

Real-Time Systems: Examples / Case Studies

- Simple Control System
- Sampling Periods
- Quality of the Control vs. Processing Cost
- Protection of Resources in Integrated Systems
- Multimedia / Real-Time Communication
- Anomalies in Asynchronous Systems
 - Example: Advanced Fighter Technology Integration (AFTI) F16
- Priority Inversion

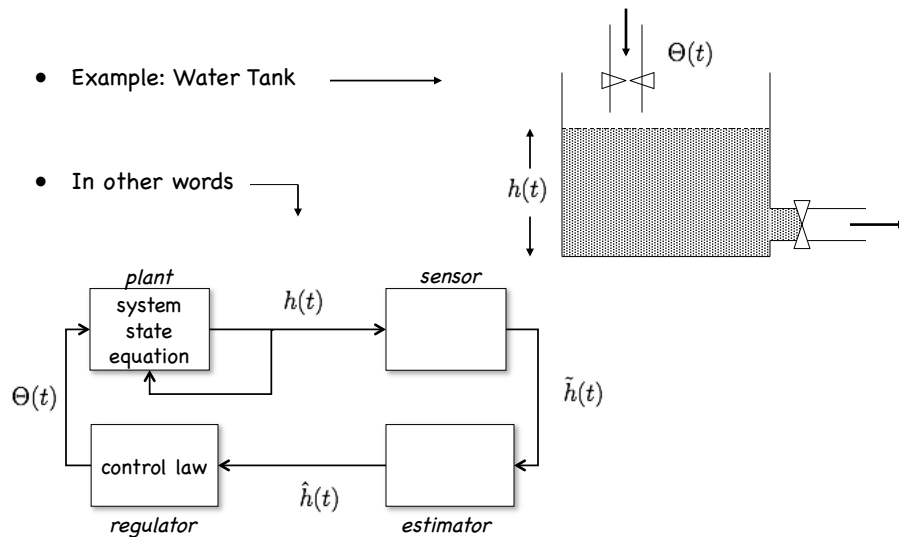
- Real-Time Systems
- Hard and soft deadlines; operational definition

© R. Bettati

Application Areas: Control Systems

- Example: Water Tank →

- In other words ↘



© R. Bettati

Control Systems (cont)

- Control Loop:

```

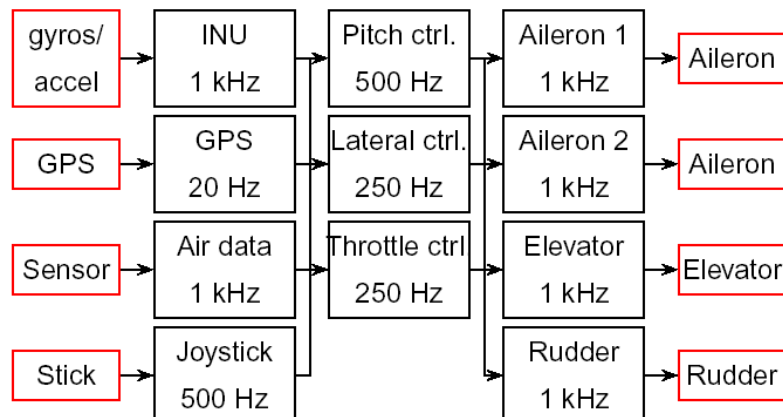
DO FOREVER
  wait_for_delay
  h      := fluid_height
  theta := valve_position
  r      := table_lookup(h, theta)
  IF     r = left THEN turn_left
  ELSE IF r = right THEN turn_right
  ELSE   do_nothing
ENDDO

```

© R. Bettati

Example: Avionics System

Hard real-time system with multirate behavior



© R. Bettati

Quality of Control vs. Processing Cost

Example: Open-Loop Temperature Control

[Simplified from : Setol, Lehoczky, Sha, and Shin, "On Task Schedulability in Real-Time Control Systems", Proceeding of the 1996 IEEE Real-Time Systems Symposium]

- System: Temperature of a unit is controlled by a burner.
- **Dynamic equation:** $\dot{x} = -ax + bu$
 - x difference between unit and ambient temperature, $x(0) = 0$
 - u control input, rate of heat
- **Control Problem:** change temperature of unit to x_d within time t_f ; consume minimum amount of fuel. Allow for a tolerance δ .
 $|x(t_f) - x_d| \leq \delta$
- **Performance Index** $J(u)$ of control system: measure of total cost of control and accuracy generated in time period $[0, t_f]$ by control u . Generally:

$$J(u) = S(x(t_f), t_f) + \int_0^{t_f} L(x(t), u(t), t) dt$$

- **Optimal control** $u^*(t)$ with performance index J^* .

© R. Bettati

Open-Loop Temperature Control (cont)

- Our case: minimize fuel.

$$\min_u J = \frac{1}{2} p (x(t_f) - x_d)^2 + \frac{1}{2} \int_0^{t_f} u^2(t) dt$$

- Resulting **optimal control:**

$$u^*(t) = \frac{x_d p a b e^{at}}{a e^{at_f} + p b^2 \sinh(at_f)}$$

- Final state:

$$x^*(t_f) = \frac{x_d p b^2 \sinh(at_f)}{a e^{at_f} + p b^2 \sinh(at_f)}$$

© R. Bettati

Open-Loop Temperature Control (cont)

- **Discretize** control input u :

- Sampling period P .

$$\dot{x}^*(t) = -ax^*(t) + bu^*(kP) \quad kP \leq t \leq (k+1)P$$

- Performance index for discrete optimal control:

$$J_D^*(P) \doteq S(x^*(t_f), t_f) + \sum_{k=0}^{n-1} \int_{kP}^{(k+1)P} L(x^*(t), u^*(kP), t) dt$$

- In our case:

$$J_D^*(P) \doteq \frac{1}{2} px_d \left(\frac{1 - e^{-aP}}{1 + e^{-aP}} \right)$$

- Constraints:

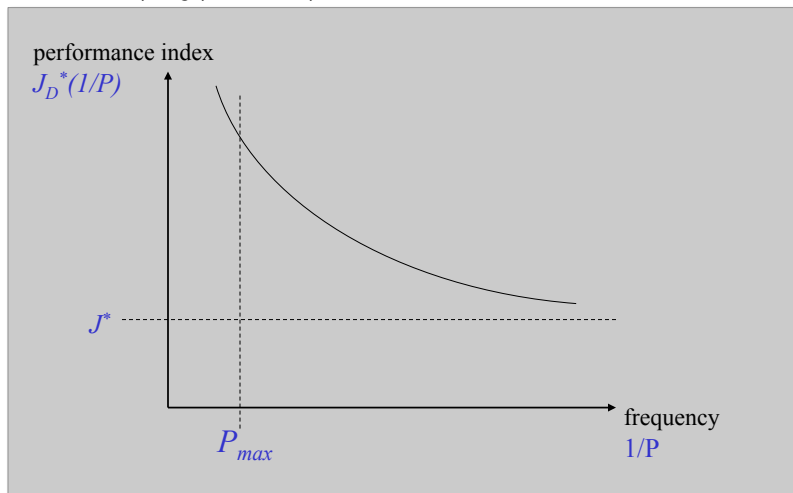
$$|x(t_f) - x_d| \leq \delta$$

$$x_d \left(\frac{1 - e^{-aP}}{1 + e^{-aP}} \right) \leq \delta \quad \Rightarrow \quad P \leq \frac{1}{a} \ln \frac{x_d + \delta}{x_d - \delta}$$

© R. Bettati

Open-Loop Temperature Control (cont)

- Effect of sampling period on performance index.



© R. Bettati

Quality of Control vs. Processing Cost (cont)

- Task frequencies must be determined to optimize the performance indices without overloading the available processing capabilities.
- Notation:

$$\Delta J^*(P) := J_D^*(P) - J^*$$

Optimization problem:

Given a set of tasks, T_1, \dots, T_n , with given $\Delta J_i^*(\bullet)$ and execution times C_i , find a set of periods P_i , such that

1. $P_i \leq P_i^{max}$ // Maintain stability

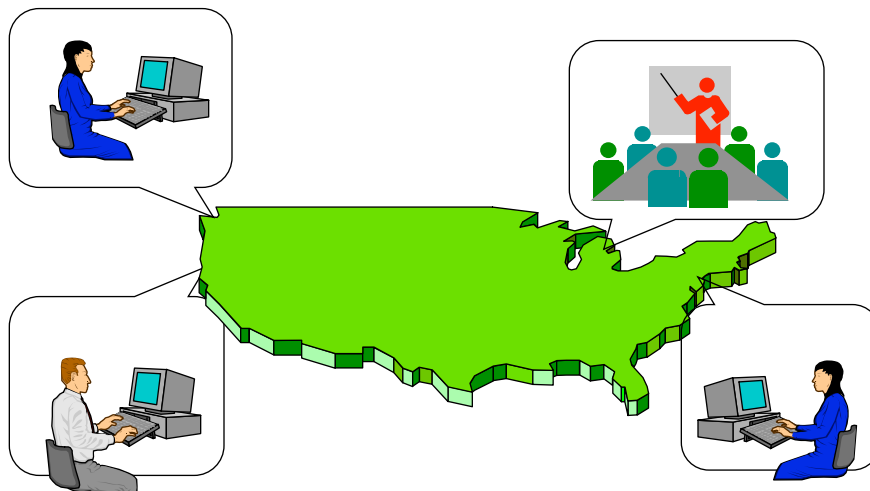
2. Minimize (maximize) $\sum_{i=1}^n \Delta J_i^*(P_i)$

// Optimize total performance index

3. Resource Constraint: $\sum_{i=1}^n \frac{1}{P_i} C_i \leq U$ // CPU capacity

© R. Bettati

Example: Multimedia



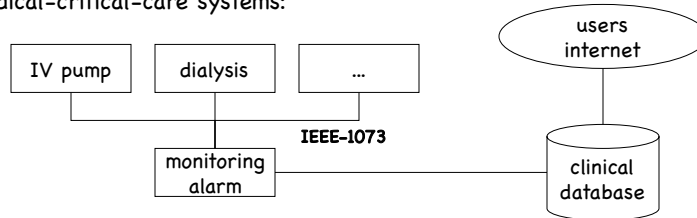
Example: Teleseminars

© R. Bettati

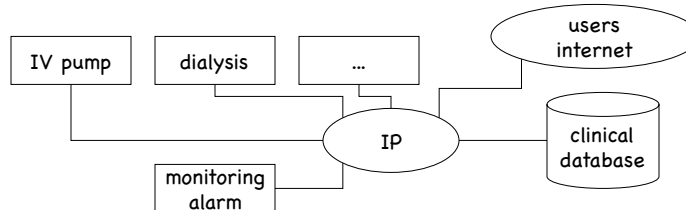
Example: Intensive Care Computing

(Ken Birman, "The Next-Generation Internet: Unsafe at any Speed?", IEEE Computer Aug 2000)

- Medical-critical-care systems:

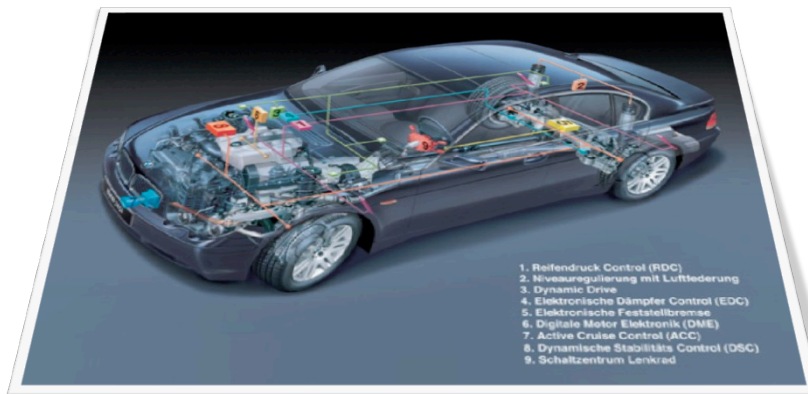


- Medical-critical-care systems over shared network:



© R. Bettati

Example: Cars as Systems-of-Systems



© R. Bettati

Cars as System of Systems (II)

Das Elektronik-Konzept der neuen 7er-Reihe

Bordnetz

Highlights:
iDrive, Dynamic Drive, Elektrische Feststellbremse,

Bussysteme:	4 Hauptbussystem + Subbusse
Diagnose-Bus-Zugang:	BMW-Fast 115kBaud
Anzahl Steuergeräte:	45 – 75 je nach Ausstattung
Verwendete Prozessoren:	im wesentlichen PPC, HC-12, C167
Betriebssystem:	CAN-BMW Standard-Core / MOST Referenzapplikationen ISIS proprietär
Flash-EPROM:	> 100 Mbyte je nach Ausstattung
Anzahl elektrische Motoren:	Bis zu 150
Leitungslänge Kupfer / LWL:	ca. 2,5 km / ca.50 m (ISIS 43m + MOST 7m)
Signale auf dem Bus:	ca. 2500
Anzahl Lieferanten:	mehr als 15 mit jeweils mehreren Standorten
Powermanagement:	Aufstart-/Abschaltverhalten und Ruhestrom

➔ **Herausforderung Integration**

BMW AG Axel Daicke EE-I
FH Deggendorf AK Mechatronik
Seite 34

© R. Bettati

Cars as SoS (III)

Das Elektronik-Konzept der neuen 7er-Reihe

Bordnetz

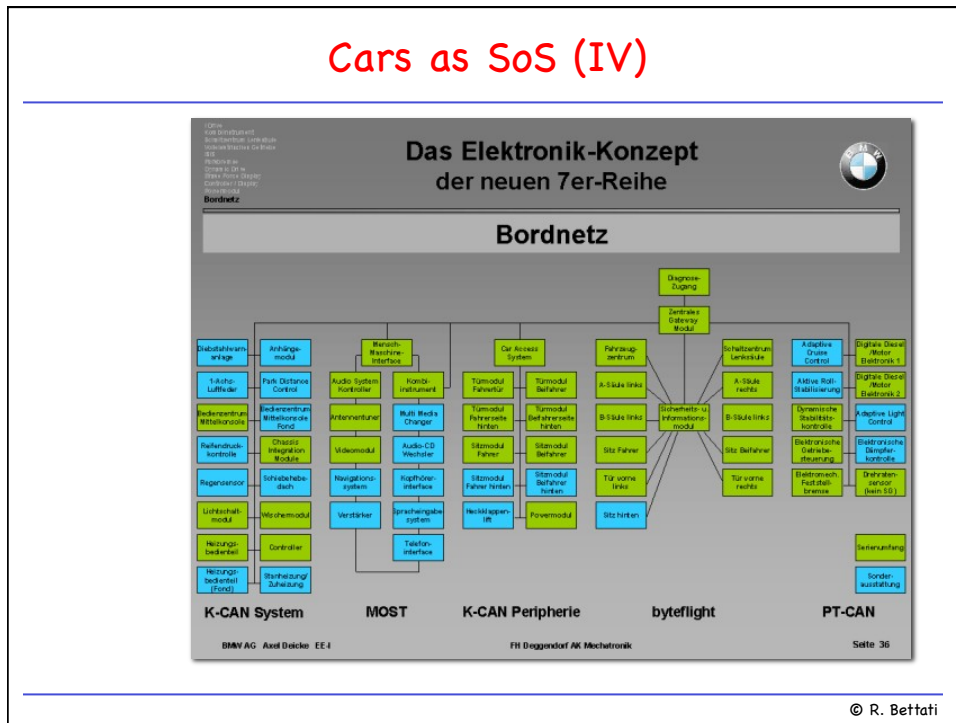
System-CAN Ausstattung	Peripherie-CAN Ausstattung	Powertrain-CAN Antrieb/Fahrwerk	byeflight Sicherheit	MOST Kommunikation
<ul style="list-style-type: none"> • EMV-Festigkeit, • Fehlertoleranz, • Ereignisgesteuert • 100 kB/s 	Wie System K-CAN + Entkopplung des Gefahrenbereichs Crash, „Angreifer“	<ul style="list-style-type: none"> • Regelvorgänge • Zyklisch + Ereignisgest. • 500 kB/s 	<ul style="list-style-type: none"> • System ISIS • Sicherheit • Teilnetzfähig • 10 MB/s 	<ul style="list-style-type: none"> • Kontinuierliche Signale • Automotivstandard • ca.22 MB/s
<p>Baumstruktur Kupferkabel</p>	<p>Baumstruktur Kupferkabel</p>	<p>Baumstruktur Kupferkabel</p>	<p>Sternstruktur Lichtwellenleiter</p>	<p>Ringstruktur Lichtwellenleiter</p>

- Integrierter Systementwurf: Homogenität über alle Busse
- Powermodul stellt Startfähigkeit selbst in Fehlerfällen sicher, Abschaltung von Verbrauchern
- Standard Core: Vereinheitlichung der STG-Basisfunktionalität.
- Gateway: Firewall zwischen den Teilsystemen

BMW AG Axel Daicke EE-I
FH Deggendorf AK Mechatronik
Seite 35

© R. Bettati

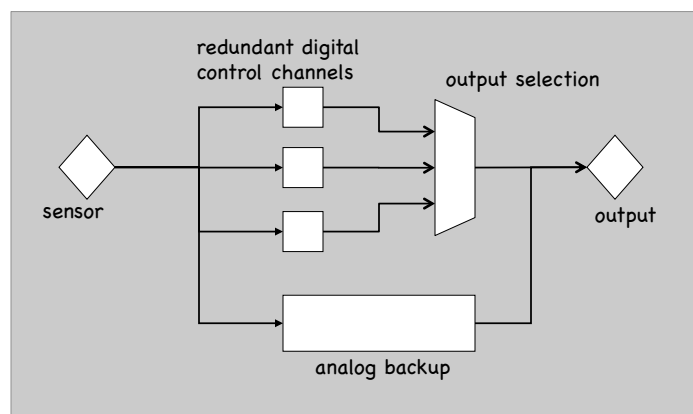
Cars as SoS (IV)



Asynchronous Design of Digital Flight Control Systems

(J. Rushby, SRI-CSL-93-07, Nov. 1993)

- Advanced Fighter Technology Integration (AFTI) F-16 DFCS:



© R. Bettati

Asynchronous Design of Digital Flight Control Systems

“... The asynchronous design of the [AFTI-F16] DFCS introduced a random, unpredictable characteristic into the system. The system became untestable in that testing for each of the possible time relationships between the computers was impossible. This random time relationship was a major contributor to the flight test anomalies. Adversely affecting testability and having only postulated benefits, asynchronous operation of the DFCS demonstrated the need to avoid random, unpredictable, and uncompensated design characteristics.”

D. Mackall, flight-test engineer AFTI-F16 flight tests

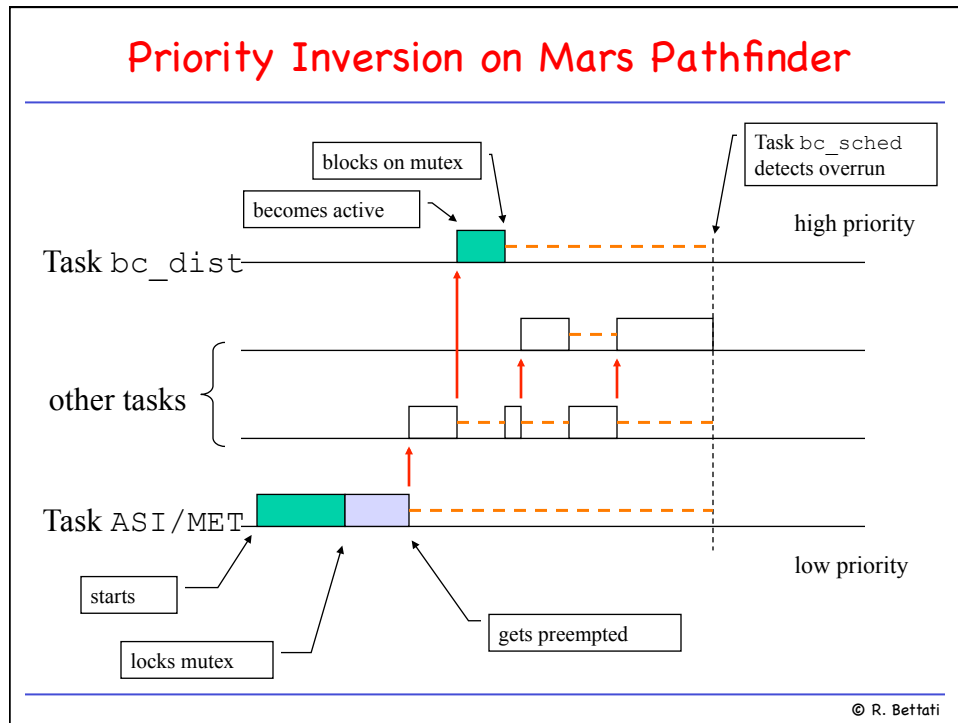
© R. Bettati

Example: Mars Pathfinder Incident



- Landing on July 4, 1997
- “experiences software glitches”
- Pathfinder experiences repeated RESETS after starting gathering of meteorological data.
- RESETS generated by watchdog process.
- Timing overruns caused by priority inversion.
- Resources:
research.microsoft.com/~mbj/Mars_Pathfinder/Mars_Pathfinder.html

© R. Bettati



Real-Time vs. Non-Real-Time Systems

Q: What distinguishes RT systems from non-RT systems?
A: Timing constraints!

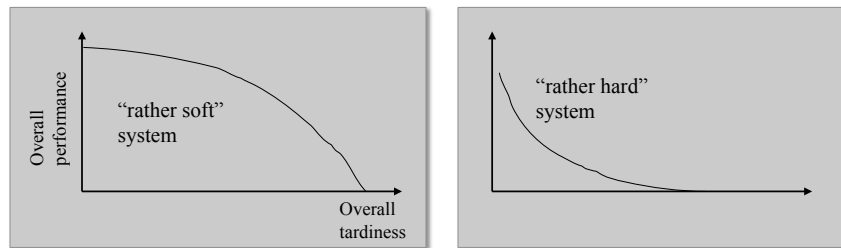
- Jobs and Processors:
 - **Job:** Unit of work executed by the system
 - **Processor:** Jobs require resource to execute (CPU, disk, network link)

No distinction necessary between types of processors!
- Timing constraints:
 - **Release Time:** time when job becomes available for execution
 - **Deadline:** time when execution must be completed
 - **Relative Deadline:** maximum response time

© R. Bettati

Hard vs. Soft Deadlines

- **Hard Deadline:** Late result may be a fatal flaw, of little use, or cause disastrous consequences
- **Soft Deadline:** Timely completion desirable. Late results useful to some degree
- **Quantitative measure:** Overall system performance as function of tardiness of jobs.



- **Operational Definition:** A job has a **hard** deadline whenever the system designer must **prove** that the job **never** misses its deadline.

© R. Bettati

Hard Real-Time Systems

Definition: A real-time system is **hard-real-time** when a large portion of the deadlines is hard.

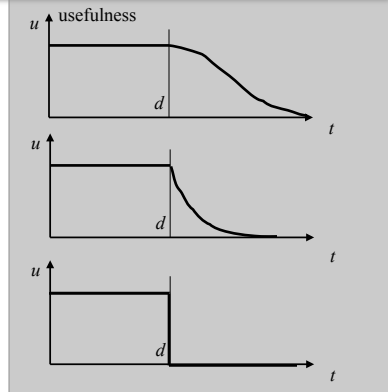
- Examples:
 - Embedded systems
 - Recovery procedures in high-availability systems
- Does real-time mean fast ?
- Verification, certification: Why not use commercial OSs?
- Why requirements to meet deadlines 100% of the time?
 - Validation of probabilistic timing requirements.
 - Assessment of compound effect of missed deadlines with other factors.

© R. Bettati

Soft Real-Time Systems

Definition: A real-time system is a **soft-real-time** system when jobs have soft deadlines.

- **Non-stringent** timing requirements
 - on-line transaction system
 - telephone switches
- **More stringent** timing requirements
 - Stock price quotation system
- **Stringent** timing requirements
 - Multimedia



- Requirements often specified in probabilistic terms; validation is done by simulation, trial use.

© R. Bettati