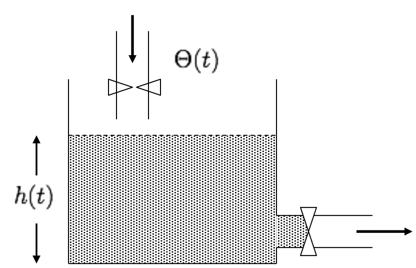
Real-Time Systems: Examples / Case Studies

- 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

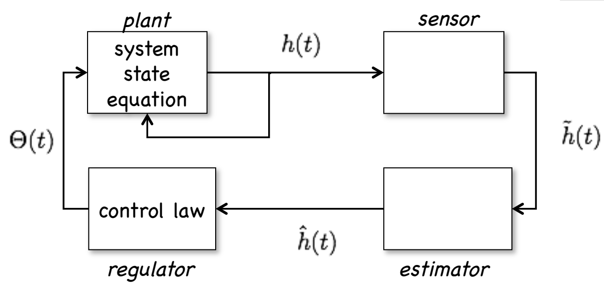
- Real-Time Systems
- Hard and soft deadlines; operational definition

Application Areas: Control Systems

- Example: Water Tank →



- In other words



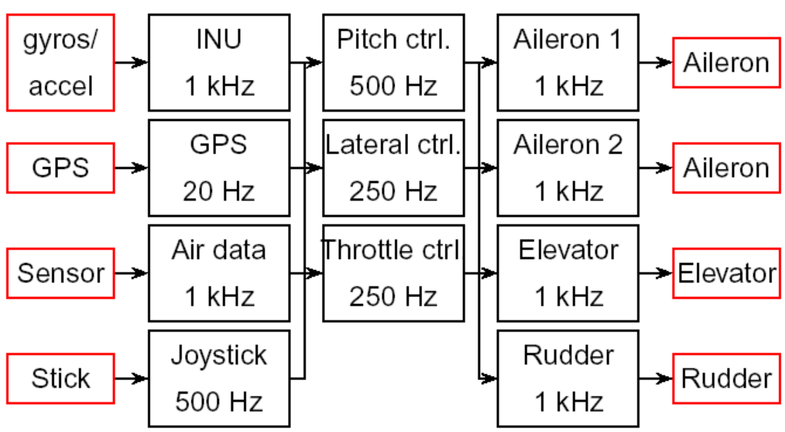
Control Systems (cont)

- 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
```


Example: Avionics System

Hard real-time system with multirate behavior



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:
$$J(u) = S(s(t_f), t_f) + \int_0^{t_f} L(x(t), u(t), t)dt$$
- **Optimal control** $u^*(t)$ with performance index J^*.

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:**
$$u^*(t) = \frac{x_d pab e^{at}}{ae^{at_f} + pb^2 sinh(at_f)}$$
- Final state:
$$x^*(t_f) = \frac{x_d pb^2 sinh(at_f)}{ae^{at_f} + pb^2 sinh(at_f)}$$

Open-Loop Temperature Control (cont)

- **Discretize** control input u:
  - Sampling period P.

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

- Performance index for discrete optimal control:

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

- In our case:

$$J_D^*(P) \tilde{=} \frac{1}{2} p x_d \left(\frac{1 - e^{-aP}}{1 + e^{-aP}} \right)$$

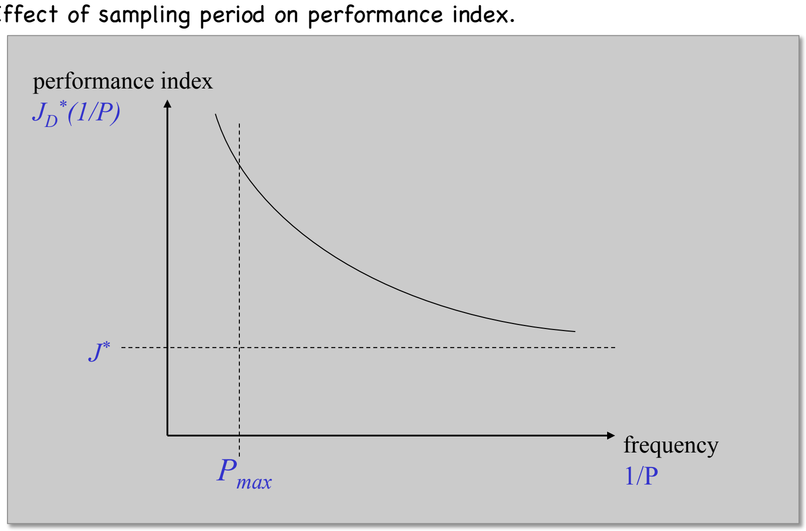
- Constraints:

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

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

Open-Loop Temperature Control (cont)

- Effect of sampling period on performance index.



Quality of Control vs. Processing Cost (cont)

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

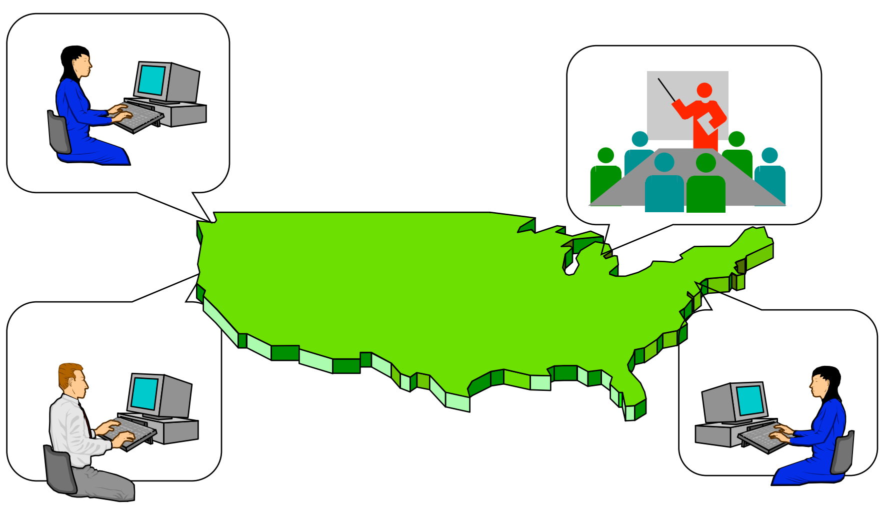
$$\Delta J^*(P) := J^*_D(P) - J^*$$

Optimization problem:

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

1. $P_i <= P_i^{max}$ // 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

Example: Multimedia

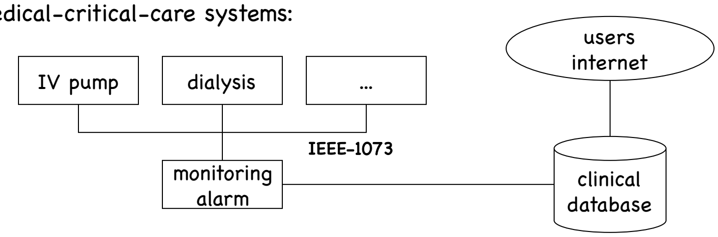


Example: Teleseminars

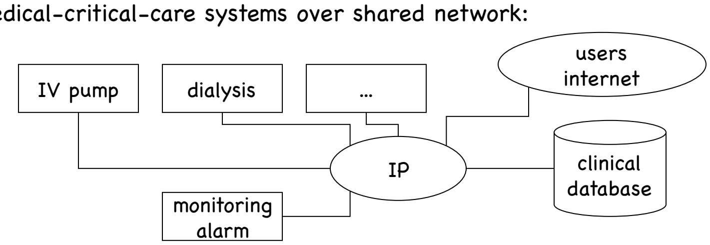
Example: Intensive Care Computing

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

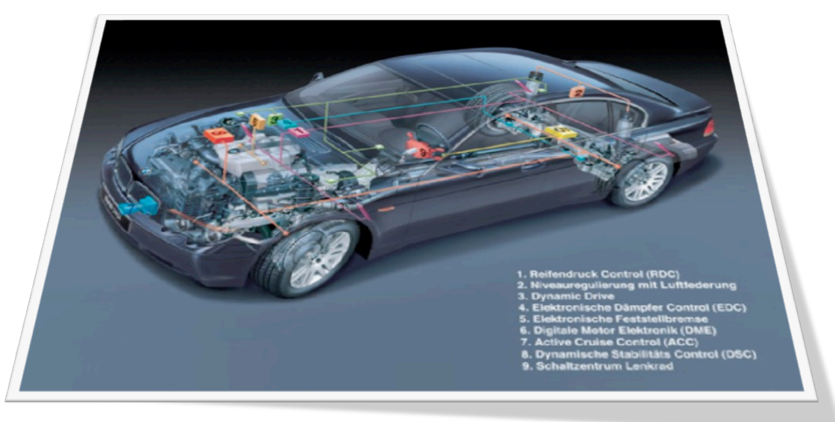
- Medical-critical-care systems:



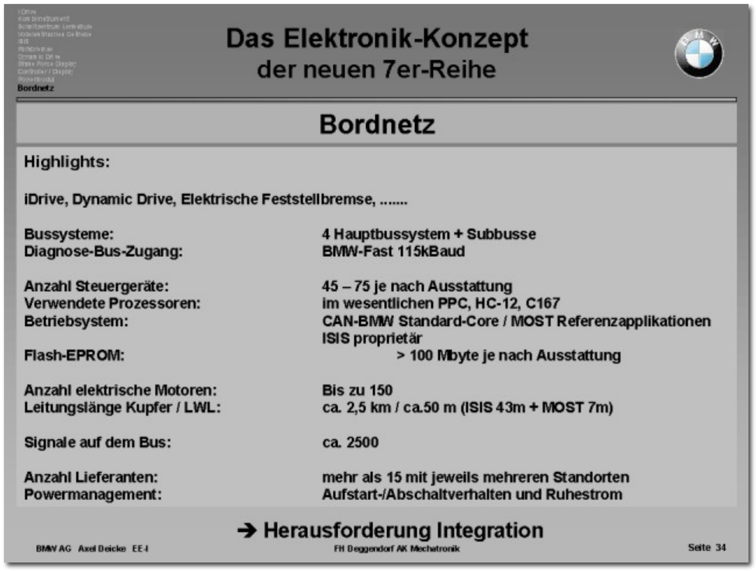
- Medical-critical-care systems over shared network:



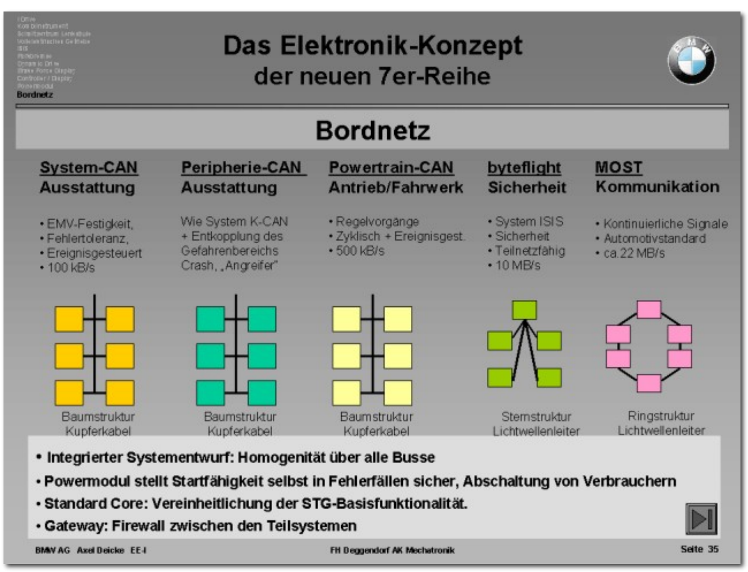
Example: Cars as Systems-of-Systems



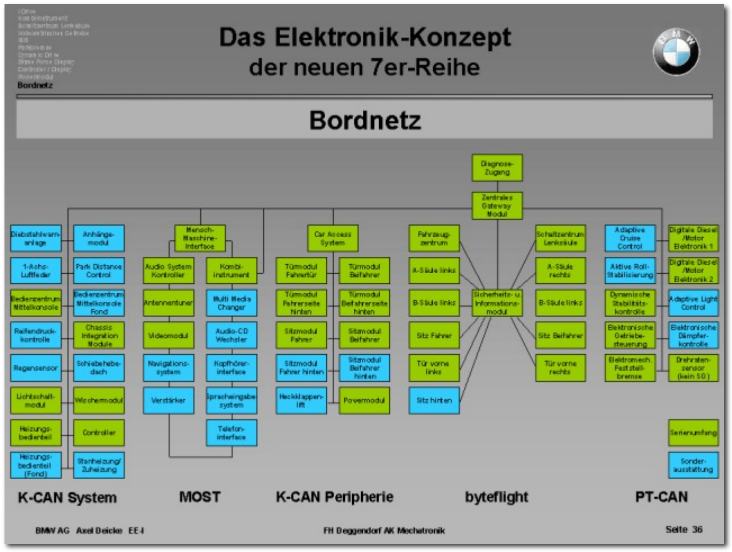
Cars as System of Systems (II)



Cars as SoS (III)



Cars as SoS (IV)

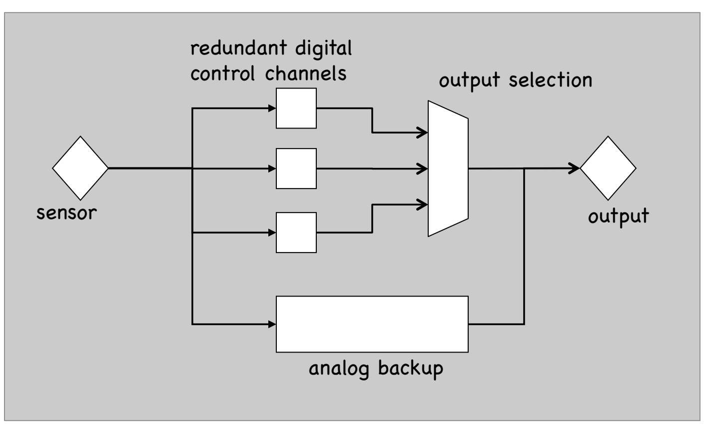


© R. Bettati

Asynchronous Design of Digital Flight Control Systems

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

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



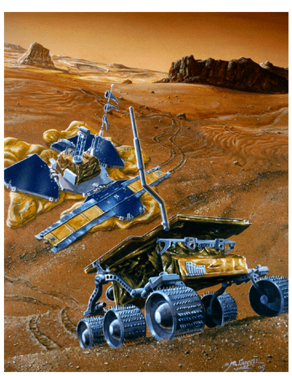
© R. Bettati

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."

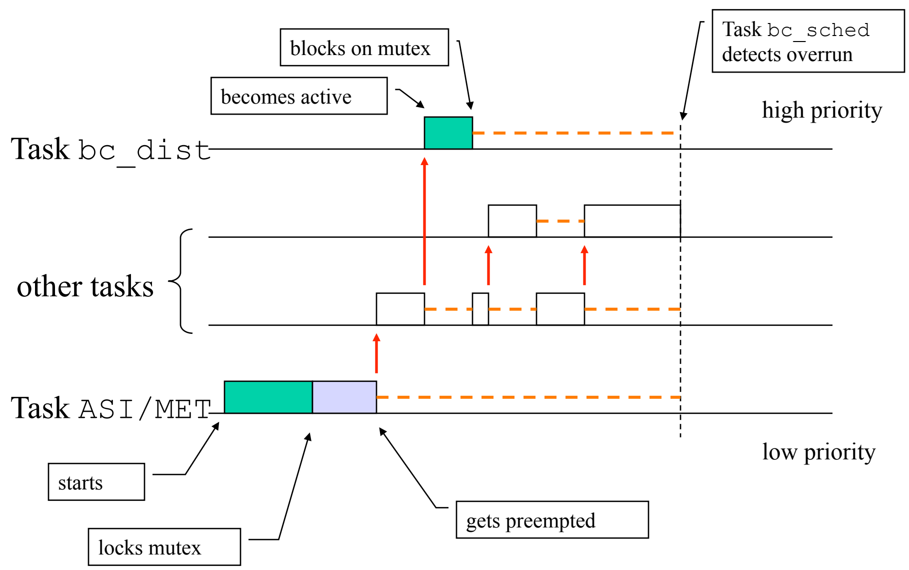
D. Mackall, flight-test engineer AFTI-F16 flight tests

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

Priority Inversion on Mars Pathfinder



Real-Time vs. Non-Real-Time Systems

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

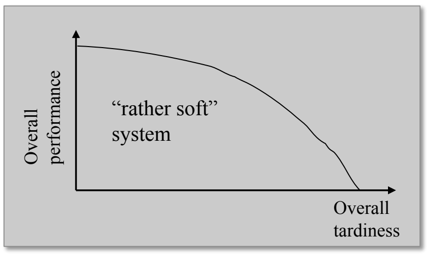
- Jobs and Processors:
  - **Job**: Unit of work executed by the system
  - **Processor**: Jobs require resource to execute (CPU, disk, network link)

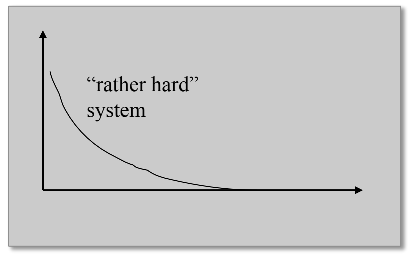
  No distinction necessary between types of processors!

- Timing constraints:
  - **Release Time**: time when job becomes available for execution
  - **Deadline**: time when execution must be completed
  - **Relative Deadline**: maximum response time

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.

Hard Real-Time Systems

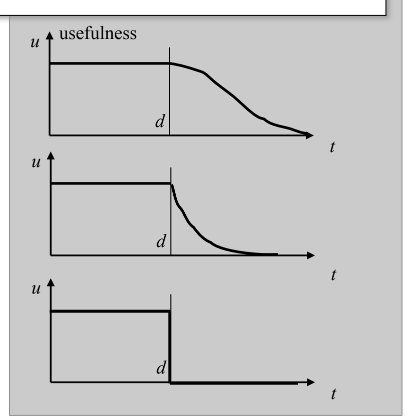
Definition: A real-time system is **hard-real-time** when a large portion of the deadlines is hard.

- 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.

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.