

Programación

Fundamentos de Robótica Industrial



Programación

- **Un robot industrial** es básicamente un manipulador multifuncional reprogramable (esto es lo que permite su uso en diversas aplicaciones, y su flexibilidad y adaptabilidad)
- **La programación de un robot** se puede definir como el proceso mediante el cual se le indica a éste la secuencia de acciones que deberá llevar a cabo durante la realización de su tarea
- **La programación se convierte en la interfaz** mediante la cual un usuario puede aprovechar todos los algoritmos de mecánica y de control subyacentes que hemos estudiado
- Hoy estudiaremos los procedimientos de programación más habituales

La fábrica

- Los manipuladores son una parte de un **proceso automatizado**.
- El término **célula de trabajo** se utiliza para describir un grupo de equipos, que puede incluir uno o más manipuladores, sistemas de transportadores, alimentadores de piezas y dispositivos fijos.
- En el siguiente nivel más alto, las células de trabajo pueden **interconectarse en redes** distribuidas en toda la fábrica, de manera que una computadora central pueda controlar todo el flujo.
- La programación de manipuladores se considera comúnmente dentro del problema más amplio de programar una variedad de máquinas interconectadas en una célula de trabajo de fábrica automatizada.

Clasificación de métodos de programación

- Se han intentado **unificar** los procedimientos de programación sin éxito (por ej: ROS)
- **Programación por guiado** o aprendizaje (uso del teach pendant)
- **Programación textual**
- **Uso mixto**

Programación por guiado

- **La 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.
- **Guiado Pasivo:** el programador aporta directamente la energía para mover el robot
 - Guiado pasivo directo: se toma el extremo del robot y se lleva hasta los puntos deseados siguiendo las trayectorias más adecuadas. La unidad de control del robot registrará, de manera automática, la señal de los sensores de posición de las articulaciones en todos los puntos recorridos.
 - La dificultad física de mover toda la estructura del robot se resuelve con la utilización del guiado pasivo por maqueta.
 - En ambos casos, el registro de las configuraciones que adopta el robot es continuo, es decir, la unidad de control muestrea y **graba a una frecuencia elevada**, las posiciones de las articulaciones y la trayectoria seguida por el robot y su velocidad queda especificada por este registro.



Programación por guiado

- **Guiado Activo:** los accionamientos son los encargados de mover las articulaciones
- Lo habitual es que la unidad de control únicamente registre aquellas configuraciones del robot que el programador indique expresamente.
- El movimiento que ha llevado el robot hasta alcanzar la configuración final es irrelevante, siendo necesario incorporar los datos que definen la trayectoria del robot desde la configuración anterior hasta la nueva (habitualmente tipo de trayectoria, velocidad, precisión). Adicionalmente a través del panel de programación se pueden incluir instrucciones para el control del flujo del programa (saltos, repetición de movimientos), atención a entradas/salidas binarias, etc.



Programación por guiado

Ventajas:

- Muy fáciles de usar
- Muy versátiles (pintura, soldadura, etc)
- No precisan disponer de las coordenadas de los elementos relevantes del entorno de trabajo
- No se producen errores de posicionamiento por una incorrecta calibración del robot o su entorno, y pueden resultar fáciles de aprender.
- Dado que el programador no hace uso directo de las coordenadas de los objetos del entorno, la unidad de control no precisa de las funciones del modelado y control cinemático, desapareciendo, consecuentemente, los posibles problemas asociados a éste como la existencia de soluciones múltiples o los puntos singulares

Desventajas:

- Necesidad de utilizar al propio robot y su entorno para realizar la programación, obligando a sacar al robot de la línea de producción interrumpiendo el proceso..
- Inexistencia de una documentación del programa y dificultad de realizar modificaciones en el mismo, inconvenientes ambos que conducen a una difícil depuración y puesta a punto de las aplicaciones.

Programación textual

- Permite indicar la tarea al robot mediante el uso de un **lenguaje de programación** específico
- El texto del programa es editado en un sistema informático que puede ser **independiente del robot**, por este motivo también se conoce a este método de **programación como fuera de línea**.
- La programación textual puede ser clasificada en tres niveles: **robot, objeto y tarea**
 - **Robot:** Refiere directamente a los movimientos de éste:

Mover_a P1 via P2 ; Situarse en un punto sobre la pieza B

Vel = 0.2 * VELMAX ; Reducir la velocidad

Pinza = ABRIR ; Abrir la pinza

Prec = ALTA ; Aumentar la precisión

Programación textual

- **Objeto:** Refiere a los estados en que deben quedar los objetos:

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 ;

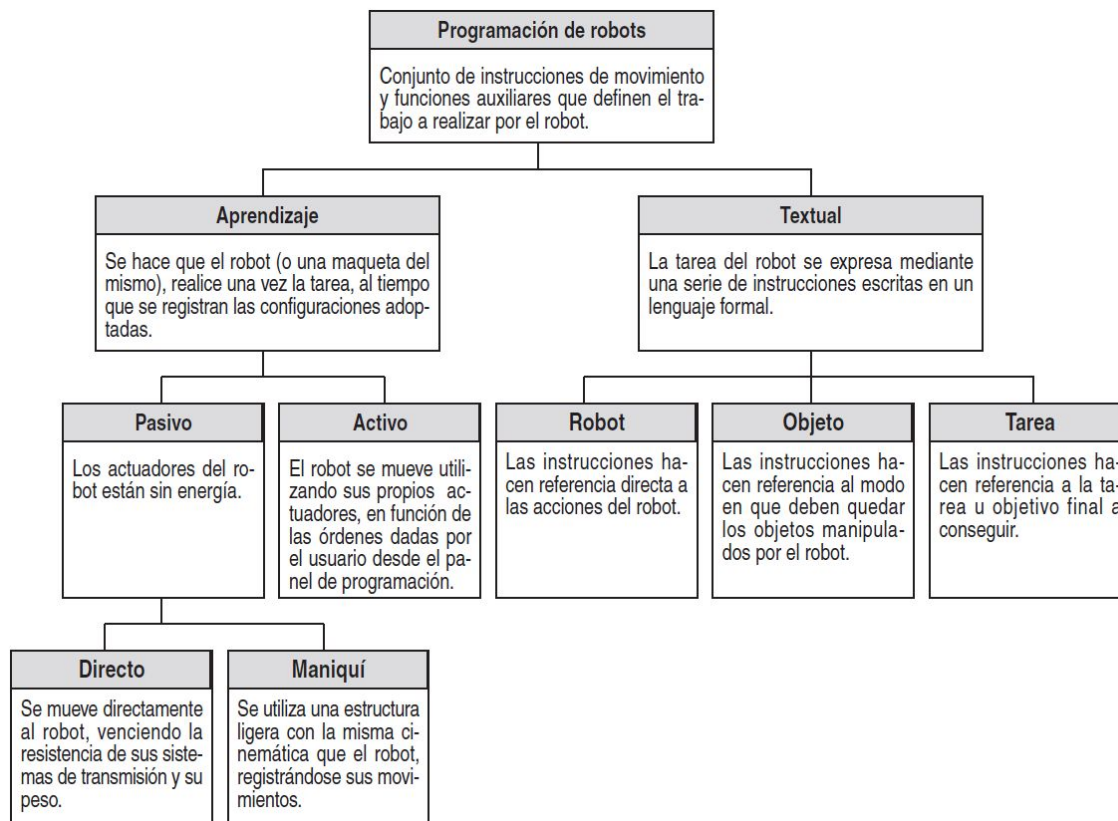
Programación textual

- **Tarea:** A un objetivo (o subobjetivos) que debe cumplir:

Ensamblar A con D

- En la práctica los lenguajes son del tipo robot

Resumen



Requerimientos de un sistema de programación

- Entorno de programación.
- Modelado del entorno.
- Tipo de datos.
- Manejo de entradas/salidas (digital y analógica).
- Comunicaciones.
- Control de movimiento.
- Control del flujo de ejecución del programa.

Entorno de Programación

- Sistemas de programación de tipo **interpretado**, pudiéndose realizar un seguimiento paso a paso de lo programado en cada momento.
- Disponibilidad de herramientas para la **programación fuera de línea**
- Uso de simuladores de la célula robotizada con posibilidad de generar el programa directamente para el robot real

Modelado del entorno

El modelo del entorno es la **representación que tiene el robot de los objetos con los que interacciona.**

Normalmente, este modelo se limita a características geométricas: **posición y orientación de los objetos**, y en ocasiones a su forma, dimensiones, peso, etc.

Para definir la posición y orientación de los objetos del modelo, lo más frecuente es asignar a cada objeto de manera **solidaria un sistema de referencia**, de manera que la posición y orientación de este sistema referidos a un sistema base, normalmente denominado sistema del mundo, definen de manera única las del objeto.

Algunos modelos del entorno permiten establecer relaciones entre objetos las cuales establecen la posible unión física entre ellos

Tipo de datos

- Datos convencionales: **enteros, reales, booleanos**, etc.
- Específicamente destinados a **definir la posición y orientación** de los puntos y objetos a los que debe acceder el robot:
 - Utilizando coordenadas **articulares** o del robot: mediante una **n-upla ($q_1 \dots q_n$)**
 - Utilizando coordenadas **cartesianas**: **px, py, pz** para posición y orientación mediante diferentes tipos de representación:
 - Ángulos de Euler: (φ, θ, ψ) .
 - Cuaternios: $(\cos\theta/2, h_x \sin\theta/2, h_y \sin\theta/2, h_z \sin\theta/2)$.
 - Matriz n,s,a (*Normal, Slide, Approach*)
- En la mayoría de los sistemas se utilizan varios métodos de representación

Tipo de datos

- **VAL II** (Stäubli) emplea $(q_1 \dots q_6)$, $(p_x, p_y, p_z, \varphi, \theta, \psi)$
- **AML** (IBM) emplea $(p_x, p_y, p_z, \varphi, \theta, \psi)$
- **ARLA** (ABB) emplea $(p_x, p_y, p_z, \cos\theta/2, h_x \text{ sen}\theta/2, h_y \text{ sen}\theta/2, h_z \text{ sen}\theta/2)$
- **RAPID** (ABB) emplea $(p_x, p_y, p_z, \cos\theta/2, h_x \text{ sen}\theta/2, h_y \text{ sen}\theta/2, h_z \text{ sen}\theta/2)$
- **V** (Adept) emplea $(p_x, p_y, p_z, \varphi, \theta, \psi)$
- **AL** (Stanford) emplea $(q_1 \dots q_6)$, y matrices de transformación homogénea

En **VAL II**, **V** y **RAPID** también existe la posibilidad de emplear matrices de transformación homogénea.

Manejo de entradas/salidas (digital y analógica)

- La comunicación del robot con otras máquinas o procesos que cooperan con él, es fundamental para conseguir su integración y sincronización en los procesos de fabricación. Esta comunicación se consigue, en el nivel más sencillo, mediante **señales binarias de entrada y salida**.
- Casos particulares de uso:
 - **Interrupciones**
 - **Integración de sensores**

Comunicaciones

El robot, debe poder comunicarse tanto con otros robots o máquinas equivalentes que participan de la producción, como con los sistemas informáticos responsables de la planificación y gestión de la producción. Esta comunicación permite entre otras cosas:

- La adaptación de los movimientos del robot a situaciones cambiantes en la línea de producción
- La carga o actualización del programa del robot en función de nuevas órdenes de producción
- La monitorización o supervisión del estado de la célula
- El control del estado de la producción

Control de Movimiento

Capaz de especificar:

- punto de destino
- tipo de trayectoria espacial que debe ser realizada
- velocidad media del recorrido
- precisión con que se debe alcanzar el punto destino

Lo habitual es que el robot admita **3 tipos de movimientos**: movimiento **libre** (equivalente al coordinado, usando algún tipo de interpolación articular), movimiento en **línea recta** y movimiento **circular**.

Control de Movimiento

Velocidad:

Habitualmente se define la velocidad como una fracción de una velocidad máxima, de forma que al modificar esta variable se ajusta en todo el programa

Precisión:

La precisión interactúa inversamente con la velocidad, por esto, usualmente podemos definir waypoints los cuales no necesariamente deben ser alcanzados con alta precisión para aumentar la velocidad del ciclo.

Control de flujo de ejecución de programa

Posee las estructuras habituales de bucles (for, repeat, while, etc.).

Tienen capacidad de procesamiento en paralelo, tanto para el control de varios robots trabajando conjuntamente bajo las órdenes de un solo programa, como para el control de un único robot en una celda de trabajo cuyos equipos se encuentran igualmente bajo el control del programa del robot (señales de sincronismo basadas en semáforos y ejecución de tareas en paralelo)

Control de interrupciones, se fija la prioridad en el tratamiento de las mismas, así como activarlas y desactivarlas durante la ejecución de diversas fases del programa.

Estandarización

- No ha tenido éxito
- Lenguajes específicos de cada marca
- ROS (mas bien un ambiente de programación)

FIN!

