

Arquitecturas de Control para Robots

CONTROL Y PROGRAMACIÓN DE ROBOTS

Grado en Electrónica, Robótica y Mecatrónica



Índice

1. Introducción
2. Arquitectura funcional
3. Entorno de programación
4. Arquitectura hardware
5. Ejemplos

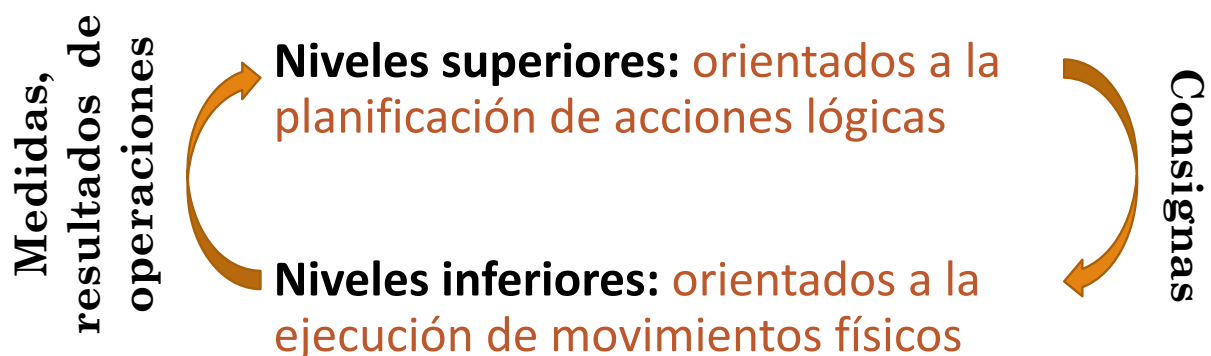
Introducción

Objetivos:

- Organización y gestión de la información dentro de la controladora de un robot.
- Incluye el sistema operativo del robot

Arquitectura funcional

Concebida para la **gestión de tareas a varios niveles** de actividades ordenados en una **estructura jerárquica**.



Arquitectura funcional

Herramientas del sistema de control para supervisar las actividades de un sistema robótico:

- Capacidad de **manipulación**
- Capacidad **sensorial**
- Capacidad de **decisión**
- Capacidad de **procesamiento de datos**

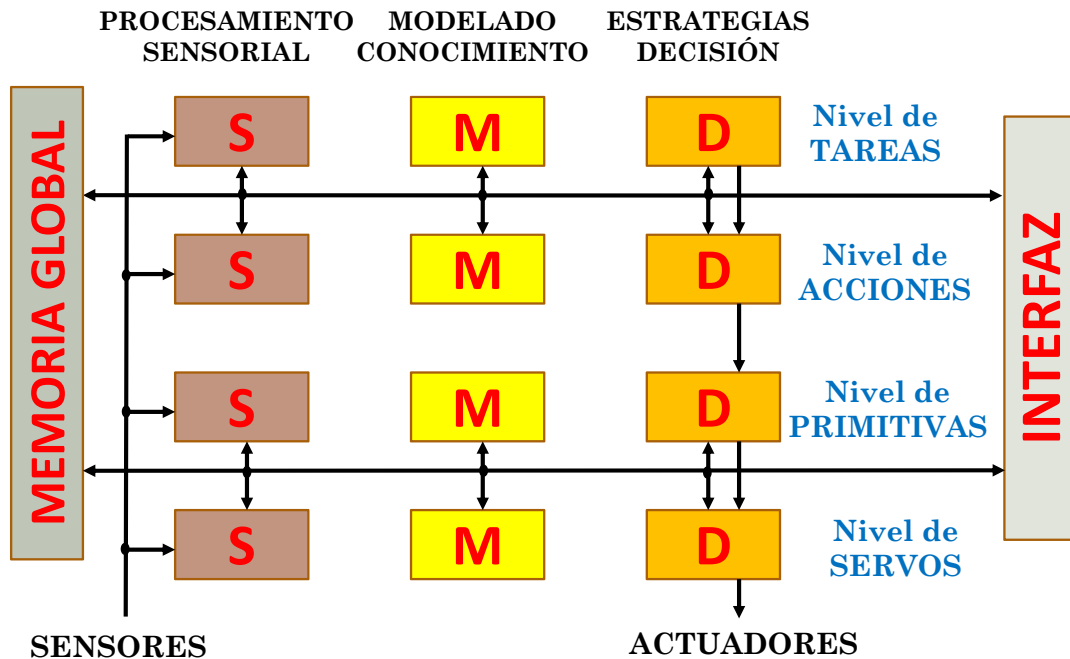
Arquitectura funcional

Módulos funcionales genéricos:

- **Módulos de Procesamiento sensorial:** adquieren, elaboran, procesan e integran datos en el tiempo y el espacio para identificar y medir el estado del sistema y las características del entorno.
- **Módulos de Modelado:** contienen modelos deducidos del conocimiento previo del sistema y del entorno.
- **Módulos de Decisión:** descomponen tareas de alto nivel en acciones de bajo nivel, incluyendo descomposición temporal (acciones secuenciales) y espacial (acciones concurrentes).

Arquitectura de control

MÓDULOS FUNCIONALES



Arquitectura funcional

Niveles jerárquicos:

- **Nivel de Tareas:** analiza y descompone la tarea (de alto nivel de abstracción) requerida por el usuario en una secuencia de acciones coordinadas en el espacio y en el tiempo.
- **Nivel de Acciones:** recibe comandos del nivel de tareas y caracterizan los caminos/trayectorias de movimiento (de alto nivel) para cada acción básica.
- **Nivel de Primitivas:** calcula trayectorias admisibles y decide la estrategia/parámetros de control adecuada, generando referencias para los servos.
- **Nivel de Servos:** implementan los algoritmos de control, generando las señales de control a los servomotores.

Entorno de programación

- Proporciona al operador la **interacción con el robot**, permitiendo indicar al robot las tareas que debe ejecutar.
- Debe **traducir las ordenes** de un lenguaje de programación.
- Debe **comprobar la correcta ejecución de las tareas** que deben ser realizadas, lo que implica la observación de que el programa produce los efectos deseados en el mundo real.
- Complejidad en función del tipo de programación del robot.

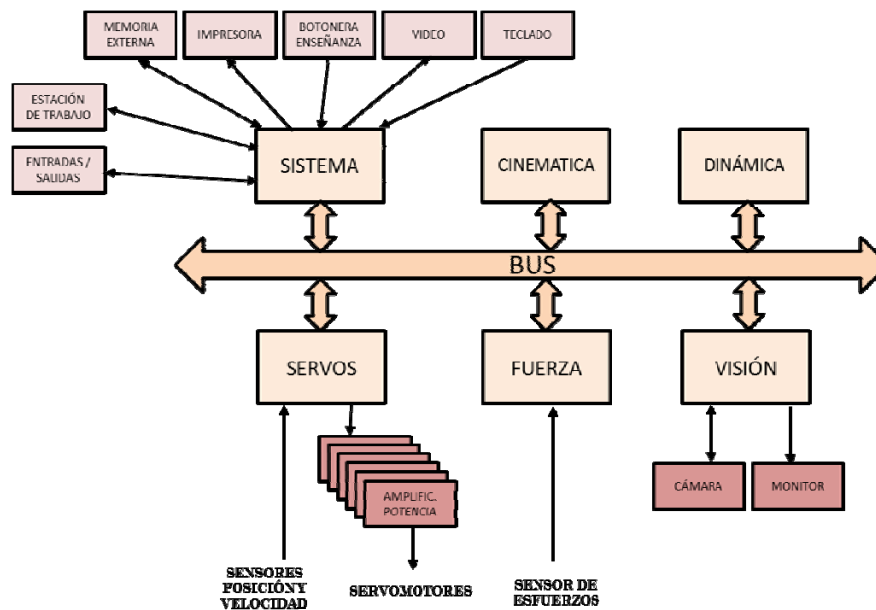
Entorno de programación

Características principales:

- Sistema operativo en tiempo real.
- Modelado del mundo.
- Control de movimiento.
- Lectura de datos del sistema sensorial.
- Interacción con el sistema físico.
- Capacidad de detección de errores.
- Recuperación de funciones operativas correctas.
- Estructura de lenguaje específica.

Arquitectura hardware

• Modelo general del sistema de control de un robot industrial



Arquitectura hardware

Tarjeta SISTEMA:

- Microprocesador con coprocesador matemático.
- Memoria EPROM para el arranque del sistema.
- Memoria RAM local.
- Memoria RAM compartida (a través del BUS).
- Puertos series y/o paralelos.
- Redes de comunicaciones (Ethernet, ...).
- Contadores, registros y temporizadores.
- Sistema de gestión de interrupciones.

Arquitectura hardware

Tarjeta CINEMÁTICA:

- Cálculo de primitivas de movimientos.
- Cálculo de cinemática directa, inversa y Jacobiana.
- Chequeo de la viabilidad de las trayectorias.
- Manejo de la redundancia cinemática.

Tarjeta DINÁMICA:

- Cálculo de dinámica inversa.

Arquitectura hardware

Tarjeta SERVOS:

- Interpolación de referencias de trayectorias.
- Cómputo del algoritmo de control.
- Conversión digital/analógica e interface con amplificadores de potencia.
- Interrupción de movimiento en caso de fallo de funcionamiento.

Tarjeta FUERZA:

- Acondicionamiento de señal del sensor de esfuerzos.
- Representación de las fuerzas en un sistema de coordenadas determinado.

Tarjeta FUERZA:

- Procesamiento de datos provenientes de la cámara.
- Extracción de las características geométricas de la escena.
- Localización de objetos en un sistema de coordenadas determinado.

Ejemplos

Arquitectura del PUMA 560

Unimation Mark II controller:

- DEC LSI-11 computer with ADAC parallel interface board, DLV11-J serial interface board, CMOS board, and EPROM board.
- Servo interface board
- Six digital servo boards.
- Two power amplifier assemblies
- Power amplifier control board
- Clock/terminator board
- Input/output interface board
- Two power supplies
- High power function board
- Arm cable board.

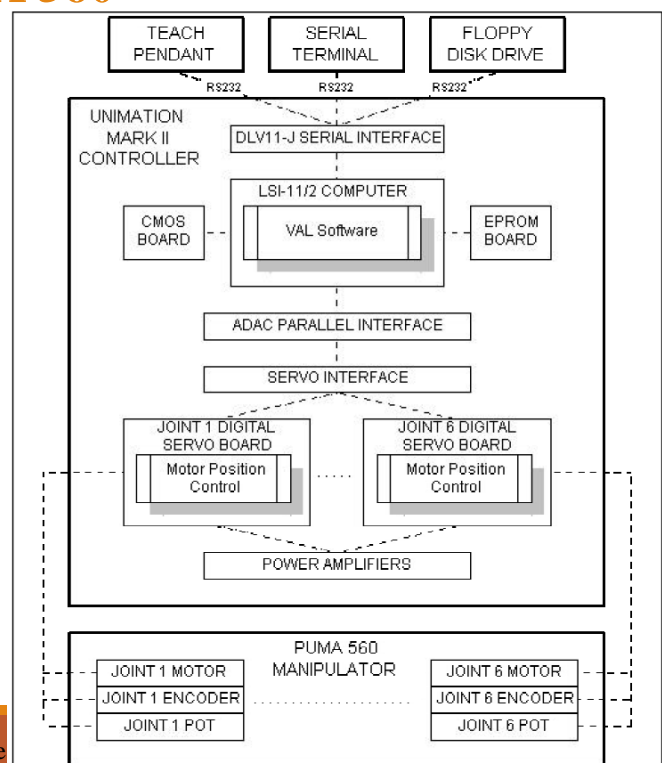
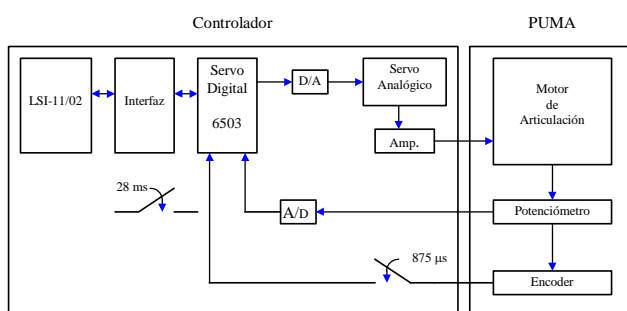
Control y Programación de Robots. GIERM

Ejemplos

Arquitectura del PUMA 560

- Esquema del controlador Unimation Mark II:
- Software: VAL-II

Detalle del servocontrolador:

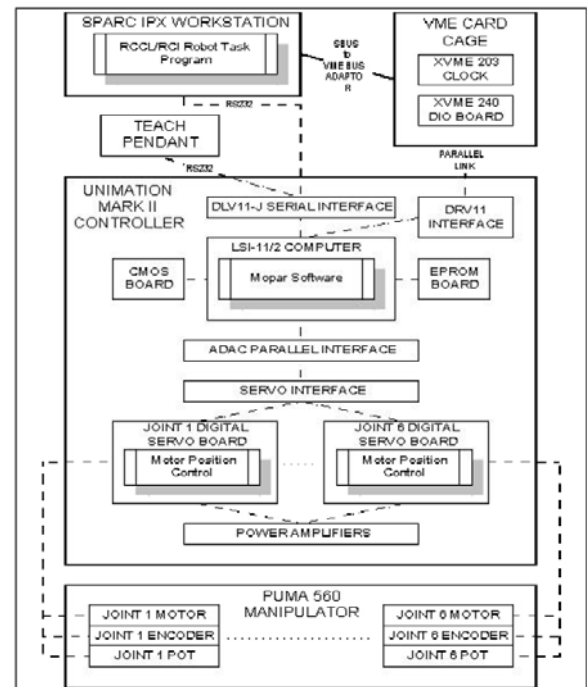


Control y Programación de

Ejemplos

Arquitectura del PUMA 560

- RCCL/RCI System
(Robot Control C Library /
Real Time Control Interface)
- Sustitución de floppy por Sparc
IPX (estación de trabajo Unix):
 - Mejora de software
 - Simulador 3D del brazo del robot
 - ...
- Software:
 - RCCL: Nivel de planificación de tareas
 - RCI: nivel de control de tareas en tiempo real

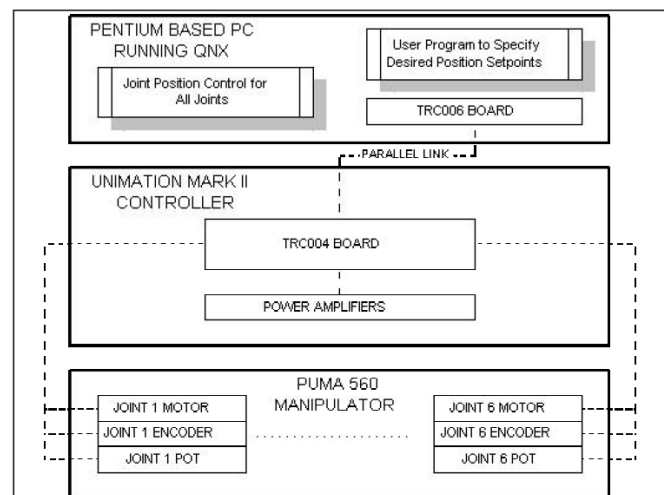


Control y Programación de Robots. GIERM

Ejemplos

Arquitectura del PUMA 560

- **TRC Board based Controller:**
(TRC: tarjetas controladoras de
servos de propósito general)
- Sustitución completa del
hardware inicial del controlador
Mark II
- Control desde PC con QNX:

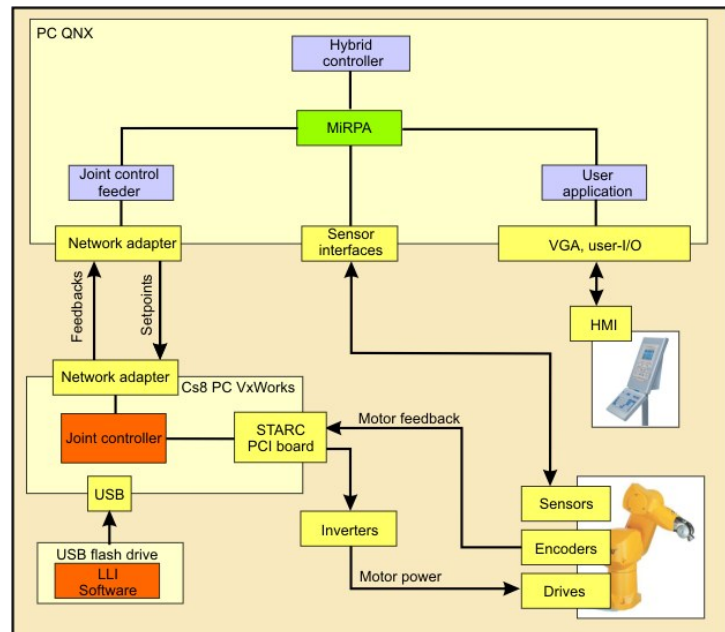
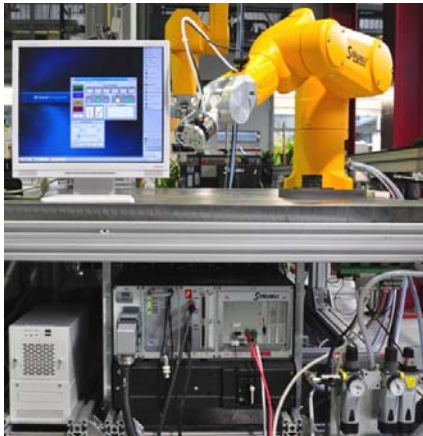


Control y Programación de Robots. GIERM

Ejemplos

Arquitectura abierta de robots de la serie TX de Stäubli

- Controlador CS8
- Low Level Interface (LLI) via USB



Institute for Robotics and Process Control

Control y Programación de Robots. GIERM

Ejemplos

Arquitectura original del robot RM-10 de Stäubli

Robot industrial de 6 grados de libertad, con capacidad de carga de 50kg y precisión de 0.2 mm.

Accionamiento indirecto.

Motores de corriente continua sin escobillas, con sensores inductivos de dos polos.

Controlador original basado en bus VME con tarjetas controladoras independientes para cada motor.



Control y Programación de Robots. GIERM

Ejemplo

Arquitectura modificada del robot RM-10 de Stäubli

(Grupo Ingeniería de Automatización, Control y Robótica. Universidad de Sevilla)

Objetivos:

- Desarrollar control basado en tarjeta DSP controladora instalada en PC y programable desde el mismo.
- Mantener la posibilidad de recurrir al controlador original.
- Establecer interfaz hardware PC-Robot.
- Desarrollar el entorno software.



Control y Programación de Robots. GIERM

Ejemplo

Arquitectura modificada del robot RM-10 de Stäubli

(Grupo Ingeniería de Automatización, Control y Robótica. Universidad de Sevilla)

Equipamiento:

- **Ordenador personal:** procesador Pentium II 350 MHz, memoria RAM 256 Mb., Disco duro 20 Gb., tarjeta DSP instalada.
- **Tarjeta dSpace DS1103:** Procesador IBM PowerPC a 333 MHz, DSP esclavo con funciones de entrada y salida, y entradas y salidas digitales y analógicas.



Control y Programación de Robots. GIERM

Ejemplo

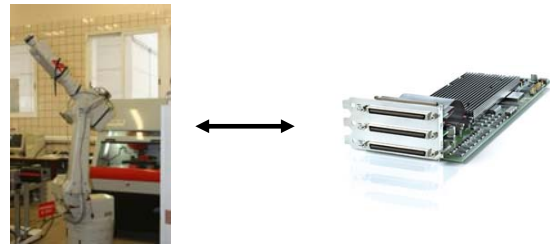
Arquitectura modificada del robot RM-10 de Stäubli

(Grupo Ingeniería de Automatización, Control y Robótica. Universidad de Sevilla)

Desarrollo de hardware:

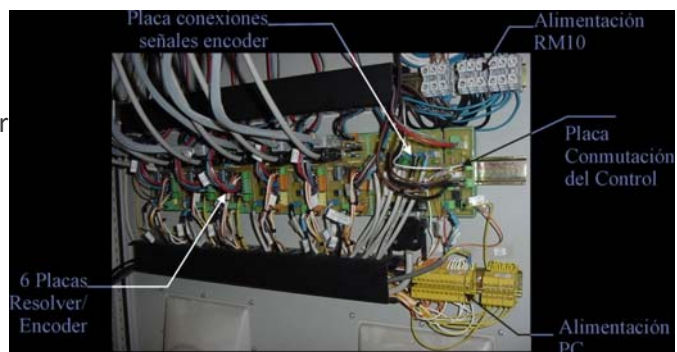
•Finalidad:

- Establecer la interfaz RM-10 - Tarjeta DSP/PC



•Tarjetas electrónicas:

- Tarjeta de conmutación de control.
- Placa de entradas/salidas digitales.
- 6 tarjetas de conversión Resolver/Encoder aislamiento de señales analógicas.



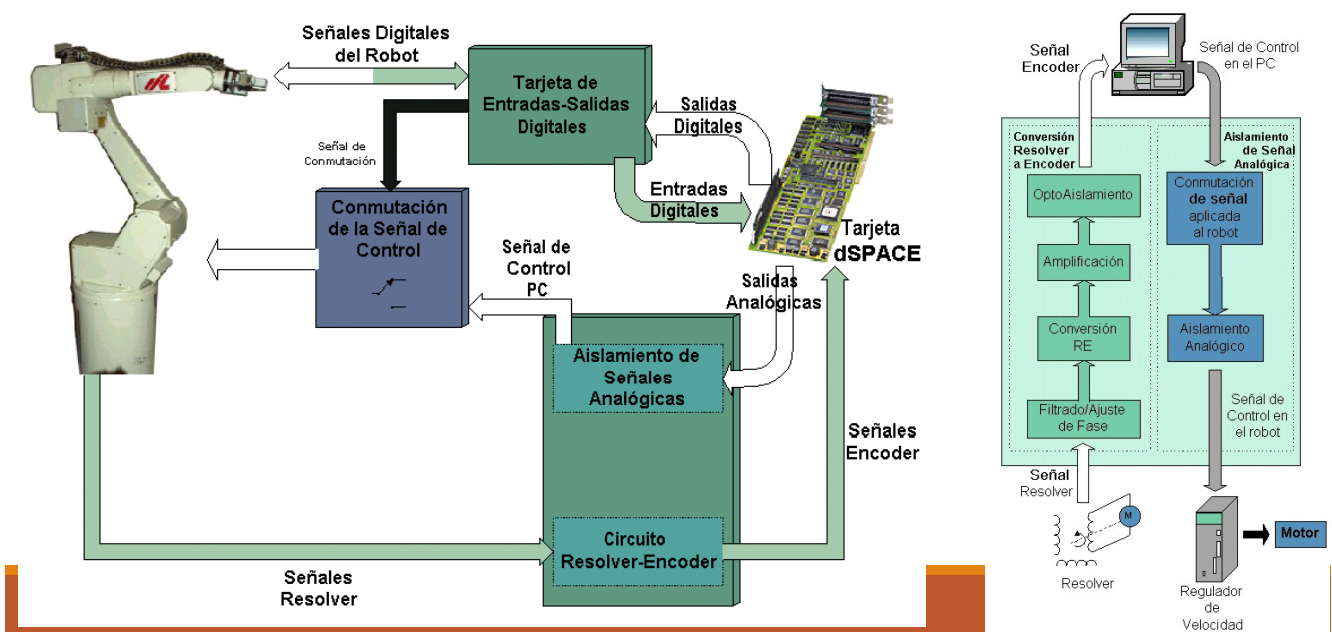
Control y Programación de Robots. GIERM

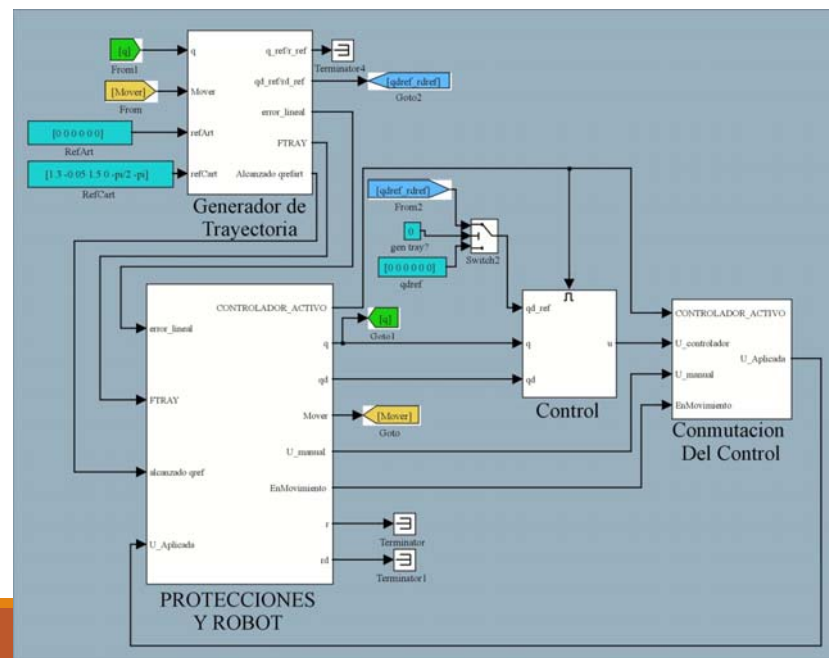
Ejemplo

Arquitectura modificada del robot RM-10 de Stäubli

(Grupo Ingeniería de Automatización, Control y Robótica. Universidad de Sevilla)

Desarrollo de hardware:





Ejemplo

Arquitectura modificada del robot RM-10 de Stäubli

(Grupo Ingeniería de Automatización, Control y Robótica. Universidad de Sevilla)

Entorno de Software:

• Máquina de estados:

- **Encendido:** Estado de reposo. Frenos activos y control deshabilitado.
- **Operación:** Robot activo. Frenos activos y control a la espera.
- **Movimiento Manual:** Velocidad constante en cada articulación.
- **Movimiento Trayectoria:** Generación de trayectorias en el espacio articular (polinomios de 5º orden) y cartesiano (lineales).
- **Movimiento Frenado:** Parada tras fin movimiento o emergencias.
- **Espera Frenos:** Control mientras se aguarda a la entrada efectiva de frenos

Control y Programación de Robots. GIERM

Ejemplo

Arquitectura modificada del robot RM-10 de Stäubli

(Grupo Ingeniería de Automatización, Control y Robótica. Universidad de Sevilla)

Entorno de Software:

Supervisión de alarmas:

- Límite de Ángulo en articulaciones.
- Límite Espacio de Trabajo.
- Límite Velocidad.
- Sin Movimiento.
- Sin Control PC.
- Frenos Activados.
- ...

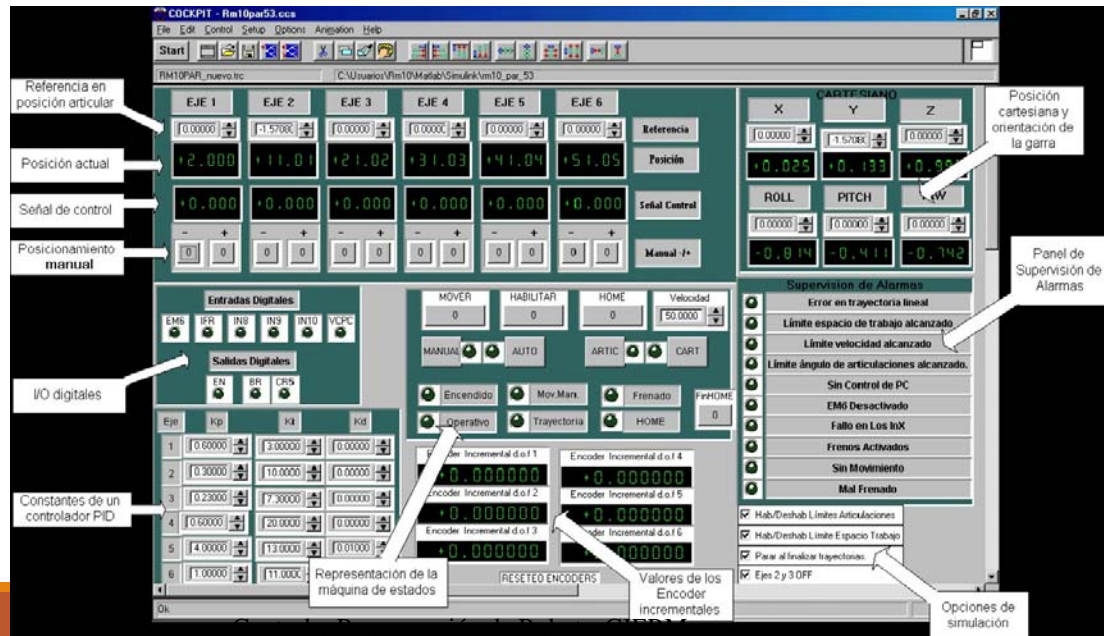
Control y Programación de Robots. GIERM

Ejemplo

Arquitectura modificada del robot RM-10 de Stäubli

(Grupo Ingeniería de Automatización, Control y Robótica. Universidad de Sevilla)

Cockpit:

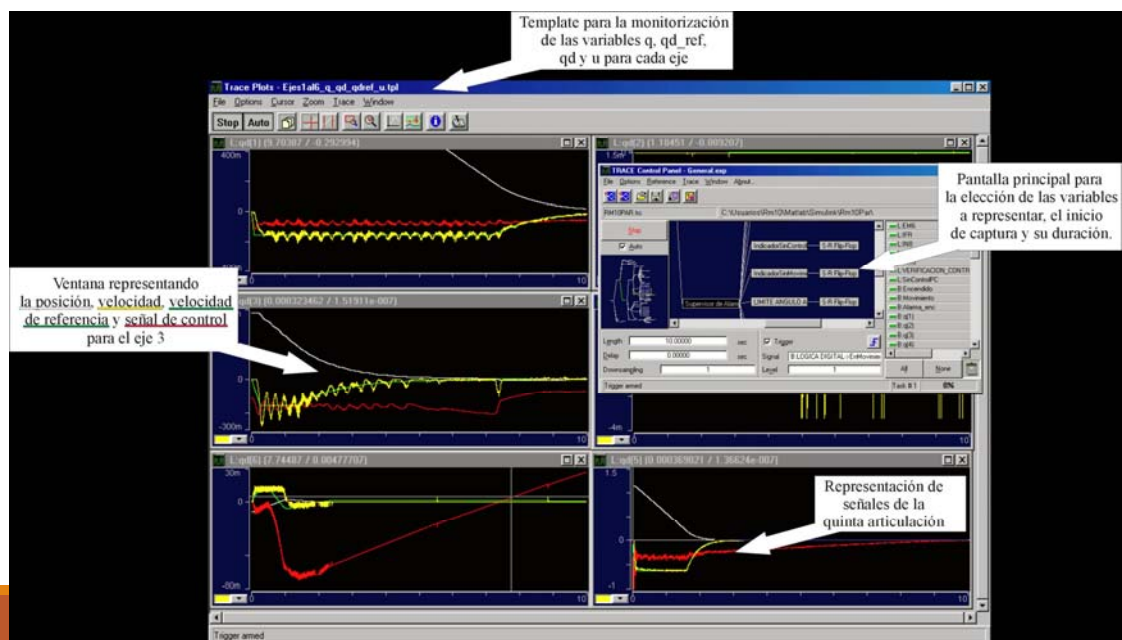


Ejemplo

Arquitectura modificada del robot RM-10 de Stäubli

(Grupo Ingeniería de Automatización, Control y Robótica. Universidad de Sevilla)

Trace:

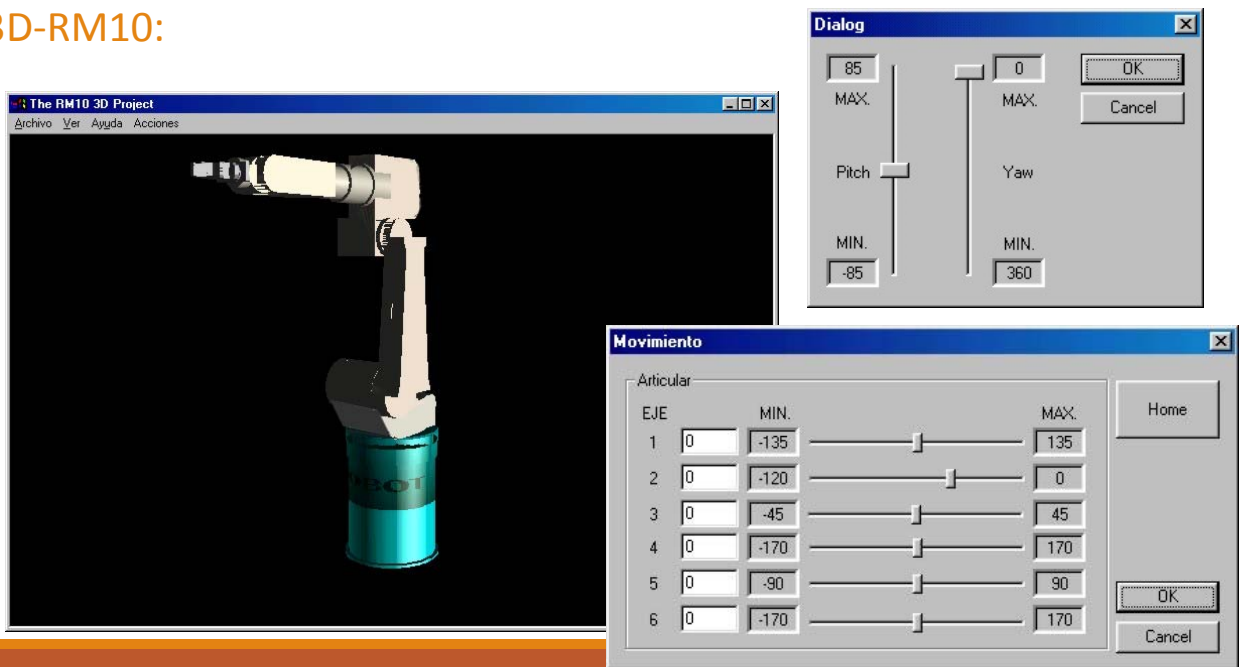


Ejemplo

Arquitectura modificada del robot RM-10 de Stäubli

(Grupo Ingeniería de Automatización, Control y Robótica. Universidad de Sevilla)

3D-RM10:



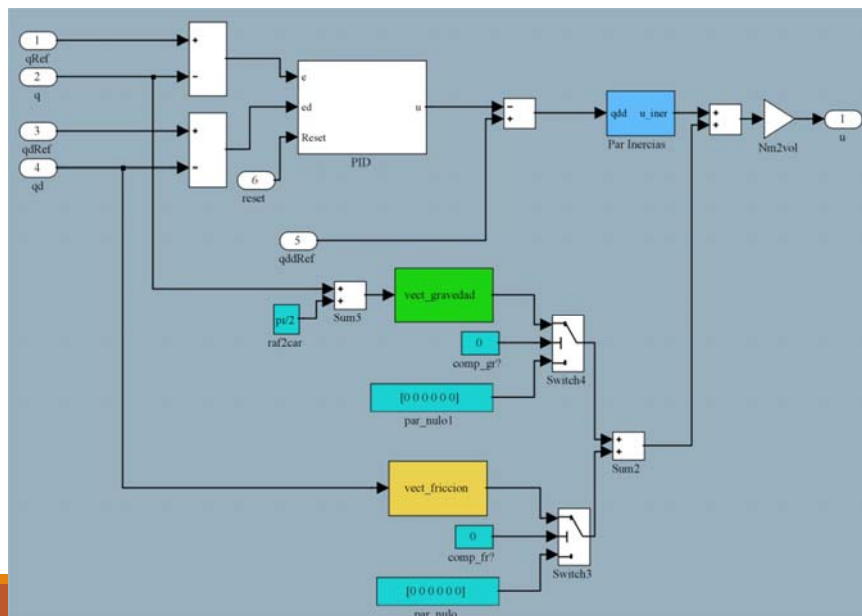
Control y Programación de Robots. GIERM

Ejemplo

Arquitectura modificada del robot RM-10 de Stäubli

(Grupo Ingeniería de Automatización, Control y Robótica. Universidad de Sevilla)

Desarrollo de controladores dinámicos:



Control y Programación de Robots. GIERM