Dependability Overview

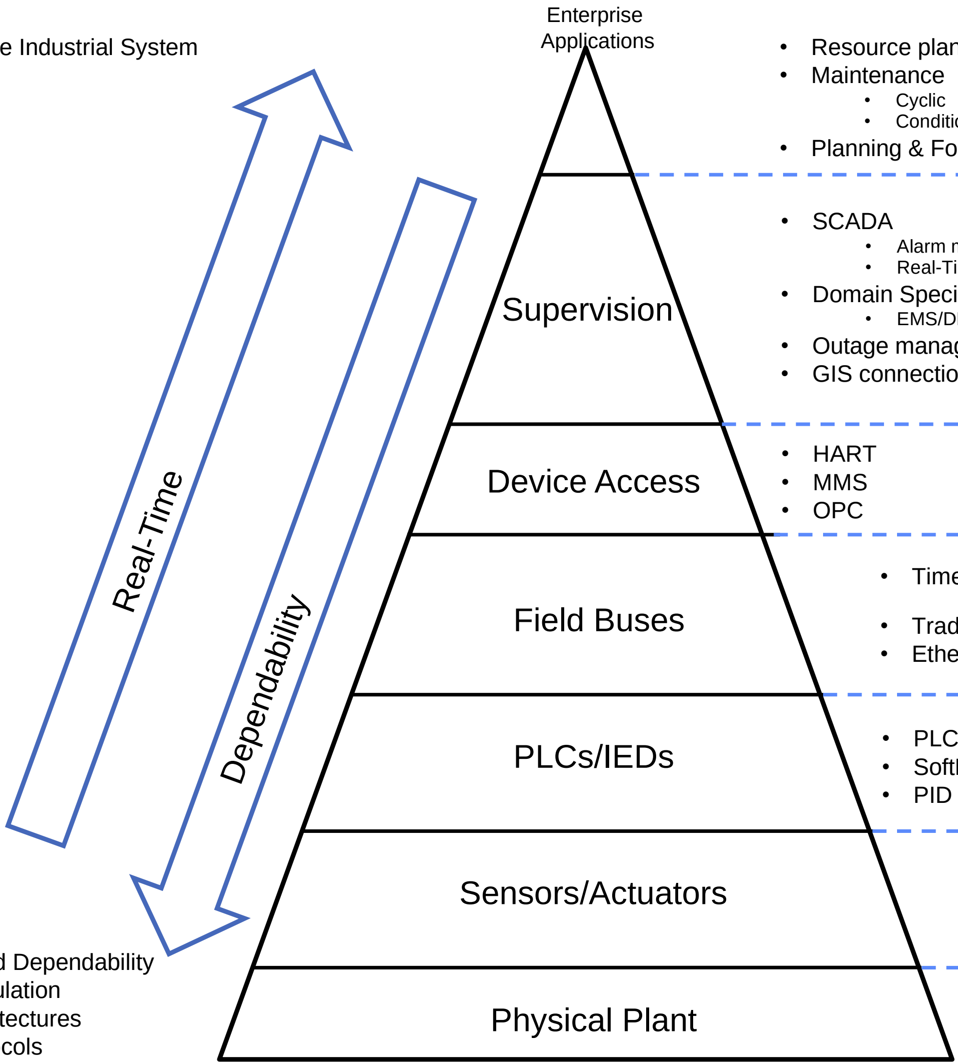
Industrial Automation

The material of this course has been initially created by Prof. Dr. H. Kirrmann and adapted by Dr. J-C. Tournier.

Dr. Jean-Charles TOURNIER

Spring 2025

- Real Time Industrial System



- Resource planning
- Maintenance
  - Cyclic
  - Condition-based
- Planning & Forecasting

- SCADA
  - Alarm management (EEMU 191)
  - Real-Time Databases
- Domain Specific Applications
  - EMS/DMS
- Outage management
- GIS connections

- HART
- MMS
- OPC

- Time Synchronization
  - PPS, GPS, SNTP, PTP, etc.
- Traditional - Modbus, CAN, etc.
- Ethernet-based - HSR, WhiteRabbit, etc.

- PLC
- SoftPLC
- PID

- Instrumentation
- 4-20 mA loop
- Sensors accuracy
- Examples (CT/VT, water, gaz, etc.)

- Plant examples
- Why supervision/control?

- Reliability and Dependability
  - Calculation
  - Architectures
  - Protocols

Dr. Jean-Charles Tournier

Schedule

08-April-2025

- 1: Overview Dependable Systems
  - Definitions: Reliability, Safety, Availability etc.,
  - Failure modes in computers
- 2: Dependability Analysis
  - Combinatorial analysis
  - Markov models
- 3: Dependable Architectures
  - Fault detection
  - Redundant Hardware, Recovery
- 4: Dependable Software
  - Fault Detection,
  - Recovery Blocks, Diversity
- 5: Safety analysis
  - Qualitative Evaluation (FMEA, FTA)
  - Examples

Schedule

- 1: Overview Dependable Systems
  - Definitions: Reliability, Safety, Availability etc.,
  - Failure modes in computers
- 2: Dependability Analysis
  - Combinatorial analysis
  - Markov models
- 3: Dependable Architectures
  - Fault detection
  - Redundant Hardware, Recovery
- 4: Dependable Software
  - Fault Detection,
  - Recovery Blocks, Diversity
- 5: Safety analysis
  - Qualitative Evaluation (FMEA, FTA)
  - Examples

6-May-2025

Schedule

- 1: Overview Dependable Systems
  - Definitions: Reliability, Safety, Availability etc.,
  - Failure modes in computers
- 2: Dependability Analysis
  - Combinatorial analysis
  - Markov models
- 3: Dependable Architectures
  - Fault detection
  - Redundant Hardware, Recovery
- 4: Dependable Software
  - Fault Detection,
  - Recovery Blocks, Diversity
- 5: Safety analysis
  - Qualitative Evaluation (FMEA, FTA)
  - Examples

13-May-2025

Schedule

- 1: Overview Dependable Systems
  - Definitions: Reliability, Safety, Availability etc.,
  - Failure modes in computers
- 2: Dependability Analysis
  - Combinatorial analysis
  - Markov models
- 3: Dependable Architectures
  - Fault detection
  - Redundant Hardware, Recovery
- 4: Dependable Software
  - Fault Detection,
  - Recovery Blocks, Diversity
- 5: Safety analysis
  - Qualitative Evaluation (FMEA, FTA)
  - Examples

20-May-2025

20-May-2025

Dependable Systems, Why?

- Systems - if not working properly in a particular situation - may cause
  - large losses of property
  - injuries or deaths of people
  - environmental disaster
- Failures <u>**are**</u> unavoidable, “mission-critical” or “dependable” systems are designed to fail in such a way that a given behaviour is guaranteed.
- The necessary precautions depend on
  - the probability that the system is not working properly
  - the consequences of a system failure
  - the probability of occurrence of a dangerous situation
  - the negative impact of an accident (severity of damage, money lost)

Domains for Dependable Systems

Space Applications	Launch rockets, Shuttle, Satellites, Space probes
Transportation	Airplanes (fly-by-wire), Railway signalling, Traffic control, Cars (ABS, ESP, brake-by-wire, steer-by-wire)
Nuclear Applications	Nuclear power plants, Nuclear weapons, Atomic-powered ships and submarines
Networks	Telecommunication networks, Power transmission networks, Pipelines
Finance	Electronic stock exchange, Electronic banking, Data stores for Indispensable business data
Medical	Irradiation equipment, Life support equipment, Technology assisted surgery
Industrial Processes	Critical chemical reactions, Drugs, Food

Definitions

EPFL

Definitions
Mission, Fault, Error, Failure

- ***Mission*** is the required (intended, specified) function of a **device** during a given time.

- ***Fault*** : abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function.
  (Fehler, *en panne*, falla) - it is a state

- ***Error***: logical manifestation of a fault in an application
  (Fehler, *erreur*, error)
  "discrepancy between a computed, observed or measured value or condition and the true, specified or theoretically correct value or condition" (IEC 61508-4)

- ***Failure****:* is the termination of the ability of a device to perform its required function.
  (Ausfall, défaillance, avería) – it is an event.

- These terms can be applied to the whole system, or to elements thereof.

Fault

- *Fault is an abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function.*
- In other words, a fault is a defect within the system
- Examples:
  - Software bug
  - Random hardware fault
  - Memory bit “stuck”
  - Omission or commission fault in data transfer

Error

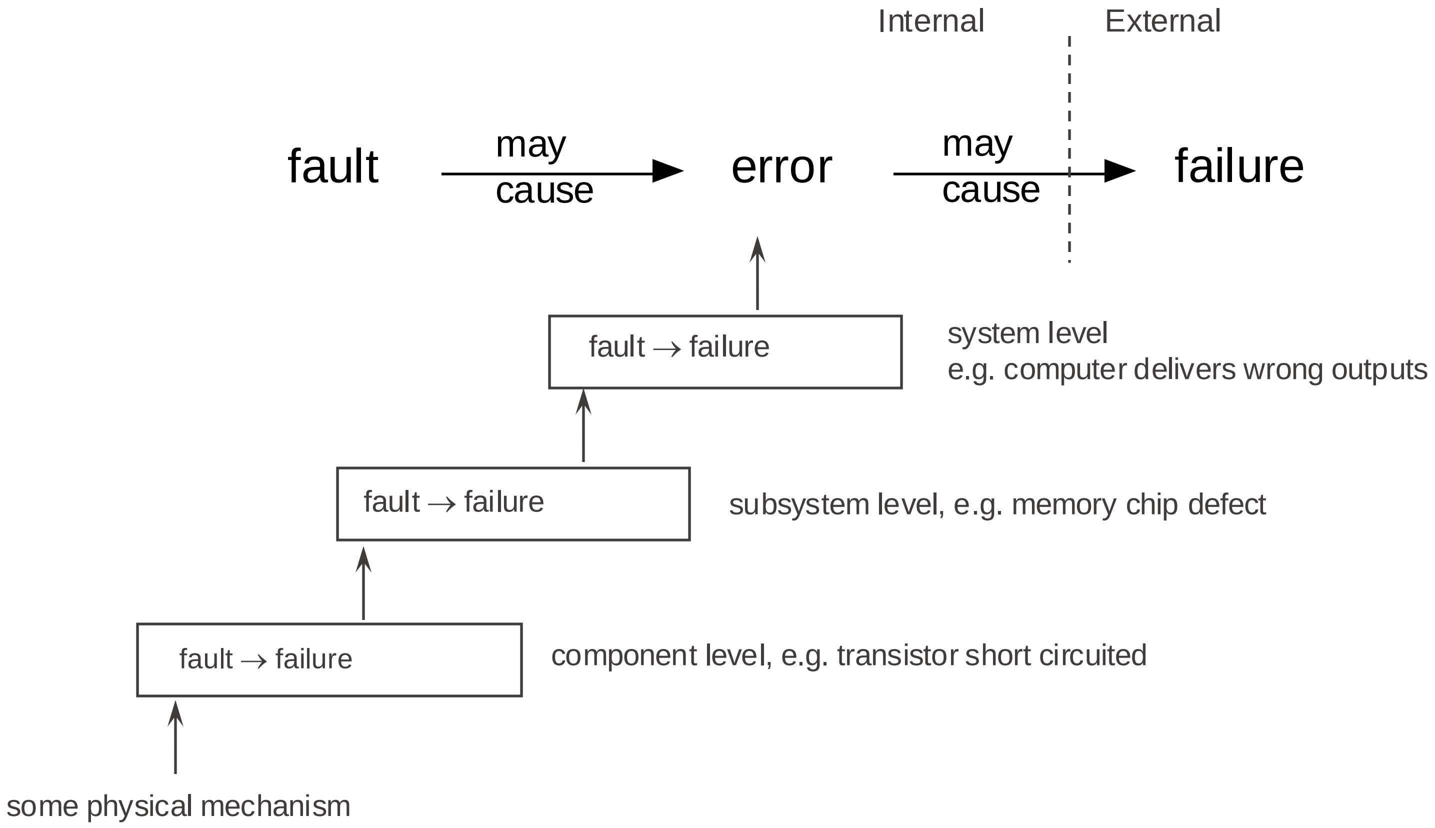
- *Error is a deviation from the required operation of system or subsystem*
  - *discrepancy between a computed, observed or measured value or condition and the true, specified or theoretically correct value or condition*
- A fault may lead to an error, i.e., error is a mechanism by which the fault becomes apparent
  - Fault activation
- Fault may stay dormant for a long time before it manifests itself as an error, therefore the term lurking fault is sometimes used in this case
- Example:
  - Faulty memory bit but CPU does not access this data
  - Broken mechanical spring in a breaker (power system protection)
  - Software “bug” in functions is not apparent until it is called

Failure

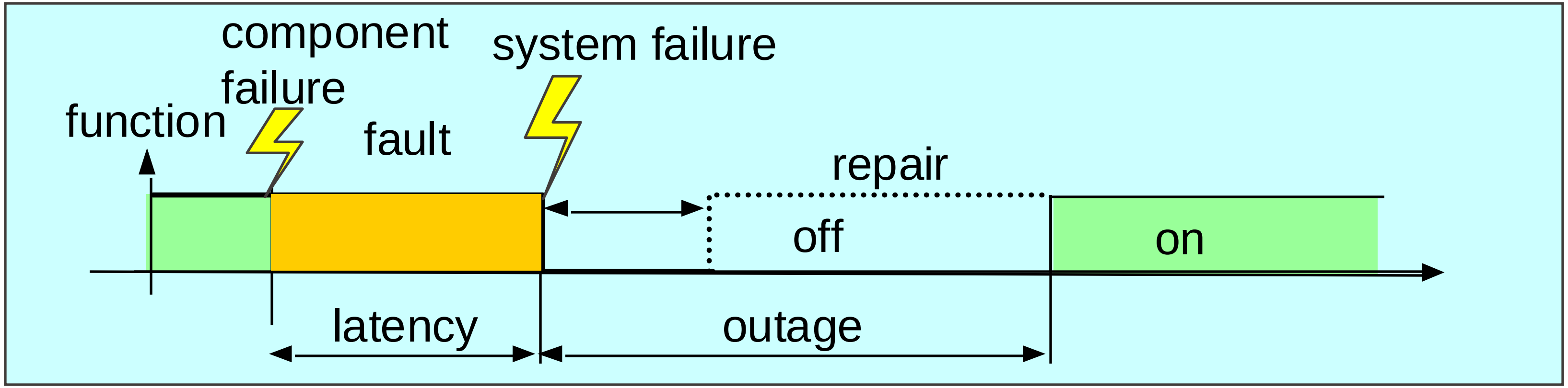
- *Failure is the termination of the ability of an item/device to perform its required function*

- A system failure occurs when the system fails to perform its required function
- Presence of an error might cause a whole system to deviate from its required operation
- An error does not necessary cause a failure
  - e.g. exception caught and handled properly in a software
- Main goal of safety-critical systems is that error should not result in system failure

Causality Chain



Definitions
Fault, Error, Failure



see International Electrotechnical Vocabulary, [IEV 191-05-01] http://std.iec.ch/iev

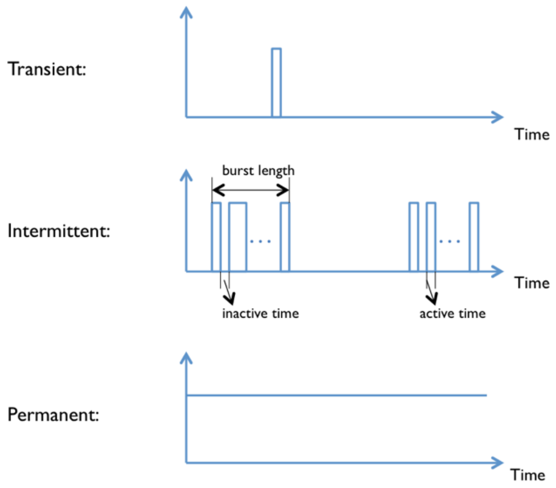
Fault Characteristics

- Fault: missing or wrong functionality (Fehler, *faute*, falla)

- Fault can be characterized either from a temporal or consistency point of view

- **Temporal** characteristics of a fault:
  - momentary = outage
  - temporary = breakdown - for repairable systems only
  - definitive

- **Consistency** characteristics of a fault**:**
  - permanent: due to irreversible change, consistent wrong functionality (e.g. short circuit between 2 lines)
  - intermittent: sometimes wrong functionality, recurring (e.g. loose contact)
  - transient: due to environment, reversible if environment changes (e.g. electromagnetic interference)



Fault Types

Systems can be affected by two kinds of faults:

physical faults

(e.g. hardware faults)

A corrected physical fault can occur again with the same probability.

design faults

(e.g. software faults)

A corrected design fault does not occur anymore.

Physical faults can originate from design faults (e.g. missing cooling fan)
Design faults can lead to physical faults (e.g. wrong regulation of a fan => over-speed)

Dr. Jean-Charles Tournier

Random and Systematic Errors

- **Systematic errors** are reproducible under given input conditions

  => from permanent fault

- **Random Error** appear with no visible pattern.

  => from intermittent fault

- Although random errors are often associated with hardware errors and systematic errors with software errors, this may not be the case

Dr. Jean-Charles Tournier

Transient Errors

- Transient errors leave the hardware undamaged.
- For instance, electromagnetic disturbances can jam network transmissions
  - Restarting work on the same hardware can be successful.

- A transient error can however be latched if it affects a memory element
- For example, cosmic rays can change the state of a memory cell, in which case one speaks of firm errors or soft errors.

Random Faults

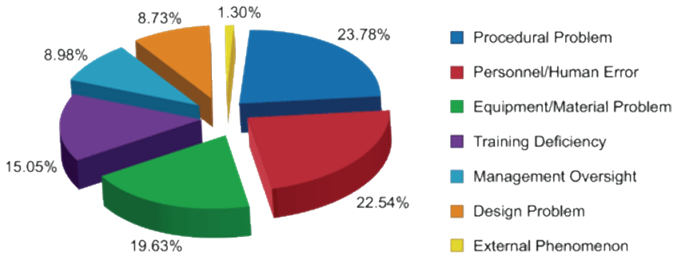
- Random faults are (usually) associated with hardware components
- When working within their correct operating environment, individual components fail randomly
- **All physical components are subject to failure**
  => all systems are subject to random faults

Random Faults

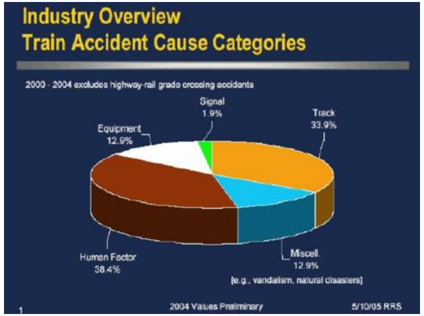
- Objectives
  - gather statistical data on large number of similar devices
  - Make prediction of the probability of a component failing within a given period of time
  - Use it to predict the overall performance of the system
  - Implement mechanism to survive random fault
    - Purpose of **fault-tolerant system**

Examples

Sources failure in a factory

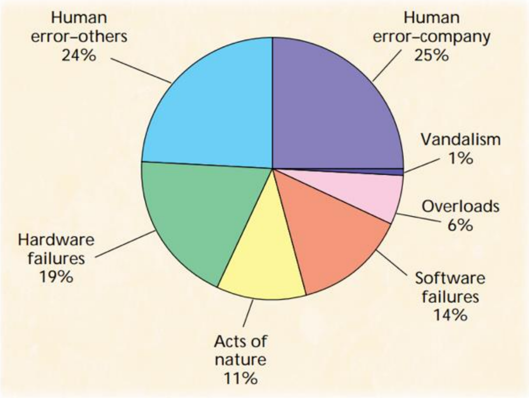


https://www.machinerylubrication.com/Read/23904/human-factors-engineering-reliability



Roop, Stephen & Morgan, Curtis & Kyte, Tobin & Arthur, Jr. Winfred & Villado, Anton & Beneigh, Theodore. (2007). Rail Crew Resource Management (CRM): The Business Case for CRM Training in the Railroad Industry. 10.13140/RG.2.1.4242.5208.

Sources failure in a public switched network



https://www.computer.org/csdl/magazine/co/1997/04/r4031/13rRUwwaKn4

Dr. Jean-Charles Tournier

Basic Concepts

Basic Concepts

- **dependability**: (*sûreté de fonctionnement*, Verlässlichkeit, seguridad de funcionamiento) collective term used to describe the availability performance and its influencing factors: reliability performance, maintainability performance and maintenance support performance.

- **availability** (*disponibilité*, Verfügbarkeit, disponibilidad):
  ability of an item to be in a state to perform a required function under given conditions at a given instant of time or over a given time interval assuming that the required external resources are provided.

- **reliability** (*fiabilité*, Zuverlässigkeit, fiabilidad):
  ability of an item to perform a required function under given conditions for a given time interval without failure

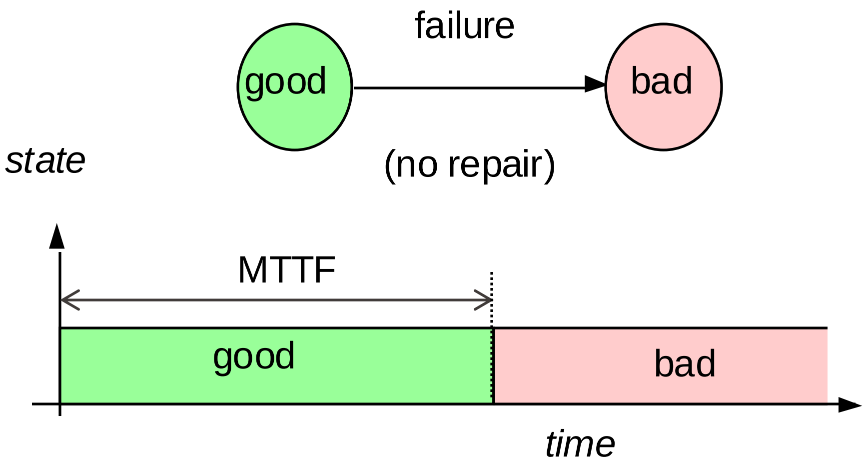
- **maintainability** (*maintenabilité*, Instandhaltbarkeit, mantenabilidad
  ability of an item under given conditions of use, to be retained in, or restored to, a state in which it can perform a required function , when maintenance is performed under given conditions and using state procedures and resources.

…. other dependability concepts:

safety (*sûreté*, Sicherheit, seguridad)
acceptable risk

security (sécurité informatique, Datensicherheit, seguridad informática)
danger to data, particularly confidentiality, proof of ownership and traffic availability

Reliability

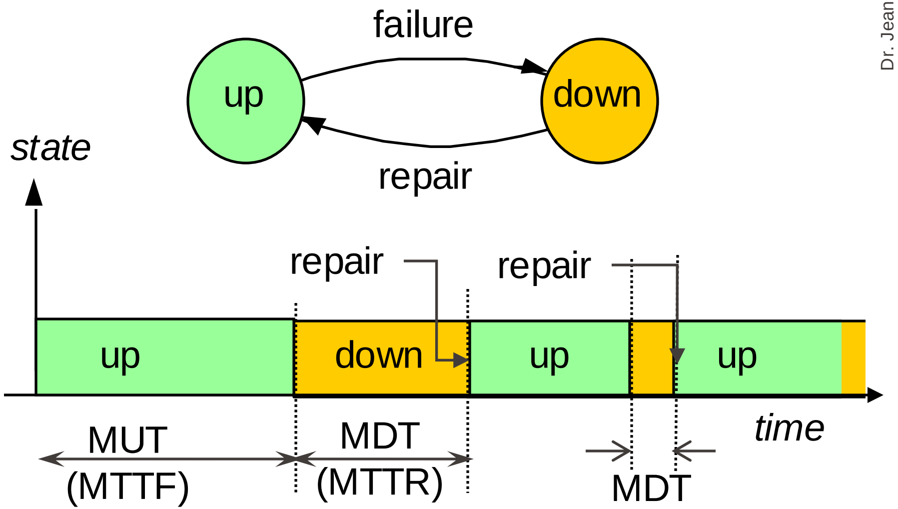


definition: "probability that an item will perform its required function in the specified manner and under specified or assumed conditions *over a given time period* ***without failure***"

Thus: reliability is a function of time → R(t),

expressed shortly by its MTTF: Mean Time To Fail

Availability



definition: "probability that an item will perform its required function in the specified manner and under specified or assumed conditions *at a given time or over a time period*"

expressed shortly by the *stationary availability*

$$A_\infty = \frac{MUT}{MUT + MDT}$$

or better its unavailability (1-A), e.g. 2 hours /year.

Repairable Systems

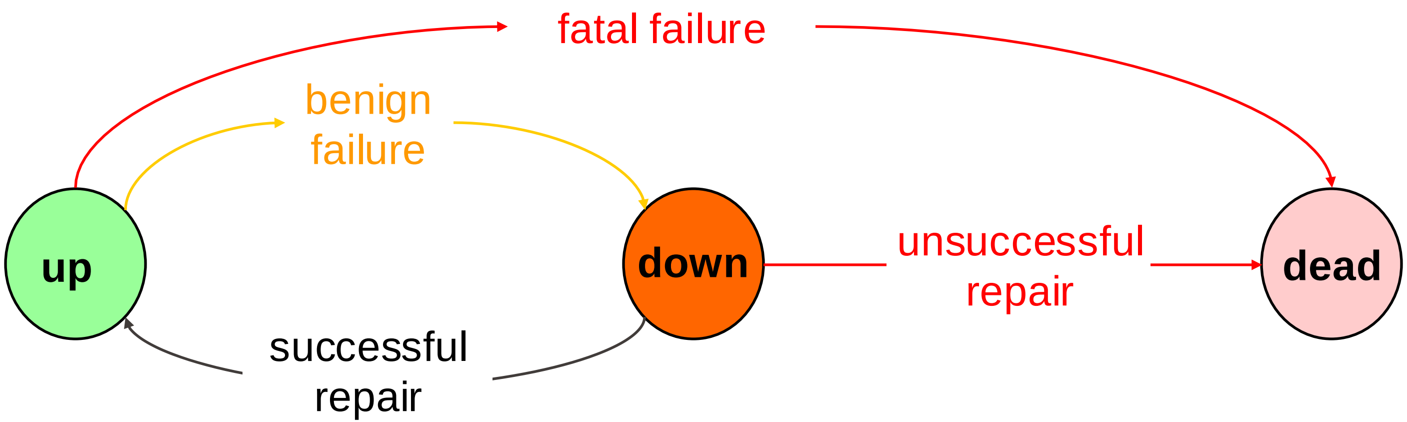
Dr. Jean-Charles Tournier

- **It is not the system that is available or reliable, it is the model of the system!**
- Considering first only benign failures (the system oscillates between the “up” and “down” states), one is interested in:
  - how much of its life does the system spend in the “up” state (Availability) and
  - how often does the transition from up to down take place (Reliability)
- For instance, a car has an MTBF (mean time between failure) of e.g. 8 months and needs two days of repair. Its availability is 99,1 %. If the repair shop works twice as fast, availability raises to 99.6%, but reliability did not change – the car still goes on the average every 8 months to the shop.
- Considering now fatal failures (the system has an absorbing state “dead”), one is interested only in how much time in the average it remains in the repairable states (“up” + “down”), its MTTF (Mean Time To Fail), is e.g. 20 years, its availability is not defined.

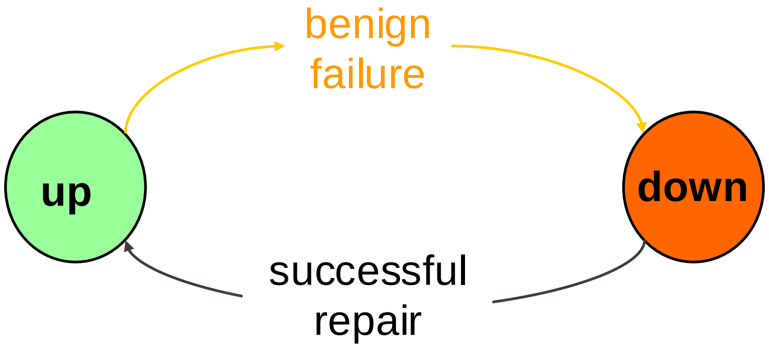
Repairable Systems

Dr. Jean-Charles Tournier

Reliability



Availability



Availability and Repair in Redundant Systems

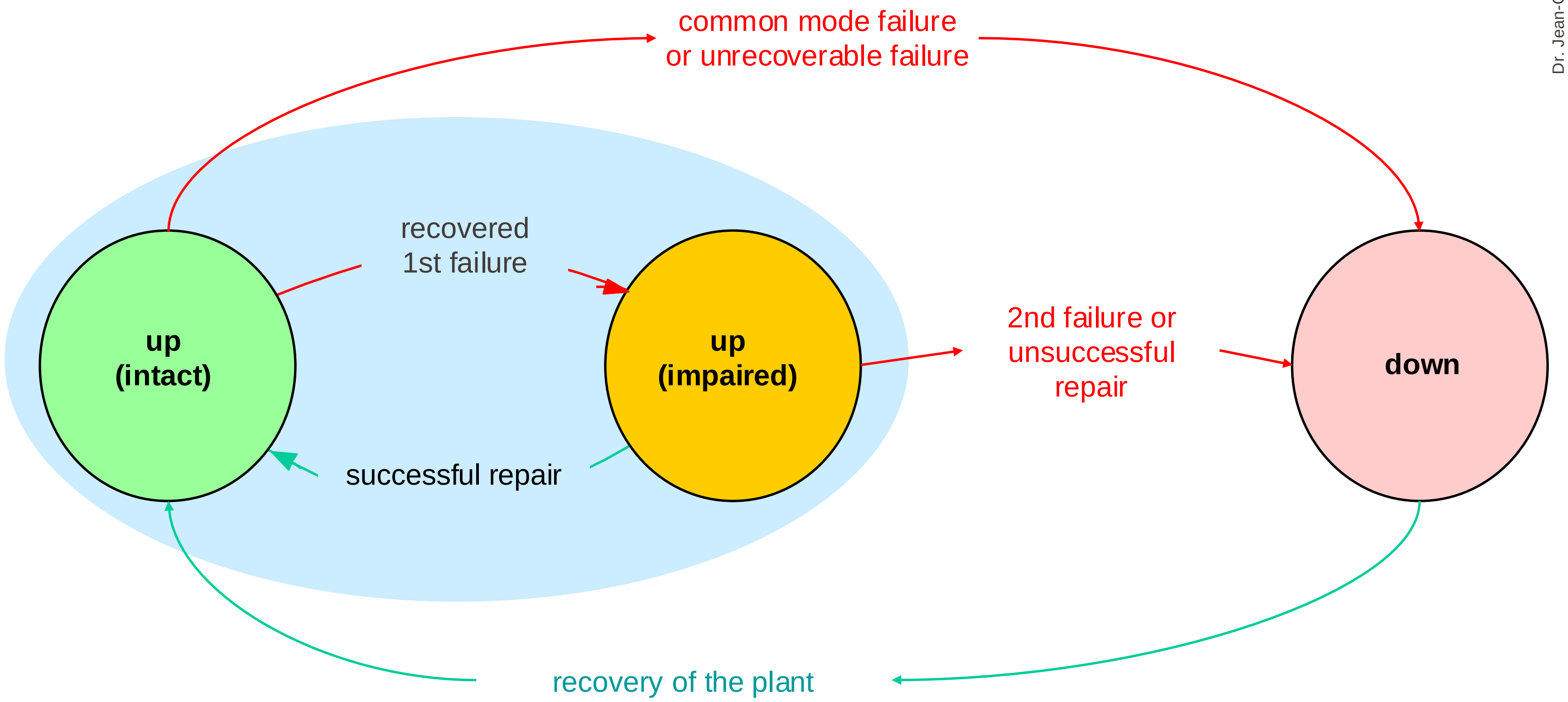
Dr. Jean-Charles Tournier

- When redundancy is available, the system does not fail
  - until redundancy is exhausted
  - or redundancy switchover is unsuccessful
  - or common error to both units

- One is however interested in its reliability
  - how often repair has to be performed
  - how long can it run without fatal failure
  - what is its availability (ratio of up to up+down state duration)

Availability and Repair in Redundant Systems

Dr. Jean-Charles Tournier



Maintenance

- *“The combination of all technical and administrative actions, including supervision actions intended to retain a component in, or restore it to, a state in which it can perform its required function.”*

- Maintenance implies restoring the system to a fault-free state
- i.e. not only correct parts that have obviously failed, but
  - restoring redundancy and degraded parts
  - test for and correct lurking faults

Maintenance Types

- **corrective maintenance**: repair when a part actually fails
  - "go to the garage when the motor fails"

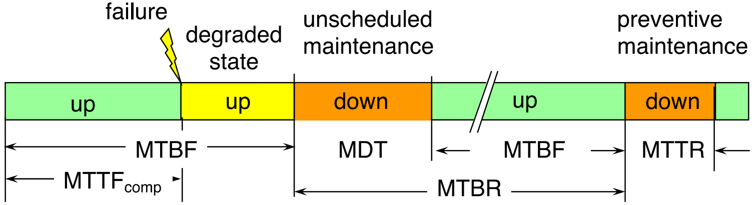
- **preventive maintenance**: restoring to fault-free state
  - "go to the garage to change oil and pump up the reserve tyre”
  - **scheduled maintenance** (time-based maintenance)
    - "go to the garage every year”
  - **predictive maintenance** (condition-based maintenance)
    - "go to the garage at the next opportunity since motor heats up"

- preventive maintenance does not necessarily stop production if redundancy is available
- "differed maintenance" is performed in a non-productive time.

- Differed maintenance is only interesting for plants that are not fully operational 24/24.

Repair and Maintenance

- Redundancy does not replace maintenance, it allows to differ maintenance to a convenient moment (e.g. between 02:00 and 04:00 in the morning).
- Redundancy increases the maintenance effort.
- The system may remain on-line or be taken shortly out of operation for repair.
- The mean time between repairs (MTBR) is the average time between human interventions
- The mean time between failure (MTBF) is the average time between failures.
- The mean time to repair (MTTR) is the average time to bring the impaired system back into operation (introducing off-line redundancy, e.g. spare parts, by human intervention).



Dr. Jean-Charles Tournier

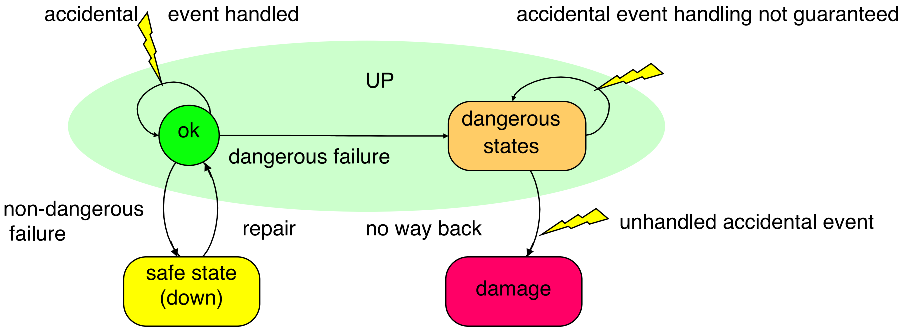
Fault Tolerance

Dr. Jean-Charles Tournier

- *“Ability of a functional unit to continue to perform a required function in the presence of faults or errors”* [IEV 191-15-05]
- Systems able to achieve a given behavior in case of failure without human intervention are fault-tolerant systems.
- The required behavior depends on the application: e.g. stop into a safe state, continue operation with reduced or full functionality.
- **Fault-tolerance requires redundancy**
  - i.e. additional elements that would not be needed if no failure would be expected.
- Redundancy can address physical or design faults.
- Most work in fault-tolerant system addresses the physical faults, because it is easy to provide physical redundancy for the hardware elements.
- Redundancy of the design means that several designs are available.

Safety

- The probability that the system does not behave in a way considered as dangerous.
- Expressed by the probability that the system does not enter a state defined as dangerous
- Difficulty of defining which states are dangerous - level of damage ? acceptable risk ?



Safe States

- Safe state
  - exists: sensitive system
  - does not exist: critical system

- Sensitive systems
  - railway: train stops, all signals red (but: fire in tunnel – is it safe to stop ?)
  - nuclear power station: switch off chain reaction by removing moderator (may depend on how reactor is constructed)

- Critical systems
  - military drones: only possible to fly with computer control system (plane inherently instable)
  - Submarines
  - Space shuttles
  - Aviation

Availability

- Availability is an economical objective.
- High availability increases productive time and yield.
- e.g. airplanes stay aloft
- The gain can be measured in additional productivity
- Availability relies on **operational redundancy** (which can take over the function) and on the quality of maintenance

Safety

- Safety is a regulatory objective
- High safety reduces the risk to the process and its environment
- e.g. airplanes stay on ground
- The gain can be measured in lower insurance rates
- Safety relies on the introduction of
  - **check redundancy** (fail-stop systems)
  - and/or **operational redundancy** (fail-operate systems)

Safety and Availability are often contradictory (completely safe systems are unavailable) since they share a common resource: redundancy.

Trade-off
Safety vs Availability

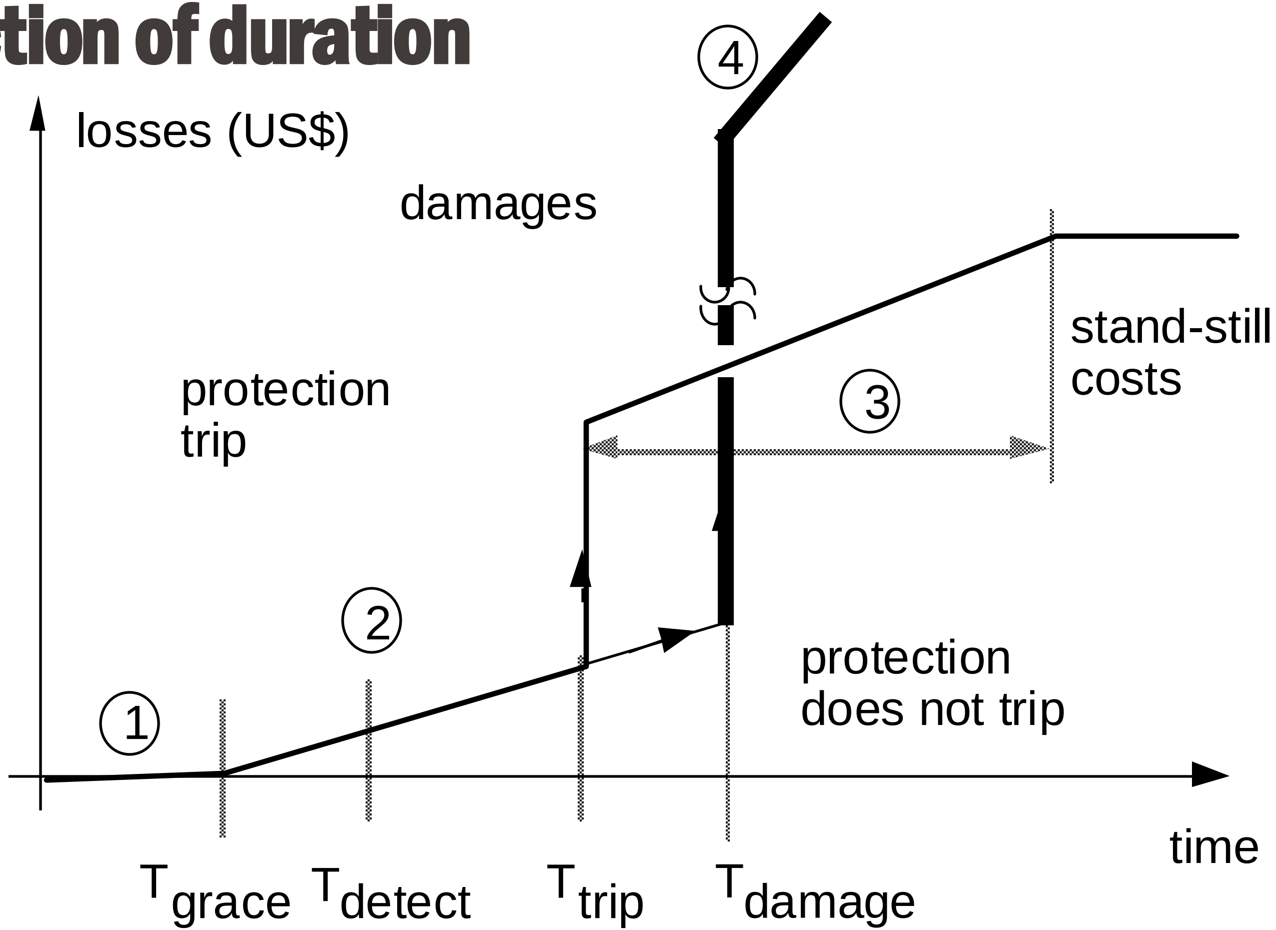
Dr. Jean-Charles Tournier

detected
fault
(don´t know
about failure)

switch to red:
decreased traffic performance
no accident risk (safe)

switch to green:
accident risk
traffic continues (available)

Cost of failure in function of duration



Dr. Jean-Charles Tournier

Safety & Security

- **Safety (*Sécurité/sûreté*, Sicherheit, seguridad):**
  - Avoid dangerous situations due to unintentional failures
    - failures due to random/physical faults
    - failures due to systematic/design faults
      - e.g. railway accident due to burnt out red signal lamp
      - e.g. rocket explosion due to untested software (→ Ariane 5)
- **Security (*Sécurité informatique*, IT-Sicherheit, securidad informática):**
  - Avoid dangerous situations due to malicious threats
    - authenticity / integrity (intégrité): protection against tampering and forging
    - privacy / secrecy (confidentialité, Vertraulichkeit): protection against eavesdropping
      - e.g. robbing of money tellers by using weakness in software
      - e.g. competitors reading production data
- The boundary is fuzzy since some unintentional faults can behave maliciously.
- *Sûreté: terme général: aussi probabilité de bon fonctionnement,* Verlässlichkeit

Dependability Approaches

- **Fault avoidance: eliminate problem sources**
  - Remove defects: Testing and debugging, Verification and Validation
  - Robust design: reduce probability of defects
  - Minimize environmental stress: Radiation shielding, etc.
  - Impossible to avoid faults completely

- **Fault tolerance: add redundancy to mask effect**
  - Additional resources needed (more after)
  - Examples:
    - Error correction coding
    - Backup storage
    - Spare tire, etc.

EPFL

How to increase dependability?

- Fault tolerance: Overcome faults without human intervention.
- Requires **redundancy**: Resources normally not needed to perform the required function.
  - Check Redundancy (that can detect incorrect work)
  - Operational Redundancy (that can do the work)
- Contradiction: Fault-tolerance increases complexity and failure rate of the system.
- Fault-tolerance is no panacea: Improvements in dependability are in the range of 10..100.
- Fault-tolerance is costly:
  - x 3 for a safe system,
  - x 4 times for an available 1oo2 system (1-out-of-2),
  - x 6 times for a 2oo3 (2-out-of-3) voting system
- Redundancy can be defeated by common modes of failure, that affect several redundant elements at the same time (e.g. extreme temperature)
- **Fault-tolerance is no substitute for quality**

Dependability

(*Sûreté de fonctionnement, Verlässlichkeit*)

goals

- reliability
- availability
- maintainability
- safety
- security

achieved by

- fault avoidance
- fault detection/diagnosis
- fault tolerance (= error avoidance)

guaranteed by

- quantitative analysis
- qualitative analysis

by error passivation

- fault isolation
- reconfiguration (on-line repair)

by error recovery

- forward recovery
- backward recovery

by error compensation

- fault masking
- error correction

Summary

- Key concepts introduced in this chapter:
  - Fault – Error – Failure
  - Reliability vs. Availability
  - Availability vs. Safety
  - Safety vs. Security

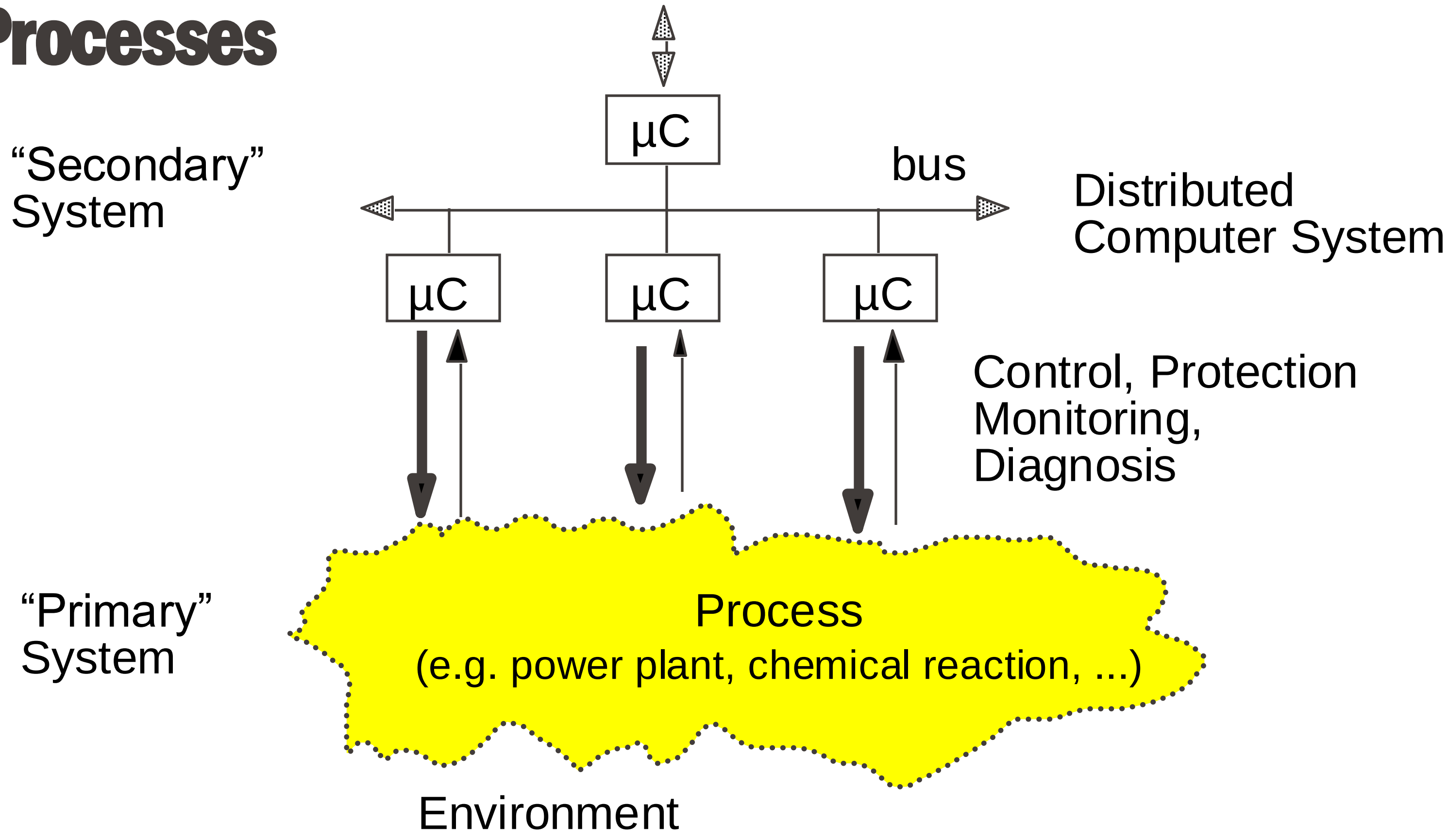
Failure Modes in Computers

Dr. Jean-Charles Tournier

Caveat

- Safety or availability can only be evaluated considering the total system including
  - controller (secondary equipment)
  - plant (primary equipment)
- We consider here only a control system

Computer and Processes



Availability/safety depends on **outputs** of computer system and process/environment.

Types of Computer Failures

Computers can fail in a number of ways

Breach of the specifications = does not behave as intended

reduced to two cases

output of wrong data

or of correct data,but at undue time

↓

integrity breach

missing output of correct data

↓

persistency breach

Fault-tolerant computers allow to overcome these situations.

The architecture of the fault-tolerant computer depends on the encompassed dependability goals

Safety Threats

depending on the controlled process,
safety can be threatened by failures of the control system:

integrity breach

not recognized, wrong data, or correct data, but at the wrong time

if the process is irreversible
(e.g. closing a high power breaker, banking transaction, aircraft takeoff)

Requirement:
fail-silent (fail-safe, fail-stop) computer
"rather stop than fail"

persistency breach

no usable data, loss of control

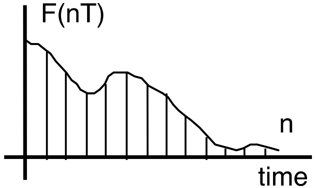
if the process has no safe side
(e.g. landing aircraft)

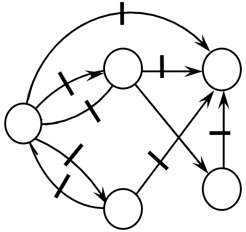
Requirement:
fail-operate computer
"rather some wrong data than none"

Safety depends on the tolerance of the process against failure of the control system

Plant Type and Dependability

continuous systems	discrete systems
modelled by differential equations, and in the linear case, by Laplace or z-transform (sampled)	modelled by state machines, Petri nets, Grafcet,....





continuous systems	discrete systems
continuous systems are generally reversible.	transitions between states are normally irreversible.
tolerates sporadic, wrong inputs during a limited time (similar: noise)	do not tolerate wrong input. difficult recovery procedure
tolerate loss of control only during a short time.	tolerate loss of control during a relatively long time (remaining in the same state is in general safe).
require persistent control	require integer control

Redundancy

- Increasing safety or availability requires the introduction of redundancy
  - resources which are not needed if there were no failures

- Faults are detected by introducing a **check redundancy**
- Operation is continued thanks to **operational redundancy** (can do the same task)

- Increasing reliability and maintenance quality increases both safety and availability

Redundancy Types

- **Massive redundancy (hardware):**
  Extend system with redundant components to achieve the required functionality
  e.g. over-designed wire gauge, use 2-out-of-3 computers

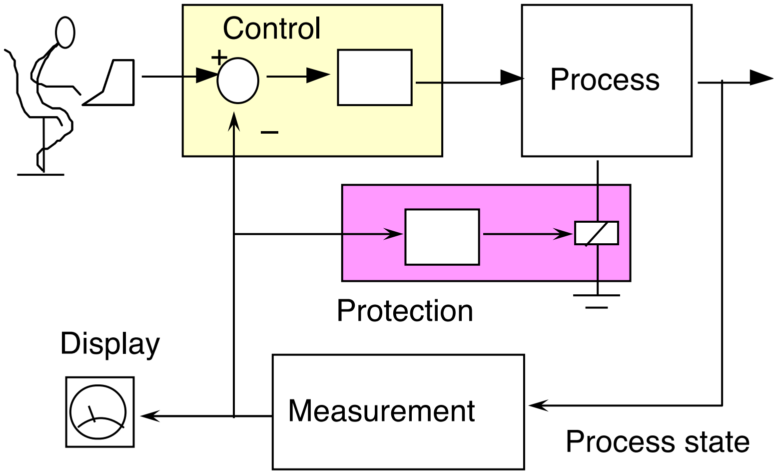
- **Functional redundancy (software):**
  Extend the system with “unnecessary” functions
  back-up functions (e.g. emergency steering)
  diversity (additional different implementation of the required functions)

- **Information redundancy:**
  Encode data with more bits than necessary
  e.g. parity bit, CRC, for error detection, Hamming code, Vitterbi code for error correction

- **Time redundancy:**
  Use additional time, e.g. to do checks or to repeat computation or re-send information

Protection and Control Systems

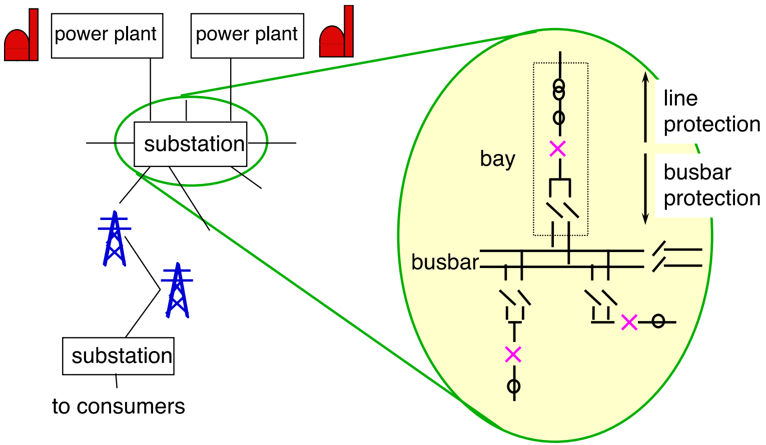
- Control system:
  - Continuous non-stop operation (open or closed loop control)
  - **Maximal failure rate given in failures per year**
- Protection system:
  - Not acting normally,
  - forces safe state (trip) if necessary
  - **Maximal failure rate given in failures per demand**



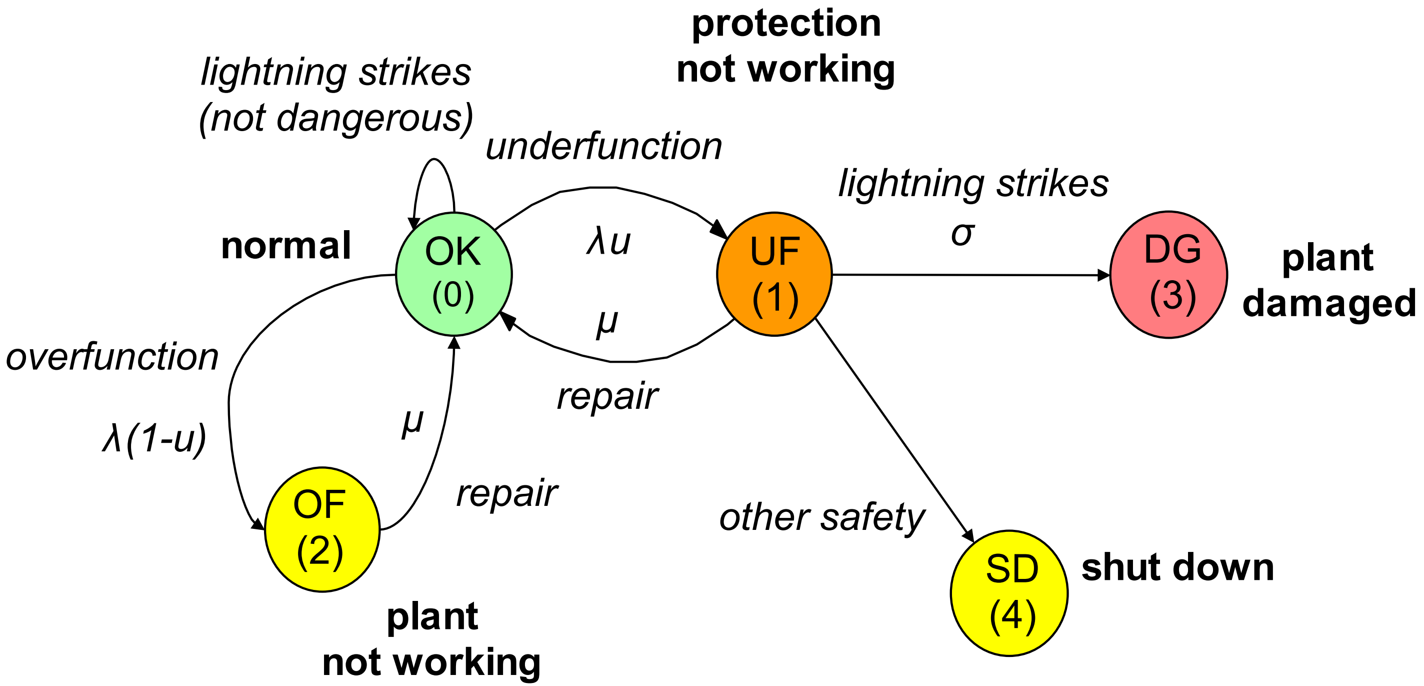
High Voltage Transmission Example

Dr. Jean-Charles Tournier

- Two kinds of malfunctions:
  - An **underfunction**
    - not working when it should of a protection system
    - It is a safety concern
  - An **overfunction**
    - working when it should not of a protection system
    - It is an availability concern



Protection Device States



safe grace time = time during which the plant is allowed to operate without protection
but for this we need to know that the protection is not working !

Summary

- Reliability and fault tolerance must be considered early in the development process,
- They can hardly be increased afterwards.

- Reliability is closely related to the concept of quality
  - its root are laid in the design process
  - It starts with the requirement specs, and accompanying through all its lifetime

EPFL

Some References

- Still an active research area
  - c.f. Software Fault Tolerance in Real-Time Systems: Identifying the Future Research Questions, ACM Computing Surveys, July 2023
- H. Nussbaumer: Informatique industrielle IV; PPUR.
- J.-C. Laprie (ed.): Dependable computing and fault tolerant systems; Springer.
- J.-C. Laprie (ed.): Guide de la sûreté de fonctionnement; Cépaduès.
- D. Siewiorek, R. Swarz: The theory and practice of reliable system design; Digital Press.
- T. Anderson, P. Lee: Fault tolerance - Principles and practice; Prentice-Hall.
- A. Birolini: Quality and reliability of technical systems; Springer.
- M. Lyu (ed.): Software fault tolerance: Wiley.

- Journals: IEEE Transactions on Reliability, IEEE Transactions on Computers
- Conferences: International Conference on Dependable Systems and Networks, European Dependable Computing Conference