Inovace tohoto kurzu byla v roce 2011/12 podpořena projektem CZ.2.17/3.1.00/33274 financovaným Evropským sociálním fondem a Magistrátem hl. m. Prahy.

OPPA

EVROPSKÁ UNIE

Evropský sociální fond

Praha & EU: Investujeme do vaší budoucnosti

Department of Distributed and Dependable Systems

Embedded and Real-time Systems

What are Embedded and Real-Time Systems?

http://d3s.mff.cuni.cz

Department of Distributed and Dependable Systems

Tomáš Bureš

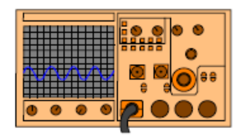
<bures@d3s.mff.cuni.cz>

CHARLES UNIVERSITY IN PRAGUE

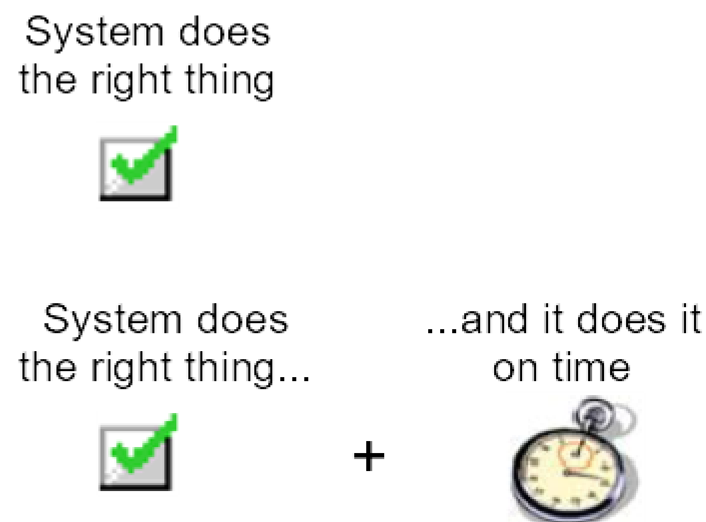
Faculty of Mathematics and Physics

What is a real-time system?

- Lots of products contain embedded computers, e.g., cars, planes and medical equipment
- In such systems it's important to deliver correct functionality on time



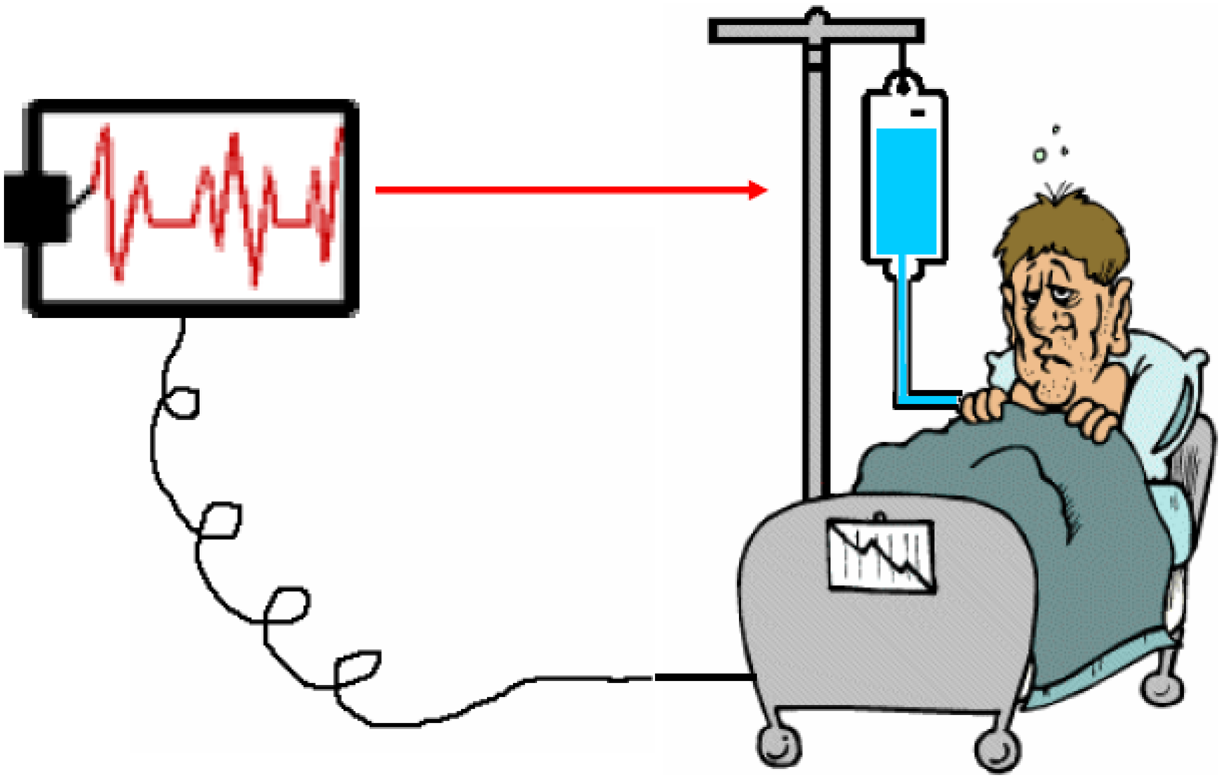

- Non-real-time systems
  - Correct function if produced result is correct
- Real-time systems
  - Correct function if produced result is correct and delivered on time



Department of
Distributed and
Dependable
Systems
D3S

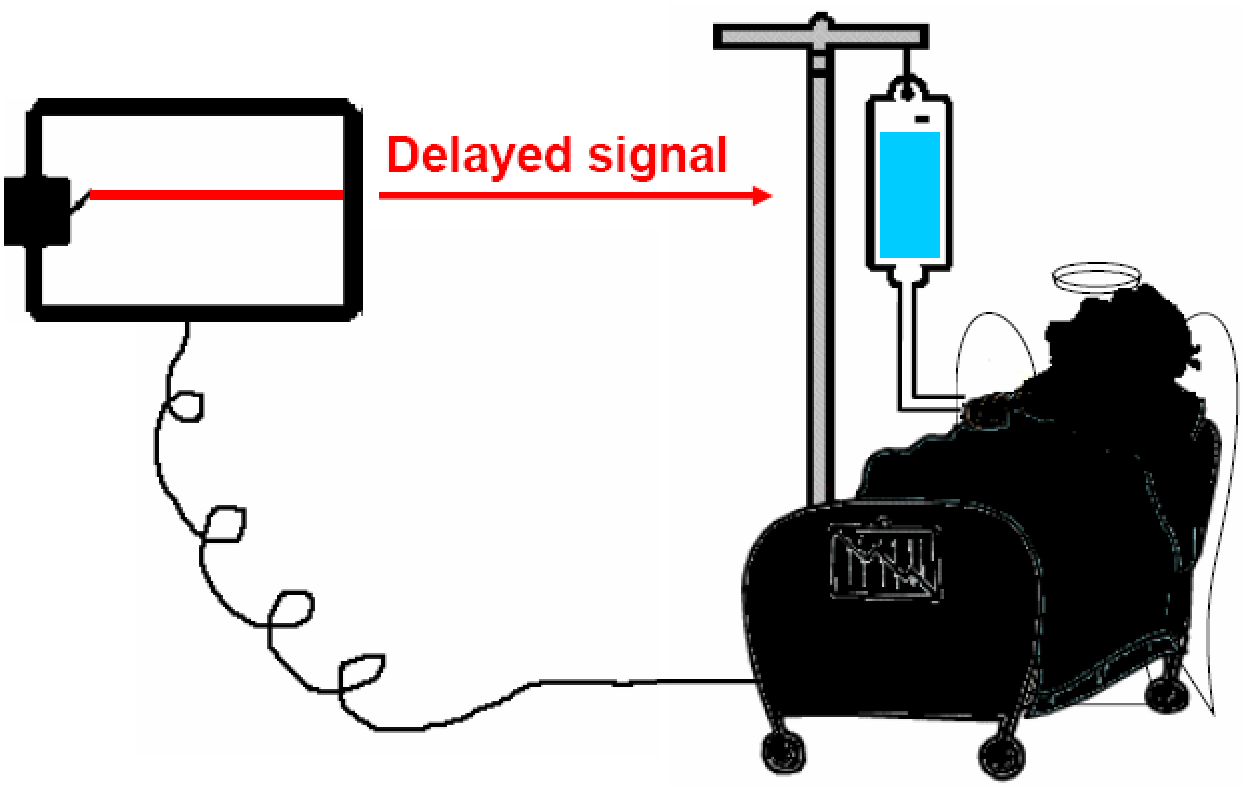
In real-time systems late data = bad data

- **Example:** Medical equipment must detect changes in the patient and respond on time...



In real-time systems late data = bad data

- **Example:** ...otherwise...

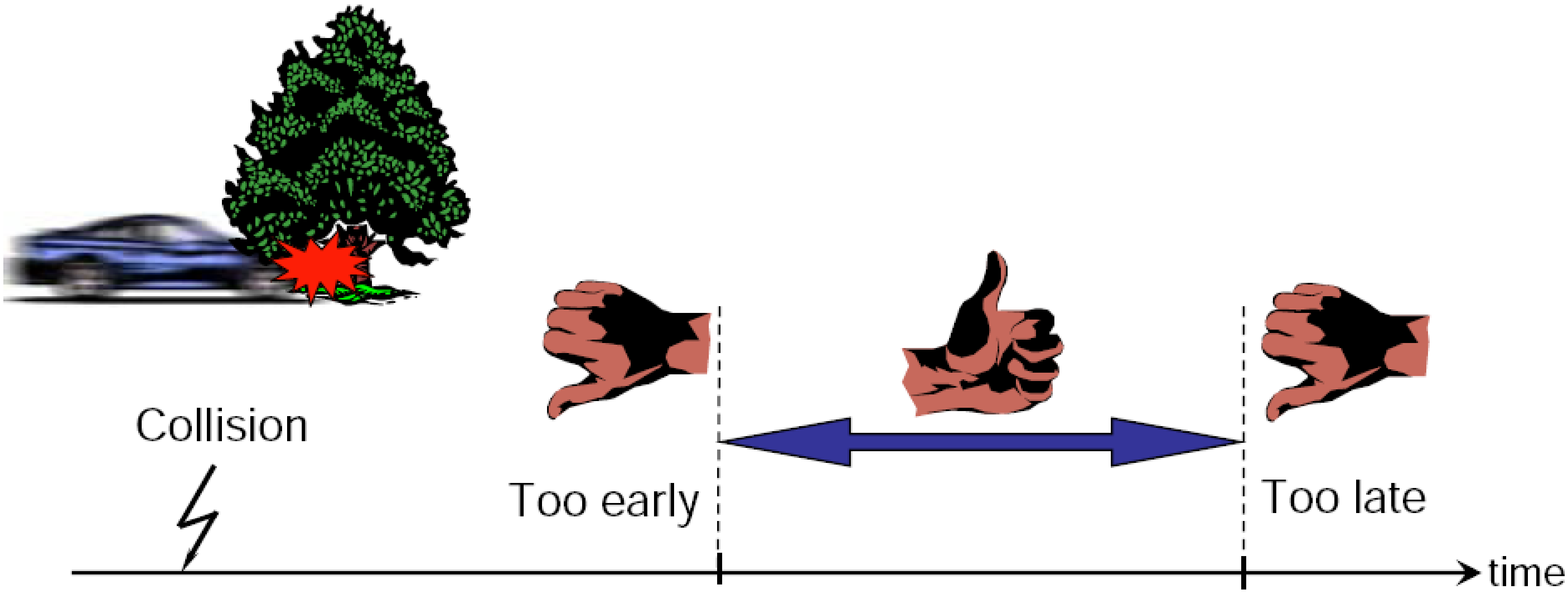


Response to a critical event must be given on time

Department of Distributed and Dependable Systems

Rather predictable than fast response

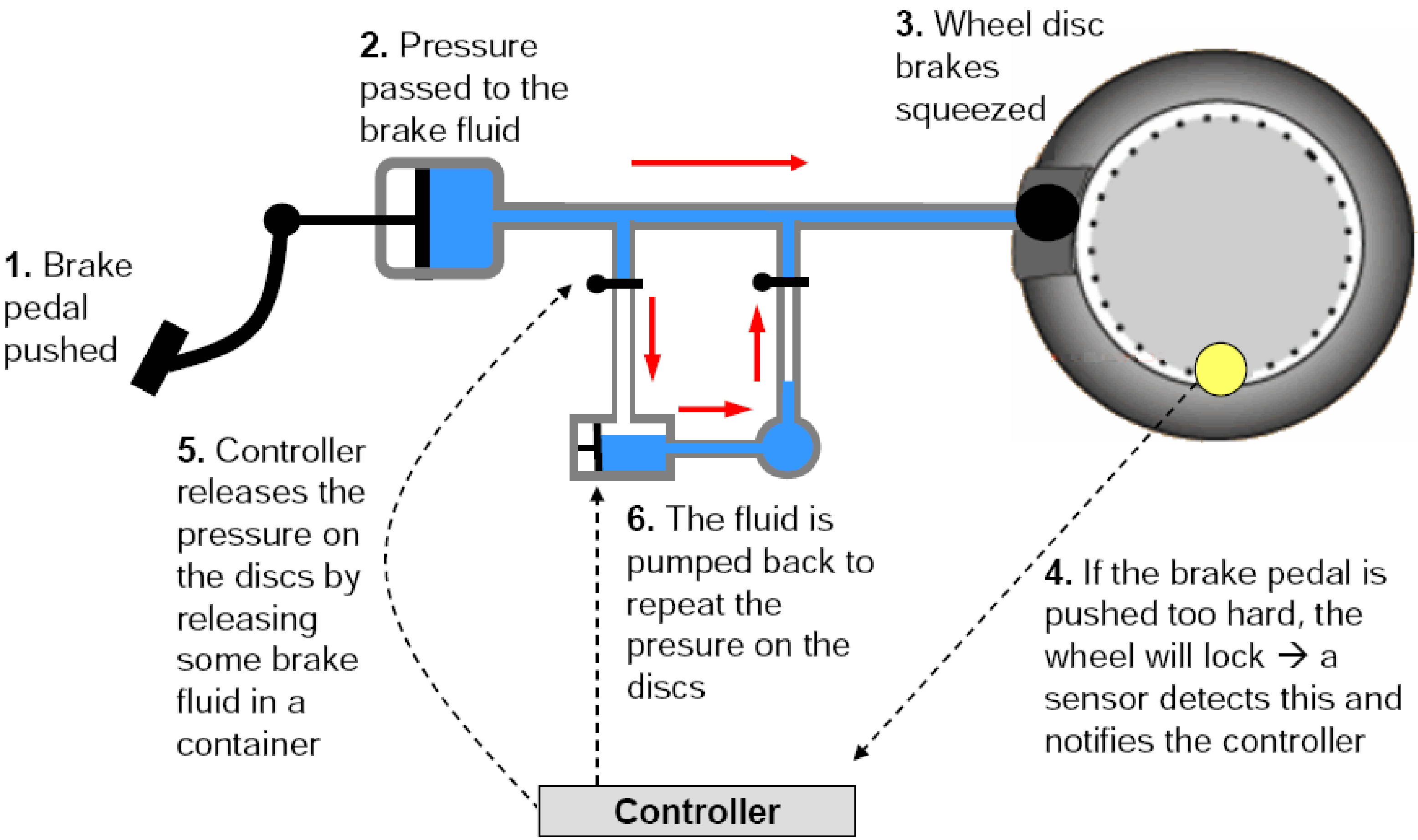
- **Example:** An air bag must not be inflated too late, nor too early!



Real-time ≠ fast !

RT systems are usually safety-critical

- **Example:** Anti-lock Braking System (ABS)



7. Entire process is repeated about 15 times/sek

Figure taken from Issovic, D.:Real-time systems, basic course

Department of Distributed and Dependable Systems

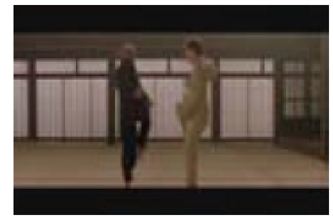
RT is not only safety-critical

Picture 1

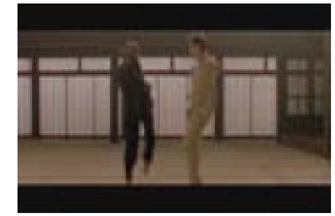
Picture 2

Picture 3

Picture 4



Picture 5



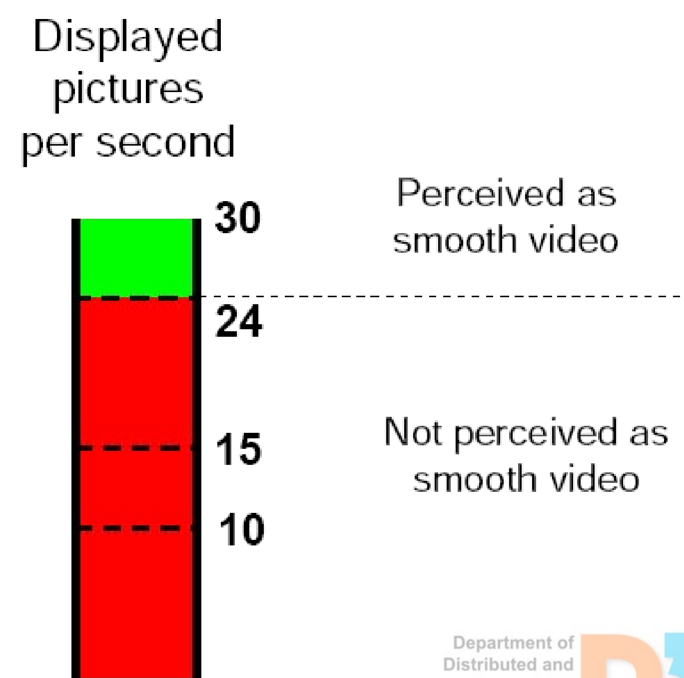
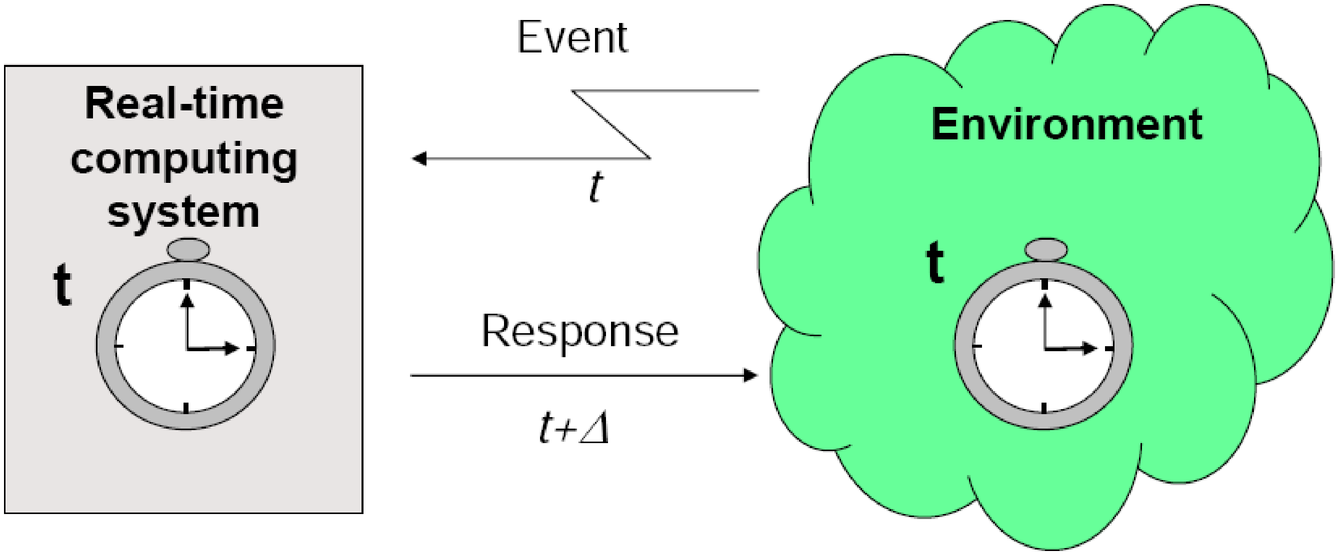


Figure taken from Issovic, D.:Real-time systems, basic course

What is a real-time system?

- "A real-time system is a system that reacts upon outside events and performs a function based on these and gives a response within a certain time. Correctness of the function does not only depend on correctness of the result, but also the timeliness of it."

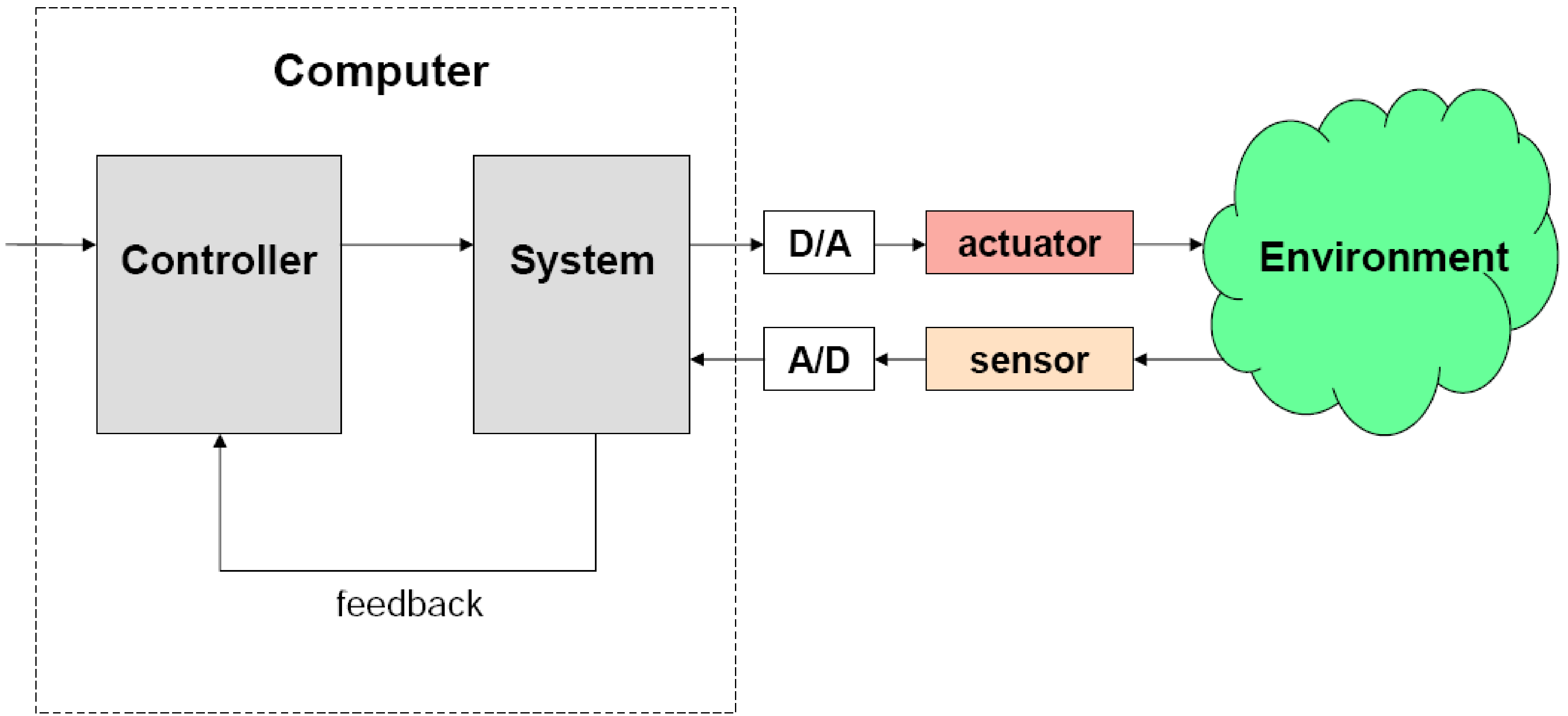


- Real-time means that the system must be synchronized with the environment. The controlled process dictates the time scale (some processes have demand on response at second-level, others at milli- or even microsecond level).

Department of Distributed and Dependable Systems

Figure taken from Issovic, D.:Real-time systems, basic course

Interaction with the environment



Example: an electrical engine

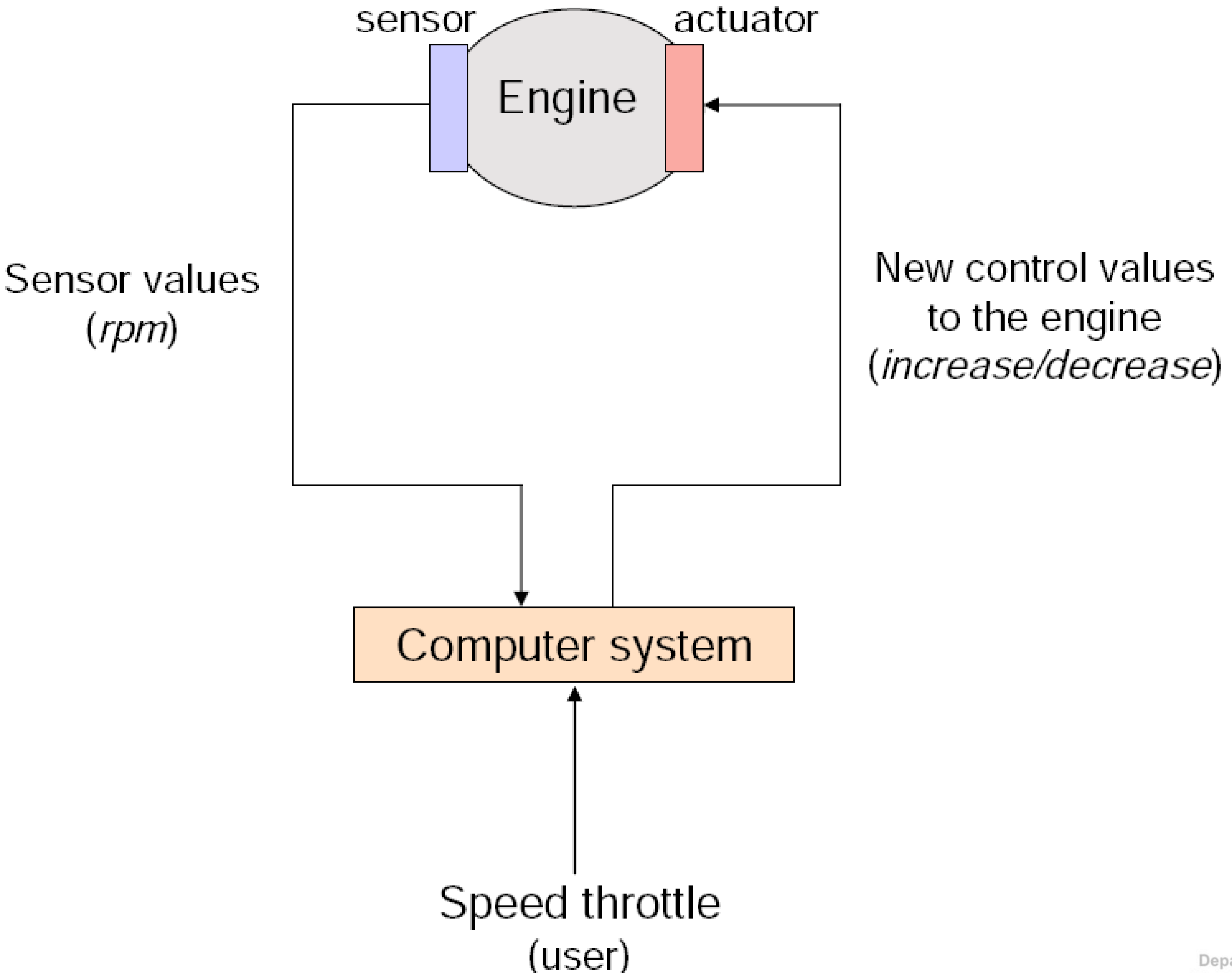
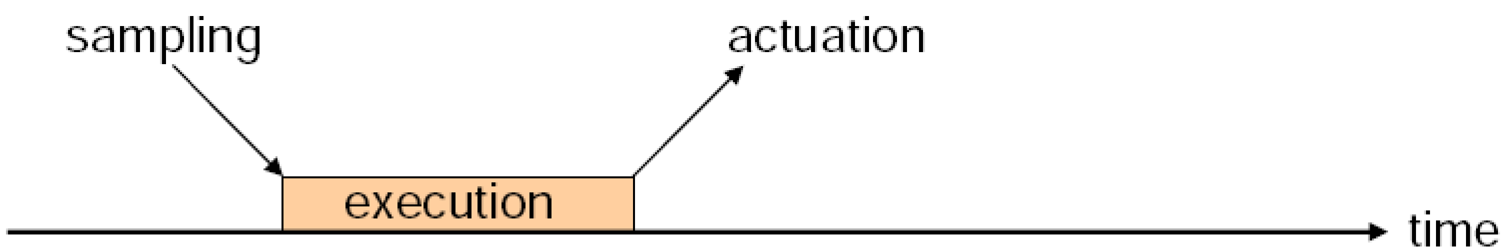


Figure taken from Issovic, D.:Real-time systems, basic course

What is real-time in this example?

- When computer system controls the speed, it has to:
  1. Observe the process, i.e., read the sensors (sampling)
  2. Decide what has to be done, i.e., execute the control algorithm
  3. Give a new control signal to the process via the actuator (actuation)



Fundamental view of real-time systems

- Temporal demands can be divided into two parts
- How fast do we have to process input and respond to the controlled process (delay)
- How often do we have to sample the environment to get a sufficiently good view of it (period)

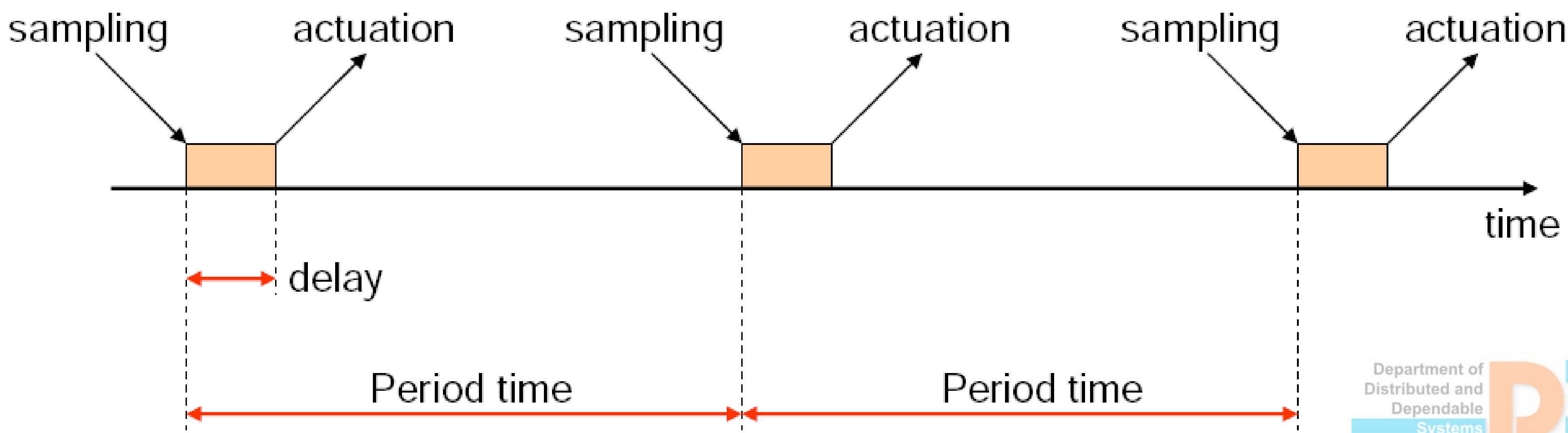
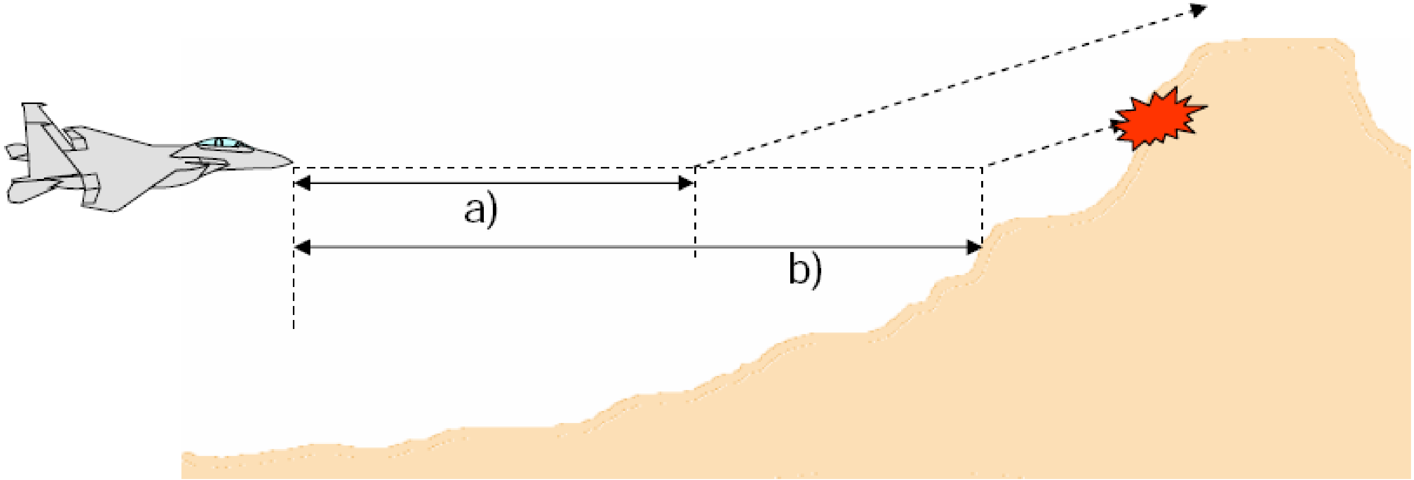


Figure taken from Issovic, D.:Real-time systems, basic course

Example: controlling the elevation with an autopilot



a) If we check the elevation often enough, we will discover changes in the terrain and have enough time to make corrections

b) Sampling done too far apart $\Longrightarrow$ catastrophic consequences

- Sampling too often means waste of resources (CPU). If too much time is spent in controlling the elevation, we might miss controlling something else.
- Thus it is important to distribute the time and resources in a good way.

Figure taken from Issovic, D.:Real-time systems, basic course

Charecteristics of real-time systems

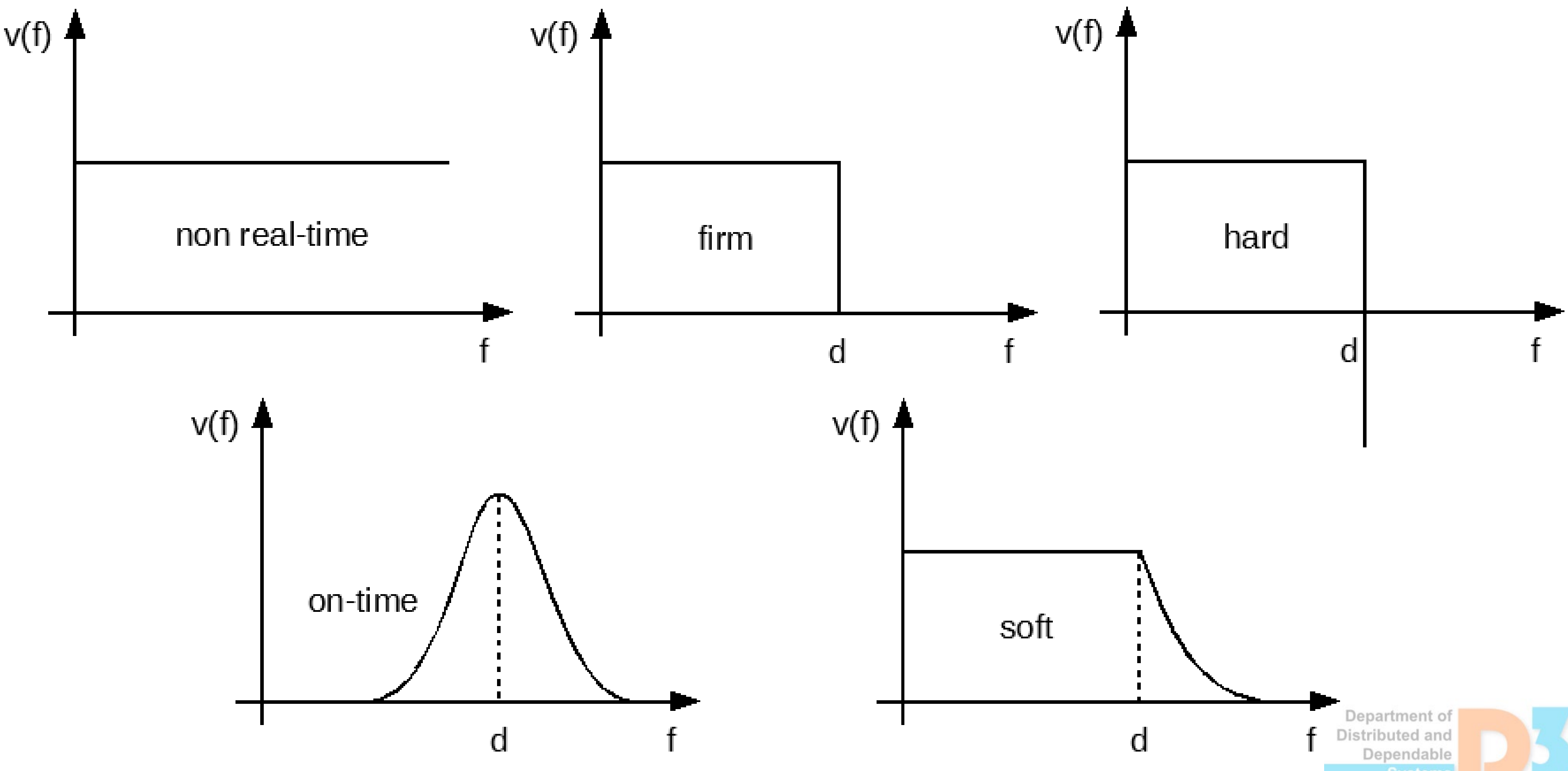
- Close coupling to process – I/O
- Predictably fast handling of events
- Handling of several system activities at the same time
- Possibility to prioritize among system activities
- Design for peak load and fault tolerance
- Configuring of program execution as cyclic or event triggered
- Internally hold a view of the process being controlled, e.g., its different states

Classification of real-time systems

- **Resources**
  - Enough resources (e.g., ABS brake system)
  - System with limited resources (e.g., telephone switches)
- **Activation**
  - Event Triggered (ET) systems (e.g., bank transaction systems)
  - Time Triggered (TT) systems (e.g., aircraft control system)
- **Service level**
  - Soft real-time systems (e.g., multimedia systems)
  - Hard real-time systems (e.g., airbag)
- **Applications**
  - Embedded real-time systems (e.g., medical equipment)
  - Not embedded real-time systems (e.g., industrial control systems)

Utility function

- The criticalness of the timing may be characterized by a cost function



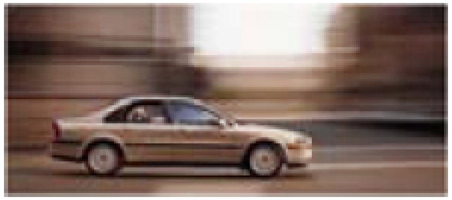
Department of Distributed and Dependable Systems

Characteristics of embedded systems

- Special-purpose computer system
  - designed to perform one or a few dedicated functions
  - often with real-time computing constraints
- Embedded as part of a complete device
  - including hardware and mechanical parts
- Optimized
  - reduced cost and size
  - mass production

Department of Distributed and Dependable Systems

Embedded systems: Volvo S80



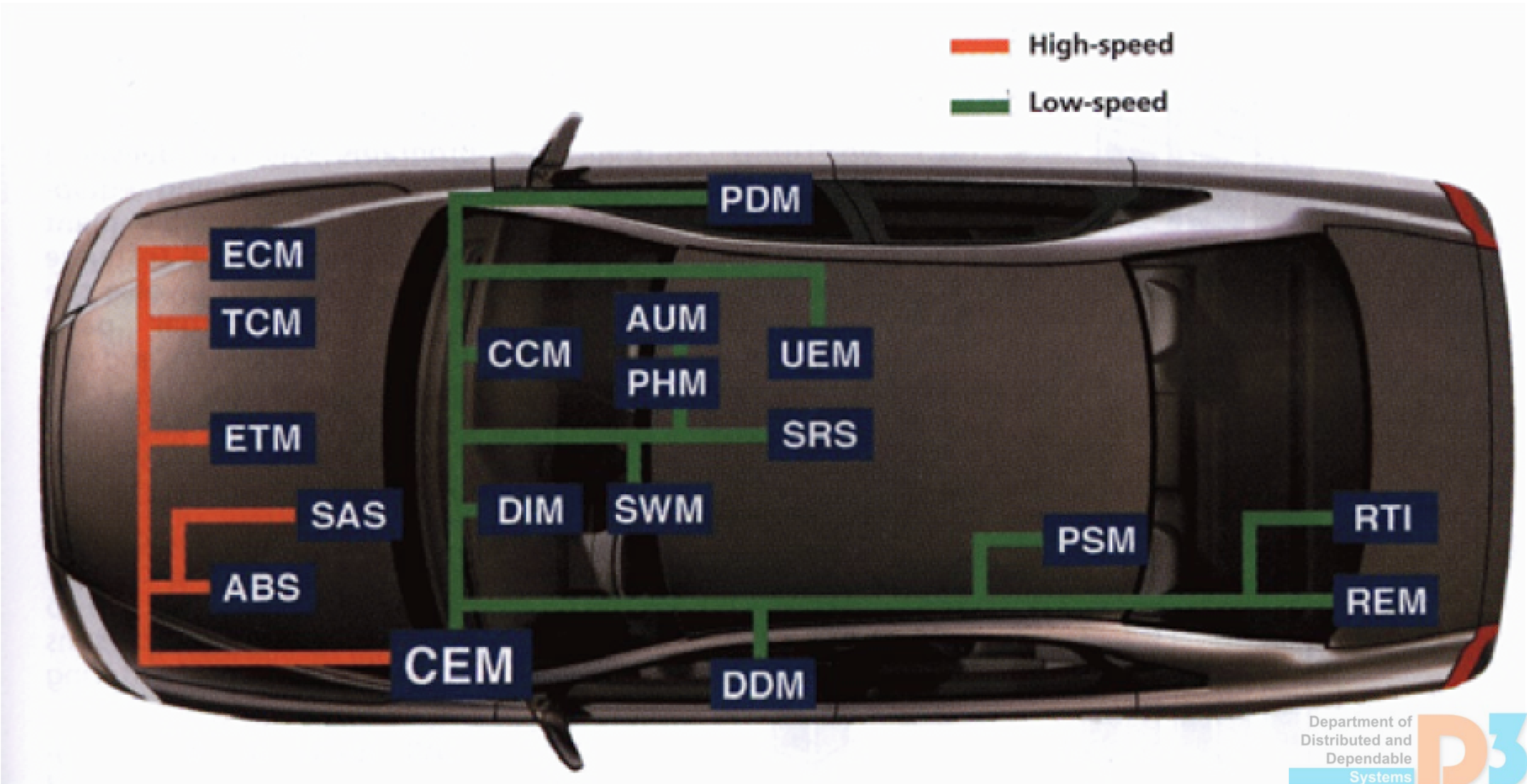
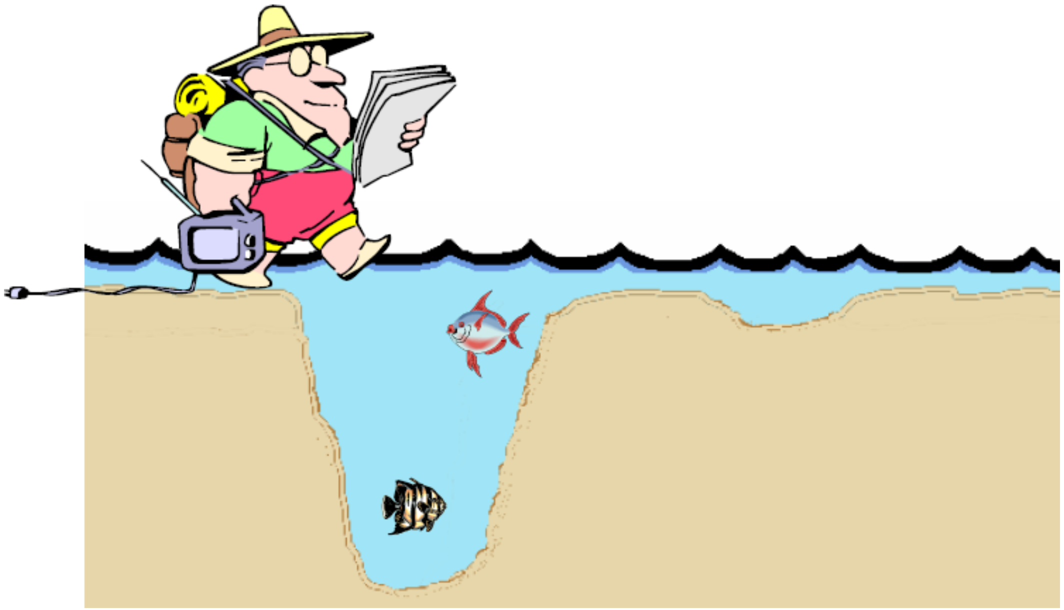


Figure taken from Issovic, D.:Real-time systems, basic course

Misconceptions about real-time systems

- Real-time ≠ fast: rather predictable than fast
  - Fast calculation: minimize average response time for entire system
  - Real-time: fulfil individual time constraints for each activity
  - "A man drowned in a river with an average depth of 20 centimeters"



Department of Distributed and Dependable Systems

Some other misconceptions...

- There is no science in real-time system design
  - We shall see...
- Advances in HW will take care of RT systems
  - Maybe better throughput, but no guarantee of the individual timing constraints
- Real-time programming implies assembly programming
  - Handcrafted assembly and device driver programming is major source of bugs
  - RT objective is to automate low-level programming
  - Application code written in C, Ada, even Java

What is so difficult about real-time systems?

- Lack of physical constraints
  - We can only measure time
- Lack of good models and methods
  - Relatively new area
- Timing constraints
  - All the problems as in non-RT systems + timing requirements
- Prediction of worst-case behavior
  - Efficiency is important, but safety is essential
- Concurrent control of separate system components
  - Parallel threads (tasks) running on the same CPU
- Complexity of modern HW architectures