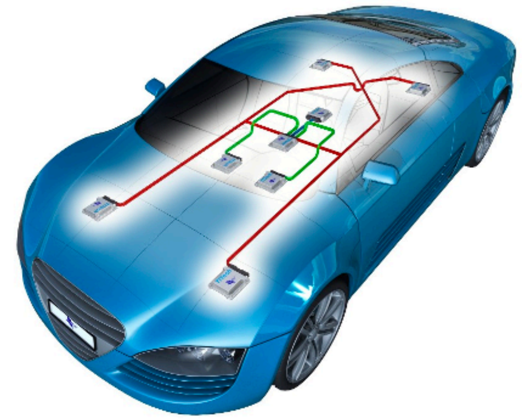






Real-Time Systems

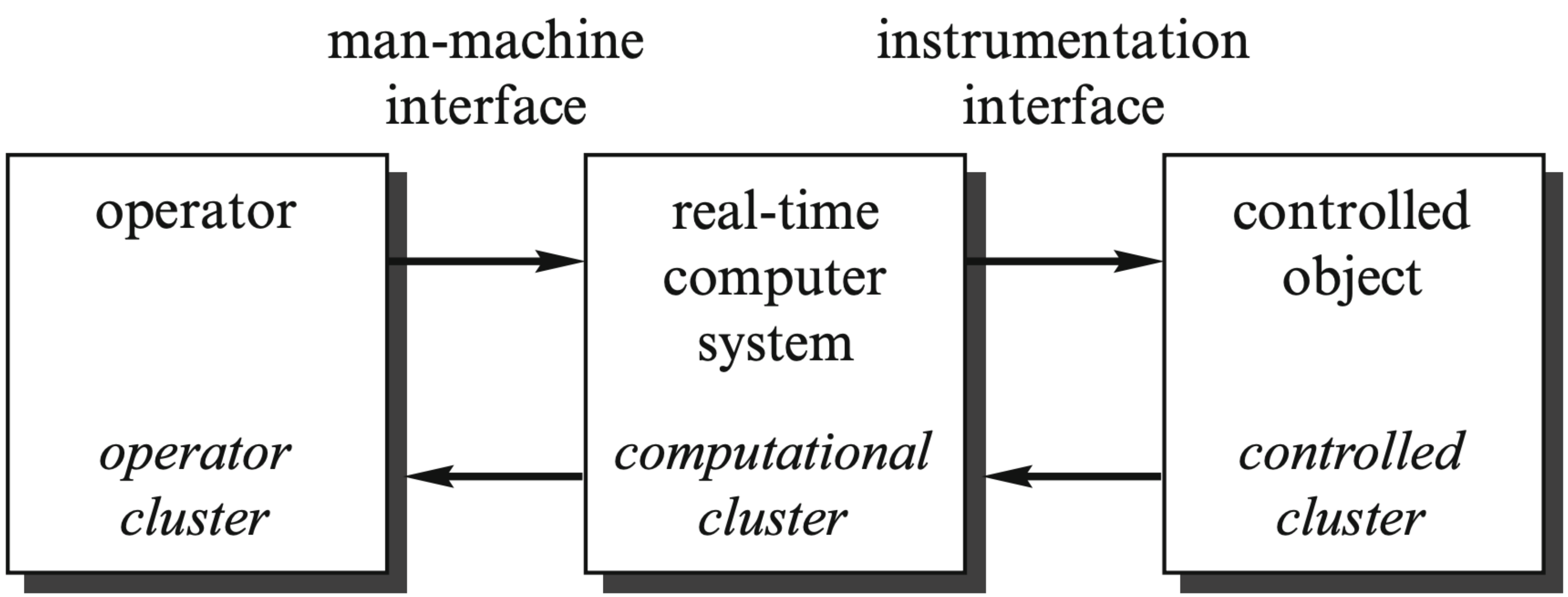
Jin Hyun Kim

What is a Real-Time Systems

- Real-Time Computer System: correct functioning depends on:
  - the values of the results produced, AND
  - the physical times at which the results are produced.
- Real-Time Computer System is embedded in a larger cyber-physical system.
- Some people prefer to call them as real-time systems or real-time embedded systems.

Real-Time Systems

- The correctness of system behaviors depends on
  - The logical results of the computations,
  - Physical instant at which these results are produced.
  - Time-dependent states

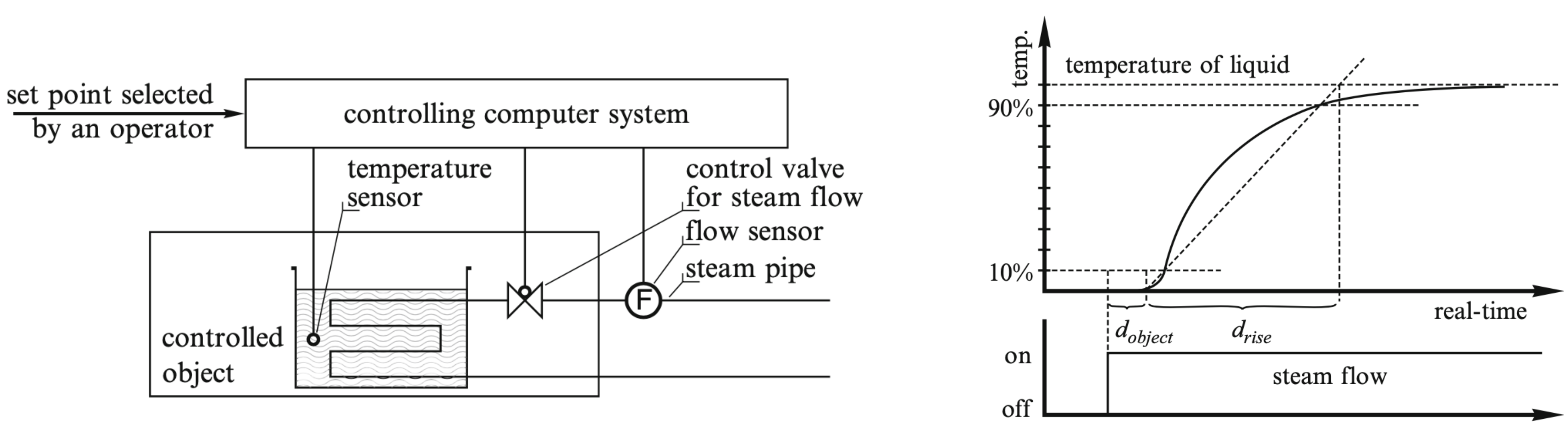


Real-Time Systems

- Time Constraints
- Dependability requirements

Where Do Temporal Requirements Come from?

- The most stringent temporal demands for real-time systems have their origin in the requirements of control loops,
    - e.g., in the control of a fast process such as an automotive engine.



A simple control loop

Properties of RTS

- A real-time computer system must react to stimuli from the controlled object (or the operator) **within time intervals** dictated by its environment.
- Deadline: instant by which a result must be produced.
  - Whose time?
  - What are different notions of time?

Real-Time

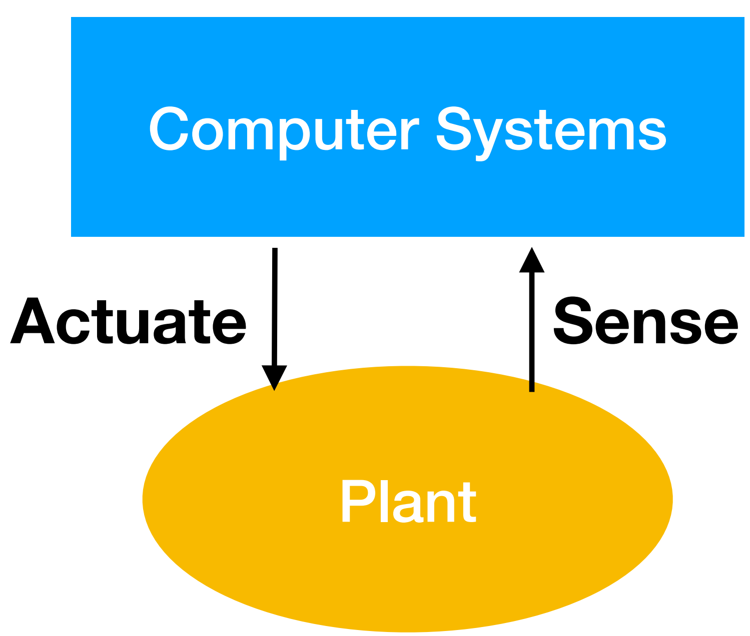
- Many tasks in distributed systems require **a common notion of time**
  - Different clocks have different clock rates,
    - Even worse, these clock rates may vary over time!
    - Communication is required to synchronize the clock

Real-Time

- Are the system time and the external physical time the same?
- Or, at least they must have a **predictable relationship**
    - bounded skew, bounded clock rates, etc.

State in RTS

- The state of the system evolves with time.
  - Position, velocity, acceleration
  - Pressure, temperature, level, concentration
- The state needs to be sensed and controlled by the computer system (in a control loop).



Temporal Constraints

- The computing system must sense, compute and actuate in a timely fashion.
- Many sensors and many actuators.
- Many control functions!
- Must **schedule computational tasks** for different control functions in a timely fashion.
- Computations must finish on time
  - QoS
  - End-to-end timing constraints

Temporal Constraints

- Real-time is about producing the correct result at the right time

Value	**Timing**	**Result**
Wrong	Too late	Failure
Wrong	On time	Failure
Correct	Too late	Failure
Correct	**On time**	**Ok**

- Temporal constraints are a way to specify when the value is on time

Temporal Constraints: Example

- Engine simulation: 1kHz recording frequency over a distributed system
- Clock synchronization: down to 1 nanosecond
- Industrial process control
    - Multiple, hierarchical control loops
- Anti-lock brakes
- Drive-by-wire
- Pacemakers
- Helicopter control
    - 200Hz pilot stick, 400Hz sensors, 200Hz flight control,1k Hz actuator electronics
- Games, user interactive systems

Types of Temporal Constraints

- Absolute temporal constraints
  - Measured with respect to a global clock
  - Xmas tree should light up between 5pm and 7am from November 27th 2006 until December 27th 2006
- Relative temporal constraints
  - Measured with respect to a local clock
  - The ventilation task should restart in five seconds
- Timing violation
  - Occurs when a temporal constraint is violated

Soft Temporal Constraints

- A soft real-time system is one where **the response time is normally specified as an average value.** **This time is normally dictated by the business or market.**

- A single computation arriving late is not significant to the operation of the system, though many late arrivals might be.

- Ex: Airline reservation system - If a single computation is late, the system's response time may lag. However, the only consequence would be a frustrated potential passenger

Hard Temporal Constraints

- A hard real-time system is one where **the response time is specified as an absolute value.** This time is normally dictated by the environment.

- A system is called **hard real-time** if tasks always must finish execution before their deadlines or if messages always are delivered within a specified time interval.

- Hard real-time is often associated with safety critical applications. A failure (e.g., missing a deadline) in a safety-critical application can lead to loss of human life or severe economical damage

Hard vs Soft RT

Characteristics	Hard RT	Soft RT
Response Time	Hard-Required	Soft-desired
Peak-Load-Performance	Predictable	Degraded
Control of Pace	Environment	Computer
Safety	Often Critical	Non-Critical
Size of Data Files	Small/Medium	Large
Redundancy Type	Active	Checkpoint-Recovery
Data Integrity	Short-term	Long-term
Error Detection	Autonomous	User assited

Desired Characteristics of Hard RTS

- Timeliness 시간준수
- Peak Load Handling 최대 사용량 조절
- Predictability 예측가능성
- Fault Tolerance - Robustness 강건성
- Maintainability

Dependability Requirements

- Reliability
- Safety
- Maintainability
- Availability

Reliability

- The Reliability $R(t)$ of a system is the probability that a system will provide the specified service until time t, given that the system was operational at the beginning, i.e., $t=t_0$.

- The probability that a system will fail in a given interval of time is expressed by the failure rate, measured in FITs (Failure In Time).

  - A failure rate of 1 FIT means that the mean time to a failure (MTTF) of a device is 10^9 h, i.e., one failure occurs in about 115,000 years.

Reliability

- If a system has a constant failure rate of λ failures/h, then the reliability at time t is given by

$$R(t) = \exp(-\lambda(t - t_o)) \quad = e^{-\lambda(t-t_0)}$$

  - where $t - t_0$ is given in hours.

- The inverse of the failure rate $1/\lambda$ = MTTF is called the Mean-Time-To-Failure MTTF (in hours).

- If the failure rate of a system is required to be in the order of 10^{-9} failures/h or lower, then we speak of a system with an ***ultrahigh reliability*** requirement.

Safety

- Safety is reliability regarding critical failure modes
  - **The cost of a failure** can be orders of magnitude higher than the utility of the system during normal operation.
    - E.g., An airplane crash due to a failure in the flight-control system, and an automobile accident due to a failure of a computer-controlled intelligent brake in the automobile
- Safety- critical (hard) real-time systems must have a failure rate with regard to critical failure modes that conforms to the **ultrahigh reliability** requirement.
- Certification - In many cases the design of a safety-critical real-time system must be approved by an independent certification agency

Maintainability

- Maintainability is a measure of the time interval required to repair a system after the occurrence of a benign failure.
- A constant repair rate μ (repairs per hour) and a Mean Time to Repair (MTTR) are introduced to define a quantitative maintainability measure.

Availability

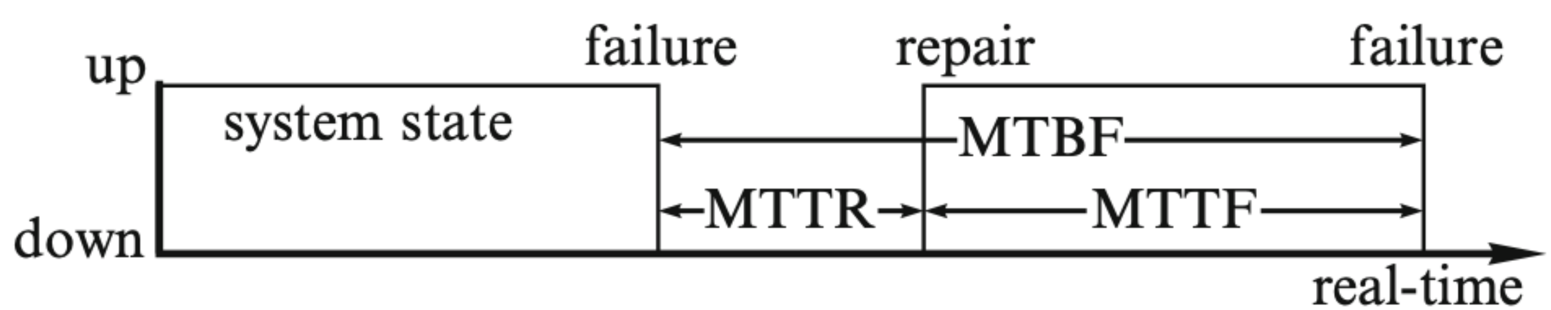
- Availability is a measure of the delivery of correct service with respect to the alternation of correct and incorrect service.
    - It is measured by the fraction of time that the system is ready to provide the service.

Availability

- In systems with constant failure and repair rates, the reliability (MTTF), maintainability (MTTR), and availability (A) measures are related by

$$A = MTTF/(MTTF + MTTR)$$

  - The sum MTTF + MTTR is sometimes called the Mean Time Between Failures (MTBF)



Security

- CIA
  - Confidentiality (기밀성)
  - Availability (가용성)
  - Integrity (무결성, 위변조방지)
- Accountability (책임추적성), 접근통제 (Access Control), Authentication (인증), Non-Reputation (부인방지)

Conclusions

- RTS is a system where timely manner is a critical concern for system correctness
- Real-time requirements and constraints are from the control loop where computer system and the environment interact with each other
- Hard and Soft RT constraints
- Dependability - Reliability, Safety, Availability, Maintainability, (Security)