دانشکده مهندسی کامپیوتر

عنوان درس:

طراحی نرم‌افزارهای اتکاءپذیر

(Dependable Software Design)

فصل ۳: فنون ارزیابی اتکاءپذیری

مدرس: محمد عبداللهی ازگمی

(Mohammad Abdollahi Azgomi)

azgomi@iust.ac.ir

Dependability Evaluation Techniques

- **Reference:**
  - **E. Dubrova,** ***Fault-Tolerant Design: An Introduction,*** **Kluwer Academic Publisher (2007)**
    - Chapter 3: Dependability Evaluation Techniques
- **--**
- *A common mistake that people make when trying to design something completely foolproof is to underestimate the ingenuity of complete fools.*

  — **Douglas Adams, Mostly Harmless**

Contents

1. Introduction

- در کنار **هزینه** (cost) و **کارایی** (performance)، سومین معیار مهم تصمیم‌گیری در مورد سیستم‌ها **اتکاءپذیری** است.
- ارزیابی **اتکاءپذیری** از این نظر اهمیت دارد که کمک می‌کند که مشخص شود که کدام جنبه از رفتار سیستم، نظیر **قابلیت اطمینان مولفه** (component reliability)، **پوشش خطا** (fault coverage) یا **استراتژی نگهداشت** (maintenance strategy) نقش بحرانی را در تعیین اتکاءپذیری کلی سیستم بازی می‌کنند.
- از اینرو ارزیابی اتکاءپذیری تمرکز درستی را برای تلاش بهبود محصول از همان مراحل ابتدایی توسعه محصول تا ساخت و آزمون فراهم می‌کند.

1. Introduction

- دو رهیافت مرسوم برای ارزیابی اتکاءپذیری عبارتند از:
  1. مدل‌سازی سیستم در مرحله طراحی، یا
  2. ارزیابی سیستم در مراحل بعدی، نوعاً بوسیله آزمون.
- نخستین رهیافت مبتنی بر مدلهای احتمالی است که از **نرخهای خرابی** (failure rate)که در کتابچه‌ها (handbooks) منتشر می‌شود یا بوسیله سازنده‌ها فراهم می‌شود استفاده می‌کند.
  - این رهیافت امکان تعیین زودهنگام اتکاءپذیری سیستم را فراهم می‌کند. اما هم مدل و داده‌های مورد استفاده در آن لازم است که با اندازه‌گیری‌های واقعی اعتبارسنجی شوند.
- رهیافت دوم نوعاً از داده‌های آزمون و **مدلهای رشد قابلیت اطمینان** (reliability growth models) بهره می‌برند.
  - این رهیافت شامل مفروضات کمتری نسبت به رهیافت قبلی است، اما می‌تواند خیلی گران باشد. هر قدر که اتکاءپذیری مورد نیاز برای یک سیستم بیشتر باشد، آزمون مورد نیاز بیشتر خواهد بود.
  - مشکل دیگر در ترجمه (و تبدیل) داده‌های قابلیت اطمینان بدست آمده بوسیله آزمون به داده‌هایی است که قابل به‌کارگیری در محیط عملیاتی هستند.

1. Introduction

- **Dependability evaluation has two aspects:**
  - The first is ***qualitative evaluation***, that aims to identify, classify and rank the **failure modes**, or the events combinations that would lead to system failures.
    - For example, component faults or environmental conditions are analyzed.
  - The second aspect is ***quantitative evaluation***, that aims to evaluate in terms of probabilities the extend to which some attributes of dependability, such as reliability, availability, safety, are satisfied. Those attributes are then viewed as measures of dependability.

1. Introduction

- In this chapter we study common dependability measures, such as **failure rate**, **mean time to failure**, **mean time to repair**, etc.
- Examining the time dependence of failure rate and other measures allows us to gain additional insight (بینش) into the nature of failures.
- Next, we examine possibilities for modeling of system behaviors using reliability block diagrams and Markov processes.
- Finally, we show how to use these models to evaluate system's reliability, availability and safety.
- We begin with a brief introduction into the probability theory, necessary to understand the presented material.

2. Basics of probability theory

- The value of probability of an event *A* lies between 0 and 1:

$$0 \leq p(A) \leq 1. \qquad (3.1)$$

- Let *A* denotes the event "not *A*". Then:

$$p(\overline{A}) = 1 - p(A). \qquad (3.2)$$

- Suppose that one event, *A* is dependent on another event, *B*. Then *P*(*A*|*B*) denotes the conditional probability of event *A*, given event *B*. the probability *p*(*AB*) that both *A* and *B* will occur

$$p(A \cdot B) = p(A|B) \cdot p(B), \text{ if } A \text{ depends on } B. \qquad (3.3)$$

$$p(A|B) = \frac{p(A \cdot B)}{p(B)} \qquad (3.4)$$

2. Basics of probability theory

- For independent events:

$$p(A \cdot B) = p(A) \cdot p(B), \text{ if } A \text{ and } B \text{ are independent events.} \quad (3.5)$$

- If *A* occurs, *B* cannot, and vice versa, i.e. A and B are mutually exclusive:

$$p(A \cdot B) = 0, \text{ if } A \text{ and } B \text{ are mutually exclusive events.} \quad (3.6)$$

- The probability *p*(*A*+*B*) is given by:

$$p(A + B) = p(A) + p(B) - p(A \cdot B) \quad (3.7)$$

- Combining (3.6) and (3.7), we get:

$$p(A + B) = p(A) + p(B), \text{ if } A \text{ and } B \text{ are mutually exclusive events.} \quad (3.8)$$

3. Common Measures of Dependability

- In this section, we describe common dependability measures:
  - **Failure rate** (نرخ خرابی)
  - **Mean time to failure (MTTF)** (میانگین زمان تا خرابی)
  - **Mean time to repair (MTTR)** (میانگین زمان تا تعمیر)
  - **Mean time between failures (MTBF)** (میانگین زمان بین خرابیها)
  - **Fault coverage** (پوشش خطا)

3.1 Failure Rate

- ***Failure rate*** **λ is the expected number of failures per unit time.**
  - □ نرخ خرابی، امید ریاضی (میانگین) تعداد خرابیها در واحد زمان است.
  - □ For example, if a processor fails, on average, once every 1000 hours, then it has a failure rate λ = 1/1000 failures/hour.

3.1 Failure Rate

- Often failure rate data is available at component level, but not for the entire system.
- This is because several professional organizations collect and publish failure rate estimates for frequently used components (diodes, switches, gates, flip-flops, etc.).
- At the same time the design of a new system may involve new configurations of such standard components.
- When component failure rates are available, a crude estimation of the failure rate of a non-redundant system can be done by adding the failure rates λ_i of the components:

$$\lambda = \sum_{i=1}^{n} \lambda_i$$

3.1 Failure Rate

- **Failure rate changes as a function of time.**
- For hardware, a typical evolution of failure rate over a system's life-time is characterized by the phases of:
  - □ **infant mortality (I) (مرگ زودرس),**
  - □ **useful life (II) (عمر مفید) and**
  - □ **wear-out (III) (فرسودگی).**
- These phases are illustrated by ***bathtub curve*** **(منحنی وان حمام)** relationship shown in Figure 3.1.

3.1 Failure Rate

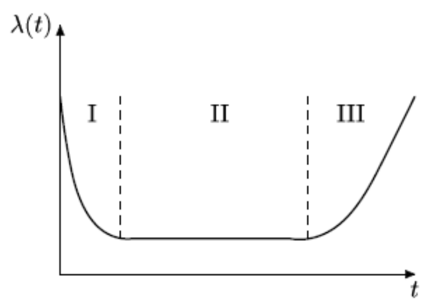


Figure 3.1. Typical evolution of failure rate over a life-time of a hardware system.

3.1 Failure Rate

- Failure rate at first decreases due to frequent failures in weak components with manufacturing defects overlooked during manufacturer's testing (poor soldering, leaking capacitor, etc.),
- then stabilizes after a certain time and
- then increases as electronic or mechanical components of the system physically wear out.

3.1 Failure Rate

- During the useful life phase of the system, failure rate function is assumed to have a constant value λ.
- Then, the reliability of the system varies exponentially as a function of time:

$$R(t) = e^{-\lambda t} \qquad (3.9)$$

  - This is based on observations and reliability measurements and estimations.
- **This law is known as *exponential failure law*.**

3.1 Failure Rate

- The plot of reliability as a function of time is shown in Figure 3.2.

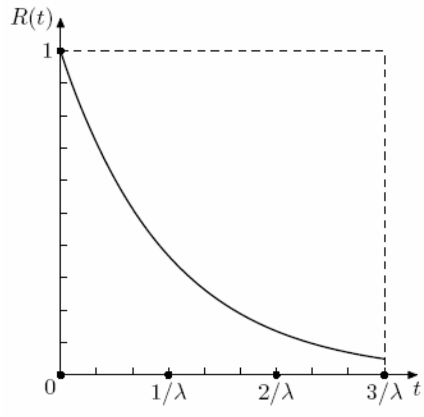


Figure 3.2. Reliability plot $R(t) = e^{-\lambda t}$.

3.1 Failure Rate

- The exponential failure law is very valuable for analysis of reliability of components and systems in hardware.
- However, it can only be used in cases when the assumption that the failure rate is constant is adequate.

3.1 Failure Rate

- **Software failure rate usually decreases as a function of time.** A possible curve is shown in Figure 3.3.
- **There three phases of evolution are:** test/debug (I), useful life (II) and obsolescence (III).

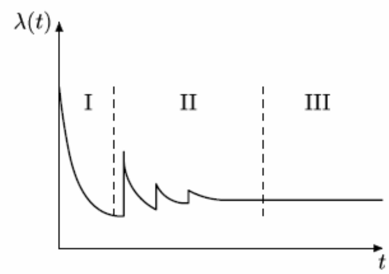


Figure 3.3. Typical evolution of failure rate over a life-time of a software system.

3.1 Failure Rate

- Software failure rate during useful life depends on the following factors:
  1. software process used to develop the design and code
  2. complexity of software,
  3. size of software,
  4. experience of the development team,
  5. percentage of code reused from a previous stable project,
  6. rigor and depth of testing at test/debug (I) phase.

3.1 Failure Rate

- **There are two major differences between hardware and software curves.**
  - One difference is that, in the useful-life phase, software normally experiences an increase in failure rate each time a feature upgrade is made. Since the functionality is enhanced by an upgrade, the complexity of software is likely to be increased, increasing the probability of faults. After the increase in failure rate due to an upgrade, the failure rate levels off gradually, partly because of the bugs found and fixed after the upgrades.
  - The second difference is that, in the last phase, software does not have an increasing failure rate as hardware does. In this phase, the software is approaching obsolescence and there is no motivation for more upgrades or changes.

3.2 Mean Time to Failure (MTTF)

- Another important and frequently used measure of interest is **mean time to failure (MTTF)** defined as follows.
- **The *mean time to failure* (MTTF) of a system is the expected time until the occurrence of the first system failure.**
  - میانگین زمان تا وقوع اولین خرابی
- If n identical systems are placed into operation at time $t = 0$ and the time t_i, $i = \{1, 2, \ldots, n\}$, that each system i operates before failing is measured then the average time is MTTF:

$$MTTF = \frac{1}{n} \cdot \sum_{i=1}^{n} t_i \qquad (3.10)$$

3.2 Mean Time to Failure (MTTF)

- In terms of system reliability $R(t)$, MTTF is defined as

$$MTTF = \int_0^\infty R(t)dt. \qquad (3.11)$$

  - ??? Next page for proof.
  - So, MTTF is the area under the reliability curve in Figure 3.2.
- If the reliability function obeys the exponential failure law (3.9), then the solution of (3.11) is given by $MTTF = 1/\lambda$ (3.12)
  - where λ is the failure rate of the system.
  - **The smaller the failure rate is, the longer is the time to the first failure.**

3.2 Mean Time to Failure (MTTF)

- فرض کنید که X یک متغیر تصادفی باشد که نشان‌دهنده مدت زندگی یک سیستم است.
- آنگاه $R(t)$ احتمال زنده بودن سیستم در زمان t خواهد بود:

$$R(t) = P(X > t) = 1\text{-}F(t)$$

- در حالت اولیه سیستم سالم است: $R(0) = 1$
- همچنین در زمان بی‌نهایت هم سیستم خراب خواهد شد: $R(\infty) = 0$
- آنگاه با مشتق گرفتن از طرفین خواهیم داشت:

$$R'(t) = \frac{d}{dt}(1 - F(t)) = -f(t)$$

- حال برای محاسبه MTTF با توجه به تعریف امید ریاضی $(E(x))$ خواهیم داشت:

$$MTTF = \int_{-\infty}^{+\infty} t\, f(t)\, dt = \int_0^{+\infty} t\,(-R'(t))\, dt = -\int_0^{+\infty} t\, R'(t)\, dt$$

انتگرال جزء به جزء:

$$\int f'(x)\, g(x)\, dx = f(x)\, g(x) - \int f(x)\, g'(x)\, dx$$

$$MTTF = -t\, R(t)\Big|_0^\infty + \int_0^\infty R(t)\, dt = \int_0^\infty R(t)\, dt$$

3.2 Mean Time to Failure (MTTF)

- In general, MTTF is meaningful only for systems that operate without repair until they experience a system failure.
- In a real situation, most of the **mission critical systems** undergo a complete check-out before the next mission is undertaken.
- All failed redundant components are replaced and the system is returned to a fully operational status.
- When evaluating the reliability of such systems, **mission time** rather than MTTF is used.
- زمان ماموریت باید کوچکتر از MTTF باشد.

3.3 Mean Time to Repair (MTTR)

- **The *mean time to repair* (MTTR) of a system is the average time required to repair the system.**
- MTTR is commonly specified in terms for a *repair rate* μ, which is the expected number of repairs per unit time: $MTTR = 1/\mu$. (3.13)

3.3 Mean Time to Repair (MTTR)

- **MTTR depends on:**
  - fault recovery mechanism used in the system,
  - location of the system, (محل سیستم و امکان دسترسی سریع به آن)
  - location of spare modules (on-site versus off-site),
    - The word “cite” is used in textbook instead of “site”!?!?
  - maintenance schedule, etc.

3.3 Mean Time to Repair (MTTR)

- **Low MTTR requirement means high operational cost of the system.**
  - For example, if repair is done by replacing the hardware module, the hardware spares are kept on-site and the site is maintained 24 hours a day, then **the expected MTTR can be 30 min.**
  - However, if the site maintenance is relaxed to regular working hours on week days only, **the expected MTTR increases to 3 days.**
  - If the system is remotely located and the operator need to be flown in to replace the faulty module, **the MTTR can be 2 weeks.**

3.3 Mean Time to Repair (MTTR)

- In software, if the failure is detected by **watchdog timers** (تایمرهای نگهبان) and the processor automatically restart the failed tasks, without operating system reboot, **then MTTR can be 30 sec.**
- If software fault detection is not supported and a manual reboot by an operator is required, **then MTTR can range from 30 min to 2 weeks, depending on location of the system.**
  - بسته به اینکه آیا اپراتور بالای سر سیستم هست یا نه که آنرا reboot کند.

3.3 Mean Time to Repair (MTTR)

- **If the system experiences *n* failures during its lifetime, the total time that the system is operational is: *n MTTF*.**
- **Likewise, the total time the system is being repaired is: *n MTTR*.**
- The **steady state availability** given by the expression (2.2) can be approximated as

$$A(\infty) = \frac{n \cdot MTTF}{n \cdot MTTF + n \cdot MTTR} = \frac{MTTF}{MTTF + MTTR}$$

- In section 5.2.2, we will see an alternative approach for computing availability, which uses Markov processes.

3.4 Mean Time Between Failures (MTBF)

- **The *mean time between failures* (MTBF) of a system is the average time between failures of the system.**
- If we assume that a repair of the system makes the system a perfect one, then the relationship between MTBF and MTTF is as follows:
  - ***MTBF = MTTF +MTTR***

3.5 Fault Coverage

- There are several types of fault coverage (پوشش خطا), depending on whether we are concerned (نگران بودن) (اهمیت داشتن) with *fault detection*, *fault location*, *fault containment* or *fault recovery*.
- **Intuitively, fault coverage is the probability that the system <u>will not fail</u> to perform the expected actions when a fault occurs.**
- More precisely, fault coverage is defined in terms of the conditional probability $P(A|B)$, read as "probability of A given B".
  - Will be discussed later …

3.5 Fault Coverage

- ***Fault detection coverage* is the conditional probability that, given the existence of a fault, the system detects it.**
  - $C = P$**(fault detection|fault existence)**
  - For example, a system requirement can be that 99% of all single stuck-at faults are detected. The fault detection coverage is a measure of system's ability to meet such a requirement.

3.5 Fault Coverage

- ***Fault location coverage* is the conditional probability that, given the existence of a fault, the system locates it.**
  - $C = P$**(fault location|fault existence)**
- It is common to require system to locate faults within easily replaceable modules. In this case, the fault location coverage can be used as a measure of success.

3.5 Fault Coverage

- **Similarly, *fault containment coverage* is the conditional probability that, given the existence of a fault, the system contains** (محدود کند) **it.**
  - □ ***C* = *P*(fault containment|fault existence)**

3.5 Fault Coverage

- **Finally, *fault recovery coverage* is the conditional probability that, given the existence of a fault, the system recovers.**
  - □ ***C* = *P*(fault recovery|fault existence)**

4. Dependability Model Types

- In this section we consider two common dependability models: **Reliability block diagrams (RBD)** (نمودارهای بلوکی قابلیت اطمینان) and **Markov processes.**
  - **Reliability block diagrams** belong to a class of *combinatorial models* (مدلهای ترکیبی), which assume that **the failures of the individual components are mutually independent**.
  - **Markov processes** belong to a class of *stochastic processes* which **take the dependencies between the component failures into account**, making the analysis of more complex scenarios possible.

4.1 Reliability Block Diagrams

- Combinatorial reliability models include:
  - reliability block diagrams,
  - fault trees,
  - success trees and
  - reliability graphs.
- In this section we will consider the oldest and most common reliability model: **reliability block diagrams**.

4.1 Reliability Block Diagrams

- A **reliability block diagram** presents an abstract view of the system.
- The **components** are represented as blocks.
- The **interconnections** among the blocks show the **operational dependency between the components**.
  - **Blocks are connected in series** if all of them are necessary for the system to be operational.
  - **Blocks are connected in parallel** if only one of them is sufficient for the system to operate correctly.

4.1 Reliability Block Diagrams

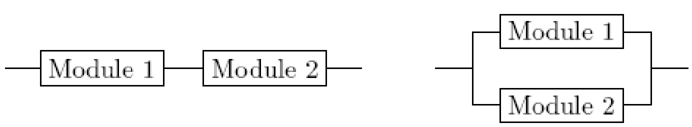


Figure 3.4. Reliability block diagram of a two-component system: (a) serial, (b) parallel.

- A diagram for a two-component serial system is shown in Figure 3.4(a).
- Figure 3.4(b) shows a diagram of a **two-component parallel system.**
- Models of more complex systems may be built by combining the serial and parallel reliability models.

4.1 Reliability Block Diagrams

- As an example, consider a system consisting of two duplicated processors and a memory. The reliability block diagram for this system is shown in Figure 3.5.

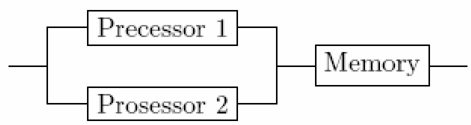


Figure 3.5. Reliability block diagram of a three-component system.

  - The processors are connected in parallel, since only one of them is sufficient for the system to be operational.
  - The memory is connected in series, since its failure would cause the system failure.

4.1 Reliability Block Diagrams

- Reliability block diagrams are a popular model, because they are easy to understand and to use for modeling systems with **redundancy**.
- In the next section we will see that they are also easy to evaluate using analytical methods.
- However, reliability block diagrams, as well as other combinatorial reliability models, have a number of serious limitations…

4.1 Reliability Block Diagrams

- Limitations of RBDs:
  - First, reliability block diagrams assume that the system components are limited to the operational and failed states and that the system configuration does not change during the mission. Hence, they cannot model **standby (یدکی) components**, **repair** as well as **complex fault detection and recovery mechanisms**.
  - Second, the failures of the individual components are assumed to be **independent**. Therefore, the case when the sequence of component failures affects system reliability cannot be adequately represented.

4.2 Markov Processes

- Contrary to combinatorial models, Markov processes take into account the interactions of component failures making the analysis of complex scenarios possible.
- Markov processes theory derives its name from the Russian mathematician A. A. Markov (1856-1922), who pioneered a systematic investigation of describing random processes mathematically.

4.2 Markov Processes

- Markov processes are a special class of stochastic processes.
- The basic assumption is that the behavior of the system in each state is **memoryless**.
- The transition from the current state of the system is determined only by the present state and not by the previous state or the time at which it reached the present state.
- Before a transition occurs, the **time spent in each state follows an exponential distribution**.
- In dependability engineering, this assumption is satisfied if all events (failures, repairs, etc.) in each state occur with **constant occurrence rates**. (و نرخها متغیر نباشند)

4.2 Markov Processes

- Markov processes are classified based on state space and time space characteristics as shown in Table 3.1.
- In most dependability analysis applications, the state space is **discrete**.

State Space	Time Space	Common Model Name
Discrete	Discrete	Discrete Time Markov Chains
Discrete	Continuous	Continuous Time Markov Chains
Continuous	Discrete	Continuous State, Discrete Time Markov Processes
Continuous	Continuous	Continuous State, Continuous Time Markov Processes

Table 3.1. Four types of Markov processes.

4.2 Markov Processes

- For example, a system might have two states: **operational** and **failed**.
  - □ The time scale is usually **continuous**, which means that component failure and repair times are random variables.
  - □ Thus, *Continuous Time Markov Chains* are the most commonly used.
    - In some textbooks, they are called *Continuous Markov Models*.
- There are, however, applications in which **time scale is discrete**.
  - □ Examples include synchronous communication protocol, shifts in equipment operation (?!??!), etc.
- If both time and state space are discrete, then the process is called *Discrete Time Markov Chain*.

4.2 Markov Processes

- Markov processes are illustrated graphically by **state transition diagrams**.
- A *state transition diagram* is a directed graph $G = (V, E)$, where
  - □ V is the set of vertices representing *system states* and
  - □ E is the set of edges representing *system transitions*.

4.2 Markov Processes

- **For dependability models, a state is defined to be a particular combination of operating and failed components.**
  - For example, if we have a system consisting of two components, then there are four different combinations enumerated in Table 3.2, where *O* indicates an operational component and *F* indicates a failed component.

Component 1	Component 2	State Number
O	*O*	1
O	*F*	2
F	*O*	3
F	*F*	4

  - *Table 3.2.* Markov states of a two-component system.

4.2 Markov Processes

- **The state transitions reflect the changes which occur within the system state.**
  - For example, if a system with two identical component is in the state (11), and the first module fails, then the system moves to the state (01). So, a Markov process represents possible chains of events which occur within a system.
- In the case of dependability analysis, these events are **failures** and **repairs**.
- Each edge carries a label, reflecting the rate at which the state transitions occur. Depending on the modeling goals, this can be **failure rate**, **repair rate** or **both**.
- We illustrate the concept first on a simple system, consisting of a single component.

4.2.1 Single-Component System

- A single component has only two states: one operational (state 1) and one failed (state 2).
- If no repair is allowed, there is a single, non-reversible transition between the states, with a label l corresponding to the failure rate of the component (Figure 3.6).

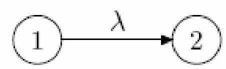


Figure 3.6. State transition diagram of a single-component system.

Single-Component System

- If repair is allowed, then a transition between the failed and the operational states is possible, with a repair rate μ (Figure 3.7). State diagrams incorporating repair are used in availability analysis.

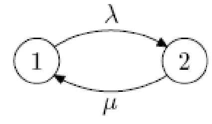


Figure 3.7. State transition diagram of a single-component system incorporating repair.

Single-Component System

- Next, suppose that we would like to distinguish between a **failed-safe** and **failed-unsafe** states, as required in safety analysis.
- Let state 2 be a failed-safe and state 3 be a fail-unsafe states (Figure 3.8).

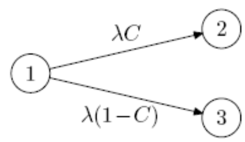


Figure 3.8. State transition diagram of a single-component system for safety analysis.

Single-Component System

- The transition between the state 1 and state 2 depends on both, **component failure rate** λ and the probability that, given the existence of a fault, the system succeeds in detecting it and taking the corresponding actions to fail in a safe manner, i.e. on **fault coverage *C***.
- The transition between the state 1 and the failed-unsafe state 3 depends on failure rate λ and the probability that a fault is *not* detected, i.e. 1-*C*.

Two-Component System

- A two-component system has four possible states, enumerated in Table 3.2.

Component 1	Component 2	State Number
O	O	1
O	F	2
F	O	3
F	F	4

Table 3.2. Markov states of a two-component system.

- O: operations F: Failed

Two-Component System

- The changes of states are illustrated by a state transition diagram shown in Figure 3.9.

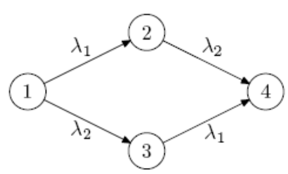


Figure 3.9. State transition diagram for a two independent component system.

- The failure rates λ1 and λ2 for components 1 and 2 indicate the rates at which the transitions are made between the states.
  - The two components are assumed to be independent and non-repairable.

Two-Component System

- If the components are in a serial configuration, then any component failure causes system failure. So, only the state 1 is the operational state. States 2, 3 and 4 are failed states.
- If the components are in parallel, both components must fail to have a system failure.
- Therefore, the states 1, 2 and 3 are the operational states, whereas the state 4 is a failed state.

State Transition Diagram Simplification

- It is often possible to reduce the size of a state transition diagram without a sacrifice (از دست دادن) in accuracy.
  - For example, suppose the components in the two component system shown in Figure 3.9 are in parallel. If the components have identical failure rates $\lambda_1=\lambda_2=l$, then it is not necessary to distinguish between the states 2 and 3.
  - Both states represent a condition where one component is operational and one is failed. So, we can merge these two states into one. (Figure 3.10).
  - The assignments of the state numbers in the simplified transition diagram are shown in Table 3.3.

Component 1	Component 2	State Number
O	*O*	1
O	*F*	2
F	*O*	2
F	*F*	3

Table 3.3. Markov states of a simplified state transition diagram of a two-component parallel system.

State Transition Diagram Simplification

- Since the failures of components are assumed to be independent events, the transition rate from the state 1 to the state 2 in Figure 3.10 is the sum of the transition rates from the state 1 to the states 2 and 3 in Figure 3.9, i. e. 2λ.

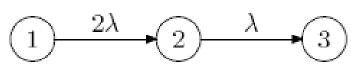


Figure 3.10. Simplified state transition diagram of a two-component parallel system.

5. Dependability Computation Methods

- In this section we study how reliability block diagrams and Markov processes can be used to evaluate system dependability:
  - **Computation Using RBDs**
  - **Computation using Markov processes**

5.1 Computation Using RBDs

- Reliability block diagrams can be used to compute system reliability as well as system availability.
  - □ **Reliability computation**
  - □ **Availability computation**

Reliability Computation

- To compute the reliability of a system represented by a reliability block diagram, we need first to break the system down into its serial and parallel parts.
- Next, the reliabilities of these parts are computed.
- Finally, the overall solution is composed from the reliabilities of the parts.

Reliability Computation

- Given a system consisting of n components with $R_i(t)$ being the reliability of the ith component, the reliability of the overall system is given by

$$R(t) = \begin{cases} \prod_{i=1}^{n} R_i(t) & \text{for a series structure,} \\ 1 - \prod_{i=1}^{n}(1 - R_i(t)) & \text{for a parallel structure.} \end{cases} \qquad (3.16)$$

- **Unreliability= 1- $R_i(t)$.**

Reliability Computation

- In a serial system, all components should be operational for a system to function correctly.
  - Hence, by rule (3.5), $R_{serial}(t) = \prod^{n}_{i=1} R_i(t)$.

$$p(A \cdot B) = p(A) \cdot p(B), \text{ if } A \text{ and } B \text{ are independent events.} \qquad (3.5)$$

Reliability Computation

- In a parallel system, only one of the components is required for a system to be operational.
- So, the unreliability of a parallel system equals to the probability that all n elements fail, i.e. $Q_{parallel}(t) = \prod^{n}_{i=1} Q_i(t) = \prod^{n}_{i=1}(1- R_i(t))$.
  - Hence, by rule 3.1,
    - $\boldsymbol{R_{parallel}(t) = 1\text{-}Q_{parallel}(t) = 1\text{-}\prod^{n}_{i=1} (1\text{-}R_i (t))}$.

$$p(\overline{A}) = 1 - p(A). \qquad (3.2)$$

Reliability Computation

- **Designing a reliable serial system is difficult.**
  - For example, if a serial system with 100 components is to be build, and each of the components has a reliability 0.999, the overall system reliability is $0.999^{100} = 0.905$.
- On the other hand, a parallel system can be made reliable despite the unreliability of its component parts.
  - For example, a parallel system of four identical modules with the module reliability 0.95, has the system reliability $1\text{-}(1\text{-}0.95)^4$ = 0.99999375.
- **Clearly, however, the cost of the parallelism can be high.**

Availability Computation

- **If we assume that the failure and repair times are independent, then we can use reliability block diagrams to compute the system availability.**
- This situation occurs when the system has enough spare resources to repair all the failed components simultaneously.

Availability Computation

- Given a system consisting of n components with $A_i(t)$ being the availability of the ith component, the availability if the overall system is given by

$$A(t) = \begin{cases} \prod_{i=1}^{n} A_i(t) & \text{for a series structure,} \\ 1 - \prod_{i=1}^{n}(1 - A_i(t)) & \text{for a parallel structure.} \end{cases} \quad (3.17)$$

Availability Computation

- The combined availability of **two components in series** is always lower than the availability of the individual components.
  - For example, if one component has the availability 99% (3.65 days/year downtime) and another component has the availability 99.99% (52 minutes/year downtime), then the availability of the system consisting of these two components in serial is 98.99% (3.69 days/year downtime).
- Contrary, a **parallel system consisting of three identical components** with the individual availability 99% has availability 99.9999 (31 seconds/year downtime).

5.2 Computation using Markov Processes

- In this section we show how Markov processes are used to evaluate system dependability.
  - Continuous Time Markov Chains (CTMCs) are the most important class of Markov processes for dependability analysis, so the presentation is focused on this model.
- The aim of Markov processes analysis is to calculate $P_i(t)$, the probability that the system is in the state *i* at time *t*.
  - Once this is known, the system reliability, availability or safety can be computed as a sum taken over all the operating states.

5.2 Computation using Markov Processes

- Let us designate the state 1 as the state in which all the components are operational.
- Assuming that at $t = 0$ the system is in state 1, we get $P_1(0) = 1$.
- Since at any time the system can be only in one state, $P_i(0) = 0$; $\forall i \neq 1$, and we have
  - $$\sum_{i \in O \cup F} P_i(t) = 1 \qquad (3.18)$$
  - where the sum is over all possible states.

5.2 Computation using Markov Processes

- To determine the $P_i(t)$, we derive a set of differential equations, one for each state of the system.
- These equations are called ***state transition equations*** because they allow the $P_i(t)$ to be determined in terms of the rates (failure, repair) at which transitions are made from one state to another.
- **State transition equations are usually presented in matrix form.**
  - The matrix M whose entry m_{ij} is the rate of transition between the states i and j is called the ***transition matrix*** associated with the system.

5.2 Computation using Markov Processes

- We use first index *i* for the columns of the matrix and the second index *j* for the rows, i.e. *M* has the following structure

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{21} & \dots & m_{k1} \\ m_{12} & m_{22} & \dots & m_{k1} \\ & & \dots & \\ m_{1k} & m_{2k} & \dots & m_{kk} \end{bmatrix}.$$

  - □ where k is the number of states in the state transition diagram representing the system.

5.2 Computation using Markov Processes

- In reliability or availability analysis the components of the system are normally assumed to be in either **operational** or **failed** states.
  - □ So, if a system consists of n components, then $k \leq 2^n$.
- In safety analysis, where the system can fail in either a safe or an unsafe way, k can be up to 3^n.
  - □ **The entries in each column of the transition matrix must sum up to 0.**
  - □ So, the entries m_{ii} corresponding to self-transitions are computed as $-\sum m_{ij}$, for all $j \in \{1, 2, \dots, k\}$ such that $j \neq i$.

5.2 Computation using Markov Processes

- For example, the transition matrix for the state transition diagram of a single component system shown in Figure 3.6 is:

$$\mathbf{M} = \begin{bmatrix} -\lambda & 0 \\ \lambda & 0 \end{bmatrix}. \tag{3.19}$$

  - The rate of the transition between the states 1 and 2 is λ, therefore the $m_{12} = \lambda$.
  - Therefore, $m_{11} = -\lambda$. The rate of transition between the states 2 and 1 is 0, so $m_{21} = 0$ and thus $m_{22} = 0$.

5.2 Computation using Markov Processes

- Similarly, the transition matrix for the state transition diagram in Figure 3.7,which incorporates repair, is

$$\mathbf{M} = \begin{bmatrix} -\lambda & \mu \\ \lambda & -\mu \end{bmatrix}. \tag{3.20}$$

5.2 Computation using Markov Processes

- The transition matrix for the state transition diagram in Figure 3.8, is of size 33, since, for safety analysis, the system is modeled to be in three different states: operational, failed-safe failed-unsafe.

$$\mathbf{M} = \begin{bmatrix} -\lambda & 0 & 0 \\ \lambda C & 0 & 0 \\ \lambda(1-C) & 0 & 0 \end{bmatrix}. \qquad (3.21)$$

5.2 Computation using Markov Processes

- The transition matrix for the simplified state transition diagram of the two component system, shown in Figure 3.10 is

$$\mathbf{M} = \begin{bmatrix} -2\lambda & 0 & 0 \\ 2\lambda & -\lambda & 0 \\ 0 & \lambda & 0 \end{bmatrix}. \qquad (3.22)$$

5.2 Computation using Markov Processes

- The examples above illustrate two important properties of transition matrices.
- **One, which we have mentioned before, is that the sum of the entries in each column is zero.**
  - Positive sign of an ij^{th} entry indicates that the transition originates in the i^{th} state.
  - Negative sign of an ij^{th} entry indicates that the transition terminates in the i^{th} state.

5.2 Computation using Markov Processes

- **Second property of the transition matrix is that it allows us to distinguish between the operational and failed states.**
  - In reliability analysis, once a system failed, a failed state cannot be leaved. Therefore, each failed state i has a zero diagonal element m_{ii}.
  - This is not the case, however, when availability or safety are computed, as one can see from (3.20) and (3.21).

5.2 Computation using Markov Processes

- Using state transition matrices, state transition equations are derived as follows.
  - Let **P**(t) be a vector whose ith element is the probability $P_i(t)$ that the system is in state i at time t. Then the matrix representation of a system of state transition equations is given by

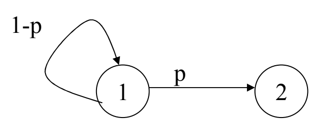
$$\frac{d}{dt}\mathbf{P}(t) = \mathbf{M} \cdot \mathbf{P}(t). \qquad (3.23)$$

- ***این رابطه از کجا بدست آمده است؟***

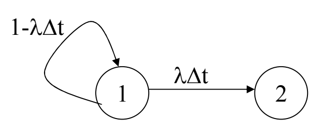
- همانگونه که قبلاً هم دیدیم، با توجه به تعریف قابلیت اطمینان داریم:
- $R(t) = P(X > t) = 1\text{-}F(t)$
- حال اگر نرخ خرابی طبق توزیع نمایی باشد، خواهیم داشت:
- $R(t) = 1\text{-}(1\text{-}e^{-\lambda t}) = e^{-\lambda t}$
- حال با توجه به تعریف عدم اطمینان خواهیم داشت:
- $Q(t) = 1 - R(t) = 1\text{-}\ e^{-\lambda t}$
  - تابع فوق احتمال خراب شدن در در زمان t مشخص می‌کند، در صورتی که در زمان صفر سیستم سالم باشد.
- می‌توانیم تابع فوق را از زمان t به t+Δt در نظر بگیریم. یعنی در t سالم بوده و در t+Δt خراب شود. آنگاه:
- $Q(\Delta t) = 1 - R(\Delta t) = 1\text{-}\ e^{-\lambda\Delta t}$

- از طرفی سری زیر را برای e^x داریم:
- $e^x = 1+x+x^2/2!+\ldots$
- آنگاه:
- $e^{-\lambda\Delta t} = 1+(-\lambda\Delta t)+(-\lambda\Delta t)^2/2!+\ldots$
- $1- e^{-\lambda\Delta t} = \lambda\Delta t-(-\lambda\Delta t)^2/2!-\ldots$
  - آنگاه چون Δt مقدار خیلی کوچکی است، خواهیم داشت: $(-\lambda\Delta t)^2/2!-\ldots \approx 0$
- آنگاه:
- $Q(t) = 1- e^{-\lambda\Delta t} \approx \lambda\Delta t$

- **حال یک سیستم یک مولفه‌ای را در نظر بگیرید که طبق توزیع نمایی با نرخ λ خراب می‌شود. یک زنجیره مارکوف برای این سیستم بدست می آوریم:**



- **احتمال اینکه پس از زمان Δt از حالت ۱ به ۲ برویم:**
  - $\mathbf{p = Q(t)} = \lambda\Delta t$
- **آنگاه:**



- حال می‌توانیم ماتریس TPM را برای MC فوق بنویسیم:
- $$P=\begin{bmatrix}1-\lambda\Delta t & 0\\ \lambda\Delta t & 0\end{bmatrix} \qquad p(t+\Delta t)=P\,p(t)$$

$$=>\begin{bmatrix}p_1(t+\Delta t)\\ p_2(t+\Delta t)\end{bmatrix}=\begin{bmatrix}1-\lambda\Delta t & 0\\ \lambda\Delta t & 0\end{bmatrix}\begin{bmatrix}p_1(t)\\ p_2(t)\end{bmatrix} => \begin{cases}p_1(t+\Delta t)=(1-\lambda\Delta t)\,p_1(t)\\ p_2(t+\Delta t)=\lambda\Delta t\,p_1(t)\end{cases}$$

$$\begin{cases}p_1(t)+p_2(t)=1\\ p_1(t+\Delta t)+p_2(t+\Delta t)=1\end{cases}$$

$$=>\begin{cases}p_1(t+\Delta t)=(1-\lambda\Delta t)\,p_1(t)\\ 1-p_2(t+\Delta t)=(1-\lambda\Delta t)(1-p_2(t))\end{cases}$$

$$=>\begin{cases}p_1(t+\Delta t)-p_1(t)=-\lambda\Delta t\,p_1(t)\\ p_2(t+\Delta t)-p_2(t)=\lambda\Delta t(1-p_2(t))\end{cases} \qquad =>\begin{cases}\dfrac{p_1(t+\Delta t)-p_1(t)}{\Delta t}=-\lambda\,p_1(t)\\ \dfrac{p_2(t+\Delta t)-p_2(t)}{\Delta t}=\lambda\,p_2(t)\end{cases}$$

$$=>\begin{cases}\lim\limits_{\Delta t\to 0}\dfrac{p_1(t+\Delta t)-p_1(t)}{\Delta t}=\lim\limits_{\Delta t\to 0}(-\lambda\,p_1(t))\\ \lim\limits_{\Delta t\to 0}\dfrac{p_2(t+\Delta t)-p_2(t)}{\Delta t}=\lim\limits_{\Delta t\to 0}(\lambda\,p_1(t))\end{cases}$$

$$=>\begin{cases}\dfrac{d}{dt}p_1(t)=-\lambda\,p_1(t)\\ \dfrac{d}{dt}p_2(t)=\lambda\,p_1(t)\end{cases} \qquad =>\frac{d}{dt}\begin{bmatrix}p_1(t)\\ p_2(t)\end{bmatrix}=\begin{bmatrix}-\lambda & 0\\ \lambda & 0\end{bmatrix}\begin{bmatrix}p_1(t)\\ p_2(t)\end{bmatrix}$$

$$\boxed{=>\frac{d}{dt}p(t)=M\;p(t)}$$

5.2 Computation using Markov Processes

- **Once the system of equations is solved and the $P_i(t)$ are known, the system reliability, availability or safety can be computed as a sum taken over all the operating states.**
- We illustrate the computation process on a number of simple examples.

Reliability Evaluation - Independent Components Case

- Let us first compute **reliability of a parallel system** consisting of **two independent components** which we have considered before (Figure 3.9).
- Applying (3.23) to the matrix (3.22) we get

$$\frac{d}{dt}\begin{bmatrix} P_1(t) \\ P_2(t) \\ P_3(t) \end{bmatrix} = \begin{bmatrix} -2\lambda & 0 & 0 \\ 2\lambda & -\lambda & 0 \\ 0 & \lambda & 0 \end{bmatrix} \cdot \begin{bmatrix} P_1(t) \\ P_2(t) \\ P_3(t) \end{bmatrix}.$$

Reliability Evaluation - Independent Components Case

- The above matrix form represents the following system of state transition equations

$$\begin{cases} \frac{d}{dt}P_1(t) = -2\lambda P_1(t) \\ \frac{d}{dt}P_2(t) = 2\lambda P_1(t) - \lambda P_2(t) \\ \frac{d}{dt}P_3(t) = \lambda P_2(t) \end{cases}$$

- By solving this system of equations, we get: (???)

$$\boxed{\begin{aligned} P_1(t) &= e^{-2\lambda t} \\ P_2(t) &= 2e^{-\lambda t} - 2e^{-2\lambda t} \\ P_3(t) &= 1 - 2e^{-\lambda t} + e^{-2\lambda t} \end{aligned}}$$

Laplace Transform

- In the branch of mathematics called functional analysis, the **Laplace transform**, $\mathcal{L}\{f(t)\}$, is a linear operator on a function $f(t)$ (*original* (time domain)) with a real argument t ($t \geq 0$) that transforms it to a function $F(s)$ (*image* (frequency domain)) with a complex argument s.
- The Laplace transform is particularly useful in solving linear ordinary differential equations such as those arising in the analysis of electronic circuits.
- The Laplace transform of a function $f(t)$, defined for all real numbers $t \geq 0$, is the function $F(s)$, defined by: $$F(s) = \mathcal{L}\{f(t)\} = \int_0^\infty e^{-st} f(t)\, dt$$

Laplace Transform Table

$f(t) = \mathcal{L}^{-1}\{F(s)\}(t)$	$F(s) = \mathcal{L}\{f(t)\}(s) = \int_0^\infty e^{-st} f(t)dt$
1	$\frac{1}{s}\ ,\quad s > 0$
$t^n\ ,\quad n$ an integer	$\frac{n!}{s^{n+1}}\ ,\quad s > 0$
e^{at}	$\frac{1}{s-a}\ ,\quad s > a$
$\sin bt$	$\frac{b}{s^2+b^2}\ ,\quad s > 0$
$\cos bt$	$\frac{s}{s^2+b^2}\ ,\quad s > 0$
$e^{at} f(t)$	$F(s-a)$
$e^{at} t^n\quad n$ an integer	$\frac{n!}{(s-a)^{n+1}}\ ,\quad s > a$

Laplace Transform Table

$e^{at} \sin bt$	$\frac{b}{(s-a)^2+b^2}\ ,\quad s > a$
$e^{at} \cos bt$	$\frac{(s-a)}{(s-a)^2+b^2}\ ,\quad s > a$
$t \sin bt$	$\frac{2bs}{(s^2+b^2)^2}\quad s > 0$
$t \cos bt$	$\frac{s^2-b^2}{(s^2+b^2)^2}\ ^*,\quad s > 0$
$u_c(t)f(t),\ c \geq 0$ $u_c(t)f(t-c),\ c \geq 0^{**}$	$e^{-cs}\mathcal{L}\{f(t+c)\}(s)$ $e^{-cs}\mathcal{L}\{f(t)\}(s)$
$y' = \dot{y} = \frac{dy}{dt}$	$sY(s) - y(0)$
$y'' = \ddot{y} = \frac{d^2y}{dt^2}$	$s^2Y(s) - sy(0) - \dot{y}(0)$

Laplace Transform Table

General

$f(t)$	$F(s) = \int_0^\infty f(t)e^{-st}\,dt$
$f + g$	$F + G$
$\alpha f \ (\alpha \in \mathbf{R})$	αF
$\frac{df}{dt}$	$sF(s) - f(0)$
$\frac{d^k f}{dt^k}$	$s^k F(s) - s^{k-1} f(0) - s^{k-2}\frac{df}{dt}(0) - \cdots - \frac{d^{k-1}f}{dt^{k-1}}(0)$
$g(t) = \int_0^t f(\tau)\,d\tau$	$G(s) = \frac{F(s)}{s}$
$f(\alpha t),\ \alpha > 0$	$\frac{1}{\alpha}F(s/\alpha)$
$e^{at}f(t)$	$F(s-a)$
$tf(t)$	$-\frac{dF}{ds}$
$t^k f(t)$	$(-1)^k \frac{d^k F(s)}{ds^k}$
$\frac{f(t)}{t}$	$\int_s^\infty F(s)\,ds$
$g(t) = \begin{cases} 0 & 0 \le t < T \\ f(t-T) & t \ge T \end{cases},\ T \ge 0$	$G(s) = e^{-sT}F(s)$

Laplace Transform Table

Specific

1	$\frac{1}{s}$
δ	1
$\delta^{(k)}$	s^k
t	$\frac{1}{s^2}$
$\frac{t^k}{k!},\ k \ge 0$	$\frac{1}{s^{k+1}}$
e^{at}	$\frac{1}{s-a}$
$\cos\omega t$	$\frac{s}{s^2+\omega^2} = \frac{1/2}{s-j\omega} + \frac{1/2}{s+j\omega}$
$\sin\omega t$	$\frac{\omega}{s^2+\omega^2} = \frac{1/2j}{s-j\omega} - \frac{1/2j}{s+j\omega}$
$\cos(\omega t + \phi)$	$\frac{s\cos\phi - \omega\sin\phi}{s^2+\omega^2}$
$e^{-at}\cos\omega t$	$\frac{s+a}{(s+a)^2+\omega^2}$
$e^{-at}\sin\omega t$	$\frac{\omega}{(s+a)^2+\omega^2}$

Solution Using Laplace Transform

$$\begin{cases} \frac{d}{dt}P_1(t) = -2\lambda P_1(t) \\ \frac{d}{dt}P_2(t) = 2\lambda P_1(t) - \lambda P_2(t) \\ \frac{d}{dt}P_3(t) = \lambda P_2(t) \end{cases}$$

$$\boxed{\mathcal{L}(f') = sF(s) - f(0)}$$

معادلات لاپلاس

$$\begin{cases} s\,p_1(s) - p_1(0) = -2\lambda\, p_1(s) \\ s\,p_2(s) - p_2(0) = 2\lambda\, p_1(s) - \lambda\, p_2(s) \\ s\,p_3(s) - p_3(0) = \lambda\, p_2(s) \end{cases}$$

حالت اولیه

$$\begin{bmatrix} P_1(0) \\ P_2(0) \\ P_3(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\Rightarrow \begin{cases} s\,p_1(s) - 1 = -2\lambda\, p_1(s) \\ s\,p_2(s) = 2\lambda\, p_1(s) - \lambda\, p_2(s) \\ s\,p_3(s) = \lambda\, p_2(s) \end{cases} \quad \Rightarrow \begin{cases} s\,p_1(s) - 1 = -2\lambda\, p_1(s) \\ s\,p_2(s) = 2\lambda\, p_1(s) - \lambda\, p_2(s) \\ s\,p_3(s) = \lambda\, p_2(s) \end{cases}$$

Solution Using Laplace Transform

$$\Rightarrow \begin{cases} p_1(s) = \dfrac{1}{s+2\lambda} \\ p_2(s) = \dfrac{2\lambda}{(s+\lambda)(s+2\lambda)} \\ p_3(s) = \dfrac{\lambda^2}{s(s+\lambda)(s+2\lambda)} \end{cases}$$

باید به فرم
قابل قبول
تبدیل شود

$$\Rightarrow \begin{cases} \dfrac{2\lambda}{(s+\lambda)(s+2\lambda)} = \dfrac{a}{(s+\lambda)} + \dfrac{b}{(s+2\lambda)} \\ \dfrac{\lambda^2}{s(s+\lambda)(s+2\lambda)} = \dfrac{x}{s} + \dfrac{y}{(s+\lambda)} + \dfrac{z}{(s+2\lambda)} \end{cases} \quad \Rightarrow \begin{cases} a = 2 \\ b = -2 \end{cases} \quad \Rightarrow \begin{cases} x = 1 \\ y = -2 \\ z = 1 \end{cases}$$

$$\Rightarrow \begin{cases} p_1(s) = \dfrac{1}{s+2\lambda} \\ p_2(s) = \dfrac{2}{(s+\lambda)} + \dfrac{-2}{(s+2\lambda)} \\ p_3(s) = \dfrac{1}{s} + \dfrac{-2}{(s+\lambda)} + \dfrac{1}{(s+2\lambda)} \end{cases} \quad \Rightarrow \quad \boxed{\begin{aligned} P_1(t) &= e^{-2\lambda t} \\ P_2(t) &= 2e^{-\lambda t} - 2e^{-2\lambda t} \\ P_3(t) &= 1 - 2e^{-\lambda t} + e^{-2\lambda t} \end{aligned}}$$

Reliability Evaluation - Independent Components Case

- Since the $P_i(t)$ are known, we can now calculate the reliability of the system.
- **For the parallel configuration, both components should fail to have a system failure.**
  - **Therefore, the reliability of the system is the sum of probabilities $P_1(t)$ and $P_2(t)$:**

$$R_{parallel}(t) = 2e^{-\lambda t} - e^{-2\lambda t} \qquad (3.24)$$

Reliability Evaluation - Independent Components Case

- In general case, the reliability of the system is computed as a function using the equation

$$R(t) = \sum_{i \in O} P_i(t),$$

  - **where the sum is taken over all the operating states *O*.**
- Alternatively, the reliability can be calculated

$$R(t) = 1 - \sum_{i \in F} P_i(t),$$

  - where the sum is taken over all the states F in which the system has failed.

Reliability Evaluation - Independent Components Case

- Comparison with RBDs: Note that, for constant failure rates, the component reliability is $R(t) = e^{-\lambda t}$.
- Therefore, the equation (3.24) can be written as $R_{parallel}(t) = 2R^2 - R$,
  - which agrees with the expression (3.16) derived using reliability block diagrams.

$$R(t) = \begin{cases} \prod_{i=1}^{n} R_i(t) & \text{for a series structure,} \\ 1 - \prod_{i=1}^{n}(1 - R_i(t)) & \text{for a parallel structure.} \end{cases} \quad (3.16)$$

- Two results are the same, because in this example we assumed the failure rates to be mutually independent.

Reliability Evaluation - Dependent Components Case

- The value of Markov processes become evident in situations in which **component failure rates are no longer assumed to be independent of the system state.**
- One of the common cases of dependence is **load-sharing components**, which we consider next.
- Another possibility is the case of **standby components**, which is considered in the availability computation section.

Reliability Evaluation - Dependent Components Case

- The word *load* is used in a broad sense of the stress on a system.

  - کلمه **بار** برای مواقع گسترده‌ای در مورد فشار کاری وارده به سیستم استفاده می‌شود.
  - This can be an **electrical load**, a **load caused by high temperature**, or an **information load**.

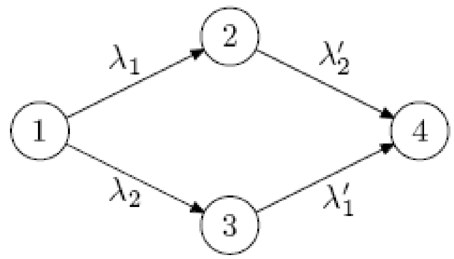
- **On practice, failure rates are found to increase with loading.**
  - Suppose that two components share a load. If one of the component fails, the additional load on the second component is likely to increase its failure rate.

Reliability Evaluation - Dependent Components Case

- State transition diagram of a two-component parallel load-sharing system?

Reliability Evaluation - Dependent Components Case

- To model load-sharing failures, consider the state transition diagram of a two-component parallel system shown in Figure 3.11.



- *Figure 3.11.* State transition diagram of a two-component parallel system with load sharing.

Reliability Evaluation - Dependent Components Case

- As before, we have four states. However, after one component failure, the failure rate of the second component increases.
- The increased failure rates of the components 1 and 2 are denoted with λ'_1 and λ'_2, respectively.

Reliability Evaluation - Dependent Components Case

- From the state transition diagram in Figure 3.11, we can derive the state transition equations for $P_i(t)$. In the matrix form they are

$$\frac{d}{dt}\begin{bmatrix} P_1(t) \\ P_2(t) \\ P_3(t) \\ P_4(t) \end{bmatrix} = \begin{bmatrix} -\lambda_1-\lambda_2 & 0 & 0 & 0 \\ \lambda_1 & -\lambda'_2 & 0 & 0 \\ \lambda_2 & 0 & -\lambda'_1 & 0 \\ 0 & \lambda'_2 & \lambda'_1 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_1(t) \\ P_2(t) \\ P_3(t) \\ P_4(t) \end{bmatrix}.$$

Reliability Evaluation - Dependent Components Case

- By expanding the matrix form, we get the following system of equations

$$\begin{cases} \frac{d}{dt}P_1(t) = (-\lambda_1-\lambda_2)P_1(t) \\ \frac{d}{dt}P_2(t) = \lambda_1 P_1(t) - \lambda'_2 P_2(t) \\ \frac{d}{dt}P_3(t) = \lambda_2 P_1(t) - \lambda'_1 P_3(t) \\ \frac{d}{dt}P_4(t) = \lambda'_2 P_2(t) + \lambda'_1 P_3(t). \end{cases}$$

Reliability Evaluation - Dependent Components Case

- The solution of this system of equation is

$$P_1(t) = e^{(-\lambda_1-\lambda_2)t}$$

$$P_2(t) = \frac{\lambda_1}{\lambda_1+\lambda_2-\lambda_2'} e^{\lambda_2' t} - \frac{\lambda_1}{\lambda_1+\lambda_2-\lambda_2'} e^{(-\lambda_1-\lambda_2)t}$$

$$P_3(t) = \frac{\lambda_2}{\lambda_1+\lambda_2-\lambda_1'} e^{\lambda_1' t} - \frac{\lambda_2}{\lambda_1+\lambda_2-\lambda_1'} e^{(-\lambda_1-\lambda_2)t}$$

$$P_4(t) = 1 - e^{(-\lambda_1-\lambda_2)t} - \frac{\lambda_1}{\lambda_1+\lambda_2-\lambda_2'} e^{\lambda_2' t} + \frac{\lambda_1}{\lambda_1+\lambda_2-\lambda_2'} e^{(-\lambda_1-\lambda_2)t} - \frac{\lambda_2}{\lambda_1+\lambda_2-\lambda_1'} e^{\lambda_1' t} + \frac{\lambda_2}{\lambda_1+\lambda_2-\lambda_1'} e^{(-\lambda_1-\lambda_2)t}.$$

Reliability Evaluation - Dependent Components Case

- Finally, since **both components should fail for the system to fail, the reliability is equal to: 1-*P*$_4$(*t*)**, yielding the expression

$$R_{parallel}(t) = e^{(-\lambda_1-\lambda_2)t} + \frac{\lambda_1}{\lambda_1+\lambda_2-\lambda_2'} e^{\lambda_2' t} - \frac{\lambda_1}{\lambda_1+\lambda_2-\lambda_2'} e^{(-\lambda_1-\lambda_2)t} + \frac{\lambda_2}{\lambda_1+\lambda_2-\lambda_1'} e^{\lambda_1' t} - \frac{\lambda_2}{\lambda_1+\lambda_2-\lambda_1'} e^{(-\lambda_1-\lambda_2)t}.$$

  - If $\lambda'_1 = \lambda_1$ and $\lambda'_2 = \lambda_2$, the above equation is equal to (3.24).

Reliability Evaluation - Dependent Components Case

- The **effect of the increased loading** can be illustrated as follows:
  - ☐ Assume that the two components are identical, i.e. $\lambda_1 = \lambda_2 = \lambda$ and $\lambda'_1 = \lambda'_2 = \lambda'$.
  - ☐ Then, the equation (3.26) reduces to

$$R_{parallel}(t) = \frac{2\lambda}{2\lambda - \lambda'} e^{-\lambda' t} - \frac{\lambda'}{2\lambda - \lambda'} e^{-2\lambda t}.$$

Reliability Evaluation - Dependent Components Case

- Figure 3.12 shows the reliability of a two-component parallel system with load-sharing for different values of λ'.

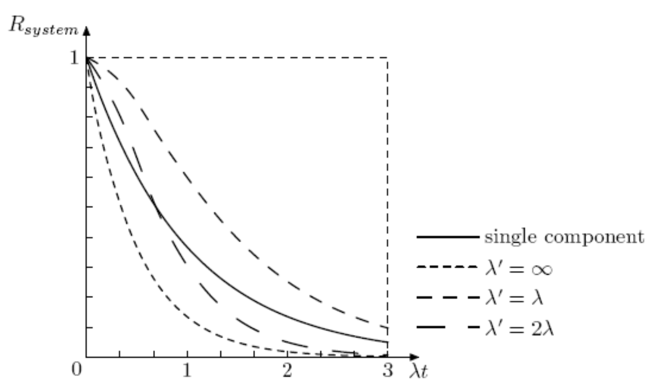


Figure 3.12. Reliability of a two-component parallel system with load sharing.

Reliability Evaluation - Dependent Components Case

- The reliability $e^{-\lambda t}$ of a single-component system is also plotted for a comparison.
- **In case of λ'=λ two components are independent**, so the reliability is given by (3.23). (چون خراب شدن یکی بر دیگری اثر ندارد)
- **λ'= ∞ is the case of total dependency.**
  - The failure of one component brings an immediate failure of another component. So, **the reliability equals to the reliability of a serial system with two components (3.16)**.
- **It can been seen that, the more the values of λ' exceeds the value of λ, the closer the reliability of the system approaches serial system with two components reliability.**

Availability Evaluation

- In availability analysis, as well as in reliability analysis, there are situations in which the component failures cannot be considered **independent of one another**.
  - These include shared-load systems and systems with standby components, which are repairable.

Availability Evaluation

- The dependencies between component failures can be analyzed using Markov methods, provided that the failures are detected and that the failure and repair rates are **time-independent**.
- **There is a fundamental difference between treatment of repair for reliability and availability analysis.**
  - In reliability calculations, components are allowed to be repaired only as long as the system has not failed. In availability calculations, the components can also be repaired after the system failure.

Availability Evaluation

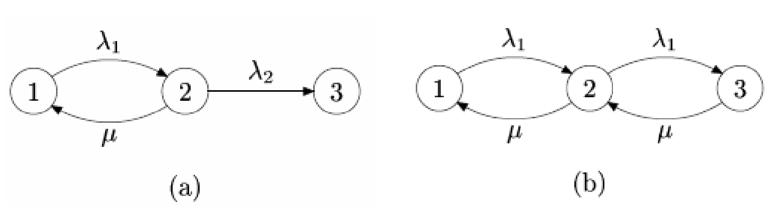
- The difference is best illustrated on a simple example of a system with two components, **one primary** and **one standby**.
  - The standby component is held in reserve and only brought to operation when the primary component fails.
  - We assume that there is a perfect fault detection unit which detects a failure in the primary component and replace it by the standby component.
  - We also assume that the standby component cannot fail while it is in the standby mode.

Availability Evaluation

- The state transition diagrams of the standby system for reliability and availability analysis?

Availability Evaluation

- The state transition diagrams of the standby system for reliability and availability analysis are shown in Figure 3.13(a) and (b), respectively.



- *Figure 3.13.* Sate transition diagrams for a standby two-component system (a) for reliability analysis, (b) for availability analysis.

Availability Evaluation

- The states are numbered according to the Table 3.4.

Component 1	Component 2	State Number
O	*O*	1
F	*O*	2
F	*F*	3

- *Table 3.4.* Markov states of a simplified state transition diagram of a two-component parallel system incorporating repair.

Availability Evaluation

- When the primary component fails, there is a transition between the states 1 and 2.
- If a system is in the state 2 and the backup component fails, there is a transition to the state 3.
- Since we assumed that the backup unit cannot fail while in the standby mode, the combination (O, F) cannot occur.
- The states 1 and 2 are operational states. The state 3 is the failed state.

Availability Evaluation

- Suppose the primary unit can be repaired with a rate μ.
  - For reliability analysis, this implies that a transition between the states 2 and 1 is possible.
- The corresponding transition matrix?

Availability Evaluation

- The corresponding transition matrix is given by

$$\mathbf{M} = \begin{bmatrix} -\lambda_1 & \mu & 0 \\ \lambda_1 & -\lambda_2 - \mu & 0 \\ 0 & \lambda_2 & 0 \end{bmatrix}.$$

Availability Evaluation

- For **availability analysis**, we should be able to repair the backup unit as well.
- This adds a transition between the states 3 and 2. We assume that the repair rates for primary and backup units are the same. We also assume that the backup unit will be repaired first. The corresponding transition matrix is given by

$$\mathbf{M} = \begin{bmatrix} -\lambda_1 & \mu & 0 \\ \lambda_1 & -\lambda_2 - \mu & \mu \\ 0 & \lambda_2 & -\mu \end{bmatrix}. \qquad (3.27)$$

Availability Evaluation

- One can see that, in the matrix for availability calculations, none of the diagonal elements is zero. This is because the system should be able to recover from the failed state.
- By solving the system of state transition equations, we can get $P_i(t)$ and compute the availability of the system as (حل نمایید)

$$A(t) = 1 - \sum_{i \in F} P_i(t), \qquad (3.28)$$

  - where the sum is taken over all the failed states F.

Availability Evaluation

- Usually, the **steady state availability** rather than the time-dependent availability is of interest.
- The steady state availability can be computed in a simpler way.
  - We note that, as time approach infinity, the derivative on the right-hand side of the equation 3.23 vanishes and we get a time-independent relationship

$$\mathbf{M} \cdot \mathbf{P}(\infty) = 0. \qquad (3.29)$$

Availability Evaluation

- In our example, for matrix (3.27) this represents a system of equations

$$\begin{cases} -\lambda_1 P_1(\infty) + \mu P_2(\infty) = 0 \\ \lambda_1 P_1(\infty) - (\lambda_2 + \mu) P_2(\infty) + \mu P_3(\infty) = 0 \\ \lambda_2 P_2(\infty) - \mu P_3(\infty) = 0 \end{cases}$$

Availability Evaluation

- Since these three equations are linearly dependent, they are not sufficient to solve for $P(\infty)$.
- The needed piece of additional information is the condition (3.18) that the sum of all probabilities is one:

$$\sum_i P_i(\infty) = 1. \tag{3.30}$$

Availability Evaluation

- If we assume $\lambda_1 = \lambda_2 = \lambda$, then we get

$$P_1(\infty) = \left[1 + \frac{\lambda}{\mu} + (\frac{\lambda}{\mu})^2\right]^{-1},$$
$$P_2(\infty) = \left[1 + \frac{\lambda}{\mu} + (\frac{\lambda}{\mu})^2\right]^{-1} \frac{\lambda}{\mu},$$
$$P_3(\infty) = \left[1 + \frac{\lambda}{\mu} + (\frac{\lambda}{\mu})^2\right]^{-1} \left(\frac{\lambda}{\mu}\right)^2.$$

Availability Evaluation

- The steady-state availability can be found by setting $t = \infty$ in (3.28)

$$A(\infty) = 1 - \left[1 + \frac{\lambda}{\mu} + (\frac{\lambda}{\mu})^2\right]^{-1} \left(\frac{\lambda}{\mu}\right)^2.$$

- If we further assume that $\lambda/\mu << 1$, we can write

$$A(\infty) \approx 1 - \left(\frac{\lambda}{\mu}\right)^2.$$

Availability Evaluation

- To summarize, steady-state availability problems are solved by the same procedure as time-dependent availability.
  - Any n-1 of the n equations represented by (3.29) are combined with the condition 3.30 to solve for the components of $\mathbf{P}(\infty)$.
  - These are then substituted into (3.28) to obtain availability.

Safety Evaluation

- The main difference between safety calculation and reliability calculation is in the construction of the state transition diagram.
- As we mentioned before, for safety analysis, the **failed state is splitted into failed-safe and failed-unsafe ones**.
- Once the state transition diagram for a system is derived, the state transition equations are obtained and solved using same procedure as for reliability analysis.

Safety Evaluation

- As an example, consider the single component system shown in Figure 3.8.
- Its state transition matrix is given by (3.21). So, the state transition equations for $P_i(t)$ are given by

$$\frac{d}{dt}\begin{bmatrix} P_1(t) \\ P_2(t) \\ P_3(t) \end{bmatrix} = \begin{bmatrix} -\lambda & 0 & 0 \\ \lambda C & 0 & 0 \\ \lambda(1-C) & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_1(t) \\ P_2(t) \\ P_3(t) \end{bmatrix}.$$

Safety Evaluation

- The solution of this system of equations is

$$P_1(t) = e^{-\lambda t}$$

$$P_2(t) = C - Ce^{-\lambda t}$$

$$P_3(t) = (1 - C) - (1 - C)e^{-\lambda t}$$

Safety Evaluation

- The safety of the system is the sum of probabilities of being in the operational or failed-safe states, i.e.

$$S(t) = P_1(t) + P_2(t) = C + (1 - C)e^{-\lambda t}$$

  - At time $t = 0$ the safety of the system is 1, as expected.
  - As time approaches infinity, the safety approaches the fault detection coverage, $S(0)= C$. So, if $C = 1$, the system has a perfect safety.

تمرینها

- شماره تمرینها:
  - 3.9
  - 3.15
  - 3.19
  - 3.25
  - 3.28
  - 3.31
- مهلت تحویل: دو هفته