

Definición de S. Tiempo Real

- Un Sistema de Tiempo Real se define como:
 - “Un sistema que debe responder ante estímulos generados por el entorno dentro de un periodo de tiempo finito especificado”
- Aspectos fundamentales:
 - Son sistemas que interactúan con el entorno (mundo físico), adquiriendo estímulos y estados del entorno y generando una acción sobre dicho entorno.
 - Son sistemas en los que el tiempo de respuesta es un aspecto crítico.
- Corrección:
 - Lógica
 - Temporal

Tiempo Real / Rapidez

- Todo sistema de tiempo real debe ser suficientemente rápido para cumplir los requerimientos, pero el que un sistema sea rápido no implica que éste sea de tiempo real
- La rapidez de un sistema de tiempo real está fijada por la naturaleza de los estímulos con los que tiene que tratar, la cual puede variar de unas aplicaciones a otras.
- Tiempo real significa trabajar con aspectos del sistema como distribución física de los componentes, manejo de estímulos impredecibles, fallos en los componentes, incertidumbre acerca del entorno, y el efectos de estos aspectos sobre el rendimiento y robustez del sistema.

Sist. Tiempo Real / Sist. Empotrados

- Entorno de un SITR:
 - **Entorno activo** que comprende los operadores que intervienen en el comportamiento del sistema modificando el modo de operación.
 - **Entorno pasivo** que comprende a los usuarios que están solamente interesados en el servicio proporcionado y al ambiente exterior que no tiene nada que ver con la existencia del sistema.
- Sistema empotrado (“embedded system”)
 - Únicamente tiene un entorno pasivo
 - Es un componente autónomo que forman parte de un sistema mayor
 - Ejemplos:
 - Disco duro, Lavadora
 - Automóvil, Sonda espacial, Equipos médicos

STR Estrictos/ No Estrictos

- Corrección:
 - Lógica – Temporal
- **STR estrictos (críticos): la corrección temporal es crítica.**
 - El tiempo de respuesta es muy importante y no puede ser sacrificado por una mejora en otros aspectos.
 - En ciertos sistemas (p.e. sistemas de seguridad críticos) la corrección temporal es tan importante que el criterio de corrección lógica puede ser relajado en aras de alcanzar un tiempo de respuesta determinado
- **STR no estrictos (acríticos): la corrección temporal no es crítica.**
 - Fallos ocasionales en generar un resultado dentro del tiempo fijado no produce consecuencias serias en el funcionamiento general del sistema.
 - Las tareas de tiempo real no estrictas son ejecutadas tan rápido como es posible, pero no están forzadas por tiempos límite absolutos, pudiendo ser sacrificada la corrección temporal bajo ciertas circunstancias

SITR: Introducción

5

STR Estrictos/ No Estrictos

- Un STR puede estar formado por tareas críticas y acríicas
- Una misma tarea puede tener tiempos límite estrictos y no estrictos.
 - La respuesta a un evento de peligro puede tener un tiempo límite no estricto de 50 ms (para una reacción con eficiencia óptima) y un tiempo límite estricto de 200 ms (para garantizar que no se produce un daño en el equipo o en las personas). Entre estos dos límites, el valor o utilidad de la salida decrece según aumenta el tiempo.
- Tipos de relajación en la corrección temporal:
 - Si no se cumple el tiempo límite el resultado no tiene ningún valor (pérdidas ocasionales de respuestas a eventos)
 - Que el servicio llegue tarde ocasionalmente pero éste sea válido dentro de un cierto intervalo de tiempo.

SITR: Introducción

6

Tolerancia a Fallos

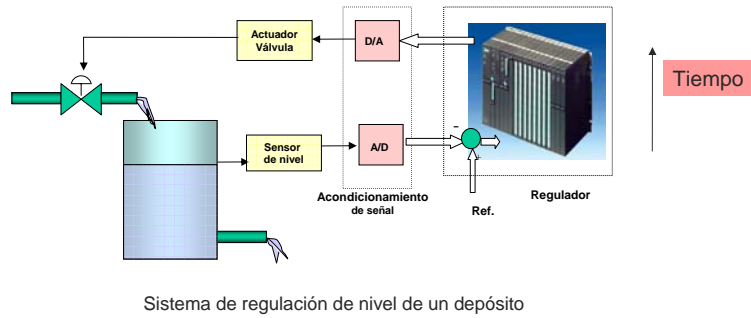
- Corrección:
 - Lógica – Temporal
- Los sistemas pueden fallar, es decir exhibir un comportamiento para el cual no fueron diseñados.
 - Ya que los fallos son inevitables, un sistema de tiempo real debe ser robusto frente a ellos, comportamiento denominado ‘tolerancia a fallos’.
- Tipos de Sistemas:
 - **Sistemas con parada segura** (“fail-safe”): en caso de fallo colocan al sistema en un estado seguro (sistemas críticos)
 - **Sistemas con degradación aceptable** (“fail-soft”): presentan una pérdida parcial de funcionalidad o prestaciones en caso de fallo, pudiendo seguir funcionando a pesar de dicha merma.

STR Eventos / Tiempo

- **Arquitectura controlada por eventos o interrupciones** (“*event-driven*” o “*interrupt-driven*”).
 - Establece la ejecución de un componente o tarea basándose en la aparición de una interrupción o señal generada por un evento externo
 - Constituyen un mecanismo mucho más eficaz para responder ante eventos externos no regulares.
- **Arquitectura controlada por tiempo** (“*time-driven*”).
 - Opera de acuerdo a los ciclos del reloj o relojes del sistema.
 - Este tipo de sistemas se utiliza cuando es preciso la ejecución de tareas periódicas o la ejecución de tareas mediante temporizadores.
 - Son más sencillos y robustos (interfaces entre subsistemas muy controlados y con un funcionamiento independiente basado en relojes locales)

Ejemplos de STR

■ Control de Procesos (1)

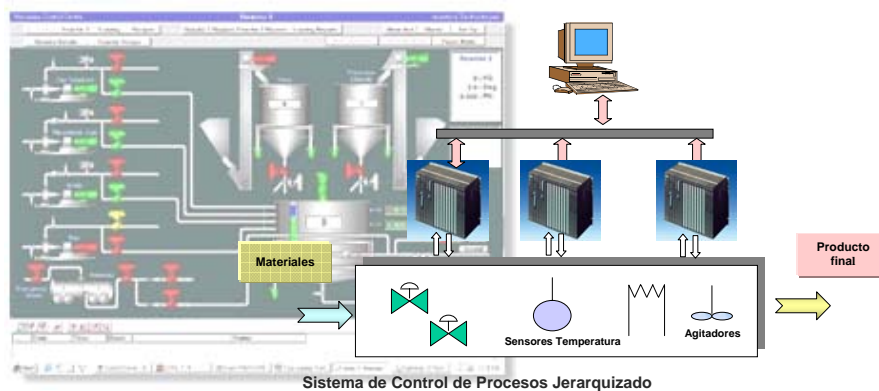


SITR: Introducción

9

Ejemplos de STR

■ Control de Procesos (2)



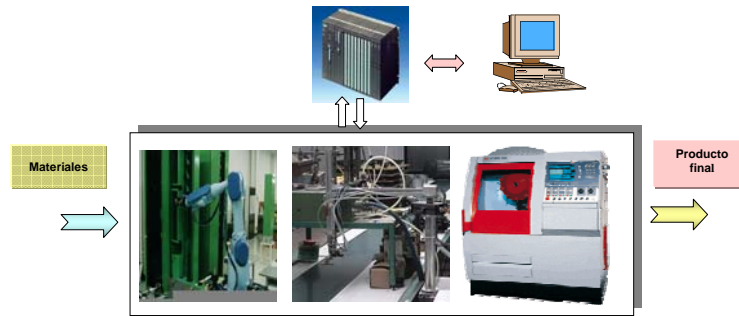
SITR: Introducción

10

Ejemplos de STR

■ Fabricación:

- Integración del proceso de fabricación (CIM)



Sistema de control de producción.

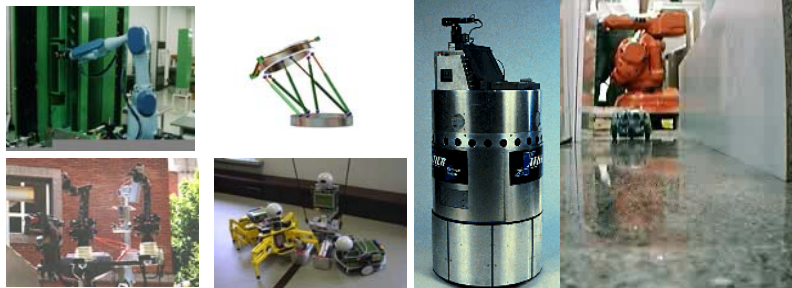
SITR: Introducción

11

Ejemplos de STR

■ Robótica

- Control dinámico
- Planificación de trayectorias



Aplicaciones de control de robots

SITR: Introducción

12

Ejemplos de STR

- **Automoción:**
 - Control de encendido
 - Control de inyección
 - Antibloqueo de frenos ABS
 - Control de tracción ASR
 - Airbag
 - Suspensión activa



Características de un STR

- Grandes y complejos.
- Manipulación de magnitudes físicas.
- Extremadamente fiable y seguro.
- Herramientas para el manejo de tiempo real.
- Implementación eficiente.
- **Concurrencia.**

Concurrencia

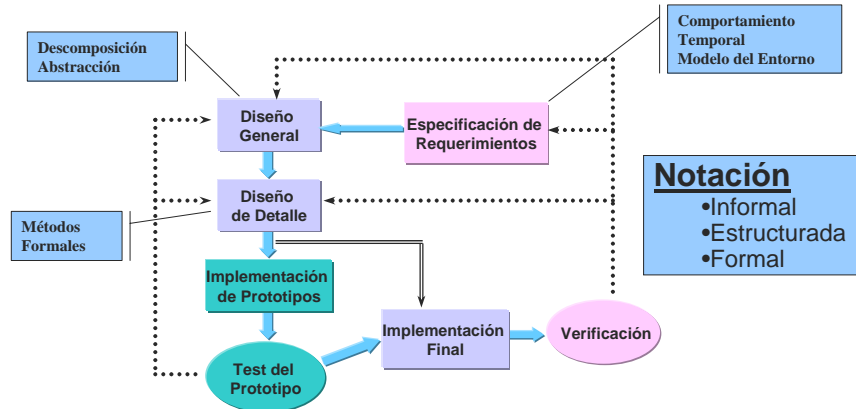
- La **concurrencia** se define como la capacidad de un sistema de mantener varias tareas en estados intermedios de su ejecución
 - Sistemas de tiempo compartido
 - Sistemas multiprocesadores
 - Sistemas distribuidos
- Expresión de la concurrencia en un programa:
 - El programador
 - El lenguaje de programación (ADA95, Occam, Pearl...)
 - El sistema operativo (POSIX)

Herramientas de Tiempo Real

- Los sistemas se diseñan en con unos márgenes de capacidad sobrantes, de forma que el comportamiento en "el peor caso" no produzca retardos en los periodos críticos de operación del sistema.
- Soporte que permita:
 - Especificar el tiempo en el cual debe ejecutarse cada tarea
 - Especificar el tiempo en el cual debe se completada cada tarea
 - Responder a situaciones en las cuales no se pueden cumplir todos los requerimientos temporales
 - Responder a situaciones en las cuales los requerimientos temporales puedan variar dinámicamente

Diseño de STR

■ Metodología "top-down"



SITR: Introducción

17

Lenguajes de programación

- Lenguaje ensamblador:
 - Muy Eficiente
 - Difícil de implementar la concurrencia.
- Lenguajes secuenciales de alto nivel
 - C, C++, Java, Pascal, Fortran, Jovial, Coral, RTL/2
 - Portables, y fáciles de programar
 - No incorporan soporte para la concurrencia (debe ser soportado por el S.O.)
- Lenguajes concurrentes de alto nivel
 - ADA95, Modula-2, PEARL, Mesa, CHILL
 - Soporte de la concurrencia y especificaciones temporales en el propio lenguaje

SITR: Introducción

18