CT420 REAL-TIME SYSTEMS

REAL-TIME SYSTEMS AND SAFETY-CRITICAL SYSTEMS

Dr. Michael Schukat

OÉ Gaillimh
NUI Galway

Lecture Overview

- Real-time systems versus safety-critical systems versus real-time safety-critical systems (RTSCS)
- Real-time systems categories
- RTSCS quality requirements
- RTS and RTSCS case studies

Recap: Real-Time Systems

- *A system is said to be real-time if the total correctness of an operation depends not only upon its **logical correctness**, but also upon **the time** in which it is performed*
  - Functional (logical correctness) versus non-functional (time constraints) requirements
  - Fancy example: Boston Dynamics robots, see https://www.youtube.com/watch?v=rVlhMGQgDkY

Response Time of an RTS

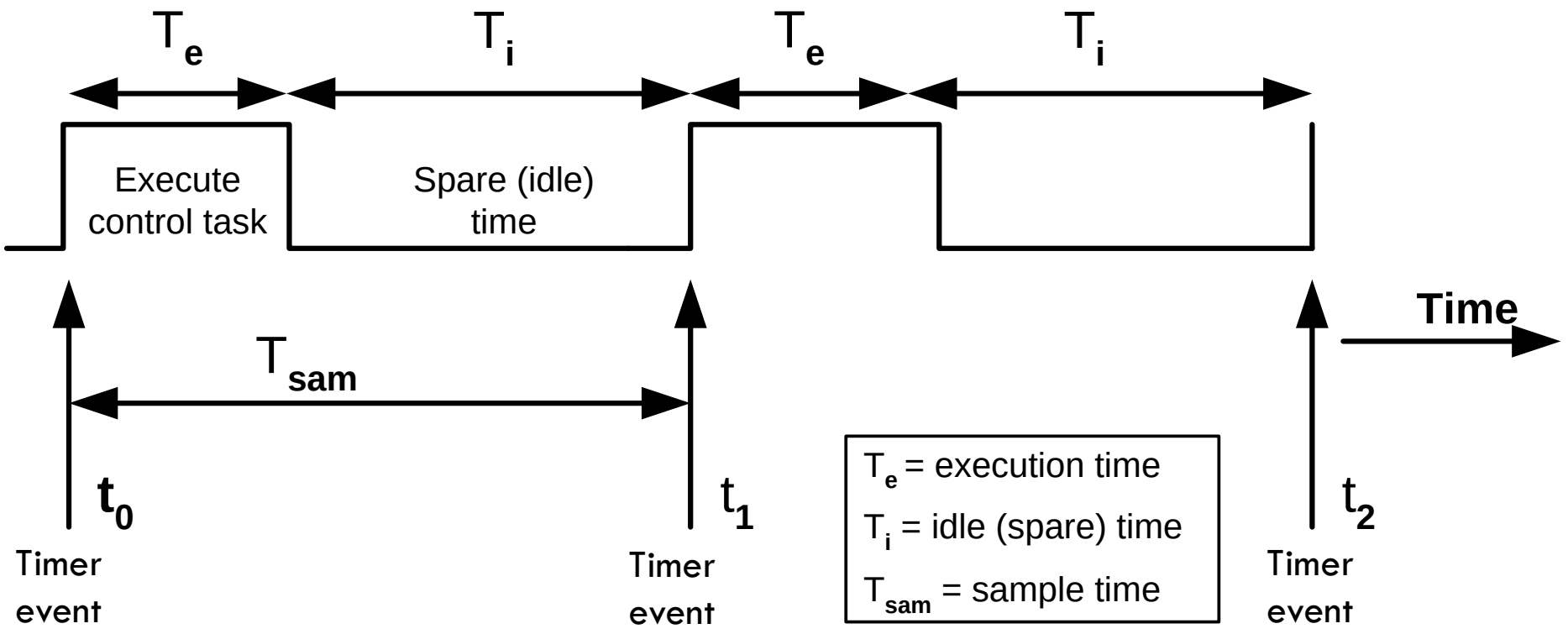
- The time between
  - the presentation of a set of inputs to a system (stimulus)
  - and the realisation of the required behaviour (response)

  is called the **response time** of the system
- The response time must be
  - bounded (i.e. response time must < x milliseconds)
  - fully deterministic (guaranteed bounded)
- Average response time is not sufficient, e.g.
  "A statistician drowned while crossing a stream that was, on average, 1 cm deep"

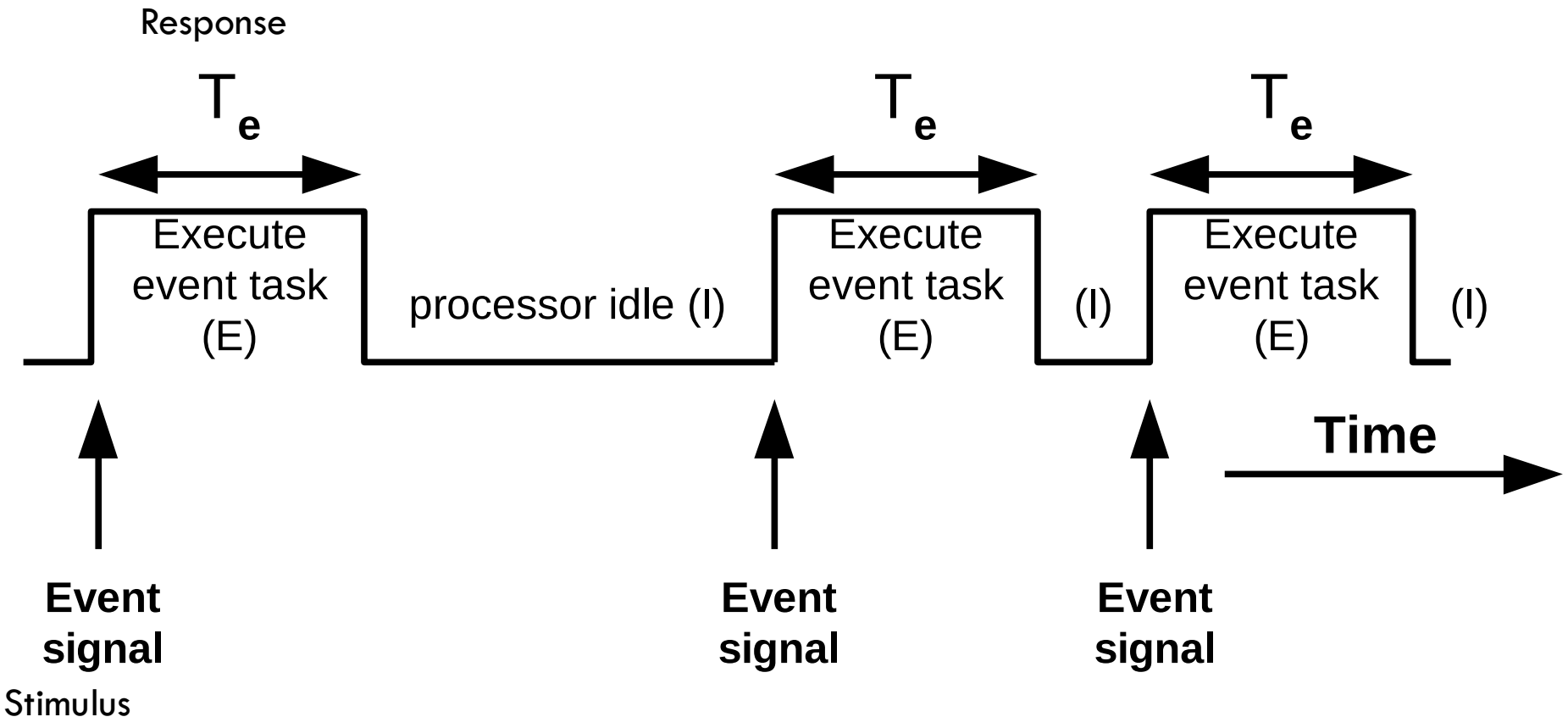
The Issue of Time Responsiveness

- Programmatically a stimulus is often provided by a hardware or software interrupt, i.e.,
  - Synchronously and repeatedly by a timer unit
  - Asynchronously by some event (e.g. impact sensor of airbag)
- The response is the execution of some code
  - Interrupt service routines, functions, or tasks / processes
- This has to happen in a timely fully deterministic and bounded fashion, despite having potentially
  - other (non-RT) tasks / processes running in the background
  - having to deal with multiple simultaneous synchronous / asynchronous events

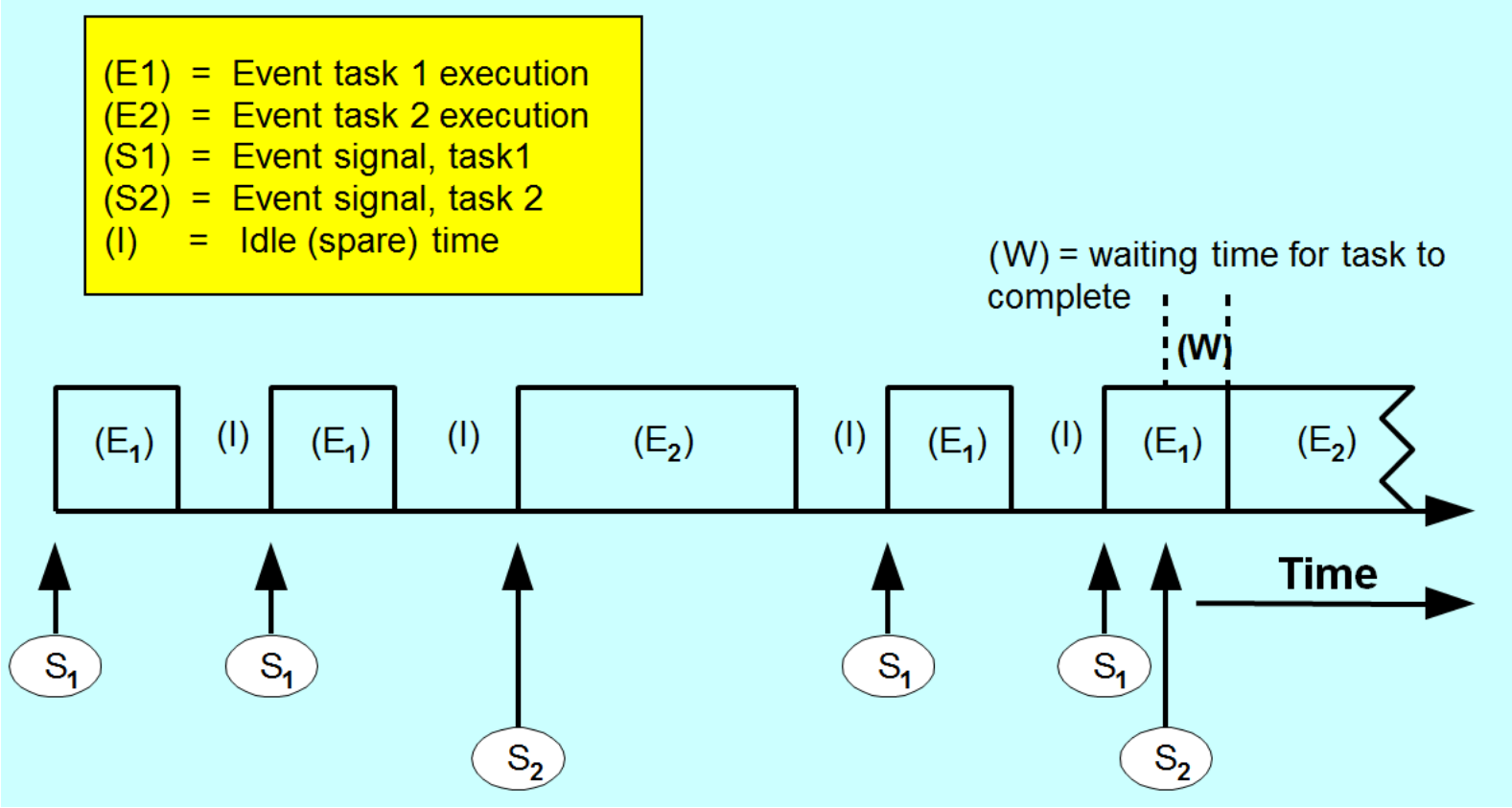
Task Model of an RTS - single synchronous (periodic) Task



Task Model of an RTS - single asynchronous (aperiodic) Task

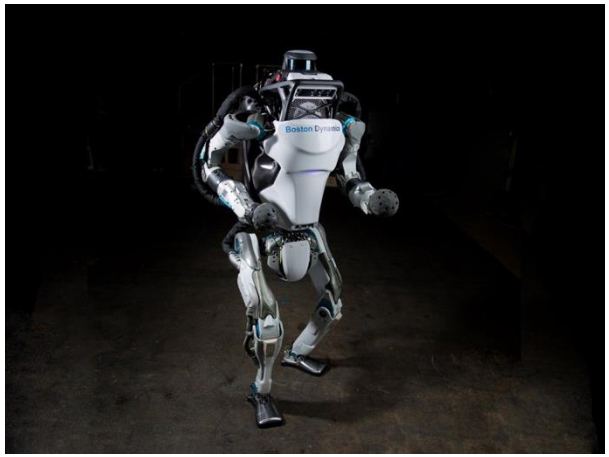


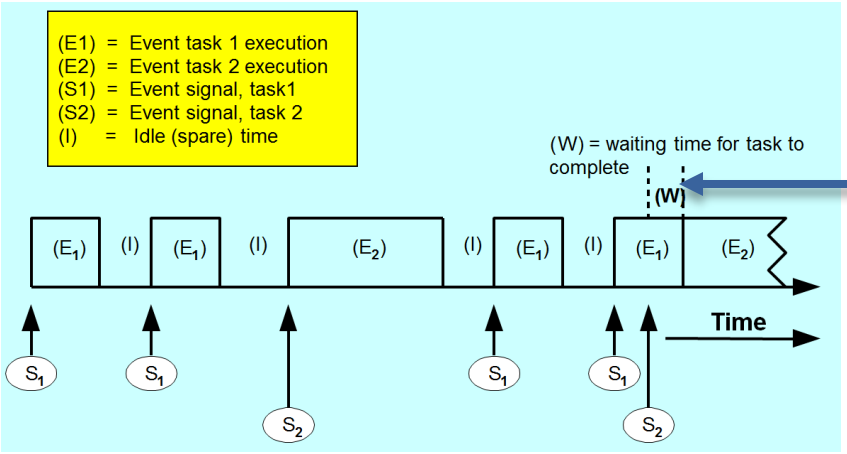
Task Model of an RTS - multiple asynchronous (aperiodic) Tasks



Example

- Consider a two-legged walking robot that is controlled by a single CPU
- The robot control software deals with two asynchronous events (triggered by some hardware sensors)
  - (S1): Robot foot touches ground
  - (S2): Robot out of balance

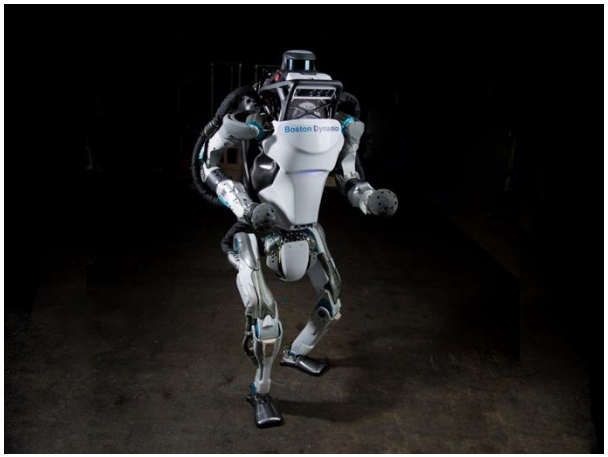


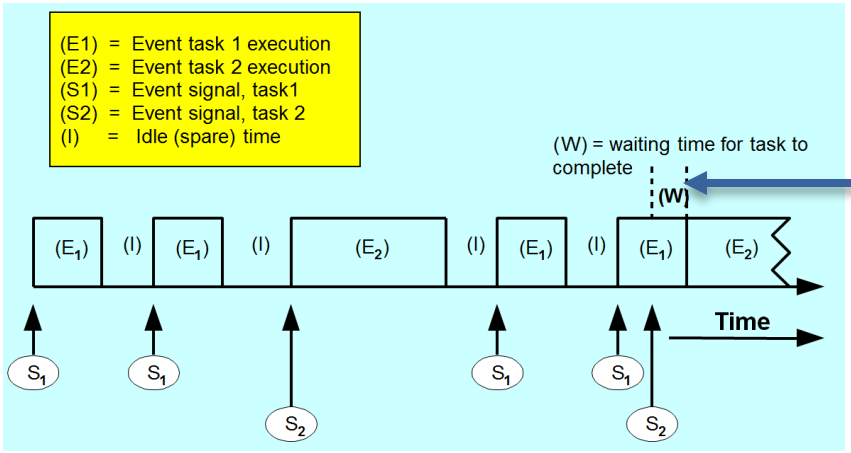


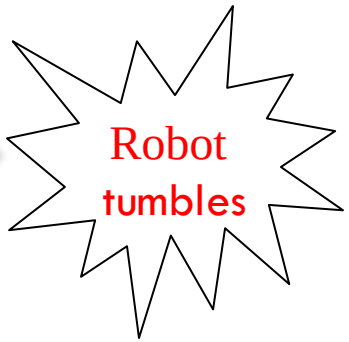
Robot tumbles

Example

- What is the an upper (acceptable) boundary for W (response time)?







Task Structure of an RTS

- In may RT applications we typically find:
  - One or more synchronous (periodic) tasks running at different frequencies
  - multiple asynchronous (aperiodic) tasks running at non-deterministic times
- **Fundamental problem: How can timeliness be guaranteed for such a system?**

RTS Categories

- Hard
  A Hard RTS is one in which failure to meet a single deadline may lead to complete and catastrophic system failure
- Firm
  Infrequent deadline misses are tolerable, but may degrade the system's quality of service. The usefulness of a result is zero after its deadline
- Soft
  A Soft RTS is one in which performance is degraded but not destroyed by failure to meet response time constraints

Soft RTS

- These are the easy ones, i.e. a Soft RTS is one in which performance is degraded but not destroyed by failure to meet response time constraints
- Examples
  - Multimedia devices and networks
    - E.g. out-of-sync dubbing, no lip-sync between audio and video
  - Real-time communications (example, VoIP, e.g. Skype)
    - E.g. poor video quality, or no lip-sync

RTS Categories

	SLOW	FAST
SOFT	Machinery condition monitoring	Man-Machine Interfacing
HARD	Missile point defence system	Airbag control system

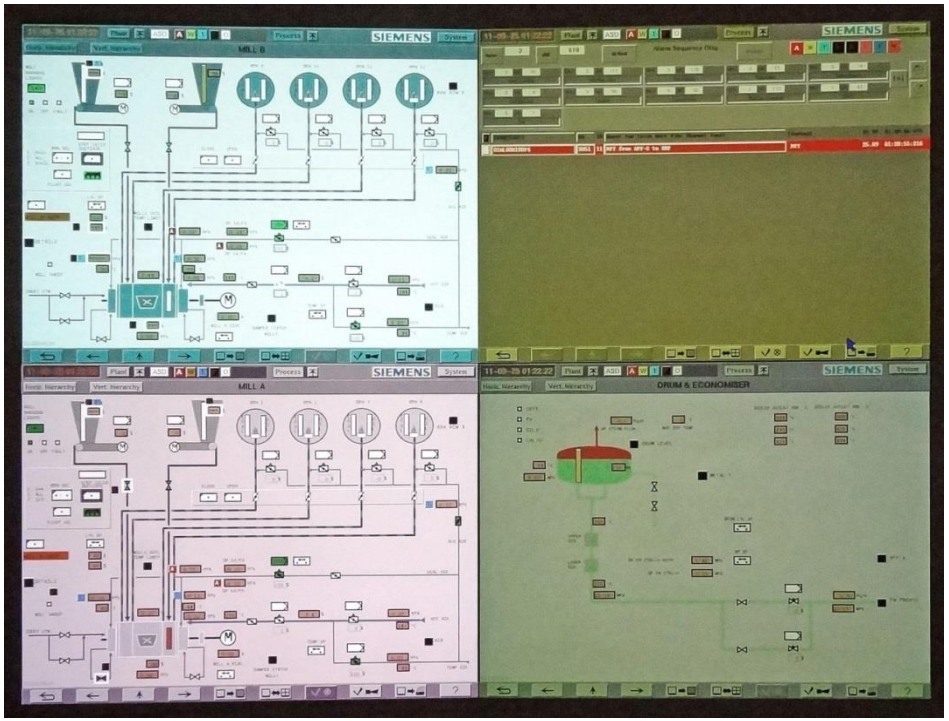
- Two major categorization factors: criticality and speed
  - Criticality:
    - Hard systems - deadlines (responsiveness) is critical. Failure to meet these have severe to catastrophic consequences (e.g. injury, damage or death)
    - Soft systems - deadlines are less critical; in many cases significant tolerance can be permitted
  - Speed:
    - Fast systems - responses in the microseconds to hundreds of milliseconds
    - Slow systems - responses in the range seconds to days

Example: Missile Point Defence System (hard-slow RTS)

- High-speed jet aircraft flying at low altitudes present a serious threat to naval forces
- Approaching under the radar of the ships, the aircraft would suddenly appear at relatively close ranges, giving the ships only seconds to respond (using their missile defense system) before the aircraft dropped their payloads and withdrew

Example: SCADA Front-End (soft-fast RTS)

- Supervisory control and data acquisition (SCADA) is a system of software and hardware elements that allows controlling industrial processes locally or remotely by monitoring and processing real-time data



Attributes of RTS

	Execution time	Deadlines	Software size	Software complexity
Hard - Fast	●●●●	●●●●	●	●
Hard - Slow	●	●●●●	● → ●●●	● → ●●●●
Soft - Fast	●●●●	●●	● → ●●●	● → ●●●
Soft - Slow	●●	●●	● → ●●●●	● → ●●●●

Attribute rating
● Low ●●●● high

Safety-Critical Systems

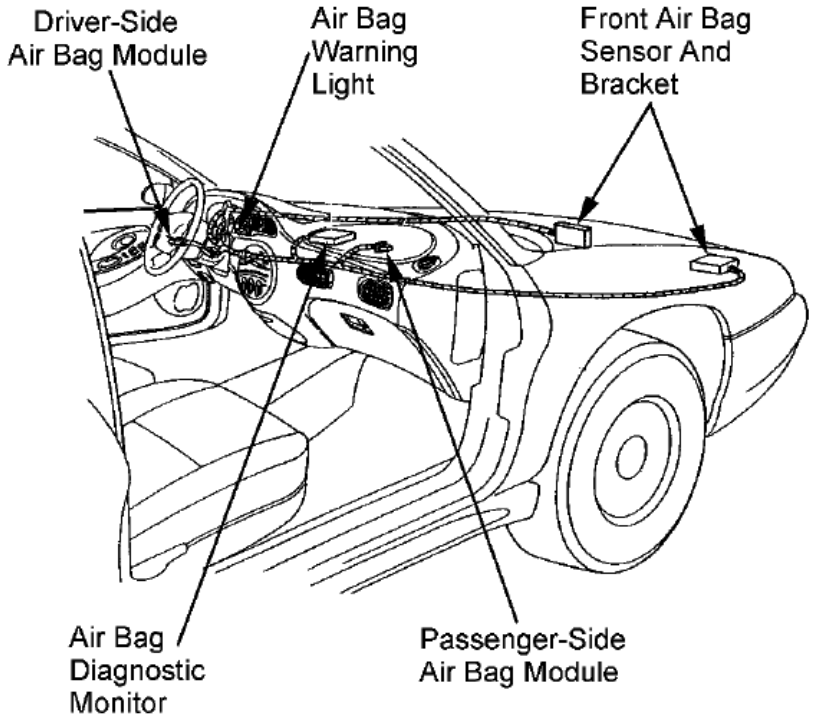
- Def. Wikipedia: A safety-critical system or life-critical system is a system whose failure or malfunction may result in one (or more) of the following outcomes:
  - death or serious injury to people
  - loss or severe damage to equipment/property
  - environmental harm
- Sounds like a hard-fast / hard-slow RTS
  - i.e. **Real-Time Safety-Critical System (RTSCS)**

Quality Requirements for RTSCS

- RTSCS must be **time responsive**
- RTSCS must be **reliable**
  - The ability to behave in accordance with its specification
- RTSCS must be **safe**
  - Conditions that lead to hazards do not occur
- RTSCS must be **secure**
  - Protect itself against intentional or accidental access, use, modification or destruction
- RTSCS must be **usable**
  - Easy to learn, understand, and use
- RTSCS must be **maintainable**
  - Return swiftly to an operational state after receiving repairs or modification (e.g. plug in-and-forget)

RTSCS Case Study: The Airbag

- How does an Airbag work?
  - https://www.youtube.com/watch?app=desktop&v=ZEWCFjbqaaE
- What are the RT-requirements of an Airbag?
  - https://www.youtube.com/watch?v=QS6ywFGcLSk
- See video:
  A failed Airbag
- What are your thoughts re the RCTCS quality requirements
  - Time responsiveness
  - Reliability
  - Safety



Possible Causes for Airbag Failure

- Mechanical fault: Airbag sensor detached from chassis / bracket
- Design fault: Airbag sensor is incorrectly positioned in this model
- Design fault: Single sensor only, no redundancy or voter
- Firmware design fault: Airbag trigger based on data from **single** sensor (rather than from multiple sensors)