

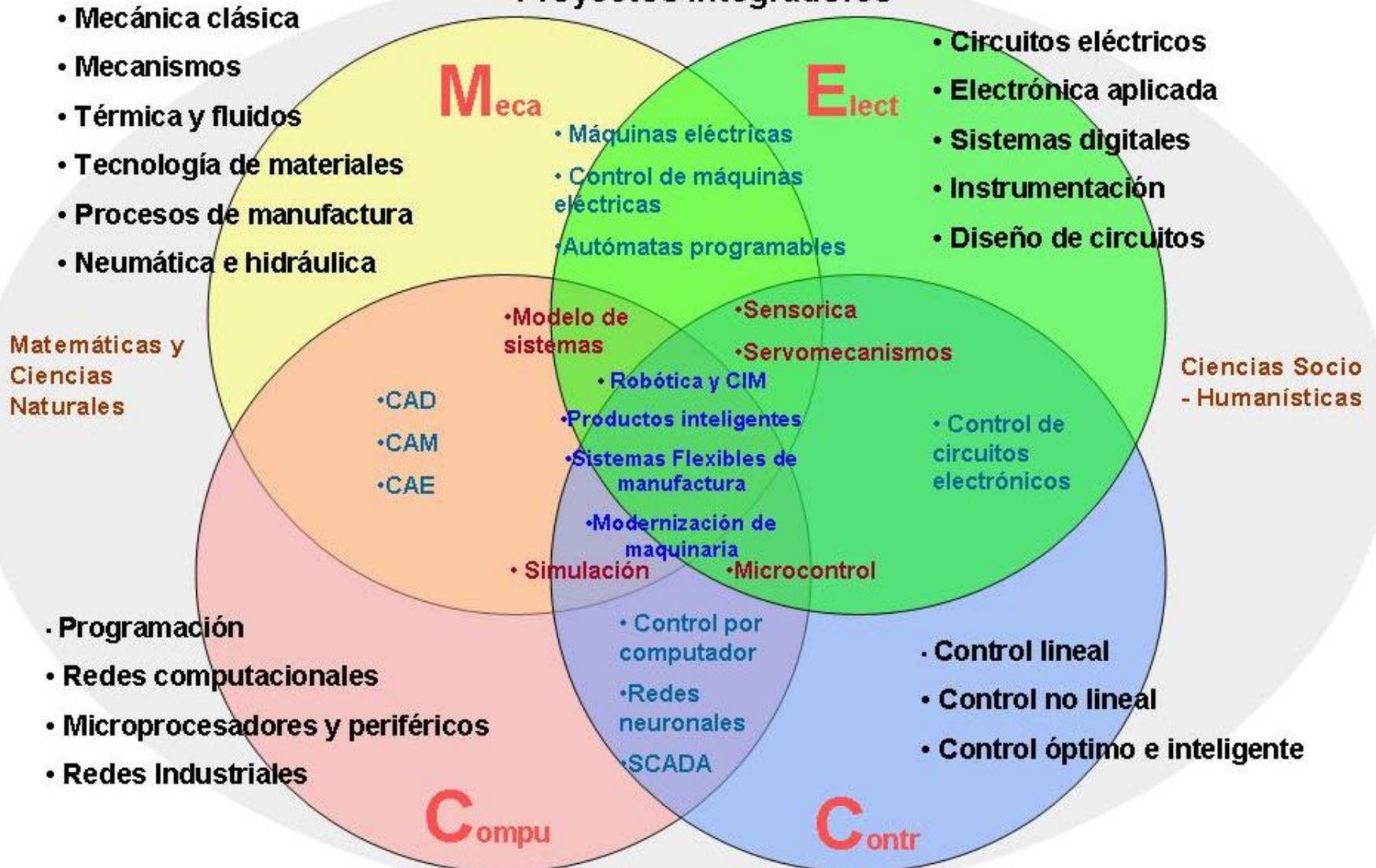
---

# Microcontroladores y Electrónica de Potencia

---

# Mecatrónica

## Proyectos Integradores





Silla



Auto eléctrico



Mano robótica



Brazo robot - Robot  
móvil



Segway



Cámara  
digital



Cuadrícóptero

# ¿Por qué “Microcontroladores y Electrónica de Potencia”?

- Un sistema mecatrónico comprende mecanismos impulsados por uno o más **actuadores** para cumplir funciones de desplazamiento, posicionamiento, esfuerzo etc.
- Estos **actuadores**, que pueden ser motores eléctricos, cilindros neumáticos o hidráulicos y otros, se acoplan a una **articulación** de **rotación** o **traslación** (prismática) para producir el desplazamiento/rotación entre dos piezas del mecanismo (eslabones), y deberán ser controlados para obtener el comportamiento buscado.
- El control es realizado generalmente mediante un sistema electrónico que incluye una etapa de **procesamiento/control** y una etapa de **potencia**.
- El **procesamiento/control** en sus inicios era analógico (ej. PID mediante operacionales), pero actualmente se realiza en plataformas con **microcontroladores, DSPs y lógica programable**, denominadas genéricamente sistemas “embebidos”.
- La etapa de **potencia** depende del tipo y tamaño de motor, y puede ser desde un simple transistor a puentes de transistores, normalmente trabajando en régimen de conmutación (como llaves) por motivos de eficiencia energética.
- Si bien los **Sistemas Embebidos** y la **Electrónica de potencia** son especialidades distintas de la Electrónica, en el análisis y diseño de sistemas mecatrónicos es conveniente su estudio conjunto, pues los fenómenos físicos que se producen en la etapa de potencia con el actuador justifican muchos de los subsistemas presentes en los **microcontroladores**, particularmente en los orientados al **control de movimiento**.

---

UNIDAD 1: Introducción al posicionamiento con ejes electromecánicos servocontrolados. Sistemas de un eje y multieje. (partes A y B)

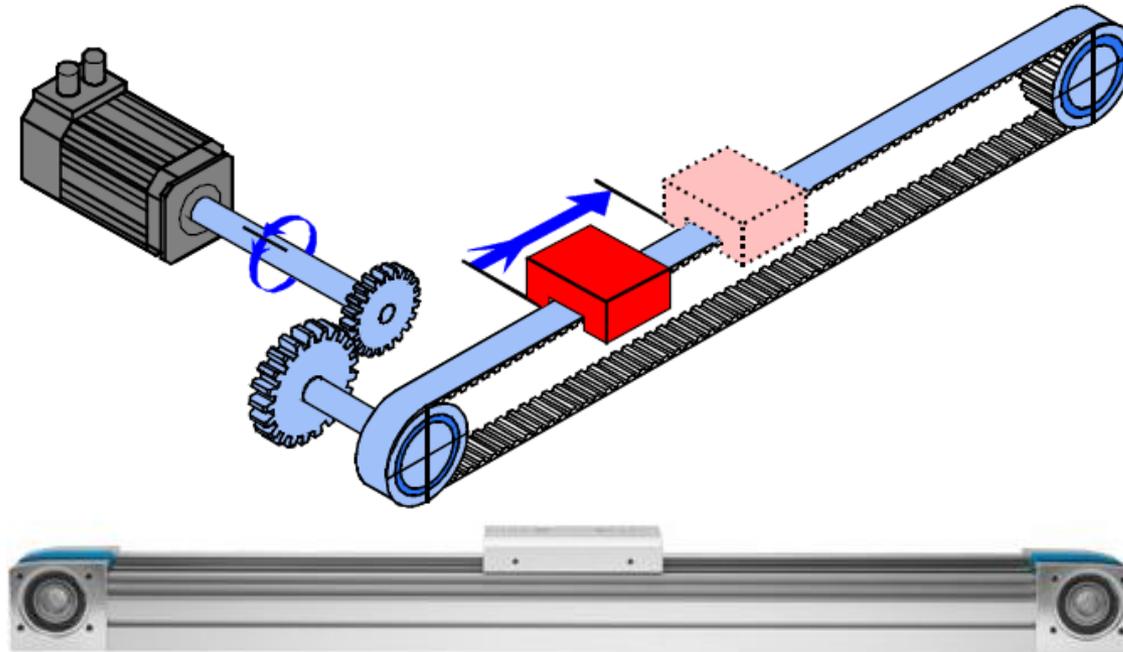
---

---

1.A - Eje mecánico servocontrolado.

---

# Eje electromecánico



- Una articulación activa, es decir impulsada por un actuador, se suele denominar **Eje**.
- Un **eje electromecánico** consta de un **motor** y un conjunto mecánico para obtener el movimiento deseado. En la figura el conjunto mecánico está formado por una caja reductora y un sistema de correa dentada para convertir el movimiento rotativo en desplazamientos lineales (hay muchos otros sistemas).

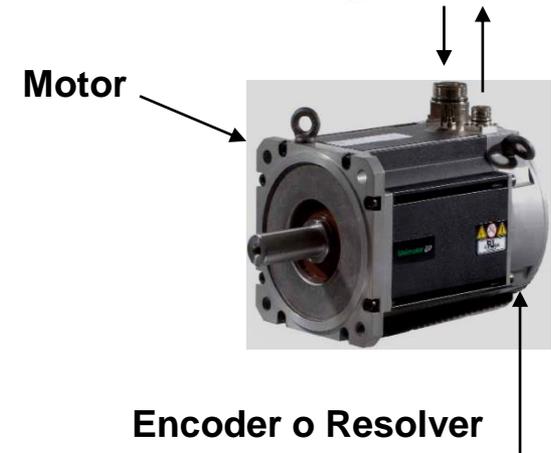
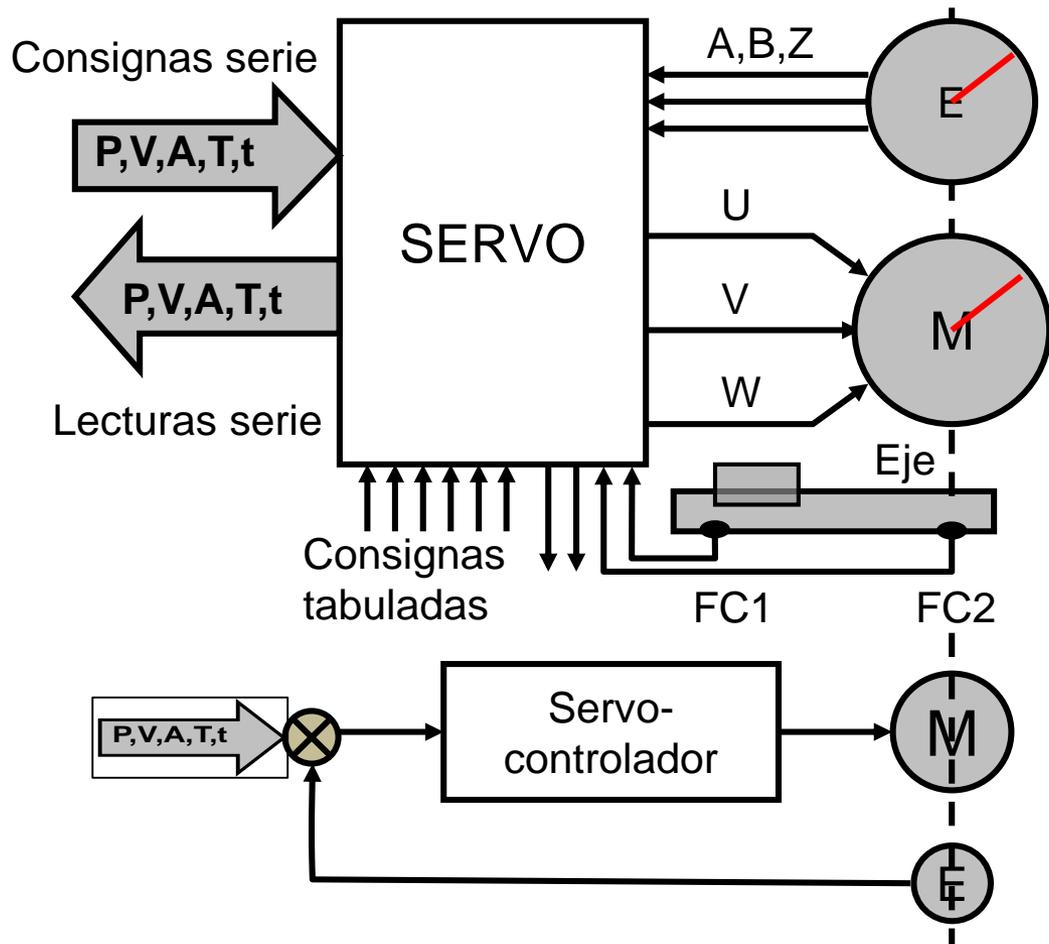
Ejemplo: <https://youtu.be/WZVvlzeVAkw>

# Control de eje

---

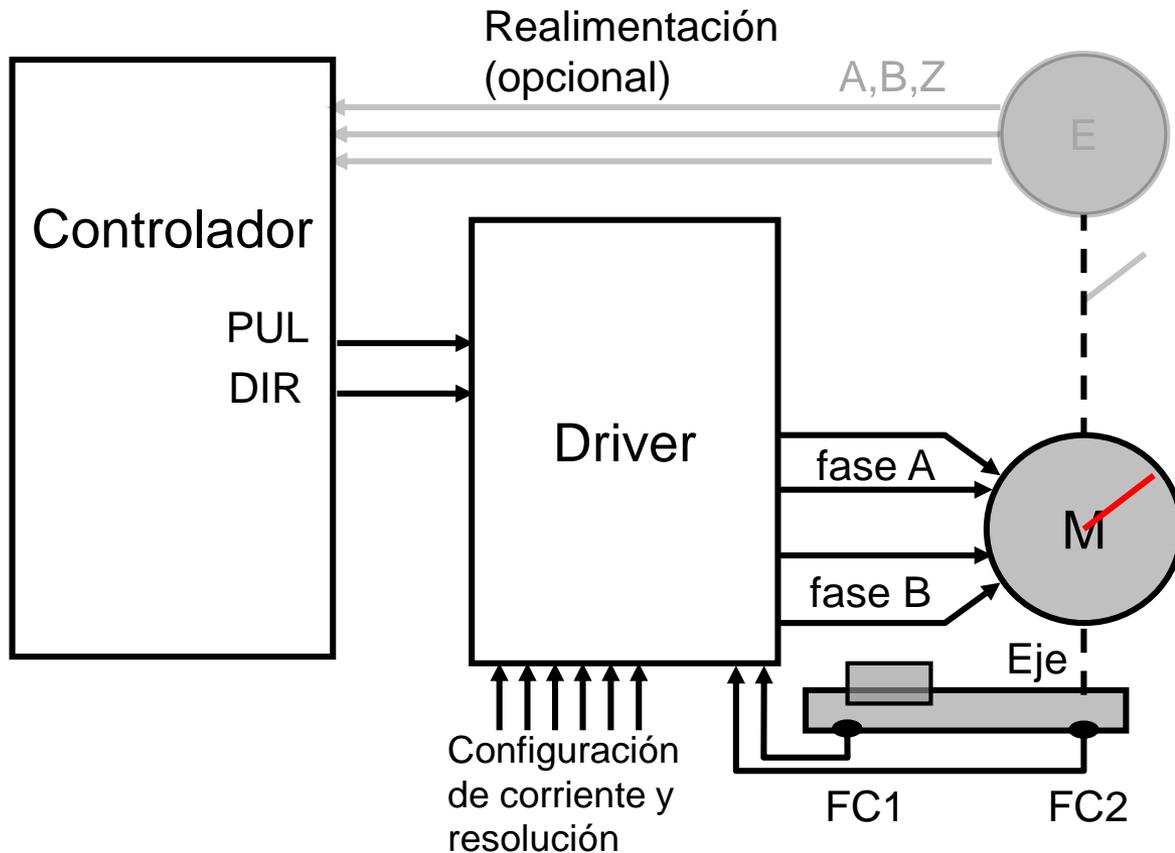
- Objetivos posibles (según aplicación):
  - Alcanzar posiciones deseadas (lineales o angulares) con velocidad/aceleración controladas.
  - Mantener una velocidad controlada o realizar un perfil de velocidad.
  - Realizar una fuerza/par controlada.
- La posición es medida mediante sensores absolutos o incrementales.
- La velocidad puede ser medida derivando posiciones sucesivas, o mediante tacogeneradores.
- El par puede ser medido directamente mediante sensores de fuerza en los ejes (*strain gages*) o indirectamente midiendo la corriente en las fases del motor.
- Cuando se utiliza posicionamiento con sensores incrementales, se requiere una maniobra inicial denominada *homing* para encontrar referencia u origen del eje.

# Eje electromecánico servocontrolado (industrial)



Un eje electromecánico **servocontrolado** está impulsado por un motor **DC** o **AC (brushless)**. Mediante un controlador electrónico con realimentación por sensores (Encoder o Resolver, finales de carrera etc) puede ejecutar consignas específicas de Posición, Velocidad y Torque, con perfiles de velocidad/aceleración. Estas consignas pueden estar tabuladas como valores P,V,A,T,t o ser comunicadas a través de un puerto. Un servocontrolador brushless económico cuesta unos 750 dólares, un servomotor unos 400 dólares

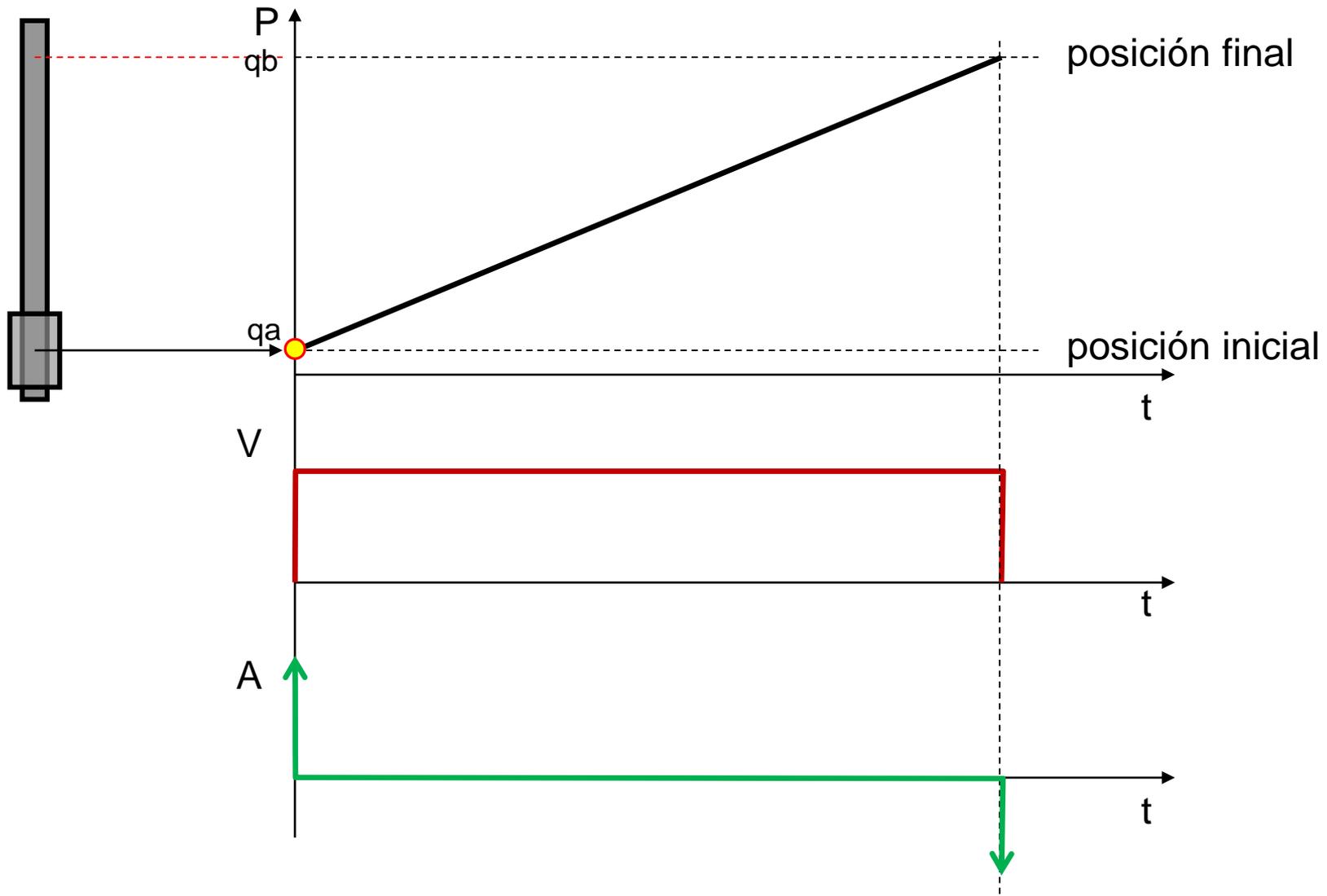
# Eje electromecánico con motor PaP (stepper)



Un eje electromecánico impulsado por **motor PaP** incorpora un controlador lógico (Ej microcontrolador), un **stepper driver** y sensores (finales de carrera, encoder opcional) para ejecutar consignas específicas de Posición y/o Velocidad. Este control es mucho más sencillo, puede realizarse a lazo abierto (sin realimentación de posición) y es muy utilizado en impresoras 3D, routers etc. Un driver puede ser un simple "pololu" de 4 dólares. Los **stepper drivers** industriales cuestan desde unos 60 dólares.

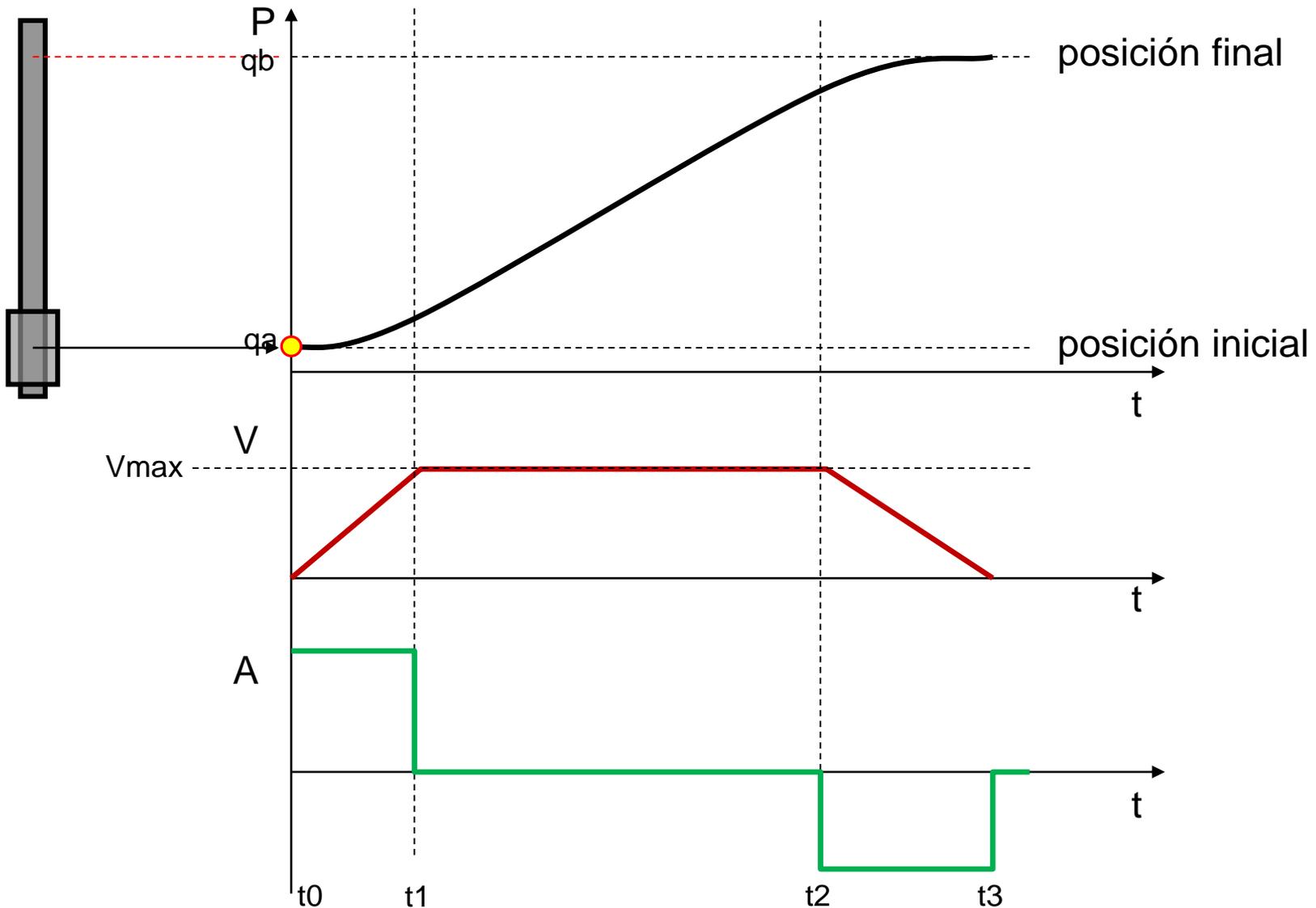
Ejemplos: <https://youtu.be/b5RubQcLpmU> <https://youtu.be/ZbSW3xin5nE>

# Posicionamiento a velocidad constante



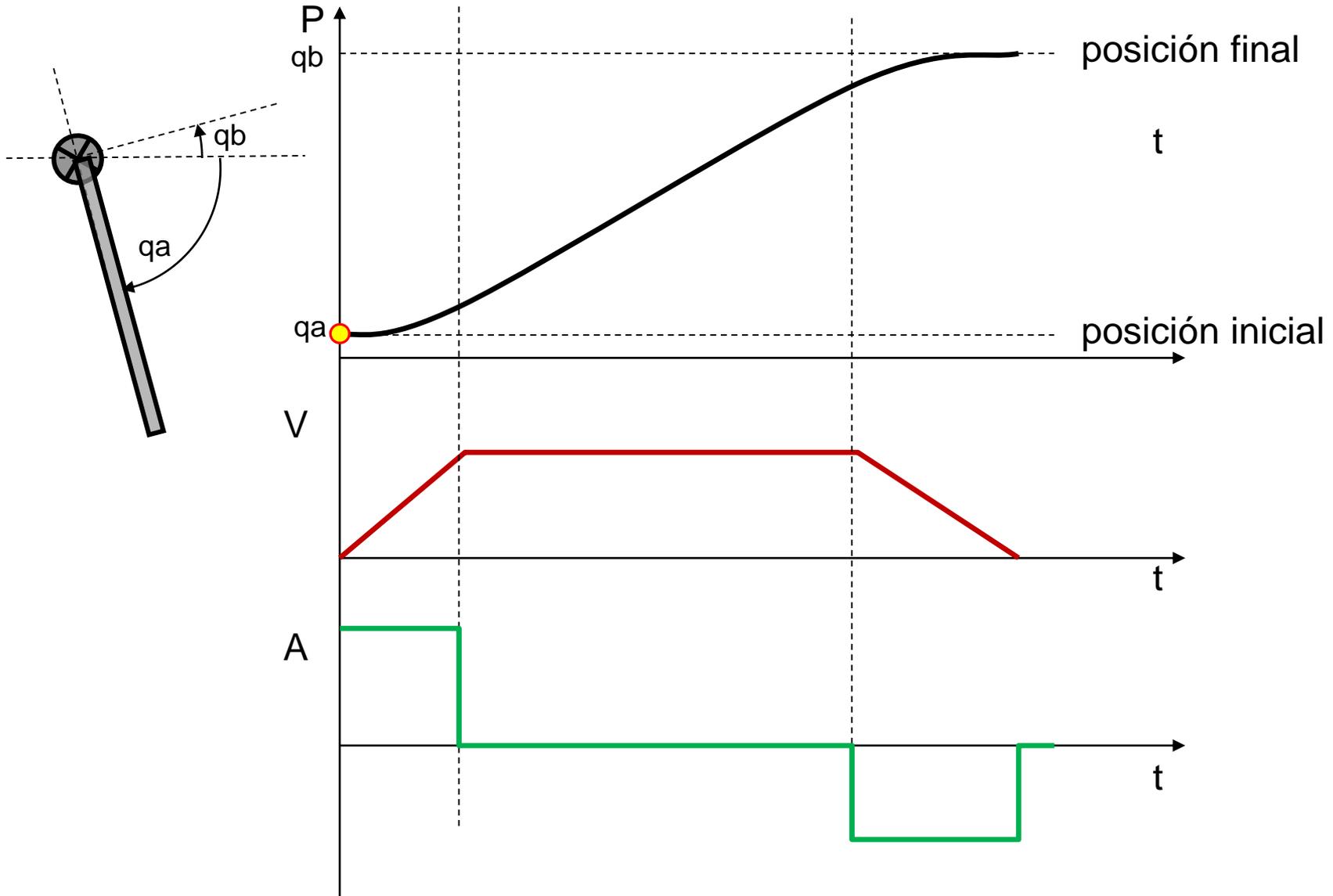
El posicionamiento consiste en ir de la posición actual (inicial) a la posición consigna (final). La idea básica de realizarlo partiendo del reposo a velocidad constante (y concluir en reposo) es físicamente imposible, pues implicaría una aceleración y desaceleración infinitas.

# Interpolador por tramos (V trapezoidal)



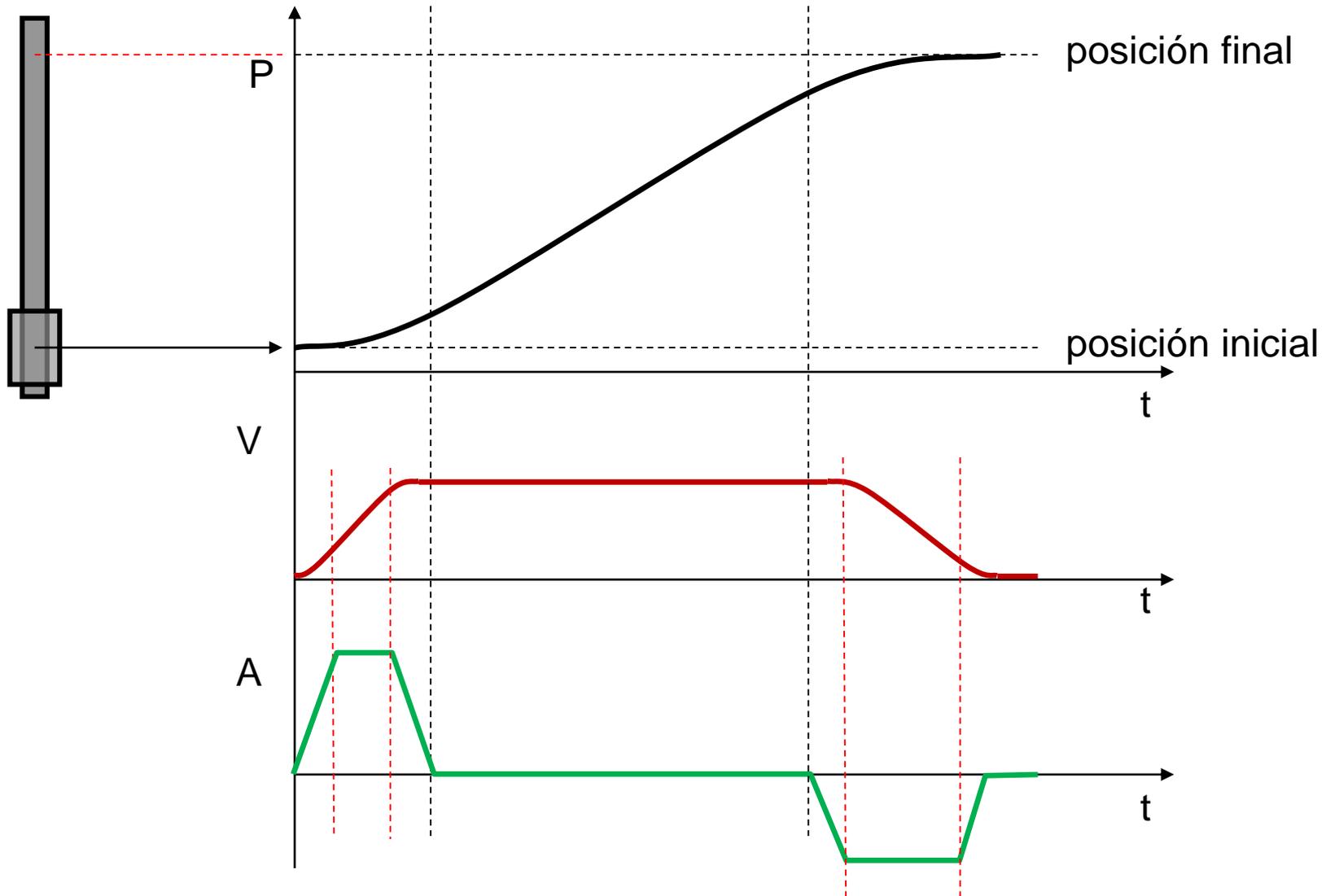
Una de las técnicas más usuales es utilizar un perfil de velocidad trapezoidal, acelerando en forma constante hasta alcanzar la velocidad máxima admisible  $V_{max}$ , y comenzar la desaceleración constante hasta detenerse en la posición final. En caso de tramos cortos el perfil de velocidad será triangular (comienza el frenado antes de alcanzar  $V_{max}$ .)

# Interpolador por tramos (V trapezoidal)



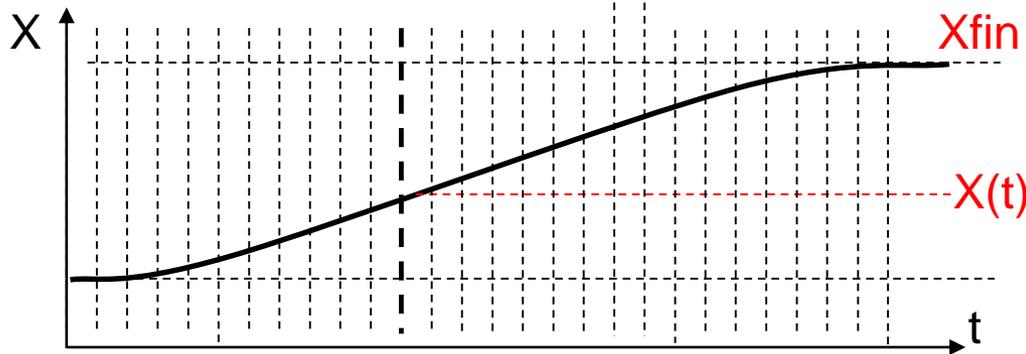
Idem anterior, pero para una articulación de rotación

# Interpolador por tramos (A trapezoidal)

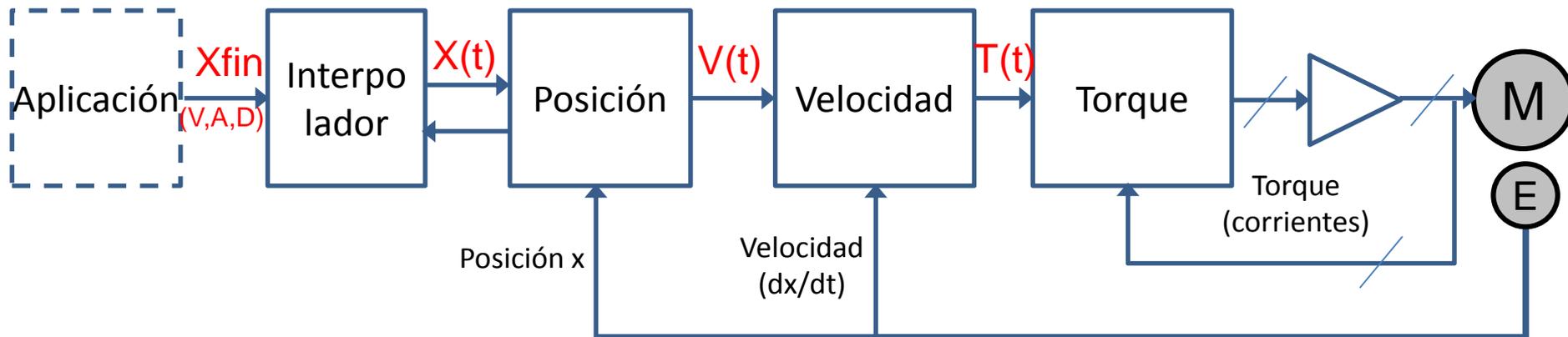


Algunos controladores permiten emplear un perfil de aceleración trapezoidal, para suavizar también el “empuje”. De esta forma se elimina el “jerk”, que es un factor de desgaste en articulaciones de mecanismos con grandes cargas inerciales.

# Modo de posicionamiento por perfil (PP)

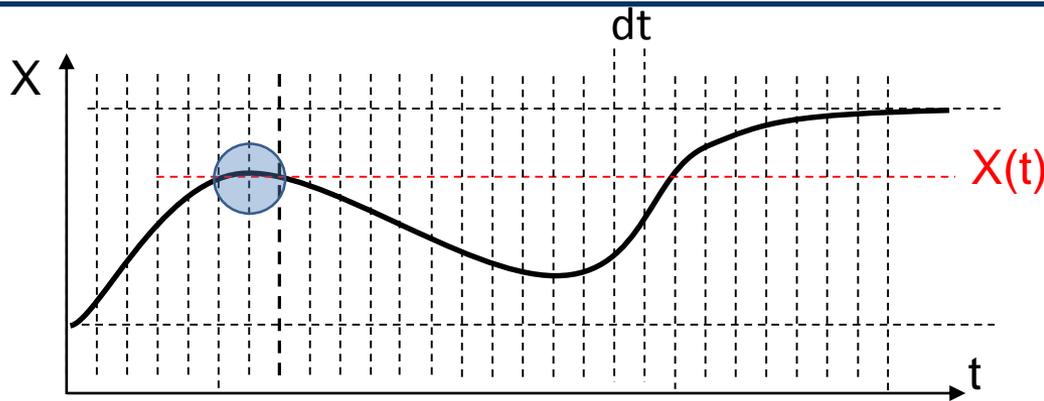


Este es el modo utilizado en sistemas de un solo eje o en control Punto a Punto.

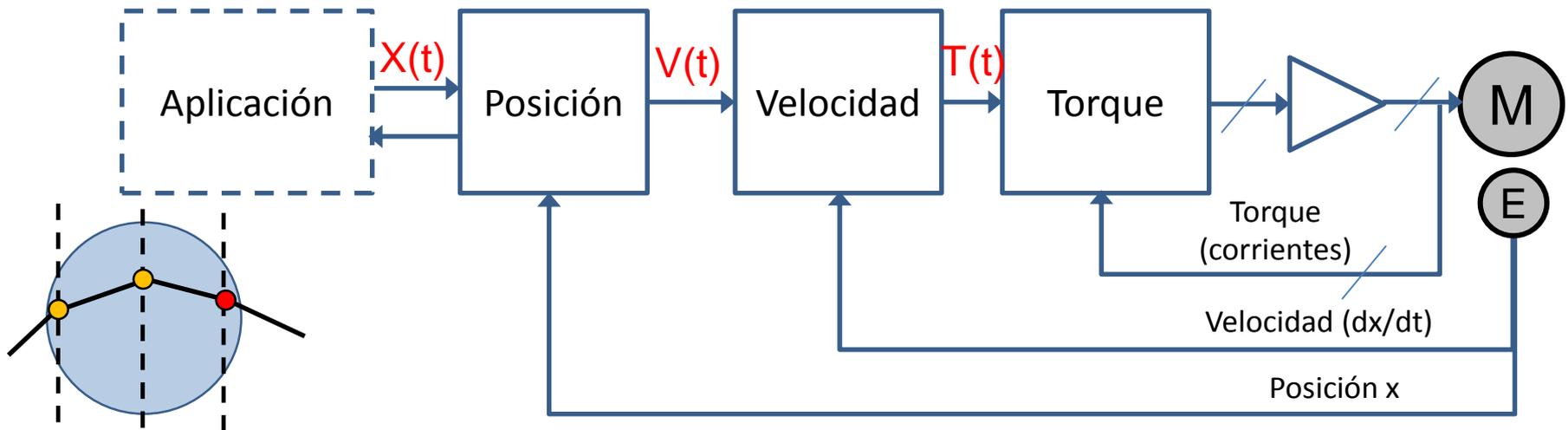


- El usuario (aplicación) configura velocidades y aceleraciones ( $V_{max}$ ,  $A$ ,  $D$  etc). Luego da consigna de posición final  $X_{fin}$ .
- El interpolador genera una secuencia de posicionamientos  $X(t)$  a intervalos iguales de tiempo (por ejemplo 10ms) para cumplir el perfil .
- El control de posición compara la consigna  $X(t)$  con  $x$ , y determina la velocidad necesaria  $V(t)$  para alcanzarla en el tiempo disponible (10ms).
- El control de velocidad compara esta consigna  $V(t)$  con su velocidad  $v=dx/dt$ , y determina la aceleración o torque necesario  $T(t)$  para alcanzarla.
- El control de torque compara esta consigna de aceleración (o torque) con la lectura de las corrientes, y actúa sobre las puertas de los transistores de salida para alcanzar la consigna.

# Modo de Posición interpolada (IP)

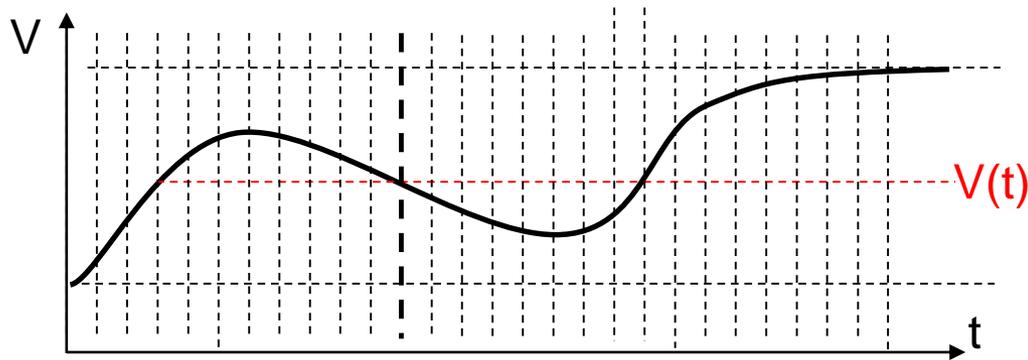


Este es el modo utilizado en sistemas de posicionamiento multi-eje, como control cinemático de robots, CNC etc.

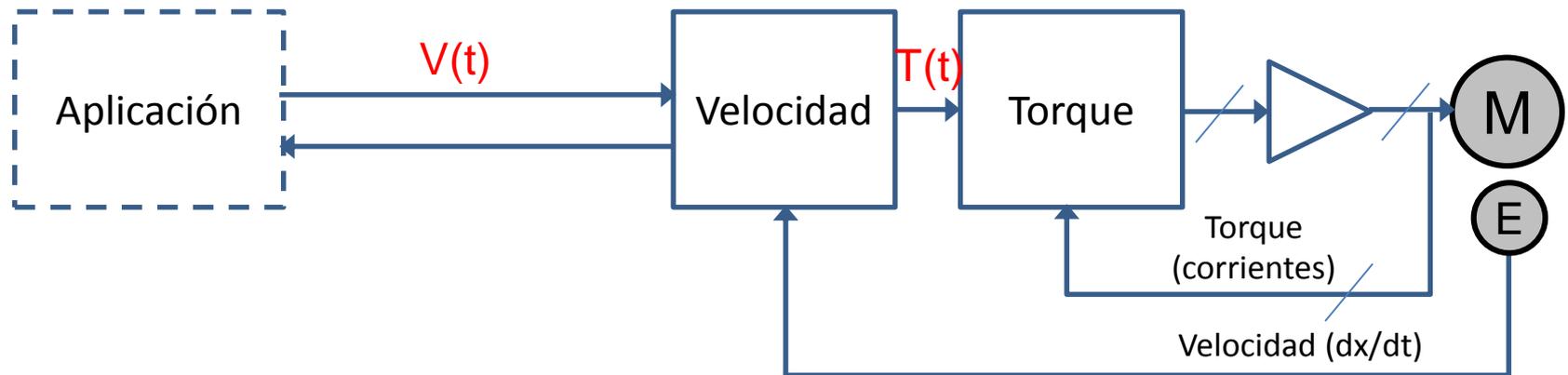


- El usuario (aplicación, por ejemplo el Controlador Cinemático) reemplaza al interpolador, y debe encargarse de generar la secuencia de posicionamientos  $X(t)$  a intervalos iguales de tiempo (por ejemplo  $dt=10ms$ ).
- El resto de las etapas de control se mantiene. En este modo el servocontrolador no aplica límites de aceleración, sino que asume velocidad constante entre 2 posiciones sucesivas (interpolación lineal de posición entre 2 puntos). Debido a que el tiempo entre consignas de posición es muy pequeño (ej 10 ms) los escalones de incremento de velocidad son reducidos. Algunos servocontroladores utilizan interpolación de grado 2 o superior utilizando 3 o más puntos de posiciones sucesivas.

# Modo de Perfil de Velocidad (PV)



Este es el modo utilizado en leva electrónica, en control cinemático multi-eje en velocidad etc.



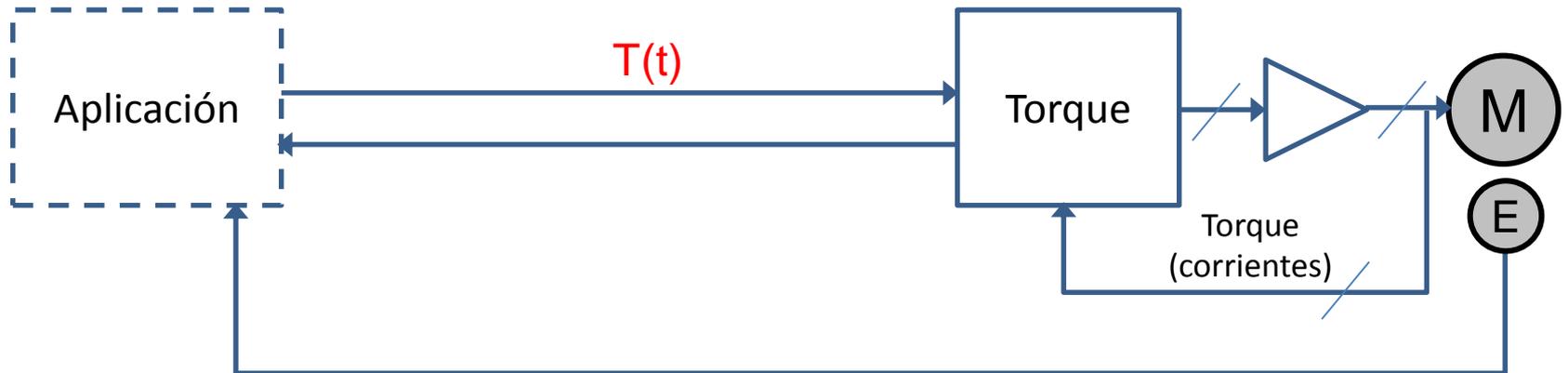
Se elimina el control de posición.

El usuario (aplicación, por ejemplo el Controlador Cinemático) genera el perfil de velocidad deseado  $V(t)$ .

El resto de las etapas de control hacia adelante se mantiene.

# Modo Torque (T)

Este es el modo utilizado en control dinámico de robots, o cuando la aplicación necesita realizar esfuerzos determinados en el eje etc.

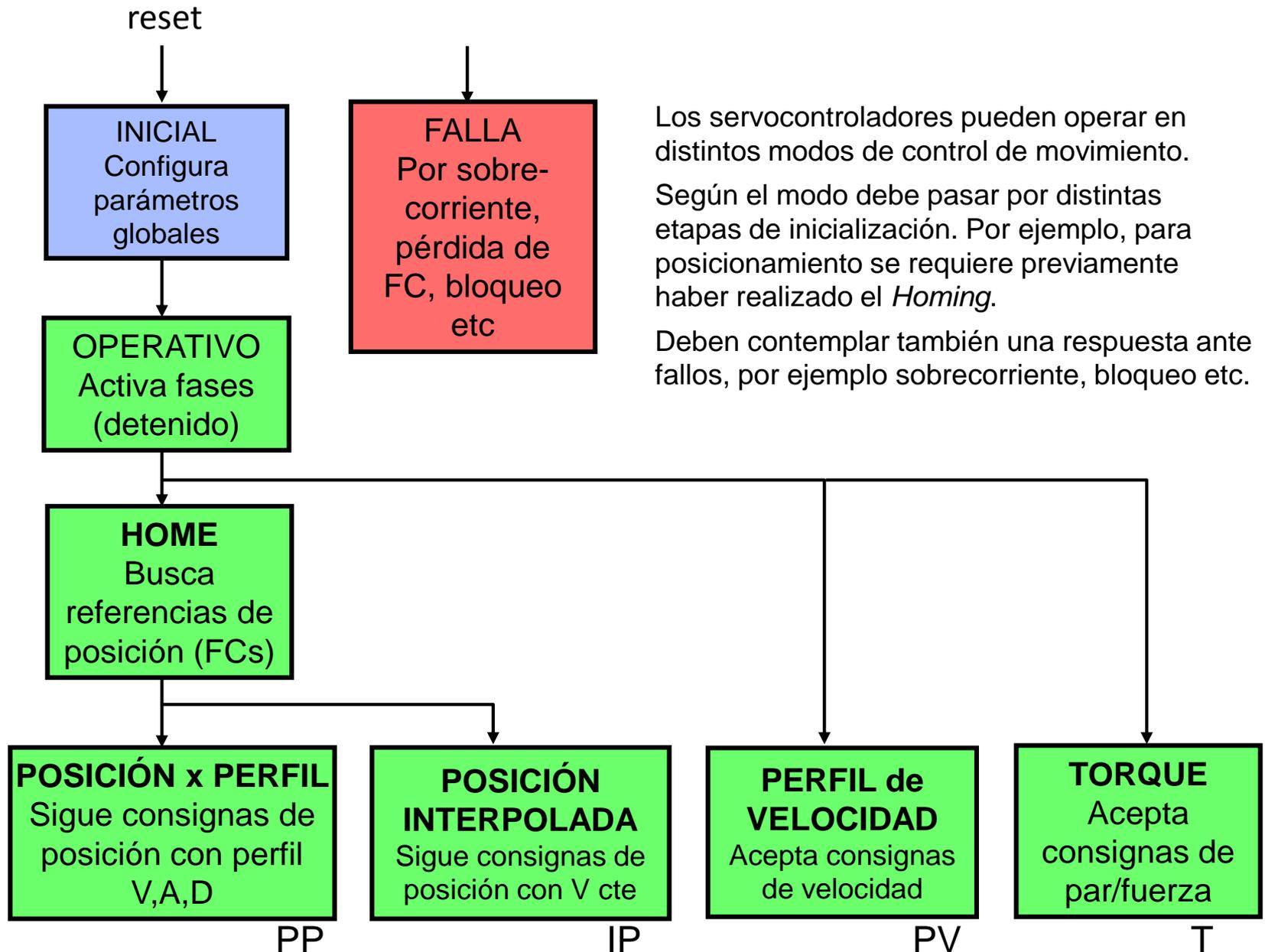


Se eliminan los controles de posición y velocidad.

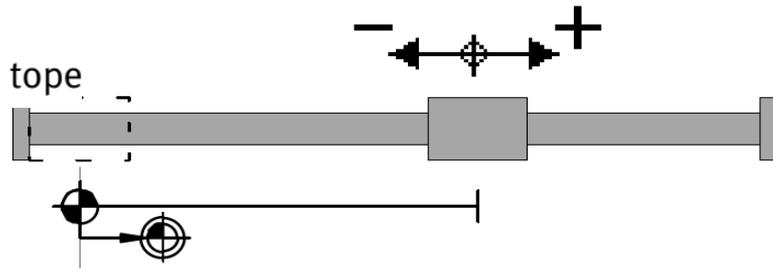
El usuario (aplicación, por ejemplo el Controlador Dinámico) comunica el Torque deseado  $T(t)$  que puede variar en el tiempo.

La etapa de control de Torque intenta cumplir esta consigna.

# Máquina de estados básica de un servo



# Métodos de referenciación (*homing*)

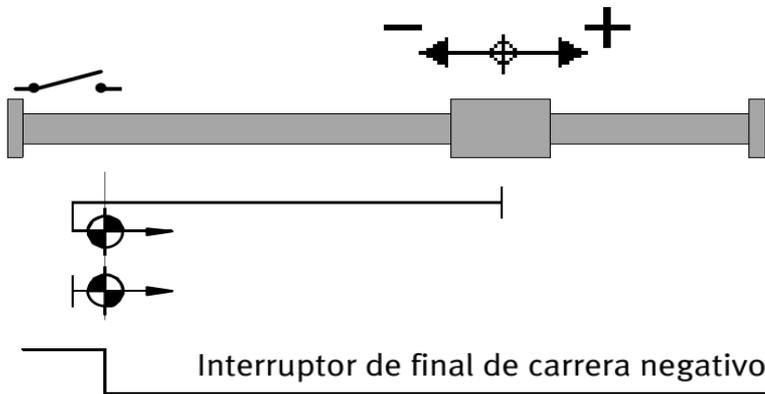


## búsqueda de tope mecánico

Retrocede a velocidad de búsqueda.

Al detectar **bloqueo (\*)** toma ese punto como referencia, y avanza hasta el 'cero', que debe estar despegado del tope

(\*) En servos brushless se hace por sobrecorriente

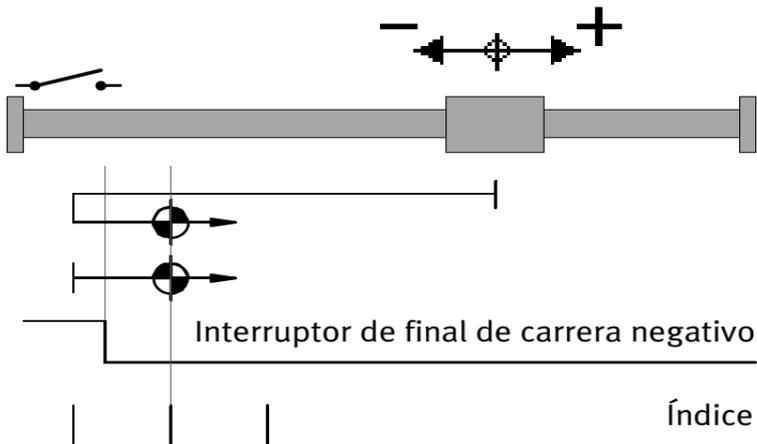


## búsqueda de final de carrera FC

Retrocede a velocidad de búsqueda.

Al detectar FC cambia de sentido a velocidad reducida hasta que deja de detectar FC, y toma ese punto como referencia.

Si retrocediendo no detecta FC luego de un tiempo mayor al necesario en el caso más desfavorable, puede indicar error.



## búsqueda de final de carrera FC con índice del sensor

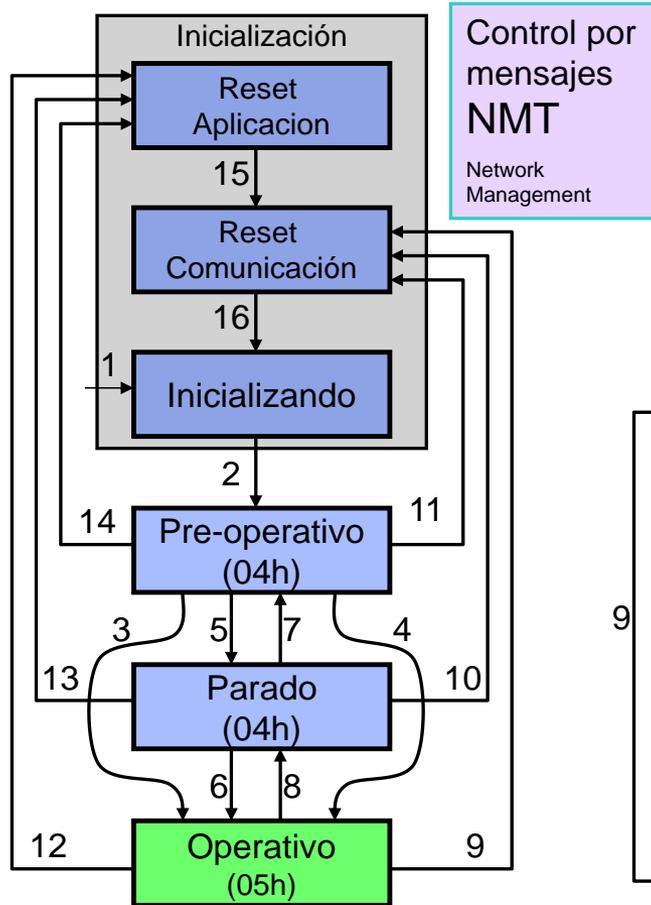
Retrocede a velocidad de búsqueda.

Al detectar FC cambia de sentido a velocidad reducida hasta que deja de detectar FC, y continúa hasta que detecta el índice del encoder vinculado al motor.

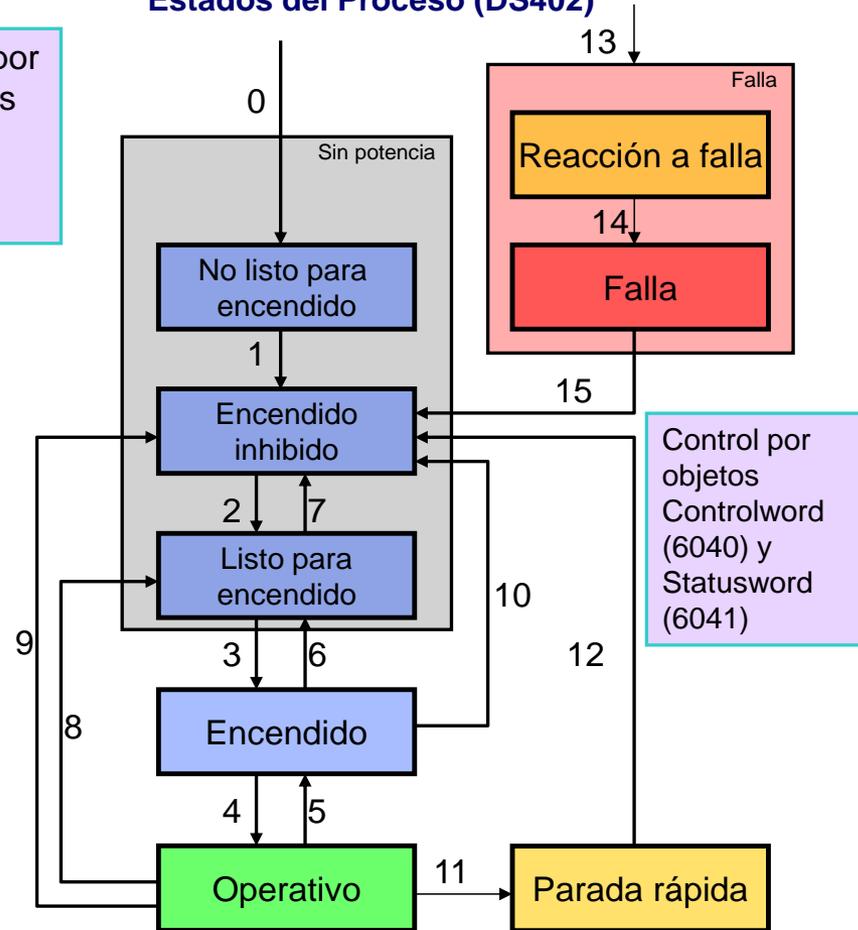
Si retrocediendo no detecta FC luego de un tiempo mayor al necesario en el caso más desfavorable, puede indicar error.

# Servocontrolador. Estados según especificaciones CANopen

## Estados de la Comunicación (DS301)



## Estados del Proceso (DS402)

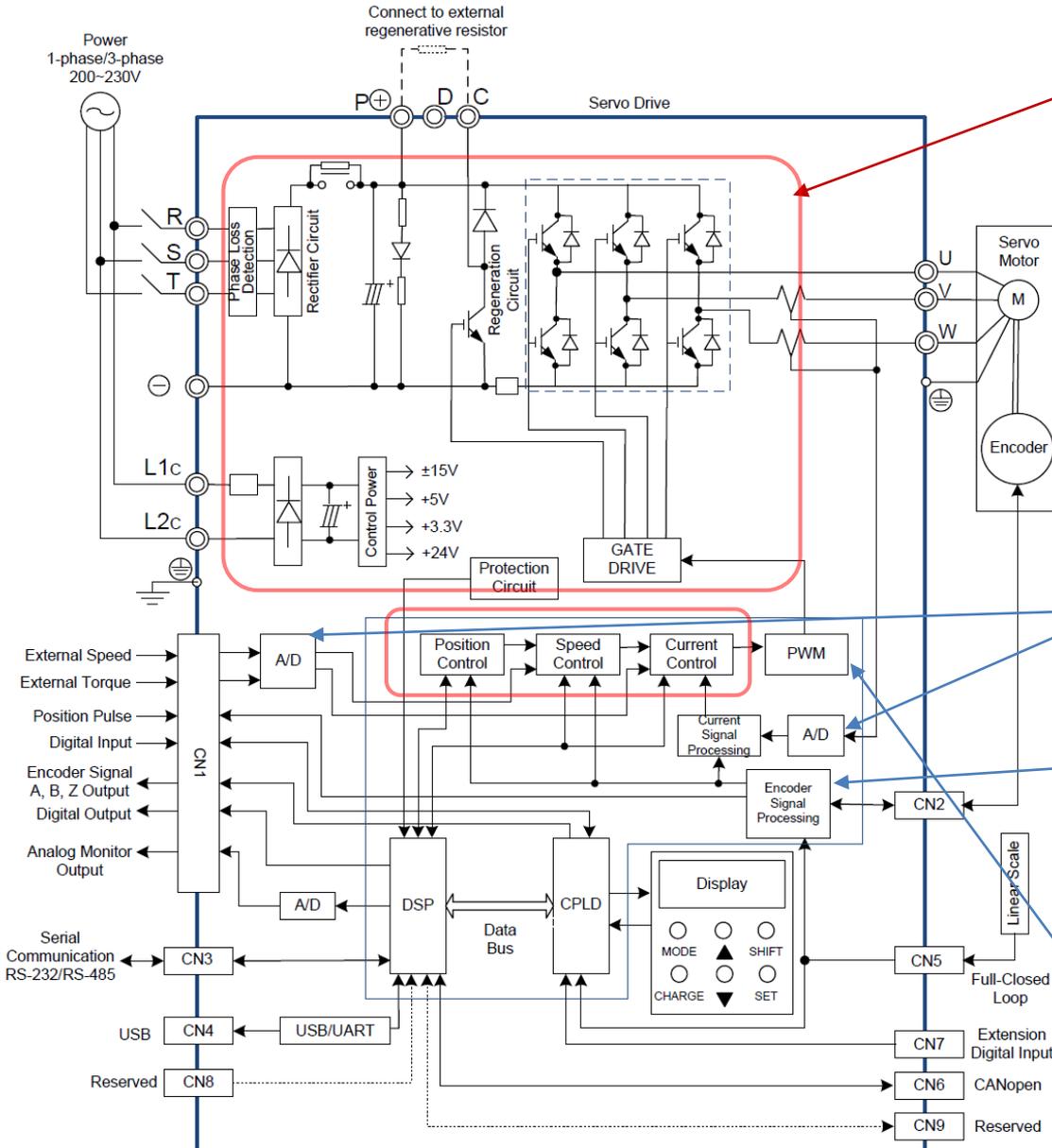


El software de servocontrolador puede contar con otros estados, por ejemplo para cumplir con algún estándar.

Un ejemplo son los servocontroladores que incluyen el protocolo CANopen.

Este protocolo especifica una máquina de estados para la Comunicación (Draft Standard 301) y una máquina de estados para el Proceso (Draft Standard 402).

# Ejemplo: Esquema del controlador DELTA ASDA-A2



La etapa de potencia está constituida por un rectificador de alta tensión (de 220V monofásicos o trifásicos) y un puente trifásico, de transistores MOSFET ó IGBT.

El hardware de procesamiento incluye núcleos DSP (digital signal processing), CPLD (lógica programable) y módulos de E/S para medición, generación de señales y comunicaciones:

Las **entradas analógicas** del  $\mu\text{C}$  miden corriente (para determinar torque), voltaje (para detectar fuerza contra-electromotriz o *Back EMF*) y consignas de Velocidad y/o Torque dadas en forma de tensión.

Las entradas de **QEI**, **Timers** conectados en modo contador o medición precisa de tiempos, y **entradas de captura** para sensores Hall permiten determinar directa o indirectamente posición y velocidad del eje, para realizar su control

Las salidas del  $\mu\text{C}$ , directas o moduladas (**PWM**) conmutan, a través de "*gate drivers*", los transistores que comandan el motor.

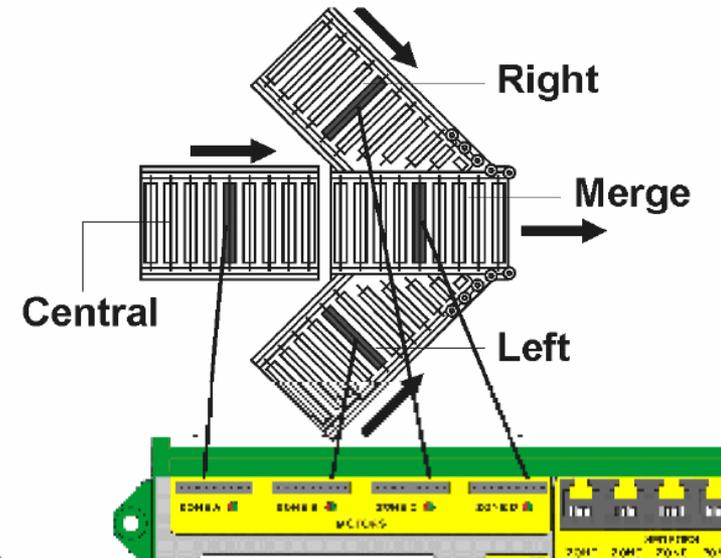
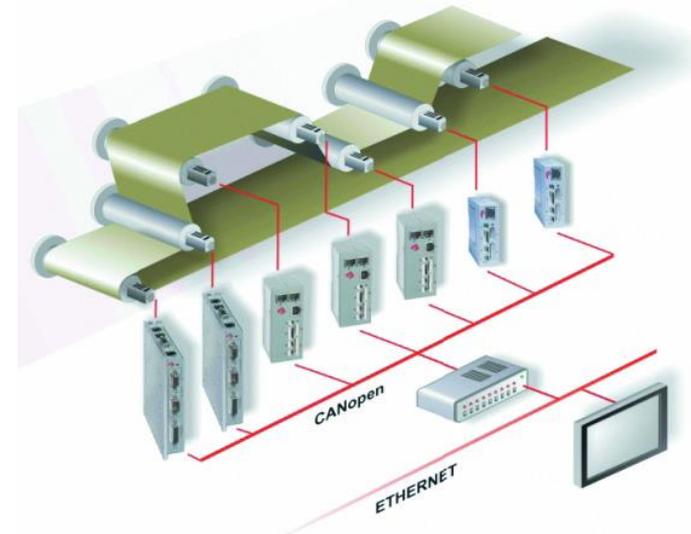
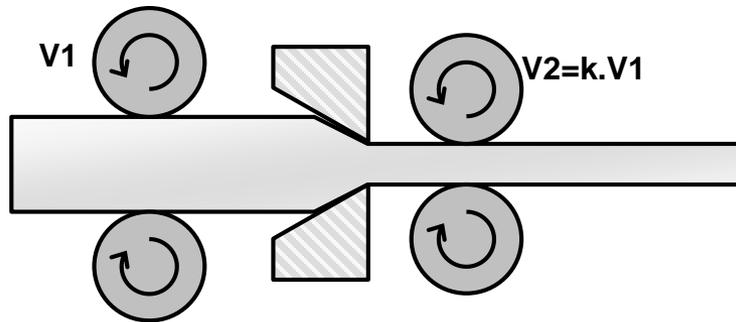
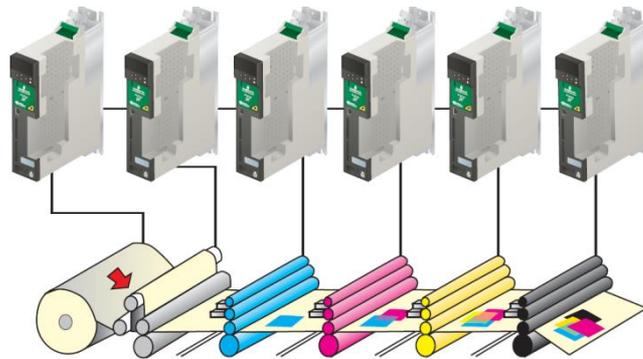
---

## 1.B Sistemas multi-eje. Generación de trayectorias.

---

# Sistemas multieje (1) Control de velocidad

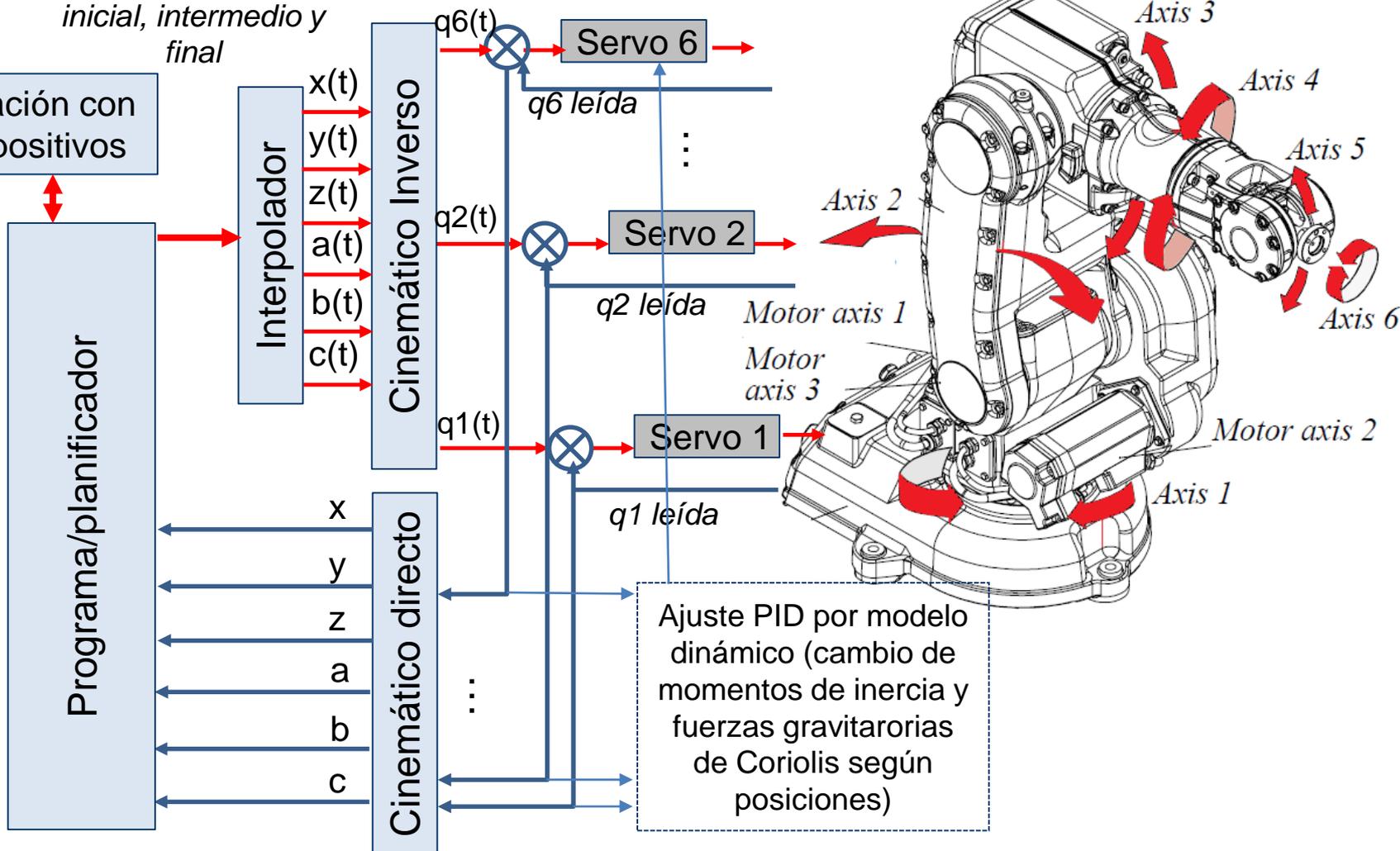
- Control en velocidad
- Trefiladoras
- Prensa y gráfica
- Industria Textil
- Sistemas de transporte interno



# Sistemas multieje (2): Control cinemático de un robot

Trayectorias  
definidas por  
puntos  $x, y, z, a, b, c$   
inicial, intermedio y  
final

Comunicación con  
otros dispositivos

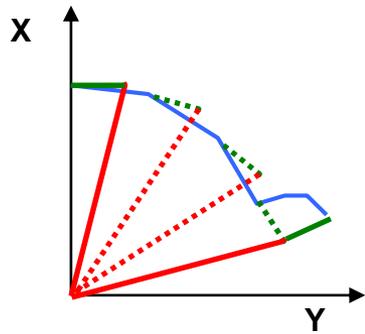
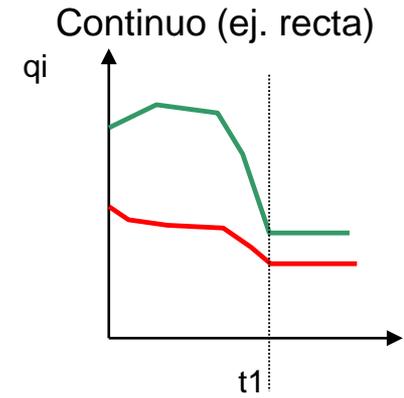
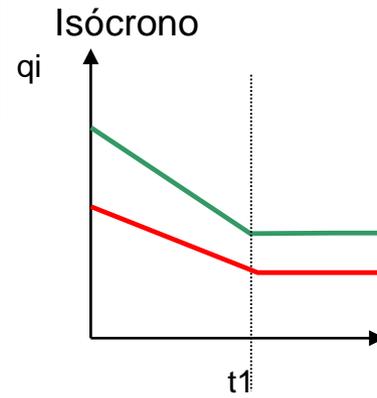
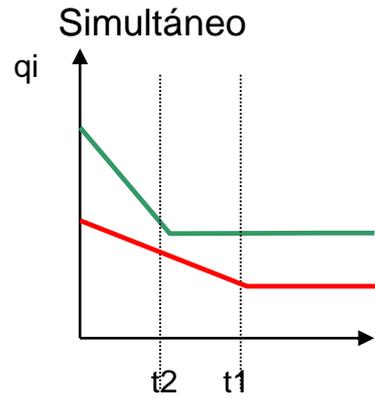
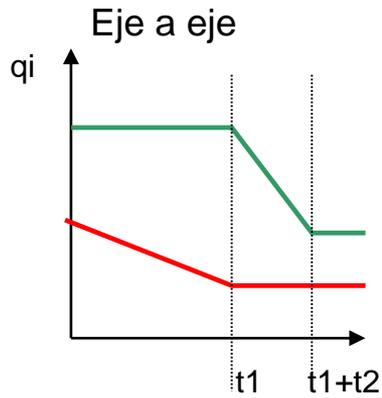


# Posicionamiento en el espacio

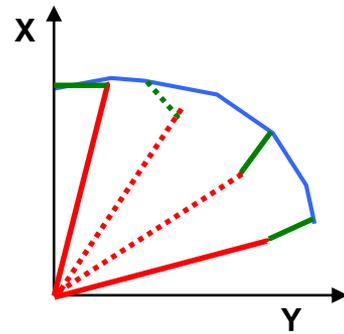
- Objetivo: Alcanzar una posición final  $(x,y,z,a,b,c)$  con velocidad/ aceleración controladas, y generalmente siguiendo una trayectoria definida, sea de forma analítica (recta, arco etc) o de forma gestual (copiada de un gesto realizado por un operador humano). La tarea de un robot será una sucesión de trayectorias y accionamientos sincronizados de herramientas, a la vez que comunicarse y sincronizarse con otros sistemas.
- También se puede buscar minimizar tiempos de ciclo (ej. *pick&place*), velocidades precisas (aplicación de material, soldaduras de arco etc).
- Especialmente deben controlarse las aceleraciones por restricciones mecánicas.

Para conseguir una trayectoria en el espacio es necesario coordinar los movimientos de sus ejes, no basta con que cada eje busque su posición final.

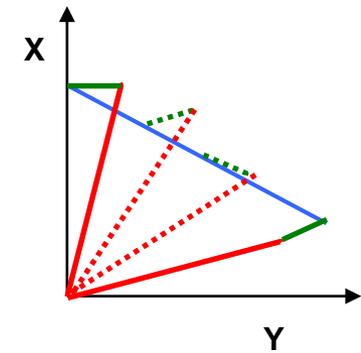
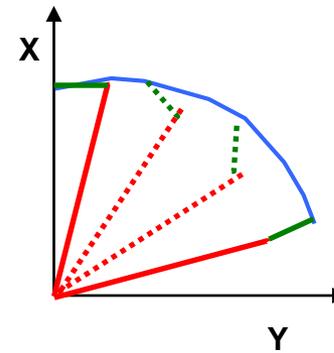
# Tipos de movimiento multieje



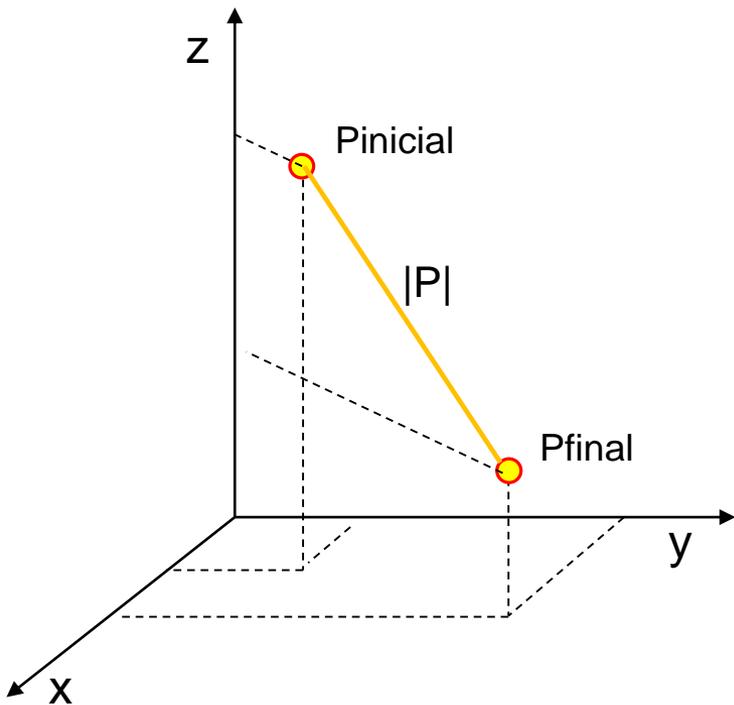
No coordinados



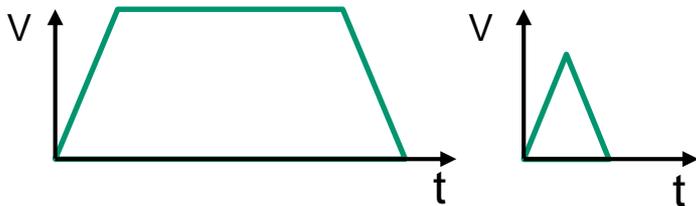
Coordinados



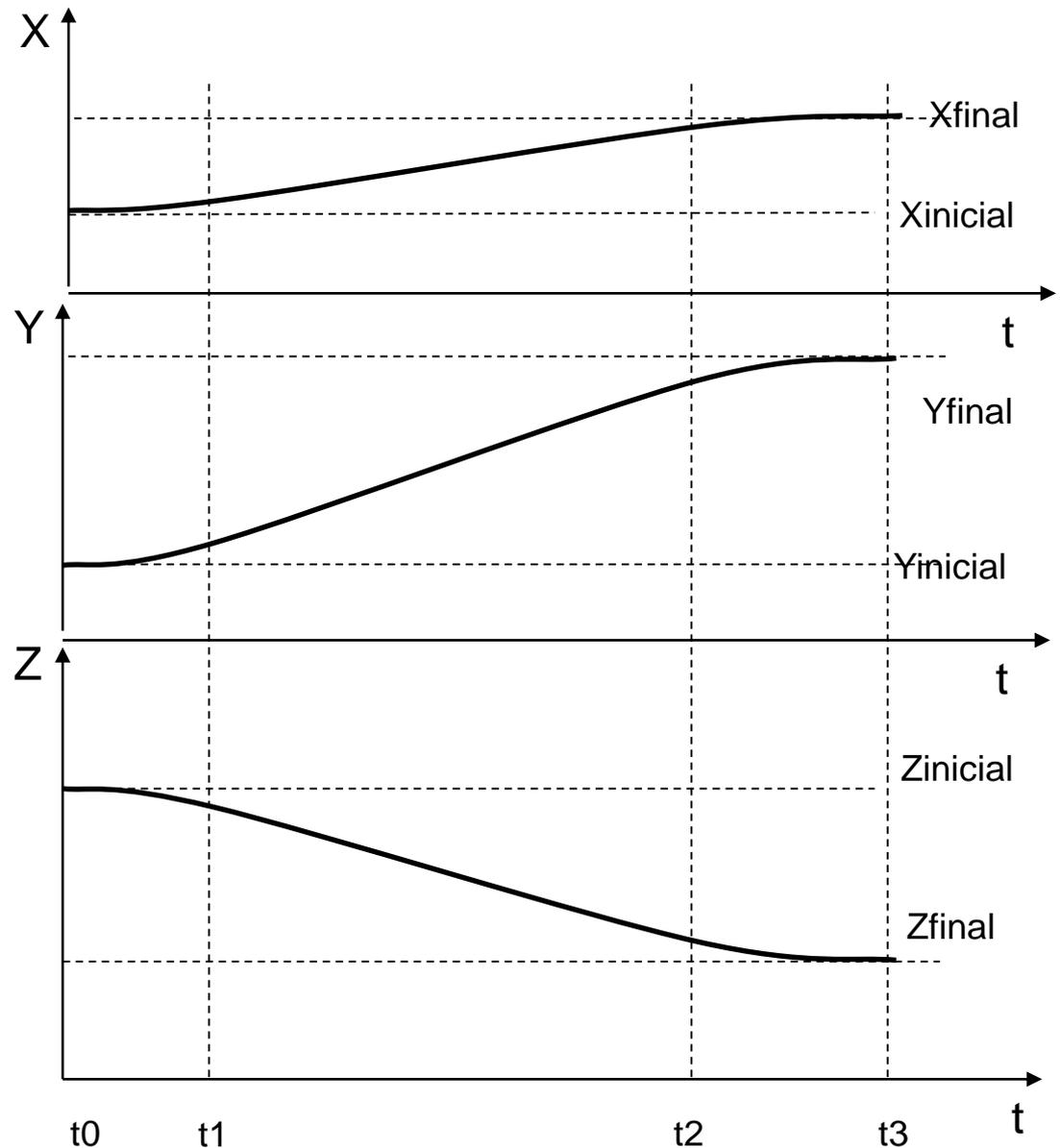
# Ejemplo de Movimiento: Recta en el espacio. 1- Obtener $|P|$ , $t_1$ , $t_2$ , $t_3$



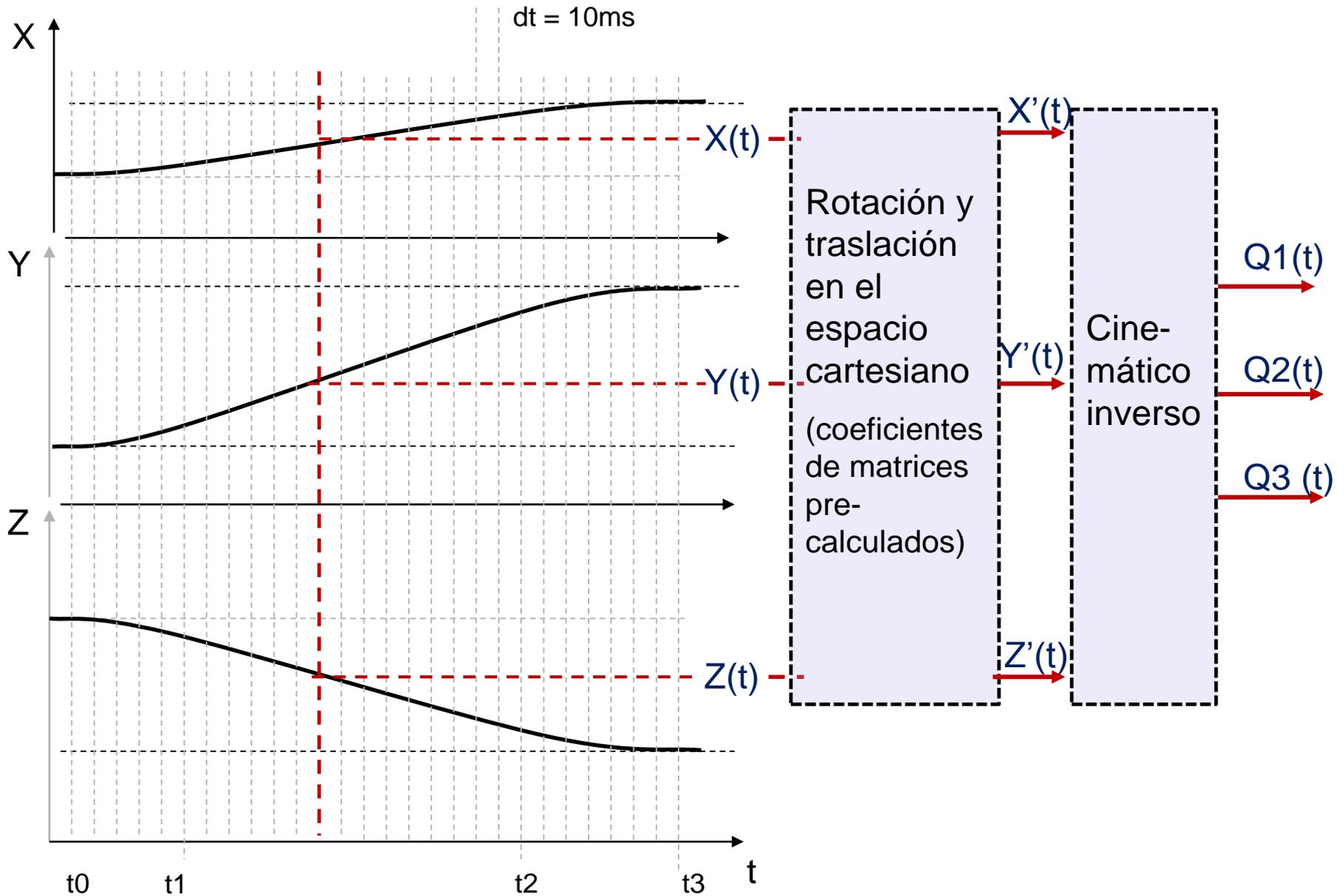
Dadas  $A$ ,  $V_{max}$  y  $D$  se calculan los instantes  $t_1$ ,  $t_2$  y  $t_3$



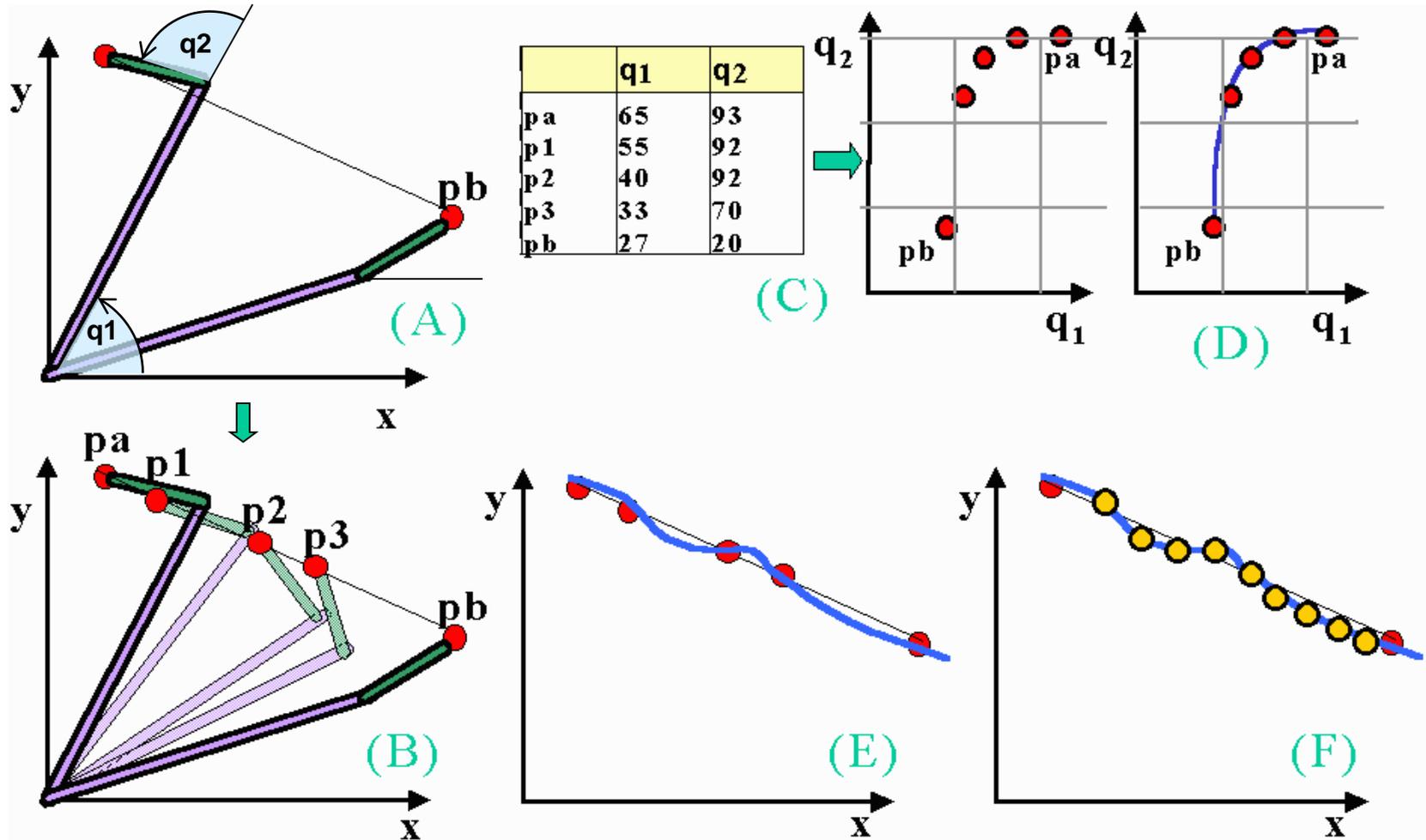
En distancias cortas no se alcanza la  $V_{max}$  (no existe el tramo  $t_1-t_2$ )



## (2) Calcular puntos interpolados cada 10 ms, enviar Q1, Q2, Q3



# Ejemplo: Generación de Trayectoria Recta en un plano x-y





# Comunicación en sistemas multieje (introducción)

## Información:

**Poco frecuente:** Configuración de sensores  
Factores de reducción

**Mediana frecuencia:** Aceleraciones

**Alta frecuencia:** Posición, Velocidad.  
Torque (en control dinámico)

**Crítica:** Control de estado, fallas

## Características requeridas:

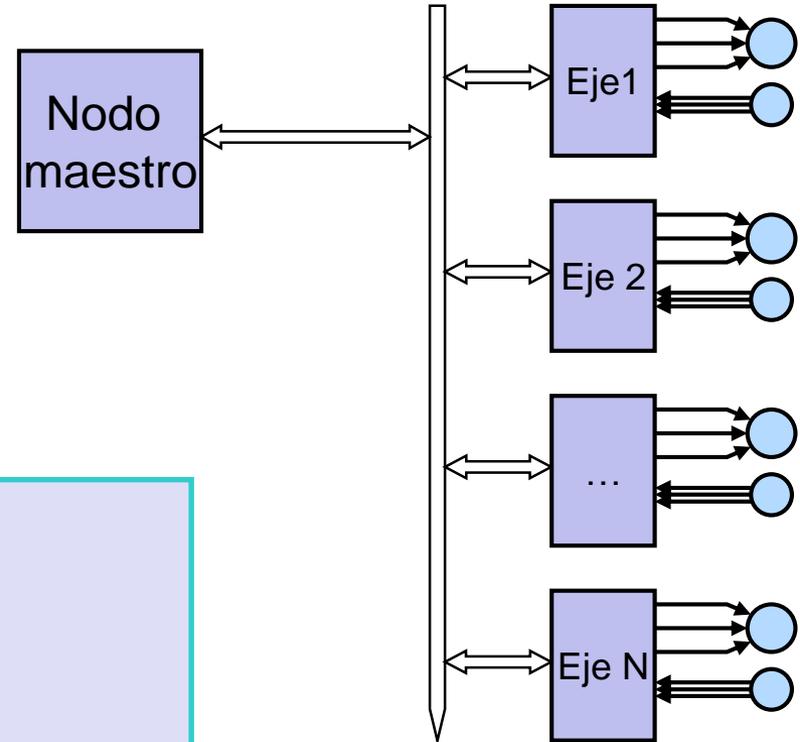
Ciclos de lazo cortos (<1ms)

Sincronización/determinismo (*jitter*<100us)

Comunicación de eventos (de Eje a nodo maestro)

Recuperación de errores

Retransmisión de mensajes erróneos



# Buses/protocolos de comunicación

---

## Interfaces de $\mu$ C

UART

SPI

I2C

CAN

Ethernet

## Protocolos

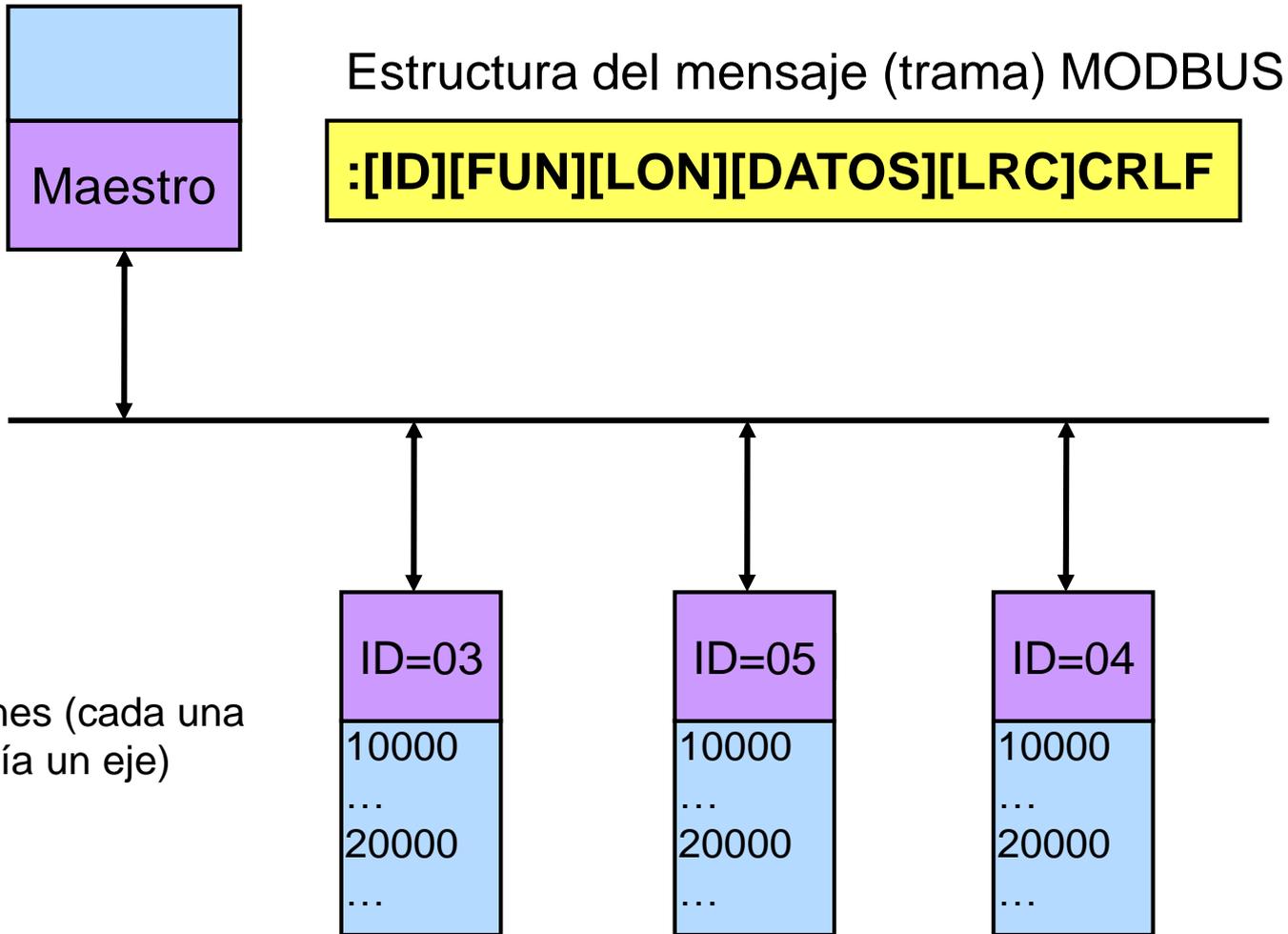
Modbus – Profibus PA

PMBus – SMBus

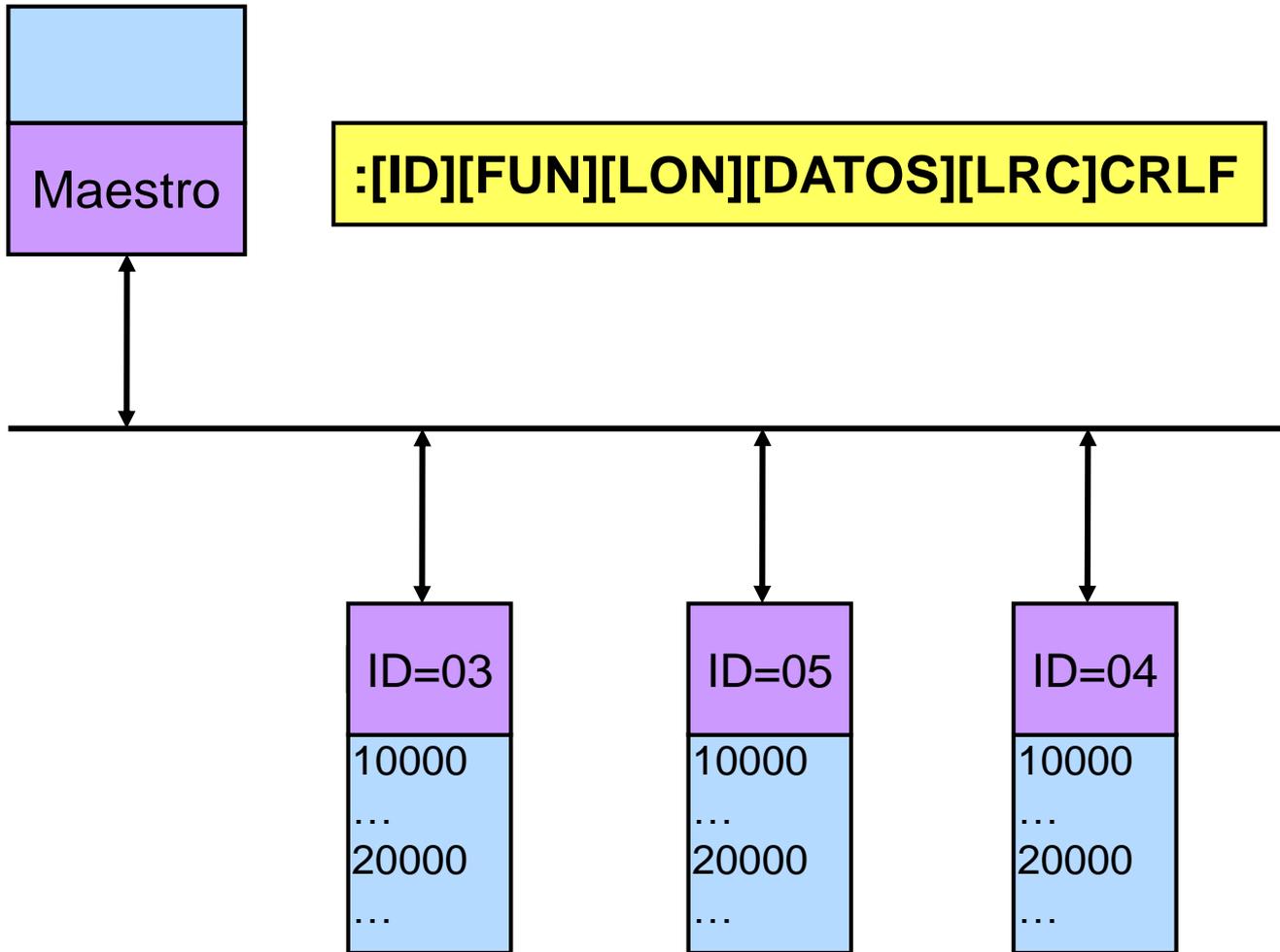
CAN/**CANopen**/DeviceNet

Modbus TCP – UDP – Profinet – **Ethercat** – **SERCOS III**

# Maestro-Esclavo (Ej. MODBUS)

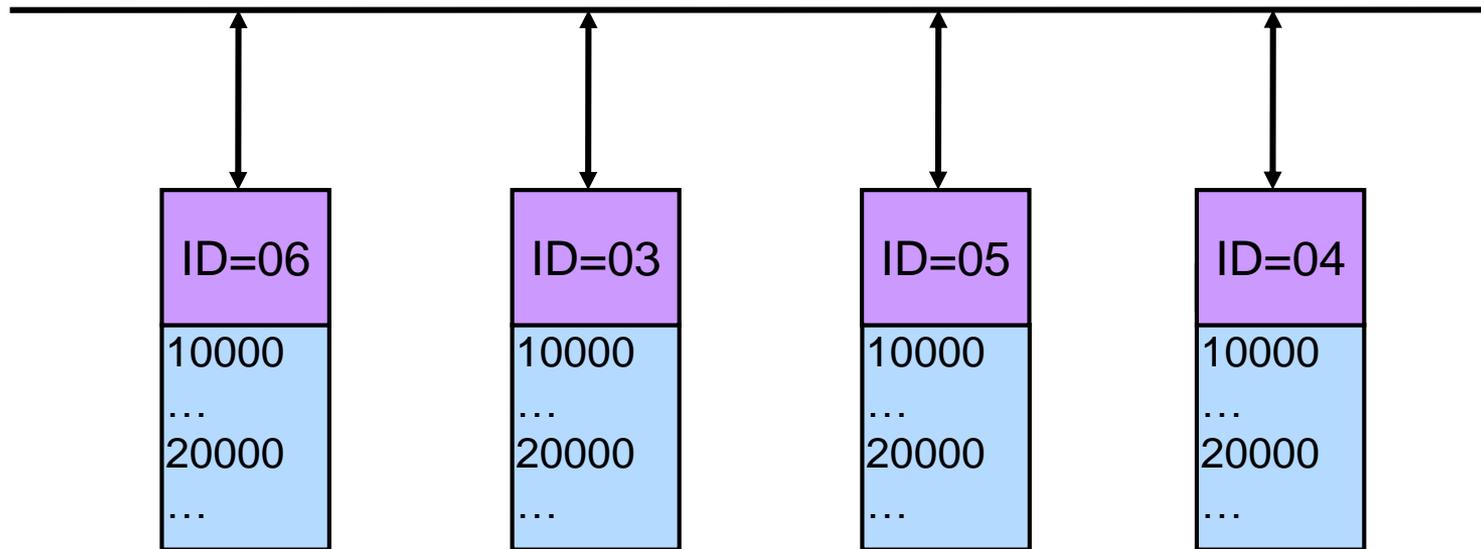


# Maestro-Esclavo (Ej. MODBUS)



# CSMA/CD (Ej. Ethernet en un mismo dominio de colisión)

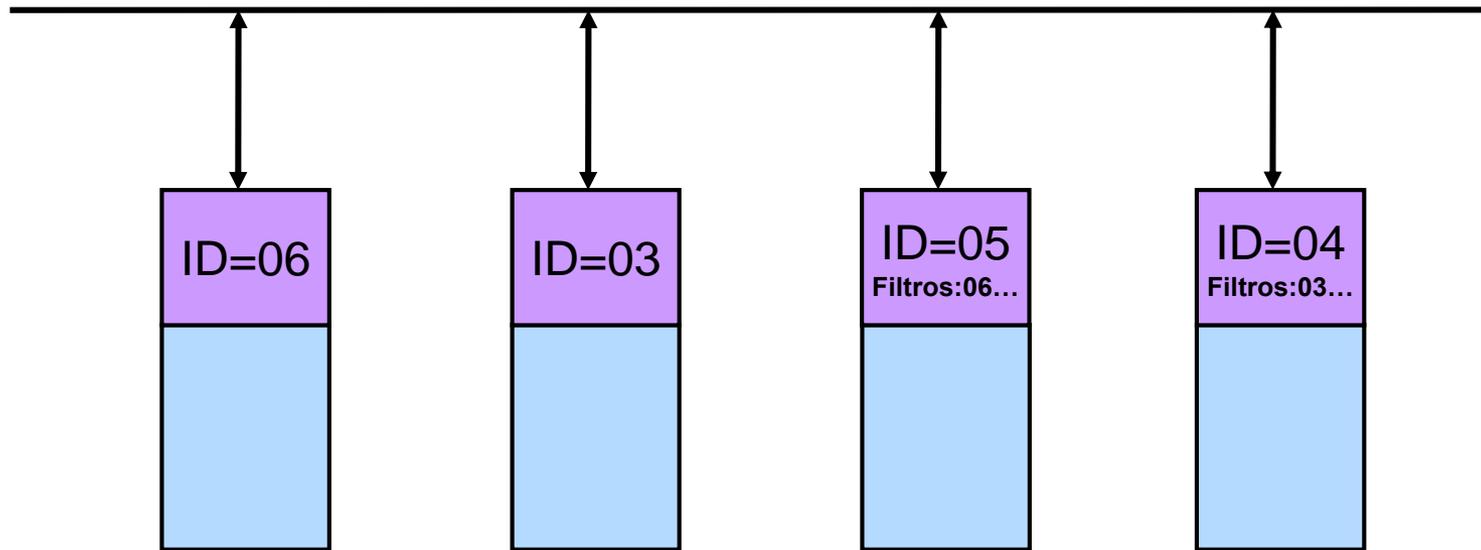
**:[DEST][ORI][FUN][LON][DATOS][verif][delim]**



*CSMA/CD: Carrier Sense Multiple Access – Collision Detection*

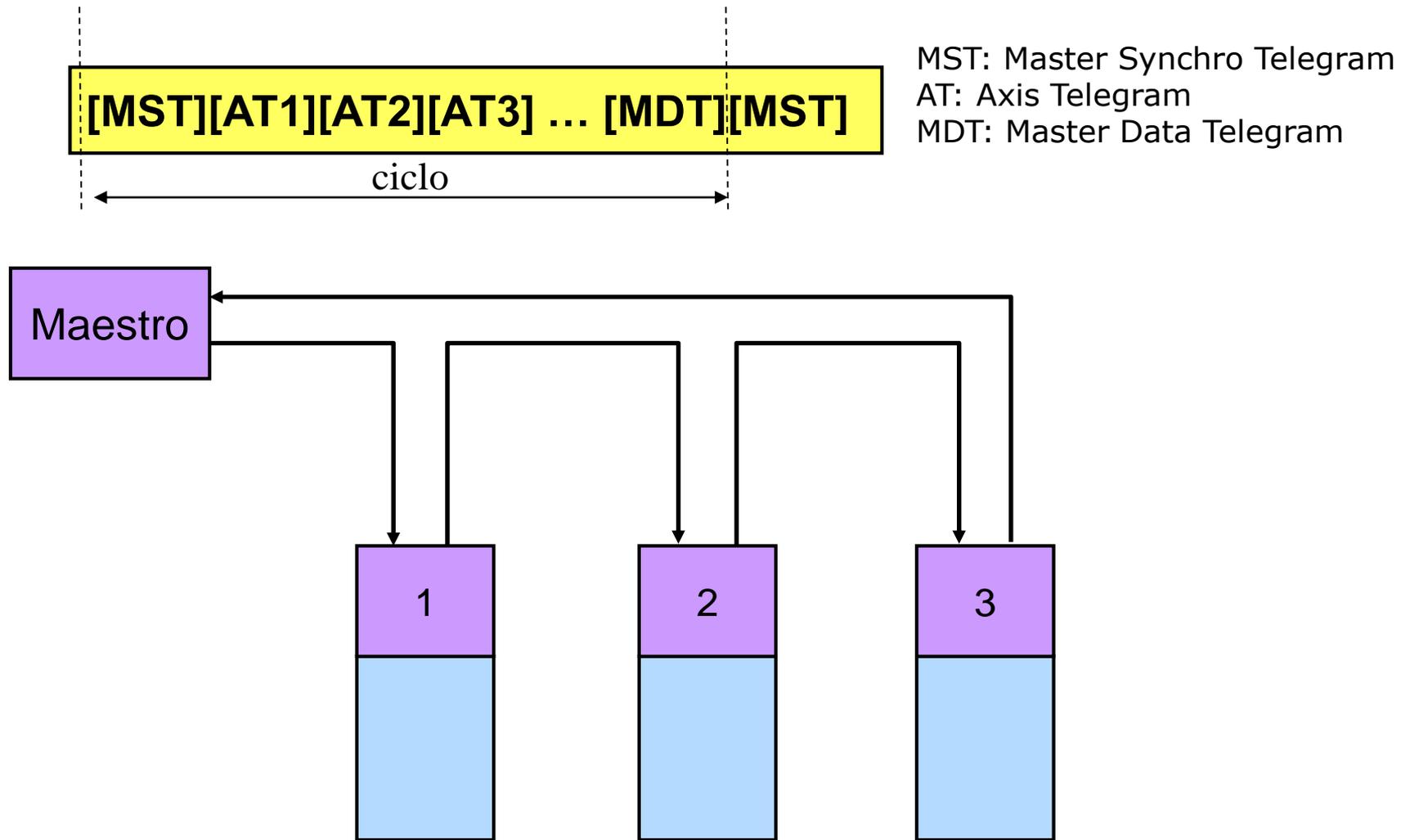
# CSMA/CA (Ej. CAN)

**:[FUN/ID][LON][DATOS][CRC][delim]**



*CSMA/CA: Carrier Sense Multiple Access – Collision Avoidance*

# Topología anillo (SERCOS III-EtherCAT)



# Resumen

- Un mecanismo está formado por eslabones y articulaciones. Cada articulación activa (con motor o impulsor) se denomina eje. Un “eje servocontrolado” es un eje controlado en posición, velocidad y/o fuerza/par (torque)
- El control individual de un eje es realizado por un hardware dedicado (microcontrolador, DSP, lógica programable) que incluye puertos de E/S para medir, activar y comunicarse, además de etapas de potencia.
- El *firmware* deberá realizar los lazos de control, implementar los distintos estados y modos de operación que debe tener el sistema lógico, y comunicarse con un sistema supervisor que le enviará consignas, consultará parámetros o cambiará estados y modos.
- Un mecanismo multi-eje tal como un robot, requiere coordinar los accionamientos de cada eje, esto es comunicar consignas de posición/velocidad y controlar sus estados y modos.
- Cuando se requiere un control continuo la coordinación debe efectuarse a alta velocidad. Mientras mayor cantidad de “puntos por segundo” se puedan solicitar, más precisión y suavidad en el movimiento se puede lograr. Esto requiere sistemas de comunicación eficientes y de “tiempo real” (bajo *jitter*).