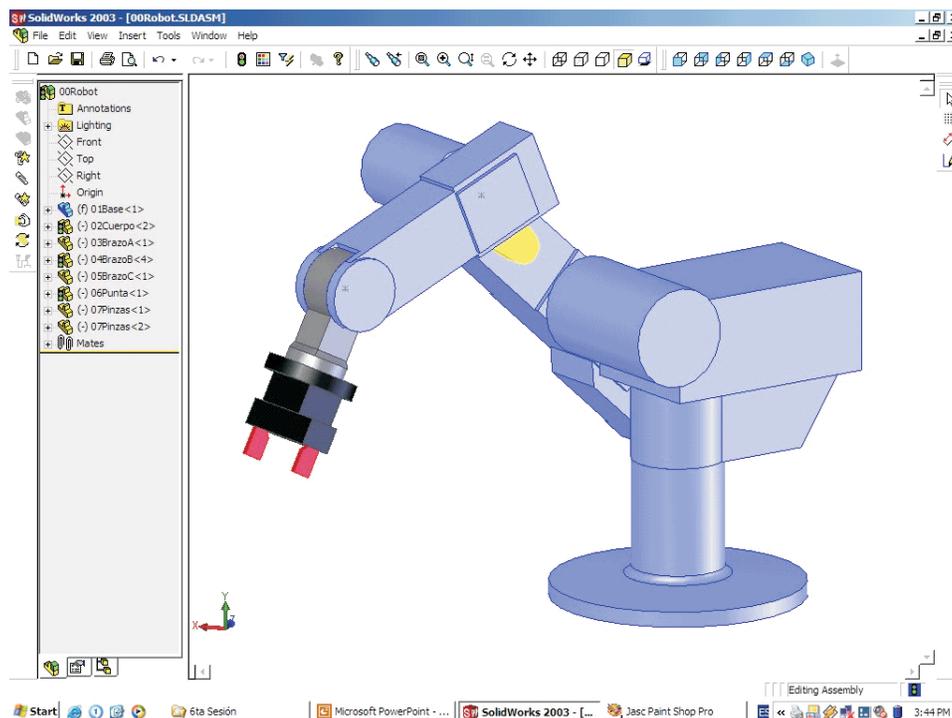




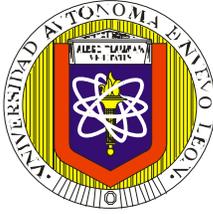
Robótica 102-711

Instructivo de laboratorio

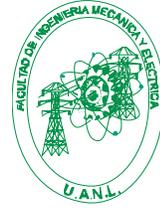


Autores:

Prof.MC. Javier de la Garza Salinas
Prof.MC. Francisco Ramírez Cruz
Prof.MC. José Luis Castillo Ocañas
Prof.Dr. F. Eugenio López Guerrero



Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica



Laboratorio de Robótica

PRÁCTICAS

1. Descripción de la estación de trabajo
2. Procedimiento de encendido, rutina de referencia y apagado del equipo
3. Movimientos manuales del robot
4. Definición de puntos y movimientos
5. Creación y edición de programas
6. Utilización de arreglos
7. Programación condicional
8. Programación con ciclos
9. Definición y uso de puntos en periféricos
10. Trayectorias especiales

PRÁCTICA #1

Descripción de la estación de trabajo

Objetivo de la práctica

El estudiante conocerá el equipamiento del laboratorio en FIME, las reglas de uso de éste y las medidas de seguridad en una celda robotizada.

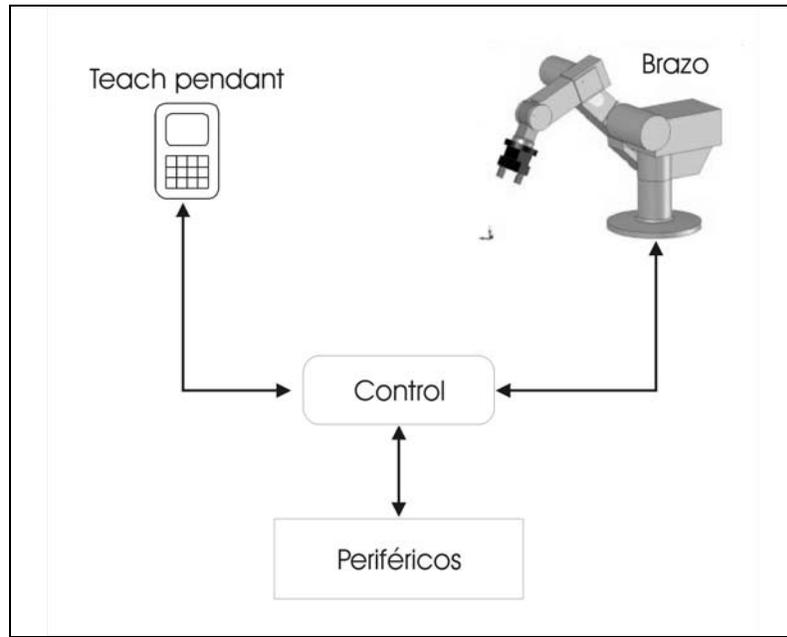


Figura 1a.- Diagrama esquemático de las conexiones de la celda robotizada del laboratorio.

La estación de trabajo consta de una computadora conectada al controlador con la cual el usuario envía órdenes a ejecutar por el robot, utilizando un software de comunicaciones.

La estación está diseñada para manipular material. Por las articulaciones con las que cuenta, el robot es de 5 grados de libertad, y su herramienta de trabajo es una pinza para sujetar objetos (comúnmente denominado *gripper*).

El controlador cuenta con:

- Un botón de encendido principal situado en la parte trasera al circuito electrónico del robot.
- Un botón de encendido de motores situado en la parte frontal. Este interruptor permite que llegue la energía a los motores del robot.
- Botones de paro de emergencia, localizados uno en el controlador y dos en el teach pendant.

REPORTE #1

1. Mencione las reglas para el uso del laboratorio de robótica.
2. Mencione las reglas de seguridad en una celda robotizada.

PRÁCTICA #2

Procedimiento de encendido, rutina de referencia y apagado del equipo

Objetivo de la práctica

El estudiante aprenderá la secuencia de encendido y apagado del equipo, aplicará el concepto de referencia para el robot Eshed.

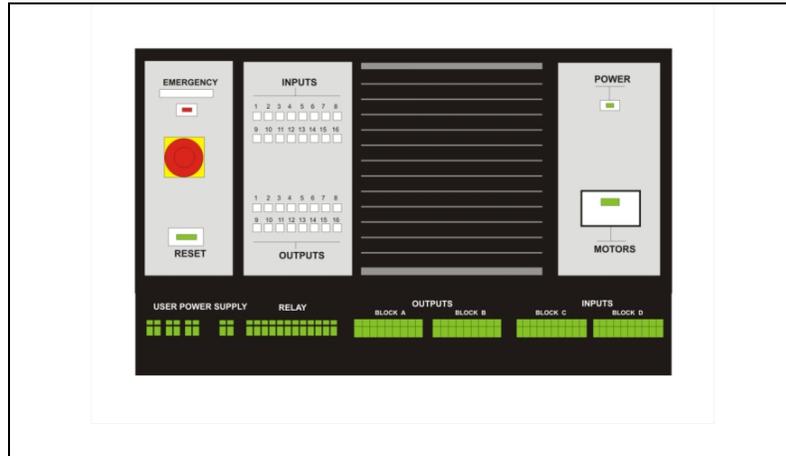


Figura 2a.- Diagrama esquemático de los interruptores de encendido y emergencia en el control del robot.

Pasos:

1. Revisar que los interruptores de emergencia en el control y el teach pendant estén desactivados.
2. Revisar que el interruptor de potencia a los motores esté apagado.
3. Encender el control (el interruptor principal está en la parte trasera del control).
4. Encender la computadora que hace la función de interfase para comunicarse con el robot.
5. Encender el interruptor de potencia de los motores.
6. Seleccionar el icono de robot en el escritorio del usuario de Windows. Esto ejecuta el programa ATS.
7. En el ambiente del ATS, enviar el comando *HOME*.
8. Esperar el resultado de la referencia. El robot debe reportar cada eje como referenciado.
9. Para salir de ATS: <Shift-F9>

REPORTE #2

1. Anexar varios printscreen del escritorio del robot durante el uso del robot desde el encendido hasta su referencia.
2. Copiar exactamente la respuesta del robot cuando se envió el comando *HOME*.

PRÁCTICA #3

Movimientos manuales del robot

Objetivo de la práctica

El estudiante adquirirá la habilidad de manejar el robot mediante el uso del teclado, en modo de juntas y en modo de ejes.

Comandos importantes dentro del programa ATS:

HOME	Realiza test de configuración de los ejes y motores del robot (Se recomienda realizarlo siempre que el robot haya sido apagado o se haya pulsado la seta de emergencia).
CON / COFF	Activa/Desactiva el control del brazo del robot. Estando apagado, la posición del brazo puede variar a causa de la acción de la gravedad.
DIR	Lista los programas almacenados en la memoria del Robot, indicando su nombre, la validez, identidad, y la prioridad de cada programa.

Movimiento manual del robot:

Se activa / desactiva pulsando ALT + m	
MOVIMIENTO POR JUNTAS	MOVIMIENTO EN EJES XYZ
Pulsando la tecla j indicamos desplazamientos por juntas.	Pulsando la tecla x se indica desplazamiento por coordenadas.
Las teclas de movimiento de cada junta son las siguientes:	Las teclas de movimiento son las siguientes:
Junta 1-Q ; Junta 2-W ; Junta 3-E ; Junta 4-R ; Junta 5-T ; Junta 6-T (pinza).	Eje. X: 1-Q ; Eje. Y: 2-W ; Eje. Z: 3-E . Las demás son las mismas que en desplazamiento por juntas.
La tecla s indica la velocidad, espera un valor entre 1:100, la tecla c activa el control y la tecla d lo desactiva.	La tecla s indica la velocidad, espera un valor entre 1:100, la tecla c activa el control y la tecla d lo desactiva.

REPORTE #3

1. Utilice el dibujo del teclado para identificar las combinaciones de teclas utilizadas para mover el robot en modo de juntas y en modo de ejes.
2. Identifique el sensor de proximidad #2 y coloque la pinza del robot normal a la banda transportadora. Tome una fotografía.

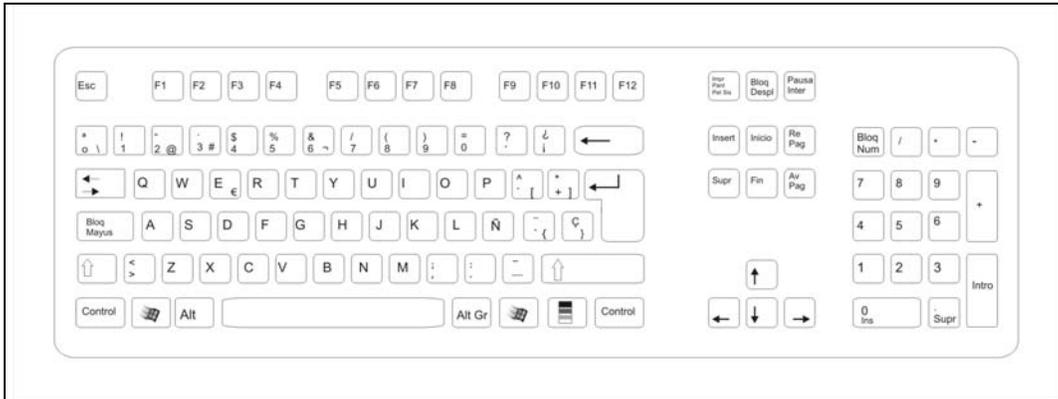


Figura 3a.- Diagrama esquemático de un teclado de computadora.

PRÁCTICA #4

Definición de puntos y movimientos

Objetivo de la práctica

El estudiante diseñará trayectorias simples para el elemento final del robot considerando el punto inicial y el final de la trayectoria.

Variables

Las variables son localizaciones de memoria que almacenan valores enteros, entre -2147483647 a +2147483647 (datos de 32 bits). El ACL usa dos tipos de variables: del sistema y del usuario.

Variables del sistema

El controlador tiene nueve variables del sistema: IN, OUT, ENC, TIME, LTA, LTB, MFLAG, ERROR, ANOUT.

Variables del usuario

Las variables definibles por el usuario pueden ser globales y locales.

Variables globales

Las variables globales se pueden usar en todos los programas. Con el comando GLOBAL se define una variable como global, y con el comando DIMG un vector de variables globales.

Variables locales

Solo se pueden utilizar en el programa en donde han sido definidas. Con el comando DEFINE se define una variable como local, y con el comando DIM un vector de variables locales.

Los nombres usados para definir variables pueden ser una combinación de hasta cinco caracteres alfanuméricos, en donde el primer carácter debe ser una letra.

Posiciones

Las posiciones están almacenadas en un área de memoria reservada para estos datos. Los datos de posiciones incluyen un valor entero para cada eje en el rango -32768 +32768 para definir las coordenadas y una palabra en el rango -32768 +32768 para indicar el tipo y nombre de la posición.

Tipos de posiciones

La memoria de posiciones esta separada para cada grupo de ejes A para los ejes del brazo, B para los accesorios y C para ejes individuales. El máximo número de posiciones para cada grupo es definible en la configuración del controlador. Por default las posiciones se definen para el grupo A

El ACL tiene seis tipos de posiciones listadas a continuación. Los comandos usados para grabar cada tipo de posición aparece en el paréntesis.

- Absoluta juntas (HERE, SETPV, SHIFT): Los datos de la posición son las coordenadas en valores de pasos de encoder.
- Absoluta XYZ (TEACH, SETPV, SHIFTC): Los valores de posición están dados en valores de coordenadas cartesianas.
- Relativa a otra posición por juntas (HERER): los datos de la posición son la diferencia entre los valores de los encoders de una posición y los valores de encoders se la otra posición ACL permite anidar hasta 32 posiciones relativas.

- Relativa a otra posición por XYZ (TEACHR): los datos de la posición son la diferencia entre los valores de coordenadas de ambas posiciones ACL permite anidar hasta 32 posiciones relativas.
- Relativa a la posición actual por juntas (HERER): los datos de la posición son calculados añadiendo los valores de los encoders de la posición a los valores de encoders de la posición actual.
- Relativa a la posición actual por XYZ (TEACHR): los datos de la posición son calculados añadiendo los valores de las coordenadas cartesianas de la posición a los valores de la posición actual.

Definición de posiciones

Los comandos DEFP, DEFPB y DEFPC se utilizan para definir posiciones y los comandos DIMPA, DIMPB y DIMPC se usan para definir vectores de posición.

Definir una posición es reservar un espacio en la memoria del controlador y dar un nombre de localización.

Tabla 4-1.- Ejemplos de definición se posiciones	
DEFP PA	define una posición llamada PA para el grupo A
DIM PP AA[10]	Define un vector 10 d Posiciones llamado AA para el grupo A
DEFPB POSB	Define una posición POSB para el grupo B
DEFPC PC 8	Define una posición PC para grupo C eje 8
DIMPC AC[30]	Define un vector de 30 posiciones AC para el grupo C eje 8

Grabar posiciones

Grabar una posición es escribir sus valores en la posición reservada de la memoria. Para grabar los valores se las posiciones se utilizan los comandos HERE, HERER, TEACH, TEACHR, SETPV y SETPVC.

Tabla 4-2.- Movimiento a una posición deseada	
MOVE <i>pos</i>	El robot se desplaza a la posición <i>pos</i>
MOVEL <i>pos</i>	El desplazamiento se realiza de forma lineal
MOVEC <i>pos</i>	El desplazamiento se realiza mediante interpolación circular
Si a la instrucción de movimiento se le añade D (MOVED,MOVELD,MOVECD), se obliga a que el robot alcance la posición para proceder a ejecutar el siguiente comando.	

Para definir y grabar una posición, debe procederse de la siguiente forma:

1. Entrar en modo manual y llevar al robot a la posición deseada. Salir después del modo manual.

2. Hay que definir la posición, igual que si se tratase de una variable. Esto se hace con el comando `defp`:

`> DEFP <posición>`

3. La posición actual queda grabada en la variable posición mediante el comando:

`> HERE <posición>`

Mediante el comando `listp` pueden listarse todas las posiciones pregrabadas (la posición 0 esta reservada a *HOME*). Es conveniente obtener un listado de posiciones antes de grabar nuevas posiciones, con el objeto de conocer que números de posición se encuentran disponibles. El comando *LISTPV* es similar al anterior, pero además proporciona las coordenadas XYZ de la posición (en décimas de milimetro), así como los ángulos de pitch y roll del robot (en décimas de grado).

Para borrar una posición es necesario ejecutar el comando:

`>DELP <posición>`

Una vez grabadas las posiciones, es posible realizar movimientos entre ellas.

REPORTE #4

1. Defina la posición *FIME1*. Coloque el brazo como se muestra en la figura 4.1(a). y grábela.
2. Defina la posición *FIME2*. Coloque el brazo como se muestra en la figura 4.1(b). y grábela.
3. Utilice para mover el robot a ambas posiciones el comando:
MOVE FIME1
MOVE FIME2
4. Diseñe y reporte una trayectoria a su elección con cuando menos 5 posiciones.

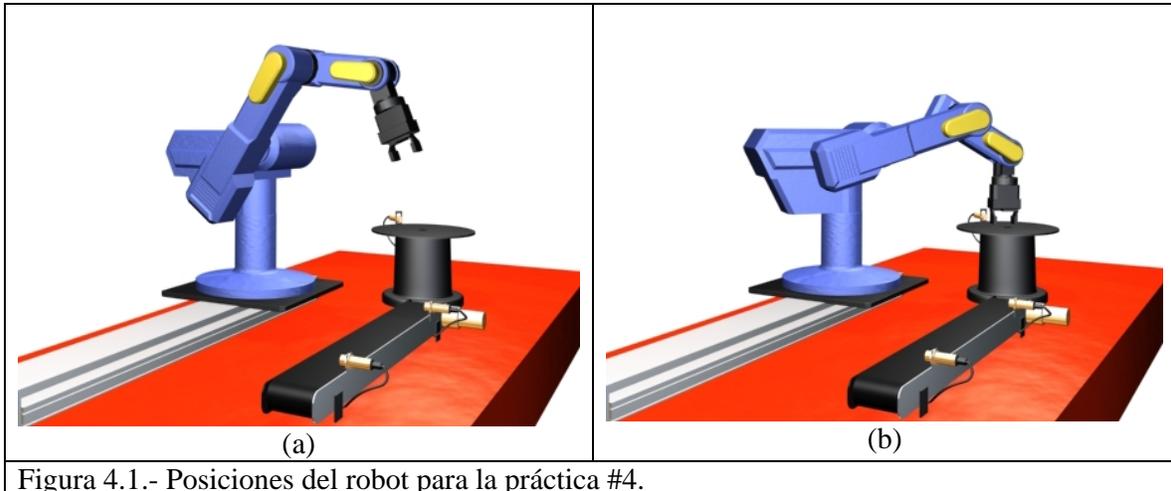


Figura 4.1.- Posiciones del robot para la práctica #4.

PRÁCTICA #5 Creación y edición de programas

Objetivo de la práctica

El estudiante programará el robot para la ejecución de una tarea sencilla de manera automática.

Antes de editar un programa, observe lo siguiente:

Las posiciones deberán estar definidas previo a la creación del programa. Asegúrese que las variables de posición tengan los valores deseados. Recuerde que otros usuarios pudieron haber modificado cualquier posición programada anteriormente. Utilizando los comandos

>SPEED 10

>MOVE pos

podrá mover el robot a la posición que se desee comprobar; mantenga al alcance el botón de paro de emergencia si observa que el robot puede colisionar.

Para la creación del programa se emplea el comando *EDIT* (Ver tabla 5.1). De este modo se podrá editar un programa, utilizando el editor propio del terminal ATS. Este editor tiene su propio funcionamiento, y editar un programa entero mediante este sistema puede ser una tarea laboriosa que editarlo mediante otro medio y transferirlo. Una vez terminada la programación y para salir del modo editor se utiliza el comando *EXIT*. Al realizar esta operación se verificarán los errores.

Tabla 5.1.- Comandos de manipulación de programas

COMANDO	SINTAXIS	EXPLICACIÓN
EDIT	EDIT <prog>	Activa la modalidad de edición. Si prog no existe, confirma la creación de éste
COPY	COPY <prog1> <prog2>	Copia el programa1 a un nuevo programa2
RENAME	RENAME <prog1> <prog2>	Cambia el nombre del programa
REMOVE	REMOVE <prog>	Borra el programa del controlador
LIST	LIST <prog>	Ver el código del programa prog
RUN	RUN <prog>	Ejecutar el código del programa prog

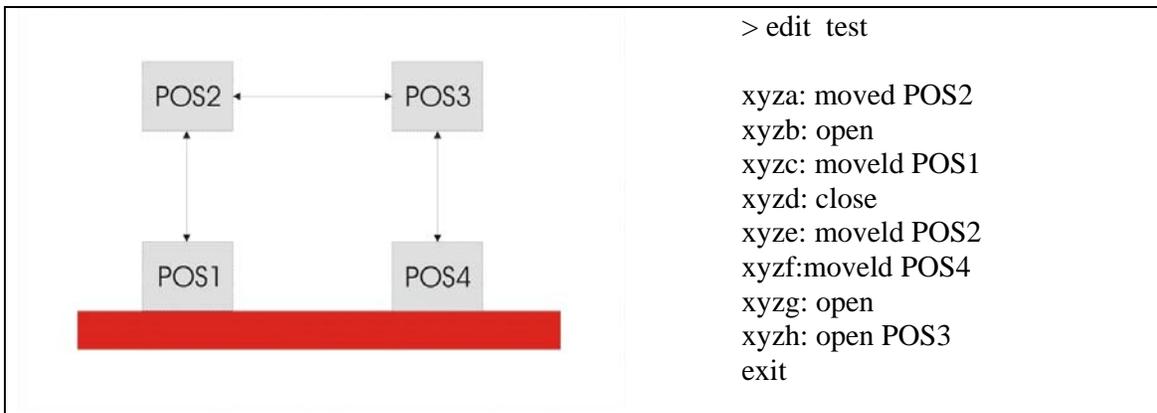


Figura 5.1.- Ejemplo de un programa llamado TEST. Transporta un objeto de la posición #1 a la posición #4.

REPORTE #5

1. Programe el robot con el programa de la figura 5.1. Defina las posiciones en donde usted lo considere adecuado. Tome un video corto con una cámara digital.
2. Imprima los valores de los encoders para cada posición programada de punto 1 de este reporte.

PRÁCTICA #6

Utilización de arreglos

Objetivo de la práctica

El estudiante programará el robot utilizando variables de arreglos de puntos. Conocerá la diferencia con respecto a utilizar variables independientes.

En el ambiente del ATS, el comando a utilizar es:

>DIM <var[n]>

En donde:

<var[n]> es un vector de n variables

Descripción

Define un vector de variables locales de n elementos. Los elementos creados se llaman var[1], var[2],...var[n]. Recuerde que una variable local sólo es reconocida por el programa en el que está definida.

Ejemplo:

DIM locv[20] crea un vector llamado locv que contiene 6 variables locales, LOCv[1],...LOCv[6].

Para borrar el vector de posiciones del control del robot utilice:

>DELP <locv>

en donde:

<locv> es el vector de posiciones definidas por el usuario.

Nota:

Este es un comando de definición. Cuando se escribe, se ejecuta el comando, pero no se crea línea de programa, ni en la modalidad de EDITOR. El primer caracter del nombre debe ser una letra; el número máximo de caracteres es cinco.

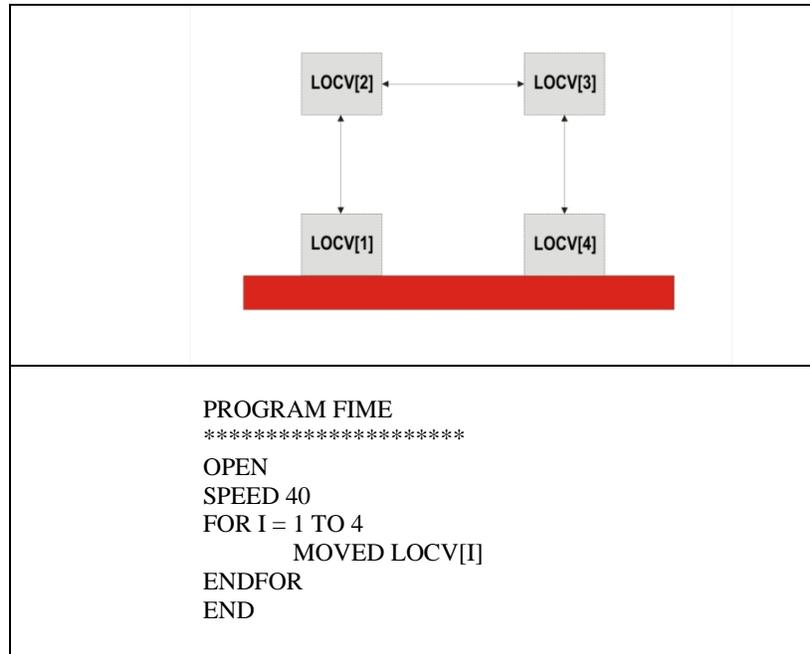


Fig. 6.1.- Ejemplo de un programa en ACL que mueve el robot recorriendo todas las posiciones LOCV previamente definidas.

REPORTE #6

1. Defina un vector de seis posiciones *LOCV* como se describió anteriormente. Grabe las primeras cuatro posiciones como se muestra en la figura 6.1. Reporte los valores de los encoders utilizando el comando *LISTPV*.
2. Haga el programa FIME de la figura 6.1 (removerlo si existe anteriormente) utilizando una variable local *i* y un ciclo *for* para recorrer todas las posiciones.

PRÁCTICA #7

Programación condicional

Objetivo de la práctica

El estudiante programará el robot para que se ejecute movimientos de acuerdo a condiciones que definan la secuencia de operaciones.

La sintaxis *completa* para el control condicional IF es la siguiente:

```
IF <cond>
    Estatutos para cuando la condicion es verdadera
ELSE
    Estatutos para cuando la condicion es falsa
ENDIF
```

Si no existe la necesidad de ejecutar comandos cuando la condición es falsa, es posible usar la versión *corta* del comando IF:

```
IF <cond>
    Estatutos para cuando la condicion es verdadera
ENDIF
```

```
PROGRAM FIME
*****
IF IN[1] = 1
    PRINTLN "sensor 1"
    MOVED POS[1]
ENDIF
IF IN[2] = 1
    PRINTLN "sensor 2"
    MOVED POS[2]
ENDIF
DELAY 30
END
```

Fig. 7.1.- Ejemplo de un programa que utiliza estatutos condicionales para mover el brazo de acuerdo al estado de los sensores de la estación. Note que el vector de posiciones deberá estar definido previamente.

Los operadores booleanos para la evaluación de la condición son:

>	mayor que
<	menor que
=	igual que
<=	menor o igual que
>=	mayor o igual que
<>	diferente que

REPORTE #7

1. Conecte el LED semáforo del robot y haga un programa que ejecute un movimiento en 4 posiciones. El color del LED deberá ser determinado por el estado de los sensores de proximidad #1 y #2 de la estación, según la siguiente tabla:

SENSOR #1	SENSOR #2	COLOR DEL LED	COMANDOS DEL ROBOT
0	0	APAGADO	SET OUT[1]=0 SET OUT[2]=0
0	1	VERDE	SET OUT[1]=1 SET OUT[2]=0
1	0	ROJO	SET OUT[1]=0 SET OUT[2]=1
1	1	AMARILLO	SET OUT[1]=1 SET OUT[2]=1

2. Tome una fotografía digital con el efecto que se muestra en la figura 7.2 y entréguela junto con la documentación completa de su programa (código, valores de posición, gráficas de posición del brazo) como parte de su reporte.



Figura 7.2.- Fotografía (a) de un artista dibujando con una lámpara y (b) del robot en movimiento realizando una figura en el aire. Ambas fotos fueron tomadas utilizando una cámara fotográfica con el obturador abierto.

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
PROGRAMA DE LA MATERIA

NOMBRE DE LA MATERIA: ROBÓTICA

CLAVE: 102-711

REQUISITOS:

AREA: Manufactura

FRECUENCIA: 3 sesión/semana

SESIONES TOTALES: 48

CREDITOS: 6

FUNDAMENTO DE LA MATERIA:

La robótica está experimentando un gran crecimiento provocado por los avances en computación, sensores, automatización, electrónica y programación de computadoras. Los robots están transformando la forma en que se produce, por lo que se hace necesario que los estudiantes de ingeniería tengan bases sólidas para su utilización, implementación y mantenimiento en trenes productivos.

OBJETIVO:

El estudiante conocerá los principios de ingeniería en que se basa la construcción y capacidades de trabajo de los brazos mecánicos.

Al finalizar este curso, el estudiante será capaz de:

- operar un robot mediante el uso de software y aditamentos necesarios para su utilización.
- acondicionar, diseñar aditamentos y programar un ambiente de interacción con un robot.

TEMARIO:

1. Introducción a los robots
2. Programación
3. Diseño de trayectorias
4. Aplicaciones

BIBLIOGRAFIA:

1. ASADA&SLOTINE. *Robot Analysis and Control*. John Wiley, 1986. ISBN 0-471-83029-1
2. ANIBAL OLLERO BATURONE. *Robótica Manipuladores y Robots Móviles*. Alfa, 2001. ISBN 84-267-1313-0.
3. YORAM KOREN. *Robotics for Engineers*. Mc.Graw Hill, 1987. ISBN 0-07-035399-9
4. WOLFRAM STADLER. *Analytical Robotics & Mechanics*. Mc.Graw Hill, 1995. ISBN 0-07-060608-0
5. LORENZO SCIAVICCO/ Bruno Siciliano. *Modeling and Control of Manipulators*.

REVISIÓN No.: 1

VIGENTE A PARTIR DEL: 2 de agosto del 2002

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
PROGRAMA DE LA MATERIA

Mc.Graw Hill, 1996. ISBN 0-07-057217-8

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
PROGRAMA DE LA MATERIA

1. Introducción a los robots

Duración aprox.: 12 sesiones

Objetivo particular: El estudiante conocerá los diferentes tipos de Robots y sus parámetros de clasificación.

Temario:

- Introducción
- Tipos de articulaciones
- Estructuras básicas
- Tipos de articulaciones
- Robots móviles

2. Programación

Duración aprox.: 12 sesiones

Objetivo particular: El estudiante adquirirá la habilidad de manejar el robot mediante el uso del Teach-Pendant.

El estudiante programará los movimientos del Robot

Temario:

- Sistemas de programación
- Programación por guiado
- Programación textual
- Sistemas de referencia
- Especificación de movimientos

3. Diseño de Trayectorias

Duración aprox.: 12 sesiones

Objetivo particular: El estudiante diseñará trayectorias para el elemento final del robot utilizando ecuaciones de movimiento, considerando puntos interpolados intermedios.

Temario:

- Planteamiento del problema
- Definición paramétrica de curvas
- Técnicas de interpolación
- Espacio cartesiano
- Trayectorias articulares

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
PROGRAMA DE LA MATERIA

- Generación en tiempo real

4. Aplicaciones

Duración aprox.: 12 sesiones

Objetivo particular: El estudiante conocerá los diferentes tipos de aplicación de robots en la industria.

Temario:

- Soldadura con robots
- Pintura con robots
- Manipulación de materiales con robots
- Ensamble con robots
- Dosificación de fluidos con robots

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
PROGRAMA DE LA MATERIA

Responsable de la elaboración del programa:

Cuerpo Académico Sistemas Integrados de Manufactura.

Prof.MC. José Luis Castillo Ocañas	Prof.MC. Fco. Javier de la Garza Salinas
Prof.MC. Francisco Ramírez Cruz	Prof.Dr. Fco. Eugenio López Guerrero

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
PROGRAMA DE LA MATERIA

Responsable de la implementación del programa:

JEFATURA DE ACADEMIA:
Prof.MC. Huemack Eduardo Rivera Zapata

JEFATURA DE DEPARTAMENTO
Prof.Dr. Fco. Eugenio López Guerrero

COORDINACIÓN DE:
División de Ingeniería Mecánica
Prof.MC. Francisco Ramírez Cruz

SUBDIRECCIÓN ACADÉMICA:
Prof.Dr. Moisés Hinojosa Rivera