

Lucrarea nr. 4: Determinarea fiabilității unui sistem cu ajutorul metodei „Monte Carlo“

Tiberiu Laurian, Radu–Florin Mirică

2014

1 Scopul lucrării

Determinarea fiabilității unui sistem prin simularea funcționării componentelor prin metoda „Monte Carlo“. Această metodă este foarte utilă în cazul sistemelor cu structură complicată și componente cu repartiții de probabilitate diferite.

2 Elemente teoretice

Analiza „Monte Carlo” – Metodă matematică folosită pentru a găsi soluția aproximativă a unei probleme complexe prin generarea unui număr suficient de mare de numere aleatoare și observarea dacă o parte din numere respectă una sau mai multe proprietăți. Această metodă este utilă în găsirea unor soluții numerice pentru probleme prea complexe pentru a putea fi rezolvate analitic. Se poate folosi pentru stabilirea probabilității de manifestare a riscurilor.

3 Punerea problemei

Se cere să se determine fiabilitatea sistemului din figura 1 prin simularea funcționării componentelor cu ajutorul metodei „Monte Carlo” și să se compare rezultatul cu cel obținut prin calculul bazat pe schema structurală. Pentru facilitarea calculului, fiabilitățile componentelor vor fi modelate cu legea de repartiție exponențială negativă, având următoarele rate de defectare:

$$\lambda = \begin{matrix} 0,0003 & 0,0001 & 0,0001 & 0,0002 & 0,0001 & 0,0003 & 0,0002 \\ 0,0004 & 0,0005 & 0,0001 & 0,0002 & & & \end{matrix}$$

Durata pentru care se face calculul este $t = 600$ ore.

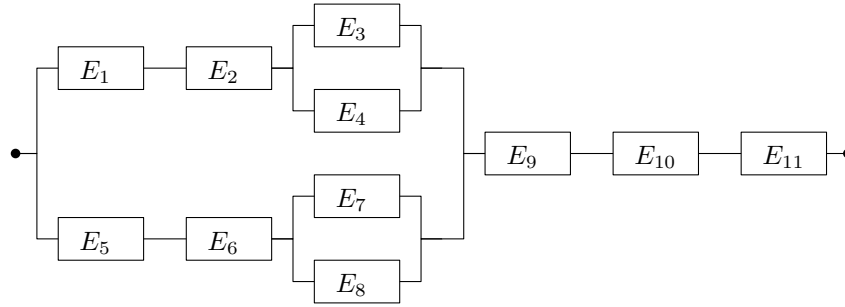


Figura 1: Schema sistemului de analizat

4 Rezolvare

Rezolvarea problemei se va face cu ajutorul calculatorului, utilizând mediul de programare MATLAB/*Octave*¹ sau un program specializat precum Mathcad². Se utilizează două metode de rezolvare pentru verificarea rezultatelor.

4.1 Calculul bazat pe schema structurală

Fiabilitatea fiecărui element se poate calcula prin relația: $R_i = \exp(-\lambda_i t)$ unde $i = 1, 2, \dots, 11$. Este util de remarcat faptul că sistemul este alcătuit dintr-o structură repetitivă „serie-paralel” (vezi figura 2) pentru care fiabilitatea are forma:

$$R_A = R_1 R_2 [1 - (1 - R_3)(1 - R_4)] = e^{-\lambda_1 t} e^{-\lambda_2 t} [1 - (1 - e^{-\lambda_3 t})(1 - e^{-\lambda_4 t})].$$

Pentru ramura de jos fiabilitatea este:

$$R_B = R_5 R_6 [1 - (1 - R_7)(1 - R_8)] = e^{-\lambda_5 t} e^{-\lambda_6 t} [1 - (1 - e^{-\lambda_7 t})(1 - e^{-\lambda_8 t})].$$

Pentru ramura din dreapta avem:

$$R_C = R_9 R_{10} R_{11},$$

iar pentru întregul sistem fiabilitatea este:

$$R_S = [1 - (1 - R_A)(1 - R_B)] R_C.$$

¹Programarea se poate face și în limbaje precum FORTRAN, C++, etc.

²Alte programe similare ce pot fi folosite pentru rezolvarea acestei probleme sunt: SMATHStudio, Mathematica, Maxima sau Maple.

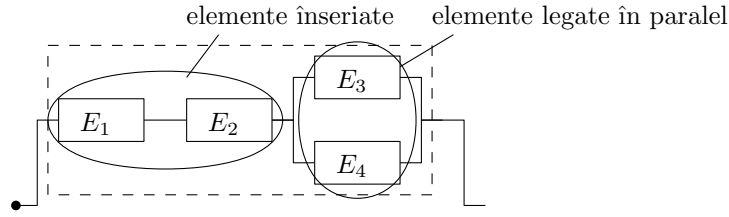


Figura 2: Detaliu repetitiv al schemei structurale

4.2 Calculul prin metoda „Monte Carlo“

Se vor parcurge următorii pași:

1. Se alege numărul de simulări (ex. $ns = 1000$).
2. Se definesc iteratorii i și j : $i = 1 \dots n$; $j = 1 \dots ns$
3. Se generează $n \times ns$ durabilități aleatoare repartizate după legea exponențială negativă utilizând funcția *exprnd*.
4. Fiecare valoare a durabilității generate aleator se compară cu durată t impusă și se determină starea elementului (1 – element în funcțiune, 0 – element defect).
5. Pentru fiecare simulare se definește un șir x_i cu valori 1 și 0 în funcție de rezultatele punctului anterior.
6. Se calculează starea sistemului pentru fiecare simulare în parte: $R_{S_j} = [1 - (1 - R_{A_j})(1 - R_{B_j})]R_{C_j}$; unde: $R_{A_j} = x_{1_j}x_{2_j}[1 - (1 - x_{3_j})(1 - x_{4_j})]$, $R_{B_j} = x_{5_j}x_{6_j}[1 - (1 - x_{7_j})(1 - x_{8_j})]$, $R_{C_j} = x_{9_j}x_{10_j}x_{11_j}$.
7. Se determină fiabilitatea sistemului ca raport al numărului de cazuri când sistemul nu este defect la numărul total de simulări

$$R_{S-MC} = \frac{\sum_j R_{S_j}}{ns}.$$

Se compară rezultatele R_S și R_{S-MC} pentru mai multe variante ale numărului de simulări (ex. $ns=100; 500; 1000$).

5 Concluzii

Se vor scrie câteva comentarii referitoare la rezultatele obținute.

Lucrarea se încheie cu un raport, în care se vor scrie textul problemei, datele de intrare și rezultatele obținute.

6 Listing MATLAB/Octave pentru rezolvarea problemei

În continuare se prezintă un listing pentru rezolvarea problemei în OCTAVE, cu afișarea rezultatelor pe ecran.

```
# -----
#
# FSM – Lucrare laborator 4
# -----
clear all;
# -----
# 1. Introducerea datelor de intrare
# -----
lambda = 1.0E-4*[3; 1; 1; 2; 1; 3; 2; 4; 5; 1; 2]; # Ratele de defectare
n = length(lambda);
t = 100; # Durata de functionare
R = exp(-lambda*t); # Fiabilitatile componentelor
# -----
# 2. Calculul bazat pe schema structurala
# -----
RA = R(1)*R(2)*(1-(1-R(3))*(1-R(4)));
RB = R(5)*R(6)*(1-(1-R(7))*(1-R(8)));
RC = R(9)*R(10)*R(11);
RS1 = (1-(1-RA)*(1-RB))*RC;
printf("-----\n");
printf("CALCUL_SCHEMA_STRUCTURALA\n");
printf("FIABILITATEA_SISTEMULUI_ESTE: _RS_=%f\n",RS1);
# -----
# 3. Calculul bazat pe metoda MONTE – CARLO
# -----
ns = 7000; # Nr. de simulari
RMC = zeros(n,ns);
x = zeros(n,ns);
for j = 1:ns,
    for i = 1:n,
        RMC(i,j) = exprnd(1/lambda(i));
        if RMC(i,j) > t,
            x(i,j) = 1;
        else
            x(i,j) = 0;
        endif
    end
    RAMC(j) = x(1,j)*x(2,j)*(1-(1-x(3,j))*(1-x(4,j)));
    RBMC(j) = x(5,j)*x(6,j)*(1-(1-x(7,j))*(1-x(8,j)));
    RCMC(j) = x(9,j)*x(10,j)*x(11,j);
    RSMC(j) = (1-(1-RAMC(j))*(1-RBMC(j)))*RCMC(j);
end
RS2 = sum(RSMC)./ns;
printf("-----\n");
printf("METODA_MONTE-_CARLO\n");
printf("FIABILITATEA_SISTEMULUI_ESTE: _RS_=%f\n",RS2);
printf("-----\n");
```