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

\[ \Theta(t) \rightarrow \exists \theta \leq \Lambda \rightarrow h(t) \rightarrow \hat{h}(t) \rightarrow \hat{h}(t) \rightarrow \text{sensor} \rightarrow \Theta(t) \]

© R. Bettati
Control Systems (cont)

• Control Loop:

\[
\begin{align*}
\text{DO \ FOREVER} \\
\text{wait\_for\_delay} \\
h &:= \text{fluid\_height} \\
\text{theta} &:= \text{valve\_position} \\
r &:= \text{table\_lookup}(h, \text{theta}) \\
\text{IF} \quad r = \text{left} \quad \text{THEN} \quad \text{turn\_left} \\
\text{ELSE IF} \quad r = \text{right} \quad \text{THEN} \quad \text{turn\_right} \\
\text{ELSE} \quad \text{do\_nothing} \\
\text{ENDDO}
\end{align*}
\]

Example: Avionics System

Hard real-time system with multitare behavior

- Gyros/accel
- GPS
- Sensor
- Stick
- INU 1 kHz
- GPS 20 Hz
- Air data 1 kHz
- Joystick 500 Hz
- Pitch ctrl. 500 Hz
- Lateral ctrl. 250 Hz
- Throttle ctrl. 250 Hz
- Aileron 1 1 kHz
- Aileron 2 1 kHz
- Elevator 1 kHz
- Rudder 1 kHz
- Aileron
- Aileron
- Elevator
- Rudder
- Rudder
Quality of Control vs. Processing Cost
Example: Open-Loop Temperature Control


- **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 \( \delta \).
  \[ |x(t_f) - x_d| \leq \delta \]
- **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**:
  \[
  u^*(t) = \frac{x_d p e^{at} b}{a e^{at} + b^2 \sinh(at_f)}
  \]
- **Final state**:
  \[
  x^*(t_f) = \frac{x_d p b^2 \sinh(at_f)}{a e^{at_f} + b^2 \sinh(at_f)}
  \]
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:
  \[
  J^*_D(P) \equiv S(x^*(t_f), t_f) + \sum_{k=0}^{n-1} \int_{kP}^{(k+1)P} L(x^*(t), u^*(kP), t)dt
  \]

- In our case:
  \[
  J^*_D(P) = \frac{1}{2} px_d \left( \frac{1 - e^{-aP}}{1 + e^{-aP}} \right)
  \]

- Constraints:
  \[
  |x(t_f) - x_d| \leq \delta
  \]
  \[
  x_d \left( \frac{1 - e^{-aP}}{1 + e^{-aP}} \right) \leq \delta \quad \Rightarrow \quad P \leq \frac{1}{a} \ln \frac{x_d + \delta}{x_d - \delta}
  \]

---

Open-Loop Temperature Control (cont)

- **Effect of sampling period on performance index.**

![Graph showing the relationship between performance index and frequency](image-url)
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 \( \Delta J^*(\cdot) \) and execution times \( C_i \), find a set of periods \( P_i \), such that

1. \( P_i \leq P_i^{\text{max}} \) // 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


- Medical-critical-care systems:

- Medical-critical-care systems over shared network:

Example: Cars as Systems-of-Systems
Cars as System of Systems (II)

Das Elektronik-Konzept
der neuen 7er-Reihe

Bordnetz

Highlights:
- iDrive, Dynamic Drive, Elektrische Feststellbremse, ......

Bussysteme:
- 4 Hauptbussystem + Subbusse
- BMW-Past 1160Bau

Anzahl Steuergeräte:
- 45 – 75 je nach Ausstattung

Verwendete Prozessoren:
- Im wesentlichen PPC, HC-12, C167

Betriebssystem:
- CAN-BMW Standard-Core / MOST Referenzapplikationen
- RS6 proprietär
- > 100 Mbyte je nach Ausstattung

Flash-EPROM:
- Bis zu 150

Leistungslänge Kupfer / LWL:
- ca. 2,5 km / ca.59 m (BBB 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

© R. Bettati

Cars as SoS (III)

Das Elektronik-Konzept
der neuen 7er-Reihe

Bordnetz

System-CAN
- Ausstattung
  - CAN, FlexRay, Environment CAN
  - Energieversorgung
  - Bussysteme

Periphery-CAN
- Ausstattung
  - Weg System K-CAN
  - Entkopplung der Getriebestände durch CAN, Anreiter

Powertrain-CAN
- Antrieb/Fahrwerk
  - Regelvorgänge
  - Energieversorgung
  - 500 Mbit/s

bytelflight
- Sicherheit
- System IS 9
- Sicherheit
- Televerbindung
- 10 Mbit/s

MOST
- Kommunikation
  - Kommunikationssignale
  - Buchstabe standard
  - ca. 22 Mbit/s

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

© 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:

```
+------------------+
| redundant digital |
| control channels  |
+------------------+
    output selection
                   +------------------+
                   | output selection  |
                   +------------------+
                   +------------------+
                   | sensor            |
                   +------------------+
                   +------------------+
                   | analog backup     |
                   +------------------+
```
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

Task `bc_dist`
- becomes active
- blocks on mutex

other tasks
- gets preempted
- becomes active

Task `ASI/MET`
- starts
- locks mutex

Task `bc_sched`
- detects overrun

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.

![Graph showing overall performance vs. overall tardiness for a "rather soft" system and a "rather hard" system.]

- **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.