Real-Time Systems

Lecture 1

Temporal constraints: source and characterization

Basic concepts about real-time
Requirements of Real-Time Systems

Adapted from the slides developed by Prof. Luís Almeida for the course "Sistemas de Tempo-Real"

A few definitions related with "Real-Time"

There is a wide variety of definitions related to **Real-Time**, **systems** dealing with Real-Time and the **services** that they provide.

All of these definitions have in common the fact that **express the dependency of a computer system on the time**, as it exists in a particular **physical process**.

Definitions related with “Real-Time”

Real-Time Service or Function

Which must be performed or provided **within finite time intervals imposed by a physical process**

Real-Time System

One who contains **at least one feature** of real-time or at least providing a service of real-time (PDC 1990)

Real-Time Science

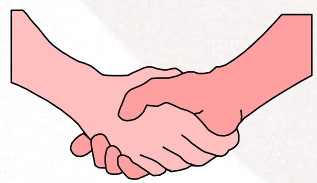
Branch of computer science that studies the introduction of Real-Time in computational systems.

Real-Time Computation

The **computation results** must be

- Logically correct
- Delivered on time

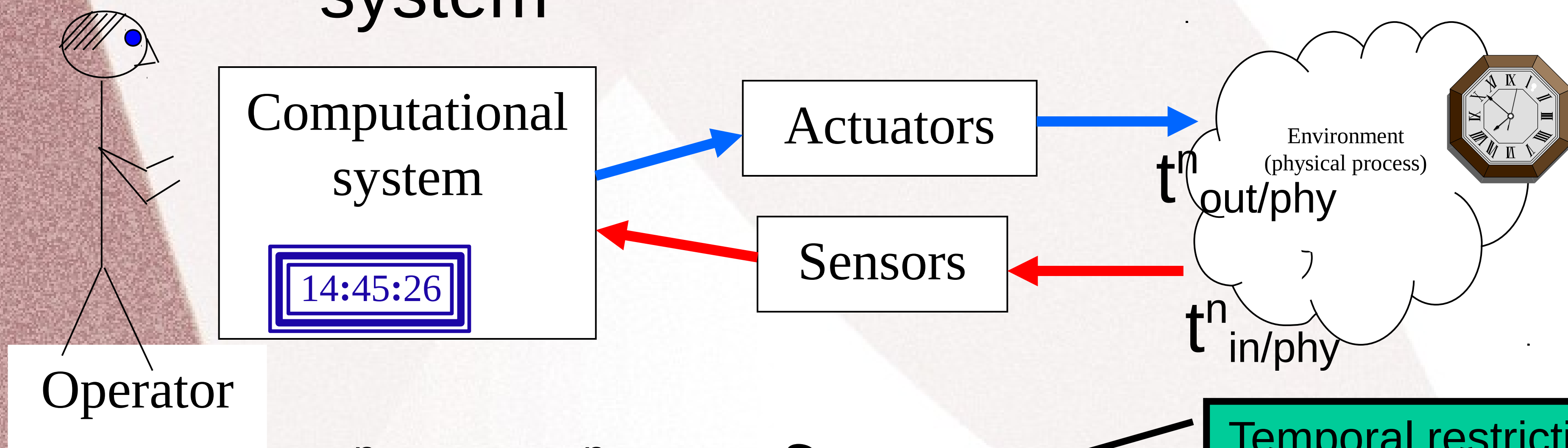
(Stankovic, 1988)

Timeliness  **Logic correction**

Real-Time System

System under control

Controller system



$$\forall n,\ t^n_{out/phy} - t^n_{in/phy} < \delta$$

$$\forall n,\ t^{n+1}_{in/phy} - t^n_{in/phy} < T$$

Temporal restrictions imposed by the environment

Notion of "Real-Time"

The **environment** with which the computer system interacts (physical process) **has its own pace of evolution**, i.e., its own dynamics.

This **rhythm** is **inherent** to the physical process itself and can not (or should not, in the case of simulators) be controlled externally. Is called a **Real-Time**.

Notion of "Real-Time"

Thus, the environment imposes on the system **timing requirements** according to its own real-time, ie, its dynamics.

For the computer system to be able to interac with its environment, it **must act on it in time**, i.e. according to the respective real-time.

Notion of “Real-Time”

Note that real-time **does not mean fast** but just the natural rhythm of a certain physical process (relative term)

Note also the **antagonism** with situations in which the pace of change **can be controlled** either by the operator or by the control system (e.g., the reservation system of air travel, banking systems control accounts). In such cases, when facing an **excess of requests**, the system **slows down** the response to these requests, continuing the processing with the fastest possible pace (best effort). Leads to **access queues** ...

Notion of "Real-Time"

- Example – **F1 pilot**
  (applies also to any driver, robots, machines ...)
  - The steering control has to be accurate, whatever the speed at which the car runs
  - All **unexpected events** (e.g. car accidents, people on the track, water in the track, flat tires) arising while the car runs at high speed **must be handled at that speed**
    - The car speed determines the real-time
    - **You can not stop instantly to think !!**

Notion of “Real-Time”

- Generally, when a control or monitoring system can monitor the state of a given physical process and, if necessary, **act on it in time**, then it is a **real-time system**.
- All living beings are **real-time in relation to their natural habitats**, which determine its real-time systems.
- On the other hand, when we build (programmable) machines to interact with physical processes, we need to use **programming** techniques and SW **infrastructures** that allow us to have **confidence in its ability to carry out timely actions**.

Objective of the study of RTS

- The main objective of the study of Real-Time Systems is the development of techniques for
  - **Design,**
  - **Analysis, and**
  - **Verification**

  that allow to obtain assurance that a given system, which is intended for real-time, has an **appropriate timing behavior**, namely satisfying the **requirements imposed by the dynamics of the system with which it interacts**

Objective of the study of RTS

Regarding the computational activities of RTS, the main aspects to consider are:

- **Execution time**
- **Response time**
- **Regularity of periodic events**

Objective of the study of RTS

Some aspects particularly important regarding

- **Execution time**
  - Code structure (language, conditional execution, cycles)
  - DMA, cache, pipeline
  - Operative System or *kernel* (*system calls*)
- **Response time and regularity**
  - Interrupts
  - *Multi-tasking*
  - Access to shared resources (*buses*, communication ports, ...)

Requirements of Real-Time Systems

The **requirements** commonly imposed to real-time systems are of three types::

- **Functional**
- **Temporal**
- **Dependability**

Functional requirements

Data gathering

- Sampling of system variables (**real-time entities**), both analog and discrete

Digital Direct Control

- Direct access of the control system to sensors and actuators

Interaction with the operator

- System status information, logs, support to correct system operation, warnings, ...

Functional requirements

Data gathering

Internally to the controller system are **local images** (internal variables) of the **entities** of real-time system.

Each image of a real-time entity has a **limited time validity** due to the temporal dynamics of the physical process.

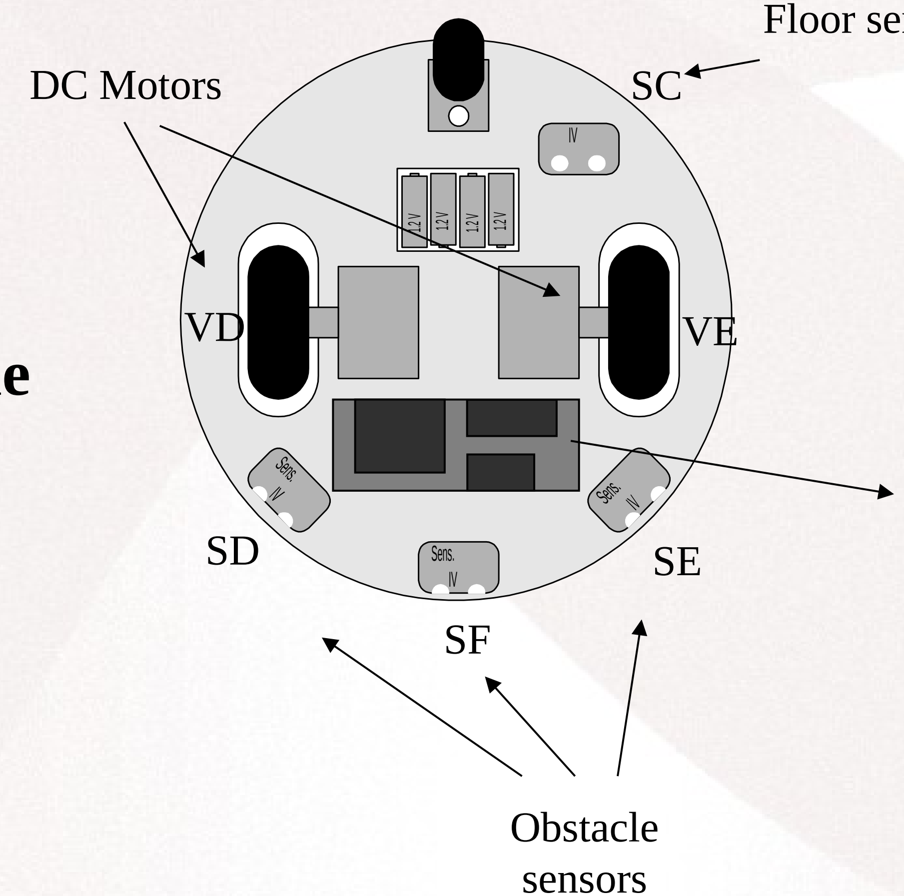
The **set of images** of the real-time entities form the **real-time database**.

The real-time database must be **updated** whenever there is a **significant change** in a value of a real-time entity.

Functional requirements

Example:

Small mobile robot



<u>RT entities</u>

Sensors:
SE, SF, SD and SC

Motor speed:
VE e VD

<u>Internal images</u>

<u>Sensors:</u>

SE’, SF’, SD’ e SC’

Motors:

VE’ e VD’

RT database

Temporal requirements

Usually arise from the **physical dynamics** of the process to be controlled

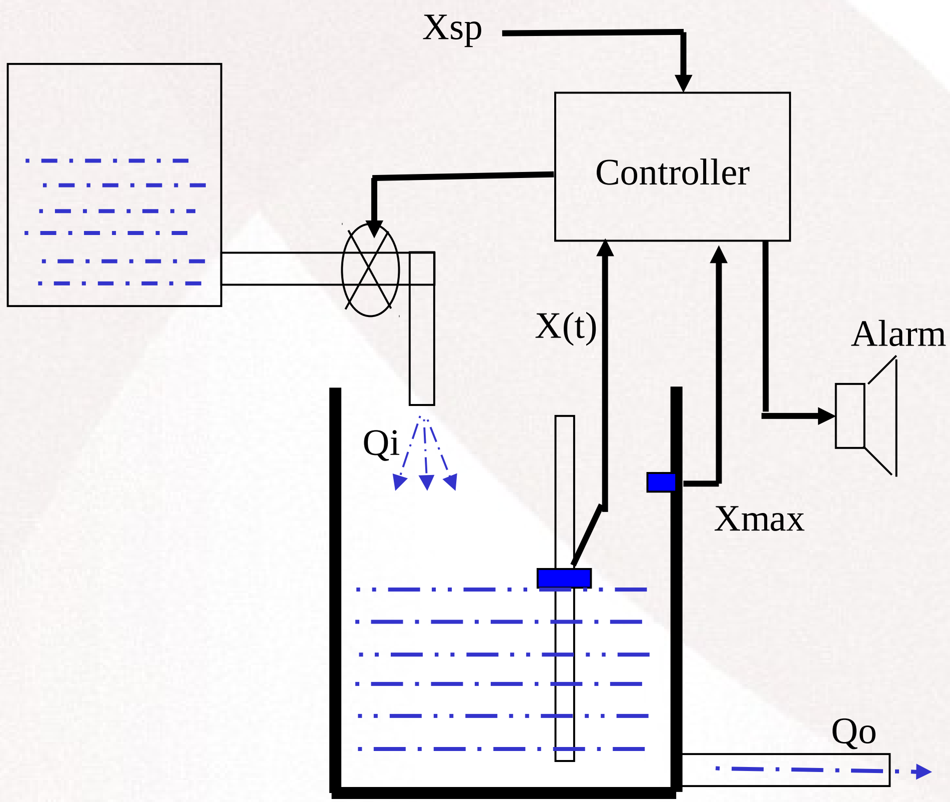
Impose **restrictions**:

- **Delays** the **observation** of the system state
- **Delays computing** the new control values (acting)
- **Variations** of previous delays (jitter)

that must be followed in all instances (including the worst case) and not only on average

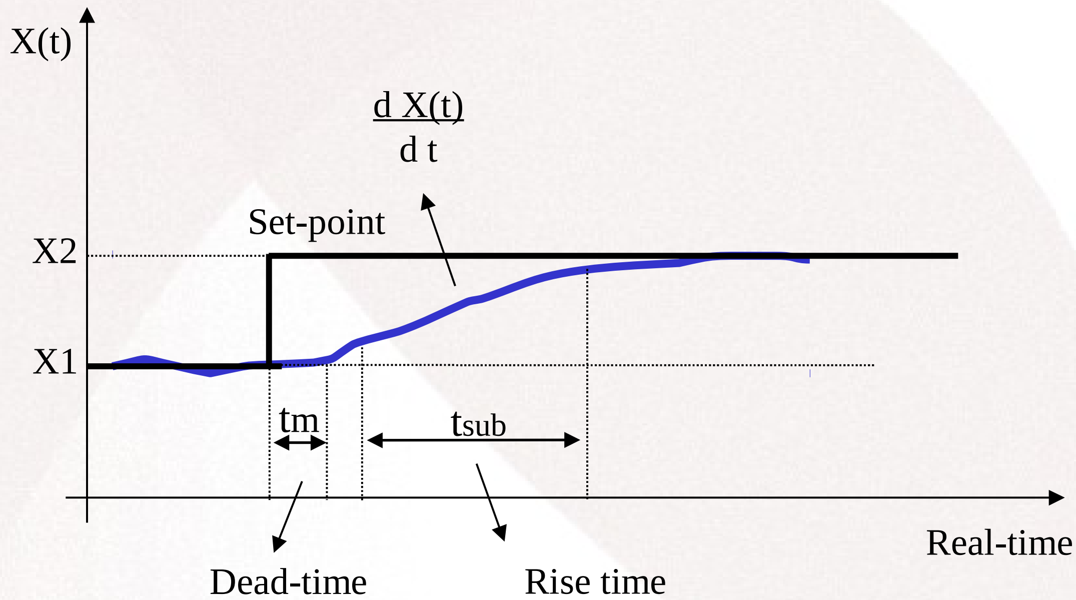
Temporal requirements

Controlling the liquid level in a container



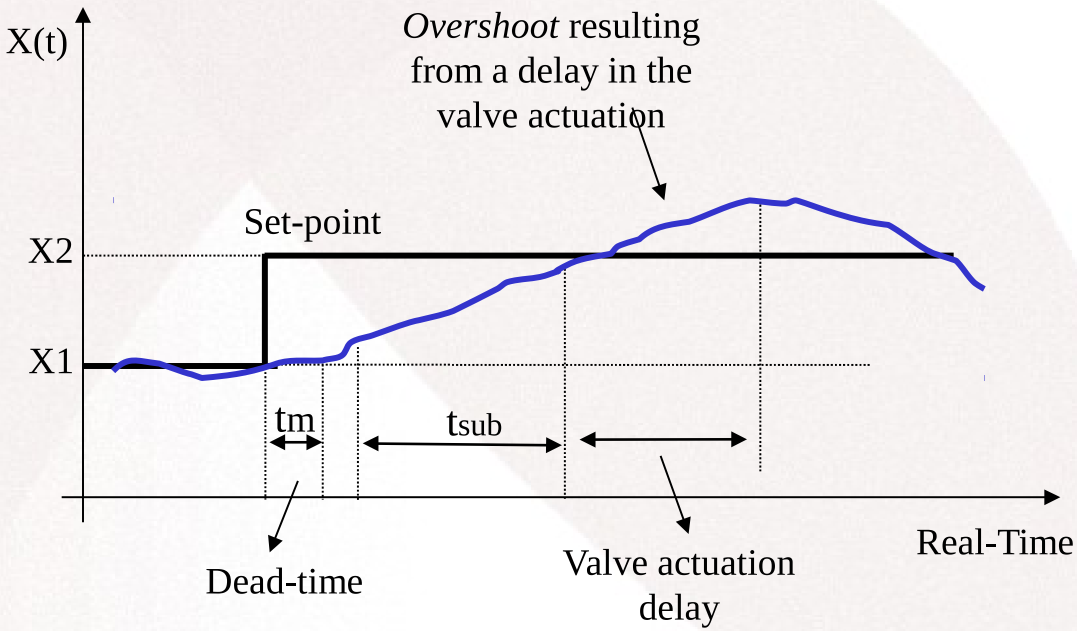
Temporal requirements

Controlling the liquid level in a container



Temporal requirements

Delay in the actuation – degradation of the control performance



Temporal requirements

The Control System imposes the following requirements

Sampling period – $\mathbf{T_s}$ (< 1/10 t_{sub} – quasi-continuous control)

Maximum delay of the valve actuation – $\mathbf{d_{max}}$ (< $\mathbf{T_s}$)

(easy to compensate by the controller)

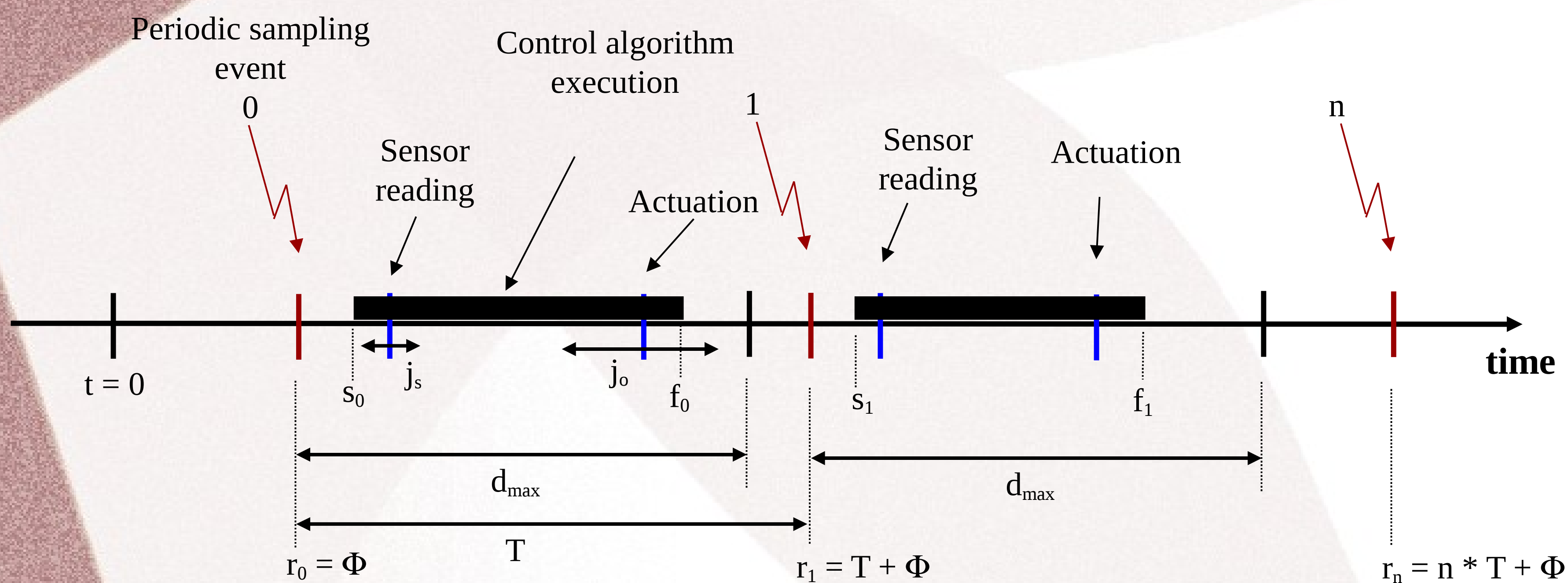
Variations on the delay of sensing the liquid level (***jitter***) – $\mathbf{j_{s,max}}$ (<< $\mathbf{d_{max}}$)

Variations on the delay of the valve actuation (***jitter***) – $\mathbf{j_{o,max}}$ (<< $\mathbf{d_{max}}$)

(hard to compensate -> degradation of the quality of control)

Maximum delay on the alam activation – $\mathbf{d_{al,max}}$

Temporal requirements



r_n – activation (release)

s_n – start of execution

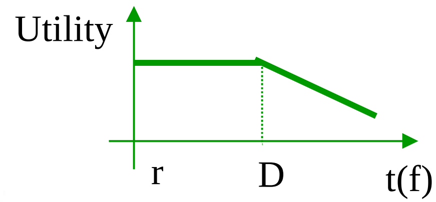
f_n – end of execution

Temporal requirements

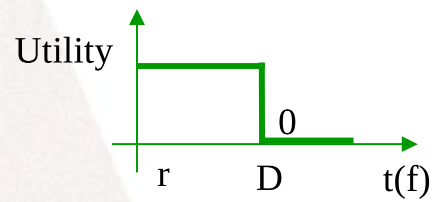
Classification of the temporal constraints:

(according with the usefulness of the result)

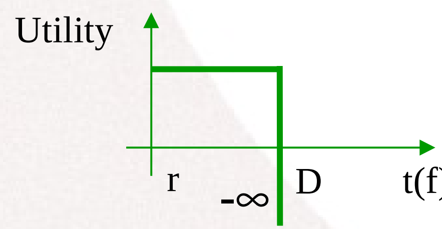
- ***Soft*** – temporal constraint in which the **result retains some utility** to the application, even after a temporal limit D, although affected by a degradation of quality of service.



- ***Firm*** – temporal constraint in which the **result loses any usefulness** to the application after a temporal limit D.



- ***Hard*** – temporal restriction that, when not met, can lead to a **catastrophic failure**.



Temporal requirements

Classification of the Real-Time Systems:

(according with the temporal constraints)

- ***Soft Real-Time*** – The system only has ***firm*** or ***soft*** real-time constraints (e.g., simulators, multimedia systems)

- ***Hard Real-Time*** – The system has at least one ***hard*** real-time constraint. These are the so-called **safety-critical systems** (e.g. airplane control, missile control, nuclear plants control, control of dangerous industrial processes)

Dependability requirements

Real-time systems are typically used in **critical applications**, in which failures may **endanger human lives** or result in high **economic impact/losses**.

This results in a requirement of:

High Reliability - Hard real-time systems have typically ultra-high reliability requirements (λ < 10-9 failures/hour). Cannot be experimentally verified!)

Dependability requirements

Important aspect to consider in safet-critical systems:

- **Stable interfaces** between the critical and the remaining subsystems, in order to avoid error propagation between each other.
- **Well defined worst case scenarios**. The system must have an adequate amount of resources to deal with worst case scenarios without resorting to probabilistic arguments, i.e. must provide service guarantees even in such scenarios.
- **Architecture** composed of autonomous subsystems, whose properties can be checked independently of the others (composability).

Summary of Lecture 1

Summary of lecture 1

- Notion of **real-time** and **real-time system**
- Antagonism between **real-time** and ***best effort***
- Objectives of the study of RTS – how to **guarantee** the **adequate temporal behavior**
- Aspects to consider: **execution time**, **response-time** and **regularity** of periodic events
- Requirements of RTS: **functional,** t**emporal** and **dependability**
- Notion of **real-time database**
- Constraints **soft**, **firm** and **hard**, and **hard real time** vs **soft real time**
- The importance of consider the **worst-case scenario**

Research work

E. Baccelli, O. Hahm et al. “Towards an OS for the Internet of Things”. INFOCOM'2013 Demo/Poster Session

- URL:http://www.embedded.com/design/real-world-applications/4430128/Towards-an-OS-for-the-Internet-of-Things