

FLOAREA BAICU

FIABILITATEA SISTEMELOR
INFORMATICE

(MODUL DE CURS)

Cuprins

CAPITOLUL 1

Concepte generale referitoare la fiabilitate

- 1.1. Definierea fiabilității
- 1.2. Obiective ale fiabilității în ciclul de viață al sistemelor
- 1.3. Defectări. Tipuri și evoluție
- 1.4. Fiabilitatea factorului uman
- 1.5. Evoluția în timp a defectelor
- 1.6. Tipuri de încercări pentru estimarea fiabilității

CAPITOLUL 2

Funcții specifice fiabilității, mentenanței și disponibilității

- 2.1. Funcții specifice fiabilității
 - 2.1.1. Funcția de fiabilitate, $R(t)$
 - 2.1.2. Probabilitatea de defectare, $F(t)$
 - 2.1.3. Densitatea de probabilitate a timpului de funcționare fără defectare, $f(t)$
 - 2.1.4. Rata de defectare, $z(t)$ sau $\lambda(t)$
 - 2.1.5. Media timpului de funcționare până la defectare, m
 - 2.1.6. Deviația standard, (σ) și dispersia (D)
 - 2.1.7. Cuantila timpului de funcționare, t_α
 - 2.1.8. Interdependența funcțiilor specifice fiabilității
- 2.2. Indicatori pentru fiabilitatea software
 - 2.2.1. Funcția de fiabilitate, $R(t)$
 - 2.2.2. Rata de manifestare a erorilor sau densitatea de defectare, $z(t)$
 - 2.2.3. Cuantila timpului de execuție, t_α
 - 2.2.4. Numărul de erori remanente, $N(t)$
 - 2.2.5. Durata medie de funcționare până la defectare
 - 2.2.6. Indicatori indirecti, specifici fiabilității previzionale a programelor
- 2.3. Metrice ce descriu fiabilitatea software
- 2.4. Mentenanța

CAPITOLUL 3

Principalele legi de distribuție statistică specifice fiabilității

- 3.1. Tipuri de distribuție statistică
- 3.2. Distribuția binominală
- 3.3. Distribuția Poisson

- 3.4. Distribuția normală (Gauss-Laplace)
 - 3.4.1. Funcția de distribuție normală
 - 3.4.2. Distribuția normală normată
 - 3.4.3. Valoarea medie și dispersia unei variabile cu distribuție normală
- 3.5. Distribuția exponențială
- 3.6. Distribuția log-normală

CAPITOLUL 4

Studiul fiabilității sistemelor pe baza blocurilor logice de fiabilitate

- 4.1. Definirea sistemelor complexe
- 4.2. Dezvoltarea modelului diagrame bloc de fiabilitate
- 4.3. Tipuri de conexiuni
 - 4.3.1. Sisteme de tip serie
 - 4.3.2. Sisteme de tip paralel
- 4.4. Arbori de evenimente
 - 4.4.1. Concepte de bază referitor la arbori de evenimente
 - 4.4.2. Descrierea și structura grafică a arborelui de defectare
 - 4.4.3. Evaluarea fiabilității sistemului sistemului utilizând arbori de defectare

CAPITOLUL 5

Metoda lanțurilor Markov pentru fiabilitatea sistemelor

- 5.1. Definirea lanțului Markov
- 5.2. Matricea stărilor de tranziție
- 5.3. Exemplu de calcul
- 5.4. Etapele aplicării metodei lanțurilor Markov

CAPITOLUL 6 și 7

Nu fac parte din modulul din curs

CAPITOLUL 8

Metode de estimare și validare a indicatorilor de fiabilitate

- 8.1. Valori estimate ale indicatorilor de fiabilitate
 - 8.1.1. Valori teoretice (adevărate) și estimate
 - 8.1.2. Calculul valorilor estimate ale indicatorilor de fiabilitate
 - 8.1.3. Caracteristicile estimărilor punctuale
- 8.2. Metode de estimare punctuală a parametrilor statici specifici fiabilității
 - 8.2.1. Metoda verosimilității maxime (maximum likelihood method)
 - 8.2.2. Metoda liniarizării
 - 8.2.3. Metoda momentelor
 - 8.2.4. Metoda celor mai mici pătrate

- 8.3. Metoda intervalelor de încredere pentru estimarea parametrilor statistici specifici fiabilității
- 8.4. Teste de verificare și validare a ipotezelor statistice
 - 8.4.1. Testul Kolmogorov-Smirnov
 - 8.4.2. Testul χ^2
 - 8.4.3. Testul secvențial al lui Wald

CAPITOLUL 9

Fiabilitatea previzională software

- 9.1. Modele de fiabilitate software și indicatori
- 9.2. Tehnici pentru îmbunătățirea software-ului bazate pe ciclul de viață al defectului software
- 9.3. Modele structurale pentru fiabilitatea software
- 9.4. Tehnici și modele pentru sisteme software tolerante la defectări
 - 9.4.1. Blocuri cu restabilire
 - 9.4.2. Structuri N-versionale
- 9.5. Teste de acceptare a rezultatelor

CAPITOLUL 10

Modele pentru fiabilitatea software în faza de testare și operare

- 10.1. Generalități
- 10.2. Modele bazate pe timpul mediu dintre defectări
 - 10.2.1. Modelul Jelinski-Moranda
 - 10.2.2. Modelul Goel-Okumoto I
 - 10.2.3. Modelul Littlewood
 - 10.2.4. Modelul Schick -Wolverton
- 10.3. Modele bazate pe numărarea defectelor
 - 10.3.1. Modelul Musa
 - 10.3.2. Modelul Shanthikumar
 - 10.3.3. Modelul Goel-Okumoto II
- 10.4. Validarea fiabilității programelor
 - 10.4.1. Etape în validarea fiabilității unui program
 - 10.4.2. Metode grafice

CAPITOLUL 11

Probleme specifice de fiabilitate hardware

- 11.1. Defecte specifice sistemelor hardware
- 11.2. Fiabilitatea circuitelor integrate VLSI
- 11.3. Fiabilitatea memoriilor semiconductoare
 - 11.3.1. Defectarea memoriilor semiconductoare
 - 11.3.3. Modalități de îmbunătățire a fiabilității memoriilor semiconductoare

11.4. Fiabilitatea microprocesoarelor

11.4.1. Aspecte specifice privind fiabilitatea microprocesoarelor

11.4.2. Autotestarea microprocesoarelor

11.4.3. Factori de care depinde fiabilitatea microprocesoarelor

11.4.4. Căi de îmbunătățire a fiabilității microprocesoarelor

11.4.4.1. Screening

11.4.4.2. Procesoare tolerante la erori tranziente

Capitolul 1. CONCEPTE GENERALE REFERITOARE LA FIABILITATE

1.1. Definirea fiabilității

Fiabilitatea sistemelor tehnice este definită ca fiind probabilitatea ca un sistem să-și îndeplinească funcția proiectată un interval de timp specificat, în condiții de utilizare prestabilite.

În această definiție sunt patru elemente cheie:

1. Fiabilitatea este definită ca *probabilitate*, pentru că defectarea sistemului este un fenomen întâmplător, poate să apară oricând și poate fi descrisă numai de funcții specifice probabilităților. Nu putem obține cu certitudine informații despre defectări individuale, cauze sau relații între defectări, ci numai cu o anumită probabilitate. Fiabilitatea este o știință stohastică, care permite estimarea statistică a comportării în timp a sistemelor cu ajutorul *probabilităților*, și în special al *probabilităților condiționate*. La începuturi a existat iluzia că este posibilă previziunea certă a funcționării unui sistem și au fost numeroase cercetări în acest domeniu. Experiența acumulată în timp a infirmat o asemenea posibilitate și a obligat la o abordare mai realistă, mai apropiată de comportamentul real, concretizată prin atașarea unui anumit *nivel de incertitudine* (sau de încredere) referitor la gradul de îndeplinire a serviciului în cursul intervalului de timp cerut. Probabilitatea de bună funcționare (succesul misiunii) poate fi foarte apropiată de certitudine, dar niciodată 100%, sigur, există întotdeauna un anumit risc de insucces, de preferință foarte mic, indiferent de cât s-ar investi în fiabilitate.

2. Fiabilitatea este corelată cu *funcția proiectată*, adică cu realizarea numai a acelei funcții avute în vedere la conceperea sistemului, înainte de dezvoltarea sa. Sistemul ca ansamblu trebuie să funcționeze fără defectare, deși pot exista componente ale sistemului care se pot defecta, fără că funcția proiectată a sistemului să fie afectată. Aceste componente pot fi înlocuite înainte de defectare în conformitate cu anumite criterii prestabilite și program de mentenanță (ex. sursele de alimentare, ventilatoare, anumite module funcționale).

3. Fiabilitatea se referă la o bună funcționare într-un anumit *interval de timp*, în afara acestui interval sistemul putând să fie funcțional sau nu. Pentru anumite sisteme timpul poate fi înlocuit cu număr cicluri de utilizare sau de rulări. Se folosește termenul de *ciclu de viață* sau *durată de viață*.

4. Fiabilitatea este restricționată de *condițiile prestabilite pentru operare*, condițiile concrete de mediu intern și extern în care funcționează sistemul și care trebuie definite explicit. Este aproape imposibil și nefeșabil să proiectezi și realizezi un sistem care să funcționeze în orice condiții, de exemplu în deșert, la pol sau în spațiu, într-o centrală nucleară sau într-un submarin. La fel de imposibil este să gândești un program pentru computer care să rezolve orice problemă matematică și să poți ține seama de toate combinațiile posibile de date de intrare. Dacă sistemul este utilizat în alte condiții de mediu decât cele avute în vedere la proiectarea și fabricarea sa, funcționarea lui poate fi grav perturbată.

Matematic, fiabilitatea se exprimă cu ajutorul **funcției de fiabilitate** (*reliability function*) $R(t)$ și reprezintă probabilitatea ca în intervalul $(0, t)$ sistemul să funcționeze fără să apară defecte [12]:

$$p(t) = P(t < T) = R(t), \quad (1.1.1)$$

unde:

$p(t)$ – probabilitatea de bună funcționare;

t – variabila timp;

T – limita specificată a duratei de funcționare, respectiv durata de funcționare până la defectare;

Atunci când ne referim la fiabilitatea unui sistem tehnic trebuie să ne referim la toate părțile componente ale acestuia care au fiabilități diferite. Componentele unui sistem tehnic pot însemna echipamente fizice, hardware, software, interfețe, conexiuni, operator uman, proceduri de utilizare și nu în ultimul rând, infrastructura suport. Un sistem trebuie să fie echilibrat din punct de vedere al fiabilității: toate componentele sale trebuie să aibă nivele de fiabilitate comparabile, altfel cea mai nefiabilă componentă determină fiabilitatea sistemului, ea este *veriga cea mai slabă* a lanțului.

Fiabilitatea unui sistem poate fi sporită folosind diferite tehnici speciale pe care le vom prezenta în capitolele următoare. Din studiile de specialitate se poate considera că în defectarea sistemelor de calcul, defectarea hardului reprezintă un procent de 19%, a softului 14% în timp ce factorul uman (proceduri incorecte sau incorect aplicate) generează 49% din defectări, actele de vandalism 1%, accidente de mediu 11%, suprasolicitarea 6%. În această carte ne vom ocupa de fiabilitatea hardware și software, de modelele care descriu defectarea acestora, metode de evidențiere a defectărilor, măsurare a parametrilor și îmbunătățirea indicatorilor de fiabilitate.

Toți factorii care influențează fiabilitatea sistemelor au caracter aleator; din acest motiv fundamentul matematic al teoriei fiabilității îl constituie teoria probabilităților și statistica matematică. Datele ce prin prelucrări matematice permit estimarea perioadei de bună funcționare se obțin fie prin încercări accelerate riguros proiectate, fie din funcționarea reală a sistemelor, cu condiția ca aceste date să fie corect colectate și suficiente.

1.2. Obiective ale fiabilității în ciclul de viață al sistemelor

Teoria fiabilității se construiește pe baza datelor referitoare la *defectarea* sistemului și componentelor acestuia. *Defectarea este procesul de pierdere a capacității sistemului (sau componente ale sistemului) de a-și realiza funcția proiectată.*

Fiabilitatea ca știință are ca obiect:

- aprecierea cantitativă a comportării sistemelor în timp, ținând seama de influența pe care o exercită asupra acestora factorii interni și externi;

- stabilirea metodelor de proiectare, dezvoltare, constructive, tehnologice și de exploatare pentru asigurarea, menținerea și creșterea fiabilității sistemelor și componentelor acestora;

- studiul defectelor și erorilor (al cauzelor, proceselor de apariție și dezvoltare), al metodelor de prevenire a apariției defectelor, de remediere a defectelor și corectare a erorilor;

- analiza fizică a defectelor;
- stabilirea modelelor și metodelor de calcul și prognoză a fiabilității pe baza încercărilor specifice și a urmării comportării în exploatare a sistemelor;
- stabilirea metodelor de selectare și prelucrare a datelor privind fiabilitatea;
- determinarea valorilor optime a indicatorilor de fiabilitate;
- stabilirea unor măsuri corective pentru reducerea riscurilor pe parcursul ciclului de viață și îmbunătățirea fiabilității.

Apectele privin managementul și procedurarea activităților utilizatorilor sistemului sunt tratate separat în științe noi de exemplu calitate, optimizare securitate și management inteligent.

Fiabilitatea este inseparabil legată de capacitatea sistemului de a fi repus în funcțiune după defectare. Probabilitatea ca starea de bună funcționare a sistemului să fie restabilită prin operații adecvate desfășurate într-un anumit interval de timp se numește *mentenabilitate*. Ansamblul tuturor activităților desfășurate pentru a menține sau restabili starea de bună funcționare se numește *mentenanță*. Există sisteme care în urma unor operații de mentenanță pot fi aduse într-o stare de funcționare apropiată de cea inițială, iar astfel de sisteme se numesc *sisteme cu restabilire* sau *reparabile* (numite și sisteme *cu reînnoire*) și sisteme care nu mai pot fi aduse în starea de bună funcționare prin nici un program de mentenanță. Astfel de sisteme se numesc *sisteme nereparabile*.

Fiabilitatea sistemelor trebuie avută în vedere în întreg ciclul de viață al sistemului. Ciclul de viață al oricărui sistem, inclusiv al sistemelor informatice, constă într-o înlănțuire de etape care se succed dar se și suprapun pe anumite perioade. În accepțiune internațională, etapele ciclului de viață ale unui sistem sunt [UP]:

1. *analiza necesității unui sistem*: identificarea cerințelor și așteptărilor beneficiarilor acestui sistem;
2. *concepere*: stabilirea funcțiilor sistemului;
3. *proiectare/dezvoltare*: realizarea proiectului pe baza cerințelor identificate ale beneficiarului referitoare la funcțiile sistemului, cu respectarea cerințelor legale și celorlalte reglementări legale din domeniu;
4. *realizare* - realizarea propriu-zisă pe baza proiectului elaborat;
5. *testarea* - utilizând metode adecvate, inclusiv funcționarea în condiții controlate/de laborator care simulează funcționarea reală;
6. *implementare, funcționare/operare* în condiții reale/nominale;
7. *verificare, testare, inclusiv validare*;
8. *întreținere și îmbunătățire, mentenanță* - corectarea erorilor care au condus la defectare, up-grade, up-date;
9. *retragere*.

1.3. Defectări. Tipuri și evoluție

Un sistem bine proiectat, corect realizat, minuțios verificat, judicios utilizat nu ar trebui să prezinte defectări în funcționare. Experiența a arătat totuși că și cele

mai bune sisteme în condițiile celei mai corecte exploatare nu exclud în întregime posibilitatea apariției unor defecte în funcționare.

Defectul se poate defini ca o anomalie în funcționarea corectă a unui sistem. Încetarea capacității unui sistem de a îndeplini funcția specificată (proiectată) poartă denumirea de *cădere* sau *defectare*. *Defectarea* poate să însemne că cel puțin una din performanțele sistemului a ieșit din limitele specificate. Performanțele sistemului sunt relevante față de o anumită aplicație, iar a ține seama de toate performanțele sistemului, chiar pentru un sistem simplu, este extrem de dificil.

Defectarea poate fi rezultatul unui proces continuu de variație a performanțelor sistemului sau al modificării bruște a valorii unei performanțe, starea de defectare aflându-se într-un raport de continuitate sau discontinuitate față de starea de bună funcționare a sistemului. Indiferent de modul de variație al performanțelor sistemului, defectarea este un proces aleator. Toate modelele matematice ale fiabilității sunt de tip *stochastic*, astfel încât previziunea comportării viitoare a unui sistem, bazată pe cunoașterea evoluției sale din trecut și a structurii sale, poate fi făcută numai pe baza teoriilor specifice probabilităților, cu un anumit nivel de încredere, cu un nivel acceptat de incertitudine.

Cauzele care pot determina defectarea pot fi datorate proiectării, fabricației și/sau utilizării sistemului. Defectarea este provocată prin depășirea unor stări limită, care se manifestă sub forma ruperii unei componente, apariției unui scurtcircuit sau erori în program etc., mecanismul defectării putând fi de natură fizică, chimică sau de altă natură. Există printre specialiști și opinia conform căreia nu orice mică defecțiune constituie o defectare, întrucât există defecțiuni care nu împiedică îndeplinirea funcțiilor de bază ale produsului.

Fiabilitatea software reprezintă probabilitatea ca *software-ul* să nu producă defectarea unui sistem care utilizează calculatorul într-un anumit interval de timp și în condiții specificate [16, 34]. Este legată de abilitatea softului de a rula cum și când este necesar în momentul integrării în sistem.

Defectarea softului nu este similară cu cea a hardului, este un eveniment cauzat de pierderea abilității de realizare a unei funcții soft în limitele specificate și se definește ca manifestarea unui defect în soft care poate împiedica realizarea performanței cerute, face să se obțină rezultate neașteptate în funcționarea acestuia, discrepanțe între valorile obținute și cele specificate sau corecte din punct de vedere teoretic. Defecțiunile softului apar în mod aleatoriu în operarea sistemului și pot surveni oricând în timpul ciclului de viață al sistemului. Atunci când apar sunt sistematice și au caracteristici similare.

Un defect soft reprezintă incapacitatea unui soft de a opera din cauza erorilor. Un defect soft rămâne latent până când este activat în anumite circumstanțe de funcționare și în mod normal devine inactiv când aceste circumstanțe nu mai există. Un defect soft poate cauza defectarea sistemului.

Eroarea soft este o eroare de obicei produsă de acțiunea greșită a unui programator în timpul codificării sau de interpretarea greșită de către programator a cerințelor specificației soft, traducerea incorectă (compilarea) în limbajele specifice sau omiteri ale cerințelor în specificația de proiectare.

În mod uzual se folosește termenul de eroare atât pentru o acțiune greșită cât și pentru manifestarea greșelii în program.

Referitor la **fiabilitatea software (a programelor)**, preocupările au fost direcționate spre elaborarea unor programe cât mai bune, care să nu conțină erori, excluzând posibilitățile ca un program foarte bun să se defecteze în tipul ciclului de viață, să apară erori noi în timpul utilizării unui anumit software [45, 80]. Pe parcursul acestei cărți pentru software vom folosi frecvent termenul soft.

În domeniul programelor *defectarea constă în punerea în evidență a unei erori latente conținute în program și care nu se datorează uzurii*. Datorită faptului că o anumită configurație de date, care pune în evidență eroarea, apare după un interval de timp aleator de testare sau utilizare, defectările software-ului pot fi tratate ca evenimente aleatoare, care se produc de-a lungul unui interval de timp. În cazul programelor defectarea reprezintă manifestarea unei erori prezente în program și care este detectată prin:

- mesaje de eroare de execuție;
- o durată de execuție nefinită a programului;
- obținerea unui rezultat clar eronat, în afara domeniului.

S-a demonstrat că în timp, chiar într-un program foarte bun, elaborat la firme mari se specialiști recunoscuți și minuțios verificate au apărut defecte. Prognoza defectării unui program este afectată de două surse de incertitudine:

- variabilitatea datelor de intrare și lipsa unei anumite legi de succesiune a acestora;
- variabilitatea programelor care pot implementa o aceeași funcție, respectiv variabilitatea raționamentelor pentru rezolvarea unei probleme.

Elementele esențiale în definiția *defectării software*, conform IEEE (ANSI) 982.2 din 1988 [34] sunt: *eroarea* (error), *neregula* (fault, bug), *defectul* (defect) și *defectarea* sau *căderea* (failure).

Monitorizarea funcționării unui sistem de calcul implică atât partea software cât și partea hardware, operatorul uman și mediul de lucru. Deseori, interacțiunile dintre aceste părți pot conduce la defectări chiar în absența unor defecte localizabile în vreuna dintre ele. De aceea, este important să se ia în considerare ansamblul om-mașină, format din hardware (HW), software (SW) și personalul utilizator, factorul uman (FU). Defectarea sistemului se poate produce din cauza unor malfuncționări ale software-ului chiar dacă acesta nu conține defecte dar este utilizat necorespunzător.

O variantă de clasificare a defectelor software este următoarea [1,16]:

a) *defecte ale specificațiilor*:

- date incomplete sau inconsistente;
- defecte de interfață între hard, soft și utilizator;

b) *defecte ale proiectului*:

- descriere incorectă;
- definiția incompletă a datelor;
- interpretarea greșită a cerințelor;
- control logic sau descriere logică incompletă;
- lipsă de comunicare;
- proceduri incomplete de identificare a posibilelor erori;

- aplicarea de standarde ieșite din uz.
- c) *defecte de programare, de codare*:
 - de logică;
 - computaționale;
 - de prelucrare;
 - de interfață;
 - de comentariu;
- d) *defecte inserate la compilare*: compilare incorectă, interfețe neadecvate;
- e) *defecte de documentare*: defecte în documentele care însoțesc produsul software.

f) *defecte introduse în timpul mentenanței*:

- corectare cu apariția unei alte erori

Din punct de vedere cantitativ se consideră două tipuri de erori în execuția unui program:

- *eroarea de discriminare*, prin care o anumită valoare de intrare este prelucrată pe o altă cale;

- *eroare de omisiune*, prin care o valoare de intrare este prelucrată în mod inadecvat pe una din căile existente, în program nefiind prevăzută o cale pentru prelucrarea corectă.

1.4. Fiabilitatea factorului uman

Fiabilitatea umană se poate defini ca *probabilitatea realizării cu succes de către om a unei activități sau sarcini în orice stadiu de operare a sistemului, într-o limită de timp specificată*.

Pornind de la premisa că studiul fiabilității se realizează utilizând procese stohastice în care decizia factorului uman este subiectivă, este absolut obligatoriu să se țină seama de factorul uman în validarea rezultatelor obținute. Teoria deciziilor reale este o ramură a psihologiei care studiază comportarea persoanelor reale în situații ideale. Nu interesează modul real de luare a deciziilor reale de către persoane reale, ci modul rațional în care ar acționa un subiect idealizat.

Prima condiție de raționalitate este *coerența strictă*, respectiv repartiția apriori considerată să acopere toate valorile adevărate ale parametrului, adică tuturor valorilor adevărate ale parametrului să li se asocieze o probabilitate apriori, astfel încât toate deciziile privind acest parametru să nu fie eronate.

A doua condiție este *condiția de simetrie*, respectiv pentru aceleași informații prelucrate de analiști diferiți să rezulte repartiții apriori identice, reducând la minimum subiectivitatea analistului.

A treia condiție se referă la modul de combinare a informațiilor apriori cu rezultatele experimentale, respectiv la faptul că *numai rezultatele experimentale să influențeze decizia analistului*.

Pentru activitatea de tip continuu, timpul mediu până la eroarea umană - *MTHE* (Mean Time to Human Error), poate fi estimat pe baza relației:

$$MTHE = \int_0^{\infty} \exp\left[-\int h(\tau)d\tau\right] dt \quad (1.4.1)$$

unde: $h(\tau)$ este rata erorii umane iar τ este timpul.

Rata erorii umane se poate determina experimental și poate fi constantă sau poate avea o distribuție exponențială, Weibull, normală, gama, Poisson etc.

Factorul uman intervine la toate fazele de dezvoltare a unui sistem fie că este mecanic, electric, tehnic, social sau software, în toate etapele ciclului de viață ale sistemului: la nivelul identificării cerințelor, conceperea și proiectarea lui, realizare și testare în exploatare precum și la culegerea datelor din exploatare sau încercări, în vederea studierii și îmbunătățirii fiabilității sistemului.

Concluziile specialiștilor spun că aproximativ 40% dintre defecțiuni sunt datorate direct erorilor umane, iar pentru echipamente complexe (CNE, aerospațial) cota ajunge la 70-80%. O parte din erorile umane sunt datorate concepției, fabricației sau instalării la locul de exploatare al sistemului. O serie din aceste erori pot fi corectate în diferite etape ale controlului de calitate, la verificările intermediare și finale, în timpul încercărilor de punere în funcțiune precum și în prima perioadă de funcționare a sistemelor.

În exploatare, cele mai multe și importante erori se datorează personalului.

Erorile umane pot fi grupate în două mari categorii: erori tehnice și psihologice. Nu ne vom referi la erorile introduse cu rea voință.

a) Erori tehnice pot fi în concepere și proiectare, exploatare, încercări sau mentenanță.

Erorile umane în faza de concepere/ proiectare pot fi:

- neînțelegeri ale cerințelor;
- comunicare incorectă între specialiștii implicați;
- descrieri incompetente sau incorecte etc.

Erorile umane în exploatare pot fi:

- manevre greșite în urma unor indicații greșit aplicate sau neaplicate, unor informații greșit interceptate sau unor fenomene rău înțelese;
- alarme neglijate sau greșit interpretate ca fiind o defectare a sistemului de alarmă;
- transmitere greșită de consemne între persoane;
- utilizarea de consemne incomplete sau perimate;

Erori umane în timpul încercărilor periodice

- încercări periodice neefectuate sau greșit efectuate, erori de etalonare și reglaj;
- utilizarea de proceduri de încercare incomplete sau perimate;
- neglijarea sau uitarea de a efectua anumite operații, care fac ca sistemul să se afle la finele încercării într-o stare diferită de cea prevăzută.

Erori umane în timpul operațiilor de mentenanță

- proceduri de întreținere nerespectate;
- utilizarea de proceduri incomplete sau perimate;
- neglijarea sau uitarea de a efectua anumite operații care fac ca la sfârșitul operației de mentenanță sistemul să nu fie în starea prevăzută.

b) Erori psihologice

Erorile psihologice ale operatorului uman se pot clasifica în: *erori de omisiune* (de memorie, de atenție) și *erori pasive* (de identificare, de interpretare, de operare).

Erori de omisiune pot proveni de la oboseală și de la lipsa de antrenament profesional. Memoria poate fi ajutată prin instrucțiuni de lucru scrise și afișate la locuri vizibile, indicarea de nivele limită, condiții critice. Atenția poate fi ajutată de diferite forme de afișare, inclusiv prin utilizarea culorilor de cod, a semnalelor auditive sau a oricăror alte semnale. Există tendința de a reduce dimensiunile și numărul de aparate de pe panouri și din camerele de comandă și de a utiliza ecrane concentratoare.

Erorile pasive pot fi:

a) *Erori de identificare* care depind atât de concepție cât și de utilizator sau operator. Testarea psihologică a operatorului unui sistem complex este absolut necesară. Identificarea poate fi ajutată prin divizarea voită a panourilor de comandă, astfel ca grupuri de aparate, lămpi de semnalizare, butoane care deserveșc o zonă sau asigură o anumită funcționalitate să poată fi grupate corespunzător. Schemele mnemonice serveșc și ele acest scop.

b) *Erori de interpretare* au loc prin înțelegerea greșită de către operator a unor informații afișate sau a unor inscripții (scale, denumiri etc.) și în consecință prin acționarea într-un mod impropriu. Culoarea diferită a unor inscripții, marcarea specială a unor descriptori alfa-numerici, iluminarea unor informații scrise ajută la evitarea unor astfel de erori.

c) *Erori de operare* sunt cele la care mișcarea cu rol de control (reglaj) este impropriu în raport cu efectul dorit cauzele sunt similare cu cele de mai sus, consecințele pot fi foarte grave.

În vederea diminuării efectului factorului uman în scăderea fiabilității se impun o serie de acțiuni, respectiv măsuri pentru proiectarea și construcția sistemelor care implică:

- efectuare a secvențelor operaționale în mod cât mai facil, uneori prin realizarea unor modele asociative mental care să câștige atenția operatorului;
- eliminarea necesității de a interpreta o informație afișată sau scrisă;
- mișcările și eforturile psihice să fie gândite cât mai ergonomic.

Pentru a preîntâmpina rezistența pasivă a personalului se impune:

- crearea unei atmosfere plăcute de lucru și de cooperare;
- stabilirea unui scop comun, recunoscut și acceptat;
- elaborarea de sarcini și responsabilități specifice;
- urmărirea modului în care progresează îndeplinirea scopului propus.

Comportamentele umane se pot ameliora prin obținerea de satisfacții morale și materiale prin analize de comportament și eficiență, prin selecția personalului (inclusiv prin chestionare de angajare, probe de lucru, teste psihologice sau alte metode).

Principalele mijloace disponibile pentru a minimiza frecvența erorilor umane sunt:

- luarea în considerare a operatorului uman încă din faza de concepție;
- punerea la punct a unor reguli de utilizare și întreținere care să reducă dificultățile de interpretare și de aplicare (livrarea sistemului împreună cu o instrucțiuni de instalare și utilizare)
- controlul calității în toate fazele (de proiectare, instalare, exploatare, scoatere din uz), existența instrucțiunilor de lucru la locul efectuării operației, afișarea regulilor de urmat;
- instruirea permanentă a personalului angajat;
- îmbunătățirea continuă a nivelului instruirii ;
- îmbunătățirea a mediului de lucru.

În realizarea unei analize a fiabilității umane (HRA-Human Reliability Analysis) se ține seama că omul operează asupra entităților calitative și realizează estimări posibile (comportament de tip fuzzy):

- integrarea tehnologiilor noi în sisteme complexe determină ca datele experimentale asupra fiabilității sistemului să fie insuficiente, caz în care se apelează la date previzionale de fiabilitate.

- operatorii sistemelor complexe sunt instruiți și educați la un nivel ridicat.
- mare parte din criteriile de decizie sunt calitative, cu scală diferită de la un caz la altul și nu pot fi comparate direct.

Există numeroase preocupări pentru un management inteligent și analiză a fiabilității umane din perspectiva implicării conșiente și responsabile a personalului în exploatarea sistemelor, limitarea acțiunilor care pot genera erori și defectări.

1.5. Evoluția în timp a defectelor

Prezența defectărilor în diferite etape din viața sistemelor este diferită în funcție de etapa din ciclul de viață, de tipul și complexitatea sistemului. La nivel internațional, în literatura de specialitate, pentru perioada de funcționare este unanim considerată pentru evoluția de tip a ratei de defecte a așa numitei curbe de forma *cadă de baie* - *bathroom curve* prezentată în figura 1.

Conform acestei reprezentări în ciclul de viață al unui sistem se pot considera trei perioade distincte.

Prima perioadă o constituie **perioada inițială** (de *mortalitate infantilă*, de *rodaj*, de *maturizare*) perioadă în care defectările au o frecvență ridicată. Elementele componente care se defectează în această perioadă sunt elementele cele mai slabe, cu vicii ascunse care se defectează chiar de la primele solicitări. După eliminarea acestor defecțiuni inițiale, precoce, defecțiunile se vor produce din ce în ce mai rar. Din acest motiv, anumite echipamente se livrează consumatorilor după o perioadă inițială de rodare, de testare, în care are loc "punerea la punct", perioadă în care echipamentele sunt puse în funcțiune în condițiile nominale specificate iar softurile se vând după ce varianta β a fost oferită liberă pe piață și a fost utilizată de mai mulți utilizatori, care au trimis observații referitor la buna funcționare. În această perioadă caracteristicile de fiabilitate ale unui sistem se îmbunătățesc. Pentru dispozitivele semiconductoare aceste defectări se elimină în timpul perioadei de ardere, prin proba numită "burn-in" realizată de producător înaintea vânzării acestora. Pentru produsele soft o parte din erorile din perioada de dezvoltare a codului sunt eliminate în timpul testării "variantei β " de către utilizatori interesați.

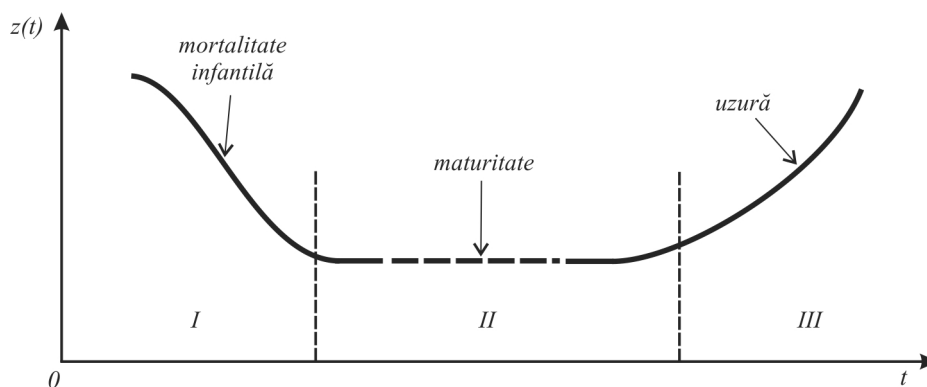


Figura 1.1. Variația tipică în timp a ratei de defectare, curba "cadă de baie"

Perioada a II-a, **perioada normală de funcționare** (*de maturitate*) reprezintă perioada principală de funcționare, cu durata cea mai lungă. Caracteristica generală a acestei perioade o constituie frecvența redusă a defectărilor și nivelul relativ constant al ratei de defectare. Defectări apar și în această perioadă, ele însă nu pot fi prevăzute. Nu trebuie făcută confuzie între aceste defecțiuni și eventualele înlocuiri periodice datorită uzurii care țin de programul de mentenanță al sistemului.

Defectele din perioada de maturitate au un caracter accidental, aleator, în general constant. Aceasta este perioada în care se efectuează studiile privind fiabilitatea. Înaintea începerii unui studiu de fiabilitate este necesar să se testeze dacă respectivul sistem se află în perioada normală de funcționare.

Perioada a III-a, **perioada finală** (*de uzură*, așa numita "bătrânețe") se caracterizează printr-o creștere bruscă a frecvenței defectărilor datorită uzurii accelerate a sistemului (degradare). În această perioadă caracteristicile de fiabilitate se înrăutățesc rapid. Pentru multe sisteme tehnice, inclusiv pentru soft, această perioadă nu se atinge în practică, ele uzându-se moral înainte de uzură și înlocuite cu altele cu caracteristici superioare.

Extinderea în timp a celor trei perioade variază în funcție de natura sistemului.

1.6. Tipuri de încercări pentru estimarea fiabilității

Estimarea parametrilor de fiabilitate ai sistemelor, oricât ar fi de complexe se face pe baza indicatorilor de fiabilitate ale tuturor elementelor componente ale sistemelor și conexiunilor dintre acestea,

Indicatorii de fiabilitate ai componentelor avute în vedere la proiectarea sistemului trebuie cunoscuți. În calculul fiabilității sistemului trebuie să se țină seama și de rolul și locul pe care îl ocupă în cadrul sistemului, de gradul de solicitare al componentei în cadrul sistemului, de criticitatea ei în funcționarea corectă a sistemului.

Indicatorii de fiabilitate ai componentelor se pot estima fie pe baza urmăririi în timp a funcționării și defectării componentelor respective în diferite sisteme, în aceleași condiții de solicitare și de mediu (fiabilitate operațională), fie prin încercări în laborator (fiabilitate experimentală sau empirică). Ambele metode sunt laborioase și scumpe, chiar dacă s-ar putea crede că urmărirea în timp a funcționării nu presupune costuri suplimentare. Înregistrarea corectă a datelor presupune completarea unor formulare bine gândite de către operatori instruiți și conștienți de faptul că date incomplete sau incorecte deformează rezultatul studiului.

O condiție esențială pentru estimarea fiabilității unui sistem o reprezintă *continuitatea variației* performanțelor sale. Este nevoie ca pentru fiecare situație în care performanța relevantă are o evoluție în trepte sau în salturi, să se găsească o variabilă măsurabilă, corelată direct cu performanța de interes și care să aibă o variație continuă.

Pentru orice mecanism de defectare, se poate găsi o ecuație a defectării, care, prin transformări convenabile, poate fi liniarizată, astfel încât ecuația performanței să poată fi scrisă în forma [8, 12]:

$$y(t) = A + \int_0^t B(t) dt, \quad (1.6.1)$$

unde: $B(t)$ = rata instantanee de variație, A = valoarea inițială a performanței.

Dacă A și $B(t)$ sunt perfect cunoscute, durata de funcționare până la defectare poate fi stabilită în mod determinist, egalând expresia (1.6.1) cu valoarea limită a performanței y .

Dar A și $B(t)$ nu sunt riguros determinate, ele fiind variabile aleatoare, previziunea defectării sistemului este probabilistă, astfel încât nu pot fi cunoscute exact și simultan valoarea inițială a unui parametru și modul (panta) de variație a acestora.

Probabilitatea reprezintă gradul de încredere *rațională* a realizării unui posibil eveniment; nu e certitudine. Faptul că funcția de fiabilitate a unui sistem în intervalul $t + \Delta t$ are valoarea R_0 afirmă că *probabilitatea* îndeplinirii cu succes a unei misiuni de durată Δt este egală cu R_0 .

Încercările de fiabilitate se pot face în scopul:

- determinării indicatorilor de fiabilitate,
- confirmării (validării) unor indicatori de fiabilitate menționați în documentul tehnic normativ al componentei/produsului/sistemului,
- stabilirea cauzelor care provoacă defecțiuni și mecanismele de defectare asociate.

Prin încercări de fiabilitate se determină indicatori sau funcții de fiabilitate empirice(e), notate în literatura de specialitate cu " \wedge " sau " $-$ " deasupra simbolului. În funcție de procedura de încercare, de durata încercării și numărul de componente defecte admise din tot lotul supus încercărilor, se pot efectua următoarele tipuri de încercări:

- *încercări complete*, când încercarea continuă până când toate elementele componente ale eșantionului s-au defectat,
- *încercări trunchiate*, când încercarea este întreruptă după o anumită durată prestabilită, înainte de defectarea tuturor elementelor eșantionului (censored type I tests),
- *încercări cenzurate*, când încercarea se întrerupe după ce s-au defectat un număr specificat de elemente (censored type II tests).

Toate aceste tipuri de încercări se pot combina între ele în vederea obținerii rezultatelor în timp util.

În funcție de nivelul solicitării în timpul încercărilor de fiabilitate, acestea pot fi:

- *accelerate*, când solicitările în condiții de laborator sunt superioare celor din funcționarea normală, în vederea scurtării timpului de încercare;
- *normale*, în cazul în care nivelul solicitării este același cu cel din funcționarea normală. De obicei, sunt încercările de fiabilitate operațională.

Metoda generală de încercare presupune supunerea unui eșantion de test cu un număr specificat de elemente, prelevate aleatoriu din lotul care urmează a fi caracterizat ("populația considerată"), la ciclul de încercare specificat și apoi însumarea pentru toate componentele supuse încercărilor a duratelor relevante de încercare și a defectărilor înregistrate, până când se poate lua decizia de întrerupere

a încercării, în cazul încercării trunchiate, sau până când este atins numărul de defectări prestabilit pentru încercările cenzurate [1, 2, 34, 44, 46].

Pe parcursul încercărilor este necesară efectuarea unor înregistrări sistematice, care să permită luarea deciziei corecte referitoare la încercări. Aceste înregistrări trebuie să cuprindă, cel puțin, următoarele informații:

- timpul (momentul începerii încercărilor, momentele de apariție a defectărilor etc.),
- identificarea elementelor înlocuite sau restabilite,
- detalii asupra condițiilor de solicitare și de mediu.

Principalul indicator de fiabilitate urmărit a fi estimat prin încercări este *media timpului de funcționare*, \bar{t}_m , notat și cu m și care poate fi corelat fie cu *MTBF*, fie cu *MTTF* sau chiar cu rata de defectare.

Un alt indicator estimat este *dispersia timpului mediu de bună funcționare*, D sau *abaterea pătratică medie* a timpului de funcționare, σ . Ceilalți indicatori de fiabilitate pot fi deduși pe baza relațiilor dintre indicatori, relații prezentate în capitolul 2 al prezentei cărți.

Încercările complete pot fi efectuate în condiții reale sau în condiții simulate [2, 30]. Încercări complete în condiții reale se fac de producător, numai când există cerințe exprese în contract. În cazul unei încercări complete, în condiții simulate, lotul de N elemente este supus încercărilor până la defectarea tuturor elementelor din lot. Acest tip de încercări se mai numește "încercări până la epuizare".

Pe parcursul încercării, pentru fiecare element defectat se înregistrează timpii de defectare, (t_i) . Durata cumulată de încercare este dată de relația:

$$T_{\Sigma} = \sum_{i=1}^N t_i. \quad (1.6.2)$$

Încercările complete pot avea durate foarte mari și să necesite cheltuieli semnificative, de aceea astfel de încercări se utilizează mai rar și numai pentru cazul, în care estimarea indicatorilor trebuie să se facă cu o precizie foarte mare, cum este cazul componentelor utilizate în sisteme de securitate (energetică nucleară, tehnică aerospațială etc.).

Pentru cazul uzual se folosesc încercări scurtate, respective cenzurate sau trunchiate care pot oferi rezultate corecte, în timp relativ scurt, cu costuri rezonabile.

Pentru încercările de fiabilitate a softului se înregistrează numărul de erori care se manifestă pe parcursul duratei de rulare sau într-un anumit ciclu de rulare dar și timpul dintre două defectări succesive. Scopul încercărilor de fiabilitate este estimarea numărului de erori existente în program la momentul inițial și predicția comportării în timp a programului. Pentru estimarea fiabilității software s-au dezvoltat modele speciale.

Capitolul 2. FUNCȚII SPECIFICE FIABILITĂȚII, MENTENANȚEI ȘI DISPONIBILITĂȚII

2.1. Funcții specifice fiabilității

Există o serie de funcții specifice fiabilității, interconectate între ele, care permit descrierea fiabilității sistemelor, numite și *indicatori de fiabilitate*.

Pentru sisteme tehnice în general, deci și pentru hardware, acești indicatori sunt: *funcția de fiabilitate și cea de nonfiabilitate, distribuția timpului de funcționare până la defectare, rata de defectare, media timpului de funcționare până la defectare, dispersia și abaterea pătratică medie, cuantila timpului de funcționare*.

Pentru software, aceste funcții se adaptează și completează cu *rata de manifestare a erorilor sau densitatea de defectare, timpul de execuție, numărul de erori remanente în program după o durată de testare, numărul de intrări/ieșiri pe modul, indicatori Halstead referitori la lungimea volumul și dificultatea programelor, indicatori de complexitate*.

În continuare vor fi prezentate funcțiile care descriu fiabilitatea sistemelor tehnice și interdependența dintre ele. Funcțiile care descriu fiabilitatea sistemului software vor fi prezentate în detaliu în &2.2.

2.1.1 **Funcția de fiabilitate, $R(t)$** (Reliability), reprezintă probabilitatea ca un sistem (sau o componentă a sistemului) să-și îndeplinească corect funcțiile prevăzute pe o perioadă de timp specificată, în condiții de utilizare prestabilite [2,18, 31, 34]:

$$p(t) = P(t < T) = R(t), \quad (2.1.1)$$

unde:

$p(t)$ – probabilitatea de bună funcționare;

t – variabila timp;

T – limita specificată a duratei de funcționare, respectiv durata de funcționare până la defectare;

$R(t)$ – reprezintă probabilitatea ca în intervalul $(0, t)$ sistemul să funcționeze fără să apară defecte.

Funcția de fiabilitate $R(t)$ are următoarele proprietăți:

- $R(0) = 1$ deoarece $\{T > 0\}$ este un eveniment sigur al experienței;

- $R(t)$ este o funcție descrescătoare;

- $R(t) \rightarrow 0$ pentru $t \rightarrow \infty$, evenimentul $\{T > \infty\}$ fiind imposibil.

Conform teoriei probabilităților se consideră că probabilitatea de bună funcționare este egală cu 1 la momentul începerii funcționării, $t = 0$ și scade în timp, astfel încât după un interval de timp suficient de mare ($t \rightarrow \infty$) ajunge asimptotic la 0.

Reprezentarea generală a unei astfel de funcții este redată în figura 2.1. În capitolul trei al acestei cărți vom prezenta mai multe tipuri de funcții matematice concrete, care descriu fiabilitate sistemelor.

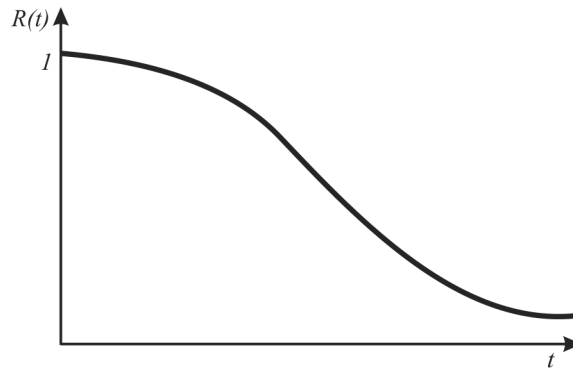


Figura 2.1. Funcția de fiabilitate

2.1.2. **Probabilitatea de defectare, $F(t)$** (unreliability) se definește cu relația:

$$F(t) = P(t \geq T). \quad (2.1.2)$$

$F(t)$ reprezintă probabilitatea de defectare a sistemului în intervalul $(0, t)$.

$R(t)$ și $F(t)$ sunt funcții complementare, funcția $F(t)$ numindu-se și *funcția de nonfiabilitate*. Matematic aceasta se scrie:

$$R(t) + F(t) = 1. \quad (2.1.3)$$

Aceste funcții se exclud reciproc, un sistem este la un moment dat, fie funcțional fie defect. Ele se pot ușor substitui, uneori este avantajos să discutăm despre probabilitatea de defectare a unui sistem, prin încercările de fiabilitate se obțin în mod uzual date despre defectele care se manifestă.

2.1-3. **Densitatea de probabilitate a timpului de funcționare fără defectare $f(t)$** (probability density function), exprimă frecvența relativă a defectărilor într-un interval de timp dat și se mai numește **lege de repartiție** sau de distribuție statistică a timpului de funcționare până la defectare. Se definește cu relația:

$$f(t) = \lim_{\Delta t \rightarrow 0} \frac{F(t + \Delta t) - F(t)}{\Delta t} = \frac{dF(t)}{dt} = -\frac{dR(t)}{dt}. \quad (2.1.4)$$

2.1.4. **Rata de defectare, $z(t)$** sau $\lambda(t)$, descrie probabilitatea de defectare în jurul unui moment dat al unui element component al unui sistem aflat în stare de bună funcționare până la acel moment.

Conform definiției probabilităților condiționate se poate scrie:

$$z(t) = \lim_{\Delta t \rightarrow 0} \frac{F(t + \Delta t) - F(t)}{R(t) \cdot \Delta t} = \frac{f(t)}{R(t)} = -\frac{1}{R(t)} \frac{dR}{dt}. \quad (2.1.5)$$

Această relație este valabilă indiferent de modul de variație în timp a ratei de defectare.

Unitatea de măsură pentru rata de defectare este inversul unității de măsură a timpului, h^{-1} . Pentru componente electronice se folosește unitatea de măsură *FIT* (Failure In Time). 1 *FIT* reprezintă un defect care apare după funcționarea unei componente timp de 10^9 h sau după funcționarea unui lot de 100 componente timp de 10^7 h, sau a unui lot de 1000 componente timp de 10^6 h sau o altă combinație între numărul de componente și timp care ajunge la 10^9 [37].

Relația dintre rata de defectare și funcția de fiabilitate rezultă din relația 1.1.5 scrisă sub forma:

$$\int_0^t z dt = - \int_1^R \frac{dR(t)}{R(t)} = -\ln R(t). \quad (2.1.5')$$

Pentru $t=0 \Rightarrow R=1$.

Pentru cazul general se poate scrie:

$$R(t) = \exp\left(-\int_0^t z(t)dt\right). \quad (2.1.6)$$

Rata de defectare permite o clasificare a sistemelor după tipul de uzură. Un sistem fără uzură are rata de defectare constantă (*RDC*), un sistem cu uzură pozitivă are rata de defectare pozitivă (*RDP*), iar un sistem cu uzură negativă are rata de defectare negativă (*RDN*). Funcția de fiabilitate are proprietăți specifice fiecărui tip de uzură.

2.1.5. Un alt indicator de fiabilitate este **media timpului de funcționare până la defectare, m** , care se definește cu ajutorul relației:

$$m = \int_0^{\infty} t \cdot f(t) dt = \int_0^{\infty} R(t) dt = \int_0^{\infty} \exp\left[-\int_0^t z(\tau) d\tau\right] dt, t \in (0, \infty). \quad (2.1.7)$$

În cazul sistemelor reparabile acest indicator devine timpul mediu între defectări, *MTBF* (**M**edian **T**ime **B**etween **F**ailures), iar în cazul sistemelor nereparabile devine *MTTF* (**M**edian **T**ime **T**o **F**ailures).

2.1.6. **Deviația standard (σ) și dispersia (D)** timpului de funcționare se definesc cu ajutorul relațiilor:

$$D = \int_0^{\infty} (t - m)^2 f(t) dt, \quad (2.1.8)$$

$$\sigma = \sqrt{D}. \quad (2.1.9)$$

Aceste două mărimi indică gradul de împrăștiere sau de uniformitate a performanțelor individuale ale unor sisteme de același tip, din punct de vedere al fiabilității.

Dacă procesul de fabricare al elementelor componente ale unui sistem este bine controlat, valorile lui D și σ vor fi mici.

2.1.7. **Cuantila timpului de funcționare, t_α** , reprezintă timpul în care proporția de elemente defectate dintr-un eșantion nu depășește o valoare prestabilită α . Este definit ca rădăcină a ecuației:

$$F(t_\alpha) = \alpha. \quad (2.1.10)$$

t_α nu depinde de timp și poate fi interpretat ca timpul garantat de producător până la atingerea unui anumit procent de defecte, un anumit nivel de încredere garantat de producător.

2.1.8. **Interdependența funcțiilor specifice fiabilității**

Așa cum se poate observa, între funcțiile de fiabilitate există relații de interdependență astfel încât pentru un anumit sistem este suficient să se estimeze anumite funcții, ceilelalte pot fi determinate prin calcul.

Relația dintre funcțiile $R(t)$, $F(t)$, $f(t)$ și $z(t)$ este prezentată sintetic în figura 2.2 și în tabelul 2.1, (fără a ține seama de o anumită formă a funcției $f(t)$):

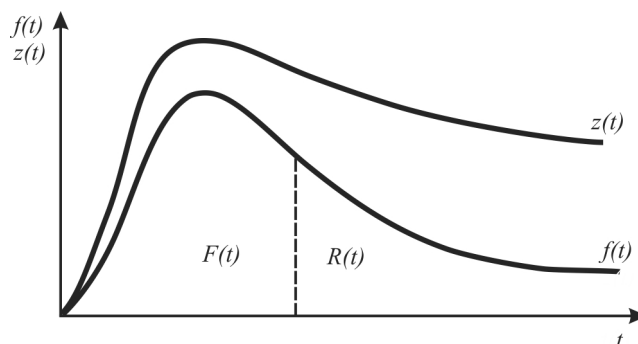


Figura 2.2. Relațiile între funcțiile specifice fiabilității [1].

Tabelul 2.1

	$F(t)$	$R(t)$	$f(t)$	$z(t)$
$F(t)$	1	$1 - R(t)$	$\int_0^t f(\tau) d\tau$	$1 - \exp\left[-\int_0^t z(\tau) d\tau\right]$
$R(t)$	$1 - F(t)$	1	$-\int_t^\infty f(\tau) d\tau$	$\exp\left[-\int_0^t z(\tau) d\tau\right]$
$f(t)$	$\frac{dF(t)}{dt}$	$-\frac{dR(t)}{dt}$	1	$z(t) \cdot R(t)$
$z(t)$	$\frac{1}{1 - F(t)} \frac{dF(t)}{dt}$	$-\frac{1}{R(t)} \frac{dR(t)}{dt}$	$\frac{f(t)}{R(t)} = -\frac{f(t)}{\int_t^\infty f(\tau) d\tau}$	1

Dacă se consideră $z(t) = \lambda = \text{constant}$, relațiile între funcțiile specifice de fiabilitate (prezentate în tabelul 2.1) se simplifică astfel:

$$R(t) = \exp(-\lambda \cdot t), \quad (2.1.11)$$

$$f(t) = \lambda \cdot \exp(-\lambda \cdot t), \quad (2.1.12)$$

$$m = \int_0^\infty \exp(-\lambda \cdot t) dt = \frac{1}{\lambda}. \quad (2.1.13)$$

Pentru a caracteriza un sistem care are impusă o anumită durată de viață sau durată a misiunii se preferă estimarea funcției de fiabilitate sau a funcției de repartiție a timpului de funcționare până la defectare. Pentru alte sisteme este preferabil să se estimeze media timpului de funcționare sau rata de defectare, ceilalți indicatori urmând a fi calculați în funcție de necesități.

O sinteză a indicatorilor generali de fiabilitate, simboluri, definiții matematice unități de măsură și interdependența lor și pentru ei este prezentată în tabelul 2.2 [34].

Nr. crt.	Indicator	Simbol pentru:			Definiție	UM
		Valoare teoretică	Valoarea estimată			
			Punc-tual	Interval de încredere		
0	1	2	3	4	5	6
1	Funcția de fiabilitate	R(t)	$\hat{R}(t)$	R_{inf} R_{sup}	Probabilitatea ca un sistem (sau o componentă a sistemului) să-și îndeplinească corect funcțiile prevăzute pe o perioadă de timp dată, în condiții de utilizare specificată, respectiv : $p(t) = P(t < T) = R(t)$	-
2	Funcția de repartiție a timpului de funcționare	F(t)	$\hat{F}(t)$	F_{inf} F_{sup}	Probabilitatea ca sistemul să se defecteze în intervalul (0, t) : $F(t) = P(t \geq T)$	-
3	Densitatea de probabilitate a timpului de funcționare	f(t)	$\hat{f}(t)$	f_{inf} f_{sup}	Frecvența relativă a defectărilor într-un interval de timp dat și se mai numește lege de repartiție a timpului de funcționare până la defectare. $\lim_{\Delta t \rightarrow 0} \frac{F(t + \Delta t) - F(t)}{\Delta t} = \frac{dF(t)}{dt} = - \frac{dR(t)}{dt}$	h ⁻¹
4	Rata (intensitatea) de defectare	$z(t)$ $\lambda(t)$	$\hat{z}(t)$	z_{inf} z_{sup}	Describe pericolul de defectare în jurul unui moment dat al unui element component al unui sistem aflat în stare de bună funcționare până la acel moment. $\lim_{\Delta t \rightarrow 0} \frac{F(t + \Delta t) - F(t)}{R(t) \cdot \Delta t} = \frac{f(t)}{R(t)} = - \frac{1}{R(t)} \frac{dR}{dt}$	h ⁻¹ sau 1FIT
5	Media timpului de funcționare: - MTTF - MTBF	m	\hat{m}	m_{inf} m_{sup}	Valoarea medie a timpului de funcționare (momentul centrat de ordinul 1): $m = \int_0^{\infty} t \cdot f(t) dt = \int_0^{\infty} R(t) dt = \int_0^{\infty} \exp[-\int_0^t z(\tau) dt]$	h
6	Dispersia timpului de funcționare	D	\hat{D}	D_{inf} D_{sup}	Momentul centrat de ordinul 2 al timpului de funcționare: $D = \int_0^{\infty} (t - m)^2 f(t) dt$	h ²
7	Abaterea medie pătratică a timpului de funcționare	σ	$\hat{\sigma}$	σ_{inf} σ_{sup}	$\sigma = \sqrt{D}$	h
8	Cuantila timpului de funcționare	t_{α}	\hat{t}_{α}	t Finf t Fsup	Timpul în care un produs funcționează cu o anumită probabilitate $1 - \alpha$, sau după care proporția de componente defecte dintr-un lot nu depășește valoarea α : $P(T \leq t_{\alpha}) = \alpha$	h

A se observa că o valoare mică a ratei de defectare este caracteristică unui sistem fiabil; cu cât rata de defectare are valori mai mici cu atât sistemul este mai fiabil. Pentru componente electronice o rată de defectare de ordinul 10^{-5} h^{-1} nu recomandă componenta respectivă pentru a fi utilizată în sisteme pentru care sunt impuse cerințe de securitate. La această oră, componentele electronice profesionale au rate de defectare apropiate de 10^{-9} h^{-1} .

2.2. Indicatori pentru fiabilitatea software

O parte din indicatorii referitori la fiabilitatea software sunt similari cu cei pentru orice sistem tehnic cum sunt funcția de fiabilitate, de nonfiabilitate sau funcția densitate probabilitate a timpului de funcționare fără defecte iar rata de defectare este înlocuită de rata de manifestare a erorilor. Există însă și o serie de indicatori specifici, indirecti ce vor fi prezentați mai în continuare.

2.2.1 Funcția de fiabilitate, $R(t)$.

Reprezintă probabilitatea ca nici o eroare să nu fie activată într-un interval de lucru precizat. În notația $R(t, t + \Delta t)$, unde Δt reprezintă intervalul de lucru, iar t momentul de inițializare al intervalului, care coincide de obicei cu sfârșitul testării. Funcția de nonfiabilitate este $F(t)$, cu aleeași atribute ca și la &2.1.2.

Argumentul timp din expresia funcției de fiabilitate poate fi uneori exprimat în unități calendaristice, dar este de preferat să se considere timpul de lucru al unității centrale de procesare (CPU time - Central Processing Unit time), astfel, atât t cât și Δt semnifică durate de execuție. Uneori argumentul funcției de fiabilitate (t și Δt) se exprimă în număr de rulări. Pentru $\Delta t = 1$, funcția de fiabilitate reprezintă probabilitatea unei rulări încununată de succes. De regulă, aceasta se evaluează pe baza a n rulări anterioare în care nu s-au detectat erori și poartă numele de **conformitate**.

2.2.2 Rata de manifestare a erorilor sau densitatea de defectare, $z(t)$.

Reprezintă probabilitatea manifestării unei erori în intervalul Δt raportată la mărimea Δt a acestui interval.

$$z(t) = \frac{F(t + \Delta t) - F(t)}{\Delta t \cdot R(t)}. \quad (2.2.1)$$

Rata de manifestare a erorilor, $z(t)$, se referă la o defectare oarecare, indiferent de numărul ei de ordine. Unitatea de măsură pentru $z(t)$ este inversa unității de măsură a timpului de execuție.

Există însă și *indicatori specifici* fiabilității software care nu-și găsesc un corespondent în teoria generală a fiabilității.

2.2.3 Cuanțila timpului de execuție, t_α

Reprezintă durata unei execuții în care funcția de fiabilitate are cel puțin valoarea α după un interval de testare t . Se mai numește *inversul funcției de fiabilitate*, adică durata de execuție în care probabilitatea activării unei erori este limitată la o valoare prescrisă egală cu $1 - \alpha$.

2.2.4 **Numărul de erori remanente, $N(t)$** , rămase în program după o durată de testare t .

Dacă pentru t se consideră valoarea 0, indicatorul va reprezenta numărul total de erori $N = N(0)$ rezidente în program. Raportându-se acest număr la numărul mediu de linii de cod (LOC), rezultă *densitatea erorilor* exprimată în $KLOC^{-1}$ (număr mediu de erori la o mie de linii de cod). Pentru codul sursă se folosește $KSLOC^{-1}$ (număr mediu de erori la o mie de linii de cod sursă).

2.2.5 Durata medie de funcționare până la defectare ($m=MTTF$ – Median Time To Failure)

Timpul mediu până la defectare se modifică după fiecare intervenție exterioară, astfel încât el depinde de timpul de testare, recomandându-se notația $m(t)$. Spre deosebire de o largă clasă de sisteme pentru care durata medie până la defectare coincide cu durata medie între defectări succesive ($MTTF = MTBF$), în cazul sistemelor software o asemenea egalitate este exclusă.

2.2.6. Indicatori indirecți, specifici fiabilității previzionale a programelor

Majoritatea indicatorilor enumerați mai sus necesită ca, înainte ca ei să poată fi evaluați, sistemul software să fie complet proiectat și implementat. Pentru a evalua nivelul de fiabilitate încă din faza de concepere, de proiectare, când se stabilesc cerințele, specificațiile, se face proiectarea simbolică, fiabilitatea se calculează în funcție o serie de indicatori specifici care se numesc *indicatori indirecți* sau chiar numai *metrice software* (*software metrics*). Acești indicatori permit previzionarea nivelului fiabilității software din faza de proiectare [16]:

a) Numărul de intrări/ieșiri pe modul, Input/Output Number.

Describe arhitectura programului în faza de proiectare. Simplist, un modul ar trebui să aibă câte o intrare și o ieșire pentru fiecare funcție principală. Pentru a trata erorile pot fi necesare însă puncte adiționale de ieșire.

Utilizând acest indicator se identifică modulele care măresc timpul de execuție prin apelări frecvente ale altor module. Minimizarea numărului de intrări/ieșiri este o caracteristică a programării structurate și oferă posibilitatea de creștere a fiabilității programului din faza de proiectare.

b) Indicatori Halstead.

Acești indicatori au fost propuși de Maurice H. Halstead în 1977 și permit evaluarea dificultății unui anumit program, pornind de la următoarele mărimi:

- n_1 – numărul de operatori distincți din program;
- n_2 – numărul operanzilor distincți din program;
- N_1 – numărul total de intervenții în program ale operatorilor;
- N_2 – numărul total de intervenții ale operanzilor în program;
- E – efortul făcut pentru dezvoltarea programului;
- B – numărul de nereguli (bugs, faults) din program

Cu ajutorul mărimilor de mai sus se pot calcula următorii indicatori Halstead:

1. Lungimea vocabularul programului:

$$n = n_1 + n_2; \quad (2.2.2)$$

2. Lungimea observată a programului:

$$N = N_1 + N_2; \quad (2.2.3)$$

3. Lungimea estimată a programului:

$$\hat{N} = n_1 \log_2 n_1 + n_2 \log_2 n_2; \quad (2.2.4)$$

4. *Volumul programului:*

$$V = N \cdot \log_2 n; \quad (2.2.5)$$

5. *Dificultatea programului:*

$$D = \frac{n_1}{2} \cdot \frac{N_2}{n_2}; \quad (2.2.6)$$

6. *Nivelul programului:*

$$NP = \frac{1}{D}; \quad (2.2.7)$$

7. *Efortul mental necesar implementării codului:*

$$E = \frac{V}{NP} = V \cdot D; \quad (2.2.8)$$

8. *Numărul de nereguli (Delivered Bugs) din program.* Acesta se estimează utilizând următoarea formulă empirică stabilită de Halstead:

$$B = \frac{V}{3000} \approx \frac{E^{2/3}}{3000}. \quad (2.2.9)$$

Se poate observa că prin micșorarea volumului programului, respectiv prin scăderea efortului mental necesar implementării codului (a lui E), se reduce numărul de erori din program, deci fiabilitatea programului crește.

9. *Timpul de implementare al unui program:*

$$t = \frac{E}{S}, \quad (2.2.10)$$

unde $S=18$ numărul de operații elementare de comparații/secundă (determinat de Halstead).

Ansamblul indicatorilor Halstead prezentați permite evaluarea dificultății programelor care nu trebuie să depășească un anumit prag pentru ca fiabilitatea să nu fie periclitată. Acest prag este specific limbajului de programare folosit.

c) **Indicatori de complexitate a sistemului software.**

Există patru indicatori ai complexității bazați pe teoria grafurilor [16]:

1) **Complexitatea statică, C_s**

Acest indicator descrie programul ca pe o rețea de module (subprograme), interconectate care pot fi executate într-o anumită succesiune, comanda de execuție fiind stabilită în program. Pentru calculul complexității se apelează la teoria grafurilor. Fiecare modul al programului reprezintă un nod al grafului iar fiecare arc reprezintă o apelare a modulului și o revenire între module.

Dacă E este numărul de arce și N este numărul de noduri ale grafului, complexitatea statică este dată de relația:

$$C_s = E - N + 1. \quad (2.2.11)$$

2) **Complexitatea dinamică C_d**

Acest indicator ține seama de faptul că întreruperile datorate apelărilor și revenirilor pot modifica, în timp, numărul de arce din graf. Ca urmare,

complexitatea dinamică se calculează folosind formula complexității statice la diferite momente de timp:

$$C_d = C_s(t) \quad (2.2.12)$$

O complexitate dinamică medie pe interval pune în evidență frecvențele diferite de execuție ale modulelor și întreruperilor în timpul execuției lor.

3) Complexitatea ciclomatică C_c

Se mai numește și complexitate condițională și a fost introdusă de Thomas J. McCabe, Sr. în 1976.

Acest indicator poate fi utilizat pentru estimarea fiabilității programului la nivel de cod sursă. Programul este reprezentat printr-un graf care are ca noduri instrucțiunile (N) și ca arce (E) trecerile controlului de la o instrucțiune la alta. Un nod din care pornesc mai multe arce se numește nod bifurcat (*Splitting Node*). Numărul ieșirilor din program este notat cu P .

Dacă SN este numărul nodurilor simplu bifurcate (în cazul nodurilor multiplu bifurcate se consideră de că nodul intervine de mai multe ori), iar RG este numărul regiunilor (regiunea este un domeniu mărginit de arce, care nu se intersectează), complexitatea ciclomatică poate fi evaluată în 3 moduri (evaluări echivalente):

$$C_c = E - N + P \quad (2.2.13)$$

Pentru programe care au o singură ieșire, cum este cazul celor mai multe programe și subprograme $P = 1$ iar $C_c = E - N + 1$ ca în cazul complexității statice;

$$C_c = RG \text{ (numărul de regiuni);} \quad (2.2.13')$$

$$C_c = SN + 1 \text{ (numărul de noduri bifurcate + 1).} \quad (2.2.13'')$$

O valoare rezonabilă pentru C_c este considerată 100.

Pentru programele care au mai multe ieșiri:

$$C_c = D - P + 2 \quad (2.2.14)$$

4) Complexitatea statică generalizată, C_g

Acest indicator consideră, alături de modulele interconectate - N , resursele - K (de memorie, de timp) alocate atunci când controlul rulării programului trece de la un modul la altul.

Fie matricea resurselor $R(K, E)$ care în teoria grafurilor se numește matricea de adiacență, ce are elementele r_{ik} egale cu 1 dacă resursa k este necesară pe arcul i și 0 – în caz contrar, unde: $k = 1, 2, \dots, K$ și $i = 1, 2, \dots, E$.

Complexitatea statică generalizată este dată de relația:

$$C_g = \sum_{i=1}^E c_i + \sum_{i=1}^E \sum_{k=1}^K d_k \cdot r_{ik}, \quad (2.2.15)$$

unde: c_i este complexitatea legată de apelarea și revenirea pe arcul $e_i, i = 1, 2, \dots, E$;

d_k - o măsură a complexității asociate cu alocarea resursei k (de exemplu complexitatea asociată procedurii folosite pentru a avea acces exclusiv la date comune, necesară pe arcul i);

r_{ik} - elementele matricei resurselor.

Cei patru indicatori de complexitate pot fi calculați pentru grafuri conectate, adică acele grafuri în care orice nod este accesibil din oricare alt nod. Acest lucru se poate realiza, adăugând un arc suplimentar între modulul de intrare și modulul de ieșire.

Fiabilitatea sistemelor software se poate îmbunătăți prin reducerea complexității dinamice. Aceasta se poate obține prin minimizarea frecvenței întreruperilor și reluărilor în timpul execuției programului.

Evaluarea complexității poate fi realizată în etapele incipiente ale proiectării programului. Ea permite o înțelegere adecvată a nodului, esențială în vederea unei mentenanțe corecte și eficiente.

Multe defecte sunt introduse în sistem în faza operațională, prin intervenții eronate, cu atât mai probabil, cu cât sistemul este mai complex.

Indicatorii complexității oferă o măsură a dificultăților în înțelegerea, testarea și mentenanța software-ului: cu cât valorile acestora sunt mai mici, cu atât fiabilitatea poate fi mai mare. Păstrând de la început un control asupra complexității, se asigură corectitudinea intervențiilor și, de aici, fiabilitatea operațională.

2.3. Metrici ce descriu fiabilitatea software

Metricii soft sunt caracteristici unui sistem soft ce pot fi măsurate și cuantificate ce permit determinarea cantitativă a valorii unui anumite performanțe a softului. Ei pot fi obținuți prin măsurare directă sau prin deducție și pot fi cuantificați.

În mod uzual pentru a măsura performanța și a evalua fiabilitatea sistemului soft sunt folosiți următorii metrici:

- *Densitatea de defecte* oferă reprezentă numărul de defecte conținute pe o mie de linii ale codului sursă (KSLOC Kilo Source Lines of Code)

- *Acoperirea codului* este un indicator al rigurozității testării softului, oferă o măsură a gradului în care codul sursă al unui program soft a fost testat sistematic. Indică procentul de defecte detectate în timpul testării execuției codului;

- *Rata de înlăturare a defectelor* reprezintă numărul de defecte detectate și corectate în soft într-o perioadă de timp determinată sau pe durata executării softului; permite stabilirea tendințelor de îmbunătățire ale softului.

- *Defectele reziduale din soft* reprezintă numărul estimat de defecte latente care au rămas în soft după testarea și înlăturarea defectelor sesizate.

- *Timpul pentru lansarea softului* reprezintă timpul estimat pentru graficul de lansare. Este bazat pe criteriile stabilite pentru nivelul de acceptare al defectelor latente rămase în soft.

- *Complexitatea softului* oferă o măsură a gradului de dificultate pentru proiectarea și implementarea unei funcții soft sau sistem soft; alte măsuri bazate pe conceptul de complexitate sunt complexitatea soft-ului, complexitatea funcțională, complexitatea operațională; metricii legați de complexitate sunt utilizați ca date de intrare pentru evaluarea fiabilității și modele de previziune.

Standardele de specialitate noi recomandă și alți metrici pentru a fi utilizați

În funcție de modul de colectare, în timpul ciclului de viață al softului, metricii soft pot fi grupați în trei categorii principale: *metricii despre defecte*, *metricii despre produsul soft*, și *metricii despre procesul de realizare a softului*

i) Metricii despre defecte colectează datele de raportare a problemei soft pentru a măsura impactul defectelor și eficiența procesului de raportare a apariției defectelor. Se referă la:

a) *Probleme de raportare a datelor*

- Data și momentul la care a fost detectat defectul
- Descrierea defectului detectat

- Defectul detectat în zona programului
- Persoana care a detectat defectul
- Simptomul defectului și statutul
- Severitatea și prioritatea

Datele colectate din proiectele soft vor fi folosite pentru raportarea problemelor de identificare.

b) Acțiuni corective

- Data corectării defectului
- Persoana care a corectat defectul
- Acțiunile de mentenanță efectuate
- Descrierea modificării
- Identificarea modulelor modificate
- Timpul necesar pentru corectarea defectului
- Data verificării corectării defectului
- Persoana care a verificat corecția

c) Defecte cumulate detectate și corectate utilizate pentru cunoașterea defectelor care necesită acțiuni corective și urmărirea efectivității acțiunilor de corectare

e) Rata de detectare a defectelor

f) Rata de corectare a defectelor pentru a stabili prioritățile pentru acțiunile de mentenanță

g) Defecte pe locație pentru identificare a ariei specifice a codului este se manifestă mai multe eroare.

h) Criticitatea defectelor, număr și procent de defecte severe

i) Complexitatea structurală și funcțională pe locație.

ii) Metricii despre proces colectează informațiile despre procesul de realizare a softului și condițiile funcționale la momentul detectării și înlăturării defectului. Se referă la:

a) Număr și procent de module care realizează mai mult decât o funcție: Este un indicator al coeziunii folosit pentru calculul complexității

b) Număr și procent de module care au o structură foarte complexă se urmărește reducerea lor prin reproiectare pentru a reduce complexitatea

e) Număr și procent de module care au exact o intrare și o ieșire: Indică un proiectului coeziv ce poate fi utilizat ca bază pentru dezvoltarea softului;

f) Număr și procent de module care sunt documentate în concordanță cu standardele utilizat pentru a determina dacă codul conține toate cerințele.

g) Număr și procent de defecte care sunt găsite în codul refolosit este un indicator al non-fiabilității codului reutilizat.

iii) Metricii despre produs colectează informațiile despre produsul soft clasificând după mărime, funcționalitate, complexitate, locația utilizatorului și alte caracteristici pentru a facilita utilizarea datelor colectate experimental ca date de intrare în scopul dezvoltării de noi produse:

Defecte introduse în ciclul de viață indică când și în ce stadiu au fost introduse defectele pentru a adopta măsurile adecvate

Defecte detectate în ciclul de viață indică când și în ce stadiu au fost detectate defectele și justifică întârzierile acțiunilor corective pentru eliminarea defectelor.

Timpul total consumat în analiză indică timpul necesar de analiză pentru identificarea problemelor și izolarea lor pentru acțiunile corective precum și resursele consumate

Timpul total consumat în proiectare inclusiv resursele consumate

Timpul total consumat în codificare inclusiv resursele consumate.

Timpul total consumat în testarea unității inclusiv resursele consumate

Timpul total consumat în testarea sistemului Indică timpul consumat de testarea sistemului și resursele consumate.

Timpul total de mentenanță Indică inclusiv resursele consumate

Timpul mediu de mentenanță - administrare timpul înainte și după corectarea defectului, inclusiv timpul petrecut cu asigurarea personalului de mentenanță, scoaterea în evidență a corecțiilor într-o versiune nouă.

Timpul mediu de acțiuni corective inclusiv resursele consumate care costul activităților de mentenanță.

Motivul acțiunii corective pentru a descoperi sursa defectelor. Motive tipice sunt:

- acțiuni de mentenanță trecute
- cerințe noi
- schimbări de cerințe
- cerințe interpretate greșit
- cerințe lipsă
- cerințe ambigue
- schimbări în mediul soft
- schimbări în mediul hard
- erori de cod/logice
- erori de performanță.

Costul acțiunilor corective inclusiv pentru izolarea defectelor, rezolvarea problemei și pentru acțiuni de mentenanță propriu-zise.

Procentul de funcții testate și verificate Indică acoperirea testelor, eficiența testelor și integralitatea.

Procentul de căi independente testate și verificate Indică acoperirea testelor structurale și integralitatea

Procentul de linii sursă ale codului testat și verificat Indică acoperirea testărilor codului soft și integralitatea.

Date despre istoricul familiei de softuri dezvoltate în timp pentru a identifica problemel de proiectare, proces și producție.

Colectarea de metrici este necesară pentru a determina tendințele în fiabilitate, frecvența și timpul necesare pentru mentenanța softului, timpul de răspuns pentru apelurile de serviciu, refacerea performanțelor degradate și cerințele pentru suportul de mentenanță. Cantitatea și tipurile de metrici trebuie să fie relativ simplu de colectat, ușor de interpretat pentru analizare și folositoare pentru evaluarea, îmbunătățirea și creșterea fiabilității softului.

2.4. Mentenanța

Mentenanța definește *ansamblul tuturor activităților tehnice și organizatorice desfășurate pentru a menține sau restabili starea de bună funcționare a unui sistem tehnic reparabil* [48].

Mentenanța poate fi:

- *Mentenanță preventivă*, atunci când se referă la supravegherea bunei funcționări și la revizii periodice efectuate în scopul reducerii probabilității de defectare sau de degradare a performanțelor unui sistem. Se efectuează la intervale de timp predeterminate sau corespunzător unor criterii prestabilite, pe baza unui program de mentenanță, pornind de la ideea că este mai ieftin să previi decât să corectezi o defecțiune.

Mentenanța preventivă poate fi sistematică, condițională și previzionară.

Mentenanța sistematică este realizată prin activități de întreținere, reparații curente, revizii și reparații capitale. Se face pe baza unui plan tehnic normat de

intervenții, specific fiecărui tip de sistem.

Mentenanță condițională se realizează prin urmărirea parametrilor de uzură ai elementelor sau subansamblurilor cheie, cu ajutorul unor instrumente specifice urmând ca intervențiile de mentenanță să fie realizate înainte de apariția defectului iar mentenanța previzională se realizează pe baza analizei de evoluție unor parametri semnificativi pentru degradare sistemului.

Timpul de mentenanță preventivă cuprinde exclusiv timpul în care sunt efectuate observațiile și activitățile impuse în programul de mentenanță preventivă și nu include timpul necesar reparării unui modul (componentă) care a fost înlocuit.

- *Mentenanță corectivă* reprezintă ansamblul de activități realizate după defectarea sistemului sau după degradarea funcției sale în mod neprevăzut. Este efectuată după apariția unei defectări, în scopul restabilirii stării de bună funcționare a sistemului. Aceste activități constau în localizarea defectelor și diagnosticul acestora, repunerea în funcțiune cu sau fără modificări și controlul bunei funcționări. Mentenanța corectivă poate fi mentenanță curativă sau mentenanță paliativă.

Mentenanță curativă reprezintă activități de mentenanță corectivă ce au ca obiectiv repunerea unui sistem într-o stare specifică de funcționare cu îndeplinirea tuturor funcțiilor sale. Activitățile de mentenanță curativă pot fi reparații, modificări sau amenajări care au ca obiect eliminarea defecțiunilor;

Mentenanță paliativă presupune activități de mentenanță pentru a permite unui sistem, în mod provizoriu, îndeplinirea integrală sau parțială a funcțiilor sale. Termenul current pentru această mentenanță este *depanare*.

Timpul de mentenanță corectivă este timp de nefuncționare și este format din suma timpilor alocați pentru:

- pregătirea activităților de mentenanță;
- verificarea deficienței;
- localizarea defectului;
- procurarea pieselor de schimb, dacă este cazul și
- timpul efectiv consumat pentru reparare sau restabilire a stării de bună funcționare.

Fiecare acțiune de mentenanță costă, beneficiarul analizează costurile și alege totdeauna ținând cont de raportul cost/beneficiu, uneori este mai avantajos a înlocui sistemul sau părți ale sistemului cu altul nou sau a face un up-grade planificat

Mentenabilitatea reprezintă *probabilitatea ca un sistem să fie repus în stare de funcționare într-o perioadă de timp prestabilită*. Mentenabilitatea cuantifică calitatea acțiunilor de mentenanță și pentru aceasta este nevoie să se determine probabilitățile de desfășurare a activităților de mentenanță și distribuția timpilor necesari pentru efectuarea acestor activități și anume:

- timpul mediu pentru efectuarea activităților de mentenanță, menționate mai sus;
- frecvența de apariție a necesității unor acțiuni de mentenanță.

Există sisteme pentru care capacitatea de a îndeplini o anumită misiune poate fi realizată printr-o recondiționare totală sau parțială, printr-o reînnoire. Se consideră că orice reînnoire a unui sistem se face în momentul defectării lui iar timpul necesar pentru reînnoire este neglijabil. Procesul de reînnoire este un

proces aleator discret pentru care se pot calcula funcțiile de repartiție ale reînnoirilor dintr-un interval de timp.

Există cel puțin două moduri de abordare a reînnoirii sistemelor:

a) se consideră că sistemul este complet transformat prin reînnoire astfel încât după fiecare reînnoire sistemul este nou din punct de vedere al fiabilității. Aceste sunt *reînnoiri propriu-zise*, caracteristice sistemelor fără uzură;

b) se consideră că reînnoirea nu schimbă total caracteristicile sistemului, ci numai îl aduce din starea de defectare în cea de funcționare dar nu anulează efectul acumulării uzurii. În acest caz fiabilitatea sistemului este fie îmbunătățită, fie înrăutățită, în funcție de performanțele activității de reînnoire.

Mentenabilitatea depinde de fiabilitatea inițială a sistemului, a întregii lui structuri.

Mentenabilitatea se poate determina în două moduri:

a) **experimental** - prin simularea în laborator, pe platforma de probe (încecări) a diferitelor categorii de defecte și înregistrarea timpilor de intervenție pentru remedierea defectelor sau

b) prin **urmărirea comportării** sistemelor la beneficiari sau grupuri țintă și organizare de așa numite *bănci pentru date tehnice, cele mai bune practici* pe baza cărora se stabilește programul de mentenanță.

Capitolul 3. PRINCIPALELE LEGI DE DISTRIBUȚIE STATISTICĂ SPECIFICE FIABILITĂȚII

3.1. Tipuri de distribuție statistică

Descrierea fiabilității sistemelor prin intermediul unei legi de distribuție și a caracteristicilor numerice aferente constituie o abordare statistică a fiabilității. Legile de repartiție studiate în statistica matematică sunt adoptate în teoria fiabilității în măsura în care ele implică o funcție de fiabilitate de formă adecvată căreia să i se poată asocia o anumită interpretare fizică.

Momentele de timp la care se manifestă defecțiunile în cazul unui lot de elemente identice se repartizează potrivit unei legi de distribuție statistică, evidențiată prin intermediul expresiei funcției de distribuție $F(t)$ sau a funcției densitate de distribuție (frecvență), $f(t)$. După cum variabila aleatoare t (timpul în cazul studiilor de fiabilitate) ia valori discrete sau continue, distribuția este discretă sau continuă [1, 8, 17, 18, 39].

Dacă variabila aleatoare ia un număr finit de valori discrete t_1, t_2, \dots, t_n cu probabilitățile p_1, p_2, \dots, p_n , distribuția statistică se poate nota schematic astfel:

$$T \begin{pmatrix} t_1 & t_2 & \dots & t_k \\ p_1 & p_2 & \dots & p_k \end{pmatrix}. \quad (3.1.1)$$

Dacă p_i este probabilitatea de realizare a evenimentului i , iar probabilitatea de nerealizare a evenimentului i este q_i , legătura dintre aceste două probabilități este de complementaritate, adică:

$$p_i + q_i = 1. \quad (3.1.2)$$

Pentru o distribuție discretă, funcția de distribuție are expresia:

$$F(t) = P(T < t) = \sum_{i=1}^k p(t_i), \quad (3.1.3)$$

unde $p(t_i)$ este probabilitatea ca variabila aleatoare să ia valoarea t_i .

Graficul acestei distribuții are forma prezentată în figura 3.1.

Funcția de distribuție discretă este o funcție crescătoare, adică pentru două valori oarecare ale variabilei ($t_a < t_b$), există relația $F(t_a) < F(t_b)$, așa cum apare și în graficul din figura 3.1.

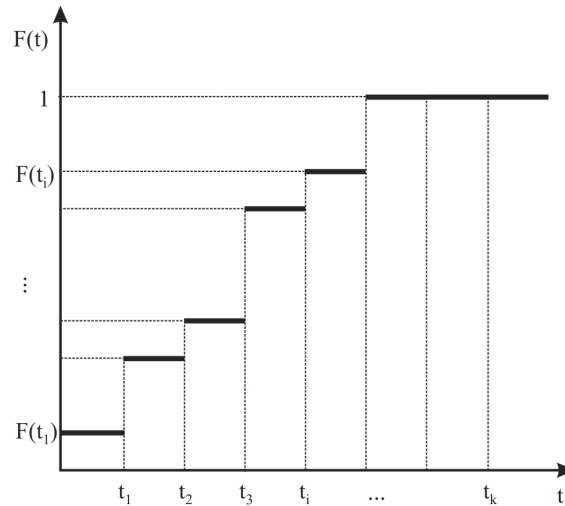


Figura 3.1. Funcția de distribuție pentru variabile aleatoare discrete

Valoarea medie a variabilei aleatoare discrete este, prin definiție:

$$m = M(T) = \bar{T} = \sum_{i=1}^n p_i t_i. \quad (3.1.4)$$

Dispersia variabilei T se definește prin relația:

$$D(T) = \sigma^2 = M[(T - M(T))^2] = \sum p_i (t_i - m)^2. \quad (3.1.5)$$

În cazul variabilei aleatoare continue nu interesează probabilitatea ca aceasta să ia o valoare bine determinată, ci probabilitatea ca să ia valori cuprinse într-un anumit interval.

Fiind dată o variabilă continuă T , probabilitatea realizării evenimentului ($a < T < b$) o vom nota prin $P(a < T < b)$.

$P(a < T < b)$ depinde de a și b , este deci o funcție de două variabile.

Funcția de distribuție a variabilei continue, T , este de forma:

$$F(t) = P(T \leq t), \quad (3.1.6)$$

și este o funcție de o singură variabilă.

Pentru orice variabilă T , discretă sau continuă, se poate scrie:

$$(T \leq t) = (T < t) \cup (T = t), \quad (3.1.7)$$

adică pentru a avea satisfăcută relația $T \leq t$, trebuie să fie satisfăcută cel puțin una din relațiile de incompatibilitate: $T < t$ și $T = t$.

De aici rezultă:

$$P(T \leq t) = P(T < t) + P(T = t). \quad (3.1.8)$$

Pentru variabilele aleatoare continue, T , pentru care $P(T = t) = 0$, oricare ar fi t , se poate scrie:

$$P(T \leq t) = F(t), \quad (3.1.9)$$

sau

$$P(T > t) = 1 - F(t) = R(t). \quad (3.1.10)$$

Într-adevăr, evenimentul sigur, E , se poate scrie:

$$E = (T > t) \cup (T \leq t), \quad (3.1.11)$$

$$1 = P(T > t) \cup P(T \leq t), \quad (3.1.11')$$

adică $1 = R(t) + F(t)$, relație cunoscută, identică cu relația (2.1.3) din capitolul 2 al acestei cărți.

Dacă $P(T = a) = 0$ nu înseamnă că evenimentul $T = a$ este imposibil, ci numai că probabilitatea sa de realizare este nulă. Totuși un eveniment imposibil are probabilitatea de apariție 0.

Funcția de distribuție are următoarele proprietăți:

$$a) F(t) \leq F(t') \text{ dacă } t < t', \text{ adică este crescătoare}; \quad (3.1.12)$$

$$b) F(-\infty) = 0, \text{ deoarece evenimentul } T < -\infty \text{ este imposibil}; \quad (3.1.12')$$

$$c) F(+\infty) = 1, \text{ deoarece evenimentul } T < +\infty \text{ este sigur}. \quad (3.1.12'')$$

Orice funcție de distribuție se poate considera definită pe R ($R =$ mulțimea numerelor reale, nu funcția de fiabilitate) chiar dacă variabila aleatoare ia valori numai într-un interval (a, b) , în restul intervalului având valorile:

$$\begin{cases} F(t) = 0, & \text{pentru } t \leq a, \\ F(t) = 1, & \text{pentru } t > b. \end{cases} \quad (3.1.13)$$

Densitatea de distribuție (sau de probabilitate) a variabilei continue care are funcția de distribuție F , este:

$$f(t) = \frac{dF}{dt}. \quad (3.1.14)$$

Orice densitate de repartiție $f(t)$ are următoarele proprietăți:

$$a) f(t) \geq 0 \text{ pentru orice } t, \quad (3.1.15)$$

deoarece este derivata întâi a unei funcții crescătoare.

$$b) \int_{-\infty}^{\infty} f(t) dt = 1, \quad (3.1.15')$$

$$c) \int_a^b f(t) dt = F(t) \Big|_a^b = F(b) - F(a). \quad (3.1.15'')$$

Dacă funcția $f(t)$ nu are toate aceste trei proprietăți, ea nu poate fi densitate de probabilitate.

De asemenea, și densitatea de distribuție se poate considera definită pe R , chiar dacă variabila aleatoare ia valori numai într-un anumit interval, în afara intervalului fiind 0.

În figura 3.2 sunt prezentate graficele funcțiilor $F(t)$ și $f(t)$ pentru o distribuție continuă oarecare:

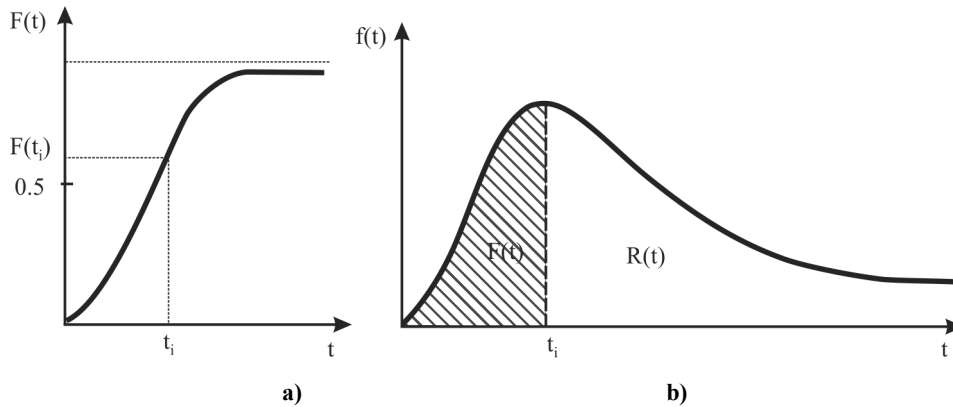


Figura 3.2. Funcția de distribuție și densitatea funcției de distribuție pentru variabile continue

$F(t)$ corespunde ariei hașurată de sub curba $f(t)$ din figura 3.2, b.

Valoarea medie a unei variabile aleatoare continue se definește prin relația:

$$m = M(T) = \int_{-\infty}^{\infty} t \cdot f(t) dt. \quad (3.1.16)$$

Dispersia unei variabile aleatoare continue este definită prin relația:

$$D = \sigma^2 = \int_{-\infty}^{\infty} (t - m)^2 f(t) dt. \quad (3.1.17)$$

O formulă practică de calcul a dispersiei este [47, 50]:

$$D(T) = M_2 - M_1, \quad (3.1.17')$$

unde:

$M_1 = m = M(T)$ valoarea mediei sau momentul centrat de ordinul 1,

M_2 – momentul centrat de ordinul 2.

Momentul centrat de ordinul r se calculează cu formulele:

$$M_r = \sum_{i=1}^n (t_i - m)^r f(x_i) \text{ pentru variabile discrete,} \quad (3.1.18)$$

sau

$$M_r = \int_{-\infty}^{\infty} (t - m)^r f(t) dt \text{ pentru variabile continue.} \quad (3.1.19)$$

Pentru $r = 1$ se obține momentul de ordinul 1, iar pentru $r = 2$ se obține momentul de ordinul 2.

În fiabilitate, pentru distribuția timpului de funcționare se utilizează ambele tipuri de distribuții, respectiv:

- distribuții discrete: binomială, polinomială, Poisson, hipergeometrică etc.;
- distribuții continue: normală, exponențială, Weibull, log-normală etc.;
- distribuții specifice: χ^2 (Hi-pătrat, Pearson), Γ (gama), t (Student), Fischer, de amestec etc.

În acest capitol se vor face scurte considerații asupra principalelor legi de distribuție folosite în teoria fiabilității, întrucât există lucrări clasice de teoria fiabilității și de statistică, în care sunt prezentate amănunțit proprietățile legilor de distribuție utilizate curent în teoria fiabilității [17,18].

3.2. Distribuția binomială

Este numită și "schema lui Bernoulli", deoarece se asociază cu experimentul care constă în n extrageri independente una de alta a unei bile de o anumită culoare dintr-o urnă, care conține un număr dat de bile de două culori diferite, dar în rest identice. Bila extrasă este reintrodusă în urnă după notarea culorii pentru asigurarea independenței extragerilor.

Fie n extracții succesive dintr-un lot de n produse, reintroducându-se în lot produsele extrase, după verificare;

- fiecare din produsele verificate poate fi defect, cu probabilitatea p (evenimentul A), sau corespunzător, cu probabilitatea q (evenimentul A negat, notat \bar{A});
- numărul de apariții ale evenimentului A în cele n experimente este k , unde k este o variabilă aleatoare, care poate lua valorile $0, 1, 2, \dots, n$;
- probabilitatea ca evenimentul A să se producă de k ori este:

$$P(A_k) = P_n(k) = C_n^k p^k q^{n-k} \quad (3.3.1)$$

iar distribuția variabilei aleatoare va avea următorul tablou:

$$\begin{vmatrix} 0 & 1 & \dots & k & \dots & n \\ q^n & C_n^1 p q^{n-1} & \dots & C_n^k p^k q^{n-k} & \dots & p^n \end{vmatrix}. \quad (3.3.2)$$

Se observă că probabilitățile $P_n(k)$ sunt tocmai termenii dezvoltării binomului $(p+q)^n$, de unde denumirea de *distribuție binomială*.

Valoarea medie este, conform definiției:

$$m = \sum_{k=1}^n k C_n^k p^k q^{n-k} = np. \quad (3.3.3)$$

Pentru a efectua calculul se ține seama de identitatea:

$$(px + q)^n = \sum_{k=0}^n C_n^k p^k x^k q^{n-k}. \quad (3.3.4)$$

Dacă derivăm relația (3.3.4) în raport cu x , se obține:

$$np(px + q)^{n-1} = \sum_{k=0}^n k C_n^k p^k x^{k-1} q^{n-k}. \quad (3.3.5)$$

Dacă se alege $x = 1$ și se ține seama că $p + q = 1$, relația (3.3.5) devine:

$$\sum_{k=0}^n k C_n^k p^k q^{n-k} = np \Rightarrow m = np. \quad (3.3.6)$$

Dispersia se calculează utilizând relația (3.1.17'). Pentru a calcula valoarea momentului de ordinul 2 se utilizează relația (3.1.18), procedând în mod asemănător ca la calculul valorii medii:

$$D = \sigma^2 = npq = np(1-p). \quad (3.3.7)$$

Densitatea de distribuție a distribuției binomiale este dată de probabilitatea realizării evenimentului A de k ori și are un singur parametru, pe k :

$$f(t, k) = P_n(k) = C_n^k p^k q^{n-k}. \quad (3.3.8)$$

Funcția de distribuție este dată de relația:

$$F(t, k) = \sum_{k=0}^n f(t, p) = \sum_{k=0}^n C_n^k p^k q^{n-k}. \quad (3.3.9)$$

Ținând seama de formula lui Stirling:

$$n! = n^n e^{-n} \sqrt{2\pi n} e^{u_n} \approx n^n e^{-n} \sqrt{2\pi n}, \quad (3.3.10)$$

unde $0 < n < \frac{1}{12n}$; cele două funcții (densitatea de distribuție și funcția de distribuție) se pot scrie:

$$f(t, k) = C_n^k p^k q^{n-k} = \frac{n!}{k!(n-k)!} p^k q^{n-k}, \quad (3.3.11)$$

$$F(t, k) = \sum_{k=0}^x \frac{n!}{k!(n-k)!} p^k q^{n-k} = \sum_{k=0}^n \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}. \quad (3.3.12)$$

Graficul funcției de repartiție a distribuției binomiale are n trepte, corespunzătoare celor $n + 1$ puncte de discontinuitate.

Există tabele care dau valorile probabilităților $P_n(k)$ și ale repartiției $F(x)$ pentru n, k, p și x cunoscute [46, 90, 93].

Această distribuție interesează în special pentru a calcula probabilitatea extragerii unor bile/numere dintr-o urnă sau în cazurile în care se urmărește a se cunoaște câte produse dintr-un lot (sau ce procentaj) se găsesc în funcțiune la un moment dat, observațiile făcându-se la intervale de timp egale. În acest ultim caz, variabila este numărul de produse în stare de funcționare.

3.3. Distribuția Poisson

Dacă o variabilă ia valori discrete, respectiv 0, 1, 2, 3, ..., k , atunci probabilitatea ca variabila să ia valoarea k este de forma:

$$P(T = k; a) = \frac{a^k}{k!} e^{-a}, \quad (3.3.1)$$

atunci variabila discretă are distribuția Poisson.

Parametrul a se numește *parametrul distribuției Poisson*.

Tabloul distribuției acestei variabile este:

$$\left(\begin{array}{cccccc} 0 & 1 & 2 & 3 & \dots & k \dots \\ e^{-a} & \frac{a}{1!} e^{-a} & \frac{a^2}{2!} e^{-a} & \frac{a^3}{3!} e^{-a} & \dots & \frac{a^k}{k!} e^{-a} \dots \end{array} \right). \quad (3.3.2)$$

Densitatea de distribuție depinde de un singur parametru, a , și are forma:

$$f(t; a) = \frac{a^k}{k!} e^{-a}. \quad (3.3.3)$$

Funcția de distribuție corespunzătoare are expresia:

$$F(t; a) = \sum_{k=0}^t \frac{a^k}{k!} e^{-a}. \quad (3.3.4)$$

Valoarea medie se calculează cu formula de definiție (1.1.4):

$$m = \sum_{k=0}^{\infty} k \frac{a^k}{k!} e^{-a} = a e^{-a} \cdot e^a = a. \quad (3.3.5)$$

Dispersia se calculează cu următoarea formulă:

$$\begin{aligned} D &= M(T^2) - [M(T)]^2 = \sum k^2 \frac{a^k}{k!} e^{-a} - a^2 = \\ &= e^{-a} (a^2 e^a + a e^a) = a^2 + a - a^2 = a. \end{aligned} \quad (3.3.6)$$

După cum se poate observa, atât valoarea medie cât și dispersia au valori egale cu parametrul distribuției.

Distribuția Poisson este un caz limită al distribuției binomiale, obținându-se când n este foarte mare, iar probabilitatea de apariție a evenimentului k examinat, este mică. Atunci când n este foarte mare în raport cu k , se pot aproxima cu n toate valorile: $n-1$, $n-2$ etc., iar densitatea de probabilitate a distribuției binomiale se poate scrie:

$$\begin{aligned} f(t; k, a) &= \frac{n^k}{k!} p^k (1-p)^{n-k} = \frac{(np)^k}{k!} (1-p)^n = \\ &= \frac{a^k}{k!} \left(1 - \frac{a}{n}\right)^n = \frac{a^k}{k!} e^{-a}. \end{aligned} \quad (3.3.7)$$

Funcția descrisă de relația (3.3.7) este tocmai densitatea de probabilitate a distribuției Poisson, având ca parametru pe $a = n \cdot p$.

Este o distribuție de tip discret, denumită și *distribuția evenimentelor rare* și este foarte utilă în studiul fiabilității producțiilor omogene, de serie mare, atunci când probabilitatea de apariție a defectelor este foarte mică.

Atunci când parametrul a crește, distribuția Poisson tinde să se suprapună cu valorile distribuției normale, ceea ce în practică are loc pentru $a > 30$. Tabelarea valorilor funcțiilor Poisson și binomială este dată în literatura de specialitate [46, 50].

3.4. Distribuția normală (Gauss-Laplace)

3.4.1. Funcția de distribuție normală

Variabila aleatoare continuă, T , are o distribuție normală de parametri m și σ , dacă densitatea sa de distribuție este de forma:

$$f(t; m, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m)^2}{2\sigma^2}}. \quad (3.5.1)$$

unde $t > 0$ și $\sigma > 0$.

Graficul funcției $f(t; m, \sigma)$ este dat în figura 3.3, depinde de doi parametri, m și σ , și are următoarele proprietăți:

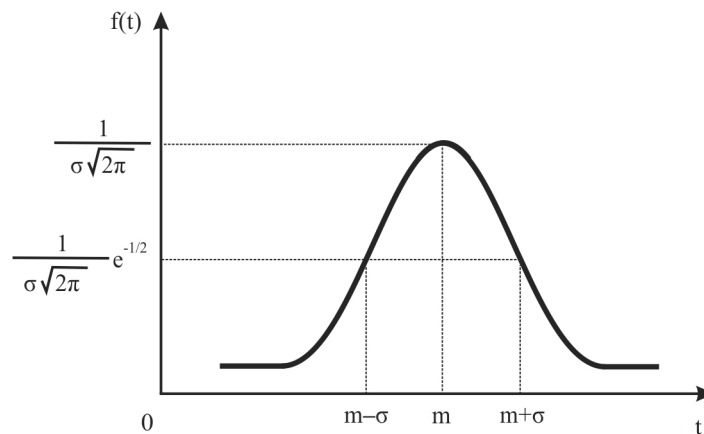


Figura 3.3. Variația în timp a densității de repartiție $f(t)$ în cazul distribuției normale

– este simetrică față de axa $t = m$:

$$f(m + \alpha; m, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(m+\alpha-m)^2}{2\sigma^2}} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\alpha^2}{2\sigma^2}}, \quad (3.5.2)$$

$$f(m - \alpha; m, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(m-\alpha-m)^2}{2\sigma^2}} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\alpha^2}{2\sigma^2}}; \quad (3.5.2')$$

– are un maxim pentru $t = m$ a cărui valoare este:

$$f(m; m, \sigma) = \frac{1}{\sigma\sqrt{2\pi}}; \quad (3.5.3)$$

– punctele $t = m \pm \sigma$ sunt puncte de inflexiune.

În aceste puncte valoarea funcției este:

$$f(m \pm \sigma; m, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}}. \quad (3.5.4)$$

Între limitele $m + \sigma$ și $m - \sigma$ se încadrează 63,27% din valorile statistice.

Indiferent de valorile parametrilor m și σ graficul acestei funcții are forma de clopot (*clopotul lui Gauss*).

Parametrul m definește axa de simetrie, iar σ stabilește înălțimea graficului în punctul de maxim. Cu cât σ este mai mic, cu atât ordonata punctului de maxim al curbei este mai mare, clopotul este mai înalt, dispersia (împrăștierea) valorilor variabilelor aleatoare este mai mică.

În figura 3.4, a sunt date graficele unor funcții $f(t; m, \sigma)$ pentru aceeași valoare a lui σ și pentru trei valori diferite ale lui m ($m_1 < m_2 < m_3$), iar în figura 3.4, b sunt date graficele pentru aceeași valoare a lui m și trei valori diferite ale lui σ ($\sigma_1 < \sigma_2 < \sigma_3$).

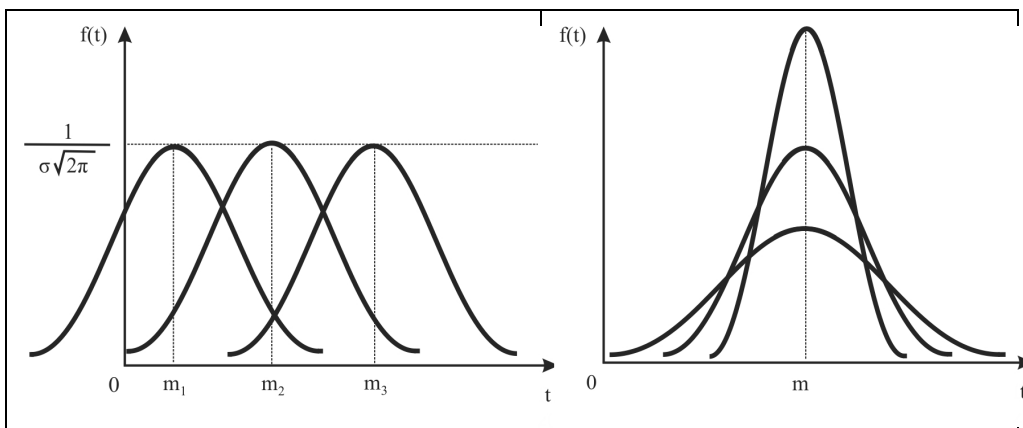


Figura 3.4. Graficul densității de distribuție în funcție de parametrul m (a) și σ (b)

Distribuția normală este o lege deosebit de importantă în statistică, deoarece foarte multe fenomene în care intervine întâmplarea, urmează această distribuție. Cu ajutorul distribuției normale pot fi approximate și alte distribuții statistice (de exemplu: distribuțiile binomială și Poisson). În fiabilitate distribuția normală caracterizează fenomenele de îmbătrânire mecanică, electrică, termică etc. a elementelor și sistemelor.

Funcția de repartiție a unei variabile cu distribuție aleatoare normală, T , este:

$$\begin{aligned}
F(T) = P(T \leq t) &= \frac{1}{\sigma\sqrt{2\pi}} \int_0^t f(\tau) d\tau = \\
&= \frac{1}{\sigma\sqrt{2\pi}} \int_0^t e^{-\frac{(\tau-m)^2}{2\sigma^2}} d\tau = F(t; m, \sigma).
\end{aligned} \tag{3.5.5}$$

Graficul acestei funcții este dat în figura 3.5.

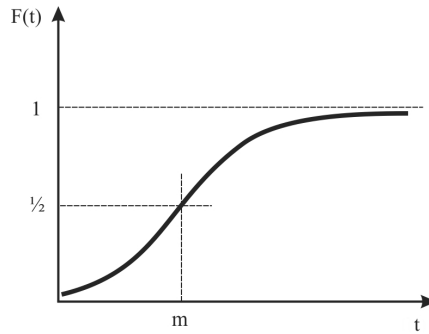


Figura 3.5. Funcția distribuției normale

3.4.2. Distribuția normală normalată

În relația (3.5.5) se poate face o schimbare de variabilă, folosind *variabila aleatoare normalată*, u , care este [46, 50]:

$$u = \frac{t-m}{\sigma}, \tag{3.5.6}$$

din care rezultă că $t = \sigma \cdot u + m$ iar

$$d t = \sigma d u. \tag{3.5.6'}$$

Dacă variabila T are o distribuție normală, cu parametrii m și σ , atunci și variabila aleatoare normalată u este normal distribuită. Pornind de la relația 3.5.5, funcția de distribuție a variabilei aleatoare normale normalate se poate scrie:

$$\begin{aligned}
F(u; 0, 1) &= \frac{1}{\sigma\sqrt{2\pi}} \int_0^{\frac{t-m}{\sigma}} e^{-\frac{u^2}{2}} \sigma du = \\
&= \frac{1}{\sqrt{2\pi}} \int_0^{\frac{t-m}{\sigma}} e^{-\frac{u^2}{2}} du = \int_0^{\frac{t-m}{\sigma}} N(u; 0, 1) du,
\end{aligned} \tag{3.5.7}$$

unde cu $N(u; 0, 1)$ s-a notat funcția $\frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}$, care corespunde valorilor $m = 0$ și $\sigma = 1$, și se numește **legea normală normată** sau *repartiția normală redusă* (*repartiție standard*).

Densitatea de distribuție a repartiției normale normate este de forma:

$$f(u; 0, 1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}. \quad (3.5.8)$$

Întrucât distribuția normală normată este simetrică față de axa ordonatelor, se poate scrie:

$$\int_{-\infty}^{+\infty} N(u; 0, 1) du = 1 = 2 \int_0^{\infty} N(u; 0, 1) du = 2 \int_{-\infty}^0 N(u; 0, 1) du, \quad (3.5.9)$$

de unde rezultă că:

$$\int_0^{\infty} N(u; 0, 1) du = \int_{-\infty}^0 N(u; 0, 1) du = \frac{1}{2}. \quad (3.5.9')$$

Se poate scrie următoarea egalitate:

$$\begin{aligned} \int_{-\infty}^{\alpha} N(u; 0, 1) du &= \int_{-\infty}^0 N(u; 0, 1) du + \\ &+ \int_0^{\alpha} N(u; 0, 1) du = \frac{1}{2} + \phi(\alpha). \end{aligned} \quad (3.5.10)$$

Funcția

$$\phi(\alpha) = \frac{1}{\sqrt{2\pi}} \int_0^{\alpha} e^{-\frac{u^2}{2}} du \quad (3.5.11)$$

se numește **funcția integrală a lui Laplace**, iar α **cuantila funcției Laplace**.

Deoarece primitiva funcției $e^{-\frac{u^2}{2}}$ nu se poate exprima cu ajutorul funcțiilor elementare, valorile funcției $\phi(\alpha)$ sunt tabelate și date în cărțile de statistică [34, 46, 47, 50].

În continuare vor fi enumerate câteva dintre proprietățile funcției Laplace:

$$\begin{aligned} \phi(-\alpha) &= \phi(\alpha), \\ \phi(0) &= 0, \end{aligned}$$

Reprezentarea densității de probabilitate a distribuției normale normalată este dată în figura 3.6, pentru $t \in (-\infty, \infty)$; (valoarea maximă a acestei funcții se obține în punctul $m = 0$, și este $1/\sqrt{2\pi} \approx 0,3989$).

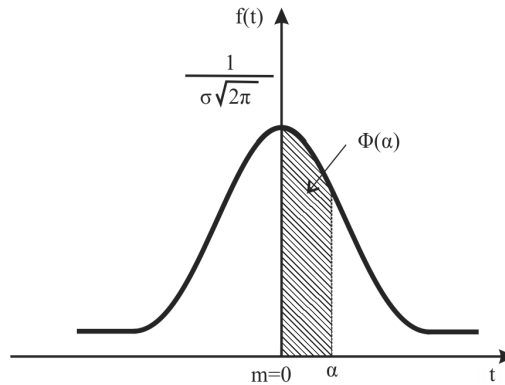


Figura 3.6. Densitatea de probabilitate a distribuției normale normalate

Funcția de repartiție (relația 3.5.7) se poate scrie cu ajutorul funcției Laplace, pe intervalul $[0, t]$:

$$F(t) = \phi\left(\frac{t-m}{\sigma}\right) = \Phi(\alpha), \quad (3.5.12)$$

unde $\alpha = \frac{t-m}{\sigma}$.

Pentru $\sigma = 1$, se obține

$$\alpha = t - m. \quad (3.5.12')$$

3.4.3. Valoarea medie și dispersia unei variabile cu distribuție normală

Fie variabila normală T , cu parametrii m și σ .

Media ei este dată de relația:

$$M(T) = \int_{-\infty}^{\infty} t \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m)^2}{2\sigma^2}} dt. \quad (3.5.13)$$

Pentru a calcula valoarea mediei se recurge la o schimbare de variabilă:

$$\frac{t-m}{\sigma} = y \Rightarrow t = \sigma y - m \Rightarrow dt = \sigma dy. \quad (3.5.14)$$

Cu această schimbare de variabilă se obține:

$$\begin{aligned}
M(T) &= \int_{-\infty}^{\infty} \frac{m + \sigma y}{\sigma\sqrt{2\pi}} e^{-\frac{y^2}{2}} \sigma dy = \\
&= \frac{m}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{y^2}{2}} dy + \frac{\sigma}{\sqrt{2\pi}} \int_{-\infty}^{\infty} y e^{-\frac{y^2}{2}} dy.
\end{aligned} \tag{3.5.15}$$

Dar:

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{y^2}{2}} dy = \int_{-\infty}^{+\infty} N(y; 0, 1) = 1,$$

și

$$\frac{1}{2} \int_{-\infty}^{\infty} y e^{-\frac{y^2}{2}} dy = \left[-e^{-\frac{y^2}{2}} \right]_{-\infty}^{\infty} = 0, \tag{3.5.16}$$

deci

$$M(T) = m. \tag{3.5.17}$$

Valoarea medie a variabilei T cu distribuție normală este tocmai parametrul m al distribuției.

Dispersia este dată de relația:

$$D(T) = \int_{-\infty}^{\infty} \frac{(t-m)^2}{\sigma\sqrt{2\pi}} e^{-\frac{(t-m)^2}{2\sigma^2}} dt = \sigma^2. \tag{3.5.18}$$

Apelând la aceeași schimbare de variabilă, se poate scrie:

$$D(T) = \frac{\sigma^2}{\sqrt{2\pi}} \int_{-\infty}^{\infty} y^2 e^{-\frac{y^2}{2}} dy. \tag{3.5.19}$$

Se face o integrare prin părți, luând

$$u = \frac{1}{\sqrt{2\pi}} y, \quad v' = y e^{-\frac{y^2}{2}}, \tag{3.5.20}$$

de unde:

$$u' = \frac{1}{\sqrt{2\pi}} \text{ si } v = -e^{-\frac{y^2}{2}}.$$

Relația 3.3.19 se scrie:

$$D(T) = \sigma^2 \left[-\frac{1}{\sqrt{2\pi}} y e^{-\frac{y^2}{2}} \right]_{-\infty}^{\infty} + \frac{\sigma^2}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{y^2}{2}} dy. \tag{3.5.21}$$

Dacă se ține seama că:

$$\lim_{y \rightarrow \pm\infty} \frac{y}{e^{-\frac{y^2}{2}}} = 0$$

și

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy = \int_{-\infty}^{\infty} N(y; 0, 1) dy = 1, \quad (3.5.22)$$

relația (3.3.21) devine:

$$D(T) = \sigma^2, \quad (3.5.23)$$

adică, abaterea medie pătratică a distribuției normale este tocmai parametrul σ al acestei distribuții.

Variația indicatorilor de fiabilitate, în cazul distribuției normale, este dată în figura 3.7.

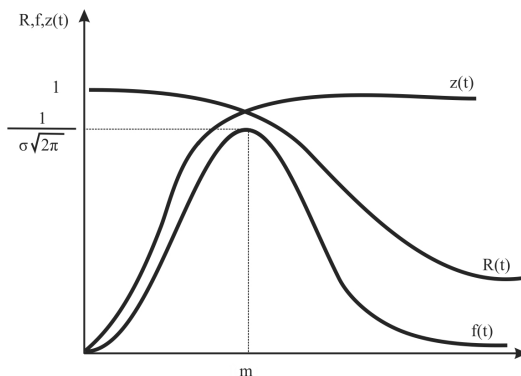


Figura 3.7. Indicatorii (caracteristicile) de fiabilitate în cazul legii normale

3.5. Distribuția exponențială

Această lege de distribuție se mai numește și exponențial negativă și este caracterizată prin aceea că rata de defectare este constantă ($z(t) = \lambda = \text{constant}$). Graficul funcției $z(t)$ este o paralelă la axa timpului; manifestarea legii are loc tocmai pe durata vieții utile a produsului, adică în zona II (perioada de maturitate), din figura 1.4.

Densitatea de distribuție este dată de relația:

$$f(t) = \lambda \cdot e^{-\lambda \cdot t} \text{ pentru } t \geq 0. \quad (3.6.1)$$

Funcția de repartiție a distribuției exponențiale este dată de relația:

$$F(t) = 1 - e^{-\lambda \cdot t} \text{ pentru } t \geq 0, \quad (3.6.2)$$

iar funcția de fiabilitate este dată de relația:

$$R(t) = e^{-\lambda \cdot t}. \quad (3.6.3)$$

Valoarea medie este:

$$m = \frac{1}{\lambda}, \quad (3.6.4)$$

iar dispersia este dată de relația:

$$D = \frac{1}{\lambda^2}. \quad (3.6.5)$$

Variația indicatorilor de fiabilitate este prezentată în figura 3.8.

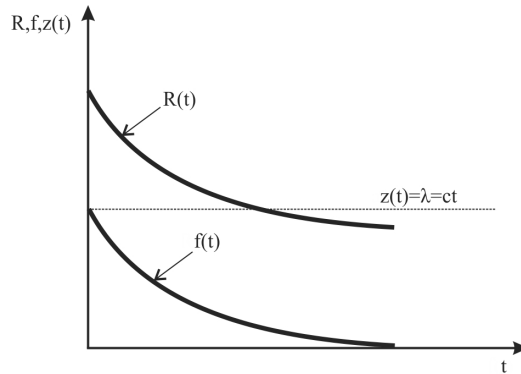


Figura 3.8. Variația indicatorilor de fiabilitate în cazul legii exponențiale

Pentru $t = m$ se obține $R(t = m) = 0,37$, iar $f(m) = 0,37/m$, așa cum se poate vedea în figura 3.9, unde sunt date dependențele de timp ale principalilor indicatori de fiabilitate, în cazul legii exponențiale, cu evidențierea valorilor acestora în punctul $t = m$.

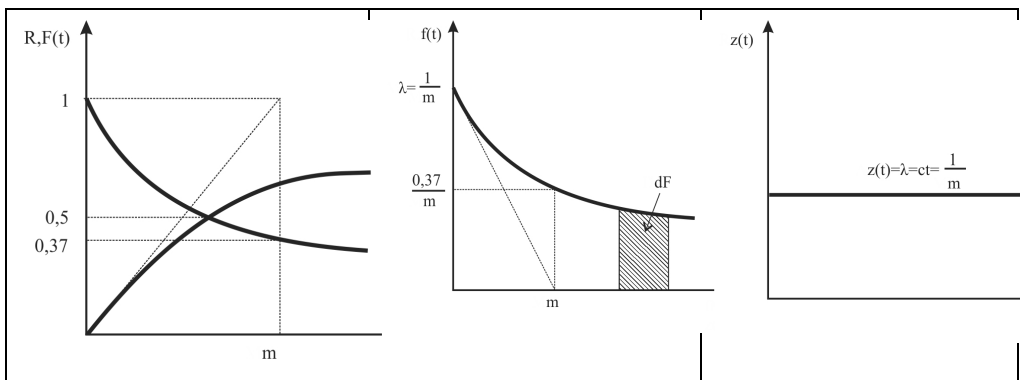


Figura 3.9. Indicatorii de fiabilitate, în cazul legii exponențiale, cu evidențierea valorilor acestora în punctul $t = m$. a) $R(t)$ și $F(t)$; b) $f(t)$; c) $z(t)$

Valorile funcției exponențiale $y = \exp(-x)$ sunt tabelate, de exemplu, în [34].

Legea distribuției exponențiale se poate aplica cu rezultate bune în cazul produselor electronice și a unor sisteme tehnice complexe.

Tabelul 3.1

Indicator	Legea de distribuție		
	exponențială	Normală	Weibull
$f(t)$	$\lambda \exp(-\lambda t)$	$\frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{t-m}{\sigma}\right)^2\right]$	$\frac{\beta}{\theta}(t-t_0)^{\beta-1} \exp\left[-\frac{(t-t_0)^\beta}{\theta}\right]$
$R(t)$	$\exp(-\lambda t)$	$1 - \Phi\left(\frac{t-m}{\sigma}\right)^*$	$\exp\left[-\frac{(t-t_0)^\beta}{\theta}\right]$
$F(t)$	$1 - \exp(-\lambda t)$	$\Phi\left(\frac{t-m}{\sigma}\right)^*$	$1 - \exp\left[-\frac{(t-t_0)^\beta}{\theta}\right]$
$z(t)$	λ	$\frac{f(t)}{\Phi\left(\frac{t-m}{\sigma}\right)}$	$\frac{\beta(t-t_0)^{\beta-1}}{\theta}$
m	$1/\lambda$	m	$t_0 + \theta^{-\beta} \Gamma\left(\frac{1}{\beta} + 1\right)$
D	$1/\lambda^2$	σ^2	$\theta^{-2\beta} \left[\Gamma\left(\frac{2}{\beta} + 1\right) - \Gamma^2\left(\frac{1}{\beta} + 1\right) \right]$

Observații:

$$* \Phi(u) = \int_0^u \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} du \text{ este funcția Laplace, unde } u = \frac{m-t}{\sigma}. \quad (3.7.6)$$

$$** \Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt - \text{funcția Gama sau integrala Euler de speța II, funcție cu valori tabelate în cărțile de statistică.} \quad (3.7.7)$$

3.6. Distribuția log-normală

Este o distribuție utilizată, mai ales, pentru mașini, utilaje și componente electronice, care se degradează, în principal, datorită fenomenului de oboseală termică.

Densitatea de distribuție are forma:

$$f(t) = \frac{1}{\sigma t \sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\ln t - m}{\sigma}\right)^2\right]. \quad (3.8.1)$$

Funcția de distribuție are expresia:

$$F(t) = \phi\left(\frac{\ln t - m}{\sigma}\right). \quad (3.8.2)$$

Valoarea medie :

$$M(T) = \int_0^{\infty} t f(t) dt = \frac{1}{\sigma\sqrt{2\pi}} \int_0^{\infty} e^{-\frac{(t-m)^2}{2\sigma^2}} dt. \quad (3.8.3)$$

Făcând schimbarea de variabilă :

$$\frac{\ln t - m}{\sigma} = u, \quad (3.8.3')$$

se obține: $M(T) = e^{\left(m + \frac{\sigma^2}{2}\right)}$.

Dispersia este:

$$D(T) = \int_0^{\infty} (t - M(T))^2 f(t) dt = e^{2m + \sigma^2} (\sigma^2 - 1). \quad (3.8.4)$$

În figura 3.10 sunt prezentate dependențele de timp pentru funcțiile de densitate de distribuție, de fiabilitate și rata de defectare [1, 8,12], ale distribuției log-normale.

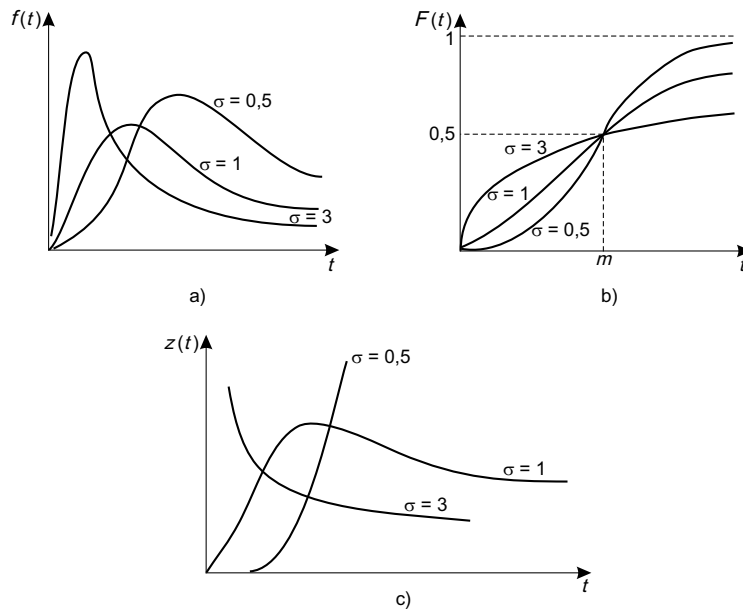


Figura 3.10. Indicatorii de fiabilitate în cazul distribuției log-normale
 a) $f(t)$; b) $R(t)$; c) $z(t)$

Distribuția log-normală prezintă două avantaje majore în fiabilitate:

1. Dacă $t = 0$, atunci $F(t) = 0$, proprietate pe care nu o are distribuția normală. Distribuția log-normală este avantajos să se folosească atunci când este nevoie de valorile parametrilor la momentul $t = 0$.

2. Produsul unor variabile aleatoare cu distribuție log-normală este tot o variabilă aleatoare cu distribuție log-normală.

Există numeroase alte funcții statistice care pot fi utilizate în fiabilitate, de exemplu distribuția uniformă, distribuția Weibul, distribuția Γ (Gama), χ^2 (Hi-pătrat), student, Fisher etc. care pot fi studiate de cei interesați în cărțile de specialitate [1,17].

Capitolul 4. STUDIUL FIABILITĂȚII SISTEMELOR PE BAZA BLOCURILOR LOGICE DE FIABILITATE

4.1. Definirea sistemelor complexe

Problemele privind fiabilitatea sunt asemănătoare din mai multe puncte de vedere pentru orice sistem tehnic. Din acest motiv, pentru *produsele complexe* vom utiliza termenul generalizator de "*sisteme tehnice*" sau numai "*sisteme*". Un sistem are bine definite intrările și ieșirile. În categoria sistemelor tehnice intră inclusiv sistemele hardware, sisteme software și componente electronice. Sistemele de software au câteva particularități, de aceea anumite aspecte specifice numai fiabilității software vor fi tratate într-un capitol separat din această carte, capitolul 5.

Prin sistem înțelegem (în contextul acestei cărți) un ansamblu de elemente legate funcțional între ele care realizează a anumită funcție tehnică concretă.

Prin *element* vom înțelege o parte componentă a sistemului. Pentru studii de fiabilitate un sistem se poate diviza într-un număr de elemente componente, în funcție de necesitățile de calcul, divizare efectuată astfel încât pentru fiecare element să se poată identifica funcția pe care o realizează în cadrul sistemului și defini intrările și ieșirile. Această divizare are un caracter convențional pentru că elementele componente ale unui sistem, o dată precizate, pot fi din nou divizate în elemente de rangul al doilea ș.a.m.d. În mod curent, elementele sistemului se numesc "subansamble", "blocuri", "unități" "module", diviziunea de rangul cel mai mic fiind "componenta" sau "linia de cod" pentru software.

Buna funcționare a produsului complex este rezultanta funcționării corecte a tuturor subansamblelor componente. La rândul lor subansamblele se compun din blocuri, module ș.a.m.d. până când procesul de divizare identifică componentele, elementele de legătură, reperele și toate celelalte elemente, care determină funcționarea corectă și fiabilitatea sistemului.

Prin fiabilitate previzională se înțelege fiabilitatea exprimată prin indicatori de fiabilitate care au rezultat din calcule de prognoză efectuate pe baza datelor de fiabilitate ale tuturor elementelor componente.

Fiabilitatea sistemului depinde de fiabilitatea tuturor elementelor componente (sau subsistemelor), de legăturile cauzale stabilite între ele și de mediul în care funcționează [1,8, 12, 47, 50]. Dacă un element se defectează, funcționarea sistemului este afectată total sau parțial și pot apare următoarele situații:

- defectarea are un **caracter minor**, sistemul continuă să funcționeze (ex. o lampă de semnalizare);
- defectarea are un **caracter major** dacă întreaga funcționare a sistemului este afectată (ex. defectarea totală a unor subsisteme componente);
- o defectare are un caracter major iar funcțiile subsistemului respectiv pot fi suplinite de un subsistem identic, aflat în rezervă (**redundanță**), funcțiile

respective pot preluate de sistemul redundant până la terminarea acțiunii de reparare. Vom reveni asupra redundanței.

Pentru studiul fiabilității unui sistem este necesar să se definească **vectorul de intrare** $U = (u_1, u_2, \dots, u_n)$, **vectorul de ieșire** $Y = (y_1, y_2, \dots, y_p)$ și **vectorul de stare** $S = (s_1, s_2, \dots, s_m)$ prin intermediul cărora se manifestă influența variabilelor de intrare asupra celor de ieșire, propriu fiecărui sistem. Toți acești vectori fiind mărimi fizice aleatoare.

În figura 4.1 sunt evidențiați aceste 3 vectori pentru un sistem descompus în două subsisteme, variabilele de stare fiind ieșiri pentru primul subsistem și intrări pentru cel de-al doilea.

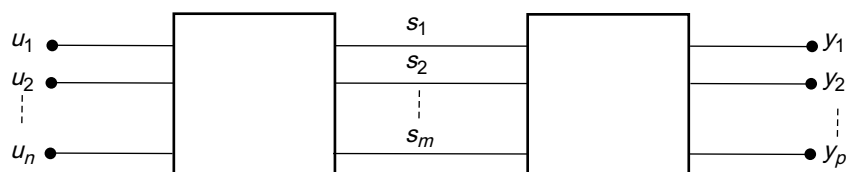


Figura 4.1. Reprezentarea simplă a unui sistem, cu evidențierea vectorilor de intrare, de ieșire și de stare

În condiții generale, un sistem poate fi descris matematic cu ajutorul ecuațiilor canonice de stare [8]:

$$\begin{cases} \dot{S} = B(S, U), & (4.1.1, a) \\ Y = C(S, U). & (4.1.1, b) \end{cases}$$

Există o largă posibilitate de alegere a vectorului de stare, oricând se poate alege un vector de stare astfel încât subsistemul, care reprezintă relația stare-ieșire, să fie simplu.

Vectorul de ieșire este complet determinat de vectorul de stare, care include și manifestarea vectorului de intrare. Relația 4.1.1, b se poate scrie:

$$Y = D(S). \quad (4.1.2)$$

Toate modelele de analiză previzională a fiabilității sistemelor se bazează pe relația generală dintre performanțele sistemului și parametrii componentelor acestuia, relație care poate fi scrisă explicit astfel:

$$y_i = f_i(y_1^1, y_1^2, \dots, y_1^{l_1}, y_2^1, y_2^2, \dots, y_2^{l_2}, \dots, y_p^1, y_p^2, \dots, y_p^{l_n}), \quad (4.1.3)$$

unde: $i = 1, \dots, p$ și $l = 1, \dots, n$.

Relațiile (4.1.2) și (4.1.3) exprimă *modelul funcțional al sistemului*.

Considerând că dependența parametrilor componentelor de solicitare se exprimă prin indicatorii de fiabilitate și funcțiile de distribuție specifice se realizează calculul indicatorilor de fiabilitate ai sistemelor pornind de la indicatorii de fiabilitate ai componentelor, elaborând astfel modele structurale.

Analiza structurală a fiabilității are drept scop stabilirea unei relații între funcția de fiabilitate a sistemului și funcțiile de fiabilitate ale elementelor componente. Analiza este precedată de o evaluare realistă a funcțiilor de fiabilitate individuale ale elementelor componente $\{R_j, j = 1, 2, \dots, n\}$, care trebuie să țină seama de criteriile de defectare reale, impuse de structura sistemului și de definirea corectă a defecțiunii elementelor și sistemului.

Dacă se cunosc caracteristicile de fiabilitate ale elementelor constituente (determinate experimental în prealabil și stocate într-o bancă de date, prezentate în cataloage sau existente în tabele) și dacă se ține seama de rolul și solicitarea acestora în sistem, se pot estima caracteristicile de fiabilitate ale sistemelor, utilizând modele structurale adecvate, arbori de defectare sau de funcționare, analiză pe baza lanțurilor Markov sau alte modele specifice. O serie de metode de evaluare a fiabilității sistemelor și componentelor acestora au fost prezentate în cărțile din bibliografie[1, 8, 12].

4.2. Dezvoltarea modelului diagrame bloc de fiabilitate

Funcția de fiabilitate a unui sistem trebuie calculată în funcție de fiabilitatea componentelor sale:

$$R_s = \Psi(R_1, R_2, \dots, R_n). \quad (4.2.1)$$

Scopul modelelor bazate pe blocuri logice de fiabilitate (BLF) este ca, pornind de la funcția de structură S a sistemului să se obțină o relație între funcția de fiabilitate a sistemului și cea a elementelor componente.

Pentru elaborarea schemei cu blocuri logice de fiabilitate se procedează astfel:

- se definește stare de bună funcționare/de defectare a sistemului;
- se identifică modulele din sistem pornind de la schema bloc a sistemului astfel încât fiecare bloc să fie independent din punct de vedere statistic de celelalte. Este de preferat ca fiecare bloc să nu conțină nici o redundanță. În schema logică de fiabilitate modulul reprezintă elementul pentru care se poate identifica clar vectorul de intrare și cel de ieșire;
- se identifică căile și direcțiile de transmitere a informației de bună funcționare;
- se conectează modulele prin linii și se realizează schema logică de fiabilitate astfel încât acestea să formeze o "*cale reușită*". Diferitele căi reușite dintre punctele de intrare și ieșire ale diagramei, trec prin acele combinații de blocuri care trebuie să funcționeze pentru ca sistemul să funcționeze.
- se definește starea de bună funcționare (succes) și cea de defect (nefuncționare) pentru fiecare modul component și pentru întregul sistem. Numărul de stări posibile ale unui sistem este 2^n , unde n este numărul de module constituente.

Se stabilește astfel expresia logică a fiabilității sistemului, în funcție de modul de conectare al elementelor componente.

Pentru cele mai multe sisteme complexe o asemenea analiză conduce la identificarea *grafului de arborescență* a fabricării, asamblării, întreținerii și reparării produsului. La grafuri de arborescență vom reveni.

4.3 Tipuri de conexiuni

4.3.1. Sisteme de tip serie

Sistemele de tip serie se caracterizează prin aceea că defectarea unui element determină ieșirea din funcțiune a întregului sistem. Reprezentarea unui astfel de sistem cu n componente poate fi urmărită în figura 4.2.

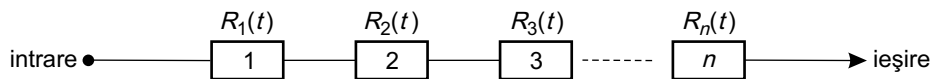


Figura 4.2. Sistem de tip serie

Se consideră că pentru fiecare element se cunoaște:

λ_i – rata de defectare a componentei i ;

$R_i(t)$ – funcția de fiabilitate, respectiv probabilitatea de bună funcționare a componentei i .

Sistemul este caracterizat de λ_s și $R_s(t)$ care sunt rata de defectare și, respectiv, probabilitatea de bună funcționare a sistemului.

Rata de defectare a sistemului, indiferent de funcția de distribuție a timpului de bună funcționare a sistemului respectiv, se calculează cu formula:

$$\lambda_s = \sum_{i=1}^n \lambda_i. \quad (4.3.1)$$

Probabilitatea de bună funcționare a acestui sistem, de tip serie, se calculează conform relației:

$$R_s(t) = R_1(t) \cdot R_2(t) \cdot \dots \cdot R_n = \prod_{i=1}^n R_i(t). \quad (4.3.2)$$

Exemplu de calcul: Fie un sistem alcătuit din $i = 4$ elemente cu ratele de defectare:

$$\lambda_1 = 0,1 \cdot 10^{-6} \text{ h}^{-1}, \quad \lambda_2 = 0,2 \cdot 10^{-6} \text{ h}^{-1}, \quad \lambda_3 = \lambda_4 = 0,5 \cdot 10^{-6} \text{ h}^{-1}.$$

Funcția de distribuție a timpului de bună funcționare a sistemului este considerată a fi exponențială.

Probabilitatea de bună funcționare a sistemului realizat cu aceste patru componente este fiind produsul probabilităților de bună funcționare a componentelor. Este tot o funcție exponențială, fiind de forma:

$$R_s(t) = \exp(-0,1 \cdot 10^{-6} \cdot t) \cdot \exp(-0,2 \cdot 10^{-6} \cdot t) \cdot \exp(-0,5 \cdot 10^{-6} \cdot t) \cdot \exp(-0,5 \cdot 10^{-6} \cdot t) = \exp(-1,3 \cdot 10^{-6} \cdot t).$$

Pentru un timp de misiune de 1000000 h rezultă o probabilitatea de bună funcționare: $R_s(1000000) = \exp(-1,3) = 0,2725$.

Pentru un timp de misiune mai mic, de exemplu 100000 h, rezultă o valoare mult mai mare, respectiv: $R_s(100000) = \exp(-0,13) = 0,8781$.

Concluzia este evidentă: probabilitatea de bună funcționare a unui sistem este cu atât mai mare cu cât durata de funcționare este mai mică.

În mod evident probabilitatea de funcționare, la momentul inițial, este maximă, $R_s(0) = 1$ și după un timp de funcționare suficient de mare (specific fiecărui sistem) probabilitatea de bună funcționare tinde spre 0, $R_s(\infty) = 0$.

Pentru un sistem complex, fiabilitatea sa scade cu cât sistemul este mai complex, valoarea indicatorului λ_s crește cu atât mai mult cu cât termenii sumei constitutive (relația 4.3.1) sunt în număr mai mare.

Din relația 4.3.2 se observă că fiabilitatea sistemului este mai mică decât fiabilitatea oricăreia din componentele sale. Dacă o componentă are o fiabilitate mult inferioară celorlalte, aceasta determină fiabilitatea sistemului și reprezintă *veriga cea mai slabă*. Este, deci, contraindicată realizarea unui sistem de tip serie cu elemente neomogene din punct de vedere al fiabilității.

4.3.2. Sisteme de tip paralel

Sistemele de tip paralel sunt caracterizate prin faptul că defectarea uneia dintre componentele sistemului nu provoacă defectarea sistemului, la defectarea unei componente intrând în funcțiune componenta legată în paralel cu cea defectată, așa numita componentă de rezervă, redundantă.

Rezervele se diferențiază în funcție de durata de conectare în sistem atunci când se defectează elementul de bază și de starea în care se află elementul de rezervă în perioada de așteptare, astfel:

- *rezervă activă* - elementul rezervă și cel de bază sunt solicitate la fel în perioada de funcționare, iar conectarea rezervei active se face imediat ce elementul din sistem s-a defectat, în timp practic egal cu zero;

- *rezervă semiactivă* - elementul de rezervă este mai puțin solicitat decât elementul de bază, iar timpul de conectare al rezervei este mic, dar nu zero;

- *rezervă pasivă* - elementul de rezervă nu este pregătit pentru înlocuirea imediată a elementului de bază, solicitarea acestora este neglijabilă în perioada de rezervă.

Cazul cel mai favorabil de redundanță, din punct de vedere al funcționării sistemului (nu și al costurilor) este cel în care un element este dublat de un altul identic, aflat în rezervă activă.

În cazul general se consideră un sistem format din n elemente legate în paralel, cu rate de defectare λ_i (pentru componenta i), μ_i - rata de reparare a componentei i și $R_i(t)$ - funcția de fiabilitate, respectiv probabilitatea de funcționare a componentei i . Funcția de structură a acestui sistem este tipul *SAU logic*:

$$S = x_1 \cup x_2 \cup x_3 \dots \cup x_n. \quad (4.3.3)$$

Reprezentarea unui sistem format din n elemente conectate în paralel este în figura 4.3.

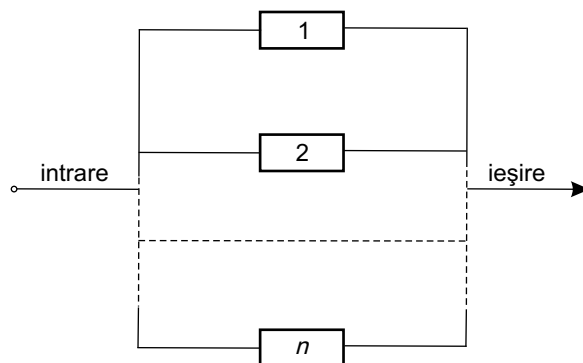


Figura 4.3. Sistem de tip paralel

Analiza fiabilității sistemului nu este imediată ca în cazul sistemului de tip serie. Pentru sistem de tip paralel se consideră funcția de stare S negată, respectiv starea de defectare a sistemului:

$$\bar{S} = \overline{x_1 \cup x_2 \cup \dots \cup x_n} = \bar{x}_1 \cap \bar{x}_2 \cap \dots \cap \bar{x}_n. \quad (4.3.4)$$

Probabilitatea de defectare a sistemului este egală cu produsul probabilităților de defectare ale elementelor sistemului:

$$F_s = P(\bar{S} = 1) = \prod_{i=1}^n P(\bar{x}_i = 1) = \prod_{i=1}^n F_i, \quad (4.3.5)$$

unde F_i este probabilitatea de defectare a elementului i .

Așa cum se cunoaște deja, probabilitatea de bună funcționare a unui sistem este complementara funcției de defectare a acestuia. Deci probabilitatea de bună funcționare a unui sistem este cu n elemente legate în paralel, R_n , este complementul funcției de defectare a sistemului. Se poate scrie:

$$R_p(t) = 1 - F_s = 1 - \prod_{i=1}^n (1 - R_i(t)), \quad (4.3.6)$$

unde R_i este probabilitatea de bună funcționare a elementului component, i .

O astfel de schemă se mai numește și **schemă redundantă**, fiind folosită în situațiile, în care un element al schemei este rezervat prin alt element identic.

Exemplu de aplicare : Fie o schemă redundantă cu două elemente în paralel: elementul 1 cu fiabilitatea $R_1(t) = 0,9$, care, la defectare, este înlocuit de rezerva sa, elementul 2, identic cu primul, deci cu aceeași valoare a funcției de fiabilitate $R_2(t) = 0,9$.

Fiabilitatea ansamblului celor două elemente este:

$$R_p(t) = 1 - (1 - R_1) \cdot (1 - R_2) =$$

$$= 1 - (1 - 0,9) \cdot (1 - 0,9) = 0,99 > 0,9. \quad (4.3.7)$$

Se constată că fiabilitatea sistemului paralel este mai bună decât fiabilitatea elementelor sale, ceea ce face ca un astfel de sistem să fie preferat unuia serie.

4.4 Arbori de evenimente

4.4.1. Concepte de bază referitor la arbori de evenimente

Studiul fiabilității sistemelor prin metoda arborilor de defectare este o metodă deductivă de analiză care se desfășoară de sus în jos și care permite identifica cauzelor care pot duce la evenimentul de vârf definit. Un arbore de defectare este o reprezentare grafică organizată a unor condiții sau factori care cauzează sau contribuie la apariția unei defectări a sistemului, denumită „eveniment de vârf”.

Evenimentul de vârf este o consecință a combinațiilor tuturor evenimentelor de intrare [1, 50]. Este numit și **eveniment final** sau **consecință de vârf**

Reprezentarea unui arbore de defect se face sub o formă care poate fi ușor înțeleasă și analizată pentru a permite identificarea:

- factorilor care afectează evenimentul de vârf considerat;
- factorilor care afectează caracteristicile de fiabilitate și performanță ale unui sistem, de exemplu, deficiențe în proiectare, solicitări de mediu sau de funcționare, moduri de defectare a componentelor, greșeli ale operatorilor, defecte ale pachetelor software;
- evenimentelor care afectează funcționarea mai multor componente, anulând beneficiile unor redundanțelor sau a unor părți ale unui sistem.

În construcția arborelui de defectare se pornește de la evenimentul de vârf și se lucrează cu următoarele elemente:

- **poartă logică** - simbol care este folosit pentru a stabili legături simbolice între evenimentul de ieșire și intrările corespunzătoare; reflectă tipul de relație logică (booleeană) între evenimentele de intrare pentru ca evenimentul de ieșire să se poată produce.

- **eveniment**- apariția unei condiții sau o acțiune care duce la defectarea sistemului;

- **eveniment primar** - eveniment care stă la baza arborelui de defect; poate fi un eveniment care nu mai poate fi dezvoltat în arborele analizat sau un eveniment care a fost dezvoltat în altă parte pe baza unui grup de evenimente și porți și care este introdus ca eveniment deja studiat;

- **defectare prin eveniment unic** - eveniment de defectare care poate cauza defectarea generală a sistemului sau care, independent de alte evenimente sau de combinațiile acestora, poate cauza evenimentul de vârf

- **cauză comună** - cauză de apariție a mai multor evenimente;

- **eveniment repetat** eveniment care este o intrare pentru mai multe evenimente de nivel superior.

Atunci când evenimentul studiat este defectarea, Metoda arborilor de evenimente devine Metoda arborilor de defectare, sau de defecte.

Metoda arborilor de defectare este adecvată pentru analiza sistemelor care cuprind mai multe subsisteme funcționale sau dependente. Este aplicată în mod uzual la proiectarea de centrale de energie nucleară, sisteme de transport, sisteme de comunicație, procese chimice sau industriale, sisteme de cale ferată, sisteme medicale, și nu în ultimul rând a sistemelor informatice

Arborii de defectare permit atât analize calitative cât și cantitative.

Scopul primar al analizei *calitative* este identificarea setului de tăieturi minimal pentru a determina modul în care evenimentele de bază influențează evenimentul de vârf.

Analiza *cantitativă* poate fi utilizată pentru calculul probabilităților de apariție a evenimentului de vârf și evenimentelor intermediare atunci când sunt cunoscute probabilitățile evenimentelor primare.

O analiză bazată pe arbori de defectare are ca obiective:

- identificarea cauzelor sau a combinațiilor acestor cauze care duc la evenimentul de vârf;

- determinarea modului în care o caracteristică de fiabilitate a unui sistem particular îndeplinește o cerință specificată;

- determinarea modurilor sau factorilor potențiali de defectare care contribuie cel mai mult la probabilitatea de defectare sau indisponibilitatea sistemului reparabil, pentru a identifica îmbunătățirile posibile ce pot fi aduse fiabilității unui sistem;

- analiza și compararea diverselor alternative de proiectare pentru a îmbunătăți fiabilitatea sistemului;

- demonstrarea valabilității ipotezelor făcute în alte analize (de exemplu lanțuri Markov și FMEA);

- identificarea modurilor potențiale de defectare care pot cauza o problemă de securitate, evaluarea probabilității corespunzătoare de apariție a evenimentelor de securitate și a posibilității de reducere;

- identificarea evenimentelor comune;

- căutarea unui eveniment sau a unei combinații de evenimente care sunt cauza cea mai probabilă a apariției evenimentului de vârf;

- evaluarea impactului apariției unui eveniment primar asupra probabilității evenimentului de vârf;

- calculul probabilităților evenimentului;

- calculul disponibilităților și al ratelor de defectare ale sistemului sau componentelor sale reprezentate în arborele de defect, dacă se poate declara o stare ca fiind stabilă, iar eventualele reparații sunt independente unele de celelalte (aceeași limitare ca și pentru diagrama căii de succes/diagrama-bloc de fiabilitate).

Metoda arborilor de defectare folosită în studiul fiabilității sistemelor pornește de la ideea că procesul de defectare poate fi cuantificat la nivel structural, astfel că orice defecțiune a sistemului este rezultatul unei secvențe cuantificate de stări ale procesului de defectare.

În figura 4.4 este dată reprezentarea cea mai simplă a unui arbore de defectare, alcătuit din evenimente primare, interconectate prin intermediul unei structuri logice booleene, care indică posibilitățile, în care evenimentele se pot combina pentru a produce avaria sistemului. Dacă sistemul are mai multe condiții de avarie, pentru fiecare dintre ele trebuie construit un arbore de defectare separat.

Arborele de defectare se construiește pornind de la evenimentul din vârf (defectarea sistemului) până când se ajunge la evenimentele primare (defectarea componentelor sau subsistemelor) studiind interacțiunile logice dintre aceste evenimente ale sistemului [8, 36, 37, 50].

În faza de proiectare, metoda arborilor de defectare permite evidențierea unor deficiențe de concepție, a locurilor și elementelor vulnerabile din sistem.

Din punct de vedere structural, arborelui de defectare i se asociază următoarele concepte:

- elementele primare - componentele sau elementele care se găsesc la nivelul de bază;
- defectări de bază - defectările elementelor primare;
- evenimentul nedorit - starea de defect;
- modul de defectare - setul de elemente defecte simultane, care conduc la defectarea sistemului;
- modul minim de defectare - cel mai mic set de componente primare, care conduc la defectarea sistemului;
- nivelul ierarhic - totalitatea elementelor care sunt echivalente structural, care ocupă poziții echivalente în alcătuirea arborelui de defectare.

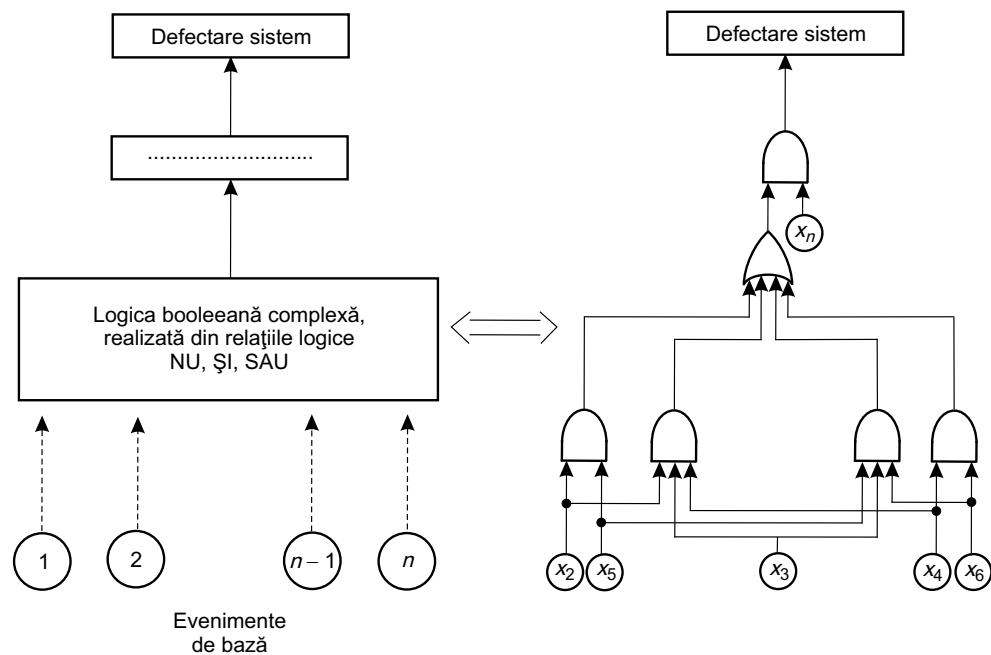


Figura 4.4. Reprezentarea simplă a unui arbore de defecte

Consecința finală a unui arbore de defectare (eveniment de vârf) poate fi o defectare în sine sau un eveniment. Aici, arborele de defectare descrie un defect sau un eveniment care rezultă din evenimentele contribuitoare sau din alte defecte. În analiza arborelui de defectare anumite combinații de evenimente pot fi stări sau evenimente, în timp ce altele trebuie să se potrivească consecinței.

4.4.2. Descrierea și structura grafică a arborelui de defectare

Componentele unui arbore de defectare sunt următoarele:

Porți logice – Simboluri care prezintă relația logică dintre evenimentele de intrare și evenimentul de ieșire. Ele pot fi statice sau dinamice.

- porți statice – consecința nu depinde de ordinea de apariție a intrărilor,
- porți dinamice – consecința depinde de ordinea de apariție a intrărilor.

Evenimente:

- Cel mai mic nivel de intrări într-un arbore de defect.

Componentele grafice ale unui arbore de defect sunt:

- a) simboluri logice ale porților arborelui de defectare;
- b) linii de conectare a intrărilor în poartă;
- c) descrieri ale evenimentelor intermediare;
- d) simboluri de transfer de intrare sau de ieșire;
- e) simboluri ale evenimentelor primare.

Toate evenimentele relevante ar trebui să fie incluse în arborele de defectare. Asemenea evenimente ar trebui să includă efectele condițiilor de mediului și ale altor solicitări la care poate fi supus elementul, inclusiv software-ul, comenzile și monitorizarea stărilor; acelea care sunt posibile în timpul funcționării, chiar și în afara specificațiilor proiectului.

Ar trebui menționate în raport chiar și evenimentele pe care analistul le-a considerat inițial, dar au fost excluse din analiza ulterioară pentru că nu puteau fi aplicate și care nu au fost incluse în arborele final de defect.

Dacă arborele de defectare atenționează asupra a două sau mai multe probleme de performanță a sistemului cauzate de un defect existent, atunci evenimentul care descrie defectul trebuie inclus în arborele de defect în mai multe locuri și marcat ca eveniment comun. În analiza cantitativă evenimentul comun este inclus în calcule numai o dată, dar ar trebui să fie aplicate toate criteriile de disjuncție.

Pentru a evita includerea accidentală a evenimentelor comune în calcule multiple, trebuie stabilită și utilizată etichetarea convențională a evenimentelor. Acest tip de etichetare trebuie să fie consecvent. Dacă este utilizat pachetul software pentru asistarea evaluării arborelui de defect, trebuie utilizate convenții și setări adecvate.

Atunci când sunt creați arborii de defect, ei pot fi prezentați în formă verticală, de sus în jos sau în formă orizontală de la stânga la dreapta. Când arborele de defectare este reprezentat pe orizontală, toate simbolurile prezentate sunt rotite la 90° în sensul invers al acelor de ceasornic.

Arborii de defectare pot fi de asemenea citați sau studiați în direcții opuse, de exemplu, în tratarea accidentelor și defectărilor produse etc.

De exemplu, într-o poartă SAU unde consecința este o stare sau un eveniment, intrările pot fi stări sau evenimente. Toate intrările într-o poartă ȘI în care consecința este un eveniment trebuie să fie evenimente, iar dacă consecința este o stare, toate intrările trebuie să fie stări.

Starea poate fi caracterizată prin probabilitatea existenței sale la momentul t , iar evenimentul poate fi caracterizat, fie prin rata sau frecvența de defectare, fie prin probabilitatea de apariție a evenimentului la momentul t .

Metoda are la bază **logica booleană** unde cele două valori sunt **Defect (D)**, corespunzând lui **0** și **Funcțional (F)**, corespunzând lui **1**. După cum se cunoaște, în sistemul binar din n variabile, se pot forma 2^n combinații binare.

Pentru orice aplicație practică, orice funcție logică poate fi obținută prin folosirea numai a celor trei *funcții logice de bază* – funcții fundamentale ale algebrei booleene [2, 50]: **funcția negație (NU)**, **produsul logic (ȘI)** și **suma logică (SAU)**. Așa cum spun matematicienii aceste trei funcții logice definesc un *sistem complet*.

Pentru fiecare eveniment care apare în arborele de defectare se recomandă să se realizeze o listă cu numele sau descrierea evenimentului, să se codifice evenimentele și să se calculeze probabilitatea de apariție.

Dacă un eveniment reprezintă un eveniment repetat sau din cauză comună, el este prezentat în arborele de defect în mod repetat, dar cu un steguleț de atenționare. Toate evenimentele repetate sau din cauză comună din arbore trebuie să aibă același cod și trebuie marcate cu un simbol transfer-intrare sau cu un alt simbol ales special pentru un anumit arbore de defectare. Această regulă se aplică tuturor evenimentelor repetate sau cu cauză comună cu excepția evenimentului de cel mai mic nivel din ansamblu, care este marcat cu un simbol transfer-ieșire. În unele diagrame ale arborilor de defect, simbolurile pentru evenimentele primare repetate sau de nivel mai mare sunt aceleași.

Dacă un eveniment a fost dezvoltat într-o altă parte sau pagină a arborelui de defect trebuie indicat acest fapt cu o poartă de transfer, de exemplu o poartă ȘI PRIORITAR.

4.4.3 Evaluarea fiabilității sistemului utilizând arbori de defectare

O evaluare rapidă, dar aproximativă a fiabilității sistemelor se poate face utilizând proprietățile porților în reprezentarea arborilor de defectare, făcând următoarele aproximații:

- evenimentele de bază sunt independente;
- evenimentele de bază sunt rare, probabilitatea de apariție este mai mică de 0,1%.

Evaluarea probabilității de defectare folosește proprietățile porților logice.

- ieșirea porților **NU**:

ieșirea = probabilitatea (A să **NU** fie defect):

$$P(\bar{A}) = 1 - P(A). \quad (4.4.1)$$

- ieșirea porților **ȘI** :

Ieșirea este probabilitatea (A defect **ȘI** B defect):

$$P(A \cap B) = P(A / B) \cdot P(B) = P(B / A) \cdot P(A). \quad (4.4.2)$$

Dacă evenimentele A și B sunt independente (nu se condiționează reciproc), se obține:

$$P(A \cap B) = P(A) \cdot P(B). \quad (4.4.3)$$

- ieșirea porților **SAU**

Ieșirea este probabilitatea (A defect **SAU** B defect)

$$P(A \cup B) = P(A) + P(B) - P(A \cap B). \quad (4.4.4)$$

Dacă evenimentele A și B sunt independente se obține:

$$P(A \cup B) = P(A) + P(B) - P(A) \cdot P(B). \quad (4.4.5)$$

Evaluarea ratei de defectare se face pe baza unor ipoteze similare.

- ieșirea porților **SAU** :

Considerăm că cele două evenimente sunt independente și la ieșire urmărim rata de defectare echivalentă λ_s .

Notând cu $P(A) = F_A(t)$ - probabilitatea ca A să se defecteze în intervalul $(0, t)$ și cu $P(B) = F_B(t)$ probabilitatea ca B să se defecteze în intervalul $(0, t)$. Probabilitatea ca un sistem cu o poartă logică SAU să se defecteze în intervalul $(0, t)$ se calculează cu relația:

$$\begin{aligned} F_s(t) &= P(A \cup B) = P(A) + P(B) - P(A) \cdot P(B) = \\ &= F_A(t) + F_B(t) - F_A(t) \cdot F_B(t). \end{aligned} \quad (4.4.6)$$

Probabilitatea de bună funcționare în intervalul $(0, t)$ a sistemului considerat, $R_s(t)$, este:

$$\begin{aligned} R_s(t) &= 1 - F_s(t) = 1 - F_A(t) - F_B(t) + F_A(t) \cdot F_B(t) = \\ &= [1 - F_A(t)] \cdot [1 - F_B(t)] = R_A(t) \cdot R_B(t). \end{aligned} \quad (4.4.7)$$

Exemplu:

Dacă:

$$R_A(t) = e^{-\lambda_A t} \text{ și } R_B(t) = e^{-\lambda_B t}, \quad (4.4.8)$$

atunci:

$$R_s(t) = e^{-\lambda_A t} \cdot e^{-\lambda_B t} = e^{-(\lambda_A + \lambda_B)t} = e^{-\lambda_s t}. \quad (4.4.9)$$

Din relația de mai sus rezultă că:

$$\lambda_s = \lambda_A + \lambda_B. \quad (4.4.10)$$

- ieșirea porților **ȘI** :

Considerând n elemente independente la intrarea porții **ȘI**, rata de defectare a sistemului se determină reluând, corespunzător, raționamentul de la poarta SAU, și rezultă:

$$\lambda_s = \frac{\sum_{i=1}^N \lambda_i (\alpha_i - 1)}{\prod_{i=1}^N \alpha_i - 1}, \quad (4.4.11)$$

unde

$$\alpha_i = \frac{1}{1 - \exp(-\lambda_i t)}, \quad \forall i \in [1, N]. \quad (4.4.12)$$

În cazul logicii paralel (redundanță), cu N elemente identice

$$\lambda_s = \frac{N \lambda}{\sum_{i=0}^{N-1} \alpha_i}, \quad (4.4.13)$$

unde $\alpha_i = \frac{1}{1 - e^{-\lambda_i t}}$. (4.4.13')

Pentru cele mai multe sisteme, o analiză a fiabilității sistemului, ținând seama de buna funcționare a tuturor elementelor componente conduce la realizarea *grafului de arborescență* [50].

Graful de arborescență este un graf finit cu următoarele proprietăți:

- graful nu conține bucle sau cicluri orientate;
- există un singur vârf, numit rădăcina arborescenței, care nu reprezintă extremitatea terminală a nici unui arc;
- oricare din vârfuri constituie extremitatea a câte unui singur arc;
- vârfurile, care nu sunt extremități inițiale ale unor arce, sunt vârfuri suspendate.

Figura 4.5 ilustrează o arborescență, rădăcina **P** fiind sistemul tehnic a cărui structură a fost reprezentată. Prin **S** s-au simbolizat subsistemele componente, indicii precizând apartenența la structură, nivel ierarhic și număr de ordine.

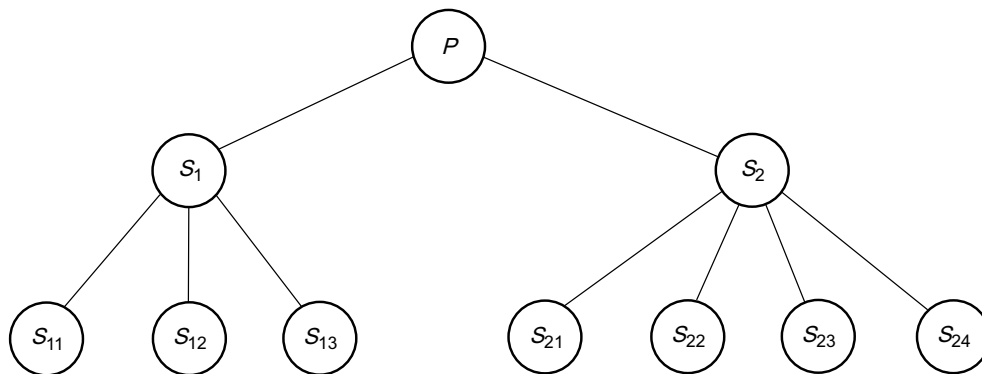


Figura 4.5. Sistem tehnic analizat ca graf de arborescență

Arborescența poate fi tratată și utilizând sistemul binar de codificare sau matrici booleene.

Un anumit produs complex, în procesul studierii fiabilității, se poate descompune în subsisteme de diferite niveluri (subansamble, blocuri, elemente etc.). În funcție de scopul analizei, fiecare asemenea ansamblu (la orice nivel) poate fi considerat ca un întreg, care se supune cercetării de sine stătător. Deci orice subsistem, sau chiar sistemul în ansamblul său, poate fi considerat ca obiect de studiu al fiabilității, care se realizează după aceeași metodologie ca și studiul unei componente elementare.

Exemplu: Fie un sistem de calcul tolerant la defecte compus din două unități hardware identice (H_1 și H_2), care execută aceeași versiune de program (V).

Înainte de afișarea rezultatului, ieșirea este supusă unui modul de decizie (D), care constă într-un test de acceptare. Dacă în rezultatul, oferit de prima unitate hardware, se detectează o eroare, rezultatul va fi cel oferit de cea de-a doua unitate.

Elementele de bază reprezintă defectările componentelor hard și soft.

Fiecare defectare software se poate datora fie (logică SAU):

- versiunii greșite (V),
- deciziei greșite (D),
- defectării software, datorită unei specificații incomplete (G).

Fiecare defectare hardware se poate datora defectării ambelor unități hard (H) (logică ȘI).

Arborele de defectare corespunzător acestui sistem este dat în figura 4.6.

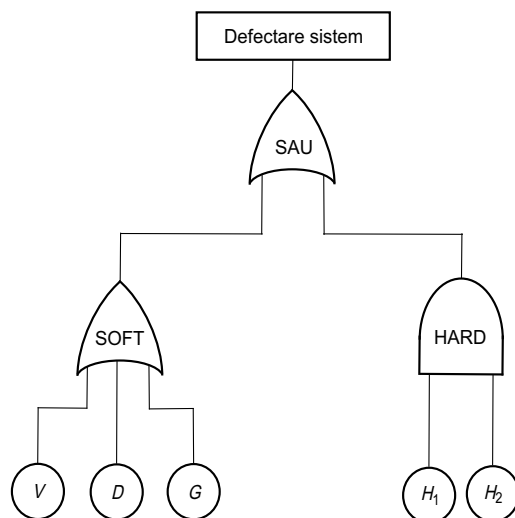
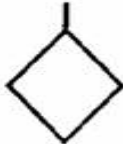
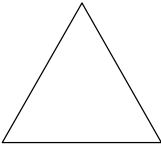



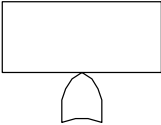
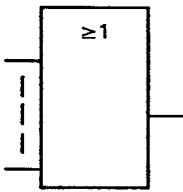
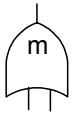
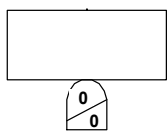
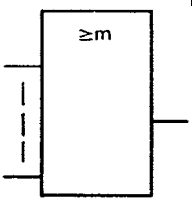
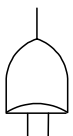
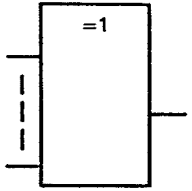
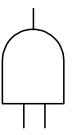
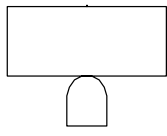
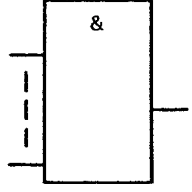
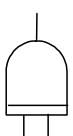
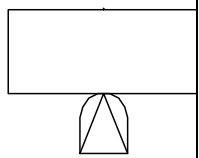
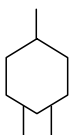
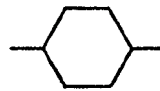
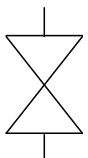
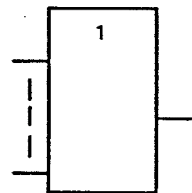


Figura 4.6. Arbore de defectare pentru un sistem de calcul tolerant la defect

Tabelul 4.1 – Simboluri utilizate frecvent pentru un arbore de defectare

Simbol			Nume	Descriere	Corelația cu fiabilitatea	Numărul de intrări
			EVENIMENT DE BAZĂ	Evenimentul de cel mai mic nivel pentru care sunt disponibile informații referitoare la probabilitatea de apariție sau fiabilitatea sa	Mod de defectare a componentei sau cauza modului de defectare	0
			EVENIMENT CONDIȚIONAT	Eveniment care este o condiție de producere a unui alt eveniment,	Eveniment care se produce pentru ca un alt eveniment să se	0

Simbol		Nume	Descriere	Corelația cu fiabilitatea	Numărul de intrări	
				atunci când pentru a apărea o ieșire trebuie să aibă loc ambele evenimente	producă Probabilitate condiționată	
			EVENIMENT „ÎN AȘTEPTARE”	Un eveniment primar care reprezintă o defectare „în adormire”; un eveniment care nu este detectat imediat dar ar putea să fie detectat printr-o inspecție sau analiză suplimentară	Mod de defectare al unei componente inactive sau cauză a defectării „în așteptare”	0
			EVENIMENT NEDEZVOLTAT	Un eveniment primar care reprezintă o parte a unui sistem care nu este încă dezvoltat	Un contribuitor la probabilitatea de defectare. Structura acestei părți din sistem nu este încă definită	0
		 Transfer IEȘIRE  Transfer INTRARE 	POARTĂ DE TRANSFER	Poartă care indică evoluția acestei părți a sistemului în altă parte sau pagină a diagramei	O diagramă parte a arborelui de defectare prezentată în altă parte a sistemului total; INTRARE înseamnă că poarta de dezvoltare se află în altă parte. IEȘIRE înseamnă că aceeași poartă dezvoltată în acest loc va fi utilizată în altă parte	0
			Poartă OR (SAU)	Evenimentul de ieșire apare dacă apare orice eveniment la intrare	Defectarea sistemului se produce dacă se defectează oricare componentă a sistemului – sistem serie	≥ 2

Simbol			Nume	Descriere	Corelația cu fiabilitatea	Numărul de intrări
			Poartă VOT MAJORITY	Evenimentul de ieșire apare dacă evenimentul de intrare apare la m sau mai multe intrări din totalul de n intrări	Redundanță k din n , unde $m = n - k + 1$	≥ 3
			Poartă OR EXCLUSIVE	Evenimentul de ieșire apare dacă evenimentul de intrare apare la o singură intrare	Defectarea sistemului apare dacă se defectează o singură componentă a sistemului	≥ 2
			Poartă AND (ȘI)	Evenimentul de ieșire apare numai dacă evenimentul de intrare apare la toate intrările	Redundanță paralel	≥ 2
			Poartă PRIORITY AND (PAND)	Evenimentul de ieșire (defectarea) apare numai dacă evenimentele de intrare apar într-o anumită secvență, de la stânga la dreapta	Bună pentru reprezentarea defectelor secundare sau pentru secvențe de evenimente	≥ 2
			Poartă INHIBIT	Evenimentul de ieșire apare dacă apar ambele evenimente la intrare, unul dintre ele fiind condiționat	Evenimentului final are o probabilitate de apariție condiționată	2
			Poartă NOT	Evenimentul de ieșire apare dacă nu apare evenimentul de intrare	Eveniment exclusiv sau măsură preventivă pentru ca evenimentul să nu apară	1

Capitolul 5. METODA LANȚURILOR MARKOV PENTRU FIABILITATEA SISTEMELOR

5.1. Definirea lanțului Markov

Din cele prezentate până acum s-a văzut că pentru sistemele cu restabilire timpiei de funcționare până la defectare, respectiv până la reparare au o distribuție statistică, aleatoare. Starea sistemului la un moment dat, t , poate fi considerată o variabilă aleatoare $\{x(t) \in S, t \in R^+\}$, unde S este spațiul stărilor sistemului.

Analiza Markov este folosită atunci când se presupune că starea viitoare a sistemului depinde numai de starea prezentă, nu și de cea din trecut, presupune că

Lanțul Markov (Markov Chain) este un proces probabilistic $\{x(t)\}$, care prezintă proprietatea lui Markov și anume, faptul că starea curentă a sistemului captează întregul istoric al acestuia, iar starea lui viitoare va depinde numai de starea lui prezentă [1,8, 12, 26]:

$$\begin{aligned} P\{X(t) = x / X(t_n) = x_n, X(t_{n-1}) = x_{n-1}, X(t_0) = x_0\} = \\ = \{X(t) = x / X(t_n) = x_n\}, \end{aligned} \quad (5.1.1)$$

unde: $t > t_n > t_{n-1}$.

Trecerea sistemului dintr-o stare i într-o stare j se numește *tranziție*. Fiecărei tranziții i se asociază o *probabilitate de tranziție* (intensitate de tranziție) pentru a desemna probabilitatea ca sistemul să fie în starea j la momentul $t + \Delta t$, condiționată de faptul că a fost în starea i la momentul θ :

$$p_{ij}(t, \theta) = P\{X(t) = j / X(\theta) = i\}. \quad (5.1.2)$$

Din definiție rezultă că estimarea stării viitoare a sistemului este complet determinată (în sens probabilistic) de cunoașterea stării lui prezente.

Analiza Markov este o tehnică cantitativă și poate fi distinctă (utilizând probabilități de schimbare între stări) sau continuă (utilizând rate de schimb între stări). Deși analiza Markov poate fi efectuată și de mână, natura tehnicilor o face adecvată utilizării pe programe informatice, multe existând în mod curent pe piață.

Ipotezele care stau la baza studiului fiabilității unui sistem, utilizând lanțuri Markov sunt:

1. fiabilitatea sistemului se poate estima în funcție de fiabilitatea tuturor elementelor sale;
2. perioada de timp în care este analizat sistemul este cea de maturitate, caracterizată prin $\lambda = \text{constant}$;
3. tranzițiile dintr-o stare în alta se pot produce în orice moment;
1. defectarea sau repararea unui element al sistemului este independentă de starea celorlalte elemente;

5. defectarea unui element al sistemului este un eveniment, a cărui probabilitate de realizare într-un interval de timp, Δt , este $\lambda \cdot \Delta t$, probabilitate care depinde numai de mărimea intervalului, nu și de timpul anterior de funcționare;

6. repararea unui element al sistemului este un eveniment a cărui probabilitate de realizare într-un interval de timp Δt este $\mu \cdot \Delta t$;

7. probabilitatea defectării și reparării unui element al sistemului în intervalul Δt este 0 ($\lambda \Delta t \cdot \mu \Delta t = 0$) adică în acest interval poate să aibă loc o singură tranziție, fie defectare, fie reparare.

Considerând că în mod sigur (cu certitudine) într-un interval de timp foarte mic, dt , sistemul nu poate decât să-și păstreze starea i de la momentul inițial sau să tranziteze într-o stare j , se poate scrie:

$$\sum_{i=1}^n p_{ij} = 1 \quad \text{sau} \quad \sum_{j=1}^n q_{ij} = 1. \quad (5.1.3)$$

Lanțul Markov este omogen, în raport cu timpul, dacă probabilitatea de trecere nu depinde de valoarea inițială a timpului de observare, ci numai de durata tranziției (dt) și că $\forall i, j \in S, \exists P_{ij} \in R^+$, astfel încât:

$$P_{ij}(t_{n+1} - t_n) = P\{X(t_{n+1}) = i / X(t_n) = j\}. \quad (5.1.4)$$

Evenimentele $\{X(t) = j\}, j \in S$, formând un sistem complet, se poate scrie:

$$P\{x(t+dt) = i\} = \sum_{j \in S} P\{x(t+dt) = i / x(t) = j\} \cdot P\{x(t) = j\}. \quad (5.1.5)$$

Introducând notația:

$$P_i(t) = P\{x(t) = i\}, \forall t \in R^+, \forall i \in S, \quad (5.1.6)$$

relația (5.1.4) se poate scrie:

$$\begin{aligned} P_i(t+dt) &= \sum_{j \in S - \{i\}} P_j(t) a_{ij} dt + P\{X(t+dt) = i | x(t) = j\} \cdot P_i(t) = \\ &= \sum_{j \in S - \{i\}} P_j(t) a_{ij} dt + \left[1 - \sum_{h \in S - \{i\}} a_{hi} dt \right] P_i(t). \end{aligned} \quad (5.1.7)$$

unde

$$a_{ij} = \lim_{\Delta t \rightarrow 0} \frac{p_{ij}(\Delta t)}{\Delta t} \quad (5.1.7')$$

reprezintă rata de tranziție din starea i în starea j cu $i \neq j$, la momentul $t + dt$.

În mod similar

$$a_{ii} = \lim_{\Delta t \rightarrow 0} \frac{p_{ii}(\Delta t) - 1}{\Delta t} \quad (5.1.7'')$$

este rata de tranziție din starea i în starea i (de fapt menținerea aceleiași stări i).

Ținând seama că

$$\sum_{j \in S} p_{ij} = 1, \quad (5.1.8)$$

se poate scrie:

$$a_{ij} = -\sum_{\substack{j \in S \\ j \neq i}} a_{ji}. \quad (5.1.9)$$

Relația 5.1.9 devine:

$$P_i(t + dt) = \sum_{\substack{j \in S \\ j \neq i}} P_j(t) dt + (1 + a_{ij}) dt \cdot P_i(t). \quad (5.1.10)$$

Pentru $dt \rightarrow 0$ ecuația 5.1.10 devine o ecuație diferențială, de fapt un sistem de ecuații diferențiale matriciale:

$$\frac{dP_i(t)}{dt} = \sum_{j \in S} a_{ij} P_j(t). \quad (5.1.10')$$

Soluția acestui sistem este dificil de obținut pentru cazul general. În caz staționar, când $t \rightarrow \infty$ (pentru un timp de observare suficient de mare), $\lim_{t \rightarrow \infty} \frac{dP_i}{dt} = 0$, rezolvarea sistemului este mai simplă. Dacă se adaugă și condiția de normare, sistemul de ecuații (5.1.10') se reduce la un sistem de ecuații liniare:

$$\begin{cases} \sum_{j \in S} a_{ij} P_j = 0, \\ \sum_{j \in S} P_j(t) = 1. \end{cases} \quad (5.1.11)$$

Lanțul Markov admite soluție staționară care este unică dacă este un lanț ireductibil. Aceasta înseamnă că mulțimea stărilor S formează o clasă închisă, care nu poate fi părăsită, adică, fiecare stare poate fi atinsă, dacă se pornește din oricare altă stare, dar numai din mulțimea stărilor sistemului.

5.2. Matricea stărilor de tranziție

Sistemul de ecuații diferențiale (5.1.10') se poate scrie și sub forma matriceală astfel:

$$\left[\frac{dP}{dt} \right] = [A(t)] \cdot [P(t)], \quad (5.2.1.)$$

unde:

- $[A(t)] = (a_{ij})$ este matricea Markov, matricea ratelor de tranziție;
- $[P(t)]$ este vectorul coloană al probabilității de stare ale sistemului;
- $\left[\frac{dP}{dt} \right]$ este vectorul coloană al derivatei vectorului probabilității de stare.

Matricea $[A(t)]$ este o matrice pătrată de dimensiune n , cu $i, j \in S$, unde termenii diagonalei (a_{ii}) reprezintă suma cu semn schimbat a tranzițiilor care

pornesc din starea i , iar termenii nediagonali (a_{ij}), cu $i \neq j$, reprezintă intensitatea tranzițiilor din starea desemnată de numărul coloanei în cea desemnată de numărul liniei.

Pentru că $a_{ii} = -\sum_{\substack{k \in S \\ k \neq i}} a_{ki}$, rezultă că suma termenilor fiecărei linii a matricii

Markov este nulă.

Soluția ecuației matriciale (5.1.12) este o matrice de forma [1, 8, 26]:

$$[P(t)] = [P(0)] \cdot e^{[A^T]t}, \quad (5.2.2)$$

unde:

- $[P(0)]$ este matricea stărilor inițiale;
- $[A^T]$ este matricea transpusă a ratelor de tranziție.

Această formă a soluției este elegantă, dar soluția este dificil de evaluat, fiind nevoie de scrierea matricii $[A^T]$.

Pentru ca matricea (5.2.2) să exprime soluția sistemului (5.2.1), este necesară diagonalizarea matricii transpuse a ratelor de tranziție.

Dacă valorile proprii ale matricii $[A^T]$ sunt distincte și ea este diagonalizată (are n vectori proprii independenți), atunci se poate scrie:

$$[A^T] = [V] \cdot [D] \cdot [V^{-1}], \quad (5.2.3)$$

unde:

- matricea D este o matrice diagonală de forma:

$$D = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sigma_n \end{bmatrix}, \quad (5.2.4)$$

iar matricea D^k este de forma:

$$D^k = \begin{bmatrix} \sigma_1^k & 0 & \dots & 0 \\ 0 & \sigma_2^k & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sigma_n^k \end{bmatrix}. \quad (5.2.5)$$

Valorile proprii $\sigma_1, \dots, \sigma_n$ se obțin ca soluții ale ecuației:

$$\det([A^T] - \sigma[T]) = 0. \quad (5.2.6)$$

- $[V]$ reprezintă matricea formată din vectorii proprii $[V_1], [V_2], \dots, [V_n]$ ai matricii $[A^T]$, adică:

$$[V] = [[V_1][V_2] \dots [V_n]], \quad (5.2.7)$$

unde vectorul propriu $[V_i]$ este o matrice coloană, obținut ca soluție a ecuației:

$$[A^T] - \sigma[I] \cdot [V_i] = 0. \quad (5.1.19)$$

Matricea $e^{[D]t}$ se poate scrie, prin dezvoltare în serie Taylor în jurul lui $t = 0$, sub forma:

$$e^{[D]t} = [I] + [D]t + \frac{[D^2]t^2}{2!} + \frac{[D^3]t^3}{3!} + \dots =$$

$$= \begin{bmatrix} 1 + \sigma_1 + \frac{\sigma_1^2 t^2}{2!} + \dots & 0 & \dots & 0 \\ 0 & 1 + \sigma_2 + \frac{\sigma_2^2 t^2}{2!} + \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 + \sigma_n + \frac{\sigma_n^2 t^2}{2!} + \dots \end{bmatrix} =$$

$$= \begin{bmatrix} e^{\sigma_1 t} & 0 & \dots & 0 \\ 0 & e^{\sigma_2 t} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & e^{\sigma_n t} \end{bmatrix}. \quad (5.2.8)$$

Cu acestea, soluția (5.2.2) se poate aduce, în final, sub forma elegantă:

$$[P(t)] = [V] \cdot e^{[D]t} \cdot [V^{-1}] \cdot [P(0)]. \quad (5.2.9)$$

Dacă modelul Markov admite soluție staționară, aceasta va fi de forma unei matrici coloană $[P] = [P_i]$ care se obține ca soluție a ecuației:

$$[A^T] \cdot [P] = [0]. \quad (5.2.10)$$

Pentru a ridica nedeterminarea, la ecuațiile (5.1.22) trebuie adăugată condiția ca stările sistemului să formeze un complet de evenimente $\left(\sum_{i \in S} P_i = 1 \right)$.

5.3. Exemplu de calcul

Fie un sistem cu restabilire, cu două stări, $S = \{F, D\}$, unde F corespunde stării funcționale și D - celei de defect. Ratele de defectare și cele de reparare sunt constante, egale cu λ și, respectiv cu μ . Probabilitățile de tranziție din starea de funcționare în cea de defect și invers (din starea de defect în stare de funcționare) sunt $a_{12} = \lambda$ și, respectiv $a_{21} = \mu$. Celelalte probabilități, de menținere în aceeași stare, de funcționare sau de defect sunt: $a_{11} = -\lambda$ și $a_{22} = -\mu$.

Graful de tranziție al stărilor este dat în figura 5.1.

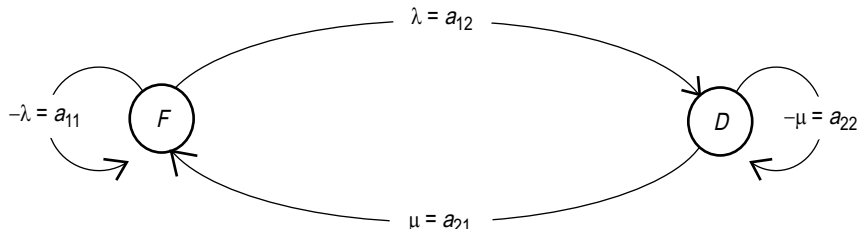


Figura 5.1. Graful de tranziție al stărilor unui sistem simplu cu două stări

Matricea tranzițiilor și transpusa sa sunt:

$$[A] = \begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix}, \quad [A^T] = \begin{bmatrix} -\lambda & \mu \\ \lambda & -\mu \end{bmatrix}. \quad (5.3.1)$$

Ecuția din care se află valorile proprii este:

$$\det([A^T] - \sigma[I]) = \begin{vmatrix} -\lambda - \sigma & \mu \\ \lambda & -\mu - \sigma \end{vmatrix} = 0, \quad (5.3.2)$$

adică

$$(-\lambda - \sigma)(-\mu - \sigma) - \lambda\mu = \sigma(\lambda + \mu + \sigma) = 0. \quad (5.3.3)$$

Din rezolvarea ecuației de mai sus se obține:

$$\sigma_1 = 0 \text{ și} \quad (5.3.4)$$

$$\sigma_2 = -(\lambda + \mu). \quad (5.3.5)$$

Matricea diagonală a valorilor proprii este:

$$[D] = \begin{bmatrix} 0 & 0 \\ 0 & -(\lambda + \mu) \end{bmatrix}. \quad (5.3.6)$$

Se poate scrie imediat și matricea $[e^{[D]t}]$

$$[e^{[D]t}] = \begin{bmatrix} 1 & 0 \\ 0 & e^{-(\lambda + \mu)t} \end{bmatrix}. \quad (5.3.7)$$

Vectorii proprii se obțin în felul următor:

$$\det([A^T] - \sigma_1[I])[V_1] = 0, \quad (5.3.8)$$

$$\begin{bmatrix} -\lambda & \mu \\ \lambda & -\mu \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0, \quad (5.3.9)$$

$$-\lambda x + \mu y = 0 \Rightarrow [V_1] = \begin{bmatrix} 1 \\ \frac{\lambda}{\mu} \end{bmatrix}. \quad (5.3.10)$$

$$\det([A^T] - \sigma_2[I])[V_2] = 0. \quad (5.3.11)$$

$$\begin{bmatrix} \mu & \mu \\ \lambda & \lambda \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0, \quad (5.3.12)$$

$$\begin{cases} \mu x + \mu y = 0, \\ \lambda x + \lambda y = 0, \end{cases} \quad (5.3.13)$$

$$\Rightarrow [V_2] = \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \quad (5.3.14)$$

Matricea vectorilor proprii este:

$$[V] = \begin{bmatrix} 1 & 1 \\ \frac{\lambda}{\mu} & -1 \end{bmatrix}, \quad (5.3.14)$$

iar inversa sa:

$$[V^{-1}] = \begin{bmatrix} \frac{\mu}{\lambda + \mu} & \frac{\mu}{\lambda + \mu} \\ \frac{\lambda}{\lambda + \mu} & -\frac{\mu}{\lambda + \mu} \end{bmatrix}. \quad (5.3.15)$$

Se obține în final, conform (5.1.21), vectorul de stare al sistemului:

$$\begin{aligned} [P(t)] &= \begin{bmatrix} 1 & 1 \\ \frac{\lambda}{\mu} & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{-(\lambda+\mu)t} \end{bmatrix} \begin{bmatrix} \frac{\mu}{\lambda + \mu} & \frac{\mu}{\lambda + \mu} \\ \frac{\lambda}{\lambda + \mu} & -\frac{\mu}{\lambda + \mu} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \\ &= \begin{bmatrix} \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda+\mu)t} \\ \frac{\lambda}{\lambda + \mu} - \frac{\lambda}{\lambda + \mu} e^{-(\lambda+\mu)t} \end{bmatrix}. \end{aligned} \quad (5.3.16)$$

5.4. Etapele aplicării metodei lanțurilor Markov

Etapele aplicării acestei metode pentru evaluarea fiabilității unui sistem sunt:

I. Stabilirea datelor de intrare:

- structura sistemului analizat,
- starea inițială a sistemului,
- ratele de defectare, λ_i , și de reparare, μ_i ale fiecărui element..

II. **Întocmirea tabelului de stări și tranziții posibile**, luând în considerare toate stările prin care pot trece toate elementele sistemului (funcțional - F , defect - D , de rezervă - Rz sau revizie - Rv).

Trecerea sistemului dintr-o stare în alta este determinată de defectarea sau repararea unui singur element al sistemului, dar poate fi determinată și de efectuarea unor operații de căutare, ca urmare a unei revizii programate

(mentenanță preventivă) sau a trecerii sistemului de pe elementul de rezervă pe cel de bază.

În funcție de structura sistemului analizat, pot apare următoarele tranziții:

- trecere de la starea de funcționare la cea de defect;
- trecere de la starea de funcționare la cea de defect și apoi la rezervă;
- trecere de la starea de funcționare la cea de defect, la cea de rezervă și apoi la revizie;
- trecerea de la starea de defect la cea de funcționare, direct sau prin starea de rezervă.

În tabelul 5.2 sunt prezentate stările caracteristice ale unui sistem format din două elemente, care poate avea numai 2 stări, funcțional (F) și defect (D). În figura 5.1 au fost prezentate tranzițiile între aceste stări (funcțional/defect), iar în figura 5.2 sunt prezentate trei stări caracteristice ale unui element: funcțional (F) - defect (D) - rezervă (Rz), cu probabilitățile de tranziție în fiecare stare.

Tabelul 5.2

Nr. stare	F	D	Starea sistemului
0	1, 2	-	F
1	1	2	D
2	2	1	D
3	-	1, 2	D

Tranzițiile marcate cu linie punctată indică faptul că acestea nu apar, dacă elementul de bază nu este disponibil: dacă el a fost reparat este pus imediat în funcțiune, nu este păstrat în rezervă.

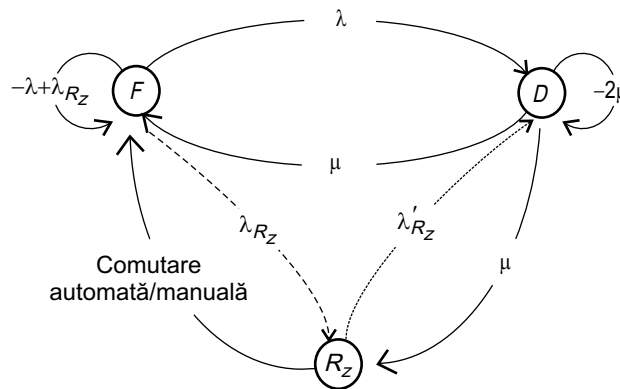


Figura 5.2. Trei stări caracteristice ale unui element de sistem: F, D, Rz

Notațiile din figura de mai sus au următoarea semnificație:

- λ , λ_{Rz} și λ'_{Rz} ratele de defectare ale unui element, care marchează tranzițiile dintre următoarele stări ($F \rightarrow D$), ($F \rightarrow Rz$) și respectiv ($Rz \rightarrow D$);
- μ - rata de reparare a unui element.

III. Trasarea grafului stărilor:

Graful stărilor se formează pe baza tabelului de stări. Acest graf oferă o imagine asupra stărilor și tranzițiilor dintre stări, facilitând scrierea matricei ratelor de tranziție și stabilirea expresiei de calcul a unor indicatori de fiabilitate. Graful stărilor se organizează pe niveluri, un nivel cuprinzând stările, care au același număr de defecte.

Trecerile sistemului dintr-o stare în alta înseamnă o trecere dintr-un nivel în nivelul imediat următor sau anterior, în funcție de evenimentul, care are loc (defectare sau reparare). Între stările aceluiași nivel nu există tranziții.

Fie un sistem format din patru componente, conform schemei din figura 5.3.

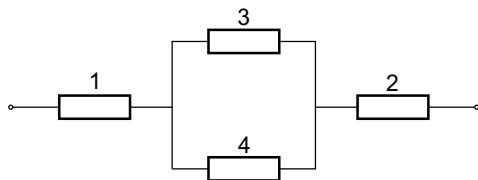


Figura 5.3. Sistemul analizat

Tabelul de stări pentru acest sistem este dat în tabelul 5.5.

Nr. stare	Elemente funcționale	Elemente defecte	Starea sistemului
0	1, 2, 3, 4	-	F
1	2, 3, 4	1	D
2	1, 3, 4	2	D
3	1, 2, 4	3	F
4	1, 2, 3	4	F
5	2, 4	3, 1	D
6	1, 4	3, 2	D
7	1, 2	3, 4	D
8	2, 3	4, 1	D
9	1, 3	4, 2	D

Pentru 3 elemente defecte, oricare ar fi, sistemul este defect (următoarele 7 stări). Graful stărilor tranzițiilor acestui sistem este prezentat în figura 5.4.

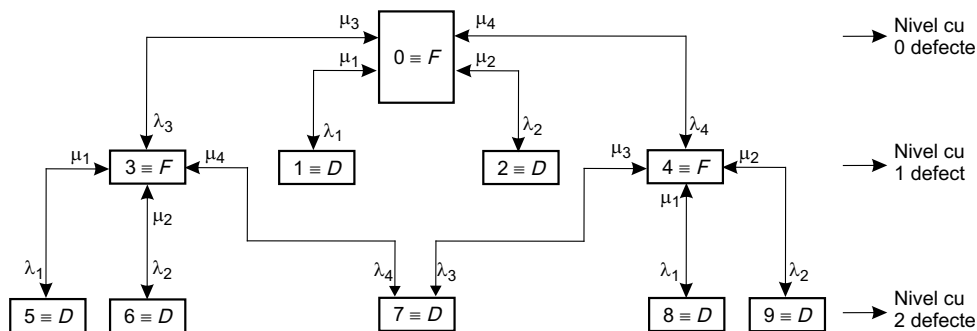


Figura 5.4. Graful stărilor sistemului analizat

Întrucât sistemul este caracterizat de elemente binare ($F - D$ sau $1 - 0$), graful stărilor este simetric.

IV. Scrierea matricei lui Markov $[A(t)] = (a_{ij})$

Matricea tranzițiilor se scrie ușor pe baza grafului stărilor, respectând regulile de formare ale acestora. Corectitudinea ei se verifică urmărind dacă:

- ordinul matricei este egal cu numărul nivelelor stărilor sistemului;
- suma intensităților de tranziție de pe fiecare linie este egală cu 0 (zero).

V. Determinarea probabilităților de ocupare a stărilor

Dacă se urmărește comportarea sistemului pe termen limitat, se vor calcula probabilitățile de ocupare a stărilor în funcție de timp, $P(t)$, din sistemul (5.1.11) sau ecuația (5.1.12), considerând $P_i(0)$ cunoscute.

Dacă se urmărește comportarea sistemului pe o durată îndelungată (putând aproxima $t \rightarrow \infty$), se vor calcula probabilitățile absolute de ocupare a stărilor prin rezolvarea sistemului (5.1.10).

VI. Calculul indicatorilor de fiabilitate

Dacă se cunosc probabilitățile de ocupare a stărilor sistemului P_i și se utilizează imaginea grafului stărilor, se pot determina indicatorii de fiabilitate pentru sistemul analizat.

În rezolvarea multor probleme de fiabilitate utilizarea proceselor Markov este mult prea complicată și se apelează atunci la procese semi-Markov [45, 58]. **Procesul semi-Markov** este un proces Markov în care se fac anumite simplificări, care ușurează identificarea stărilor sistemului.

În cazul sistemelor de programare, aceste simplificări pot fi:

- se consideră că la un moment dat se execută doar unul din modulele programului;

- tranziția controlului de la un modul la altul este aleatoare, astfel încât probabilitatea de a fi apelat modulul j după executarea modulului i depinde numai de cele două module i și j , nu și de restul modulelor din program.

Pentru durata de execuție a modulelor se admite orice lege de distribuție care descrie funcționarea acestui modul.

Capitolele 6 și 7 din cartea „Fiabilitatea sistemelor informatice” nu fac parte din modulul de curs.

Capitolul VIII. METODE DE ESTIMARE ȘI VALIDARE A INDICATORILOR DE FIABILITATE

8.1. Valori estimate ale indicatorilor de fiabilitate

8.1.1. Valori teoretice (adevărate) și estimate

Valorile teoretice ale indicatorilor de fiabilitate, care caracterizează o mulțime omogenă de produse industriale sau componente ale unor sisteme, se pot determina prin încercări pe **toate** elementele mulțimii sistemelor (produselor) sau componentelor sistemelor respective.

Relațiile de calcul prezentate în tabelul 8.1, preluat din STAS 10307/75 *Fiabilitatea produselor industriale - Indicatori de fiabilitate* [49], se referă la timpul de funcționare până la defectare, în cazul produselor nereparabile, sau până la prima defectare, în cazul produselor reparabile, valorile astfel obținute se numesc **valori adevărate** sau în populație.

Valorile estimate ale indicatorilor de fiabilitate se determină prin prelucrarea statistică a datelor experimentale, obținute prin observații efectuate asupra unui eșantion prelevat din mulțimea de produse/componente considerată și supus unor încercări de fiabilitate [1, 12, 39]. Precizia estimării este cu atât mai mare, cu cât cantitatea de informații disponibilă este mai mare.

Pentru produsele ale căror parametri au asociate plaje de toleranță (instrumente de măsură, aparatură de automatizare etc.), pentru care se poate urmări evoluția unui parametru față de care se poate determina momentul de apariție al defectului, estimarea fiabilității se face urmărind evoluția în timpul încercării a unui sau mai multor parametri caracteristici pentru un element sau sistem.

Se consideră că un produs este defect atunci când unul sau mai mulți parametri caracteristici s-au modificat în timp, ieșind din limitele normale, stabilite pentru funcționarea corespunzătoare de către proiectant, deși produsul poate continua să funcționeze (exemplu: condensatoarele, care își modifică valoarea capacității, sistemul este defect, fără ca această defecțiune să fie catastrofică).

Valorile estimate sunt corecte numai dacă *eșantionul este reprezentativ* pentru populația studiată, iar informațiile deținute nu provin din încercări realizate în condiții diferite sau pe eșantioane diferite.

Tabelul 8.1

Indicator	Valoare teoretică
$F(t)$	$F(t) = \frac{N-n}{N}$
$\bar{f}(t, t + \Delta t)$	$\bar{f}(t, t + \Delta t) = \frac{n(t) - n(t + \Delta t)}{\Delta t \cdot N}$
$R(t)$	$R(t) = \frac{n(t)}{N}$
$\bar{z}(t, t + \Delta t)$	$\bar{z}(t, t + \Delta t) = \frac{n(t) - n(t + \Delta t)}{\Delta t \cdot n(t)}$
m	$m = \frac{1}{N} \sum_{i=1}^N t_i$
D	$D = \frac{1}{N} \sum_{i=1}^N (t_i - m)^2$
σ	$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (t_i - m)^2}$
t_α	Timpul până la care se defectează un anumit procent α din numărul total de $\alpha \cdot N$ elemente din lotul testat

Notă: Simbolurile utilizate în tabel au următoarele semnificații:

N -numărul total de produse la momentul inițial (volumul eșantionului supus încercărilor);

$n(t)$ -numărul de produse, în bună stare, la momentul t ;

t_i -timpul de funcționare a produsului i ($i = 1, 2, \dots, N$);

Δt -interval de timp convenabil ales.

Pentru $f(t)$ și $z(t)$ se pot determina numai valorile medii, nu și cele instantanee.

Valorile indicatorilor de fiabilitate pot fi **estimate punctual** sau cu **interval (limite) de încredere** prin metode specifice, care vor fi prezentate în acest capitol.

Conform terminologiei statistice, estimarea unui parametru este punctuală dacă rezultă din calculul unei valori orientative a parametrului pe baza unei relații matematice folosind rezultatelor experimentale. Pentru ca valoarea estimată punctual să fie cât mai apropiată de cea adevărată se folosesc metode adecvate, ce vor fi prezentate în acest capitol.

Limitele de încredere definesc, în jurul estimatorului punctual, un interval de încredere care include valoarea adevărată a indicatorului de fiabilitate, cu o anumită probabilitate, numită **nivel de încredere**.

Intervalul de încredere este cu atât mai mic (deci estimația este mai precisă) cu cât pentru calcul sunt disponibile de un volum mai mare de informații, rezultate din observații (fie din încercări în laborator, fie din exploatare).

Intervalul de încredere poate fi unilateral sau bilateral.

În cazul unui interval unilateral pentru anumiți indicatori și anumite legi de repartiție statistică se dă limita inferioară (*inf*), pentru alți indicatori sau alte legi de repartiție statistică se dă limita superioară (*sup*). Dacă se dă o singură limită și nu se specifică felul ei, este vorba de limita inferioară.

În cazul unui interval bilateral, se dă, în același timp, o limită inferioară și o limită superioară. Intervalul de încredere include valorile adevărate ale indicatorilor de fiabilitate, cu o anumită probabilitate. Această probabilitate este dată în general în funcție de α sau β , care au următoarea semnificație:

- α reprezintă probabilitatea de a respinge o ipoteză, deși ea este adevărată. Aceasta este considerată o *eroare de ordinul I*. În statistica referitoare la fiabilitate, la fel ca și în calitate, α se mai numește și *riscul furnizorului* și reprezintă probabilitatea de a respinge un lot de produse/componente, deși el este bun;

- β reprezintă probabilitatea de a admite o ipoteză deși ea este falsă. Aceasta este considerată o *eroare de ordinul II*. În statistica referitoare la fiabilitate β se mai numește *riscul beneficiarului* și reprezintă probabilitatea de a accepta un lot de produse/componente, deși el **nu** este bun.

8.1.2. Calculul valorilor estimate ale indicatorilor de fiabilitate

Estimarea valorilor indicatorilor de fiabilitate se poate face prin **metode neparametrice**, a căror aplicare nu necesită identificarea legii de repartiție a timpului de funcționare, sau prin **metode parametrice**, a căror aplicare necesită în prealabil indicarea legii de repartiție a timpului de funcționare.

Observațiile asupra eșantionului se pot efectua utilizând un **plan trunchiat** de încercări, când observațiile se încheie după un timp stabilit inițial, sau **plan cenzurat** de încercări, când observațiile se încheie după apariția unui număr de defectări stabilit inițial, sau planuri care utilizează încercări combinate, trunchiate și cenzurate sau secvențiale. În tabelul 8.2, preluat tot din [39], sunt prezentate relațiile pentru calculul valorilor estimate punctual (coloanele 2 și 3) sau cu interval de încredere bilateral simetric, cu nivel de încredere impus $1-\alpha$ (coloanele 4 și 5), *prin metode parametrice*, pentru cazul în care legea de repartiție a timpului de funcționare este cunoscută.

Tabelul 8.2

Indicator	punctual, pentru:		cu interval de încredere, pentru:	
	<i>produse nereparabile</i>	<i>produse reparable</i>	<i>produse nereparabile</i>	<i>produse reparable</i>
$\hat{F}(t)$	$(*) \frac{r}{N}$	$\frac{r'}{q}$	$\sum_{K=0}^r C_n^K F_{\text{sup}}^K (1 - F_{\text{sup}})^{n-K} = \frac{\alpha}{2}$	$\sum_{K=0}^{r'} C_q^K F_{\text{sup}}^K (1 - F_{\text{sup}})^{q-K} = \frac{\alpha}{2}$
	$(**) \frac{r}{N+1}$		$\sum_{K=r}^n C_n^K F_{\text{inf}}^K (1 - F_{\text{inf}})^{n-K} = \frac{\alpha}{2}$	$\sum_{K=r'}^q C_q^K F_{\text{inf}}^K (1 - F_{\text{inf}})^{q-K} = \frac{\alpha}{2}$
$\tilde{f}(t, t + \Delta t)$	$\frac{n(t) - n(t + \Delta t)}{\Delta t \cdot N}$	-	-	-
$\hat{R}(t)$	$(*) \frac{N-r}{N}$	$\frac{q-r'}{q}$	$\sum_{K=0}^r C_n^K R_{\text{inf}}^{n-K} (1 - R_{\text{inf}})^K = \frac{\alpha}{2}$	$\sum_{K=0}^{r'} C_q^K R_{\text{inf}}^{q-K} (1 - R_{\text{inf}})^K = \frac{\alpha}{2}$
	$(**) \frac{N-r+1}{N+1}$		$\sum_{K=r}^n C_n^K R_{\text{sup}}^{n-K} (1 - R_{\text{sup}})^K = \frac{\alpha}{2}$	$\sum_{K=r'}^q C_q^K R_{\text{sup}}^{q-K} (1 - R_{\text{sup}})^K = \frac{\alpha}{2}$
$\bar{z}(t, t + \Delta t)$	$\frac{n(t) - n(t + \Delta t)}{\Delta t \cdot n(t)}$	-	-	-
\hat{m}	$\frac{T_{\Sigma}}{r}$	$\frac{1}{p} \sum_{j=1}^p t'_j$	-	-
	$\frac{\sum_{i=1}^n (t_i - \hat{m})^2}{N-1}$		-	-

\hat{D}	$\frac{\sum_{i=1}^n (t_i - \hat{m})^2}{N - 1}$	$\frac{\sum_{j=1}^p (t'_j - \hat{m})^2}{p - 1}$	-	-
Indicator	punctual, pentru:		cu interval de încredere, pentru:	
	<i>produse nereparabile</i>	<i>produse reparable</i>	<i>produse nereparabile</i>	<i>produse reparable</i>
$\hat{\sigma}$	$\sqrt{\frac{\sum_{i=1}^n (t_i - \hat{m})^2}{N - 1}}$	$\sqrt{\frac{\sum_{j=1}^p (t'_j - \hat{m})^2}{p - 1}}$	-	-
\hat{t}_F	Timpu până la care se defectează $F \cdot N$ produse	Durata fiecăreia din cele q misiuni inițiate, din care $F \cdot q$ sunt ratate	-	-

Observații:

1. Simbolurile utilizate au următoarele semnificații:

$N = n(0)$ - numărul de produse/componente supuse testului, la momentul $t = 0$, respectiv volumul eșantionului;

$n(t)$ - numărul de produse/componente din eșantion, aflate în bună stare la momentul t ;

r - numărul de defectări observate până la un moment dat;

t_i - momentul apariției unei defectări ($i = 1, 2, \dots, r$);

T_Σ - durata cumulată de încercare a produselor până la punctul de decizie:

$$T_\Sigma = \sum_{i=1}^r t_i + (n-r)t;$$

Δt - interval de timp convenabil ales;

q - numărul de misiuni de durată t , inițiate asupra produsului;

r' - numărul de misiuni, din cele inițiate, în care produsul s-a defectat;

p - numărul de încercări efectuate asupra unui produs până la defectarea acestuia

t_j - durata încercării ($j = 1, 2, \dots, p$).

2. Relațiile notate cu (**) sunt valabile în cazul în care ultima defectare coincide, în timp, cu sfârșitul intervalului de observare ($t_r = t$), iar cele notate cu (*) sunt valabile în caz contrar $t_r \neq t$.

3. Pentru $f(t)$ și $z(t)$ se pot estima numai valorile medii, nu și cele instantanee.

8.1.3. Caracteristicile estimărilor punctuale

Estimarea punctuală a unui parametru constă în calculul unei valori orientative a parametrului respectiv, pe baza rezultatelor experimentale.

Estimația punctuală $\hat{\theta}$ a unui parametru este ea însăși o variabilă aleatoare, cu o distribuție $f(\hat{\theta}/\theta)$, condiționată de valoarea adevărată a parametrului θ și caracterizată de mărimi specifice estimărilor [8]:

a) deplasarea

O estimație este nedeplasată, dacă valoarea sa medie coincide cu valoarea parametrului de estimat. Relația matematică, care descrie această afirmație, este:

$$\int_0^{\infty} \hat{\theta} \cdot f\left(\frac{\hat{\theta}}{\theta}\right) d\hat{\theta} = \theta. \quad (8.1.2)$$

b) consistența

O estimație este consistentă dacă tinde spre valoarea adevărată a parametrului o dată cu creșterea volumului observațiilor.

c) precizia

Precizia estimării reprezintă măsura apropierii valorii estimate punctual de valoarea adevărată a parametrului. Precizia se exprimă, de obicei, cu ajutorul abaterii medii pătratice a estimatorului sau al dispersiei estimatorului. Estimația punctuală, care are dispersia minimă, se numește *estimație eficientă*.

Precizia unei estimări punctuale se mai definește cu ajutorul *abaterii normale*, egală cu modulul diferenței dintre estimația punctuală și valoarea adevărată a parametrului, raportată la valoarea adevărată:

$$\Delta\theta = \frac{|\hat{\theta} - \theta|}{\theta}. \quad (8.1.3)$$

În figura 8.1 sunt reprezentate repartiții $f(\hat{\theta})$, asociate mai multor estimări ale aceluiași parametru θ . Se observă că estimatorii $\hat{\theta}_1$ și $\hat{\theta}_2$ sunt nedeplasați, iar estimatorul $\hat{\theta}_3$ este deplasat. Estimatorul θ_1 are o dispersie mai mică (curba este mai strânsă), conduce la valori mai apropiate de valoarea reală a parametrului θ .

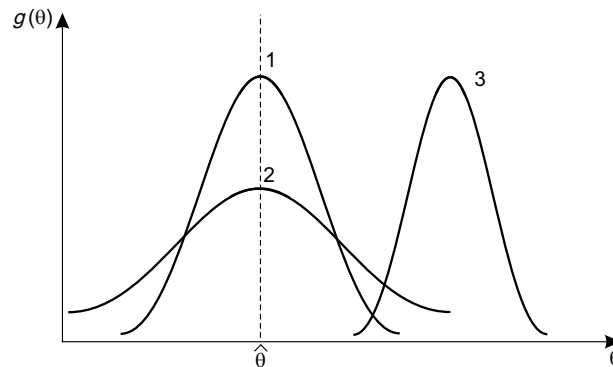


Figura 8.1. Precizia estimării punctuale

d) funcția de pierderi

Funcția de pierderi este definită ca medie a pătratului diferenței dintre valoarea estimată punctual și valoarea adevărată a parametrului:

$$L_s(\theta) = \int_0^{\infty} (\hat{\theta} - \theta)^2 \cdot f(\theta) d\hat{\theta}. \quad (8.1.4)$$

Pentru estimări nedeplasate, funcția de pierderi coincide cu abaterea medie pătratică a estimatorului.

e) cantitatea de informații

Cantitatea de informație poate fi definită ca medie a logaritmului repartiției estimatorului:

$$I_s = - \int_0^{\infty} f(\hat{\theta}) \cdot \log f(\hat{\theta}) \cdot d\hat{\theta}. \quad (8.1.5)$$

Precizia estimăției depinde de cantitatea de informații referitoare la estimatorul respectiv și informație.

f) **energia informației** este definită ca integrala pătratului distribuției estimatorului.

$$E = \int_0^{\infty} f^2(\hat{\theta}) d\theta. \quad (8.1.6)$$

Cu cât energia informației este mai mare cu atât localizarea parametrului este mai precisă.

8.2. Metode de estimare punctuală a parametrilor statistici specifici fiabilității

8.2.1. Metoda verosimilității maxime (maximum likelihood method)

Una dintre cele mai utilizate metode de estimare punctuală este metoda verosimilității maxime, conform căreia valoarea estimată punctuală este aceea care maximizează probabilitatea de apariție a rezultatelor experimentale [9, 16, 26]. Probabilitatea asociată rezultatelor experimentale se numește *funcție de verosimilitate* și se notează cu $L\left(\frac{t}{\theta}\right)$, unde t este vectorul rezultatelor experimentale iar θ este parametrul de estimat.

Metoda verosimilității maxime este o metodă uzuală pentru estimarea punctuală a parametrilor unei distribuții statistice, servește la estimarea unui indicator de fiabilitate fără a face ipoteze privind natura legii de repartiție a timpului de funcționare; este deci o metodă neparametrică.

Dezavantajul metodelor neparametrice este legat de faptul că valorile estimate nu pot fi extrapolate dincolo de intervalul de timp în care se desfășoară experimentul.

Se consideră că densitatea de probabilitate este cunoscută și depinde de s parametri $\theta_1, \theta_2, \dots, \theta_s$, adică este de forma $f(t_i, \theta_k)$ ($i = 1, \dots, n$; $k = 1, \dots, s$).

Funcția de verosimilitate $L(\theta)$ se definește ca funcția de densitate de probabilitate reunită a variabilelor aleatoare t_1, t_2, \dots, t_n :

$$L(\theta_k) = \prod_{i=1}^n f(t_i, \theta_k). \quad (8.2.1)$$

Valoarea cea mai verosimilă (cea mai probabila a fi obținută) a parametrului θ este aceea pentru care funcția $L(\theta_k)$ are valoarea maximă. După cum se știe de la Analiza matematică, această cerință se îndeplinește dacă derivata parțială a funcției în raport cu parametrul respectiv se anulează.

Estimarea de maximă verosimilitate pentru $\hat{\theta}_k$ (semnul pus deasupra simbolului desemnează, așa cum am mai specificat valoare estimată a lui θ ; se mai notează și ca valoare medie a estimatorului $\bar{\theta}_k$) se obține prin maximizarea funcției de verosimilitate $L(\theta_k)$, în raport cu θ_k . Valorile parametrilor $\theta_1, \theta_2, \dots, \theta_n$ se determină rezolvând sistemul de ecuații de verosimilitate, respectiv sistemul de derivatele parțiale ale funcției de verosimilitate în raport cu parametrii de interes.

Ecuația (8.2.1) în practica se dovedește dificil de aplicat, practic este mai uzual a se deriva logaritmul natural al funcției $L(\theta_k)$ (pentru că funcția logaritmică este strict crescătoare), respectiv:

$$\frac{\partial \ln L(\theta_k)}{\partial \theta_k} = 0. \quad (8.2.2)$$

Soluțiile sistemului de ecuații (8.2.2) se numesc *estimații de verosimilitate maximă*.

Exemple:

1) Să se estimeze valorile ratei de defectare, λ , pe baza duratelor de defectare (t_1, t_2, \dots, t_n) pentru sistem pentru care, pentru timpul dintre defectării este valabilă distribuția exponențială.

Rezolvare:

Estimatorul (parametrul) în acest caz, este λ .

Funcția de verosimilitate este:

$$L(t, \lambda) = \prod_{i=1}^n \lambda e^{-\lambda t_i} = \lambda^n e^{-\lambda \sum_{i=1}^n t_i}, \quad (8.2.3)$$

Valoarea maximă a estimatorului se obține din egalarea cu 0 a derivatei logaritmului funcției $L(t, \lambda)$ în raport cu estimatorul λ

$$\ln L(\lambda) = n \ln \lambda - \lambda \sum_{i=1}^n t_i, \quad (8.2.3')$$

$$\frac{\partial \ln L(\lambda)}{\partial \lambda} = \frac{n}{\lambda} - \sum_{i=1}^n t_i = 0. \quad (8.2.4)$$

Soluția ecuației 8.2.4 este:

$$\bar{\lambda} = \frac{n}{\sum_{i=1}^n t_i} = \frac{1}{\bar{t}}. \quad (8.2.5)$$

2) Pentru cazul distribuției binomiale (Bernoulli), se consideră θ_k variabilă aleatoare care poate lua valorile 1 și 0, cu probabilitățile p și respectiv $q = 1 - p$. Să se estimeze probabilitatea p pe baza unei selecții repetate $\theta_1, \dots, \theta_k, \dots, \theta_n$.

Probabilitatea ca $\theta_1 = x_1, \dots, \theta_n = x_n$ este egală cu $\Psi(x_1, \dots, x_n) = p^k q^{n-k}$, dacă $n - k$ dintre valorile x_1, \dots, x_n sunt egale cu 0, iar k sunt egale cu 1.

Ecuația verosimilității maxime se poate scrie:

$$\frac{\partial \ln \Psi}{\partial p} = \frac{\partial}{\partial p} \ln(p^k q^{n-k}) = \frac{k}{p} - \frac{n-k}{q}, \quad (8.2.6)$$

$$\frac{k}{p} - \frac{n-k}{q} = 0 \Rightarrow \frac{k}{p} - \frac{n-k}{1-p} = 0 \quad (8.2.7)$$

$$p = \frac{k}{n} = \frac{x_1 + \dots + x_n}{n} \quad (8.2.8)$$

Așadar, valoarea medie a selecției, $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ (8.2.8') este o estimație de verosimilitate maximă a probabilității p .

8.2.2. Metoda liniarizării

Este tot o metodă de estimare punctuală, care constă în reprezentarea grafică a funcției de distribuție empirică într-un sistem de coordonate convenabil ales astfel încât reprezentarea funcției de distribuție teoretică să fie o dreaptă.

Metoda liniarizării se utilizează nu numai pentru estimarea parametrilor distribuției, dar și pentru stabilirea tipului de funcție de repartiție statistică, atunci când aceasta nu este cunoscută.

Pentru fiecare tip de distribuție se utilizează hârtie de probabilitate specială, denumirea standardizată fiind *rețea probabilistică* [12, 43, 45].

Parametrii funcției teoretice (necunoscuți) rezultă din parametrii dreptei empirice (panta, termenul liber, intersecții cu anumite drepte).

a) În cazul **distribuției exponențiale** (prezentate la paragraful 3.5, $F(t) = 1 - e^{-\lambda \cdot t}$), se poate scrie:

$$\lambda t = \ln[1 - F(t)]^{-1}. \quad (8.2.9)$$

Introducând notația $y = \ln[1 - F(t)]^{-1}$, ecuația (8.2.9) devine ecuația unei drepte:

$$y = t \cdot \lambda, \quad (8.2.10)$$

care trece prin origine și are panta λ .

Valoarea empirică a funcției $F(t)$ este

$$\hat{F}(t_i) = \frac{i}{n+1}. \quad (8.2.11)$$

Funcția care trebuie reprezentată grafic și care să fie o dreaptă este:

$$\hat{y}_i = \ln\left(\frac{n+1}{n+1-i}\right). \quad (8.2.12)$$

Dacă punctele obținute din experiment se aliniază pe o dreaptă, atunci se confirmă caracterul exponențial al distribuției.

Parametrul λ se poate estima din panta dreptei reprezentată.

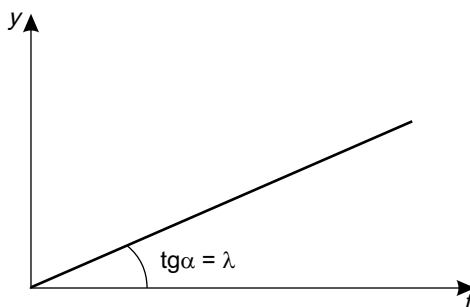


Figura 8.2. Metoda liniarizării (pentru distribuția exponențială)

c) Distribuția normală

Parametrii distribuției normale sunt m și σ , după cum s-a prezentat în capitolul 3 al acestei cărți. Liniarizarea distribuției normale se obține pornind de la forma exprimată prin funcția Laplace, formula 3.5.12 din capitolul 3:

$$F(t; m, \sigma) = \Phi\left(\frac{t-m}{\sigma}\right). \quad (8.2.18)$$

Inversa funcției Laplace, pentru o anumită valoare t_i , este:

$$\Phi^{-1}[F(t_i, m, \sigma)] = \frac{t_i - m}{\sigma} = \frac{1}{\sigma}t_i - \frac{m}{\sigma}. \quad (8.2.19)$$

Coordonatele care liniarizează distribuția normală sunt:

- analitice: (y_i, t_i) ;

- empirice (\hat{y}_i, \hat{t}_i) ,

unde:

$$y_i = \frac{1}{\sigma}t_i - \frac{m}{\sigma} \quad \text{și} \quad \hat{y}_i = \Phi^{-1}(\hat{F}). \quad (8.2.20)$$

(După cum am menționat în capitolul 2, valorile funcției Laplace și ale inversei sale sunt tabelate [12, 17].

Parametrii drepte sunt:

- intersecția la origine: $a = \frac{1}{\sigma}$; (8.2.20')

- panta: $b = -\frac{m}{\sigma}$. (8.2.20'')

În planul (y, t) se reprezintă punctele $P_i(\hat{y}_i, \hat{t}_i)$, care, dacă se aliniază după o dreaptă, validează ipoteza distribuției normale. Și pentru distribuția normală există o hârtie specială de probabilitate, prezentată în figura 8.3 [43].

Rețeaua probabilistică normală (Gauss) este astfel concepută încât funcția de repartiție cumulată a timpilor până la defectare, care urmează o distribuție normală, să se reprezinte printr-o dreaptă, conform figurii 8.6. Axele acestei rețele sunt:

-abscisa -valorile timpilor (t_i) obținuți prin încercări,

-ordonata -valorile $F(t) = \frac{t_i}{P_{50}(t_i)}$, unde valorile $P_{50}(t_i)$ se calculează cu

relația: $P_{50} = 100 \frac{i-0,3}{n+0,4}$ (8.2.21)

Poate fi utilizată și pentru încercările în care nu s-au defectat toate elementele din lotul supus încercărilor. Dacă încercarea este întreruptă înainte de defectarea tuturor elementelor eșantionului de test, se reprezintă numai valorile mediane până

la $i = r$. În exemplul din figura 8.3. eșantionul de test are 10 elemente, iar încercarea a fost oprită după $r = 8$ defectări.

Estimatorul timpului mediu până la defectare, \hat{m}_F este egal cu abscisa punctului de pe dreapta de ordonată egală cu 50%.

Estimatorul abaterii medii pătratice $\hat{\sigma}$ este egal cu diferența dintre abscisa punctului de pe dreapta de ordonată, egală cu 84% și \hat{m}_F , așa cum se poate urmări în figura 8.3.

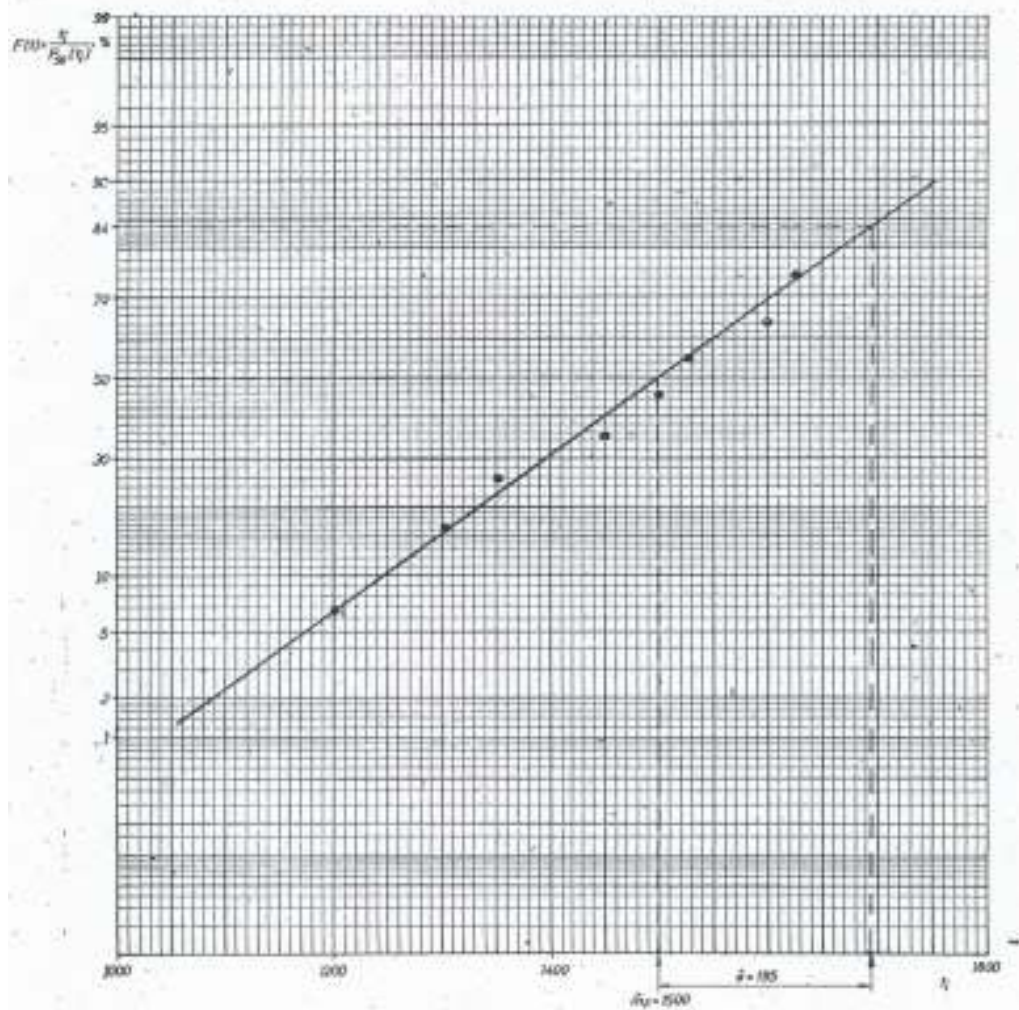


Figura 8.3. Rețeaua probabilistică pentru distribuția normală

8.2.3. Metoda momentelor

Constă în compararea momentelor empirice de diferite ordine (k) cu momentele corespunzătoare ale distribuției teoretice luate ca ipoteză.

Momentele empirice, considerate ca estimări ale șirului (t_1, t_2, \dots, t_n) , se determină cu relația:

$$\hat{M}_k = \frac{1}{n} \sum_{i=1}^n t_i^k. \quad (8.2.22)$$

Aceste momente se compară cu momentele distribuțiilor analitice definite de relația:

$$M_k = \int_0^{\infty} t^k f(t, \theta) dt. \quad (8.2.22')$$

8.2.4. Metoda celor mai mici pătrate

Estimarea parametrilor prin această metodă constă în formarea unei sume S , egală cu suma diferențelor pătratelor dintre funcția de repartiție empirică (\hat{F}_i) și funcția de repartiție analitică considerată:

$$S = \sum_{i=1}^n [\hat{F}_i(t_i) - F(t_i, \theta)]^2. \quad (8.2.23)$$

Se pune condiția ca această sumă să fie minimă.

Determinarea parametrului θ se realizează din condiția de minim a sumei S , respectiv prin anularea derivatei sumei S , în raport cu parametrul distribuției analitice utilizate.

Dacă:

$$\frac{\partial S}{\partial \theta} = 0 \Rightarrow \theta. \quad (8.2.24)$$

Dacă funcția analitică are mai mulți parametri, suma S se va deriva în raport cu fiecare parametru. Determinarea parametrilor se face prin rezolvarea unui sistem de ecuații independent.

a) Pentru distribuția exponențială, relația pentru calcul lui λ este:

$$\lambda = \frac{\sum_{i=1}^n y_i t_i}{\sum_{i=1}^n t_i^2}. \quad (8.2.25)$$

b) pentru distribuția Weibull, parametrii β și η sunt calculați din relațiile:

$$\beta = \frac{\sum_{i=1}^n \hat{y}_i \cdot \sum_{i=1}^n (\ln t_i)^2 - \sum_{i=1}^n (\hat{y}_i \cdot \ln t_i) \cdot \sum_{i=1}^n \ln t_i}{n \sum_{i=1}^n (\ln t_i)^2 - \left(\sum_{i=1}^n \ln t_i \right)^2}, \quad (8.2.26)$$

$$-\beta \ln \eta = \frac{n \sum_{i=1}^n (\hat{y}_i \cdot \ln t_i) - \sum_{i=1}^n \hat{y}_i \cdot \sum_{i=1}^n \ln t_i}{n \sum_{i=1}^n (\ln t_i)^2 - \left(\sum_{i=1}^n \ln t_i \right)^2}. \quad (8.2.27)$$

c) pentru distribuția normală, m și σ se calculează din relațiile:

$$\frac{1}{\sigma} = \frac{n \sum_{i=1}^n (\hat{y}_i \cdot t_i) - \sum_{i=1}^n \hat{y}_i \cdot \sum_{i=1}^n t_i}{n \sum_{i=1}^n (t_i)^2 - \left(\sum_{i=1}^n t_i \right)^2}, \quad (8.2.28)$$

$$-\frac{m}{\sigma} = \frac{\sum_{i=1}^n \hat{y}_i \cdot \sum_{i=1}^n t_i^2 - \sum_{i=1}^n (\hat{y}_i \cdot t_i) \sum_{i=1}^n t_i}{n \sum_{i=1}^n (\ln t_i)^2 - \left(\sum_{i=1}^n \ln t_i \right)^2}. \quad (8.2.29)$$

8.3. Metoda intervalelor de încredere pentru estimarea parametrilor statistici specifici fiabilității

De cele mai multe ori, estimarea parametrilor statistici nu se face punctual, ci cu un interval de încredere $[\theta_{\text{inf}}, \theta_{\text{sup}}]$, cu limita superioară și inferioară stabilite pe baza statisticii evenimentului (t_1, t_2, \dots, t_n) .

Metoda presupune stabilirea unui interval de încredere pentru parametrul urmărit și precizarea probabilității ca parametrul să aparțină acestui interval:

$$P(\theta_{\text{inf}} \leq \theta \leq \theta_{\text{sup}}) = 1 - \alpha, \quad (8.3.1)$$

unde: $1 - \alpha$ reprezintă nivelul de încredere, respectiv probabilitatea ca intervalul de încredere să cuprindă valoarea adevărată a parametrului estimat; α este nivelul de semnificație sau riscul furnizorului, așa cum am precizat la paragraful 8.1.1.

Intervalul de încredere poate fi bilateral și simetric în jurul valorii prezise a parametrului θ , sau unilateral, dacă una din limitele intervalului este 0 sau ∞ .

Nivelul de semnificație α pentru intervalul bilateral reprezintă probabilitatea de a respinge o ipoteză, deși ea este adevărată, iar în cazul intervalului unilateral, α , reprezintă valoarea maximă a acestei probabilități.

În mod uzual $\alpha = 0,01$ sau $0,05$, respectiv o șansă din 100 sau o șansă din 20, în funcție de riscul pe care este dispus să și-l asume fabricantul sau utilizatorul.

Există mai multe posibilități de determinare a valorilor $\hat{\theta}_{\text{inf}}$ și $\hat{\theta}_{\text{sup}}$. Una dintre acestea este identificarea lor cu cuantile de ordinul $\alpha/2$, respectiv $1 - \alpha/2$ față de repartiția estimatorului.

Limitele $\hat{\theta}_{\text{inf}}$ și $\hat{\theta}_{\text{sup}}$ depind de valoarea reală, necunoscută, a parametrului θ , ambele fiind funcții crescătoare de acest parametru, conform figurii 8.4.

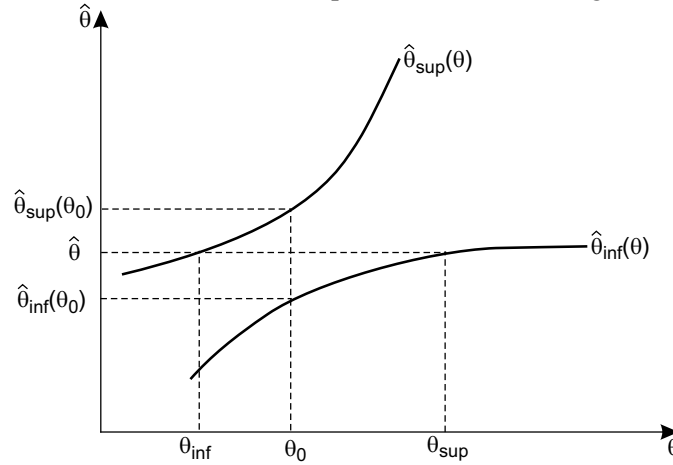


Figura 8.4. Construcția intervalului de încredere pentru parametrul estimat

Pentru o valoare fixată θ_0 a parametrului de estimat din figura 8.4 rezultă că există două valori $\hat{\theta}_{\text{inf}}$ și $\hat{\theta}_{\text{sup}}$, astfel încât:

$$P[\hat{\theta}_{\text{inf}}(\theta_0) < \hat{\theta} < \hat{\theta}_{\text{sup}}(\theta_0)] = 1 - \alpha. \quad (8.3.2)$$

Invers, pentru o valoare particulară $\hat{\theta}$ a estimației punctuale se pot determina, pe abscisă, două valori θ_{inf} și θ_{sup} astfel încât:

$$\hat{\theta} = \hat{\theta}_{\text{sup}}(\theta_{\text{inf}}) = \hat{\theta}_{\text{inf}}(\theta_{\text{sup}}). \quad (8.3.3)$$

Datorită monotoniei funcțiilor $\hat{\theta}_{\text{inf}}(\theta)$ și $\hat{\theta}_{\text{sup}}(\theta)$ rezultă faptul că intervalul $[\theta_{\text{inf}}, \theta_{\text{sup}}]$ conține parametrul θ_0 , cu aceeași probabilitate, cu care valoarea estimată $\hat{\theta}$ se găsește în intervalul $[\hat{\theta}_{\text{inf}}(\theta_0), \hat{\theta}_{\text{sup}}(\theta_0)]$:

$$P[\theta_{\text{inf}} < \theta_0 < \theta_{\text{sup}}] = P[\hat{\theta}_{\text{inf}}(\theta_0) < \hat{\theta} < \hat{\theta}_{\text{sup}}(\theta_0)] = 1 - \alpha. \quad (8.3.4)$$

Valorile limitelor intervalului de încredere $\hat{\theta}_{\text{inf}}$ și $\hat{\theta}_{\text{sup}}$ se pot determina pornind de la repartiția $f\left(\frac{\hat{\theta}}{\theta}\right)$, utilizând relațiile [8]:

$$\int_0^{\theta_{\text{inf}}} f\left(\frac{\hat{\theta}}{\theta_0}\right) d\hat{\theta} = \frac{\alpha}{2}, \quad (8.3.5)$$

$$\int_{\theta_{\text{sup}}}^{\infty} f\left(\frac{\hat{\theta}}{\theta_0}\right) d\hat{\theta} = \frac{\alpha}{2}. \quad (8.3.5')$$

Metoda intervalelor de încredere se poate aplica atât pentru estimări neparametrice (asociat probabilității de defectare), cât și pentru estimări parametrice (asociat unui parametru al legii de repartiție, fie λ , fie t_m).

În cazul intervalului unilateral de încredere, interesează limita superioară a probabilității de defectare F_{sup} .

Ecuția este:

$$\sum_{k=0}^r C_n^k F_{\text{sup}}^k (1-F_{\text{sup}})^{n-k} = \alpha. \quad (8.3.6)$$

Valorile $F_{\text{inf}}, F_{\text{sup}}$ respectiv $R_{\text{inf}}, R_{\text{sup}}$ se calculează utilizând tabele specializate.

O metodă de calcul a limitelor superioară și inferioară pentru rata de defectare, utilizând date obținute atât prin încercări cenzurate cât și încercări trunchiate, utilizând repartiția binomială și χ^2 , este dată în [1, 8].

În standardele specifice sunt date tabele pentru calculul indicatorilor de fiabilitate atât încercări trunchiate, cât și cenzurate, cu și fără înlocuire, cu interval de încredere bilateral simetric, pentru repartiția exponențială.

Cu cât intervalul $(\theta_{\text{inf}}, \theta_{\text{sup}})$ este mai mic, cu atât avem o indicație mai precisă asupra parametrului θ .

Există însă o contradicție între nivelul de încredere și precizia estimării: pentru un anumit volum al observațiilor, cu cât nivelul de încredere este mai înalt, cu atât estimarea este mai puțin precisă. Efectuând un număr mare, k , de încercări de fiabilitate asupra unor eșantioane extrase din aceeași populație, se construiesc intervale de încredere:

- $(1-\alpha) \cdot k$ intervale care conțin valoarea adevărată,
- $k \cdot \alpha$ intervale care lasă în afara lor valoarea adevărată.

Cu cât nivelul de încredere este mai mare, cu atât șansa de a greși este mai mică. Creșterea corectitudinii estimării împiedică o lărgire a intervalului de încredere, adică o micșorare a preciziei. În caz extrem, se poate afirma cu un nivel de încredere de 100% că fiabilitatea unui sistem este cuprinsă între 0 și 1, dar precizia acestei estimări este nulă.

Precizia estimării cu interval de încredere este definită prin relațiile [1, 8]:

$$P\left(\frac{\theta_{\text{sup}} - \theta}{\theta} > l_d\right) = \varepsilon_1, \quad (8.3.7, a)$$

$$P\left(\frac{\theta - \theta_{\text{inf}}}{\theta} > l_s\right) = \varepsilon_2, \quad (8.3.7, b)$$

unde:

- l_d și l_s sunt preciziile la dreapta și la stânga;
- ε_1 și ε_2 -erori de estimare.

Mărimile l și ε sunt subunitare.

Aceste relații afirmă următoarele :

-limita superioară a intervalului de încredere nu depășește valoarea adevărată a parametrului cu mai mult de o fracțiune l_d din valoare, cu o probabilitate de $1 - \varepsilon_1$;

-cu o probabilitate de $1 - \varepsilon_2$, se poate afirma că limita inferioară a intervalului de încredere nu se află sub valoarea adevărată a parametrului cu mai mult de o fracțiune l_s din această valoare.

La proiectarea încercării de fiabilitate se aleg convenabil aceste mărimi pentru a putea calcula numărul de defectări care trebuie înregistrate în timpul încercării astfel încât să fie atinsă precizia impusă, la nivelul de încredere stabilit.

8.4. Teste de verificare și validare a ipotezelor statistice

Testele de verificare și validare pot fi și ele:

- parametrice atunci când vizează parametrii funcției de distribuție utilizate;
- neparametrice atunci când vizează forma analitică a funcției de distribuție.

Ipotezele statistice care se fac asupra parametrilor sau funcțiilor de repartiție sunt presupuneri asupra populației din care s-a făcut eșantionarea (selectivă) și nu numai asupra eșantionului. Confirmarea oricărei ipoteze se face cu un anumit risc, deoarece se lucrează pe un eșantion din populație, așa cum am mai precizat. Există mai multe teste de verificare și validare a ipotezelor statistice. Vom prezenta numai trei teste, cele mai frecvent utilizate de specialiștii în fiabilitate.

8.4.1. Testul Kolmogorov-Smirnov

Permite verificarea unei ipoteze H_0 conform căreia funcția de repartiție a variabilei aleatoare ξ este F_0 .

În statistică se demonstrează că pentru $\lambda > 0$ există egalitatea:

$$\lim_{n \rightarrow \infty} P \left[\sup_{-\infty < x < \infty} \left| \hat{F}(x) - F(x) \right| \leq \frac{\lambda}{\sqrt{n}} \right] = \sum_{-\infty}^{\infty} (-1)^k e^{-2k^2 \lambda^2} = K(\lambda) \quad (8.4.1)$$

De aici rezultă că, dacă n este suficient de mare, se poate construi un test de verificare a unei ipoteze H , bazat pe regiunea critică:

$$V = \left\{ (x_1, \dots, x_n) / \sup_{-\infty < x < \infty} \left| \hat{F}(x) - F_0(x) \right| > \frac{\lambda_\alpha}{\sqrt{n}} \right\}. \quad (8.4.2)$$

λ_α se află din condiția:

$$K(\lambda_\alpha) = 1 - \alpha, \quad (8.4.3)$$

α fiind pragul de semnificație al testului.

Testul Kolmogorov-Smirnov este un test neparametric, aplicabil oricărui tip de repartiție, recomandat mai ales pentru cazul în care funcția de repartiție empirică se construiește prin puncte și pune în evidență abaterea maximă dintre funcția de repartiție empirică (experimentală) (\hat{F}) și cea teoretică (F) [1, 8, 12, 17].

Etapele aplicării testului:

-se calculează valorile teoretice $F(t_i)$ și cele empirice $\hat{F}(t_i)$ în toate punctele selecției $t_n(t_1, t_2, \dots, t_n)$;

-se calculează abaterea maximă K_{\max} cu formula:

$$K_{\max} = \max_{1 \leq i \leq n} |\hat{F}(t_i) - F(t_i)|; \quad (8.4.4)$$

-se calculează valoarea

$$K_{\alpha \text{ critic}} = \frac{\lambda_{\alpha}}{\sqrt{n}}, \quad (8.4.5)$$

unde $K_{\alpha \text{ critic}}$ reprezintă abaterea critică a testului, corespunzătoare nivelului de semnificație α .

Mărimea λ_{α} se determină din tabelul cu valori critice ale testului Kolmogorov, în funcție de nivelul de încredere, $k = 1 - \alpha$.

Se compară valoarea calculată cu valoarea critică, și dacă $K_{\max} \leq K_{\alpha \text{ critic}}$ ipoteza făcută este acceptată.

8.4.2. Testul χ^2

Testul χ^2 (*hi pătrat* sau după unii autori *chi pătrat*) se folosește, în special, pentru a verifica dacă o anumită funcție de distribuție modelează suficient de bine comportarea unui lot de produse, din care un eșantion a fost supus la încercări de fiabilitate [1, 8, 12, 17].

Fie X o variabilă aleatoare, care poate lua k valori distincte x_1, \dots, x_k , cu probabilitățile p_1, \dots, p_k . Notăm cu $\theta_1, \dots, \theta_k$ frecvențele de apariție ale valorilor x_1, \dots, x_k într-o selecție repetată de volum n . Variabila aleatoare X^2 de forma:

$$X^2 = \sum_{i=1}^k \frac{(\theta_i - np_i)^2}{np_i}, \quad (8.4.6)$$

are, pentru $k \rightarrow \infty$ o repartiție χ^2 cu $k - 1$ grade de libertate (unii autori notează cu ν gradele de libertate).

Pentru k suficient de mare:

$$P(X^2 \leq \chi^2) \cong \frac{1}{2^{\frac{k-1}{2}} \Gamma\left(\frac{k-1}{2}\right)} \int_0^{\chi^2} x^{\frac{k-1}{2} - 1} e^{-\frac{x}{2}} dx. \quad (8.4.7)$$

Funcția $\Gamma\left(\frac{k-1}{2}\right)$ este *funcția Gama*, numită și funcția Euler de speța II.

Funcția $\Gamma(\alpha)$ este de forma:

$$\Gamma(\alpha) = \int_0^{\infty} t^{\alpha-1} \cdot e^{-t} dt. \quad (8.4.7')$$

Valorile $\Gamma(\alpha)$ sunt tabelate pentru $1 \leq \alpha \leq 2$. Alte valori se pot calcula ținând seama că funcția Γ are proprietatea $\alpha \Gamma(\alpha) = \Gamma(\alpha + 1)$.

Pentru a verifica ipoteza $H_0 : p_i = p_{i0}, i = 1, \dots, k$, trebuie ca:

$$P\left(\sum_{i=1}^k \frac{(\theta_i - np_{i0})^2}{np_{i0}} \leq \chi^2\right) \cong \frac{1}{2^{\frac{k-1}{2}} \Gamma\left(\frac{k-1}{2}\right)} \int_0^{\chi^2} x^{\frac{m-1}{2}-1} e^{-\frac{x}{2}} dx. \quad (8.4.8)$$

Alegând $\chi^2 = \chi_\alpha^2$ astfel încât:

$$\frac{1}{2^{\frac{k-1}{2}} \Gamma\left(\frac{k-1}{2}\right)} \int_0^{\chi_\alpha^2} x^{\frac{m-1}{2}-1} e^{-\frac{x}{2}} dx = 1 - \alpha, \quad (8.4.9)$$

rezultă că ipoteza H_0 va fi acceptată la nivelul de semnificație α , dacă:

$$\sum_{i=1}^k \frac{(\theta_i - np_{i0})^2}{np_{i0}} \leq \chi_\alpha^2; \quad (8.4.10)$$

în caz contrar ipoteza este respinsă.

Etapele aplicării testului sunt următoarele:

-se ordonează crescător timpii de defectare obținuți experimental ($t_1 < t_2 < \dots < t_n$);

-se împarte plaja de timp în k intervale egale, delimitate de valorile T_0, T_1, \dots, T_k ;

-pentru fiecare interval $[T_{j-1}, T_j], j = 1, \dots, k$ se determină numărul de timpi de încercare, care aparțin intervalului respectiv; frecvențele de realizare se notează cu θ_i ;

-se calculează, pentru fiecare interval, probabilitatea p_i ca variabila aleatoare să aparțină intervalului respectiv:

$$p_i = \int_{\theta_{i-1}}^{\theta_i} f(t) dt = F(\theta_i) - F(\theta_{i-1}); \quad (8.4.11)$$

-se determină măsura apropierii dintre numărul de realizări observate și cele așteptate, utilizând relația:

$$d_{k-1} = \sum_{i=1}^k \frac{[\theta_i - np_i]^2}{np_i} \quad (8.4.12)$$

-dacă $a \cong \frac{1-\beta}{\alpha}$ unde α este nivelul de semnificație al testului, ipoteza se acceptă. Funcția χ_α^2 este tabelată în cărțile de statistică, de exemplu [12, 18].

8.4.3. Testul secvențial al lui Wald

Testul secvențial al lui Wald (Abraham Wald, născut la Cluj în 1902) constituie o metodă de obținere rapidă a deciziei de acceptare sau respingere a unui lot încercat prin completarea unui grafic, numit **Planul lui Wald**. Este denumit și *planul secvențial al lui Wald* întrucât pornește de la încercări de fiabilitate de tip secvențial [8, 26, 52].

Graficul planului are axele T_{Σ} (timpul cumulat de încercare până la luarea deciziei de întrerupere a încercării) și r (numărul de defectări înregistrate). În aceste coordonate se trasează două drepte paralele A_N și R_N , corespunzătoare limitelor de acceptare, respectiv de respingere a lotului de N elemente. Se procedează secvențial, marcându-se pe plan, după fiecare defectare ($r = 1, 2, 3, \dots, n$), punctul de coordonate (T_{Σ}, r) , așa cum se poate urmări în figura 8.4.

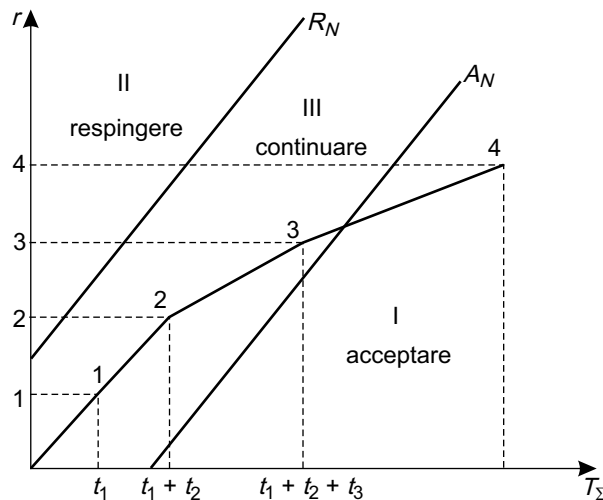


Figura 8.4. Testul secvențial al lui Wald

În momentul în care curba experimentală obținută intersectează una din drepte, încercarea se oprește și se ia decizia de acceptare sau respingere.

Dreptele limită A_N și R_N se determină astfel:

a) Se stabilesc valorile următoarelor mărimi:

θ_0 -valoarea acceptabilă *MTTF*,

θ_1 -o valoare inacceptabilă a *MTTF* ($\theta_1 > \theta_0$),

α -riscul furnizorului (probabilitatea maximă de a respinge un lot acceptabil cu $\theta = \theta_0$),

β -riscul beneficiarului (probabilitatea maximă de a admite un lot inacceptabil cu $\theta = \theta_1$).

b) Pe caracteristica operativă pentru încercare de forma celei prezentată în figura 8.5, se deduc probabilitățile de acceptare:

$$P_a(\theta_0) = 1 - \alpha, \quad P_a(\theta_1) = \beta. \quad (8.4.13)$$

c) Pentru încercarea cu numărul de ordine r , se introduc notațiile:

- $P_0(r)$ -probabilitatea ca să fie valabilă ipoteza $H_0 : \theta = \theta_0$,
- $P_1(r)$ -probabilitatea ca să fie valabilă ipoteza $H_1 : \theta = \theta_1$.

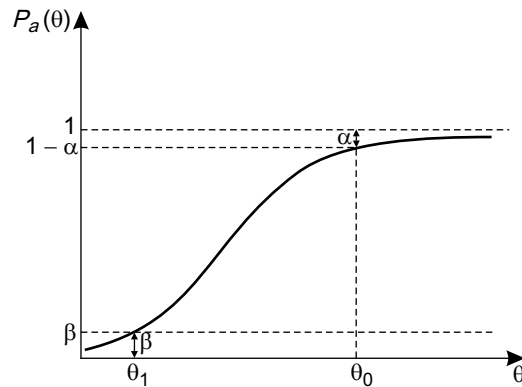


Figura 8.5. Caracteristica operativă pentru determinarea probabilității de acceptare

Notă: reprezentarea din figură nu păstrează proporțiile pentru a pune în evidență mărimile α și β .

d) Se calculează raportul $\frac{P_1(r)}{P_0(r)}$ ținând seama de faptul că numărul de defecțiuni r în timpul T_Σ are o repartiție Poisson, respectiv:

$$P(r) = \left(\frac{T_\Sigma}{\theta}\right)^r \frac{e^{-\frac{T_\Sigma}{\theta}}}{r!}. \quad (8.4.14)$$

Prin urmare:

$$\frac{P_1(r)}{P_0(r)} = \left(\frac{\theta_0}{\theta_1}\right)^r e^{-T_\Sigma \left(\frac{1}{\theta_1} - \frac{1}{\theta_0}\right)}. \quad (8.4.15)$$

e) Testul lui Wald constă în a forma pentru fiecare valoare a lui r raportul $\frac{P_1(r)}{P_0(r)}$ și a-l compara cu două numere date a și b aflate în relația $b < a$, ambele stabilite pe baza caracteristicilor operative conform relațiilor:

$$a \cong \frac{1-\beta}{\alpha} \quad \text{și} \quad b \cong \frac{\beta}{1-\alpha}. \quad (8.4.16)$$

Dacă

$$\frac{P_1(r)}{P_0(r)} \leq \frac{\beta}{1-\alpha}, \quad (8.4.17)$$

atunci lotul se acceptă, conform ipotezei H_0 .

Dacă

$$\frac{P_1(r)}{P_0(r)} \geq \frac{1-\beta}{\alpha}, \quad (8.4.18)$$

atunci lotul se respinge, conform ipotezei H_1 .

Dacă

$$\frac{\beta}{1-\alpha} < \frac{P_1(r)}{P_0(r)} < \frac{1-\beta}{\alpha}, \quad (8.4.19)$$

testul se continuă.

Condiția de continuare a testului, cu notațiile de la (8.4.16) se mai poate scrie:

$$b < \frac{P_1(r)}{P_0(r)} < a \quad (8.4.19')$$

sau
$$\ln b \leq r \ln \frac{\theta_0}{\theta_1} + T_\Sigma \left(\frac{1}{\theta_1} - \frac{1}{\theta_0} \right) \leq \ln a. \quad (8.4.19'')$$

Împărțind cu $\ln \frac{\theta_0}{\theta_1}$, relația (8.4.19'') devine:

$$\frac{\ln b}{\ln \frac{\theta_0}{\theta_1}} + \frac{T_\Sigma \left(\frac{1}{\theta_1} - \frac{1}{\theta_0} \right)}{\ln \frac{\theta_0}{\theta_1}} \leq r \leq \frac{\ln a}{\ln \frac{\theta_0}{\theta_1}} + \frac{T_\Sigma \left(\frac{1}{\theta_1} - \frac{1}{\theta_0} \right)}{\ln \frac{\theta_0}{\theta_1}}. \quad (8.4.20)$$

Introducând notațiile:

$$m = \frac{\frac{1}{\theta_1} - \frac{1}{\theta_0}}{\ln \frac{\theta_0}{\theta_1}}, \quad (8.4.21, a)$$

$$n_1 = \frac{\ln b}{\ln \frac{\theta_0}{\theta_1}} = \frac{\ln \frac{\beta}{1-\alpha}}{\ln \frac{\theta_0}{\theta_1}}, \quad (8.4.21, b)$$

$$n_2 = \frac{\ln a}{\ln \frac{\theta_0}{\theta_1}} = \frac{\ln \frac{1-\beta}{\alpha}}{\ln \frac{\theta_0}{\theta_1}}; \quad (8.4.21, c)$$

relația (8.4.20) se poate scrie: $mT_\Sigma + n_1 \leq r \leq mT_\Sigma + n_2. \quad (8.4.22)$

Se constată că în partea dreaptă și în partea stângă a inecuației sunt ecuațiile a două drepte paralele:

$$A_N = mT_\Sigma + n_1, \quad (8.4.22, a)$$

$$R_N = mT_\Sigma + n_2. \quad (8.4.22, b)$$

Aceste drepte împart planul (T_Σ, r) în trei regiuni: regiunea I - de acceptare a ipotezei H_0 , regiunea II - de respingere a ipotezei H_0 și regiunea III - de continuare a testului, așa cum se poate observa în figura 8.4.

Capitolul 9. FIABILITATEA PREVIZIONALĂ SOFTWARE

S-a arătat în capitolul 1 al acestei cărți că defectarea software înseamnă manifestarea în timpul exploatării acestuia a unor erori latente existente în program. Defectarea software constă în punerea în evidență a unei erori latente conținute în program. Testarea software nu poate demonstra absența erorilor, ci numai prezența lor.

Durata până la manifestarea erorii este o variabilă aleatoare, ceea ce permite ca fiabilitatea programelor să fie modelată similar cu fiabilitatea sistemelor tehnice. Programul poate fi considerat un sistem cu reînnoire, fiecare defectare (manifestare a unei erori) implică o intervenție exterioară prin care se tinde spre micșorarea numărului de erori latente conținute în program.

Modelarea fiabilității unui sistem de programare în fazele de proiectare și dezvoltare are ca scop predicția funcției de fiabilitate, a timpului mediu de funcționare (exprimat fie ca durată medie până la prima defectare sau ca o durată medie între două defectări) dar și a funcției de reînnoire.

9.1. Modele de fiabilitate software și indicatori

Un model de fiabilitate software determină forma unui proces aleator care descrie manifestarea erorilor software-ului în timp.

Modelele de fiabilitate software urmează, în general, următoarele ipoteze:

i) Profilul operațional în care se va măsura fiabilitatea este același cu profilul de testare în care modelul de fiabilitate a fost parametrizat.

ii) În cazul manifestării unei defectări, eroarea care a cauzat defectarea este corectată.

iii) Procesul de înlăturare a erorii nu va introduce noi erori.

iv) Numărul de erori intrinseci din software și modul lor de manifestare respectă o anumită lege matematică care poate fi identificată prin metode adecvate.

Întrucât numărul de erori intrinseci conținute în soft scade în timp și astfel fiabilitatea crește, se poate considera că sistemul software este un sistem cu uzură negativă, (vechi mai bun decât nou), iar modelele specifice sunt denumite adeseori *modele de creștere a fiabilității software-ului* (SRGM - Software Reliability Grow Model).

Începând din 1963, când Jelinsky și Moranda au propus primul model de creștere a fiabilității, au apărut numeroase modele SRGM de fiabilitate, cum ar fi modelele cu creștere exponențială a fiabilității, modelele Weibull, modelele Bayesiene și altele [5, 48]. Marile provocări ale acestor modele nu constau în claritatea lor matematică, ci în depistarea erorilor, colectarea datelor de intrare, validarea lor și aplicabilitatea în software real, la utilizator.

În timpul proiectării și codificării se pot introduce erori care pot fi grupate în următoarele categorii:

- i. erori legate de alegerea și descrierea algoritmului:
 - algoritm incorect, sau inadecvat problemei de rezolvat;
 - algoritm mai puțin performant ca precizie sau timp necesar rezolvării problemei;
 - omiterea, interpretarea greșită sau incompletă a unor părți ale algoritmului;
 - validarea incorectă și/sau incompletă a datelor de intrare;
 - inversarea răspunsurilor la un bloc de decizie etc.
- ii. erori în definirea și utilizarea datelor ce provin din variabile neinițializate, formate impropriu de citire, contoare de capacitate insuficientă, neverificarea datelor de intrare, aliniere/redefinire incorectă a câmpurilor, utilizarea unor cuvinte cheie ca variabile, variabile ilegale (formate prin concatenare sau despărțite între două linii de program);
- iii. erori de calcule care pot avea ca surse:
 - expresii complicate care fac posibile eroari necontrolabile;
 - conversii implicite de tip (cu eroare de conversie, rotunjire, trunchiere etc.);
 - neinterceptarea cazurilor de depășire/subdepășire a intervalului definit etc.
- iv. erori produse în tehnica de programare cum sunt:
 - variabile și structuri de date globale;
 - acces necontrolat la zone de memorie partajate;
 - interfețe program - subprogram nerespectate;
 - pasarea constantelor ca parametri transmiși prin adresă;
 - pasarea parametrilor de intrare/ieșire prin valoare;
 - automodificarea programului în timpul execuție;
 - utilizarea necontrolată a mai multor limbaje cu convenții de apel diferite etc.
- v. erori produse din neatenție situație în care logica de control e defectuoasă, apar salturi în afara limitelor programului, condiții logice compuse sau incorect negate, neprelucrarea primei sau ultimei înregistrări, neluarea în considerare a posibilității de existență a fișierelor goale (vide), neprelucrarea erorilor de intrare/ieșire, depășirea capacității stivei, adresare incorectă, necontrolarea indecșilor etc.
- vi. erori în contextul execuției datorate memoriei dinamice insuficiente sau nealocată, periferice neoperaționale, comunicare defectuoasă cu sistemul de operare.

O mare parte din erorile menționate mai sus depistate în faza de compilare a programului și sunt extrase în fișierul de ieșire, într-o formă specifică fiecărui compilator. Tot ca erori de codificare sunt considerate și cele detectate în faza de editare a legăturilor.

9.2. Tehnici pentru îmbunătățirea fiabilității software-ului bazate pe ciclul de viață al defectului software

În literatura de specialitate au fost propuse o serie de metode tehnice care să ajute la îmbunătățirea fiabilității software-ului, tehnici bazate pe ciclul de viață al defectului software.

Realizarea de software de înaltă calitate din perspectiva beneficiarului reprezintă o sarcină ce necesită mult efort din partea tuturor inginerilor software și a celor de fiabilitate.

Pentru obținerea de sisteme software fiabile pot fi utilizate fiind patru tehnici specifice:

i) *Prevenirea defectului* (erorii): pentru a evita, prin construcție, manifestările defectului.

ii) *Înlăturarea erorilor*: eliminarea erorilor care au fost detectate prin verificare și validare.

iii) Realizarea de soft *Tolerant la Defectări*: pentru a oferi, prin redundanță, servicii care satisfac specificațiile, fără a ține seama de erorile apărute sau în curs de manifestare.

iv) *Prognozarea defectului/căderii*: pentru a estima prezența erorilor și modul de manifestare, precum și consecințelor căderilor.

Mecanismele de prevenire a erorilor nu pot garanta evitarea tuturor defectelor software. Atunci când erorile sunt inserate în cod, înlăturarea erorilor reprezintă următorul mijloc de protecție. Două dintre abordările practice ale metodei de înlăturare a erorilor sunt: testarea softului și examinarea softului, ambele devenind practici standard industriale în asigurarea fiabilității softului.

Atunci când erorile intrinseci rămân nedetectate în cadrul proceselor de testare și inspecție (examinare), aceste erori vor rămâne în cod atunci când acesta este lansat. Toleranța la erori reprezintă ultima linie de apărare pentru prevenirea defectelor în a se manifesta ca defectări ale sistemului. Toleranța la erori reprezintă atributul de supraviețuire al sistemelor software, datorită abilității de a oferi permanent servicii fiabile către beneficiari.

Tehnicile de toleranță la erori software permit sistemelor software:

i) să prevină ca erorile software latente să nu fie activate prin programarea defensivă, în care se verifică parametrii de intrare/ieșire și se blochează operațiile ilegale;

ii) să blocheze erorile software manifestate, între anumite limite fără posibilitatea de propagare mai departe, de exemplu prin rutinele de tratare a excepțiilor prin care se tratează operațiile fără succes;

iii) să restabilească operațiile software blocate de condițiile eronate prin mecanismele de verificare de tip "checkpoint" și de restabilire a condițiilor anterioare manifestării erorii și

(4) să tolereze sistematic erorile la nivel de sistem, de exemplu prin aplicarea diversității în proiectare în procesul dezvoltării produsului software.

Dacă defectările software se vor manifesta este important ca acestea să fie evaluate și anticipate prin predicții. Prognozarea erorii/căderii implică formularea relației eroare-cădere, cunoașterea profilului operațional, stabilirea modelelor de fiabilitate software, dezvoltarea de proceduri și mecanisme pentru măsurarea fiabilității software-ului și analiza și evaluarea rezultatelor măsurătorii.

Capacitatea de determinare a fiabilității software-ului nu ne oferă numai o orientare referitor la calitatea software-ului și despre momentul la care testarea poate fi oprită ci oferă informații pentru necesitățile de mentenanță software. Ca metodă de creștere a fiabilității software se consideră și mentenanța planificată, ca tehnică de reînnoire software.

9.3. Modele structurale pentru fiabilitatea software

Există mai multe modele și metode disponibile pentru determinarea fiabilității și pentru măsurarea caracteristicilor softului în toate etapele ciclului de viață al softului. Toate modelele de fiabilitate au dezvoltate pentru potrivirea la cerințele de fiabilitate ale utilizatorilor, pe baza datele care au putut fi colectate pentru intrările

modelului. Validitatea și acuratețea modelului aplicat și rezultatelor la ieșire depind de ipotezele făcute în formularea modelelor și de relevanța ieșirilor obținute.

Au fost dezvoltate modele pentru estimări pentru toate etapele din ciclul de viață al softului. Există modele predictive pentru faza de proiectare, modele pentru a determina timpul necesar pentru testare înainte de lansarea soft-ului, pentru predicția fiabilității soft-ului înainte de scrierea codului. În astfel de cazuri datele de intrare sunt bazate pe date de istoric de la sisteme și aplicații similare.

Alte modele au fost dezvoltate pentru perioada de testare pentru estimarea tendințelor de creștere a fiabilității pe baza datelor de intrare necesare testării sau bazează pe datele raportate despre comportarea în funcționare a softurilor, pe erorile constatate și durate dintre manifestarea erorilor. Nu există un model capabil să acopere întregul ciclu de viață al soft-ului.

În practică, sunt încercate și folosite mai multe modele pentru același soft, pentru fiecare etapă din ciclul de viață. Pentru a verifica și selectarea ce model se potrivește cel mai bine cu setul de date din observații se folosesc și tehnici statistice, de genul *goodness of fit*. Majoritatea modelelor de fiabilitate a soft-ului sunt executate automat datorită necesității de calcul iterativ. Interpretarea rezultatelor modelelor de fiabilitate necesită specialiști cu experiență practică și expertiză în modelarea fiabilității.

În acest capitol ne vom ocupa de modele pentru etapa de proiectare, urmând ca a modele pentru etapele de testare și utilizare să fie prezentate în capitolul următor.

Modelele structurale permit predicția fiabilității programelor încă din faza de proiectare și dezvoltare a unui program. Un astfel de model structural este propus de standardul militar american MIL HDBK 217/1976 [37] și deși a trecut mult timp de la elaborarea acestui model este încă folosit de specialiștii din întreaga lume cu rezultate foarte bune. Conform acestor modele, pe baza unei analize funcționale, programul este descompus în module iar pentru fiecare modul se calculează funcția individuală de fiabilitate, ținând seama de caracteristicile funcționale ale programului și de factorii externi care influențează fiabilitatea sa.

Funcțiile de fiabilitate individuale ale modulelor sunt apoi combinate ținând seama de structura programului și de constrângerile impuse de misiunea acestui program.

Funcția de fiabilitate asociată unui modul m_i are forma:

$$R_{m_i} = R_i + E_i(1 - R_i), \quad (9.3.1)$$

unde: R_i - funcția de fiabilitate intrinsecă a modulului i ;

E_i - un coeficient, care depinde de metodele folosite în elaborarea programului, astfel încât numărul de erori latente să fie cât mai mic.

Funcția de fiabilitate intrinsecă a modulului (R_i) depinde de interfețe, de dimensiunea programului, exprimată în linii de cod, de experiența grupului care a elaborat programul. MIL HDBK 217/1979 propune cu următoarea formulă de calcul:

$$R_i = I \cdot S \cdot O, \quad (9.3.2)$$

unde: I = nivelul primar al fiabilității,

S = factor dependent de complexitatea programului cu valori cuprinse între 0 și 1 iar

O = factor dependent de experiența elaboratorului.

Evaluarea nivelului primar al fiabilității (I) se face asociind fiecărui modul un coeficient X_n , cu n luând valori de la 1 la 7, fiecare indice având o semnificație anume, respectiv:

- 1 - tipul aplicației,
- 2 - complexitatea funcțională,
- 3 - interacțiunile,
- 4 - interfața hardware,
- 5 - interfața software,
- 6 - interfața cu operatorul,
- 7 - variabilitatea datelor de intrare.

Formula de calcul utilizată pentru calculul primar al fiabilității este:

$$I = \prod_{n=1}^7 X_n^{1/n}. \quad (9.3.3)$$

Coeficientul E_i din relația 9.3.1 se calculează cu formula:

$$E_i = \frac{A}{1 - D(1 - A)}, \quad (9.3.4)$$

unde: A este un coeficient care exprimă în ce măsură metoda folosită pentru elaborarea programului asigură prevenirea apariției erorilor cu valori cuprinse între 0 și 1 iar

D - alt coeficient care exprimă capacitatea de detectare și eliminare a erorilor, de asemenea cu valori cuprinse în intervalul $[0, 1]$.

Pentru limbaje avansate de proiectare, coeficienții A și D pot avea valori foarte apropiate de 1.

Admițând că modelarea programelor se poate face utilizând un proces semi-Markov (prezentat la pct. 5.4 din prezenta carte) și considerând că procesul de manifestare a erorilor latente într-un modul de program este un proces Poisson, rata de apariție a erorilor într-un program complex este de forma:

$$h = \sum_i t_i \alpha_i + \sum_{i,j} b_{ij} \lambda_{ij}, \quad (9.3.5)$$

unde: t_i = este fracțiunea de timp petrecută în modulul i din timpul total de execuție al modulului respectiv (la limită);

α_i = parametrul procesului Poisson, legat de funcția de fiabilitate a modulului i :

$$\alpha_i = \ln R_{m_i}, \quad (9.3.5')$$

b_{ij} = frecvența transferului de control de la i la j ,

λ_{ij} = probabilitatea de manifestare a unei erori atunci, când modulul i apelează la modulul j .

Rezolvarea ecuațiilor (9.3.5) se poate face numai aproximativ, nu se pot cunoaște exact toate valorile λ_{ij} și nu se poate ține cont de interfețele dintre module, care introduc potențiale noi surse de defectare.

Fiabilitatea întregului program se calculează apoi apelând la una din metodele de calcul a fiabilității unui sistem complex prezentate în capitolul 4 al acestei cărți, de exemplu schemele logice de fiabilitate sau metoda arborilor de defecte.

9.4. Tehnici și modele pentru sisteme software tolerante la defectări

Toleranța la defectări, atunci când se poate aplica, reprezintă una dintre abordările principale de a obține sisteme software de înaltă fiabilitate. Există două categorii diferite pentru tehnicile toleranței la defectări:

- tehnici software cu o singură versiune și
- tehnici software cu versiuni multiple.

Cea din dintâi, metoda cu o singură versiune, înglobează modularitatea programului, proprietatea de sistem închis, atomicitatea acțiunilor, detecția erorilor, tratamentul excepțiilor, utilizarea punctelor de control și a repornirii execuției, perechi de procese și diversitatea datelor [32].

Metoda cealaltă, cu versiuni multiple, așa numită *diversitatea proiectării*, este implementată atunci când mai multe versiuni ale softului sunt dezvoltate independent de către echipe diferite de programare, utilizând metode diferite de proiectare, dar care totuși furnizează servicii echivalente, conform cu aceleași specificații ale cerințelor. Tehnicile principale ale acestei abordări a software-ului cu versiuni multiple sunt reprezentate de *blocurile cu restabilire*, *structurile N-versionale*, *structurile N-autotestabile* precum și alte variante bazate pe aceste trei metode fundamentale.

9.4.1. Blocuri cu restabilire

Blocurile cu restabilire sunt structuri insensibile la erori, realizate printr-un procedeu de mascare a erorilor potențiale prin elaborarea mai multor variante ale aceluiași program, corespunzătoare aceluiași specificații, dar în care probabilitatea apariției aceleiași erori să fie foarte mică. Sunt structuri inspirate din sistemele redundante de comutație (stand-by), în care un element de rezervă este conectat în cazul defectării elementului de bază.

Structura unui bloc cu restabilire conține o variantă primară a programului și una sau mai multe variante secundare, mai simple, care oferă o prelucrare mai sumară a datelor, dar care au o fiabilitate mai ridicată (de multe ori versiuni mai vechi și verificate ale aceluiași program).

În cazul manifestării unei erori în varianta primară a programului, intră în funcțiune prima variantă secundară. O eroare în această variantă activează o nouă variantă secundară și tot așa în continuare.

Ieșirea din varianta primară poate avea loc înaintea terminării ei, dacă eroarea este evidentă sau numai în final când se efectuează testul de acceptare. Același test de acceptare este utilizat pentru verificarea corectitudinii rezultatelor tuturor variantelor. În figura 9.1 este prezentată o schemă logică de funcționare a blocului cu restabilire, conform [16].

Pentru blocuri cu restabilire este important de stabilit punctul de reluare, care permite restaurarea stării programului și păstrarea unei copii a stării sistemului la intrarea în fiecare bloc cu restabilire. Memoria rapidă dintr-un sistem de calcul, memoria *cache*, păstrează valorile inițiale ale variabilelor modificate în timpul executării blocurilor cu restabilire iar, în cazul manifestării unei erori, permite revenirea în punctul de reluare cel mai apropiat. Structura de tip bloc cu restabilire pentru un program poate fi compusă din mai multe blocuri cu restabilire pentru module ale acestuia sau pentru subprograme.

Testul de acceptare executat după terminarea fiecărei variante decide dacă blocul cu restabilire este părăsit sau se readuce sistemul la starea inițială și se execută o variantă secundară.

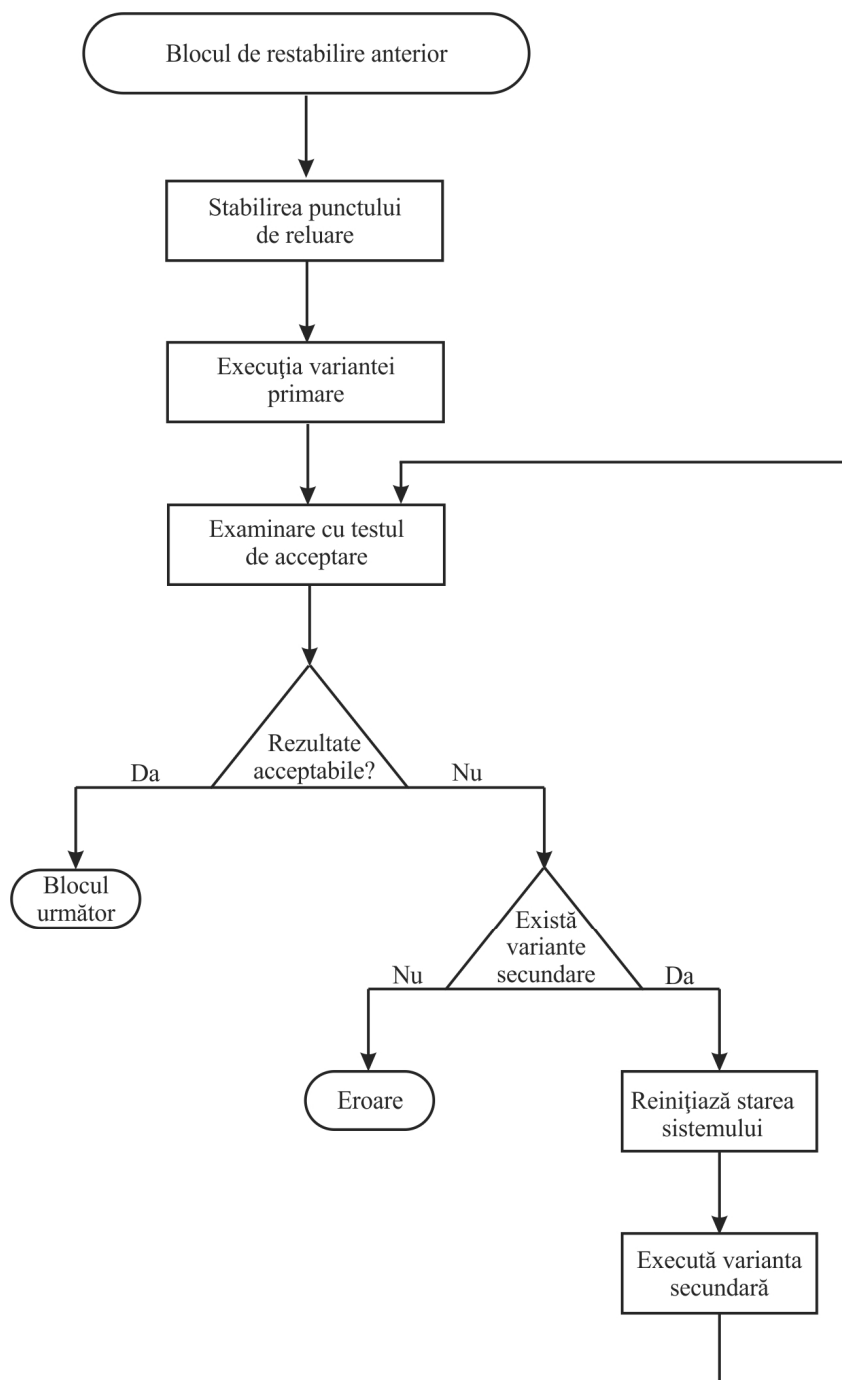


Figura 9.1. Schemă logică de funcționare a unui bloc cu restabilire

În continuare, este prezentată pe scurt o analiză a fiabilității unui bloc cu restabilire care conține două variante de rezolvare, o variantă primară (P) și una secundară (R), utilizând un model de tip semi-Markov. O analiză detaliată este făcută în [16].

În figura 9.2 este prezentat modelul stărilor unui bloc cu restabilire pentru studiul fiabilității pe baza lanțurilor de tip semi-Markov.

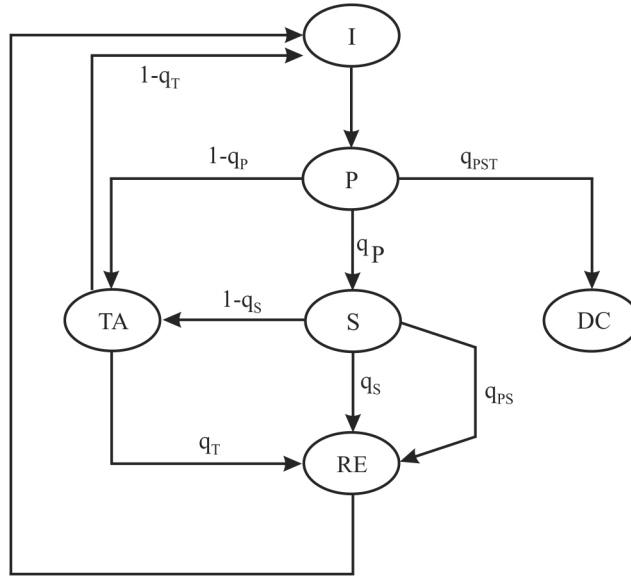


Figura 9.2. Modelul stărilor pentru studiul fiabilității unui bloc cu restabilire

Starea I corespunde stării inițiale a blocului neactivat.

Starea P corespunde executării variantei primare.

Starea S corespunde executării variantei secundare, în cazul eșecului variantei primare.

Starea TA corespunde executării testului de acceptare;

Starea DC este starea de defectare catastrofică, în care toate versiunile au eșuat dintr-o eroare comună, iar testul de acceptare este și el afectat de eroare. Probabilitatea de a ajunge în starea DC este notată cu q_{PST} .

Starea RE corespunde unor rezultate eronate ale ambelor versiuni, dar detectate de testul de acceptare. Ea se atinge atunci când ambele versiuni eșuează din cauza unor erori comune, erori care se manifestă cu probabilitatea q_{PS} , sau din cauza unor erori independente descrise prin q_P - probabilitatea de activare a unei erori în varianta primară, q_S - probabilitatea de activare a unei erori în varianta secundară și q_T - probabilitatea de nesesizare a erorii de către testul de acceptare.

Eșecul versiunilor este sesizat de testul de acceptare, iar sistemul este readus în starea inițială pentru a fi cuplat în alt bloc de restabilire.

În starea RE se poate ajunge și dacă testul de acceptare nu recunoaște corectitudinea rezultatelor furnizate de varianta primară P, respectiv de varianta secundară S.

Dacă se admite că se poate manifesta un singur tip de eroare în blocul cu restabilire și că nu au loc compensații ale erorilor, rezultă că probabilitatea de a atinge una din stările de defectare este:

$$\begin{aligned} Q_{BR} &= q_{PST} + q_{PS} + q_P \cdot q_S + (1 - q_S) \cdot q_P \cdot q_T + (1 - q_P) \cdot q_T = \\ &= q_{PST} + q_{PS} + q_T + (1 - q_T) \cdot q_P \cdot q_S. \end{aligned} \quad (9.4.1)$$

Într-un sistem de programare în care nu există sisteme redundante și nici posibilități de detecție a erorilor, probabilitatea de defectare a sistemului va fi:

$$Q = q_{PST} + q_{PS} + q_P. \quad (9.4.2)$$

Impunând condiția ca $Q_{BR} < Q$, adică probabilitatea de defectare a sistemului cu restabilire să fie mai mică decât a unui sistem neredundant de aceeași dimensiune, se poate scrie:

$$q_T + (1 - q_T)q_P \cdot q_S < q_P \Rightarrow q_T < \frac{q_P(1 - q_S)}{1 - q_Pq_S}. \quad (9.4.3)$$

Această condiție impune ca testul de acceptare al sistemului să fie mai simplu și mai fiabil decât cel pentru variantele primare și secundare, astfel încât probabilitatea de defectare a sistemului datorită testului de acceptare să fie cât mai mică.

9.4.2. Structuri N-versionale

Sunt structuri care utilizează un număr impar de versiuni concepute în mod independent, conform aceluiași set de specificații și care pot funcționa în mod independent [16, 23, 31].

Sunt tot structuri insensibile la erori.

Versiunile sunt activate de un modul supervisor, numit **driver**, care furnizează datele printr-o facilitate numai de citire (read only), colectează ieșirile variantelor și le combină conform unei reguli de tipul **votării majoritare**.

Rezultatul sintetic corespunde majorității rezultatelor oferite de versiunile independente, iar erorile, care se presupune că afectează variantele minoritare, nu influențează rezultatul de la ieșire.

Structura N-versională depinde de calitatea procesului de votare, ea poate fi îmbunătățită prin includerea în specificațiile versiunilor a unor valori intermediare de control care vor fi furnizate driver-ului odată cu rezultatele de ieșire. Confruntând valorile intermediare, voterul va detecta și erori în stările interne ale versiunilor, nu numai în stările lor externe, ceea ce limitează independența versiunilor care trebuie să convergă toate în punctele de control ale valorilor intermediare.

Modulul driver trebuie să țină seama de duratele diferite de execuție ale versiunilor, astfel încât decizia să nu depindă de diferențele dintre duratele de execuție, votarea să fie făcută în timp real sau rezultatele unor versiuni să fie păstrate în așteptarea celorlalte rezultate.

În sistemele critice din punct de vedere al siguranței se preferă o decizie bazată pe unanimitate nu pe majoritate. În cazul, în care rezultatele tuturor versiunilor nu sunt echivalente, sistemul intră într-o stare de "defectare nepericuloasă".

În cartea *Când calculatoarele greșesc* [16] profesorul A. Mihalache calculează probabilitatea de defectare a unei structuri 3-versionale, făcând o analiză comparativă între performanțele structurii cu restabilire și a celei 3-versionale (pag. 137÷142), utilizând modelul semi-Markov în două variante:

I. Versiunea afectată de erori nu este eliminată din structură ci este reutilizată după revenirea într-o stare compatibilă cu a celorlalte și

II. Versiunea afectată de erori este eliminată din structura 3-versională.

Pentru sistemul considerat, sunt identificate 5 tipuri de erori prezentate sintetic în tabelul 9.1. împreună cu probabilitățile de manifestare a acestor erori.

Dacă versiunile dau rezultate net distincte între ele sau dacă driver-ul nu recunoaște similaritatea lor, se ajunge în starea de defectare, *RE*, recunoscută ca atare, iar starea inițială a versiunilor este restabilă.

Tabelul 9.1

Nr. crt.	Tipul de eroare	Probabilitatea de manifestare
1	Eroare comună tuturor versiunilor și driver-ului	q_{VD}
2	Eroare comună celor 3 versiuni	q_{3v}
3	Eroare comună în oricare 2 versiuni	q_{2v}
4	Eroare independentă într-o versiune	q_{1v}
5	Eroare independentă în driver	q_D

Stările structurii, în ipoteza eliminării versiunii eronate, sunt prezentate în figura 9.3.

În cazul în care se obțin două sau chiar trei rezultate similare dar eronate, driver-ul nu poate detecta incorectitudinea lor și se ajunge în starea de defectare, C , ignorată de sistem.

Celelalte stări ale sistemului sunt:

I - starea inițială,

V - starea în care se execută cele 3 versiuni,

D_1 - rezultatele celor 3 versiuni sunt echivalente,

D_2 - eroare independentă manifestată într-o versiune,

D_3 - erori independente manifestate în 2 sau 3 versiuni,

D_4 - erori dependente manifestate în 2 sau 3 versiuni sau eroare comună manifestată în toate modulele (3 versiuni + driver).

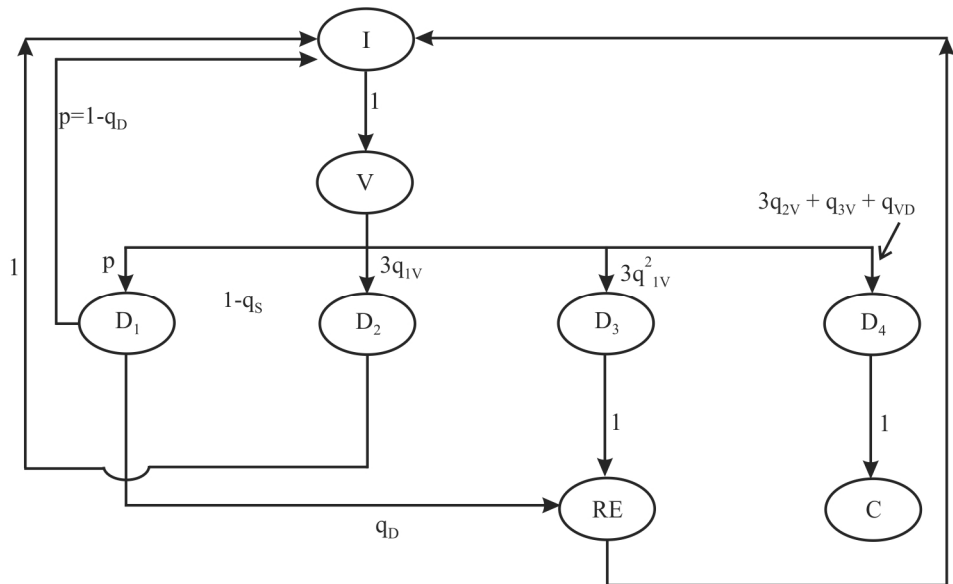


Figura 9.3. Modelul semi-Markov pentru analiza fiabilității structurii 3-versiionale

Probabilitatea de tranziție din starea V în starea D_2 , deoarece ea corespunde manifestării erorii doar în una din cele 3 versiuni, este egală cu:

$$3q_{1V}(1-q_{1V})^2 \approx 3q_{1V}. \quad (9.4.4)$$

Tranziția din starea V în starea D_3 se face cu probabilitatea ca în oricare două versiuni sau în toate 3, să se manifeste erori independente, adică:

$$3q_{1V}^2(1-q_{1V}) + q_{1V}^3 \approx 3q_{1V}. \quad (9.4.5)$$

Probabilitatea de tranziție din starea V în starea D_4 se face cu probabilitatea de manifestare a erorilor comune în oricare două versiuni sau în toate modulele, inclusiv driver-ul:

$$3q_{2V} + q_{3V} + q_{VD}. \quad (9.4.6)$$

Probabilitatea de defectare a structurii este:

$$Q_V = pq_D + 3q_{1V}^2 + 3q_{2V} + q_{3V} + q_{VD}, \quad (9.4.7)$$

unde p este probabilitatea ca în toate versiunile să nu se manifeste nici o eroare (probabilitatea de succes a structurii).

Modelul semi-Markov din figura 9.3 este realizat considerând că în timpul execuției se manifestă un singur tip de eroare dintre cele prezentate în tabelul 9.3 și nu există compensări ale erorilor.

9.5. Teste de acceptare a rezultatelor

Testele de acceptare, așa cum am prezentat în capitolul 9.4.1 se execută după finalizarea unui anumit program (o versiune) sau secvență de program în scopul:

- detectării de deviații față de comportarea prevăzută a programului;
- prevenirii furnizării unor rezultate greșite care să fie utilizate în continuare, evitând astfel prelucrări inutile sau decizii periculoase.

Nu se aplică numai în structurile cu restabilire ci și unor altfel de structuri și, întotdeauna, programelor noi care sunt supuse unor teste de acceptare foarte severe, sau celor pentru care se solicită un nivel de fiabilitate ridicat.

Principalele tipuri de teste de acceptare sunt:

- teste de rezonabilitate,
- teste de satisfacerea cerințelor,
- teste contabile.

a) **Teste de rezonabilitate** - examinează dacă rezultatele furnizate de un program (versiune, bloc), se încadrează în gama de valori admisibilă, urmăresc deviațiile variabilelor față de valoarea medie, corelațiile dintre diverse variabile etc.

În mod uzual se stabilește mulțimea rezultatelor eronate dar aceasta nu poate fi niciodată completă. Un test de acceptare evaluează funcția de apartenență a mărimii de ieșire la această mulțime și, dacă valoarea acestei funcții nu aparține mulțimii, rezultatul se validează.

Funcția de apartenență se evaluează subiectiv pentru fiecare intrare posibilă și se mediază pe mulțimea datelor de intrare, în conformitate cu legea de distribuție care descrie frecvența lor relativă de apariție.

Se aplică, în special, pentru programele care controlează variabilele fizice în timp real.

b) Teste de satisfacere a cerințelor

Aceste teste se referă la cerințele clar formulate în definirea problemei pe care programul trebuie să o rezolve.

Sunt eficiente pentru verificarea pe segmente nici ale unui program de calcul, pentru care cerințele pot fi simplu și clar formulate. Acolo unde cerințele sunt ample și greu de explicitat, aplicarea acestor teste nu este practică. Uneori testele de acceptare sunt foarte complexe.

Exemple:

1) Un program de ordonare a unor elemente ar necesita un test de acceptare care să verifice faptul că mulțimea de ieșire este o permutare a mulțimii de intrare. Un astfel de test ar fi mai complex decât programul de testat și atunci se aplică un test mai simplu, care verifică dacă mulțimile de intrare și de ieșire au același număr de elemente verificând în același timp uniformitatea regulii de succesiune în mulțimea de ieșire.

2) În cazul în care se utilizează teste de acceptare care constau în inversarea operațiilor efectuate de program și compararea rezultatelor obținute cu datele de intrare, apar dificultăți legate de faptul că inversa unei funcții nu este totdeauna ușor de calculat, iar unele funcții realizate de program nu au inversă.

Astfel de teste de satisfacere a cerințelor precizate sunt practice mai ales în cazul programelor de editare și compilare.

c) **Teste contabile** - se utilizează în programe care supervizează tranzacții, utilizând principiul contabilității duble și anume că "totalul creditelor din toate conturile trebuie să fie egal cu totalul debitelor din toate conturile, în orice interval de timp", cu condiția ca tranzacțiile din conturi să fi fost corect înregistrate.

În unele cazuri, inventarul fizic se modifică în timp, cum este cazul depozitelor de material nuclear unde stocul existent este estimat pe baza măsurărilor de câmp de radiații. La intervale prestabilite modificarea nivelului radiațiilor din depozit se compară cu modificarea rezultată în urma îmbătrânirii materialelor și a tranzacțiilor autorizate [1].

Aceste teste sunt completate cu o serie de verificări, urmând proceduri speciale de testare și validare.

Cele mai multe calculatoare sunt prevăzute cu **programe de autotestare** continuă a stărilor anormale ale sistemului care sunt semnalizate operatorului în timp real, utilizând un bit de stare sau mai mulți, aplicând tehnici de detectare și corectare a erorilor (bitul de paritate, codul Hamming, Codurii corectoare de erori, (de exemplu CRC – Cyclic Redundancy Checking, FEC Forward Error Corection).

În sistemul de operare sunt incluse, în mod curent, structuri de date și teste integrate, care permit **autotestarea** și **monitorizarea** specială de exemplu cu "supervizorul de interacțiuni", care cer declararea pentru fiecare modul a apelurilor autorizate și a surselor acestor apeluri. În situația în care accesul sau ieșirea dintr-un modul se referă la adrese neautorizate, supervizorul ia o decizie de respingere.

Încercările de scriere în zone protejate ale memoriei sunt controlate în timp real și împiedicate.

Capitolul 10. MODELE PENTRU FIABILITATEA SOFTWARE ÎN FAZA DE TESTARE ȘI OPERARE

10.1. Generalități

Modelele utilizate pentru estimarea fiabilității software în fazele de testare și în faza de operare cu remediarea defectelor sesizate sunt modele bazate pe timpul mediu dintre defectări sau pe numărarea defectelor. Aceste modele tratează programul ca pe un produs, ca pe o cutie neagră fără să țină seama de structura acestuia și de procesul de codificare și dezvoltare. În fazele de testare și depanare se consideră că erorile detectate sunt corectate și că nu sunt introduse noi erori la intervenția în program. Pentru estimarea fiabilității în faza de proiectare există modele specifice bazate pe procedee formale. Am prezentat câteva astfel de modele în capitolul precedent.

Modelele se pot alege în funcție de modul de testare ales. Dacă se utilizează o metodă de testare aleatoare pot fi utilizate atât modele bazate pe numărarea defectelor cât și modelele bazate pe timpul dintre defectări, considerând că pe măsură ce sunt eliminate erori, intervalul dintre defectări crește. Dacă se utilizează o testare deterministă, de exemplu orientată pe câte o cale a programului, se preferă modelele bazate pe urmărirea defectelor, întrucât independența intervalelor între defectări nu mai este respectată.

În faza de operare a programelor, intrările nu mai pot fi considerate aleatoare, ele sunt corelate deoarece se utilizează în mod uzual numai anumite căi din program. Pentru calculul fiabilității în această fază s-au dezvoltat metode și tehnici speciale și au fost dezvoltate modele bazate pe numărarea defectelor.

A apărut o știință nouă, numită Ingineria fiabilității, care studiază fiabilitatea pe întreg ciclul de viață al unui produs, dar nu numai aplicând metode matematice sau statistice, ci și din prisma managementului, a costurilor și riscurilor care pot fi generate de produse non-fiabile.

În scopul studierii fiabilității software și reducerii riscurilor de apariție de erori concomitent cu reducerea costurilor realizării de software de înaltă fiabilitate se dezvoltă proceduri specifice.

Prima etapă în aceste proceduri este *stabilirea unui obiectiv* al fiabilității din perspectiva beneficiarului pentru a maximiza satisfacția acestuia și se definește modul de utilizare al softului de către beneficiar. Se dezvoltă apoi un așa numit *profil operațional*.

Apoi software-ul este *testat* conform cu profilul operațional stabilit, sunt colectate datele de defectare și se studiază fiabilitatea în timpul testării pentru a se determina punctele în care trebuie intervenit și timpul de lansare al produsului. Această activitate este repetată până când se obțin date suficiente și se atinge un nivel de fiabilitate satisfăcător.

În final fiabilitatea este *validată* la utilizatorul final pentru a se obține date necesare îmbunătățiri viitoare ale software-ului. Vom reveni la testarea și validarea software în finalul acestui capitol, 10.4

În ciuda existenței unui număr mare de modele, problema selecției și aplicării unui model este o problemă complexă, existând diverse orientări și metode statistice pentru selecția unui model adecvat fiecărei aplicații. Experiența a arătat că pot fi luate în considerare numai câteva modele și, unele dintre acestea sunt deja implementate în diverse unelte software.

Utilizând metode statistice, se obțin “cele mai bune” estimări ale fiabilității în timpul testării. Acestea sunt apoi utilizate pentru a proiecta fiabilitatea în perioada operării pentru a determina dacă s-a atins sau nu obiectivul de fiabilitate. Acest proces este iterativ, el fiind reluat ori de câte ori este nevoie până când obiectivul fixat a fost atins.

Atunci când profilul operațional nu este dezvoltat în totalitate, se poate aplica un *indice de acoperire al testării* pentru estimarea fiabilității pe teren, indice care este definit prin raportul dintre timpul de execuție necesar în faza operațională și timpul de execuție necesar în etapa de testare, pentru a se acoperi întreg vectorul de intrare al programului.

Indicele de acoperire al testării reprezintă scăderea ratei de defectare (sau creșterea fiabilității) în perioada de operare, relativ la fiabilitatea măsurată în timpul testării, deoarece personalul de testare caută rapid, prin vectorul de intrare, ambele condiții de execuție, normală și dificilă, în timp ce utilizatorii execută programul software pas cu pas.

Fiabilitatea proiectată trebuie să fie validată prin compararea acesteia cu cea observată. Această validare nu doar stabilește criteriile de referință și nivelele de încredere pentru estimările fiabilității ci oferă un feedback pentru procesul ingineriei fiabilității software-ului, în vederea îmbunătățirii continue și a ajustării cât mai bune a parametrilor. Când este asigurat un feedback, îmbunătățirea procesului vine în mod natural: se validează modelul, se determină creșterea fiabilității și se îmbunătățește indicele de acoperire al testării.

Cei mai mulți producători de software utilizează teste numite *Alpha Test* și *Beta Test* pentru a descoperi erori pe care numai utilizatorii finali le descoperă.

Testul Alpha se realizează de către o serie de clienți selectați, este condus de către dezvoltătorii de software și se realizează, de obicei, într-un mediu controlat. Aplicația este utilizată având în spate dezvoltatorul pentru a înregistra erorile și problemele apărute.

Testul Beta este făcut de unul sau mai mulți clienți finali fără nici un control din partea dezvoltatorului. Acesta este un test într-un mediu necontrolat (ambient real) în care clientul înregistrează toate problemele reale sau imaginare și pe care le

raportează la intervale regulate către dezvoltător. În schimb dezvoltatorul îi poate oferi produsul fără plata costurilor, înainte ca utilizatorii ceilalți să-l poată utiliza.

10.2. Modele bazate pe timpul mediu dintre defectări

10.2.1. Modelul Jelinski-Moranda

Este unul dintre cele mai cunoscute modele pentru predicția fiabilității programelor.

Ipotezele modelului :

a) Intervalele de timp între defectările succesive sunt variabile aleatoare independente, distribuite după legi exponențiale cu parametrii diferiți;

b) Rata de defectare este proporțională cu numărul de erori latente ale programului și este o constantă pe un anumit interval, având forma:

$$z(t) = \lambda_k = \varphi \cdot N(t) = \varphi[N - (k - 1)] = \varphi[N - k + 1] \text{ pentru } t_{k-1} \leq t \leq t_k, (10.2.1)$$

unde:

– N este numărul inițial de erori $N(t=0)$ iar

– φ este o constantă de proporționalitate.

c) La fiecare defectare se efectuează o intervenție în program prin care se elimină o singură eroare, și anume, aceea pusă în evidență de către defectarea observată.

Variația în timp a ratei de defectare așa cum este prezentată în figura 10.1.

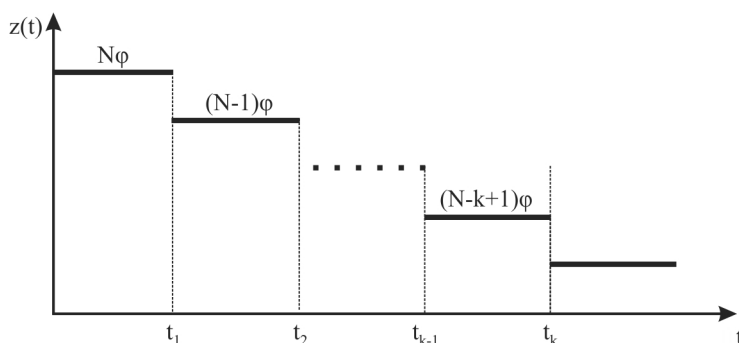


Figura 10.1. Variația în timp a ratei de defectare pentru Modelul Jelinski-Moranda

După cum se poate urmări în figură, rata de defectare scade la fiecare defectare a programului, deci fiabilitatea acestuia crește; se poate spune că este un proces de reînnoire cu reînnoiri negative.

Caracteristici și parametri de fiabilitate:

a) Este un model de creștere exponențială a fiabilității.

b) Funcția de fiabilitate și funcția densitate de probabilitate pe intervalul k sunt.

$$R_k(t) = e^{-(N-k+1)\varphi t} \quad (10.2.2)$$

$$f(t) = \frac{dR}{dt} = (N-k+1) \cdot \varphi \cdot e^{-(N-k+1)\varphi t_n} \quad (10.2.2')$$

c) Media timpului între defectările $k-1$ și k este:

$$m = \int_0^{\infty} R(t) dt = \frac{1}{(N-k+1) \cdot \varphi}. \quad (10.5.3)$$

d) Funcția de reînnoire este media numărului de defectări în intervalul $(0, t)$ și este dată de relația:

$$H(t) = N(1 - e^{-\varphi t}). \quad (10.2.4)$$

iar densitatea de reînnoire este

$$h(t) = \frac{dH(t)}{dt} = N\varphi e^{-\varphi t} \quad (10.2.4')$$

e) Durata medie până la eliminarea tuturor erorilor:

$$D(t) = \frac{1}{(N-n)\varphi} + \frac{1}{(N-n-1)\varphi} + \dots + \frac{1}{\varphi}, \quad (10.2.5)$$

unde n reprezintă primele defectări observate.

Estimarea parametrilor modelului (N și φ) se poate face utilizând metoda verosimilității maxime, considerând ca estimății punctuale perechea de valori $(\hat{N}, \hat{\varphi})$ care maximizează probabilitatea de apariție a rezultatelor experimentale obținute.

Funcția de verosimilitate este definită ca funcția de densitate de probabilitate reunită a variabilelor aleatoare:

$$L(t_1, t_2, \dots, t_n / N, \varphi) = \prod_{k=1}^n \varphi^n (N-k+1) e^{-\sum_{k=1}^n (N-k+1)\varphi t_k} \quad (10.2.6)$$

Funcția de verosimilitate își atinge maximul pentru acele valori în care derivatele parțiale în raport cu N și φ ale logaritmului funcției de verosimilitate se anulează:

$$\frac{\partial \ln L}{\partial N} = 0 \text{ și } \frac{\partial \ln L}{\partial \varphi} = 0 \quad (10.2.7)$$

Din rezolvarea acestor ecuații se obține:

$$\begin{cases} \sum_{k=1}^n \frac{1}{N-k+1} = \frac{nt_n}{Nt_n - \sum_{k=1}^n (k-1)t_k}, \\ \varphi = \frac{n}{Nt_n - \sum_{k=1}^n (k-1)t_k}, \end{cases} \quad (10.2.8)$$

unde am introdus notația:

$$t_n = \sum_{k=1}^n t_k \quad (10.2.8')$$

t_n reprezintă durata scursă de la începutul observațiilor până la înregistrarea defectării cu numărul de ordine n .

Rezolvarea ecuațiilor din sistem conduce la obținerea estimațiilor punctuale \hat{N} și $\hat{\varphi}$ care depind de numărul de defectări observate și de momentul la care se face estimarea.

Rezolvarea acestor ecuații și discuții referitoare la soluțiile obținute pot fi studiate în bibliografie [16]. Pentru ca soluțiile să fie acceptabile se impune acumularea unui număr foarte mare de date de la utilizatorii programului.

10.2.2. Modelul Goel-Okumoto I

Este un model care admite că depanarea programului s-a făcut incorect, respectiv că starea de defect a fost înlăturată dar fără a fi siguri că eroarea eliminată este cea care a adus sistemul în această stare, ci probabilitatea ca eroarea să fie eliminată prin intervenția exterioară în momentul apariției ei este p .

Ipotezele modelului :

a) intervalele de timp între defectările succesive sunt variabile aleatoare independente, distribuite după legi binomiale;

b) rata de defectare este proporțională cu numărul de erori latente rezidente în program:

$$z(t) = \varphi \cdot N(t) = \varphi [N - (k-1)]; \quad (10.2.1)$$

c) la fiecare defectare se elimină exact o eroare și anume aceea care a dus la defectarea programului;

d) probabilitatea ca o eroare să fie eliminată prin intervenția exterioară în momentul apariției ei este presupusă constantă și egală cu p .

Caracteristici și parametri de fiabilitate :

a) Probabilitatea eliminării a r erori în intervalul $(0, t)$, cu $r = 1, 2, \dots, N$ este:

$$P_r(t) = C_N^r (e^{-p\varphi t})^{N-r} (1 - e^{-p\varphi t})^r. \quad (10.2.2)$$

Aceasta este o distribuție binomială, având parametrii N , φ și p .

b) Numărul mediu de erori corectate coincide cu media distribuției binomiale, respectiv:

$$M_c(t) = N(1 - e^{-p\varphi t}). \quad (10.2.3)$$

c) Numărul de defectări observate în intervalul $(0, t)$ este legat de numărul de erori corectate, prin relația:

$$M_D = \frac{M_C}{p} = \frac{N}{p}(1 - e^{-p\varphi t}). \quad (10.2.4)$$

d) Numărul mediu de erori remanente, la un moment dat este:

$$M_r(t) = N e^{-p\varphi t}. \quad (10.2.5)$$

Probabilitatea ca, la un moment dat, numărul erorilor remanente să fie mai mic sau egal cu o valoare dată A este:

$$P(N(t) \leq A) = \sum_{k=0}^A Q_k(t) = \sum_{k=0}^A C_N^k (e^{-p\varphi t})^k (1 - e^{-p\varphi t})^{N-k}. \quad (10.2.27)$$

Pentru $A = 0$, rezultă că probabilitatea de eliminare a tuturor defectelor în intervalul $(0, t)$ este:

$$Q_0(t) = (1 - e^{-p\varphi t})^N, \quad (10.2.6)$$

iar durata medie până la eliminarea tuturor erorilor este:

$$D = \sum_{i=1}^N \frac{1}{i p \varphi}. \quad (10.2.7)$$

Predicția comportării programului este dificilă atunci când numărul curent de erori $N(t)$ nu este univoc determinat de numărul de defectări înregistrate, fiind necesar calculul distribuției numărului de erori reziduale în timp după înregistrarea unui număr oarecare de defectări.

Estimarea parametrilor modelului se poate face prin metoda celor mai mici pătrate, punând condiția ca suma pătratelor diferențelor dintre defectările înregistrate în fiecare interval de timp și valorile prezise de model să fie minimă.

Numărul mediu de defectări în intervalul (t_{i-1}, t_i) este dat de relația:

$$M_D(t_{i-1}, t_i) = M_D(t_i) - M_D(t_{i-1}) = \frac{N}{p}(e^{-p\varphi t_{i-1}} - e^{-p\varphi t_i}), \quad (10.2.8)$$

cu $i = 1, 2, \dots, n$.

Introducând notațiile:

$$M_0 = \frac{N}{p} \quad \text{și} \quad b = p\varphi, \quad (10.2.9)$$

expresia 10.2.30 se poate scrie:

$$M_D(t_{i-1}, t_i) = M_0 (e^{-bt_{i-1}} - e^{-bt_i}). \quad (10.2.10)$$

Suma pătratelor abaterilor dintre valorile experimentale și cele prezise este:

$$S(M_0, b) = \sum_{i=1}^L [n_i - n_{i-1} - M_0 (e^{-bt_{i-1}} - e^{-bt_i})]^2. \quad (10.2.11)$$

Parametrii M_0 și b_0 se estimează rezolvând sistemul de ecuații neliniare:

$$\begin{cases} \frac{\partial S(M_0, b)}{\partial M_0} = 0, \\ \frac{\partial S(M_0, b)}{\partial b} = 0, \end{cases} \quad (10.2.12)$$

iar parametrii modelului N și φ se calculează din relațiile:

$$\begin{cases} \hat{N} = \hat{p} \cdot \hat{M}_0, \\ \hat{\varphi} = \frac{b}{\hat{p}}. \end{cases} \quad (10.2.13)$$

10.2.3. Modelul Littlewood

Ipotezele modelului:

a) Fiecare eroare are ponderea sa proprie, φ_i , în rata de defectare a sistemului,
b) Intervalele între defectări sunt variabile aleatoare independente și identic distribuite după legi exponențiale,

c) Fiecare intervenție elimină cu certitudine o eroare din program.

Modelul permite estimarea numărului de defecte din program și creșterea fiabilității acestuia.

Caracteristici și parametri de fiabilitate :

a) Funcția de fiabilitate are forma:

$$R(t, t+x) = \frac{(\beta+t)^{\alpha(N-n)}}{(\beta+t+x)^{\alpha(N-n)}} = \frac{1}{\left(1 + \frac{x}{\beta+t}\right)^{\alpha(N-n)}}, \quad (10.2.14)$$

n = numărul erorilor manifestate până la un moment dat ,

$N - n$ = numărul erorilor reziduale la momentul respectiv.

α, β = parametrii distribuției.

b) Rata de defectare, după manifestarea a n erori, este egală cu suma ponderilor celor $N - n$ erori reziduale:

$$z(t) = \lambda_{n+1} = \sum_{i=1}^{N-n} \varphi_i \text{ pentru } t_k \leq t < t_{k+1}; \quad (10.2.15)$$

c) Durata medie de funcționare până la proxima defectare este dată de relația:

$$m = \int_0^{\infty} R(t, t+\Delta t) dx = \frac{\beta+t}{\alpha(N-n)-1}. \quad (10.2.16)$$

d) Funcția de reînnoire, adică numărul mediu de defectări în intervalul $(t, t+x)$ este de forma:

$$H(t, t+\Delta t) = (N-n) \left[1 - \left(\frac{\beta+t}{\beta+t+\Delta t} \right)^{\alpha} \right]. \quad (10.2.17)$$

Parametrii modelului sunt N , α și β se pot calcula utilizând funcția de verosimilitate după modelele prezentate anterior:

$$\ln L(x_1, x_2, \dots, x_n / N, \alpha, \beta) = n \ln \alpha + \alpha N \ln \beta + \sum_{i=1}^n \ln(N - i + 1) - (\alpha + 1) \sum_{i=1}^n \ln(\beta + t_i) - \alpha(N - n) \cdot \ln(\beta + t). \quad (10.2.18)$$

A existat o variantă mai veche a acestui model, numit *Littlewood-Verrall* care considera că o eroare constatată nu este sigur eliminată. Printre parametrii modelului nu era numărul total de erori latente N . Predicțiile în acest caz se referă numărul de erori ce pot fi estimate și la intervalul de timp dintre două erori sesizate sau până la apariția erorii.

10.2.4. Modelul Schick-Wolverton

Ipotezele modelului :

- Intervalele între două defectări sunt variabile aleatoare independente.
- Orice eroare este eliminată în momentul manifestării ei.
- Rata de defectare este proporțională cu numărul curent de erori latente și crește liniar cu durata de testare între două defectări succesive, revenind la zero la fiecare manifestare a unei erori și crește cu o pantă din ce în ce mai mică între două defectări succesive, conform relației:

$$z(t_n + x) = (N - n) \varphi \cdot x \text{ cu } 0 \leq x \leq t_{n+1} - t_n. \quad (10.2.19)$$

Dependența ratei de defectare de timp este prezentată în figura 10.2.

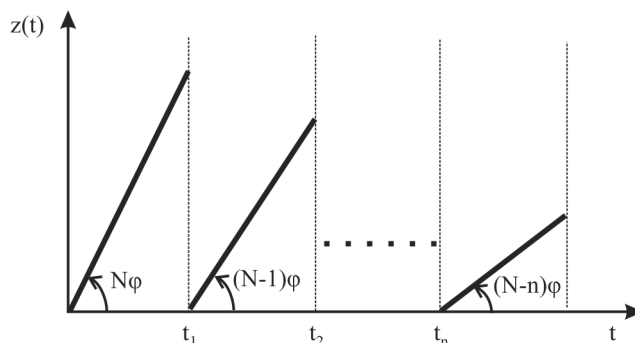


Figura 10.2. Variația ratei de defectare pentru modelul Schick-Wolverton

Caracteristici și parametri de fiabilitate :

- Funcția de fiabilitate este o funcție exponențială:

$$R(t_n, t_n + x) = \exp \left[- \int_0^x (N - n) \cdot \varphi \cdot x \cdot dx \right] =$$

$$= \exp\left[-(N-n)\frac{\varphi \cdot x^2}{2}\right], \quad (10.2.20)$$

unde: t_1, t_2, \dots, t_n - momentele primelor n defectări observate,
 N - numărul inițial de erori latente,
 φ - constanta de proporționalitate.

b) Durata medie a intervalului dintre două defectări succesive este:

$$m(t_n) = \sqrt{\frac{\pi}{2(N-n)\varphi}}. \quad (10.2.21)$$

c) Durata medie până la eliminarea tuturor defectelor:

$$D(t_n) = \sum_{k=n}^{N-1} \sqrt{\frac{\pi}{2(N-n)\varphi}} = \sum_{i=1}^{N-n} \sqrt{\frac{\pi}{2i\varphi}}. \quad (10.2.22)$$

Parametrii modelului sunt N și φ și ei pot fi estimați utilizând metoda verosimilității maxime, funcția de verosimilitate fiind în acest caz:

$$L(x_1, \dots, x_n / N, \varphi) = n \ln \varphi + \sum_{k=1}^n \ln(N-k+1) + \sum_{k=1}^n \ln x_k - \frac{\varphi}{2} \sum_{k=1}^n (N-k+1) x_k^2. \quad (10.2.22)$$

Există mai multe versiuni ale modelului Schick-Wolverton care se pot studia din bibliografie [16].

10.3. Modele bazate pe numărarea defectelor

10.3.1. Modelul Musa

Ipotezele modelului :

a) Intervalele de timp dintre defectările succesive sunt variabile aleatoare independente, distribuite după legi exponențiale cu parametri diferiți;

b) Rata de defectare este proporțională cu numărul de erori latente ale programului:

$$z(t) = \varphi \cdot N(t) = \varphi[N - (k-1)]. \quad (5.3.1)$$

c) La fiecare defectare se efectuează o depanare prin care se elimină tocmai eroarea pusă în evidență de către defectarea observată, dar eroarea se poate repeta.

d) Variabila timp din funcționare este timpul de execuție al unității centrale (t_{CPU}).

Se poate folosi atât pentru perioada de testare cât și pentru perioada de operare, introducând un factor specific acestui model denumit *factor de compresie*, c . Factorul de compresie este definit ca raportul între durata echivalentă de operare

și durata de testare și indică în ce măsură timpii de rulare din faza operațională au fost reduși prin proiectarea și selecția testelor; o oră de testare poate reprezenta câteva ore de operare, în faza de exploatare.

Caracteristici și parametri de fiabilitate:

a) Funcția de fiabilitate:

$$R(t_n, t_n + x) = \sum_{i=N-k}^N e^{-i\varphi x} C_k^{N-i} (1-q)^{k-N+1} \cdot q^{N-i} =$$

$$= e^{-N\varphi x} [1 - q + q \cdot e^{\varphi x}]^k. \quad (5.3.2)$$

unde: t_1, t_2, \dots, t_n - momentele de producere a n defecte succesive;

q - probabilitatea condiționată ca o eroare care nu s-a repetat să fi fost corectată. Expresia pentru q este:

$$q_j = \frac{p}{p + (1-p)e^{-\varphi(t_n - t_j)}}. \quad (5.3.3)$$

cu p probabilitatea ca o eroare să fie eliminată prin intervenția exterioară în momentul apariției ei. Este presupusă constantă

b) Numărul mediu de defectări înregistrate în intervalul $(0, t)$, ținând seama de factorul de compresie, este:

$$M_D(t) = \frac{N}{p} (1 - e^{-p\varphi c t}) = \frac{N}{1-q} (1 - e^{-(1-q)\varphi c t}) =$$

$$= \frac{N}{1-q} (1 - e^{-\frac{c t (1-q)}{m_0 \cdot N}}), \quad (5.3.4)$$

unde: $m_0 = \frac{1}{N\varphi}$ este durata medie până la prima defectare. (5.3.4')

Parametrii modelului sunt tot p , φ și N și se pot determina în mod similar cu parametrii modelului Goel-Okumoto I.

10.3.2. Modelul Shanthikumar

Ipotezele modelului :

a) Rata de defectare este proporțională cu numărul erorilor remanente:

$$z(t) = (N - k + 1) \cdot \varphi(t). \quad (10.3.5)$$

b) Factorul de proporționalitate este o funcție de timp:

$$\varphi = \varphi(t). \quad (10.3.6)$$

Este un model de creștere exponențială a fiabilității.

Caracteristici și parametri de fiabilitate :

a) Rata de defectare este de forma:

$$R(t, t+x) = e^{-(N-n)q_t(x)}, \quad (10.3.7)$$

unde

$$q_t(x) = \exp \left[- \int_t^{t+x} \varphi(t) dt \right] \quad (10.3.7')$$

reprezintă probabilitatea ca o eroare să nu se manifeste în intervalul de timp de urmărire, Δt .

b) Funcția de reînnoire este:

$$H(t) = N [1 - q(t)] = N \left[1 - e^{-\int_0^t \varphi(t) dt} \right]. \quad (10.3.8)$$

Parametrii modelului sunt N și $\varphi(t)$. Estimarea acestor parametri se poate face numai presupunând anumite forme pentru funcția $\varphi(t)$ și verificând experimental corectitudinea predicției făcute. Modelul poate fi adaptat și situației depanării imperfecte, presupunând că probabilitatea corectării unei erori în momentul depistării ei este p , iar factorul de proporționalitate este $p \cdot \varphi(t)$.

În cazul modelului Shanthikumar, procesul de manifestare a erorilor în intervalul $(0, t)$ este de tip binomial.

Dacă $\varphi(t) = \text{constant} = \varphi$ modelul Shanthikumar se reduce la modelul Jelinski-Moranda primul model studiat în acest capitol și cel mai mult utilizat în studiile internaționale de fiabilitatea software.

10.3.3. Modelul Goel-Okumoto II

Ipotezele modelului :

Modelul Goel-Okumoto II este definit ca un proces Poisson în care numărul erorilor inițiale este o variabilă aleatoare cu valori particulare foarte ridicate iar erorile detectate în intervale de timp disjuncte sunt independente între ele.

Caracteristici și parametri de fiabilitate :

a) Funcția de fiabilitate nu depinde de numărul de erori manifestate anterior momentului predicției și are forma:

$$R(t, t+\Delta t) = \exp(-a \cdot e^{-bt} - e^{-b(t+\Delta t)}), \quad (5.3.9)$$

unde: a și b sunt parametrii modelului și reprezintă:

a - numărul mediu de erori detectabile într-un interval de timp infinit de lung;

b - rata de detecție a erorilor, o constantă de proporționalitate, care leagă numărul mediu de erori detectate într-un interval de timp scurt, $(t, t+\Delta t)$, de numărul mediu de erori nemanifestate încă, în intervalul $(0, t)$;

b) Numărul mediu de erori manifestate în intervalul $(0, t)$ este funcția de reînnoire care are formă exponențială:

$$H(t) = a(1 - e^{-bt}). \quad (5.3.10)$$

c) Numărul mediu de erori ce se vor manifesta în intervalul $(t, t + x)$ este dat de relația:

$$H(t, t + \Delta t) = H(t + x) - H(t) = a(e^{-bt} - e^{-b(t+\Delta t)}). \quad (5.3.11)$$

10.4. Validarea fiabilității programelor

10.4.1. Etape în validarea fiabilității unui program

Adoptarea unui model pentru estimarea fiabilității programelor urmează următoarea schemă logică [12, 16]:

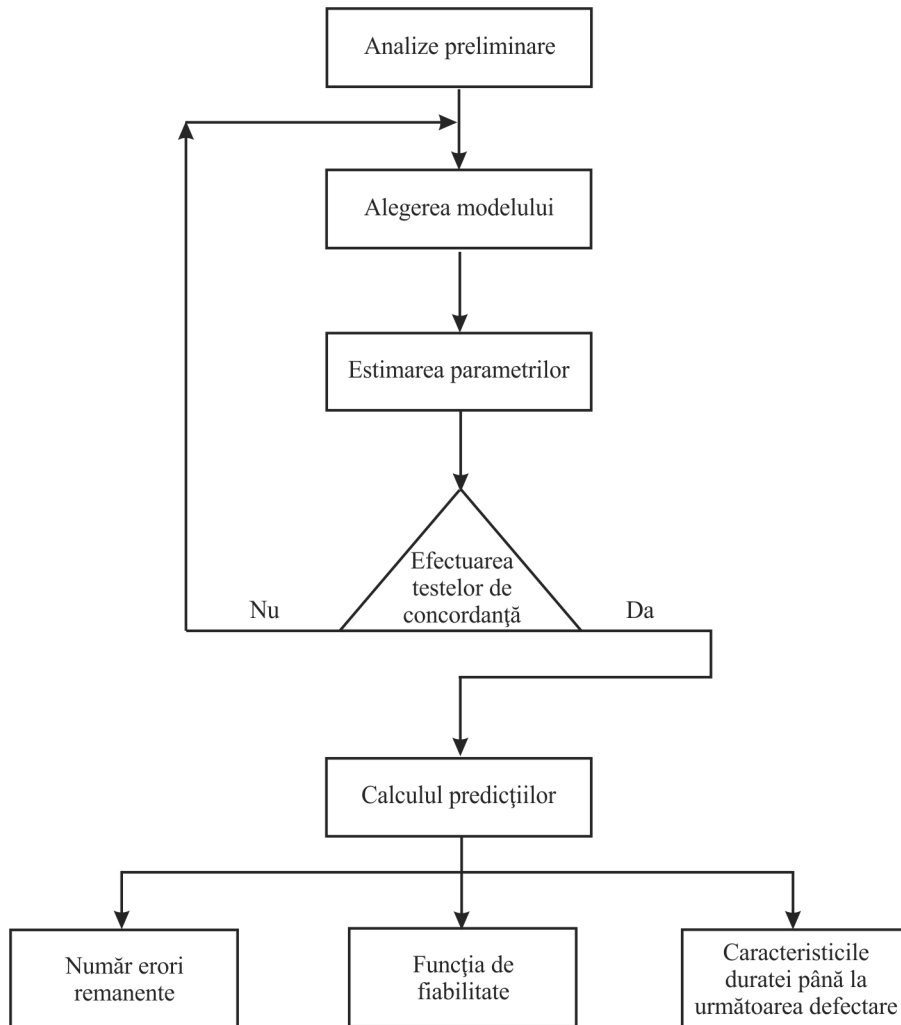


Figura 10.3. Schemă logică preliminară pentru analiza fiabilității programelor

Datele statistice de ieșire trebuie să fie obținute pentru sisteme suficient de complexe și corect înregistrate.

Pentru a selecta un anumit model pot folosite următoarele criterii:

- profile de defectare;
- maturitatea produsului de soft;
- caracteristicile dezvoltării soft-ului;
- caracteristicile testării soft-ului;
- metrice și date existente.

Pentru aplicarea modelului se recomandă utilizarea de instrumente performante de calcul care permit obținerea de rezultate corecte, economisire de timp și bani pentru implementarea lor și creșterea nivelului de încredere. Se recomandă ca datele să fie prelucrate cu mai multe instrumente, pentru mai multe modele, iar rezultatele să fie comparate pentru a determina care se potrivește mai bine.

Pentru selectarea unui instrument trebuie luate în considerare următoarele criterii:

- disponibilitatea unui instrument compatibil cu sistemul de calcul pe care rulează softurile;
- tipurile de softuri care urmează să fie studiate;
- număr de studii care pot fi făcute cu respectivul instrument;
- costurile de instalare și întreținere a softurilor studiate;
- ușurința de învățare și aplicare a instrumentelor;
- calitatea, flexibilitatea și puterea (precizia) instrumentelor;
- suportul tehnic al instrumentelor.

În funcție de datele obținute se poate aplica unul sau mai multe din modelele specifice, unele prezentate la începutul capitoului. Dacă se înregistrează numărul de erori cumulate în timp, fără a ține seama de duratele dintre defectări, pot fi utilizate, în special, modele de tip Poisson.

Pentru cazul în care se consideră durata dintre defectări se constată că sunt valabile distribuții exponențiale, logaritmice, binomiale sau Weibull.

Confruntarea empirică a unui model cu rezultatele experimentale constă în compararea directă a valorilor estimate de model cu cele direct observate. Pericolul constă în aceea că este posibil ca aceleași date experimentale să accepte ipoteze diverse, asociate unor modele diferite.

De exemplu ipotezele constanței ratei de defectare între două defectări, independența duratelor între defectări, proporționalitatea dintre rata de defectare și numărul de erori latente sunt ipoteze valabile pentru mai multe modele. Dacă aceste ipoteze sunt acceptate, atunci concordanța dintre numărul mediu de erori permis de model și numărul cumulat de erori efective, observate, poate valida mai multe modele.

Așa cum se poate urmări în figura 10.3, etapa esențială în adoptarea unui model, este testul de concordanță, care este aplicabil în condiții specifice, bine făcute.

Un astfel de test este cel al preciziei cu care au fost estimați parametrii modelului, care trebuie în primul rând, să fie posibil de realizat și estimarea să fie cât mai precisă.

Evaluarea performanțelor modelelor pentru fiabilitate software nu poate utiliza totdeauna metode statistice, întrucât numărul erorilor latente este finit, deci numărul de observații experimentale nu este suficient pentru prelucrare statistică.

În general, orice model permite predicția caracteristicilor numerice ale duratei scursă de la defectarea $i - 1$ la defectarea i , pe baza cunoașterii duratelor dintre defectări, predicție efectuată pe baza unui număr restrâns de observații.

Vor fi prezentate în continuare după tipuri de metode utilizate pentru validarea experimentală a modelelor fiabilității software și validarea unui anumit model.

10.4.2. Metode grafice

Fie $F_i(x_i)$ funcția de repartiție a parametrului x_i , ea reprezentând pentru soft probabilitatea manifestării unei erori în intervalul $(t_{i-1}, t_{i-1} + x)$.

Fie $\hat{u}_i = F_i(x_i)$ valoarea prezisă a acestei probabilități de către un anumit model, unde:

- i ia valori de la k la n ; $k - 1$ este numărul minim de date începând de la care se poate calcula o predicție;

- n numărul total de erori observate în cursul funcționării softului.

Presupunând că valoarea prezisă, $\hat{u}_i = F_i(x_i)$, coincide cu valoarea adevărată a funcției de repartiție și știind că valorile oricărei funcții de repartiție sunt distribuite în intervalul $[0,1]$, rezultă că în cazul ideal, valorile \hat{u}_i ($i = k, k + 1, \dots, n$) sunt realizări particulare ale unei variabile uniform distribuite în intervalul $[0, 1]$.

În acest caz, se poate aplica testul de validare Kolmogorov-Smirnov, comparând abaterea maximă a curbei empirice cu cuantila distribuției Kolmogorov-Smirnov ($\Delta\alpha_{\text{critic}}$), corespunzătoare numărului de observații $n - k + 1$ și riscului α adoptat.

Pentru a trasa curba empirică se ține seama că valorile nu sunt ordonate crescător, iar funcția lor de repartiție are formă de scară cu pași neuniformi, de mărime $1/(n - k + 2)$, așa cum este prezentat în figura 10.4, a [1, 8, 16].

Pentru $n - k$ suficient de mare, graficul are aspectul unei curbe continue. Dacă ea ar coincide cu prima bisectoare, care reprezintă funcția de repartiție teoretică, predicția ar fi ideală. Situarea curbei empirice deasupra primei bisectoare indică probabilități de defectare estimate mici, semnificând un sistem de predicție optimist. Situarea curbei empirice sub prima bisectoare indică un sistem de predicție pesimist.

Curba 1 din figura 10.4, b corespunde unei predicții optimiste, curba 2 corespunde unei predicții ideale, iar curba 3 corespunde unei predicții pesimiste.

Dacă abaterea maximă a curbei empirice față de funcția de repartiție teoretică este mai mică decât abaterea critică admisă ($K_{\alpha \text{ critic}}$) predicția a fost corectă și, cu nivelul de semnificație $1 - \alpha$, ipoteza se admite.

Metoda poate fi aplicată și pentru altfel de repartiții.

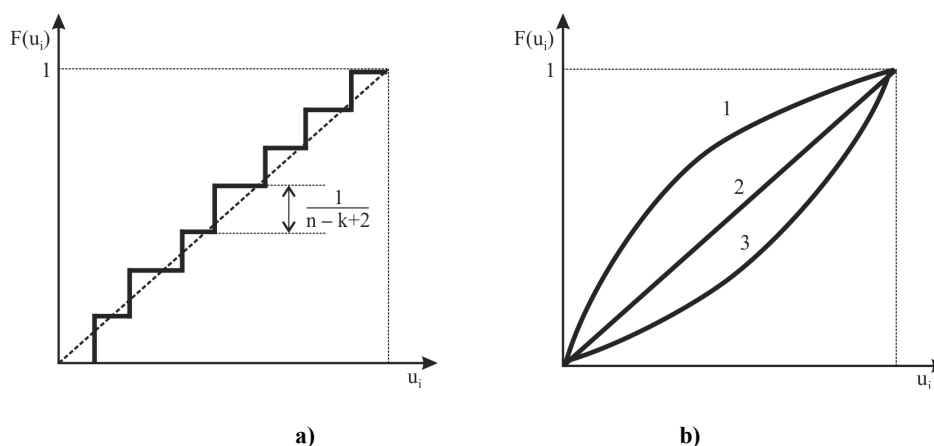


Figura 10.4. Forma funcției de repartiție a) discretă, b) continuuă

10.4.3. Acuratețea predicțiilor

Acuratețea unei predicții, numită de unii autori *zgomotul predicției*, se poate exprima, de cele mai multe ori, utilizând metoda celor mai mici pătrate.

Dacă predicțiile $F_i(x_i)$ sunt nedepășite, mediile lor coincid cu valorile adevărate, înscrise pe bisectoare, astfel încât abaterile sistematice constatate pe graficele $f(u_i)$ pot fi considerate ca o indicație a dispersiei. Întrucât se presupune că fiabilitatea unui program nu are salturi mari prin detectarea și eliminarea unei erori, rezultă că dispersia mare se datorează procedurilor de predicție.

Pentru a compara modelele din punctul de vedere al variabilității predicțiilor, este necesară o măsură globală a dispersiei, respectiv a împrăștierii valorilor față de valoarea teoretică.

Există mai multe expresii cu care estimează împrăștieria în funcție de estimatorul prezis de model, astfel:

a) În cazul în care se prezice media m_i a unui interval x_i , împrăștieria (abaterea pătratică medie) este dată de relația:

$$\sigma = \frac{n-k}{n-k+1} \cdot \frac{\sum_{i=k}^n (x_i - m_i)^2}{\sum_{i=k}^n (x_i - \bar{x})^2}, \quad (10.4.1)$$

unde n și k au semnificațiile precizate mai sus.

b) Dacă se evaluează media $x_{i;0,5}$ a duratei x_i , împrăștieria este dată de relația:

$$\sigma = \frac{x_{i;0,5} - x_{i-1;0,5}}{x_{i-1;0,5}}. \quad (10.4.2)$$

c) Dacă se evaluează densitatea de apariție a erorilor, h_i , calculată la momentul apariției erorii i , împrăștierea se calculează cu relația:

$$\sigma = \frac{h_i - h_{i-1}}{h_{i-1}}. \quad (10.4.3)$$

O mărime globală a calității estimatorului este *verosimilitatea previzională*, definită ca probabilitatea asociată observațiilor x_k, x_{k+1}, \dots, x_n , calculată pe baza observațiilor x_1, x_2, \dots, x_{k-1} :

$$L = \prod_{i=k}^n \hat{f}_i(x_i). \quad (10.4.4)$$

Pentru a evalua *performanțele predictive* relativ la două modele A și B, se calculează raportul verosimilității previzionale ale celor două modele:

$$P = \frac{\prod_{i=k}^n \ln \hat{f}_i^A(x_i)}{\prod_{i=k}^n \ln \hat{f}_i^B(x_i)}. \quad (10.4.5)$$

Cu cât raportul performanțelor predictive este mai mare, cu atât modelul A oferă predicții mai bune față de modelul B.

În literatura de specialitate sunt prevăzute metode de combinare a predicțiilor în vederea recalibrării predicțiilor oricărui model, pentru toate erorile detectate. În cazul validării modelelor fiabilității software, subiectivitatea factorului uman joacă un rol major, întrucât analistul trebuie să selecteze, pe baza experienței personale și inspirației proprii, modelul care face predicția cea mai adecvată în raport cu situația analizată.

Calitatea predicțiilor pe termen scurt nu implică, în mod automat, o aceeași calitate a predicțiilor pe termen lung, analistul fiind acela care trebuie să schimbe modelul în funcție de rezultatele obținute și cele așteptate și să le aplice succesiv.

Scuze pentru eventualele erori de numerotare. Modulul de curs face parte dintr-o carte mai mare, nu le-am cuprins pe toate în modul.

Capitolul 11. PROBLEME SPECIFICE DE FIABILITATE HARDWARE

11.1. Defecte specifice sistemelor hardware

Un calculator numeric conține subsisteme hardware și software interconectate funcțional pentru a prelucra, transfera și stoca informații la solicitarea unui utilizator, conform unui algoritm de calcul.

În mod uzual, când se discută de fiabilitatea hardware trebuie avute în vedere fiabilitatea memoriei calculatorului, a microprocesorului cu tot ce conține el – Unitatea Centrală de Procesare cu Unitatea de Control, ALU și aria de registre, Unitatea de Memori (internă și externă, inclusiv HDD în multiplele lui forme) Unitatea Intrare/Ieșire, magistralele de transfer și nu în ultimul rând conexiunile.

Pentru studiul fiabilității acestor subsisteme se aplică metodele specifice sistemelor tehnice, considerând că sunt alcătuite din mai multe module funcționale, cuplate logic. Se folosesc mai ales metodele bazate pe blocuri funcționale logice, metoda arborilor de defecte și/sau de evenimente, capitolul 5 din această carte, metoda grafurilor interconectate, metoda Markov, de asemenea prezentată în această carte în capitolul 5.

Din punctul de vedere al *utilizatorului* fiabilitatea înseamnă buna funcționare a tuturor acestor subsisteme, ori de câte ori are nevoie. Pentru aceasta achiziționează sisteme pe care le consideră fiabile, sigure și la prețuri rezonabile. Uneori pentru a achiziționa sisteme performante rebuie plătite costuri foarte mari care se justifică atunci când sistemele de calcul îndeplinesc funcții critice, de exemplu în aplicații nucleare, aerospațiale, militare, în domeniul bancar, securitate națională..

Din punctul de vedere al *producătorilor de calculatoare*, pentru toate subsistemele și componente lor, trebuie să obțină date despre fiabilitatea lor, indicatori privind probabilitățile de defectare dar și despre condițiile de funcționare pentru care au fost testate, dar și compatibilitatea cu celelalte componente ale sistemului.

Din punctul de vedere al *producătorilor de componente* pentru sisteme de calcul, pentru fiecare dintre aceste componente ei trebuie să demonstreze că sunt fiabile, că probabilitatea apariției diferitelor defecte este suficient de mică și sunt compatibile între ele. Pentru aceasta el trebuie să proiecteze teste prin care să obțină datele necesare pentru a calcula cu o precizie cât mai bună probabilitățile de defectare și să ofere garanții referitoare la durata de bună funcționare. Ei trebuie să conceapă tehnologii de producție astfel încât să iasă în întâmpinarea cerințelor de fiabilitate și securitate a producătorilor de sisteme. Tot mai mulți producători de hardware au început să producă și software pentru a asigura compatibilitatea lor.

Toți cei implicați în realizarea și exploatarea calculatoarelor sunt interesați de creșterea fiabilității hardware, de găsirea de metode evitate a defectelor sau de sisteme redundante la prețuri rezonabile.

Evitarea defectelor este o metodologie idealizată care presupune că toate componentele sunt perfecte. Pentru că hardware-ul de astăzi are o calitate excepțională, nivelul software în calculatoarele obișnuite adoptă o astfel de viziune idealizată. Programatorii presupun că sistemul pe care se rulează programele lor este extrem de fiabil.

Fiabilitatea hardware este obținută printr-o combinație de tehnici de redundanță, proiectare și fabricație cu precizie foarte ridicată, testare riguroasă și "ardere" (*burn-in*) pentru eliminarea defectelor din perioada de mortalitate infantilă.

Producătorii de hard proiectează și testează sisteme de calcul în condiții mai nefavorabile decât cele cunoscute pentru funcționarea lor normală și propun sisteme tolerante la defectări tranziente sau sisteme redundante

O metodă foarte simplă dar scumpă de hard tolerant la defectare este de a multiplica fiecare componentă a sistemului, realizând *sisteme redundante spațial* sau de a lansa în execuție fiecare instrucțiune de mai multe ori (*redundanță temporală*). Apariția unui defect poate fi detectată comparând rezultatele obținute din subsistemele redundante.

Prima schemă de toleranță a defectelor a fost propusă de John von Neumann în 1956 și se numește *redundanță modulară triplă* (*Triple Modular Redundancy*) [21, 32]. În această schemă trei module execută aceeași operație și un modul numit de *vot* alege rezultatul după numărul de apariție. Există și scheme în care sistemul de votare este replicat, pentru a nu depinde de o singură componentă. Un astfel de sistem de votare este folosit în calculatoarele care controlează procese critice sau de securitate. De exemplu pentru o navetă spațială sistemul este compus din cinci calculatoare, din care patru fac aceleași calcule și al cincilea este folosit pentru operațiuni necritice. Rezultatele oferite de cele patru calculatoare sunt transmise către un motor de comparare care calculează local rezultatul votului, iar fiecare calculator compară rezultatele sale cu ale celelalte trei; când unul dintre calculatoare oferă rezultate diferite este scos din funcțiune.

Dacă două calculatoare se defectează sistemul intră într-un mod de funcționare în care rezultatele sunt comparate și atunci când diferă, recalculat. Al cincilea calculator conține un sistem de control complet separat, dezvoltat de altă companie, care intră în funcțiune numai când o eroare identică este detectată în celelalte patru programe.

Vor fi prezentate și alte metode redundanță în acest capitol.

Un alt mecanism de creștere a fiabilității sistemelor de calcul este utilizarea de *servere redundante* - Redundant Servers. Serverele principale au copii absolut identice fie în aceeași locație sau de preferat în altă locație pentru a asigura și protecții în caz de calamitatea naturală (ex: cutremur, inundații). Dacă serverul principal se defectează, indiferent de motiv, serverul secundar care funcționează în paralel cu acesta preia instantaneu sarcinile de lucru astfel că pentru beneficiar, problema apărută este insesizabilă.

Avantajul principal al acestui sistem este acela că oferă cel mai mare grad de siguranță la ora actuală, dezavantajul principal este legat de menținerea sincronă a celor 2 sau mai multe servere redundante și de prețul mare pentru menținerea în funcțiune paralelă a 2 servere performante. În cazul în care serverele sunt în locuri

diferite, apar probleme legate de lățimea de bandă la transfer care poate afecta interceptarea fluxurilor de date.

Memoriile și microprocesorul sunt dispozitive semiconductoare digitale, integrate la scară foarte largă, ultra largă (VLSI) iar studiul mecanismelor lor de defectare are la bază studiul mecanismelor de defectare a componentelor discrete care le alcătuiesc, tranzistoare bipolare sau MOS. Miniaturizarea progresivă și creșterea densității de integrare a dispozitivelor integrate a dus la probleme specifice legate străpungeri între trasee alăturate și instabilități ale sarcinii, care afectează informația stocată sau transferată.

11.2. Fiabilitatea circuitelor integrate VLSI

Așa cum bine se cunoaște, memoriile semiconductoare și microprocesorul sunt circuitele digitale VLSI iar aceste circuite, oricât de complexe și specializate ar fi cuprind tranzistoare bipolare sau tranzistoare MOS, realizate (cel mai frecvent) pe substrat de siliciu monocristalin, interconectate prin intermediul unor trasee metalice (de regulă din aur dar pot fi și din platină sau aluminiu) extrem de subțiri și apoi încapsulate etanș. Au o complexitate foarte mare, practic nelimitată, și dimensiuni ale cipurilor foarte mici (câțiva nm). Conțin un număr impresionant de componente discrete, de exemplu AMD din 2011 la 3,2÷3,6 GHz, are un cip de 32nm și conține 1,2milioane de tranzistoare, POWER7+ de la IBM, tot pe cip de 32nm, conține 2,1milioane de tranzistoare, iar Itanium de la Intel 3,1 milioane de tranzistoare (2012).

Fiabilitatea circuitelor integrate este determinată, în principal, de trei mari factori:

- fiabilitatea componentelor sale, respectiv a tranzistoarelor;
- materialul din care este realizată capsula și etanșitatea pe care o realizează capsula;
- conexiunile interne și externe.

Fiabilitatea circuitelor integrate este mai mare decât a oricărui circuit realizat cu componente discrete care îndeplinește aceeași funcție, dar mai mică decât a oricărei componente a sa. Numărul mare de pini presupune mai multe puncte de sudură și, respectiv, o distanță mai mică între pini învecinați și o secțiune mai mică a acestora, toate acestea conducând la capsule fragile care se pot deteriora în cursul manipulării.

Rata de defectare a circuitelor integrate, ca și a oricăror componente semiconductoare discrete, este influențată de condițiile termice, electrice și mecanice din mediul în care funcționează circuitul, de umiditate și compoziția chimică a mediului, nivelul de radiații de la locul de funcționare [1, 35].

Temperatura ridicată este o formă de solicitare care reduce fiabilitatea oricărei componente; rata de defectare este determinată de creșterea vitezei reacțiilor chimice la creșterea temperaturii. Multe dintre mecanismele de defectare sunt cauzate de reacții chimice care au loc între materialele din care sunt realizate circuitului respectiv și impuritățile apărute ca urmare a etanșării imperfecte a structurii. Umiditatea poate influența fiabilitatea circuitelor integrate, mai ales în situația când sunt încapsulate în material plastic, care este permeabil la umiditate.

În acest caz, apar defectări catastrofale prin corodarea metalizărilor de aluminiu studiate foarte mult de specialiști pentru reducerea ratei de defectare a echipamentelor electronice. Fiabilitatea circuitelor integrate a putut fi crescută prin următoarele acțiuni:

- fabricație industrială automatizată, în condiții controlate, reproductibile în timp;
- utilizarea de materii prime de calitate controlată, compatibile între ele și cu mediul de funcționare;
- încercări de selecție (screening) pe produsul finit, astfel încât anumite clase de circuite realizate cu un anumit nivel de fiabilitate, să aibă destinații adecvate. De exemplu: cerințele de securitate ale energiei nucleare sau industriei aerospațiale și militare impunând un nivel de fiabilitate foarte înalt sunt testate foarte sever. Circuitele destinate uzului casnic nu necesită astfel de nivele de fiabilitate sau testare, și nici nu pot costa la fel de mult.

Cercetătorii consideră că cele mai eficiente încercări de selecție de fiabilitate pentru circuite integrate MOS, trebuie să cuprindă următoarele verificări [51]:

- inspecție optică cu o mărire minimă de $200\times$;
- stocare la temperatură ridicată, respectiv 96 ore la temperatura maximă de funcționare, admisă de circuit;
- zece cicluri termice (-65°C , $+125^{\circ}\text{C}$);
- teste de vibrații la o accelerație 20 g ;
- măsurarea parametrilor semnificativi, critici la 25°C ;
- funcționare sub tensiunea nominală la temperatura maximă admisă, timp de 24 ore;
- test electric bun/defect la 25°C ;
- ardere (burn-in), timp de 24 ore, la temperatura maximă admisibilă.

Costurile acestor verificări sunt mari, dar ele pun în evidență circuitele integrate cu defecte de fabricație, defecte de mortalitate infantilă, care pot fi eliminate numai prin astfel de probe.

Pot fi puse în evidență, de asemenea, degradări ale unor parametri critici, problemele de structură, de sudură și încapsulare.

În urma încercărilor accelerate efectuate pe circuite integrate bipolare s-au constatat derive ale parametrilor de ieșire care duc scăderea fiabilității circuitelor.

Astfel de defectări pot fi induse de mai multe mecanisme de defectare:

- a) defecte de mascare și “pin-hole”,
- b) impurități metalice originare în sistemul și sursele de difuzie,
- c) dislocații la marginea zonei de difuzie n^{+} , de emitor,
- d) impurități în și la suprafața oxidului.

Aceste mecanisme le-am studiat amănunțit și prezentat în capitolul 7 *Fiabilitatea dispozitivelor semiconductoare* din cartea *Elemente de fiabilitate* [1].

În cazul circuitelor integrate care conțin tranzistoare cu efect de câmp (MOS și CMOS), se manifestă mecanisme de defectare specifice circuitelor integrate cu tranzistoare bipolare la care se adaugă mecanisme specifice tranzistoarelor MOS, respectiv străpungeri datorate descărcărilor electrostatice sau stăpungerea oxidului

de poartă prin tunelare Fowler-Nordheim, degradare datorită purtătorilor fierbinți sau pătrunderea intrinsecă.

Un mecanism specific care se manifestă numai în circuitele integrate se datorează interacțiunilor secundare între diferite regiuni cu nivel diferit de dopare. Cele mai importante dintre aceste interacțiuni secundare au fost modelate sub forma unor dispozitive parazite. Apariția unor dispozitive parazite pentru anumite configurații structurale și raport câmp/curent în dispozitiv, pot perturba semnificativ funcționarea acestora, efectele putând merge până la degradarea lor ireversibilă [1, 6].

Preocupările privind modelarea acestor fenomene sunt determinate de următoarele aspecte:

- caracterul inerent al acestor interacțiuni;
- creșterea nivelului interacțiunilor, determinată de micșorarea dimensiunilor dispozitivelor;
- creșterea complexității interacțiunilor determinată de creșterea complexității circuitelor integrate;
- favorizarea unor interacțiuni secundare de către mediu, specifice unor domenii de aplicații.

Cel mai de întâlnit dispozitiv parazit care se manifestă în tehnologia CMOS este *tiristorul parazit* de volum [6]. El se formează din două tranzistoare bipolare parazite, conectate într-o buclă cu reacție pozitivă, care în situația în care colectorul unui tranzistor comandă baza celuilalt cele două tranzistoare formează o structură de tiristor $p-n-p-n$ care este bistabilă. La comutarea tiristorului din starea blocat (corespunzătoare funcționării corecte a dispozitivului) în starea deschis, se formează o cale de curent de mică rezistență între alimentare și masă. În funcție de limitarea din exterior, se stabilește un curent mare pe calea parazită, care provoacă perturbarea funcționării dispozitivului și, eventual, degradarea lui permanentă.

Înteruperea curentului de tiristor se poate realiza numai prin întreruperea alimentării dispozitivului.

Fenomenul, în literatura de specialitate, este denumit *latch-up* și, la nivelul unei statistici din 2010, era responsabil pentru 19% din totalul defectărilor din exploatarea a circuitelor CMOS [1].

Pentru diminuarea mecanismului de defectare latch-up este nevoie de optimizarea proiectului și a proceselor tehnologice, optimizarea layout-ului circuitului, modelarea analitică a mecanismului pentru a cuprinde cât mai multe dintre interacțiunile care apar într-un dispozitiv.

Un alt mecanism care produce degradarea dispozitivelor VLSI este așa numitul *tranzistor MOS parazit*, apărut ca o consecință a dezalinierei măștilor în procesul de fabricare, etapa mascare. O ușoară dezaliniere împreună cu sarcina indusă în dielectric fac ca un strat de inversie să fie indus în zona de câmp dintre difuzii adiacente, formându-se în această configurație un canal de conducție n-MOS. Efectul cel mai important este pierderea informației stocate în celula de memorie.

Geometria fină a circuitelor integrate face ca unele aglomerări locale ale dopării, care pot apare în timpul proceselor tehnologice, în volumul stratului epitaxial să fie

resimțite ca o joncțiune parazită, denumită *joncțiune fantomă*, care se inserează joncțiunilor dispozitivului. Amănunte privind aceste mecanisme de defectare, și altele sunt tratate în cartea *Elemente de fiabilitate* capitolul 7.

11.3. Fiabilitatea memoriilor semiconductoare

11.3.1. Defectarea memoriilor semiconductoare

Memoriile semiconductoare se realizează încă sub forma integrată în tehnologie bipolară (cu tranzistoare bipolare) sau în tehnologie MOS (cu tranzistoare MOS). În acest capitol vom discuta despre fiabilitatea memoriilor, presupunând cunoscut modul de funcționare al acestora, modul de înregistrare și ștergere a conținutului, tehnologia de realizare.

Din punct de vedere al fiabilității, memoriile semiconductoare se comportă, într-o primă aproximație, ca și circuitele integrate deoarece se produc prin aceleași metode, cu aceleași tehnologii de realizare, au structuri asemănătoare, diferă însă prin gradul de integrare și numărul de pini. Fiabilitatea memoriilor depinde, în principal, de trei factori :

- nivelul de integrare;
- maturitatea tehnologiei de fabricație;
- tipul încercărilor de selecție pentru eliminarea defectelor de mortalitate infantilă.

Cauzele apariției defectelor memoriilor semiconductoare pot fi:

- metode de montare inadecvate, care pot provoca stresuri mecanice, stresuri care pot determina fisuri în cristalul semiconductor;
- firele metalice de conexiune, care trec peste trepte de oxid, au o secțiune și mai mică la aceste treceri și pot ceda în timpul utilizării;
- capsulele neetanșe și/sau impurități chimice (rezultate din curățarea suprafeței) pot produce coroziunea metalizării;
- legăturile electrice dintre cristalul semiconductor și pin, precum și conexiunile interne ale circuitului;
- din cauza distanțelor foarte mici dintre pini, pot apare scurtcircuite între doi pini alăturați.

În funcție de tipul memoriei, fiabilitatea poate să varieze după [3, 22, 30, 54]:

- tehnologie (bipolară, MOS etc.);
- forma semnalului;
- tipul de celulă (dinamică sau statică);
- persistența conținutului (volatil, nevolatil);
- tipul memoriei (numai citire, scriere/citire, reprogramabilă);
- modul de programare;
- modul de ștergere.

Pentru cele mai multe tipuri de memorie, prin încercări de selecție se elimină aproximativ 80% din defecte. Totuși, defectarea chiar a unui procent de 1% din populația majoritară a lotului de memorii asamblate în sisteme de calcul complexe, poate provoca efecte catastrofale.

În memoriile semiconductoare pot apare două categorii de erori :

- *permanente*, care constau în defecte fizice ale unei celule, linii sau a unui bit individual, cum ar fi întreruperi sau scurtcircuit,

- *tranziente*, care sunt erori ale unui bit, care nu se repetă și nu cauzează o deteriorare permanentă. Aceste erori constau în trecerea defectoasă a unui bit din starea "0" în starea "1" sau invers. Pot fi determinate de zgomotul sistemului, de oscilații ale tensiunii de alimentare etc. Aceste erori sunt mai numeroase față de cele permanente (cu aproape un ordin de mărime), dar nu sunt catastrofale.

Pentru memorii **PROM bipolare**, cu fuzibil, mecanismele de defectare critice sunt [1, 3]:

– formarea ocazională a unor punți de topire, datorate arderii insuficiente a siguranțelor în timpul topirii;

– în cursul citirii este posibil ca unele siguranțe să se ardă și să schimbe programarea inițială.

Ambele mecanisme pot fi eliminate printr-o probă de burn-in de 168 ore, în funcționare nominală.

Mecanismele de defectare critice ale **memoriilor MOS** sunt :

– defecte în oxid, datorită faptului că poarta de oxid nu este protejată;

– impurități în și la suprafața oxidului sau impurități metalice, ca la circuitele integrate;

– defecte de cristal, dislocații, defecte de împachetare sau fisuri, care produc perechi de purtători de sarcină, care pot conduce la apariția unui curent rezidual.

În cazul memoriilor dinamice predomină defectele de oxid din cauza grosimii mari a stratului de oxid și a faptului că poarta de oxid nu este protejată. Stratul mai subțiri de oxid nu pot fi utilizate pentru că ar putea conduce la pierderi de sarcină prin curenții reziduali.

Străpungerea oxidului de poartă se poate produce în anumite puncte, acolo unde stratul de oxid are o grosime mai mică sau unde au apărut impurificări necontrolate (în timpul sau după depunerea oxidului de poartă), în același mod cu cel prezent în tranzistoarele MOS. Deoarece proporția stratului porților cu oxid față de suprafața totală a cipului este mult mai mare decât la circuitele integrate logice, această cauză de defectare este cea mai importantă pentru memoriile MOS. Scurtcircuitele poartă-substrat duc la defectarea totală a celulelor individuale ale unei întregi linii sau ale unei coloane și pot produce defectarea întregii memorii. Capacitatea porților este de ordinul câtorva zecimii de pF și se pot deteriora chiar la energii mici.

Defectările în oxid se produc după câteva microsecunde, astfel încât aceste defecte pot fi eliminate prin aplicarea unui câmp electric de intensitate mare timp de câteva secunde, în cadrul încercărilor de selecție.

Pentru memoriile reprogramabile, la care ștergerea se realizează cu radiații ultraviolete, chiar sub acțiunea radiației solare pot apare ștergeri ale memoriei.

Un mecanism de defectare important, dar care apare numai în anumite condiții de exploatare, este pierderea datelor din una sau mai multe celule.

Condițiile, care pot determina pierderea datelor memorate pot fi:

- *pomparea încărcării*: de câte ori o comandă face să crească potențialul de comandă, o mică parte din încărcarea celulei poate fi pompată în celulele vecine,

astfel încât, după o anumită perioadă de timp, între două cicluri de împrăștiere, poate apărea modificarea informațiilor memorate;

- atât pentru memoriile statice cât și pentru cele dinamice se poate întâmpla ca decodificatorul de adresă să fie prea lent.

- în cazul memoriilor dinamice, timpul de pauză al sistemului de citire a unor linii poate să fie prea lung și să genereze erori de citire;

- unele defecte ale monocristalului din care este realizat cipul pot bloca funcția de ansamblu a unei celule sau pot forma legături între două celule vecine iar înregistrarea să se facă în celula învecinată;

- limitele de toleranță ale parametrilor dinamici ai memoriilor semiconductoare sunt uneori prea mici, iar sub influența temperaturii se pot reduce și mai mult;

- curenții de scurgere prea mari pot conduce la pierderea rapidă a datelor memorate.

Analiza defectelor memoriilor EPROM 2716, la firma INTEL, a pus în evidență următoarea distribuție a mecanismelor de defectare specifice [1, 51]:

- pierderea informațiilor pe unul sau mai mulți biți: 6,15%;

- modificarea timpului de acces (prin contaminare, polarizare, sarcină superficială): 18,9%;

- curent rezidual în oxid : 12%;

- defecte ale metalizării (circuite interne deschise): 5,5%;

- defecte de fabricație: 2,1%.

În cazul memoriilor PROM bipolare, principalele mecanisme de defectare sunt [1, 3]:

- formare ocazională a unor punți de topire din Ni-Cr și TiW;

- modificarea aparentă a unui bit programat;

- scurtcircuite ale comenzilor cuvânt (biți de scriere și citire);

- probleme de stabilitate;

- curenți de scurgere în celule;

- deficiențe în controlul și menținerea stabilității procesului.

Fabricarea memoriilor în tehnologie bipolară presupune mai multe etape de fotograură și difuzie, ceea ce face ca defectele datorate procesului de fabricație să fie mai frecvente. Memoriile bipolare pot fi însă de 10 (zece) ori mai rapide decât cele MOS, deși densitatea circuitelor MOS este de 5 (cinci) ori mai mare față de a celor bipolare.

11.3.2. Modalități de îmbunătățire a fiabilității memoriilor semiconductoare

O memorie defectă poate crea multe probleme unui sistem de calcul, indiferent dacă echipează un server sau un PC. În funcție de tipul și rolul ei, defectarea unei memorii poate fi catastrofică (de exemplu, dacă ea este memoria ROM în care este stocat BIOS-ul), poate duce la pierderi de date sau poate corupe fișierele sistemului. Există o preocupare foarte mare pentru îmbunătățirea fiabilității memoriilor atât la producătorii de memorii dar și la realizatorii de sisteme de calcul unde pot apărea defectări la montarea modulelor de memorie în sistem.

Din punct de vedere al fiabilității pot fi considerate trei tipuri de memorie: memorii neprotejate, memorii cu paritate și memorii ECC.

Memorii neprotejate stochează fiecare bit de date în mod separat și nu oferă nici o protecție împotriva erorilor. Ca atare sunt cele mai ieftine. Cum însă dimensiunea memoriilor a crescut foarte repede, la ora actuală această soluție este riscantă, căci probabilitatea ca un bit să se defecteze este foarte mare.

Memorii cu paritate, numite și **Memorii cu verificarea** folosesc o metodă foarte simplă pentru a detecta erori de un bit în fiecare octet, metodă numită metoda *bitului de paritate*. Prin această metodă este adăugat câte un bit suplimentar, numit *bit de paritate* la fiecare cuvânt care reține dacă în biții fiecărui cuvânt există un număr par sau impar de cifre 1. Dacă numărul de biți egali cu 1 în fiecare cuvânt este par, se consideră paritate pară, iar dacă numărul de biți egali cu 1 este impar, se numește paritate impară. Bitul de paritate este memorat împreună cu cuvântul care prezintă interes iar când este accesată memoria, automat este recalculată și paritatea iar rezultatul se compară cu bitul de paritate citit.

Dacă paritatea nu este corectă se declanșează o excepție iar sistemul de operare decide cum trebuie să acționeze, de exemplu blocarea programului care folosea acea memorie și marcarea memoriei ca fiind defectă, astfel încât alte programe să nu o poată refolosi. Verificarea parității este o operație foarte rapidă, care se poate face foarte simplu în hardware în paralel cu transferul informației. Bitul de paritate se poate adăuga la început sau la sfârșit, în funcție de convenția aleasă. Metoda bitului de paritate poate fi extinsă la nivel de bloc, adăugând un bit de paritate și pentru coloana blocului de cuvinte, câștigând astfel spațiu pentru date.

Metoda permite detectarea modificării unui singur bit din cuvânt sau bloc dar nu și erorile multiple și nici nu poate corecta erorile detectate. Dacă probabilitatea ca doi biți să se modifice este mare trebuie folosite alte metode de detectare a erorilor.

Memoriile ECC sunt protejate cu un cod complex de corecție a erorilor (*Error Correcting Code*). Acest cod poate corecta automat orice eroare de 1 bit care apare într-un șir de 64 de biți. Pentru aceasta memoria stochează fiecare șir de 64 de biți folosind un cod de 72 de biți. "Risipa de memorie" în această schemă este de 8 biți la fiecare 72.

La fiecare acces la memorie, sistemul verifică dacă cuvântul de cod este corect; dacă nu, automat calculează cel mai apropiat cuvânt de cod pe care apoi îl decodifică.

Aceste operații de verificare și decodificare sunt complicate și consumatoare de resurse, astfel încât un sistem cu memorii ECC este cu aproximativ 5% mai lent decât unul cu memorii cu paritate.

Rata de defectare a memoriilor semiconductoare este de ordinul $10^{-6} \div 10^{-9} \text{h}^{-1}$ și se poate îmbunătăți cu un ordin de mărime prin măsuri adecvate de testare, selecție și funcționare în condiții normale, atât de către producător cât și de utilizator. În cazul memoriilor reprogramabile, fiabilitatea se referă la numărul de cicluri de reprogramare. Nu au fost încă realizate memorii care să permită un număr de reprogramări mai mare de 10000.

Producătorul de memorii poate îmbunătăți fiabilitatea prin alegerea unor tehnologii reproductibile și sigure, ținerea sub control a tuturor proceselor tehnologice pentru a evita impurificarea sau neuniformitatea straturilor de oxid și a traseelor electrice.

Utilizatorul de memorii poate apela la următoarele metode de îmbunătățire a fiabilității:

- alegerea memoriei după evaluarea producătorului, respectiv verificarea algoritmului de testare și a limitelor de utilizare;
- verificarea inițială a caracteristicilor electrice ale memoriei, la temperatura maximă de utilizare;
- utilizarea de memorii în sistem numai după selecția prin încercări adecvate (temperatură înaltă, tensiune de alimentare mărită);
- teste de durată de viață. În cele mai multe situații, durata de viață impusă este de 25 ani, iar testele se fac prin *încercări accelerate* riguros proiectate. În mod curent memoriile ajung să se uzeze moral după mult mai scurt timp [1, 35, 37];
- mânăuire corectă și protecție față de suprasarcină;
- funcționarea în condiții normale.

Dacă multă vreme au fost preferate memoriile dinamice care puteau fi realizate cu o putere disipată mai mică și o densitate de integrare mare, se revine la memoriile statice, care au timpi de reacție mai mici, dar fiabilitate mai bună.

11.4. Fiabilitatea microprocesoarelor

11.4.1. Aspecte specifice privind fiabilitatea microprocesoarelor

Microprocesoarele sunt tot circuite integrate pe scară foarte largă (VLSI) care cuprind elemente logice complexe și față de alte circuitele integrate logice ridică probleme de testare speciale, datorită faptului că nu pot fi testate separat, pe standuri de test. Fiind elemente ale sistemului de calcul, testarea lor este dependentă de memorii, interfețe, de modul de succesiune al comenzilor etc. Pe de altă parte pentru testare, în mod uzual se folosesc sisteme mai performante decât cele testate, astfel că pentru testarea unui microprocesor de ultimă oră nu există altul mai performant. Au fost dezvoltate metodele de încercare specifice care să ofere rezultate concludente. Astfel de metode pot fi:

a) *simularea logică a prototipului*, construind funcțiile de ansamblu ale unui microprocesor pe un calculator performant, care să aibă viteza necesară pentru simulare;

b) *test de comparație pe un exemplar înrudit*. Pe baza funcționării acestui exemplar se elaborează o previziune de tip “bun/rău”;

c) *metode de autotestare*, pe anumite segmente cu funcții logice bine definite, pornind de la un nucleu minimal de testare și continuând până la verificarea tuturor funcțiilor elementare (citire, scriere, memorare și comparare) precum și a celorlalte operații, cu ajutorul comenzilor deja testate.

Tendința este să se renunțe la echipamentele de testare scumpe și să se realizeze structuri care să se autotesteze, prin încorporarea în structura microprocesorului a unor circuite speciale de autotestare care să permită microprocesorului ca, la fiecare pornire a sistemului din care face parte, să își verifice singur capacitatea de funcționare.

11.4.2. Autotestarea microprocesoarelor

Scopul metodei de “autotestare” este să se realizeze un test orientat spre necesitățile specifice ale utilizatorului, care renunță complet la informațiile care nu interesează sau nu figurează în specificația tehnică a microprocesorului respectiv. Nu se face nici o presupunere privind realizarea internă a unităților funcționale conținute în interior, presupunând că:

- funcționarea microprocesorului poate fi complet descrisă de ansamblul său de instrucțiuni,
- toate erorile de funcționare se traduc prin funcționări eronate sau defectoase ale instrucțiunilor și pot fi recunoscute cu ajutorul testului setului de instrucțiuni.

Verificarea setului de instrucțiuni este efectuată în timpul autotestării, fiecare funcționare necontrolată va fi verificată într-o secțiune a programului de test de către candidat, cu ajutorul funcțiilor testate anterior (figura 11.1) [2, 10, 28]. Programul complet de autotestare este compus din segmente asemănătoare, în prima secțiune de testare (așa-numitul “test de bază”) fiind verificată o cantitate minimă de funcțiuni, ca bază a algoritmului de testare (figura 11.2).

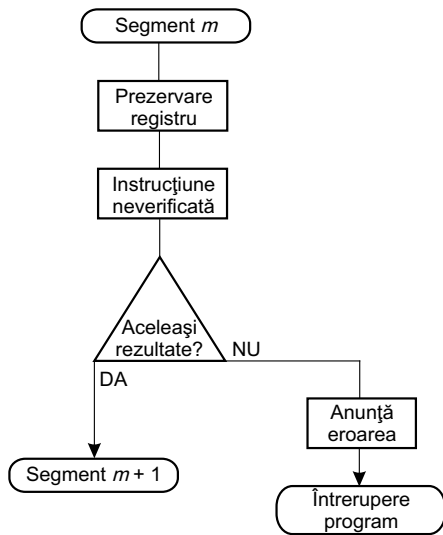


Figura 11.1. Test funcțional secvențial

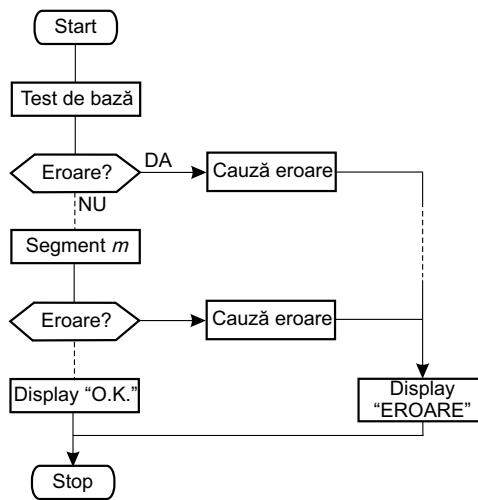


Figura 11.2. Exemplu de program de autotestare

Circuitele pentru recunoașterea erorilor cuprind, așa cum se poate urmări în figura 11.3 :

- memoria de programare cu capacitate mare;
- registrul de întreruperi (microprocesorul generează el însuși semnale de întrerupere și controlează tratamentul lor),
- supravegherea secțiunii programului,
- comanda derulării programului (se generează pentru autotest un semnal de pornire, se supraveghează lungimea maximală a testului și se comandă cele trei dispozitive de afișare: “test în curs”, “în ordine” și “eroare”). Toate funcțiile pot fi realizate pe aceeași placă. De notat că pe baza măsurărilor practice efectuate s-a putut demonstra că o memorie testată poate indica un comportament parțial eronat

și că este capabil a atenționa asupra erorilor din programul de autotestare și întrerupe chiar derularea programului de testare.

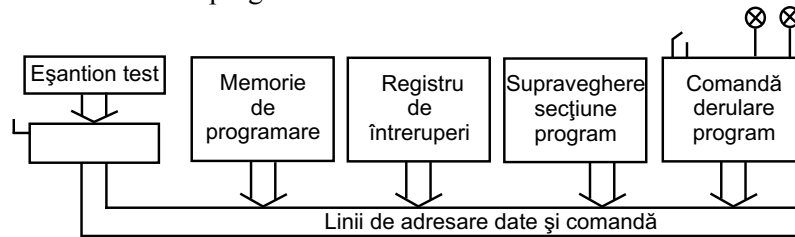


Figura 11.3. Circuit suplimentar pentru identificarea erorilor

Pe baza celor prezentate, se poate realiza un sistem de testare care să poată fi utilizat la controlul de intrare, ca sistem de măsură de tip “bun/rău”. La ansamblul circuitelor, indicate în figura 11.3, trebuie adăugate alimentarea, generatorul de tact și respectiv circuitele de interfață.

Pentru testarea diferitelor microprocesoare este necesară:

- testarea completă a tuturor funcțiilor de comandă și întrerupere,
- derularea testului în condiții de timp real.

Ca dezavantaj trebuie menționată imposibilitatea de a măsura întârzierile semnalelor, amplitudinile acestora, timpii de acces la memorii etc. Pentru a crește rigurozitatea testului de control de intrare, se pot varia tensiunea de alimentare, frecvența și temperatura de funcționare.

În general, câmpul valorilor funcționării corespunzătoare depășește cu mult domeniul valorilor nominale.

11.4.3. Factori de care depinde fiabilitatea microprocesoarelor

Fiabilitatea microprocesoarelor depinde atât de materialele și tehnologiile de realizare, rigurozitatea testării, de condițiile de funcționare și nu în ultimul rând de programarea lor. Este unanim acceptat că

– rata de defectare ce depinde de mai mulți parametri, dintre care cei mai importanți sunt: temperatura, tensiunea de alimentare, tehnica de preîncărcare, criteriul de defectare și nivelul de încredere admis pentru indicatorii de fiabilitate, nu numai pentru rata de defectare

– tehnica de preîncărcare dinamică a circuitelor VLSI este preferabilă celei statice pentru testele de durată de viață, deoarece corectările și întreruperile repetate ale modurilor interne simulează mai bine utilizarea reală.

– măsurările electrice trebuie să cuprindă programe de testare cu secvențe de exploatare, în condițiile cele mai proaste și impulsuri în timp. Criteriile *bun/rău* trebuie să fie definite pentru fiecare circuit pe baza parametrilor de c.c, c.a. și timpii dați de către producător în fila de catalog. Microprocesoarele care nu satisfac condițiile testului respectiv sunt selecționate după modurile de defectare, apoi are loc o înregistrare a datelor și o analiză a defectelor pentru a putea determina mecanismul specific de defectare.

– calculul factorilor de accelerare și al ratelor de defectare trebuie făcut în funcție de temperatura joncțiunii (și nu a mediului ambiant).

– ratele de defectare ale microprocesoarelor cu capsulă de plastic sunt comparabile cu cele cu capsule ceramice, deși capsulele din plastic sunt sensibile la umiditate și coroziune.

La circuitele integrate defectele de mascare pot duce în timp la deriva parametrilor și o verificare incompletă sau imperfectă a funcționării poate avea drept rezultat mascarea unei defecțiuni existente, care să nu poată fi detectată decât după începerea funcționării sistemului.

O dată cu creșterea vitezei de funcționare și cu scăderea tensiunii de utilizare, crește sensibilitatea circuitelor față de suprasolicitări, fiind necesară o protecție împotriva încărcărilor electrostatice și a perturbațiilor dinamice pe traseele semnalelor.

11.4.4. Căi de îmbunătățire a fiabilității microprocesoarelor

11.4.4.1. Screening

Metoda cea mai folositoare este aceea de eliminare a defectelor de fabricație prin probele de selecție atât în timpul fabricației cât și după încapsularea microprocesoarelor, permit eliminarea unui număr foarte mare de defecte (ajungând chiar până la 99,9% la fima INTEL). Aceste încercări de selecție complexe sunt foarte scumpe și de durată. Ele presupun mai multe etape. Un exemplu de succesiune de etape de încercare este următorul [5, 51]:

a) Teste de funcționare în curent continuu (*DC functional tests*). Pentru microprocesoarele INTEL pot fi 52 de teste diferite pentru evidențierea scurtcircuitelor, circuitelor întrerupte, curenților de scurgere și tensiunilor de străpungere. Încercările de verificare, în funcție de tensiunea de intrare, trebuie efectuate în așa fel încât să poată fi determinată eficacitatea circuitelor de protecție ale tuturor intrărilor.

b) Teste de parametri în curent alternativ (*AC parametric tests*). Dacă un microprocesor dispune de 256 comenzi, prin combinarea acestora rezultă circa 400000 de comenzi posibile, care necesită timp și specialiști.

c) Încărcarea celulelor de înaltă tensiune (*High voltage cell stress*). Este controlată funcționarea corectă a tuturor structurilor, cu ajutorul unei tensiuni de încărcare de circa 2 ori mai mare decât tensiunea nominală. pentru a elimina majoritatea defectelor de străpungere ale oxidului.

După încapsularea microprocesoarelor sunt prevăzute și alte teste de clasificare, pentru a se stabili dacă specificațiile prescrise sunt respectate și apoi stabilite destinațiile și prețurile. Acestea se efectuează în condiții dure de tensiune și durată, la temperatură a mediului de $85^{\circ}C$. Metoda nu permite verificarea fiecărui circuit intern, ci doar stabilirea proastei sau bune funcționări generale a unui tranzistor de pe plachetă (în total, în această fază de fabricație se efectuează circa 1300 de teste diferite).

11.4.4.2 Procesoare tolerante la erori tranziente

Toți marii producători de microprocesoare sunt preocupați de realizarea de procesoare tolerante la defectări. Primul astfel de procesor, un procesor superscar tolerant la erori tranziente, a fost propus la conferința de microarhitectură MICRO

2001 de un grup de cercetători de la universitatea Carnegie Mellon. Ei propuneau o schemă care folosește redundanță temporală, pornind de la procesor superscalar obișnuit căruia i-au făcut câteva modificări simple astfel încât fiecare instrucțiune citită să fie lansată în execuție în mod repetat iar rezultatele seturilor lansate în execuție sunt comparate între ele. Executarea instrucțiunii suplimentare care să nu afecteze sistemul a fost posibilă printr-un mecanism de redenumire a regiștrilor folosiți în procesorul superscalar. Îmbunătățirea fiabilității depinde de gradul de redundanță: dacă fiecare instrucțiune este executată de două ori, o eroare se manifestă prin rezultate diferite și instrucțiunea trebuie reexecutată; dacă o instrucțiune este executată de mai mult de două ori, se poate folosi o schemă de votare cu majoritate, prezentată în capitolul 9. paragraful 9.4.2.

Un astfel de procesor poate fi proiectat să lucreze fie în mod normal, fie în mod cu fiabilitate crescută, depinzând de tipul de program executat. Performanța în modul cu fiabilitate ridicată este invers proporțională cu gradul de redundanță; de exemplu, dacă fiecare instrucțiune este executată de două ori, ne-am aștepta la o scădere a vitezei de calcul la 50%. În realitate, penalizarea este ceva mai mică, din cauză că un program nu folosește toate resursele computerului. De exemplu, dacă un program folosește 80% din resurse, când execută un program cu duplicarea fiecărei instrucțiuni are nevoie de 160% resurse, ceea ce se traduce într-o degradare a performanței cu 37,5% ($100/160 = 62,5 = 100 - 37,5$).

Un sistem tolerant la erori tranziente a fost folosit de compania IBM care a creat microprocesorul G5 pentru calculatoarele mainframe. El conține două benzi de execuție identice, controlate de același ceas. Toate instrucțiunile sunt executate în mod sincron de ambele benzi, iar la sfârșitul benzii rezultatele sunt comparate. Dacă rezultatele sunt identice, rezultatul instrucțiunii este scris în registrul destinație sau în memorie. Dacă nu, se generează o excepție software, care de obicei se soldează cu reexecuția instrucțiunii problemă. Erorile tranziente sunt astfel reparate în mod transparent. Această schemă este funcțională pentru că probabilitatea ca o eroare tranzientă să afecteze ambele benzi în același fel este foarte foarte mică.

BIBLIOGRAFIE

1. Floarea BAICU, *Elemente de fiabilitate*, Ed. Victor, București, 2005.
2. Floarea BAICU, *Fiabilitatea sistemelor informatice*, Modul de curs, Universitatea Hyperion, 2012
3. Floarea BAICU, Sorina MIHĂITĂ, *Aspects of semiconductor memories reliability* Analele Universității Hyperion din București, 2014
4. Floarea BAICU, *Allocation Methods for Systems Reliability During Design Elaboration Phase*, Rev. *Optimum Q*, Vol. XIX nr. 1-2, pg. 40-45, 2008
5. Floarea BAICU Andrei Mihai BAICU - *The Complexity and Diversity of Technologies Involved in HDD Production and Operation - Consequences on Reliability*, ENEC 2013
6. T. I. BĂJENESCU - *Fiabilitatea componentelor electronice*, Editura Tehnică București, 1996
7. T. BARON, Al. ISAC – MANIU și alții *Calitate și fiabilitate*, manual practic, Editura Tehnică, București 1988
8. V. M. CĂTUNEANU, A. MIHALACHE, *Bazele teoretice ale fiabilității*, Editura Academiei RSR, București, 1983
9. V. M. CĂTUNEANU, A. MIHALACHE - *Reliability Fundamentals*, ELSEVIER, Amsterdam – Oxford – New York, Tokyo, 1989
10. R.A. Evans *Reliability Prediction*, IEEE Transactions on Reliability, vol. 48, June 1999, pg. 105;
11. Corina GEORGIAN, P. TARAȘ, Floarea BAICU, *Hazard Plotting for Incomplete Data - Computer Program*, International Symposium on Nuclear Energy, SIEN' 93, București, 25, 1993
12. T. HOHAN - *Tehnologia și fiabilitatea sistemelor*, Editura Didactică și Pedagogică București, 1982,
13. Z. JELINSKI, P. B. MORANDA, *Software Reliability Research*, in Statistical Computer Performance Evaluation, We. Freiberger, Ed. Academic Press, 1972
14. B. LITTLEWOOD, L. STRIGINI, *Software reliability and dependability: a roadmap*, Proceedings of the Conference on The Future of Software Engineering, Limerick, Ireland, 2000
15. MICHAEL R. LYU. *Handbook of Software Reliability Engineering*. Internet URL: <http://www.cse.cuhk.edu.hk/~lyu/book/reliability/>
16. A. MIHALACHE - *Când calculatoarele greșesc.... - Fiabilitatea sistemelor de programare (software)*, Editura Didactică și Pedagogică R.A. - București, 1995
17. Gh. MIHOC, Aneta MUJA, E. DIATCU - *Bazele matematice ale teoriei fiabilității*, Editura Dacia, Cluj Napoca 1976
18. Aneta MUJA, E. DIATCU - *Matematica pentru economiști* Editura Victor, București 1999
19. J. D. MUSA, *Software reliability engineering: more reliable software faster and cheaper*. Bloomington: Authorhouse, 2004.

20. J. D. MUSA AND K. OKUMOTO, *Software reliability models: concepts, classification, comparisons, and practice*, in *Electronic Systems Effectiveness and Life Cycle Costing*, Heidelberg, 1983.
21. J. Von Neumann, "*Probabilistic Logics*," *Auromafa Studies*, Princeton University Press, 1956.
22. J.D PALMER *Electronic-Module Environmental-Stress-Screening Data-Evaluation Technique*, Annual Reliability and Maintainability Symposium 1999 Proceedings, 50;
23. B. PARHAMI *Defect, Fault, Error ..., or Failure?* IEEE Transaction on Reliability, Vol. 46, No. 4, 450, 1997
24. A. PASQUINI s.a *Reliability Analysis of Systems Based on Software and Human Resources*, IEEE Transaction on Reliability, Vol. 50, No. 4, 337, 2001
25. M. PERMAN s.a. *Semi-Markov Models with an Application to Power-Plant Reliability Analysis*, IEEE Transaction on Reliability, IEEE Transaction on Reliability, Vol. 46, No. 4, 526, 1997
26. D. C. SECUI – *Fiabilitatea stațiilor electrice de distribuție* Editura Universității din Oradea, 2003
27. C. SMIDTS s.a. *Software Reliability Modeling: An Approach to Early Reliability Prediction*. IEEE Transaction on Reliability, Vol. 47, No. 2, 268, 1998
28. N. D. STOJADINOVIC, S. D. RISTIC, "*Failure Physics of Integrated Circuits and Relationship to Reliability*, Phys. Stat. Sol. (a) 75, 11, (1983)
29. V. SUNE, J.A.CARRASCO *A Failure Distance Based to Bound the Reliability of Non-Repairable Fault Tolerant Systems without the Knowledge of Minimal Cuts* IEEE Transaction on Reliability, Vol. 50, No. 1, 60, 2001
30. V. K. ȘTEFAN – *Evaluarea fiabilității sistemelor de calcul*, Editura Universității din Oradea, 2003
31. X. TENG, H. PHAM *A Software Reliability Growth Model or N-Version Programing Systems*, IEEE Transaction on Reliability, Vol. 51, No.32, 311, 2002
32. VANDERKUL W., *The Use of Triple-Modular Redundancy to Improve Computer Reliability*, 1962, <http://www.ccs.neu.edu/course/csg712/resources/Lyons-Vanderkulk-62.pdf>,
33. WALD, Abraham. *Contributions to the Theory of Statistical Estimation and Testing Hypotheses*. Ann. Math. Statist. 10 (1939), no. 4, 299--326.doi:10.1214/aoms/1177732144. <http://projecteuclid.org/euclid.aoms/1177732144>.
34. IEEE (ANSI) Standard 982.2/1988 *Software Reliability Terminology*
35. IEEE Std. 381/1977 *IEEE Standard Criteria for Type Tests of Class 1E Modules Used in Nuclear Power Generating Station*
36. IEEE 352/75 - *Guide for General Principles of Reliability Analysis of Nuclear Power Generating Station Protection Systems* ;
37. MIL HDBK 217C/1979 – *Military Standardization Handbook Reliability Prediction of Electronic Equipment*
38. MIL STD 19.500E/1969 – *Military Standard General Specification for Semiconductor Devices*

39. SR ISO 2854/2000 - *Interpretarea statistică a datelor. Tehnici de estimare și teste referitoare la medii și dispersii*
40. SR CEI 60863/1998 - *Prezentarea rezultatelor previziunii caracteristicilor de fiabilitate și disponibilitate*
41. SR CEI 61070/1998 - *Proceduri de încercare a conformității pentru disponibilitate în regim stabil*
42. SR CEI 61164/1998 - *Creșterea fiabilității. Încercări și metode de estimare statistică*
43. STAS R 12007/4-89 - *Metode de calcul al estimatorilor punctuali și al limitelor de încredere, rezultate din încercările de determinare a fiabilității echipamentelor*
44. STAS R 12007/5-86 - *Încercarea de fiabilitate a echipamentelor – Planuri de eșantionare pentru confirmarea proporției succesului*
45. STAS R 12007/6-89 - *Teste de validare a ipotezei unei rate de defectare constante*
46. STAS R 12007/7-81 - *Încercarea de fiabilitate a echipamentelor – Planuri de eșantionare pentru confirmarea ratei de defectare și a timpului mediu între defectări, în ipoteza unei rate de defectare constante*
47. STAS 12689/88 - *Analiza modurilor de defectare și a efectelor defectărilor*
48. STAS 8174/1,2,3/77 - *Fiabilitate/Mentenabilitate/Disponibilitate - Terminologie*
49. STAS 10307/75 - *Fiabilitatea produselor industriale – Indicatori de fiabilitate*
50. SR EN 61025/2007 - *Analiza fiabilității pe baza arborilor de defectare*
51. <http://www.e-mtbf.com/>INTEL Reliability Report, RR febr. 2001
52. <http://ep.etc.tuiasi.ro/files/CID/memorii.pdf>
53. http://en.wikipedia.org/wiki/Configurable_Fault_Tolerant_Processor
54. <http://www.ace.tuiasi.ro/users/103/2011-Huzum%20rezumat%20.pdf>