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

\[ \Theta(t) \rightarrow h(t) \rightarrow \text{sensor} \rightarrow \tilde{h}(t) \]

\[ \text{plant} \rightarrow \text{system state equation} \rightarrow \text{control law} \rightarrow \text{regulator} \rightarrow \tilde{h}(t) \]

\[ h(t) \rightarrow \text{estimator} \rightarrow \tilde{h}(t) \]

© R. Bettati
Control Systems (cont)

- Control Loop:

```plaintext
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

- Gyros/accel
  - INU 1 kHz
  - GPS 20 Hz
- Sensor
  - Air data 1 kHz
  - Joystick 500 Hz
- Stick
- 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

© R. Bettati
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 \( x_f \) within time \( t_f \);
consume minimum amount of fuel. Allow for a tolerance \( \delta \).

\[ |x(t_f) - x_f| \leq \delta \]

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:
\[ u^*(t) = \frac{\frac{\text{d}p}{\text{d}t} x_d e^{at} e^{-bt}}{a e^{at} + pb^2 \sinh(at_f)} \]

Final State:
\[ x^*(t_f) = \frac{\frac{\text{d}p}{\text{d}t} x_d b e^{at} \sinh(at_f)}{a e^{at} + pb^2 \sinh(at_f)} \]
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:**

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

In our case:

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

**Constraints:**

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

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

---

**Open-Loop Temperature Control (cont)**

- Effect of sampling period on performance index.

![Performance Index Diagram](image)
Quality of Control vs. Processing Cost (cont)

Task Frequencies must be determined to optimize the performance indices without overloading the available processing capabilities.

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

Optimization problem:

Given a set of tasks, $T_1, \ldots, T_n$ with given $\Delta J_i(*)$ 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

Example: Multimedia

Example: Teleseminars
Example: Intensive Care Computing


Traditional medical-critical-care systems:
- IV pump
- dialysis
- monitoring
- alarm
- IEEE-1073
- clinical database
- users

Medical-critical-care systems over shared network:
- IV pump
- dialysis
- monitoring
- alarm
- IP
- clinical database
- users

Example: Industrial Applications

Control Actions deployed via IRN

© R. Bettati
Example: Internet of Things

Internet of Things (II)
Example: Cars as Systems-of-Systems

Cars as System of Systems (II)

Bordnetz

<table>
<thead>
<tr>
<th>Highlights:</th>
</tr>
</thead>
<tbody>
<tr>
<td>iDrive, Dynamic Drive, Elektrische Feststellbremse, ......</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Busysteme:</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Hauptbusysteme + Subbusse</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Diagnosis-Bus-Zugang:</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMW-Fast 11-kBit/s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Anzahl Steuergeräte:</th>
</tr>
</thead>
<tbody>
<tr>
<td>45 – 75 je nach Ausstattung</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Verwendete Prozessoren:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Im wesentlichen PPC, HC-12, C167</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Betriebssystem:</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAN-BMW Standard-Core / MOST Referenzanwendungen</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flash-EPROM:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 100 Mbyte je nach Ausstattung</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Anzahl elektrische Motoren:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bis zu 150</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Leistungslänge Kupfer / LWE:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ca. 2,5 km / ca.50 m (BSI 43s + MOST 7m)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Signale auf dem Bus:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ca. 2500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Anzahl Lieferanten:</th>
</tr>
</thead>
<tbody>
<tr>
<td>mehr als 15 mit jeweils mehrerenStandorten</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Powermanagement:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aufstart/Abschaltverhalten und Ruhestrom</td>
</tr>
</tbody>
</table>

Herausforderung Integration
### Cars as SoS (III)

#### Bordnetz

<table>
<thead>
<tr>
<th>System-CAN Ausstattung</th>
<th>Peripherie-CAN Ausstattung</th>
<th>Powertrain-CAN Antrieb/Fahrrwerk</th>
<th>bytflyflight Sicherheit</th>
<th>MOST Kommunikation</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMC/EMI-Festigkeit</td>
<td>Erfassungskapazität</td>
<td>Energieeffizienz</td>
<td>System IS/IS</td>
<td>Komunikationseigenschaften</td>
</tr>
<tr>
<td>System K/CAN</td>
<td>Energiemessung</td>
<td>Torque</td>
<td>Sicherheit</td>
<td>Autonome Standards</td>
</tr>
<tr>
<td>Energiemessung</td>
<td>Torque</td>
<td></td>
<td>Temporäraufzeichnung</td>
<td>21 MHz</td>
</tr>
</tbody>
</table>

- Integriertes Systemontwurf: Homogenität über alle Busse
- Einsatzmodul stellt Stabilität selbst in Fällen sicher, Abschaltung von Verbrauchern
- Standard Core: Vereinheitlichung der STG-Basisfunktionalität
- Gateway: Firewall zwischen den Teilsystemen

---

### Cars as SoS (IV)

#### Bordnetz

- K-CAN System
- MOST
- K-CAN Peripherie
- bytflyflight
- PT-CAN

---

© R. Bettati
Cars as SoS

Example of a Backbone Architecture with FlexRay

Example: Asynchronous Design of Digital Flight Control Systems
(J. Rushby, SRI-CSL-93-07, Nov. 1993)

- Advanced Fighter Technology Integration (AFTI) F-16 DFCS:
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:
http://research.microsoft.com/en-us/um/people/mbj/Mars_Pathfinder/Mars_Pathfinder.html
Priority Inversion on Mars Pathfinder

Task `bc_dist`
- blocks on mutex
- becomes active
- high priority

Task `ASI/MET`
- starts
- locks mutex
- gets preempted
- low priority

Task `bc_sched`
- detects overrun

Real-Time vs. Non-Real-Time Systems

Q: What distinguishes RT systems from non-RT systems?

A: Timing Constraints!
Players in Real-Time Systems

**Jobs and Processors:**
- **Job**: Unit of work executed by the system
- **Processor**: Jobs require resource to execute (CPU, disk, network link)
  (We don’t distinguish 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
  - many others ...

---

**Q:** Does **real-time** mean **fast**?
**Hard Real-Time Systems**

**Q:** Why not use commercial (general purpose) OSs?

**A:** Verification, Certification

---

**Hard Real-Time Systems**

**Q:** Why do we need to meet deadlines 100% of the time?

**A:**
- 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 *soft-real-time* 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 often done by *simulation, trial use.*