# EDA122/DIT061 Fault-Tolerant Computer Systems
# DAT270 Dependable Computer Systems

## Welcome to Lecture 5

Availability modeling
Safety modeling






## Outline

**Availability (Swedish: tillgänglighet)**

- Definition
- Steady-state availability
- Simplex system
- Birth-death processes
- Hot stand-by system with one spare

**Safety (Swedish: säkerhet mot olyckor)**

- Simplex system with coverage factor

**Reliability modeling av large systems**

- Primary subsystems
- Fault and error containment regions

# Definition

**Availability:** the probability that a system is working at a given time *t*.

When calculating the availability we consider both failures and repairs. We must make assumptions about the ***up time*** *(function time)* and the ***down time*** *(total repair time)*.

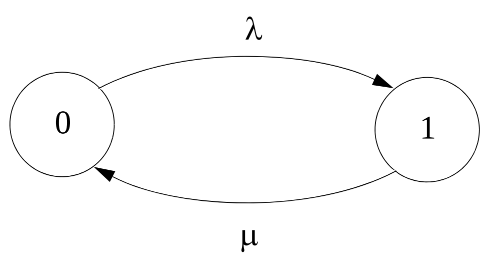
The ***down time*** consists of the time period from a system failure until the system is up and running again, including the time from the occurrence of the failure until repair is started, the time it takes to perform the repair, and the time it takes to restart the system after the repair is completed.

The availability can be defined for different ***service levels*** if the system allows ***graceful degradation***. The notion of a ***working system*** may therefore have different meanings depending the service level considered.





# Markov chain model for a simplex system




State
- 0: System OK
- 1: System failure

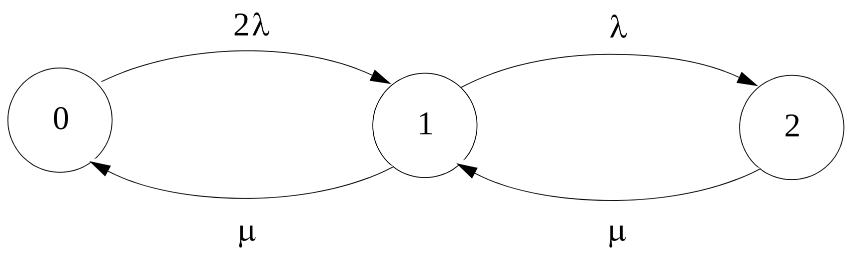
Failure rate: $\lambda$
Repair rate: $\mu$

Availability (Swe: Tillgänglighet): $A(t) = P_0(t)$

Reliability (Swe: Funktionsannolikhet): $R(t) = e^{-\lambda t}$

Maintainability (Swe: Underhållsgodhet): $M(t) = 1 - e^{-\mu t}$

# Markov chain for a hot stand-by system




Availability: $A(t) = P_0(t) + P_1(t)$

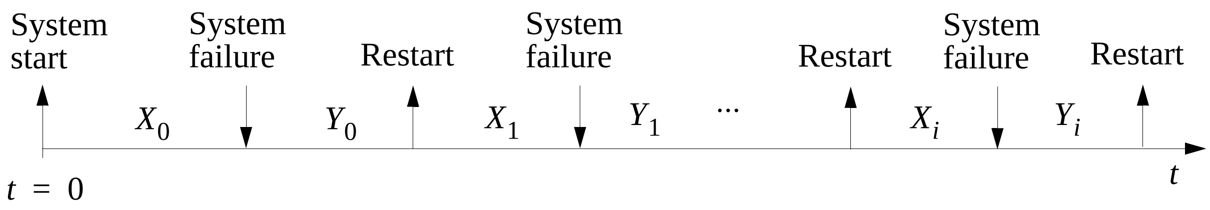
We assume that one repair-person works with the system whenever at least one module is faulty.

The fact that there is only have one repair-person implies that the repair rate is the same in State 1 and State 2.





# Steady-state availability




$E[X_0] = MTTFF$ (mean time to first failure)

$E[X_i] = MTTF$ (mean time to failure)

$E[Y_i] = MTTR$ (mean time to repair)

$MTTF + MTTR = MTBF$ (mean time between failures)

# Steady-state availability (cont’d).

$$\lim_{t \to \infty} A(t) = \frac{MTTF}{MTTR + MTTF} = \frac{MTTF}{MTBF}$$

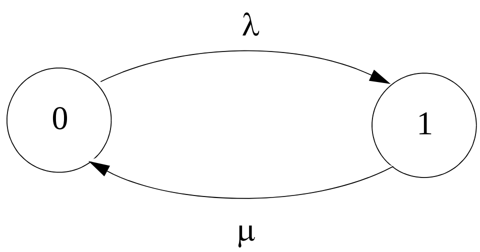
Assuming exponentially distributed function times and repair times, we get

$$A(\infty) = \frac{\frac{1}{\lambda}}{\frac{1}{u} + \frac{1}{\lambda}} = \frac{\mu}{\lambda + u}$$





# The availability for a simplex system




We obtain the following system of differential equations.

$$\mathbf{P}'(\mathbf{t}) = \mathbf{P}(\mathbf{t}) \cdot \mathbf{Q} \quad , \qquad \mathbf{P}(\mathbf{0}) = \begin{bmatrix} P_0(0) & P_1(0) \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix}$$

## Solution sketch

We have a system of two differential equations

$$\begin{cases} P'_0(t) = -\lambda \cdot P_0(t) + \mu \cdot P_1(t) & (1) \\ P'_1(t) = \lambda \cdot P_0(t) - \mu \cdot P_1(t) & (2) \end{cases}$$

We also know that

$$P_0(t) + P_1(t) = 1 \qquad (3)$$

If we substitute $P_1(t)$ with $1 - P_0(t)$ in (1), we obtain

$$P'_0(t) + (\lambda + \mu) \cdot P_0(t) = \mu \qquad (4)$$





## Solution sketch

The solution to equation (4) is

$$P_0(t) = \frac{\mu}{\lambda + \mu}(1 - e^{-(\lambda + \mu) \cdot t}) + P_0(0) \cdot e^{-(\lambda + \mu) \cdot t} \qquad (5)$$

We can obtain $P_1(t)$ simply by exchanging $\lambda$ and $\mu$ in (5)

$$P_1(t) = \frac{\lambda}{\lambda + \mu}(1 - e^{-(\lambda + \mu) \cdot t}) + P_1(0) \cdot e^{-(\lambda + \mu) \cdot t} \qquad (6)$$

Let $\Pi_0$ and $\Pi_1$ denote the steady-state probabilities

$$\Pi_0 = \lim_{t \to \infty} P_0(t) = \frac{\mu}{\lambda + \mu}$$

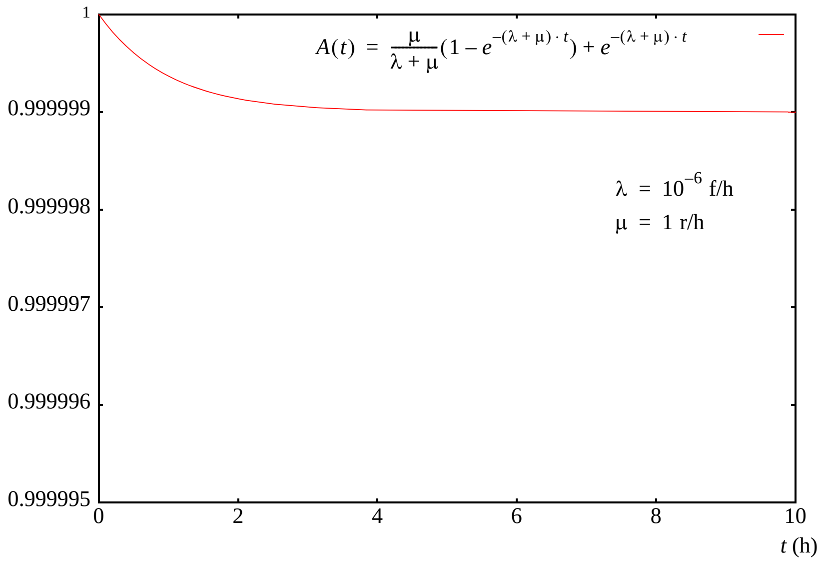
$$\Pi_1 = \lim_{t \to \infty} P_1(t) = \frac{\lambda}{\lambda + \mu}$$

The steady-state availability is $\lim_{t \to \infty} A(t) = \Pi_0$



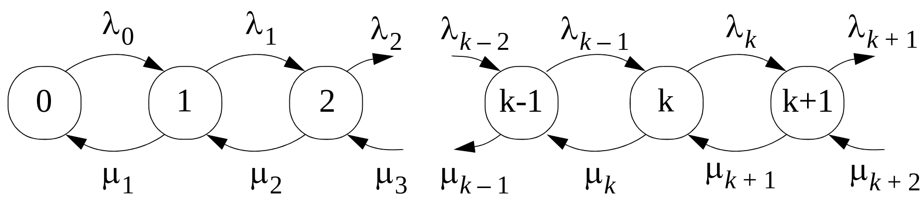


# Availability for a simplex system



# Birth-death processes

A birth-death process can be described by the following state diagram




The transition rate matrix becomes

$$\mathbf{Q} = \begin{bmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & 0 & 0 & \ldots \\ \mu_1 & -(\lambda_1+\mu_1) & \lambda_1 & 0 & 0 & 0 & \ldots \\ 0 & \mu_2 & -(\lambda_2+\mu_2) & \lambda_2 & 0 & 0 & \ldots \\ 0 & 0 & \mu_3 & -(\lambda_3+\mu_3) & \lambda_3 & 0 & \ldots \\ 0 & 0 & 0 & \mu_4 & -(\lambda_4+\mu_4) & \lambda_4 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & 0 & \ldots \end{bmatrix}$$





# Birth-death processes (cont'd)

We obtain the following system of differential equations

$$\mathbf{P}'(\mathbf{t}) = \mathbf{P}(\mathbf{t}) \cdot \mathbf{Q}$$

We calculate the steady-state probability distribution over the states of the process by making the following assumption

$$\Pi_k = \lim_{t \to \infty} P_k(t),\ k = 0, 1, 2, \ldots$$

We assume that the derivatives of the state probabilities tend to zero as time tends to infinity

$$\lim_{t \to \infty} P'_k(t) = 0$$

The differential equations can be written as

$$\begin{cases} P'_0(t) = -\lambda_0 \cdot P_0(t) + \mu_1 \cdot P_1(t) \\ P'_k(t) = \lambda_{k-1} \cdot P_{k-1}(t) - (\lambda_k + \mu_k) \cdot P_k(t) + \mu_{k+1} \cdot P_{k+1}(t), \quad k = 1, 2, \ldots \end{cases}$$

Let $\Pi_i = \lim_{t \to \infty} P_i(t)$ and assume $\lim_{t \to \infty} P'_i(t) = 0$.

We then obtain the following algebraic equations for the steady state probabilities

$$\begin{cases} 0 = -\lambda_0 \cdot \Pi_0 + \mu_1 \cdot \Pi_1 \\ 0 = \lambda_{k-1} \cdot \Pi_{k-1} - (\lambda_k + \mu_k) \cdot \Pi_k + \mu_{k+1} \cdot \Pi_{k+1} \end{cases}$$

which can be rewritten as

$$\begin{cases} -\lambda_0 \cdot \Pi_0 + \mu_1 \cdot \Pi_1 = 0 \\ (-\lambda_k \cdot \Pi_k + \mu_{k+1} \cdot \Pi_{k+1}) - (-\lambda_{k-1} \cdot \Pi_{k-1} + \mu_k \cdot \Pi_k) = 0 \end{cases}$$



If we make the following substitution

$$z_k = -\lambda_k \cdot \Pi_k + \mu_{k+1} \cdot \Pi_{k+1}$$

then the equation system can be written as

$$\begin{cases} z_0 = 0 \\ z_k - z_{k-1} = 0 \end{cases}$$

which has the solution $z_k = 0$ for $k = 0, 1, 2, \ldots$ , which gives

$$\Pi_1 = \frac{\lambda_0}{\mu_1} \cdot \Pi_0 \qquad (1)$$

$$\Pi_{k+1} = \frac{\lambda_k}{\mu_{k+1}} \cdot \Pi_k \qquad (2)$$

$$\Pi_1 = \frac{\lambda_0}{\mu_1} \cdot \Pi_0 \qquad (1)$$

$$\Pi_{k+1} = \frac{\lambda_k}{\mu_{k+1}} \cdot \Pi_k \qquad (2)$$

By repeated use of (2) we obtain

$$\Pi_1 = \frac{\lambda_0}{\mu_1} \cdot \Pi_0$$

$$\Pi_2 = \frac{\lambda_1 \cdot \lambda_0}{\mu_2 \cdot \mu_1} \cdot \Pi_0$$

$$\Pi_3 = \frac{\lambda_2 \cdot \lambda_1 \cdot \lambda_0}{\mu_3 \cdot \mu_2 \cdot \mu_1} \cdot \Pi_0$$

...

$$\Pi_k = \frac{\lambda_{k-1} \cdot \lambda_{k-2} \cdot \ldots \cdot \lambda_0}{\mu_k \cdot \mu_{k-1} \cdot \ldots \cdot \mu_1} \cdot \Pi_0$$





We also know that

$$\sum_{i=0}^{k} \Pi_i = 1$$

We can now determine $\Pi_0$ as

$$\Pi_0 + \Pi_0 \cdot \sum_{i=1}^{k} \frac{\lambda_{i-1} \cdot \lambda_{i-2} \cdot \ldots \cdot \lambda_0}{\mu_i \cdot \mu_{i-1} \cdot \ldots \cdot \mu_1} = 1$$

## Example

Calculate the steady-state availability for a hot stand-by system with one spare module. Assume that the system is repaired by one person.

**We solve the problem on the black-board.**





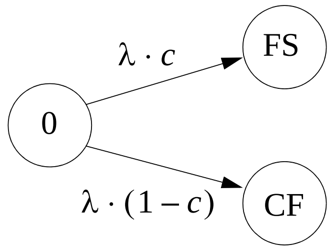

## Safety Modelling

**Safety:** The probability that a system is working correctly or has failed in a safe way.

- Calculating safety is similar to calculating reliability.
- In a reliability model there is usually only ***one absorbing state***, while in a safety model there are at least ***two absorbing states***.
- Among the absorbing states in a safety model, at least one represents that the system is in a ***safe shut-down state***, and at least one represents that system is in a ***catastrophic (or critical) failure state***.

## Safety for a simplex system with coverage factor

We obtain the following Markov chain model




State
- 0: system OK
- FS: fail safe
- CF: critical failure

and the corresponding transition-rate matrix

$$\mathbf{Q} = \begin{bmatrix} -\lambda & \lambda \cdot c & \lambda \cdot (1-c) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$





## Safety for a simplex system with coverage factor (cont'd.)

The solutions of the differential equations are:

$$P_0(t) = e^{-\lambda t}$$

$$P_{FS}(t) = c - ce^{-\lambda t}$$

$$P_{CF}(t) = (1-c) - (1-c)e^{-\lambda t}$$

The safety of the system is

$$S(t) = P_0(t) + P_{FS}(t) = e^{-\lambda t} + c - ce^{-\lambda t} = c + (1-c)e^{-\lambda t}$$

The steady-state safety is:

$$\lim_{t \to \infty} S(t) = c$$

# Reliability modelling of large systems

- Example: Hot stand-by control system
- Divide and conquer
- Primary independent subsystems
- Methodology for reliability modelling
- Fault/error containment regions





# Hot Stand-by Control System

## Physical Architecture

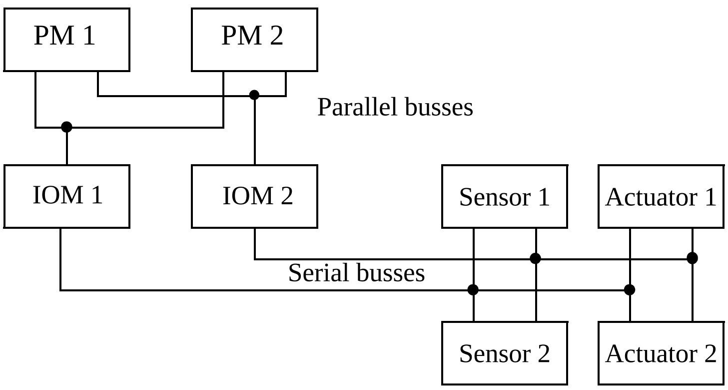



Figure 1

# Hot Stand-by Control System

## Textual description

Figure 1 shows the physical architecture for a fault-tolerant computer system for a real-time control application.

The system consists of two processor modules (PM 1 and PM 2), two I/O-modules (IOM 1 and IOM 2), two parallel and two serial buses, and two sensors and two actuators.

All ***primary subsystems*** operate as hot stand-by systems.

The processor modules execute the control program which calculates the outputs for the actuators based on sensors values.

The I/O-module handles the data communication between the processor modules and the sensors/actuators. The I/O-module is the bus master for both the parallel bus and the serial bus.





# Reliability Analysis of Large Systems

Reliability and availability analysis using Markov chain models becomes increasingly difficult as the number of modules in a system increases.

If we have $n$ modules in the system, we must (in principle) consider $\boldsymbol{2^n}$ states, since each module can assume one of two states: ***operational (working)*** or ***non-operational (broken)***.

For small systems we can often manually reduce the number of states. For example, we have previously used a model with three states for a TMR system, although $2^3 = 8$ combinations of failed and working units can occur in a TMR system. Each of these combinations corresponds to an ***elementary state*** of the system.

The reason why we can reduce the number of states to three is that there are more *elementary states* than ***significant states***, e.g., there are several elementary states that correspond to a system failure.

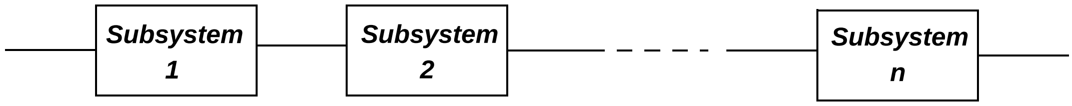
For large systems that consist of many modules of different types it becomes difficult to define markov chain models manually, as the number of ***significant states*** in the model is large.

# Divide and Conquer

One approach for simplifying the analysis of large systems is to divide the system into a number of ***independent subsystems***, which we can call ***primary subsystems.***

We assume that a system consists of several primary subsystems, which ***all must function in order for the system to function***.

Thus, at the highest level of abstraction the system is a series system




Reliability block diagram

**Note**: Not all system can be divided into independent subsystems!






# Independent Primary Subsystems

Definitions:

1 A ***primary subsystem*** is one which is essential to the system, i.e., a failure of a primary subsystems always results in a system failure.

2 If all failures of a primary subsystem are mutually independent of all failures of all other subsystem, then it is an ***independent primary subsystem****.*

# Method for reliability analysis

Starting from a physical architecture and a functional description of the fault-tolerance features, we conduct the analysis in three steps:

1 Determine the ***independent primary subsystems***.

2 Calculate the reliability of each independent primary subsystem.

3 Calculate the reliability of the system as $R_{system} = \prod_{i=1}^{n} R_i$, where $R_i$ is the reliability of subsystem i.



# Primary Subsystems

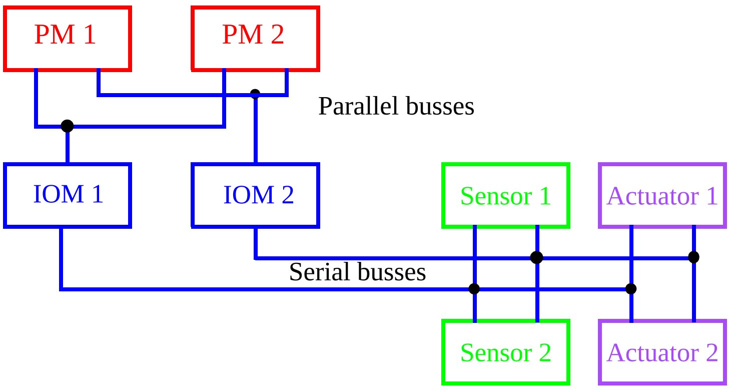



Figure 1

We have four primary subsystems:

- The processor modules (in red)
- The I/O modules including the serial and parallel busses (in blue)
- The sensors (in green)
- The actuators (in purple)

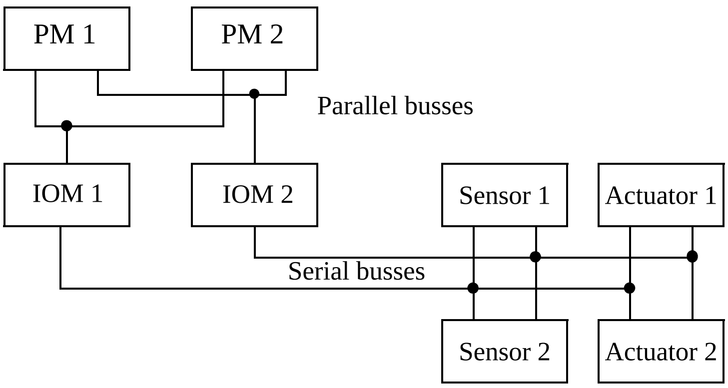
# Fault/Error Containment Regions

- Fault/error containment aims at preventing faults/errors in one unit from affecting other units.
- A fault-tolerant computer system consists of several fault/error containment regions.
- Fault/error containment should be maintained at all unit interfaces where fault and error propagation may lead to a reduction of system reliability.
- Fault/error containment is not needed between units that constitute a series system.





# Fault Containment Regions




There are 8 fault containment regions in this system: ***PM 1***, ***PM 2***, ***IOM 1 (including buses)***, ***IOM 2 (including buses)***, ***Sensor 1***, ***Sensor 2***, ***Actuator 1*** and ***Actuator 2***.

# Modified Hot Stand-by Control System

## Physical Architecture

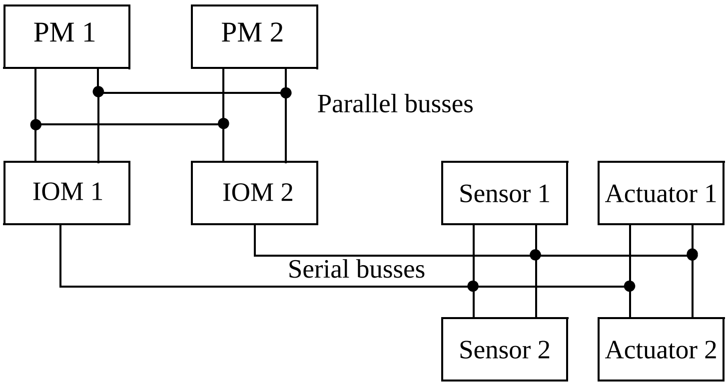



Figure 1

We introduce a cross-coupling of the parallel buses.

There are now ***10*** fault containment regions: ***PM 1, PM 2***, ***Parallel Bus 1, Parallell Bus2***, ***IOM1 and Serial bus 2, IOM2 and Serial bus 2***, ***Sensor 1, Sensor 2***, ***Actuator 1*** and ***Actuator 2***.





# Overview of Lecture 6

- Case Study: Hewlett-Packard’s Non-Stop Advanced Architecture
  Preparations:
  Course book: Section 6.1, 6.3, 6.4, 6.5, 6.8
  Paper by Bernick et al., “Non-Stop Advanced Architecture”

- Case Study: Ariane 501 disaster
- Introduction to redundancy in software
  Preparations:
  Course book: Section 6.1, 6.2 (software faults), 6.3, 6.6
  Report on Ariane 501 failure (**Very important to read before the lecture!**)