Ingeniería Informática

AUTOMATIZACIÓN Y ROBÓTICA

TEMA 15 (Teoría - Práctica)

Programación de robots

1

Programación de robots

- Robot industrial: manipulador multifuncional y programable.
- Objetivo: Indicar la secuencia de acciones a realizar durante la tarea.
  - Movimientos a puntos predefinidos.
  - Manipulación de objetos.
  - Interacción/sincronización con el entorno.
- Partes del robot involucradas:
  - Memoria.
  - Sistema de control cinemático/dinámico.
  - E/S.
- Manejo de localizaciones y sistemas de coordenadas.

2

Programación de robots

- Programación on-line y off-line
- Programación por aprendizaje o guiado
- Programación textual
- Características de los sistemas y lenguajes de programación

3

Ingeniería Informática

PROGRAMACIÓN ON-LINE Y OFF-LINE

4

Programación On-line

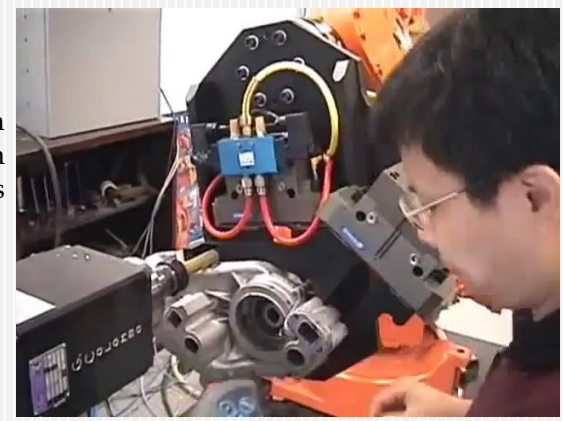
- Se ha de disponer del robot para desarrollar su programación.
- Ejemplo: Programación a partir de paletas de control como el flex-pendant de ABB.

Fuente: Youtube. Jim Kosmala de ABB Robotics USA

5

Programación On-line

- Ventajas:
  - Robot programado en concordancia con la posición actual del equipamiento y los objetos a manipular.
  - Buena precisión.



Fuente: Youtube

6

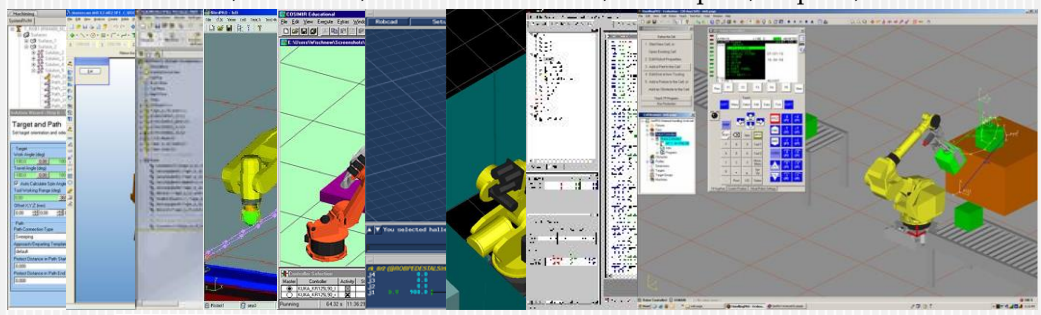
Programación On-line

- Inconvenientes:
  - El robot no se encuentra operativo durante la programación.
  - Coste equivalente al valor de producción.
  - Movimiento lento del robot durante la programación.
  - Difícil programación de la lógica y cálculos del programa.
  - Programa sin documentación.

7

Programación Off-line

- Utilización de entornos complejos de programación y simulación:
  - RobotStudio de ABB, KUKA Roboter, MELSOFT de Mitsubishi, FANUC Robotics' ROBOGUIDE, COSIMIR, Robcad de Siemens, WorkSpace, Ropsim, SimPRO…



8

Programación Off-line

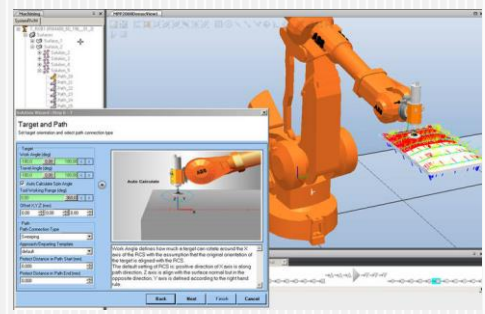
- Ventajas:
  - Programación efectiva de la lógica y cálculos necesarios para la tarea.
  - Simulador de célula robotizada y posibilidad de generar el programa directamente para el robot real.
  - Depuración de programas.
  - Reutilización de datos existentes de tipo CAD.
  - Coste independiente de la producción. El robot puede seguir trabajando mientras se programa.

Fuente: Youtube

9

Programación Off-line

- Inconvenientes:
  - Modelado 3D del entorno. Posibilidad de errores.
  - Errores al considerar la cinemática ideal.
  - Necesidad de personal experto en la programación de robots.



10

Programación híbrida

- Se almacenan las posiciones de manera on-line.
- Se crea el programa o procedimiento de manera off-line.
- El conocimiento preciso de las piezas a manipular al conocer el modelo CAD permite programar off-line las posiciones de manipulado relativas.
- Los tiempos de parada del robot se reducen.

11

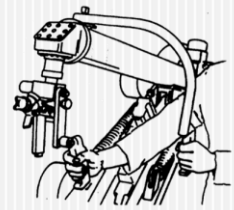
Ingeniería Informática

PROGRAMACIÓN POR APRENDIZAJE O GUIADO

12

Programación por aprendizaje

- Indicar al robot los movimientos a realizar para su posterior repetición.
  - Pasivo. Se realiza el movimiento manual del robot.
    - Directo. Se mueve el robot físicamente.
    - Uso de maquetas.



Guiado pasivo directo
(Robot Gaiotto)

Guiado pasivo con maniquí
(Robot Nordson)

13

Programación por aprendizaje

- Indicar al robot los movimientos a realizar para su posterior repetición.
  - Pasivo. Se realiza el movimiento manual del robot.
    - **Directo**. Se mueve el robot físicamente.
    - Uso de maquetas.

Fuente: Youtube

14

Programación por aprendizaje

- Indicar al robot los movimientos a realizar para su posterior repetición.
  - Pasivo. Se realiza el movimiento manual del robot.
    - **Directo**. Se mueve el robot físicamente.
    - Uso de maquetas.

Fuente: Youtube

15

Programación por aprendizaje

- Indicar al robot los movimientos a realizar para su posterior repetición.
  - Pasivo. Se realiza el movimiento manual del robot.
    - Directo. Se mueve el robot físicamente.
    - **Uso de maquetas**.

Fuente: Youtube

16

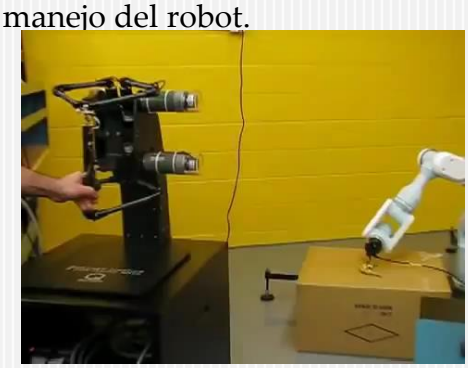
Programación por aprendizaje

- Indicar al robot los movimientos a realizar para su posterior repetición.
  - Activo. Se emplean dispositivos para el manejo del robot.
    - Joystick.
    - Dispositivo de enseñanza.

17

Programación por aprendizaje

- Indicar al robot los movimientos a realizar para su posterior repetición.
  - Activo. Se emplean dispositivos para el manejo del robot.
    - Joystick.
    - Dispositivo de enseñanza.



Fuente: Youtube

18

Programación por aprendizaje

- Tipos de aprendizaje.
  - Aprendizaje básico. Se indican únicamente los puntos por los que ha de pasar el robot.
  - Aprendizaje extendido. Se pueden indicar velocidades, tipo de trayectoria, precisión, control de flujo.
- Ventajas:
  - No precisa disponer de las coordenadas de los elementos del entorno de trabajo.
  - No se producen errores de posicionamiento por incorrecta calibración del robot o su entorno.
  - Fáciles de aprender.
- Inconvenientes:
  - Se ha de disponer del robot para realizar la programación.
  - Son limitados: fundamentalmente instrucciones de movimiento. No son reactivos.
  - Difícil depuración e inexistencia de documentación.

19

Ingeniería Informática

PROGRAMACIÓN TEXTUAL

20

Programación textual

- Se indica la tarea mediante un lenguaje de programación.
- Un programa se corresponde con una serie de órdenes que se pueden editar y posteriormente ejecutar.
- Existe un texto o código alto nivel (que se traducirá al código máquina de la CPU).
- No precisa la presencia del robot durante la fase de desarrollo del programa.

21

Programación textual

- Intentos de estandarización.
  - IRDATA (Univ. Karlsruhe).
    - Código intermedio entre el sistema de programación utilizado y el propio sistema de programación del robot.
  - IRL (Industrial Robot Language).
    - Notación independiente de cualquier robot
    - Puede ser traducido a un código intermedio para el control de un robot.
    - Implementado en Cosimir.

22

Ejemplo IRL I

```
PROGRAM ejemplo IRL
VAR
      INPUT BOOL: pieza at 1;        {entrada digital que indica la presencia de pieza}
                  repetir at 2;      {entrada digital que indica el deseo de repetir la acción}
      OUTPUT BOOL:electrm at 1;      {salida digital que activa el electroimán del efector final}
BEGIN
      IF NOT pieza THEN HALT;ENDIF;
      R_ACC:=300.0;                  {especifica la aceleración}
      R_SPEED:=100.0;                {especifica la velocidad}
```

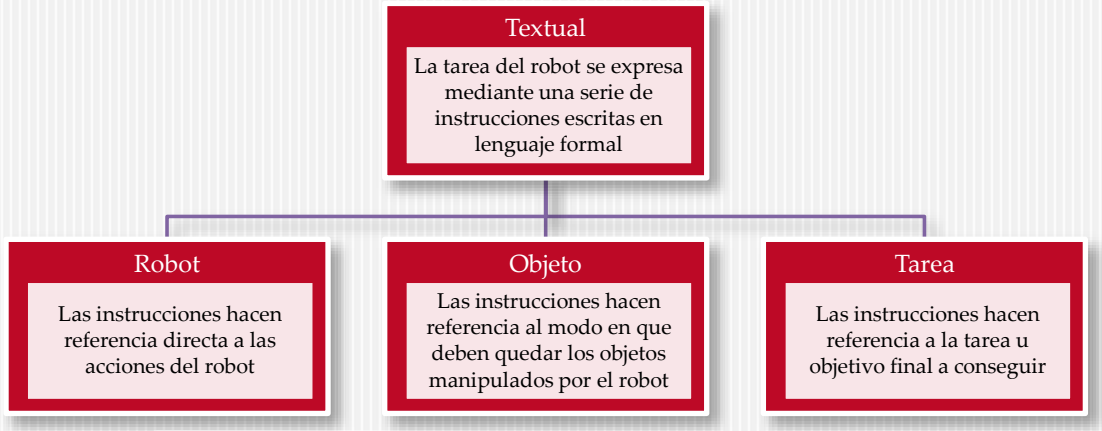
23

Ejemplo IRL II

```
Bucle:
      MOVE LIN p_sb_pieza;           {movimiento en línea recta a punto sobre pieza}
      MOVE LIN p_pieza;              {movimiento en línea recta a punto recogida pieza}
      electrm:=TRUE;                 {activa el electroimán del efector para coger la pieza}
      MOVE LIN p_sb_pieza;           {movimiento en línea recta a punto sobre pieza}
      MOVE LIN p_intermedio;         {movimiento en línea recta a punto intermedio}
      MOVE LIN p_destino;            {movimiento en línea recta hasta situarse sobre destino}
      electrm:=FALSE;                {desactiva el electroimán del efector para dejar la pieza}
      MOVE LIN p_intermedio;         {regresa al punto intermedio}
      IF repetir = TRUE THEN GOTO bucle;ENDIF;          {¿Repetir → coger nueva pieza?}
      MOVE LIN home;                 {Fin de tarea. Se retorna a la posición de reposo}
      HALT;                          {Para la ejecución del programa}
END PROGRAM
```

24

Programación textual



25

Lenguajes orientados a robot

- Las instrucciones hacen referencia a las acciones del robot.
- Ventajas (respecto a programación por aprendizaje o guiado):
  - Estructuras de programación similares a las de los computadores (iterativas, condicionales, subrutinas).
  - Acceso a E/S con comandos (interacción entorno).
  - Sincronización fácil (órdenes "Espera", "Mientras Entrada no activada...", etc.).
- Inconvenientes: más complejidad.
  - Necesita computador y programador.
  - Los comandos suelen depender del robot y lenguaje concretos.
  - Número elevado de instrucciones.
- Ejemplos:
  - AS (KAWASAKI), KAREL (FANUC), KRL (KUKA), RAPID (ABB), V+ (ADEPT), VALII (UNIMATION).

26

Lenguajes orientados a objeto

- Las instrucciones hacen referencia al modo en que deben quedar los objetos manipulados por el robot.
- 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.
- Ejemplos:
  - LAMA.
  - AUTOPASS.
  - RAPT.

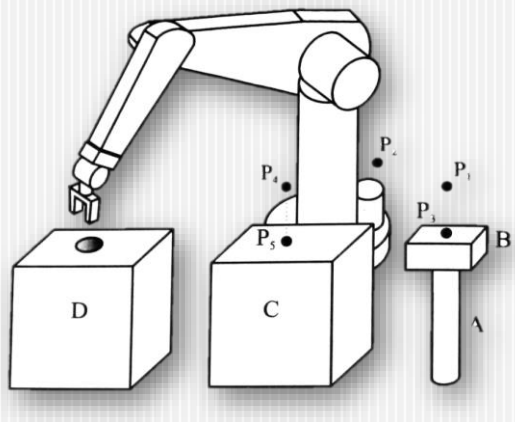
27

Lenguajes orientados a tarea

- Se indica qué se desea realizar.
- Utilización de inteligencia artificial.
- Modelo del entorno (CAD).
- Uso de simuladores.
- Sistema sensorial.
- En investigación.
- Inconveniente: complejidad.
- Ventajas:
  - El programador piensa en la cadena de montaje; el robot es una simple herramienta cuyas características exactas no tiene por qué conocer.
  - El chequeo de situaciones imprevistas las hace el compilador o intérprete.

28

Ejemplo niveles programación

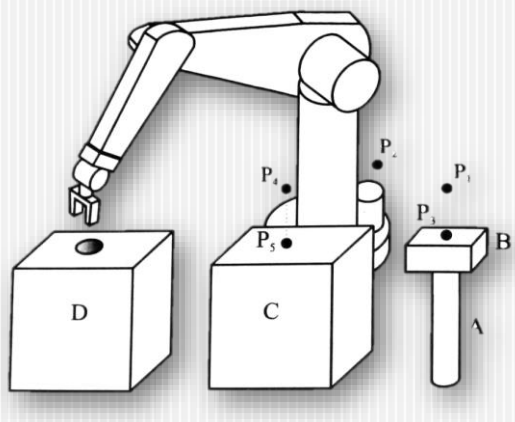


- Lenguaje orientado a robot:

```
Mover_a P1 via P2
Vel = 0.2 * VELMAX
Pinza=ABRIR
Prec=ALTA
Mover_recta_a P3
Pinza=CERRAR
Espera=0.5
Mover_recta_a P1
Prec=MEDIA
Vel=VELMAX
Mover_a P4 via P2
Prec=ALTA
Vel=0.2*VELMAX
Mover_recta_a P5
Pinza=ABRIR
```

29

Ejemplo niveles programación



- Lenguaje orientado a objeto:

```
Situar B sobre C
   haciendo coincidir
LADO_B1 con LADO_C1
Situar A dentro D
   haciendo coincidir
EJE_A con EJE_HUECO_D y
   BASE_A con BASE_D
```

- Lenguaje orientado a tarea:

```
Ensamblar A con D
```

30

Ingeniería Informática

CARACTERÍSTICAS DE LOS SISTEMAS Y LENGUAJES DE PROGRAMACIÓN

31

Características lenguajes programación

- Entorno de programación.
- Modelado del entorno.
- Tipos de datos.
- Manejo de entradas/salidas.
- Comunicaciones.
- Control de movimiento.
- Control de flujo de ejecución del programa.

32

Entorno de programación

- Aumenta la productividad de la programación.
- Depuración de programas. Ejecución paso a paso.
- Monitorización del desarrollo del programa.
- Otras opciones:
  - Programación fuera de línea.
  - Simulador de célula robotizada y posibilidad de generar el programa directamente para el robot real.

33

Modelado del entorno

- Representación geométrica de los objetos con los que interacciona el robot (posición, orientación, forma, peso, dimensiones…)
- Definición de sistemas de coordenadas asociados a cada objeto.
- Definición de relaciones entre objetos:
  - Independientes. El movimiento de un objeto no afecta al otro.
  - Unión rígida. El movimiento de uno implica el del otro y viceversa
  - Unión no rígida. El movimiento de uno implica el del otro pero no al revés.

34

Tipos de datos

- Tipos de datos convencionales (booleanos, enteros, reales…).
- Definición de posiciones y orientaciones.
  - Coordenadas articulares.
  - Coordenadas cartesianas.
    - Posición. Coordenadas cartesianas del origen del sistema
    - Orientación:
      - Ángulos de Euler.
      - Cuaternios.
      - Matriz de rotación.

35

Manejo de entradas-salidas

- Comunicación del robot con otras máquinas o procesos.
- Instrucciones para leer y activar entradas y salidas digitales.
- Ejecución de interrupciones.
- Monitorización del robot a través de conexión serie o LAN.
- Integración de sensores.
  - Modificar la trayectoria.
  - Elegir entre diversas alternativas.
  - Obtener la identidad y posición de objetos y sus características.
  - Cumplir con restricciones externas.

36

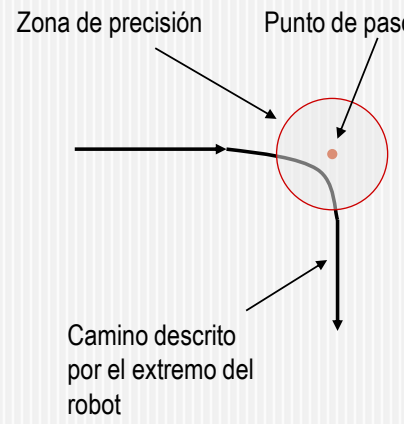
Comunicaciones

- El robot debe poder comunicarse con otros robots o máquinas que participan en la producción con el objetivo de:
  - Adaptar los movimientos del robot a situaciones cambiantes.
  - Actualización del programa en función de nuevas órdenes de producción.
  - Monitorización o supervisión del estado de la célula.
  - Control del estado de la producción.
- Comunicación mediante RS232, Devicenet, CanBus, Profibus, Ethernet, etc.

37

Control del movimiento del robot

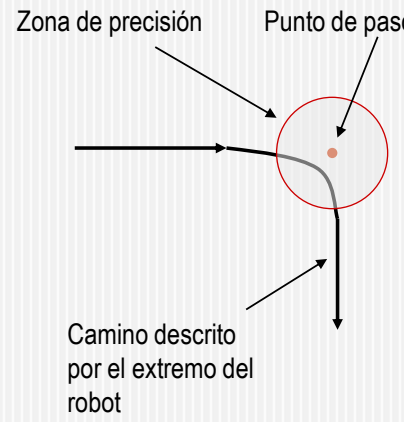
- Especificación del movimiento del robot.
- Tipos de trayectorias.
  - Punto a punto (eje a eje, simultaneo ejes)
  - Coordinadas.
  - Trayectoria continua.
- Velocidad.
- Precisión.



38

Control del movimiento del robot

- Especificación de puntos de paso.
  - Precisión baja. El robot se encamina hacia la siguiente configuración sin haber llegado a la anterior. Poca modificación de velocidad.
  - Precisión alta. Menor continuidad.



39

Control del flujo de ejecución

- Ejecución de bucles.
- Sentencias condicionales.
- Capacidad de procesamiento paralelo.
- Sincronización.
- Gestión de interrupciones (prioridades, activar/desactivar interrupciones…)

40


Ingeniería Informática
AUTOMATIZACIÓN Y ROBÓTICA
TEMA 15 (Teoría - Práctica)
Programación de robots

41