

Introducción a la Programación de Robots

CONTROL Y PROGRAMACIÓN DE ROBOTS

Grado en Electrónica Robótica y Mecatrónica



Índice

1. Introducción
2. Métodos de programación
3. Requerimientos de un sistema de programación
4. Ejemplo programación de robot:
 - El robot SCORBOT-V
 - El lenguaje ACL
 - Prácticas de laboratorio

Introducción

Objetivo: indicar la secuencia de acciones a realizar durante una tarea

- Movimiento a puntos predefinidos
- Manipulación de objetos
- Interacción/sincronización con el entorno

Partes del robot involucradas en la ejecución de un programa

- Memoria
- Sistema de control cinemático/dinámico
- E/S

Métodos de programación

Programación por guiado o aprendizaje

- Pasivo
 - directo
 - por maniquí
- Activo

Programación textual

- Nivel robot
- Nivel objeto
- Nivel tarea

Es frecuente la utilización conjunta de programación por guiado y textual

Métodos de programación

Programación por guiado o aprendizaje:

Consiste en hacer realizar al robot, o a una maqueta del mismo, la tarea (llevándolo manualmente por ejemplo) al tiempo que se registran las configuraciones adoptadas, para su posterior repetición de manera automática.

- **Pasivo:** El programador debe aportar directamente la energía para mover el robot. Se registran todas las posiciones.
 - directo
 - por maniquí
- **Activo:** Para mover el robot se utiliza el sistema de accionamiento del robot a través de una botonera o Joystick. Sólo se registran las posiciones que indique el operador.



Métodos de programación

Programación textual

- La tarea queda especificada mediante una serie de instrucciones textuales (Lenguaje formal)
- Similar a un programa informático
- El texto del programa es editado en un sistema informático, que puede ser independiente del robot, no precisando la presencia de éste durante la fase de desarrollo del programa (programación fuera de línea).
- Diferentes sintaxis, con mayor o menor grado de abstracción

Métodos de programación

Programación textual:

Nivel robot (movimientos a realizar por el robot)

- AS (KAWASAKI)
- KAREL (FANUC)
- KRL (KUKA)
- **RAPID (ABB)**
- V+ (ADEPT)
- VAL II (UNIMATION).

Nivel objeto (estado en que quedan los objetos)

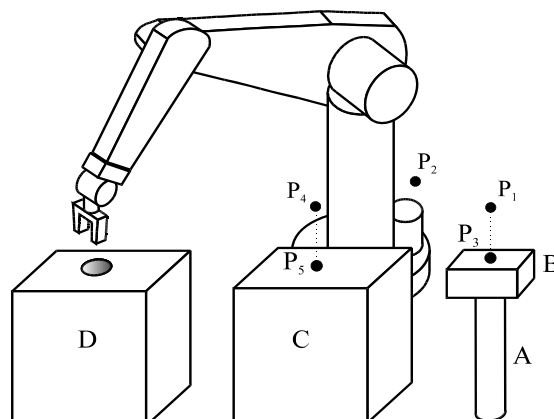
- LAMA (MIT - 1976).
- AUTOPASS (IBM - 1977).
- RAPT (Universidad de Edimburgo - 1978).

Nivel tarea (objetivo o sub-objetivo a conseguir)

Métodos de programación

Ejemplo de tarea a programar:

Se pretende situar la pieza A, sobre la que se apoya la pieza B, en el interior del orificio de la pieza D.

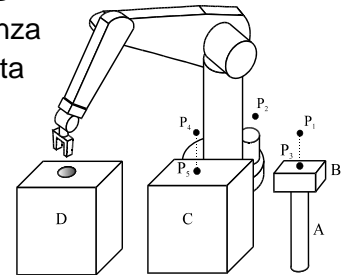


Métodos de programación

Programación a nivel de robot:

Operación de colocar B sobre C

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

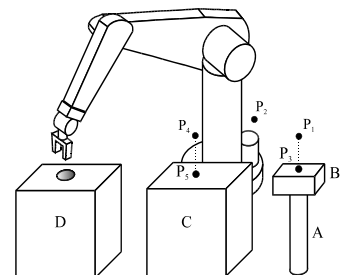


Métodos de programación

Programación a nivel de objeto:

Situar B sobre C haciendo coincidir
LADO_B1 con LADO_C1 y LADO_B2 con LADO_C2 ;

Situar A dentro D haciendo coincidir
EJE_A con EJE_HUECO_D y BASE_A con BASE_D ;

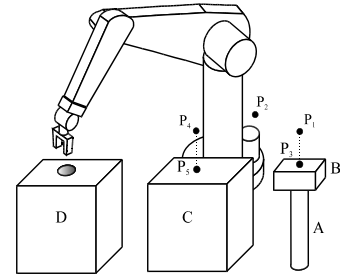


- Disminuye la complejidad del programa.
- La programación se realiza de manera más cómoda.
- Un planificador de la tarea se encargará de consultar una base de datos y generar las instrucciones a nivel de robot.

Métodos de programación

Programación a nivel de tarea:

Ensamblar A con D



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.

Requerimientos de un sistema de programación

- Entorno de desarrollo
- Modelado del entorno
- Tipos de datos
- Manejo de entradas/salidas (analógicas y/o digitales)
- Comunicaciones
- Control de movimiento
- Control del flujo de ejecución del programa
- ...

Requerimientos de un sistema de programación

Modelado del entorno

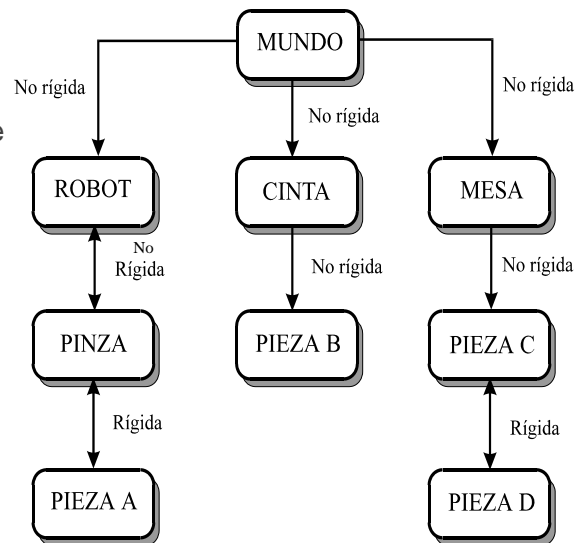
Modelo del entorno: Representación de los objetos con los que interacciona.

Localización definida mediante un sistema de referencia asociado al objeto.

Relaciones de unión entre objetos:

- Objetos **independientes**: el movimiento de uno no afecta al otro
- Objetos con **unión rígida**: el movimiento de uno implica el del otro y viceversa
- Objetos con **unión no rígida**: el movimiento de uno implica el del otro, pero no al revés.

Las relaciones pueden ser actualizadas por el programador (haciendo uso de composición de transformaciones) o automáticamente.



El robot SCORBOT-V



**Utilizado
para
prácticas**

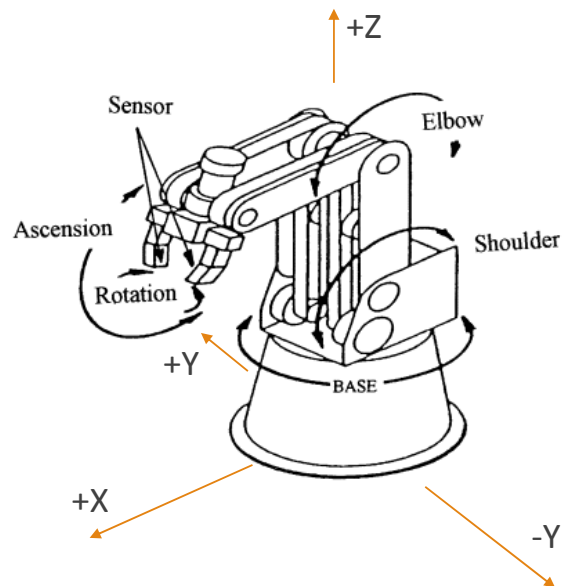


- **Robot de enseñanza**
 - Menor carga útil
 - Menor repetitividad y precisión
 - Peor software
- **Programación en ACL**
- **Componentes:**
 - El brazo del robot
 - El controlador
 - Botonera de enseñanza
 - Terminal de programación

El robot SCORBOT-V

Brazo del robot

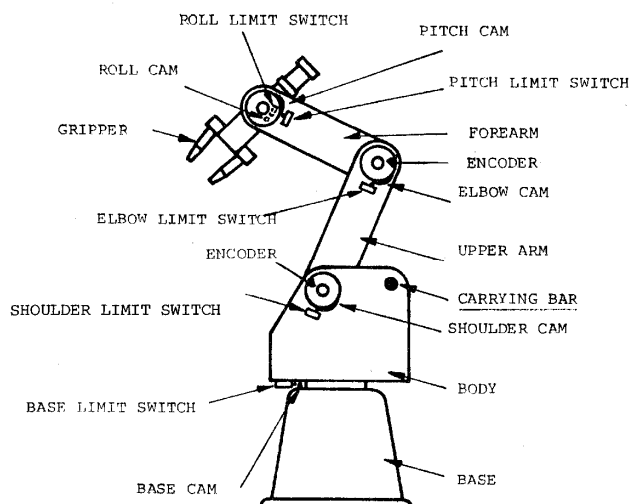
- Tipo RRR
- Grados de libertad:
 - Base
 - Hombro
 - Codo
 - Elevación
 - Giro
- Garra con movimiento casi paralelo



El robot SCORBOT-V

Brazo del robot

- **Servomotores de continua (24V, 2A) con reductora**
- **Codificadores de posición:**
 - Ópticos
 - Incrementales
 - Posición HOME por microinterruptores
- **Transmisiones indirectas**
 - Engranajes
 - Correas
- **Repetitividad: 0.5 mm**
- **Velocidad máxima: 600 mm/s**
- **Carga máxima: 1 Kg**
- **Peso: 11 Kg aproximadamente**



El robot SCORBOT-V

Controlador

- **Sistema microprocesador:**
 - Motorola 68010
 - 384K EPROM
 - 64K RAM de usuario
 - Multitarea en tiempo real
 - Sistema operativo propio (ordenador solo como terminal via RS-232)
- **Servocontroladores:**
 - 4 tarjetas para 8 motores
 - Fuente de alimentación interna (24V, 18^a) para potencia de los motores
 - Microinterruptores como protecciones térmicas



- **Señales entrada/salidas:**
 - 16 entradas
 - 4 salidas tipo relé (1-4)
 - 12 salidas de colector abierto (5-16)
- **Otros:**
 - Pulsador de emergencia
 - Interruptor de motores
 - Fuente de alimentación adicional (12V, 2A)
 - Indicadores led para entradas/salidas

El robot SCORBOT-V

Botonera de enseñanza

- **Mover brazo:**
 - Articulares
 - Cartesianas
- **Almacenar posicioneas**
- **Fijar velocidad del brazo**
- **Mover brazo a posición almacenada**
- **Ejecutar/abortar programas en ACL**
- **Activar/desactivar control de ejes**

Terminal de programación

- **Para programar robot desde una computadora**
- **ATS: Advanced Terminal Software**



El lenguaje ACL

ACL: Advanced Control Language

Lenguaje de programación del robot SCORBOT-V

- Ejecución directa de comandos
- Control de datos de entrada/salida
- Botonera de enseñanza
- Programación por parte del usuario
- Sincronización en la ejecución de programas
- Manejo simple de ficheros
- Conexión con el puerto serie (RS-232)
- Modo de ejecución de comandos
 - Directos: de ejecución inmediata
 - Indirectos o de edición: escritos en programas

El lenguaje ACL

Tipos de comandos:

- Definición y almacenamiento de posiciones
- Manipulación de posiciones
- Definición y manipulación de variables numéricas
- Control de movimientos del robot
- Tiempo real y control de programas
- Control del flujo de un programa
- Control de entradas y salidas
- Visualización e interfaz con el usuario
- Otros (interfaz con usuario, ...)

El lenguaje ACL

Definición y almacenamiento de posiciones:

- Posiciones conocidas por todos los programas
- En zona especial de memoria
- Formato del nombre:
 - ATS: cinco caracteres como máximo
 - Botonera de enseñanza: números
 - Posición '0': HOME

El lenguaje ACL

Definición y almacenamiento de posiciones:

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
DEFP <pos>	D, E	Reserva memoria para una posición
DIMP <vpos[n]>	D, E	Reserva memoria para un vector de n posiciones
HERE <pos>	D, E	Graba una posición absoluta en coordenadas articulares
HERE <vpos[2]>		
HERER <pos>	D	Graba una posición relativa a la posición actual del robot
HERER <pos2> <pos1>	D, E	Graba una posición <pos2> relativa a la posición <pos1>

El lenguaje ACL

Definición y almacenamiento de posiciones:

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
TEACH <pos>	D	Graba una posición absoluta en coordenadas cartesianas
TEACHR <pos>	D	Graba una posición relativa a la posición actual del robot
TEACHR <pos2> <pos1>	D	Graba una posición <pos2> relativa a la posición <pos1>
UNDEF <pos>	D	Borra las coordenadas de la posición pero sigue estando definida
DELP <pos>	D, E	Borra posición o vector de posiciones de la memoria RAM de usuario

El lenguaje ACL

Manipulación de posiciones:

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
SETP <pos2> = <pos1>	D, E	Copia la posición <pos1> en la posición <pos2>
SHIFT <pos> BY <eje> <valor>	D, E	Cambia el valor de un eje de una posición <pos> asignando el nuevo valor al eje especificado
SHIFTC <pos> BY <coord> <valor>	D, E	Igual, pero en en coord. cartesianas <coord> : X, Y, Z, P, R <valor> : en décimas de mm

El lenguaje ACL

Manipulación de posiciones:

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
SET <var>=PVAL <pos> <eje>	D, E	Asigna a la variable <var> el valor del <eje> de la posición <pos> Igual, pero con la coordenada <coord>
SET <var>=PVALC <pos> <coord>	D, E	
SETPV <pos> <eje> <val>	D, E	
SETPVC <pos> <coord> <val>	D, E	Graba <val> como coordenada para el <eje> de la posición <pos>
	D, E	Graba <val> como coordenada cartesiana <coord> de la posición <pos>

El lenguaje ACL

Definición de variables numéricas:

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
DEFINE <var1> ... <var8>	E	Crea variables “locales”
GLOBAL <var1> ... <var8>	D, E	Crea variables “globales”
DIM <var[n]>	E	Crea un vector de n variables “locales”
DIMG <var[n]>	D, E	Crea un vector de n variables “globales”
DELVAR <var>	D, E	Borra una variable de la RAM de usuario

El lenguaje ACL

Manipulación de variables numéricas:

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
SET <var1> = <var2>	D, E	Asigna el valor de <var2> a <var1>
SET <var1> = NOT <var2>		Asigna el valor negativo lógico de <var2> a <var1>
SET <var1> = COMPLEMENT <var2>		Asigna el valor complementario de <var2> a <var1>
SET <var1> = ABS <var2>		Asigna el valor absoluto de <var2> a <var1>
SET <var1> = <var2> <oper> <var3>		<oper> : +, - , * , / , SIN , COS, TAN, ATAN, EXP, LOG, MOD, OR, AND

El lenguaje ACL

Control de movimientos del robot:

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
MOVE <pos>	D, E	Mueve al robot a la posición <pos> a la velocidad especificada
MOVE <pos> {<tiempo>}	D, E	Mueve al robot a la posición <pos> en un tiempo <tiempo> en 10 ms/unidad
MOVED <pos> {<tiempo>}	E	Igual, salvo que el programa no continúa hasta que no se haya alcanzado la posición <pos>
MOVEL <pos> {<tiempo>}	D, E	Igual con movimiento lineal
MOVELD <pos> {<tiempo>}	E	
MOVEC <pos1> <pos2>	D, E	Movimiento circular del robot a <pos1> pasando por <pos2>
MOVECD <pos1> <pos2>	E	

El lenguaje ACL

Control de movimientos del robot:

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
MOVES <vpos> <inicio> <fin> {<tiempo>}	D, E	Mueve al robot por todas las posiciones del vector <vpos>, desde inicio hasta fin
MOVESD <vpos> <inicio> <fin> {<tiempo>}	E	
SPEED <valor>	D, E	Especifica la velocidad de movimiento (entre 1 y 100)
MPROFILE PARABOLE TRAPEZE	D, E	Ajusta el perfil de movimiento
INT_ON <eje1> ... INT_OFF <eje1> ...	D, E	Activa/desactiva el servocontrol integral

El lenguaje ACL

Control de movimientos del robot:

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
HOME {<n>}	D, E	Busca la posición de referencia
CON	D	Activa el servocontrol de los ejes
COFF	D	Desactiva el servocontrol de los ejes
TON	D	Activa (desactiva) las protecciones térmicas de los motores
TOFF	D	
OPEN	D, E	Abre la garra
CLOSE	D, E	Cierra la garra

El lenguaje ACL

Tiempo real y control de programas:

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
RUN <prog> {<prioridad>}	D, E	Ejecuta un programa
STOP {<prog>}	E	Aborta la ejecución de un programa
ABORT {<prog>}	D	Igual, y detiene el movimiento de los ejes
SUSPEND <prog>	D, E	Detiene la ejecución de un programa
CONTINUE <prog>	D, E	Continúa con la ejecución de un programa
SET <var>=TIME	D, E	Asigna a la variable var el valor de la variable de sistema TIME en ese instante
DELAY <tiempo>	E	Suspende el programa durante el tiempo especificado (en 10 ms)

El lenguaje ACL

Tiempo real y control de programas:

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
WAIT <var1> <cond> <var2>	E	Suspende la ejecución de un programa hasta que se cumpla la condición especificada.
PRIORITY <prog> <valor>	E	Especifica la prioridad de un programa (entre 1 y 10)
TRIGGER <prog> BY IN/OUT <n> <estado>	E	Ejecuta un programa condicionado al estado de la entrada/salida n (0: OFF 1:ON)

El lenguaje ACL

Control del flujo de programas:

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
IF <var1> <cond> <var2>	E	Ejecución condicional <cond>: < , > , = , <= , >= , <>
ANDIF <var1> <cond> <var2>	E	Condición lógica producto
ORIF <var1> <cond> <var2>	E	Condición lógica suma
ELSE	E	Sigue al IF y precede al ENDIF
ENDIF	E	Final de un bloque IF
FOR <var1> = <valor1> TO <valor2>	E	Bucle FOR
ENDFOR	E	Final de bucle FOR
LABEL <n>	E	Etiqueta (n entre 0 y 9999)
GOTO <n>	E	Salto incondicional
GOSUB <prog>	E	Salto a subrutina

El lenguaje ACL

Control de entradas y salidas:

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
SET OUT[<n>] = <estado>	D, E	Fija el estado de la salida n a 0 o a 1 <estado> : 0 -> OFF 1-> ON
IF IN[<n>] = <estado>	D, E	Ejecución condicionada al valor de la entrada n.

El lenguaje ACL

Visualización e interfaz con el usuario:

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
PRINT "cadena" <arg>	E	Presenta por pantalla un texto y el valor de una variable
PRINTLN "cadena" <arg>	E	Igual, pero introduce inicialmente un retorno de página
READ "cadena" <var>	D	Presenta por pantalla una cadena y espera la introducción del valor de la variable var por teclado
GET <var>	E	Espera que se introduzca un carácter por teclado y asigna a <var> el código ASCII del carácter

El lenguaje ACL

Otros:

- Manipulación de programas
- Funciones de edición
- ...

En el laboratorio