6BAT para el brazo robot RV-2AJ
- › 1 pila ER6 para el controlador CR1-71

El temporizador de la batería se puede consultar en el T/B, realizando lo siguiente: MENU / 5 MAINT / 5 POWER / "Battery ____"

Cuando desciende la capacidad de las baterías, el robot envía el código de error 7520. Entonces es preciso cambiarlas inmediatamente para asegurar que no se borren los datos de posición, en este caso se sugiere que todos los programas y datos importantes del robot sean respaldados mediante una PC.

Las baterías tienen una vida aproximada de 14.000 horas, o año y medio. Se recomienda el cambio anual de las baterías.

Si no se cumple el intervalo de mantenimiento, es posible que sea necesario enviar el robot a reparar con cargo al cliente para volver a configurar sus ejes⁷⁰.

Pasos para la sustitución de baterías

(El margen de tiempo por seguridad para realizar esta sustitución es de 15 minutos)

› **Baterías del brazo robótico RV-2AJ**

1. Encender el controlador CR1-57, previamente conectado al brazo robótico RV-2AJ, para evitar la pérdida de los datos de posición, cabe señalar que, si el cable no está conectado correctamente o si el controlador está apagado, los datos de posición se perderán.
2. Con ayuda del T/B, mover el brazo robótico a una posición tal, que permita facilitar el acceso y extracción de la cubierta F que se muestra en la figura E.1.

⁷⁰ Ver apéndice F: "Configuración de origen de los ejes", p. 99.

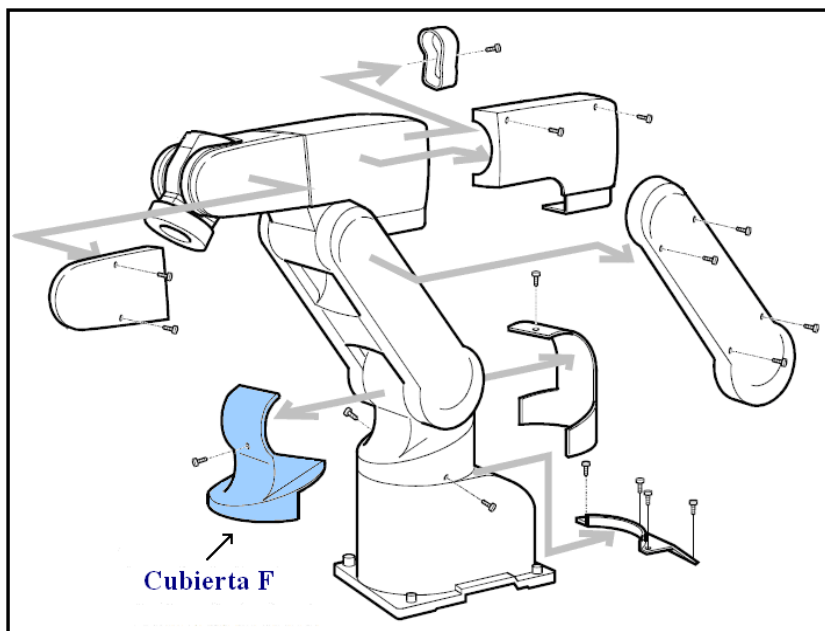


Figura E.1: Ubicación de la Cubierta F del brazo robótico RV-2AJ.

3. Una vez realizado el paso anterior, presionar el botón de paro de emergencia.
4. Desatornillar y retirar la cubierta F.
5. Una vez retirada la cubierta, retirar los tornillos de la caja metálica, extraer la base donde se encuentran las baterías y removerlas, sustituyéndolas por las nuevas y reconectar los cables. Es importante mencionar que se tienen que este reemplazar todas las baterías por otras nuevas al mismo tiempo, ver figura E.2.

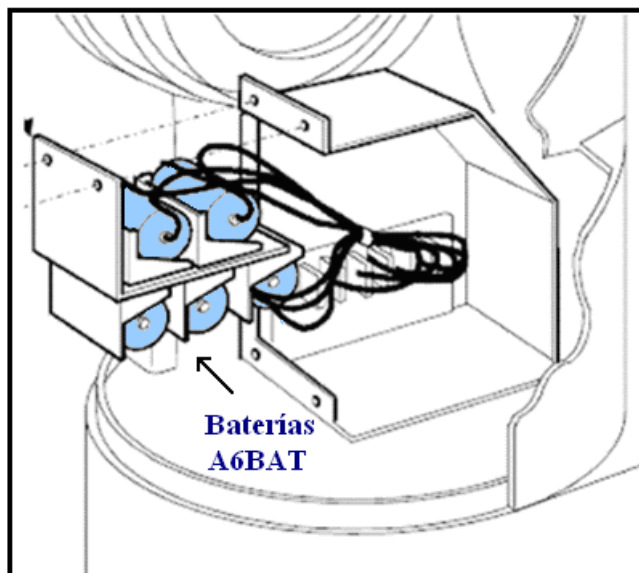


Figura E.2: Ubicación de las baterías del brazo robótico RV-2AJ.

6. Colocar la base metálica en su lugar y atornillar, luego coloque la tapa F y atornille.

7. Inicializar el tiempo de consumo de la batería con ayuda del T/B siguiendo los siguientes pasos:
 - › Presionar la tecla [5] y seleccionar MAINT.
 - › Presionar la tecla [1] y seleccionar la opción INIT.
 - › Presionar la tecla [2] y después la tecla [INP] para seleccionar en la pantalla la inicialización del consumo de la batería.
 - › Cuando la tecla [INP] es presionada y después se presiona [1], el tiempo de consumo de batería se ha inicializado. La pantalla de inicialización aparecerá después de la ejecución.

› **Baterías del controlador CR1-571**

1. Conectar el controlador aproximadamente un minuto.
2. Desconectar el controlador y espere, por lo menos, tres minutos.
3. Desatornillar y retirar la cubierta superior como se muestra en la figura E.3.
4. Desconectar y retirar la batería vieja de su ubicación.
5. Colocar la batería nueva
6. Conectar los cables de la batería nueva en los conectores de la tarjeta de relés.

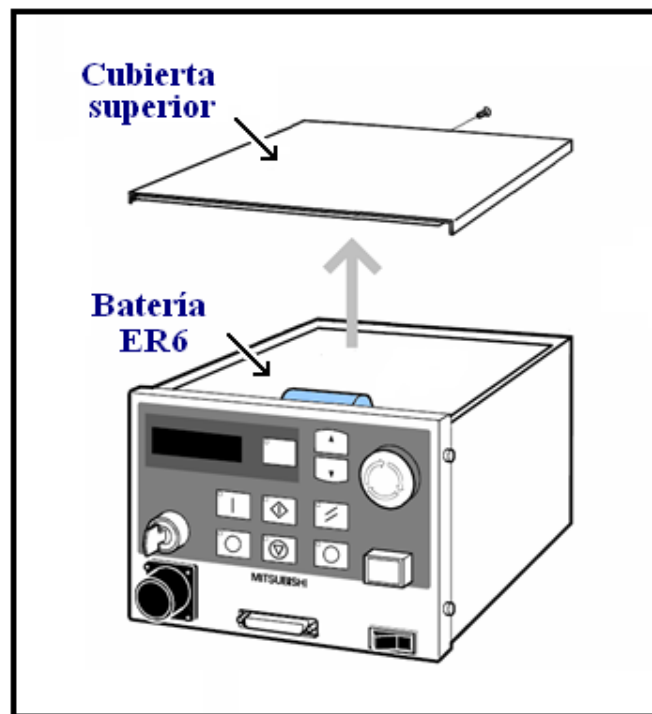


Figura E.3: Ubicación de la baterías del controlador CR1-571.

APÉNDICE F

CONFIGURACIÓN DE ORIGEN DE LOS EJES

Cuando la batería es sustituida⁷¹ porque se ha agotado, es necesario establecer el origen utilizando el método de límite mecánico, esto se lleva a cabo con ayuda del T/B.

PASOS GENERALES PARA LA CONFIGURACIÓN DEL ORIGEN DE TODOS LOS EJES DEL BRAZO ROBÓTICO RV-2AJ

1. Seleccione el modo TEACH en el controlador.
2. Primeramente, se debe fijar el origen del eje J5 (se menciona más adelante).
3. Para prevenir impactos del eje J5 en el brazo robótico, mover J5 lo más cercano a 0°, como se muestra en la figura F.1.

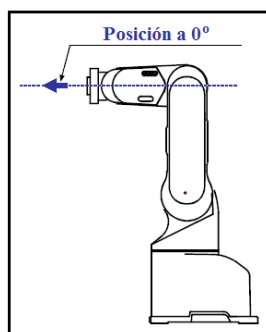


Figura F.1: Posición del brazo robótico para prevenir impactos.

4. Establecer las posiciones de origen de J1, J2, J3 y J6 (se mencionan más adelante).
5. Al terminar con el establecimiento de posiciones de origen, la postura del brazo robótico debe ser igual a la que se muestra en la figura F.2.

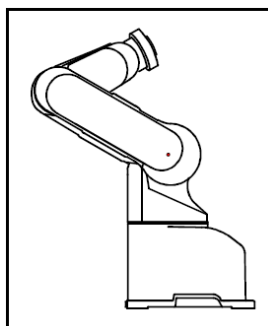


Figura F.2: Postura final del brazo robótico.

⁷¹ Ver apéndice E: "Cambio de baterías", p. 96.

6. Pulsar la tecla [↓] para situarse en SET AXIS.
7. Designar el eje para el origen, escribiendo “11101100” en SET AXIS y luego presionando la tecla [INP].
8. Presionar la tecla [1] y luego la tecla [INP], así, la posición de origen será fijada.
9. El ajuste de origen se ha completado.
10. Actualizar los datos.

En la figura F.1 se muestran de manera gráfica los pasos para la selección del método de límite mecánico que a continuación se mencionan:

- a) Presionar la tecla [MENU] del T/B.
- b) Dentro de MENU presionar la tecla [5], aparecerá el menú MAINT.
- c) Dentro de MAINT presionar la tecla [4], aparecerán los métodos para establecer el origen del robot, llamado menú ORIGIN.
- d) Dentro de ORIGIN presionar la tecla [2] para seleccionar el método MEACH, luego teclear [1] seguido de la tecla [INP].
- e) MECHA: Menú de método mecánico para establecimiento del origen de los ejes.



Figura F.1: Pasos para seleccionar el método de límite mecánico.

❑ Configuración del origen del eje J1:

- › En BRAKE escribir “10000000” para establecer la liberación del freno del eje J1.
- › Confirmar el eje para que los frenos se han puestos en libertad.
- › Pulsar las teclas [DEADMAN] + [MOVE] + [+X] a la vez para liberar los frenos.
- › Con las manos, mover lentamente el eje J1 en dirección negativa, hasta llegar al límite, como se muestra en la figura F.3.

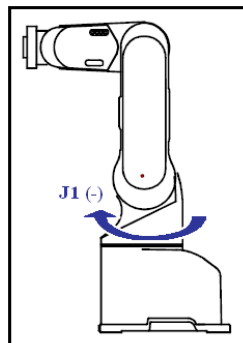


Figura F.3: Dirección negativa del eje J1.

- › Presionar la tecla [↓] para situarse en SET AXIS.

- › Designar el origen del eje. Colocando “1000000” en SET AXIS, luego presione la tecla [INP].
- › Presionar la tecla [1] y luego la tecla [INP].
- › La postura de origen se ha completado.
- › Actualizar los datos.

❑ Configuración del origen del eje J2:

- › En BRAKE escribir “01000000” para establecer la liberación del freno del eje J2.
- › Confirmar el eje para que los frenos se han puestos en libertad.
- › Sujetar con las manos al brazo robótico a medida que al liberar el freno del eje J2, este mantenga su postura.
- › Pulsar las teclas [DEADMAN] + [MOVE] + [+X] a la vez para liberar los frenos.
- › Con las manos, mover lentamente el eje J2 en dirección negativa, hasta llegar al límite, como se muestra en la figura F.4.

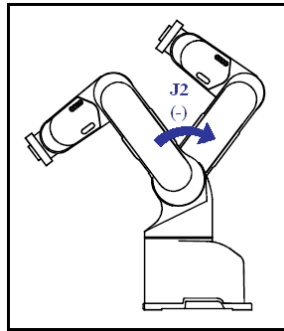


Figura F.4: Dirección negativa del eje J2.

- › Presionar la tecla [↓] para situarse en SET AXIS.
- › Designar el origen del eje. Colocando “01000000” en SET AXIS, luego presione la tecla [INP].
- › Presionar la tecla [1] y luego la tecla [INP].
- › La postura de origen se ha completado.
- › Actualizar los datos.

❑ Configuración del origen del eje J3:

- › En BRAKE escribir “00100000” para establecer la liberación del freno del eje J3.
- › Confirmar el eje para que los frenos se han puestos en libertad.
- › Sujetar con las manos al brazo robótico a medida que al liberar el freno del eje J3, este mantenga su postura.
- › Pulsar las teclas [DEADMAN] + [MOVE] + [+X] a la vez para liberar los frenos.
- › Con las manos, mover lentamente el eje J3 en dirección positiva, hasta llegar al límite, como se muestra en la figura F.5.

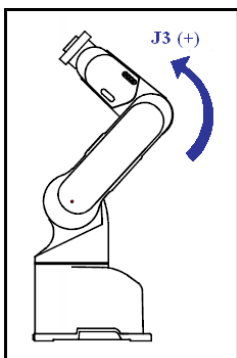


Figura F.5: Dirección positiva del eje J3.

- > Presionar la tecla [\downarrow] para situarse en SET AXIS.
- > Designar el origen del eje. Colocando “00100000” en SET AXIS, luego presione la tecla [INP].
- > Presionar la tecla [1] y luego la tecla [INP].
- > La postura de origen se ha completado.
- > Actualizar los datos.

❑ Configuración del origen del eje J5:

- > En BRAKE escribir “00001000” para establecer la liberación del freno del eje J5.
- > Confirmar el eje para que los frenos se han puestos en libertad.
- > Sujetar con las manos al brazo robótico a medida que al liberar el freno del eje J5, este mantenga su postura.
- > Pulsar las teclas [DEADMAN] + [MOVE] + [+X] a la vez para liberar los frenos.
- > Con las manos, mover lentamente el eje J5 en dirección negativa, hasta llegar al límite, como se muestra en la figura F.6.

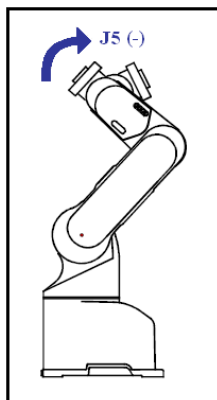


Figura F.6: Dirección negativa del eje J5.

- › Presionar la tecla [↓] para situarse en SET AXIS.
- › Designar el origen del eje. Colocando “0000100” en SET AXIS, luego presione la tecla [INP].
- › Presionar la tecla [1] y luego la tecla [INP].
- › La postura de origen se ha completado.
- › Actualizar los datos.

❑ Configuración del origen del eje J6:

- › En BRAKE escribir “00000100” para establecer la liberación del freno del eje J6.
- › Confirmar el eje para que los frenos se han puestos en libertad.
- › Sujetar con las manos al brazo robótico a medida que al liberar el freno del eje J6, este mantenga su postura.
- › Pulsar las teclas [DEADMAN] + [MOVE] + [+X] a la vez para liberar los frenos.
- › Con las manos, tome la pinza y rote lentamente el eje J6 hasta que coincidan las marcas, así como se muestra en la figura F.7.

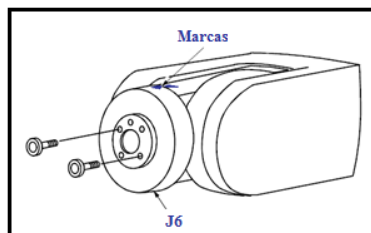


Figura F.7: Dirección negativa del eje J6.

- › Presionar la tecla [↓] para situarse en SET AXIS.
- › Designar el origen del eje. Colocando “00000100” en SET AXIS, luego presione la tecla [INP].
- › Presionar la tecla [1] y luego la tecla [INP].
- › La postura de origen se ha completado.
- › Actualizar los datos.

*** Actualización de datos ***

En la figura F.8 se muestran de manera gráfica los pasos para la selección del método DATA que a continuación se mencionan:

1. Presionar la tecla [MENU] del T/B.
2. Dentro de MENU presionar la tecla [5], aparecerá el menú MAINT.
3. Dentro de MAINT presionar la tecla [4], aparecerán los métodos para establecer el origen del robot, llamado menú ORIGIN.

4. Dentro de ORIGIN presionar la tecla [1] para seleccionar el método de entrada de datos DATA. Aparecerá en la pantalla “SERVO OFF”.
5. Presionar la tecla [1] y luego la tecla [INP] para apagar el servomotor.
6. Dentro del menú DATA (figura F.8a) escribir los códigos del dato “D” y del eje que se está configurando (figura F.8b). Utilizando las teclas [↓] + [INP] de manera conjunta para desplazarse a la siguiente línea de código.

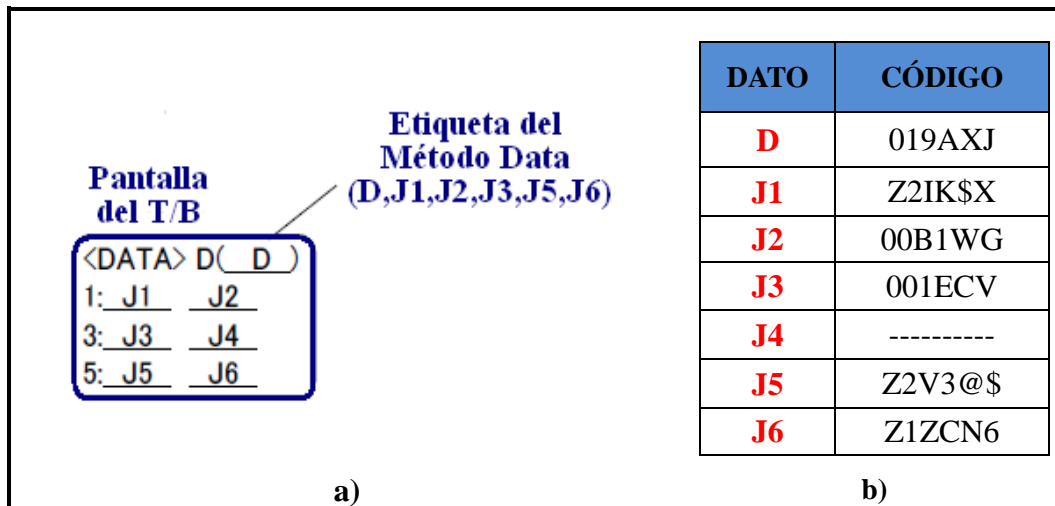


Figura F.8: a) Visualización del orden de entrada de datos, b) Códigos de datos para Configuración del origen de los ejes.

7. Después de introducir los códigos, pulsar la tecla [INP].
8. Confirmar pulsando la tecla [1] y luego la tecla [INP] para finalizar la configuración del origen.

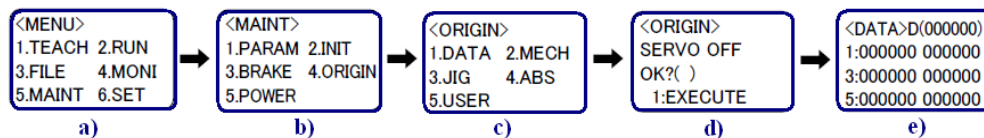


Figura F.9: Pasos para seleccionar el método DATA.

BIBLIOGRAFÍA

Libros:

- Aníbal Ollero, Baturone. “Robótica. Manipuladores y robots móviles”. México: Alfaomega. 2007.
- Jhon J., Craig. “ROBÓTICA”. 3^{er} Edición. México: Prentice Hall. 2006.
- Lñigo Madrigal, Rafael Y Vidal Idiarte, Enric. “Robots industriales manipuladores”. 1^a Edición. México: Alfaomega. 2004.
- Domínguez, Sergio et al. “Control en el Espacio de Estado”. 2^a Edición. Madrid: Prentice Hall, 2006.

Manuales:

- MITSUBISHI
Mitsubishi Industrial Robot
CR1/CR2/CR3/CR4/CR7/CR8/CR9 Controller INSTRUCTION MANUAL
Detailed explanation of functions and operations
http://www.ene.ttu.ee/elektrijamid/oppeinfo/materjal/AAR0080/Mitsubishi_Robot_Englisch.pdf
- MITSUBISHI
Mitsubishi Industrial Robot
CR1/CR2/CR3/CR4/CR7/CR8/CR9 Controller
INSTRUCTION MANUAL
Troubleshooting
<http://download.mitsubishi-automation.com/resources/manuals/132314.pdf>
- MITSUBISHI
Mitsubishi Industrial Robot
RV-1A/2AJ SERIES INSTRUCTION MANUAL
Robot arm setup & maintenance
<http://www.rixan.com/Portals/0/RV-1A-2AJ/1n2setup.pdf>
- MITSUBISHI
Mitsubishi Industrial Robot
CR1/CR2 Controller INSTRUCTION MANUAL
Explanations of MOVEMASTER COMMANDS
<http://download.mitsubishi-automation.com/resources/manuals/160277.pdf>

- MITSUBISHI
 Mitsubishi Industrial Robot
 CR1/CR1B Controller
 INSTRUCTION MANUAL
 Controller Setup, basic operation, and maintenance
[http://suport.siriustrading.ro/02.DocArh/07.RI/04.Controllers,%20extensii/01.Controllers/01.CRn/CR1%20Controller%20-%20Instruction%20Manual%20BFP-A8054-H%20\(02.05\).pdf](http://suport.siriustrading.ro/02.DocArh/07.RI/04.Controllers,%20extensii/01.Controllers/01.CRn/CR1%20Controller%20-%20Instruction%20Manual%20BFP-A8054-H%20(02.05).pdf)

- MITSUBISHI
 Mitsubishi Industrial Robot
 RV-1A/RV-2AJ Series
 Standard Specifications Manual (CR1-571 Controller)
<http://www.rixan.com/Portals/0/RV-1A-2AJ/1n2specs.pdf>

- Mitsubishi Electric
 Mitsubishi Electric Industrial Robot
 Safety Manual
http://download.mitsubishi-automation.com/resources/manuals/safety_manual_RVRH.pdf

- Manual COSIMIR Industrial 4.1:
www.festo-didactic.com/ov3/media/customers/1100/00521457001096461528.pdf

Páginas web:

- http://cfievalladolid2.net/tecno/cyr_01/robotica/sistema/morfologia.htm
- <http://html.rincondelvago.com/robots-industriales.html>
- <http://www.google.com.mx/#hl=es&q=sensores+internos+y+externos&meta=&aq=f&oq=sensores+internos+y+externos&fp=934b0323ebe816b3>
- http://cfievalladolid2.net/tecno/cyr_01/robotica/industrial.htm
- <http://www.monografias.com/trabajos16/estacion-robotica/estacion-robotica.shtml>
- <http://www.monografias.com/trabajos10/robap/robap.shtml>
- <http://proton.ucting.udg.mx/robotica/r166/r69/r69.htm>