MANUAL DE ESTUDIO · SISTEMAS ROBÓTICOS

Configuraciones del brazo y aplicaciones industriales

Capítulo 02 · En el video: **minuto 15:49** · Duración 15:17

Objetivo. Elegir la configuración de un robot según su espacio de trabajo y comprobar que su ciclo cabe en el takt de la planta.

Ideas clave

Cartesiano y pórtico (PPP)

Tres ejes prismáticos perpendiculares: el espacio de trabajo es un prisma, la rigidez es uniforme y la programación es directa. Los pórticos (gantry) cubren áreas grandes sobre la línea.

Cilíndrico (RPP) y polar (RRP)

Una base rotacional con brazo que sube y se extiende genera un cilindro hueco; el polar cambia la columna por un giro de elevación y su envolvente es un casquete esférico.

SCARA

Dos brazos rotacionales de ejes verticales y un eje Z: es rígido en vertical y flexible en el plano horizontal, ideal para insertar piezas con pequeños desalineamientos.

Articulado (RRR) y delta

El brazo antropomórfico tiene el mayor alcance para su tamaño y llega a orientaciones difíciles. El delta mueve una plataforma liviana con tres brazos paralelos y supera las 100 recogidas por minuto.

Aplicaciones

Soldadura por arco (trayectoria continua y velocidad constante) y por puntos (pinza servoneumática); pintura (muñeca hueca y protección para atmósferas explosivas); paletizado con garras de vacío Venturi y sincronía con la cinta por encoder; desbarbado, que exige rigidez frente a las fuerzas de corte.

Tiempo de ciclo y takt

El takt es el ritmo que pide la planta: tiempo disponible dividido por la demanda. El ciclo del robot (aproximar, procesar, retirar y trasladar) debe quedar por debajo, con margen para paradas.



Desglose del ciclo de configuraciones_y_takt.py.

Fórmulas y relaciones

Relación	Expresión	Cómo leerla
Volumen cartesiano	$V = x \cdot y \cdot z$	Prisma recto.
Volumen cilíndrico	$V = \pi(r_2^{\,2} - r_1^{\,2}) \cdot h \cdot (\varphi / 360°)$	φ: giro de la base.
Volumen esférico	$V = (r_2^{\,3} - r_1^{\,3})/3 \cdot (\operatorname{sen} e_{máx} - \operatorname{sen} e_{mín}) \cdot \varphi$	φ en radianes; e: elevación.
Takt	$T_{takt} = \text{tiempo disponible} / \text{demanda}$	Ritmo de la línea.
Intercepción en cinta	$t = (x_{robot} - x_{pieza}) / v_{cinta}$	Cuándo llega la pieza.

Ejemplo resuelto: una celda de soldadura frente al takt

1. Turno de 8 h con 30 min de pausas: 27 000 s disponibles; demanda 1800 piezas.
2. $T_{takt} = 27\,000/1800 = 15{,}0$ s.
3. Ciclo: 1,2 + 9,5 + 1,0 + 2,1 = 13,8 s.
4. Ocupación: 13,8/15 = 92 %.

La celda cumple, pero con 1,2 s de margen por pieza. configuraciones_y_takt.py también calcula volúmenes de trabajo e intercepción en cinta.

Selección de la configuración

Configuración	Espacio de trabajo	Fuerte en	Uso típico
Cartesiano PPP	Prisma	Rigidez, áreas grandes	Pórticos, CNC, dispensado
Cilíndrico RPP	Cilindro hueco	Carga y descarga radial	Máquinas herramienta
Polar RRP	Casquete esférico	Alcance	Fundición, forja
SCARA	Cilindro	Velocidad en el plano	Ensamble electrónico
Articulado RRR	Esfera irregular	Alcance y orientación	Soldadura, pintura, manipulación
Delta	Cúpula pequeña	Velocidad muy alta	Envasado, alimentos

Código del capítulo

Los archivos completos vienen en el paquete ZIP del capítulo (carpeta `codigo/`).

configuraciones_y_takt.py — volúmenes de trabajo, takt, ocupación e intercepción de una pieza en la cinta.

```
"""
configuraciones_y_takt.py — Capítulo 02: volumen de trabajo, tiempo de ciclo e intercepción en cinta.
"""
import math


def vol_cartesiano(x, y, z):
    return x * y * z


def vol_cilindrico(r_min, r_max, h, giro_grados=360):
```

```
    return math.pi * (r_max ** 2 - r_min ** 2) * h * giro_grados / 360


def vol_esferico(r_min, r_max, giro_grados=360, elev_min=-30, elev_max=60):
    """Casquete esférico entre dos radios y dos ángulos de elevación."""
    franja = math.sin(math.radians(elev_max)) - math.sin(math.radians(elev_min))
    return (2 * math.pi / 3) * (r_max ** 3 - r_min ** 3) * franja * giro_grados / 360


def takt(tiempo_disponible_s, demanda):
    return tiempo_disponible_s / demanda


def intercepcion(x_pieza, v_cinta, x_robot, t_robot):
    """Instante en que la pieza llega a x_robot y si el robot alcanza a estar listo."""
    t_llegada = (x_robot - x_pieza) / v_cinta
    return t_llegada, t_llegada >= t_robot


if __name__ == '__main__':
    print('Volumen de trabajo aproximado (m³)')
    print(f'  Cartesiano 1,2 × 0,8 × 0,5 m          : {vol_cartesiano(1.2, 0.8, 0.5):.3f}')
    print(f'  Cilíndrico r 0,3–0,9 m, h 0,6 m, 300° : {vol_cilindrico(0.3, 0.9, 0.6, 300):.3f}')
    print(f'  Esférico r 0,4–1,2 m, 340°, −30/+60°  : {vol_esferico(0.4, 1.2, 340):.3f}')

    disponible = 8 * 3600 - 30 * 60          # turno menos pausas
    t_takt = takt(disponible, 1800)
    etapas = {'aproximación': 1.2, 'proceso': 9.5, 'retirada': 1.0, 'traslado': 2.1}
    ciclo = sum(etapas.values())
    print(f'\nTakt = {disponible} s / 1800 piezas = {t_takt:.1f} s')
    print(f'Ciclo del robot = {ciclo:.1f} s -> ocupación {100 * ciclo / t_takt:.0f} %'
          f' {"(cumple)" if ciclo <= t_takt else "(NO cumple)"}')

    t, ok = intercepcion(x_pieza=0.0, v_cinta=0.25, x_robot=0.80, t_robot=2.6)
    print(f'\nPieza a 0,25 m/s llega a la zona del robot en {t:.2f} s;'
          f' el robot necesita 2,6 s -> {"alcanza" if ok else "no alcanza"}')
```

Errores frecuentes

- Calcular el takt con el turno completo sin descontar pausas.
- Elegir un SCARA para tareas que necesitan inclinar la herramienta.
- Olvidar el tiempo de aproximación y retirada en el ciclo.

Autoevaluación

1. ¿Qué configuración tiene rigidez vertical alta y conformidad horizontal?
2. Si el takt es 15 s y el ciclo 13,8 s, ¿qué ocupación tiene el robot?
3. ¿Por qué los robots de pintura usan muñeca hueca?

Respuestas. 1. SCARA. **2.** 92 %. **3.** Para llevar las mangueras por dentro y evitar enredos y chispas.