

Universidad Nacional de Quilmes

INFORME: IACI00-01

TEMA: Scrobot ER-Vplus
y Scrobot ER-IX

AUTOR: Claudio Martín Syd

FECHA: abril del 2000

LUGAR: Universidad Nacional de Quilmes
(sede Florencio Varela)

CONTENIDOS:

Descripción de los brazos robóticos SCORBOT ER-Vplus y SCORBOT ER-IX, modos de operación, sistemas de coordenadas, periféricos y lenguaje de programación ACL

Indice

Introducción	pag 3
--------------------	-------

Capitulo 1 SCORBOT-ER Vplus

1.1 Descripción	pag 5
1.2 Puesta en marcha	pag 6
1.3 Modos de operación	pag 6
1.4 Teach pendant	pag 7
1.5 Inicialización	pag 8
1.6 Operación del robot	pag 8
1.7 Grupos de control de ejes	pag 9
1.8 Sistemas de coordenadas	pag 9
1.9 Disposición de periféricos y sensores	pag 10

Capitulo 2 SCORBOT-ER IX

2.1 Descripción	pag 12
2.2 Puesta en macha	pag 12
2.3 Operación del robot	pag 13
2.4 Teach pendant	pag 14
2.5 Sistemas de coordenadas	pag 15

Capitulo 3 Lenguaje de programación ACL

3.1 Definición y grabado de posiciones	pag 17
3.2 Movimiento a posiciones grabadas	pag 19
3.3 Comandos de control de ejes	pag 19
3.4 Comandos de control de programas	pag 20
3.5 Comandos de manipulación de programas	pag 21
3.6 Comandos de reporte	pag 21
3.7 Comandos de control I/O	pag 21
3.8 Comandos de definición y manipulación de variables	pag 22
3.9 Comandos de programación	pag 22
3.10 Ejemplo de programación	pag 23

Capitulo 4 Scorbace para windows.

4.1 Descripción	pag 26
4.2 Edición de programas	pag 27
4.3 Definición y grabado de posiciones	pag 27
4.4 Ejecución de programas	pag 27

Precauciones	pag 28
Conclusiones	pag 29
Referencias	pag 29

Introducción

El presente informe fue desarrollado para el SCORBOT ER Vplus y el SCORBOT ER IX, está concebido para las exigencias de la enseñanza orientada hacia la práctica. Los objetivos del mismo son proveer información rápida acerca de ambos equipos, desde las características físicas principales hasta la operación del brazo robótico, pasando por la disposición de los periféricos y sensores. Se describen tanto el software como el hardware en un enfoque global facilitando el manejo. Cabe destacar el carácter superficial de este informe ya que es una síntesis de los manuales de operación dados por el fabricante.

El Capítulo 1 se presenta de la siguiente manera: En un primer paso se describe el robot y se menciona el nombre de sus brazos y juntas, luego se abordan temas como la puesta en marcha y los modos de operación, así como el uso de un dispositivo de gran ayuda en el aprendizaje como el teach pendant. También se explican temas como: sistemas de coordenadas y grupos de control de ejes, útiles a la hora de operar el robot.

En cuanto a los contenidos, el Capítulo 2 se presenta de la siguiente manera: En un primer paso se describe el robot y se menciona el nombre de sus brazos y juntas, luego se abordan temas como la puesta en marcha y los modos de operación, así como el uso de un dispositivo de gran ayuda en el aprendizaje como el teach pendant. Otros temas como: sistemas de coordenadas y grupos de control de ejes pueden leerse en el capítulo 1 correspondiente al SCORBOT ER Vplus ya que son compatibles.

En el Capítulo 3 se describen las principales instrucciones del ACL que es el lenguaje de programación utilizado en el robot, culminando el capítulo con un ejemplo práctico.

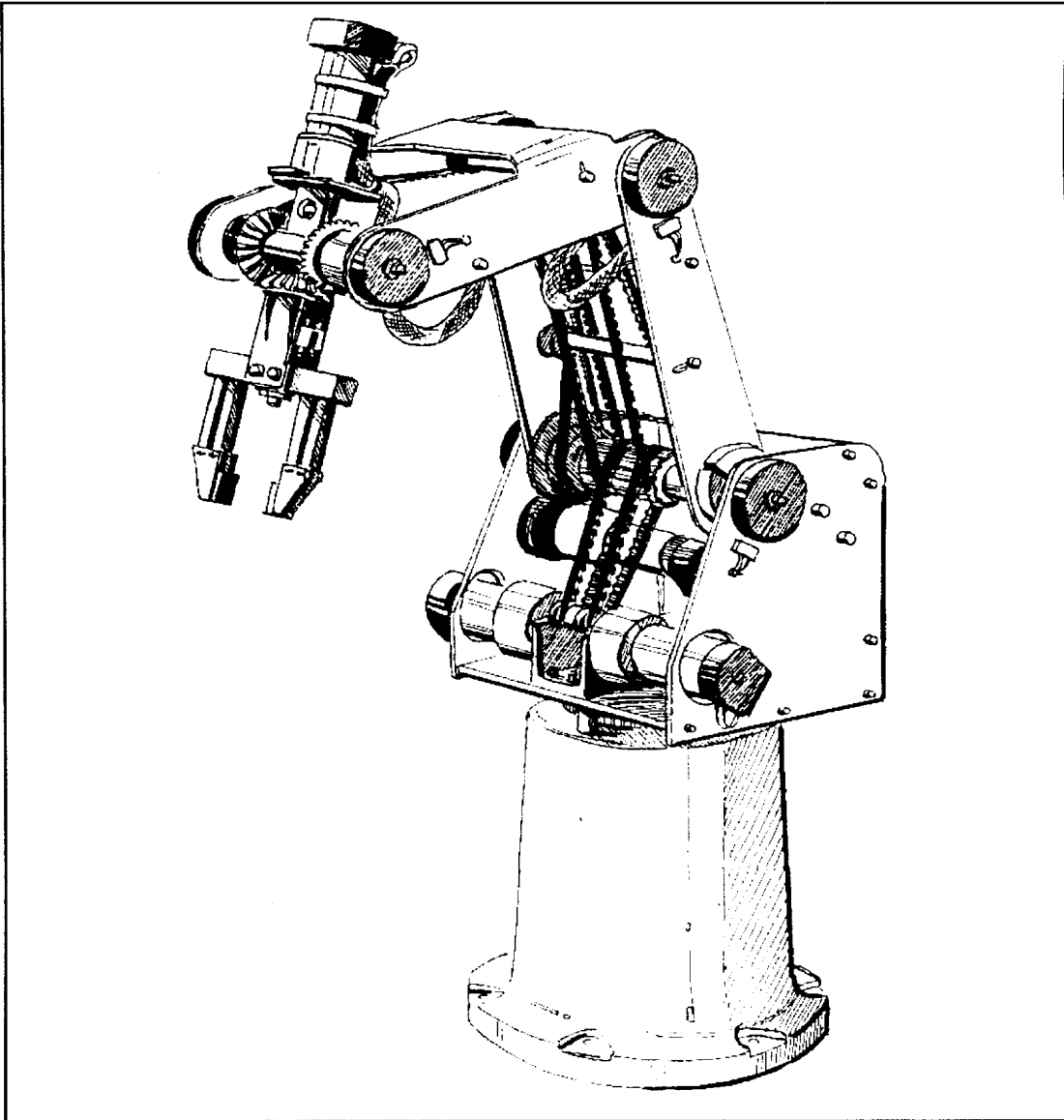
Por último veremos las precauciones que se deben tener en cuenta a la hora de operar el robot, ya que un uso indebido del mismo puede causar daños en el equipo o accidentes en el operador.

El informe no cuenta con detalles de los mecanismos ni con hojas de datos correspondientes a cada una de las partes, esto implica que para reparaciones o para detectar errores de funcionamiento deberá consultarse otro material de referencia o acudir al fabricante.

Para principiantes se recomienda comenzar operando el SCORBOT ER Vplus ya que en principio es igual que el SCORBOT ER IX pero más pequeño y manejable, esto proporciona al aprendiz la posibilidad de introducirse al mundo de la robótica con mayor facilidad.

Capitulo 1

Scorbot – ER Vplus

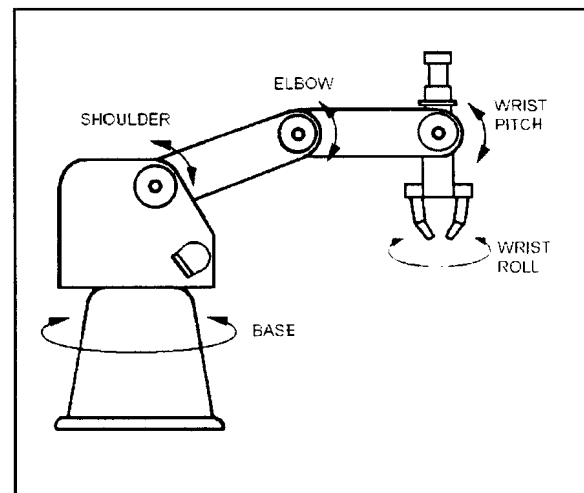
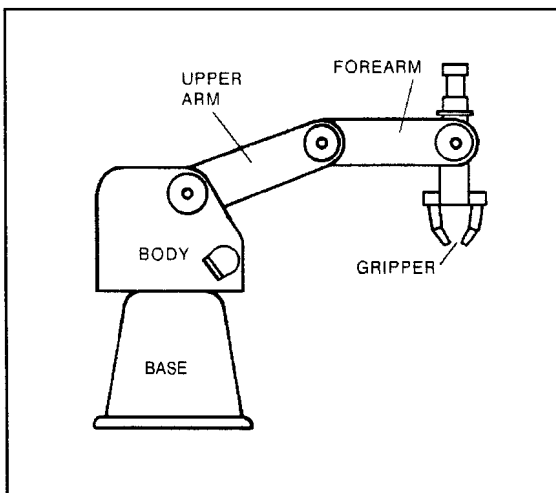


1.1 Descripción:

El **Scorbot – ER Vplus** es un robot vertical articulado con cinco juntas rotativas. Con el agregado de la mordaza, el robot posee seis grados de libertad.

Los movimientos de las juntas están descritas en la siguiente tabla:

Num. De eje	Nombre de la junta	Acción
1	Base	Gira el cuerpo (body)
2	Shoulder	Sube o baja el brazo superior (uper arm)
3	Elbow	Sube o baja el antebrazo (fore arm)
4	Wrist pitch	Sube o baja la pinza (gripper)
5	Wrist roll	Gira la pinza (gripper)



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

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

1.2 Puesta en marcha:

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

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

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

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

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

1.3 Modos de operación:

Software

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

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

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

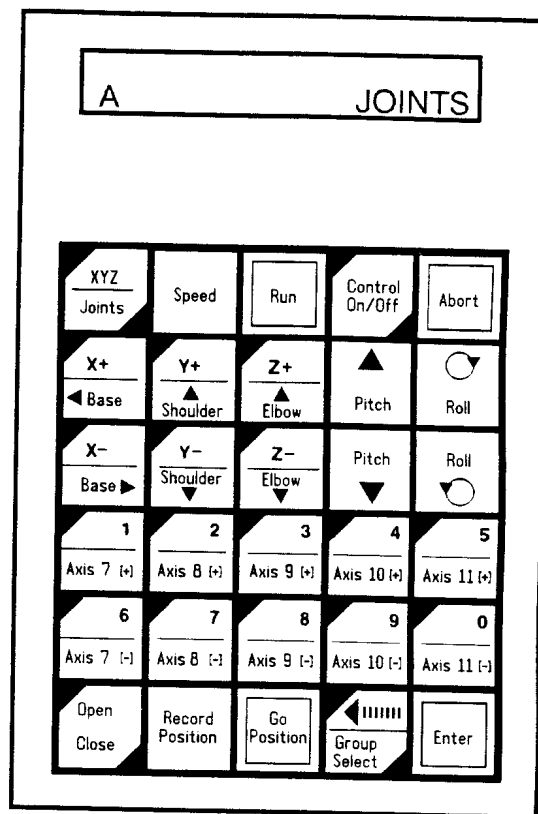
- Forma corta de configurar el controlador
- Definición de periféricos
- Editor de programas
- Teclas de short-cut para entrar
- Backup manager

1.4 Teach pendant

El **teach pendant** es un dispositivo opcional, es una terminal de mano, usado para controlar el robot y los periféricos conectados al controlador. Es practico para el movimiento de ejes, grabado de posiciones, direccionamiento de los ejes a posiciones grabadas y ejecución de programas.

Trabaja tanto en coordenadas joint como en cartesianas.

El teach pendant puede verse en la siguiente figura:



Teach Pendant

1.5 Inicialización

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

Para ejecutar el homing desde la PC tipee:

>home <enter>

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

run – 0 - enter

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

>home *nº de eje*

1.6 Operación del robot

MANUAL MODE (Modo manual)

El modo manual es accionable cuando el sistema esta en modo directo, este modo permite el control directo de los ejes sin la necesidad de usar el TEACH PENDANT.

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

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

DIRECT MODE (Modo directo)

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

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

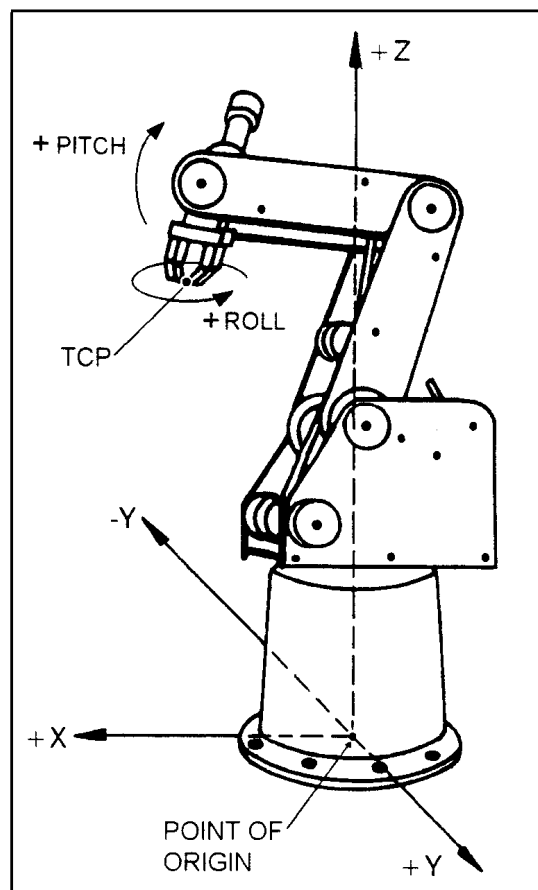
1.7 Grupos de control de ejes

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

1.8 Sistemas de coordenadas

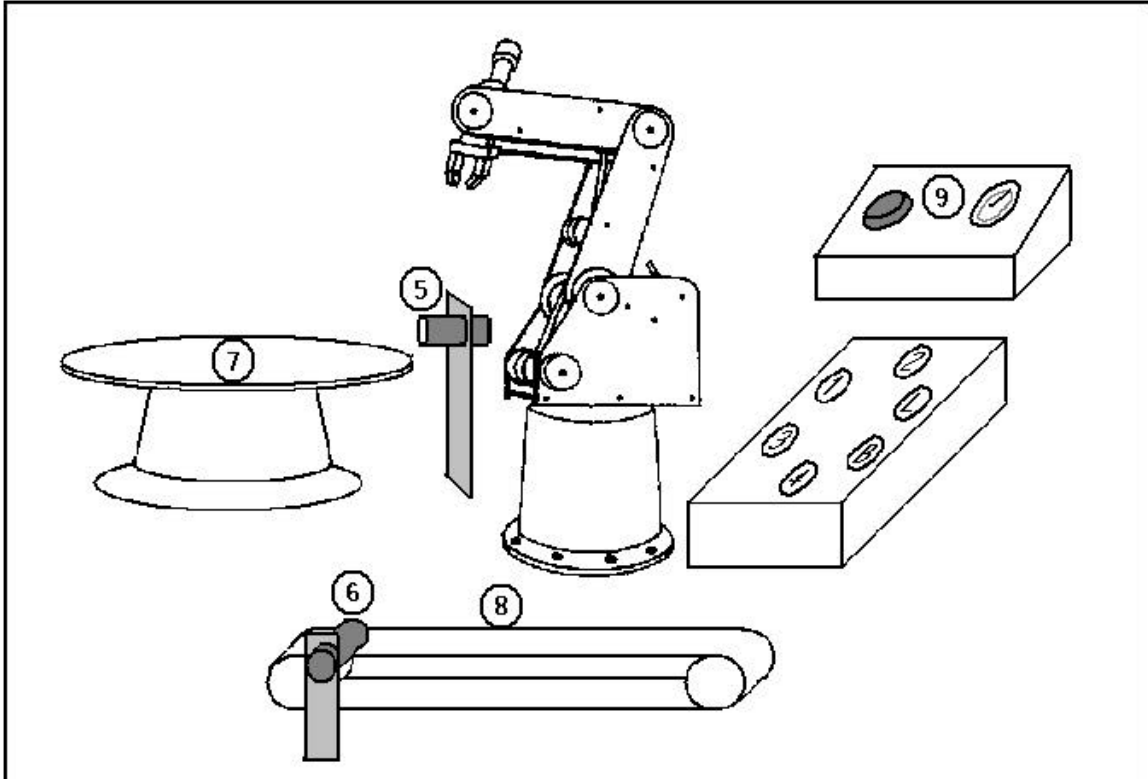
Coordenadas joint: especifican la posición de los ejes robóticos en unidades de encoders, ya que estos proporcionan un número de señales proporcional al movimiento de los ejes.

Coordenadas cartesianas: es un sistema geométrico para especificar la posición del punto central de la herramienta del robot TCP definiendo la distancia en unidades lineales (décimas de milímetros) desde el origen situado en la base



1.9 Disposición de periféricos y sensores

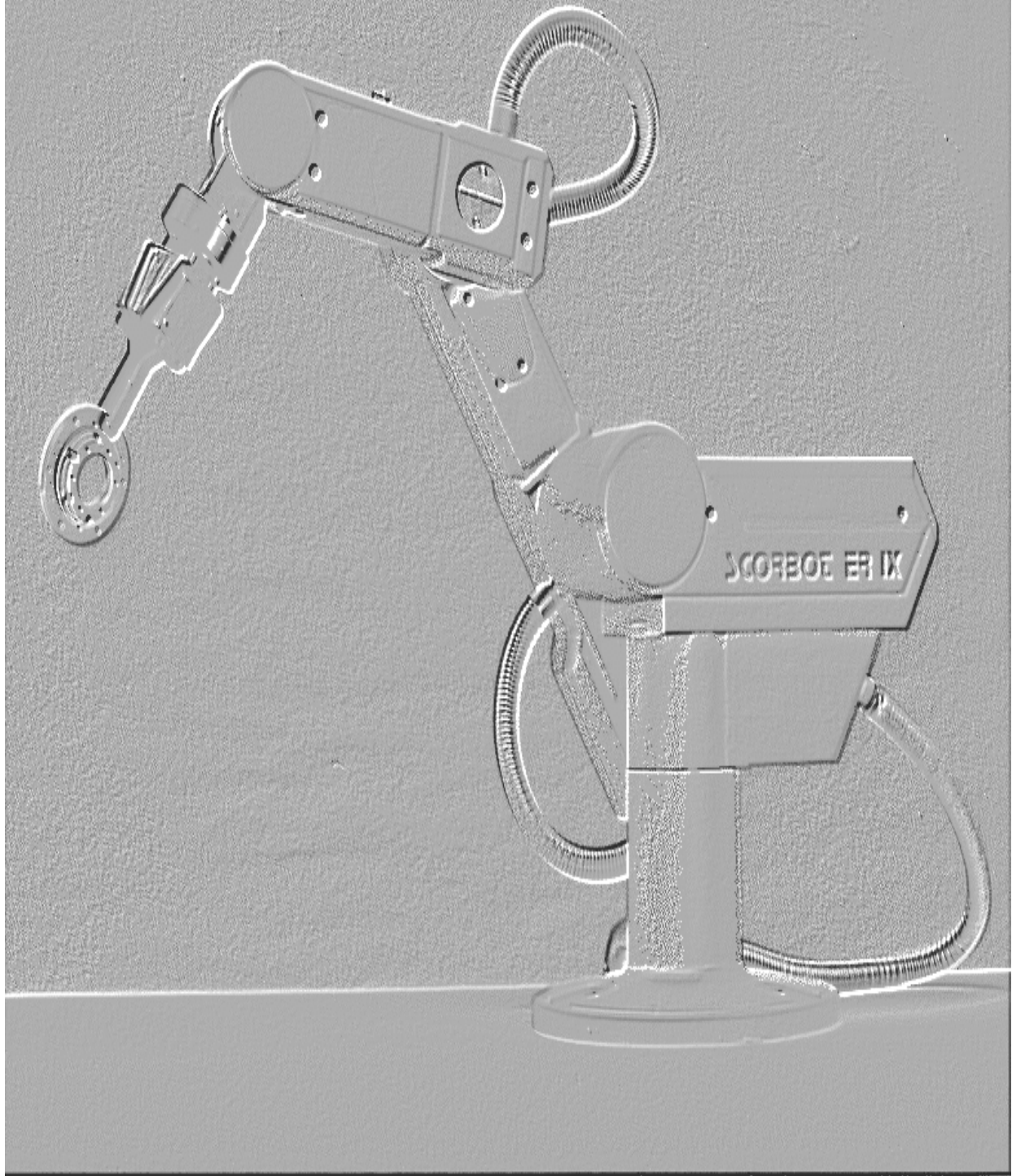
La siguiente es la disposición de sensores y periféricos de la instalación del equipo en los laboratorios de la Universidad Nacional de Quilmes



REF	DESCRIPCION	CONECTADO A
1	Sensor mecánico	IN 1
2	Sensor mecánico	IN 2
3	Sensor mecánico	IN 3
4	Sensor mecánico	IN 4
L	Luz	OUT 1
B	Bocina	OUT 2
5	Sensor de proximidad	IN 5
6	Sensor de proximidad	IN 6
7	Disco giratorio	AXES 7
8	Cinta transportadora	AXES 8
9	Controlador neumático	OUT 3

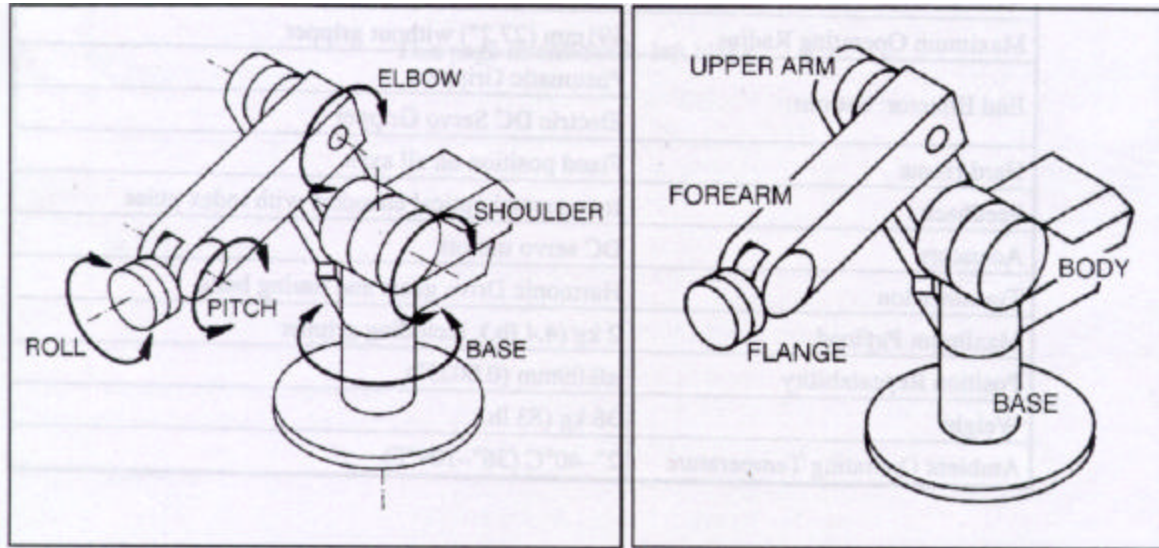
Capitulo 2

Scorbot ER-IX



2.1 Descripción:

El **scorbot – ER IX** es un robot vertical articulado con cinco juntas rotativas. Con el agregado de la mordaza y el riel, el robot posee siete grados de libertad.



Los movimientos de las juntas están descritas en la siguiente tabla:

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

2.2 Puesta en marcha:

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

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

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

4- encienda su computadora y arranque en modo DOS
posicionese en el directorio ATS y tipee: `ats / c2`

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

2.3 Operación del robot

MANUAL MODE (Modo manual)

El modo manual es accionable cuando el sistema esta en modo directo, este modo permite el control directo de los ejes sin la necesidad de usar el TEACH PENDANT.

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

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

DIRECT MODE (Modo directo)

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

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

.

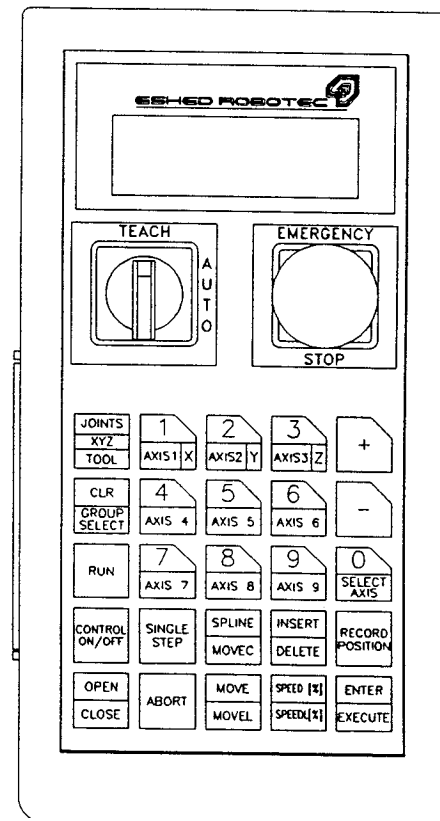
2.4 Teach pendant

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

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

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

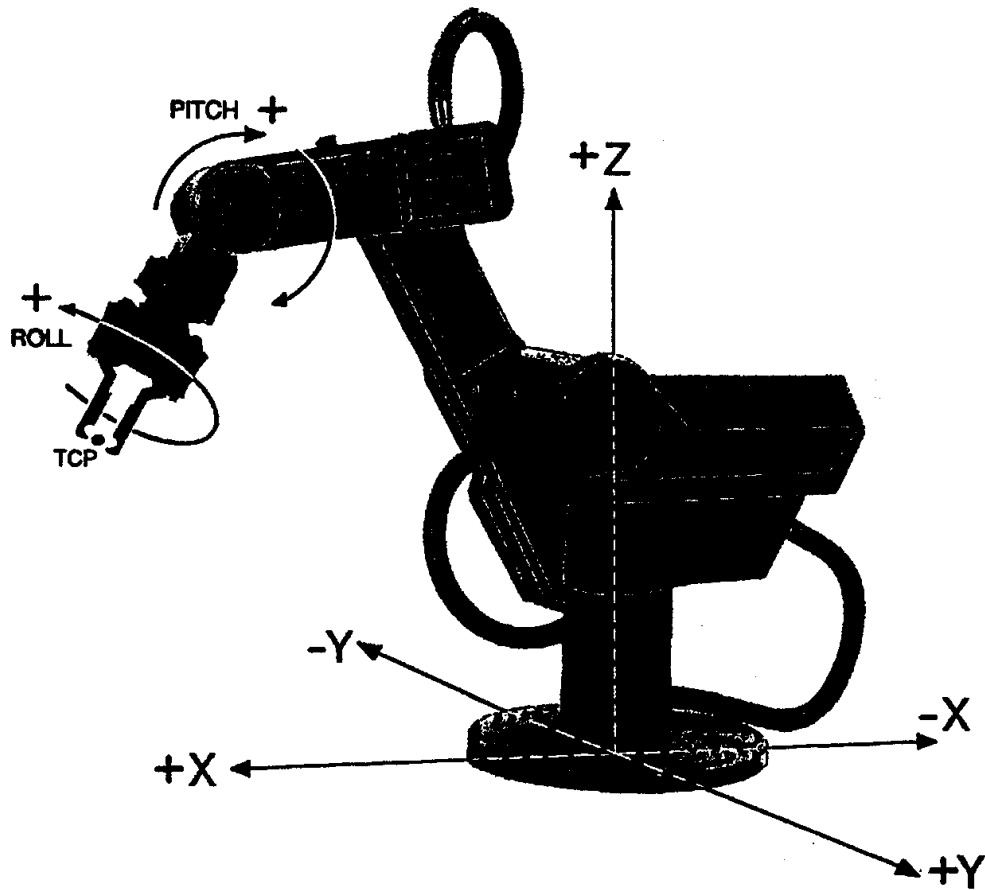
El teach pendant puede verse en la siguiente figura.



2.5 Sistemas de coordenadas

Coordenadas joint: especifican la posición de los ejes robóticos en unidades de encoders, ya que estos proporcionan un número de señales proporcional al movimiento de los ejes.

Coordenadas cartesianas: es un sistema geométrico para especificar la posición del punto central de la herramienta del robot TCP definiendo la distancia en unidades lineales (décimas de milímetros) desde el origen situado en la base



Capitulo 3

ACL

ADVANCED CONTROL LANGUAGE

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

3.1 Definición y grabado de posiciones

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

- (D,E) **Defp** *pos* define una posición llamada *pos* en el grupo A
- (D,E) **Defpb** *pos* define una posición llamada *pos* en el grupo B
- (D,E) **Defpc** *pos* define una posición llamada *pos* en el grupo C

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

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

Cada posición se llamará *pvec* [1], *pvec* [2], ... *pvec* [n]

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

- (D,E) **Delp** *pos* borra la posición existente llamada *pos*
- (D,E) **Delp** *pvec* borra el vector de posiciones *pvec*

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

- (D,E) **Undef** *pos* borra el contenido de la posición existente llamada *pos*
- (D,E) **Undef** *pvec* borra el contenido del vector de posiciones *pvec*

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

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

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

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

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

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

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

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

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

```
>teach pos  
  X -- [.] >  
  Y -- [.] >  
  Z -- [.] >  
  P -- [.] >  
  R -- [.] >
```

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

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

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

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

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

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

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

3.2 Movimiento a posiciones grabadas

- (D,E) **Move** *pos* el robot se mueve a la posición grabada en *pos*.
- (D,E) **MoveI** *pos* la pinza se mueve desde la posición actual hasta la posición *pos*. en línea recta, siempre que esto sea posible.
- (D,E) **MoveC** *pos1 pos2* mueve la pinza desde la posición en que se encuentra hasta la posición *pos1* siguiendo una trayectoria circular a través de la posición *pos2* .
- (E) **Moved** *pos* el robot se mueve a la posición grabada en *pos* y hasta que no llega hasta dicha posición no continúa con el programa .
- (E) **MoveId** *pos* la pinza se mueve desde la posición actual hasta la posición *pos*. en línea recta, siempre que esto sea posible y no continúa con el programa mientras no haya completado el .
- (E) **MoveCd** *pos1 pos2* mueve la pinza desde la posición en que se encuentra hasta la posición *pos1* siguiendo una trayectoria circular a través de la posición *pos2* y no pasa a la siguiente línea de programa hasta que no se haya completado el movimiento.

3.3 Comandos de control de ejes

- (D,E) **Open** abre la pinza
- (D,E) **Close** cierra la pinza
- (D) **Con** habilita el servo control de todos los ejes, o de algún grupo en particular
- (D) **Con axis** habilita el servo control de un eje especificado en *axis*
- (D) **Coff** deshabilita el servo control de todos los ejes, o de algún grupo en particular
- (D) **Coff axis** deshabilita el servo control de un eje especificado en *axis*
- (D,E) **Set anout[n] = DAC** deshabilita el control del eje n y setea el valor de la tensión

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

3.4 Comandos de control de programas

Run: permite ejecutar programas ya editados.

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

A: aborta la ejecución de programas.

- (D) **A prog** aborta la ejecución del programa *prog*.
- (D) **<ctrl> + A** aborta la ejecución de todos los programas que estén corriendo

Stop: aborta la ejecución de programas.

- (E) **Stop prog** aborta la ejecución del programa *prog*.
- (E) **Stop** aborta la ejecución de todos los programas que estén corriendo

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

- (D,E) **Suspend prog** interrumpe la ejecución del programa.
- (D,E) **Continue prog** continua la ejecución del programa interrumpido
- (E) **Delay var** suspende la ejecución del programa durante el tiempo(en centésimas de segundo) especificado en *var*.
- (E) **Wait var1 oper var2** suspende la ejecución del programa hasta que la condición se satisfaga.
- (E) **Trigger prog by {in/out} n {0/1}** ejecuta un programa condicional al cambio en el estado de la entrada o salida *n*.

3.5 Comandos de manipulación de programas

- (D) **Copy *prog1 prog2*** copia el programa *prog1* a un nuevo programa llamado *prog2*
- (D) **Rename *prg1 prg2*** cambia el nombre del programa *prg1* a *prg2*
- (D) **Remove *prog*** borra el programa *prog* y libera la memoria ocupada por él.
- (D) **Empty *pro*** borra las líneas del programa *prog* manteniéndolo existente y válido.
- (D) **Edit *prog*** activa el modo edit y llama al programa *prog* y permite modificar su contenido.
- (E) **Exit** sirve para salir del modo edit.

3.6 Comandos de reporte

- (D) **Dir** proporciona la lista de programas guardados en ram
- (D) **List *prog*** muestra el listado de instrucciones que componen el programa *prog*
- (D) **Listp** muestra una lista de todas las posiciones definidas
- (D) **Listpv *pos*** muestra en pantalla las coordenadas correspondientes a la posición *pos*

3.7 Comandos de control I/O

- (D) **Disable {in/out} *n*** deshabilita la entrada o salida *n*
- (D) **Enable {in/out} *n*** habilita la entrada o salida *n*
- (D) **Force{in/out} *n*{1/0}** fuerza la entrada o salida *n* a un estado 1 o 0
- (D) **Show din** muestra el estado de las 16 entradas
- (D) **Show dout** muestra el estado de las 16 salidas/entradas
- (D) **Set out[n]={1/0}** setea el estado de una salida.

3.8 Comandos de definición y manipulación de variables

- (E) **Define** *var1 ... var12* define hasta 12 variables locales.
- (D,E) **Global** *var1 ... var12* define hasta 12 variables globales.
- (E) **Dim** *var[n]* define un vector de n variables locales.
- (D,E) **Dimg** *var[n]* define un vector de n variables globales.
- .
- (D,E) **Delvar** *var* elimina la variable *var*.
- (D) **Purgue** elimina todas las variables no utilizadas

3.9 Comandos de programación

- (E) **If** *var1 oper var2* inicia una subrutina si la relación entre dos variables es verdadera, el operador puede ser <, >, =, <=, >=, <>
- (E) **Andif** *var1 oper var2* combinación lógica con otro comando if
- (E) **Orif** *var1 oper var2* combinación lógica con otro comando if
- (E) **Else** se ubica entre el if y el endif, comienza una subrutina cuando la condición if es falsa
- (E) **Endif** fin de la subrutina if
- (E) **For** *var1=var2 to var3* ejecuta una subrutina para todos los valores de la variable
- (E) **Endfor** fin de la subrutina for
- (E) **Label** *n* marca una subrutina de programa para ser ejecutada con goto
- (E) **Goto** *n* continua la ejecución del programa en la línea especificada después de label
- (E) **Gosub** *prog* transfiere el control al programa *prog* el programa actual se suspende mientras se ejecuta la subrutina.

3.10 Ejemplo de programación en ACL

A continuación, veremos un ejemplo de programación en ACL, describiendo la función de cada línea.

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

```
PROGRAM DEMO
*****

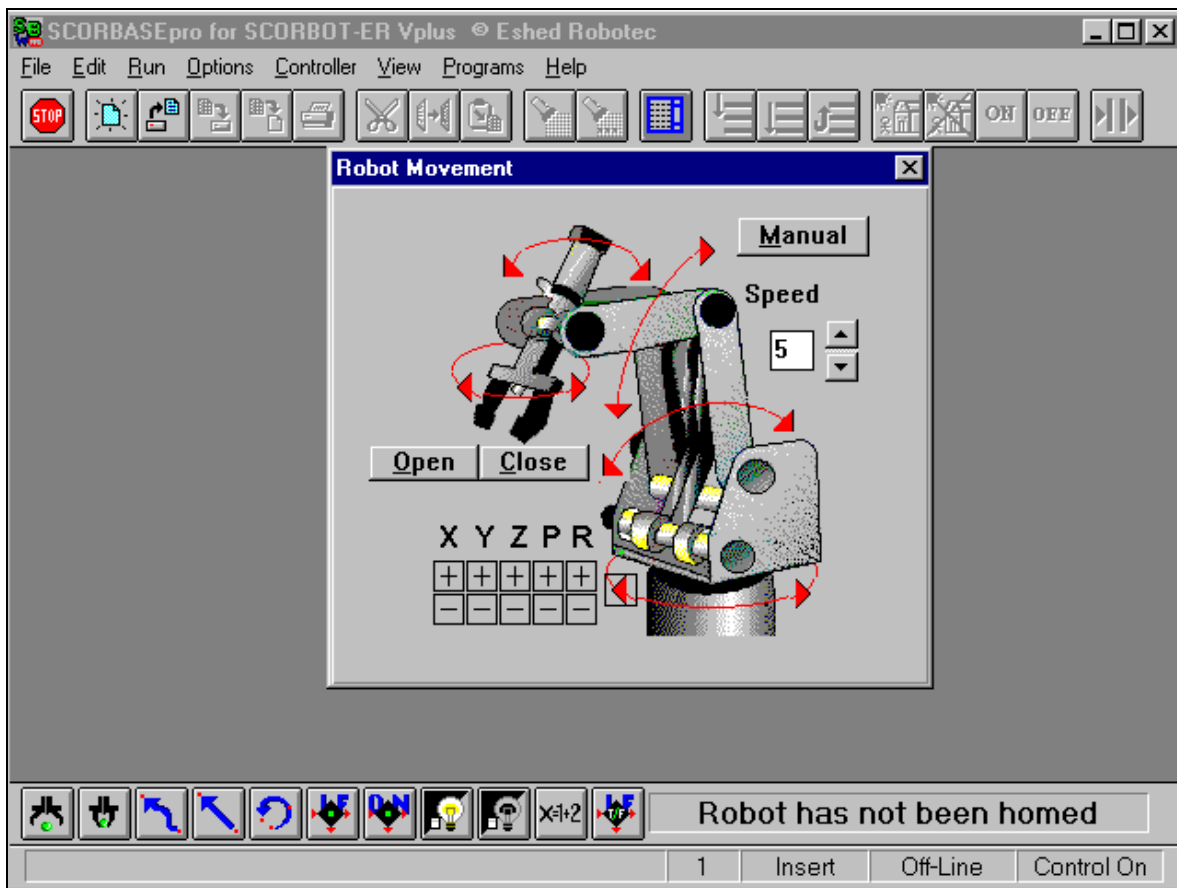
MOVED 0 (lleva la pinza a la posición 0)
SPEED 20 (fija la velocidad a un 20% de la velocidad máxima)
LABEL 1 (establece una marca en el programa)
IF IN[3] = 0 (ejecuta la subrutina siempre que el estado de la entrada 3 sea 0)
GOTO 1 (el programa continua en la línea marcada con label 1)
ENDIF (fin de la subrutina if)
OPEN (abre la pinza)
MOVE DEMO[10] (mueve la pinza a la posición demo[10])
MOVED DEMO[1] (mueve la pinza a la posición demo[1])
CLOSE (cierra la pinza y toma el objeto)
MOVECD DEMO[3] DEMO[2] (va con movimiento circular a la posición demo[2] pasando por demo[3])
OPEN (abre la pinza)
MOVED DEMO[4] (lleva la pinza a la posición demo[4])
SET ANOUT[8] = -1000 (fija una tensión de -1 volt en el eje 8)
WAIT IN[6] = 1 (detiene el programa hasta que se active el sensor 6)
DELAY 160 (detiene el programa 1,6 seg.)
SET ANOUT[8] = 0 (fija una tensión de 0 volt en el eje 8)
MOVED DEMO[5] (lleva la pinza a la posición demo[5])
CLOSE (cierra la pinza)
MOVED DEMO[6] (lleva la pinza a la posición demo[6])
MOVED DEMO[7] (lleva la pinza a la posición demo[5])
OPEN (abre la pinza)
MOVED DEMO[8] (lleva la pinza a la posición demo[8])
SET ANOUT[7] = 1500 (fija una tensión de 1,5 volt en el eje 7)
DELAY 1000 (detiene la ejecución del programa durante 10 segundos)
WAIT IN[5] = 1 (detiene la ejecución del programa hasta que no se accione el sensor 5)
SET ANOUT[7] = 0 (fija una tensión de 0 volt en el eje 7)
MOVED DEMO[7] (lleva la pinza a la posición demo[7])
CLOSE (cierra la pinza)
MOVED DEMO[8] (lleva la pinza a la posición demo[8])
MOVE DEMO[10] (lleva la pinza a la posición demo[10])
MOVED DEMO[9] (lleva la pinza a la posición demo[9])
OPEN (abre la pinza)
MOVE DEMO[10] (lleva la pinza a la posición demo[8])
MOVE 0 (lleva la pinza a la posición 0)
END (fin del programa)
```

Posiciones:

	X	Y	Z	P	R
DEMO[1]	125	3623	918	-868	56
DEMO[2]	1755	3270	2260	-900	204
DEMO[3]	3334	2815	1010	-895	514
DEMO[4]	2786	279	3428	-895	825
DEMO[5]	3556	-2680	972	-894	1305
DEMO[6]	2127	-3835	2535	-894	1305
DEMO[7]	100	-3986	2239	-894	1792
DEMO[8]	76	-3032	4130	-874	1791
DEMO[9]	697	4319	944	-869	184
DEMO[10]	836	3758	2259	-850	48

Capitulo 4

SCORBASE para Windows



Descripción:

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

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

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

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

- ❑ **File:** el menú file contiene las usuales funciones de windows las cuales permiten cargar, grabar e imprimir archivos (los cuales contienen programas y posiciones). También se puede salir del software.
- ❑ **Edit:** el menú edit contiene las funciones usuales de windows para editar archivos que contengan programas de control de robots.
- ❑ **Run:** el menú run contiene los comandos de SCORBASE para efectuar el homing de los ejes del robot y periféricos y los comandos para la ejecución de programas
- ❑ **Options :** el menú options permite definir las preferencias a la hora de operar el software
- ❑ **Controller:** el menú controller permite configurar el controlador y ajustar la performance de las operaciones del controlador.
- ❑ **View:** las opciones de pantalla en el menú view permiten mostrar un conjunto predefinido de ventanas de dialogo y menús del SCORBASE.
- ❑ **Programs :** SCORBASE graba cada programa en ventanas separadas, esto permite editar programas mientras se usa otra ventana de SCORBASE.
- ❑ **Help:** este menú brinda ayuda para la operación del software.

Edición de programas:

Para editar un programa se necesitarán dos ventanas:

- Program window
- Command list.

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

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

Definición y grabado de posiciones:

Para grabar posiciones se utilizaran las siguientes ventanas:

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

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

Ejecución de programas:

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

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

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

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

PRECAUCIONES:

- DESCONECTE LOS MOTORES MEDIANTE EL SWITCH DEL CONTROLADOR ANTES DE ENTRAR EN EL AREA DE TRABAJO DEL ROBOT.
- NO SOBRECARGUE EL BRAZO DE ROBOT, EL PESO ENTRE LA PINZA Y LA CARGA DE TRABAJO NO DEBE EXCEDER LO RECOMENDADO (1KG. PARA EL VPLUS Y 2KG PARA EL IX)
- NO USE LA FUERZA FISICA PARA MOVER O DETENER ALGUNA PARTE DEL BRAZO ROBÓTICO
- NO CONDUZCA AL ROBOT A TRAVÉS DE OBSTACULOS.
- NO DEJE EL BRAZO EXTENDIDO POR MAS DE ALGUNOS SEGUNDOS.
- NO DEJE NINGUN EJE DEL ROBOT BAJO ESFUERZOS MECÁNICOS DURANTE MUCHO TIEMPO. ESPECIALMENTE LA PINZA.
- NO ENTRE AL ÁREA DE TRABAJO DEL ROBOT CUANDO ESTE ESTÁ OPERANDO.
- ASEGÚRESE DE NO TENER ROPA O CABELLO SUELTO CUANDO TRABAJA CON EL ROBOT.
- NO PONGA EL DEDO ENTRE LAS MORDAZAS DE LA PINZA.
- EL EQUIPO DEBERÁ OPERARSE SOLO EN PRESENCIA DE ALGUN RESPONSABLE
- SE RECOMIENDA CUBRIR EL ROBOT UNA VEZ FINALIZADA LA OPERACIÓN DEL MISMO
- TENGA EN CUENTA QUE EL SCORBOT ER-IX ES MAS ROBUSTO QUE EL VPLUS, POR LO TANTO ES MAS PELIGROSO

Conclusiones

Por lo visto en el capítulo 1, el SCORBOT ER Vplus es un elemento didáctico que permite una introducción a la robótica. Este dispositivo es de manejo similar a los utilizados en la industria, por lo que brinda la posibilidad de adquirir un aceptable entrenamiento en el manejo de brazos robóticos.

Como conclusión general encontramos en el SCORBOT ER Vplus una herramienta útil, tanto para el aprendizaje como para aplicaciones de laboratorio.

Por lo visto en el capítulo 2, el SCORBOT ER IX permite adquirir cierto manejo en la robótica. Este dispositivo es de manejo similar a los utilizados en la industria, por lo que brinda la posibilidad de adquirir un aceptable entrenamiento en el manejo de brazos robóticos. Se aconseja a los principiantes utilizar algún robot más pequeño de la misma familia antes que centrarse en el estudio del SCORBOT ER IX ya que debido a su robustez no es el más indicado para una introducción a la robótica.

En el capítulo 3 se listan las principales instrucciones del ACL el lenguaje de programación de ambos robots. Por otra parte, este lenguaje permite programar a diferentes robots de su familia por lo que, aprendiendo el manejo de este robot, se podrán manejar otros con diferentes características.

Como conclusión general encontramos en los SCORBOT ER Vplus y SCORBOT ER IX una herramienta útil, para el aprendizaje en un nivel intermedio y para aplicaciones de laboratorio.

El informe intentó ser breve y conciso, otorgando un enfoque genérico, apto para quien quiera utilizar el equipo en aplicaciones simples, para aplicaciones más complejas deberá hacerse un estudio más completo del equipo y del material disponible.

Referencias:

- SCORBOT-Vplus user's manual
- ATS for controller-A, reference guide
- ACL for controller-A, reference guide
- SCORBOT-ER IX user's manual
- CONTROLLER-B user's manual
- TEACH PENDANT for controller-B, user's manual
- ATS for controller-B, reference guide
- ACL for controller-B, reference guide