一个多节点声纳系统中同步时钟机制的可靠性评估和系统优化问题

第一组 whh allesgutewh@gmail.com

摘要

本课题意在研究一个多节点声纳系统中同步时钟机制的可靠性评估和系统优化问题。本文将该同步时钟机制分解为切换器、节点、系统三个层次，其中切换器寿命概率密度为指数分布。利用马尔科夫链描述工作状态，最终通过仿真得到不同节点个数条件下系统的平均使用寿命、可靠性等指标，为系统的设计做出指导。

关键词: 可靠性评估, 系统优化, 指数分布, 马尔科夫链

RELIABILITY EVALUATION AND SYSTEM OPTIMIZATION OF SYNCHRONOUS CLOCK MECHANISM IN A MULTI-NODE SONAR SYSTEM

ABSTRACT

This subject is intended to study the reliability evaluation and system optimization of the synchronous clock mechanism in a multi-node sonar system. In this paper, the synchronous clock mechanism is decomposed into three levels: switch, node, and system, and the lifetime probability density of the switch is exponentially distributed. The Markov chain is used to describe the working state, and finally the indicators such as the average service life and reliability of the system under different n umber o f nodes are obtained through simulation to guide the system design.

Keywords: Reliability assessment, System Optimization, Index distribution, Markov chain

1 问题重述

1.1 问题背景

一个功能复杂的系统往往由许多功能较为单一的次级系统组成，次级系统又由许多更次一级的系统部件构成……各级部件最终构成的系统能否正常工作取决于各级部件的工作情况。

在了解系统构成之后，由最底层的元件状态可逐级推算各级的工作状态进而得到整个系统的工作状态。因此可以根据最底层的元件不可靠元件的工作寿命的概率分布推算系统的预期工作寿命，由此得到的结论可用于指导设计各级的元件个数，以最低成本达到不同水平的工作寿命或其他指标的要求。

1.2 问题的提出

本课题研究的是一个多节点声纳系统中的同步时钟机制，整个系统的结构可分为系统、节点、切换器三个层次。系统的能否正常工作取决于是否有足够多功能正常的节点，节点是否具有正常功能取决于 A、B 两个切换器的工作状态。

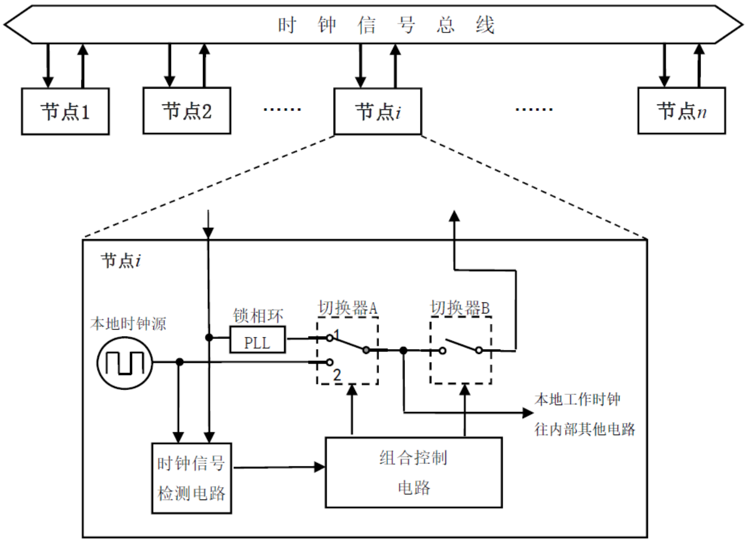

图 1: 一个多节点声纳系统中的时钟同步机制示意图 [1]

本课题的目标是针对图1所示多节点声纳系统中的时钟同步机制，根据最底层的切换器元件的工作寿命概率分布，仿真分析整个系统的工作寿命（单位：小时）及可靠性，并根据仿真结论给出最优的系统设计方案。

2 模型假设

1. 切换器 A、B 的工作寿命服从指数分布且系数已知。
2. 构成节点的元件中除切换器 A、B 外均为可靠元件，即不可能损坏的元件。
3. 故障一旦发生就无法修复，即切换器 A、B 的故障不可逆。
4. 各节点状态相互独立。
5. 系统的最大寿命为 90000 小时。

3 符号说明

表1列出了本文需要的符号。

表 1: 符号说明

符号	符号描述
n	构成系统的节点个数
k	系统正常工作的最少正常节点个数, 本案例固定为 5
$1(X)$	X 为 True 时为 1，否则为 0
P_{EX}	元件发生故障，故障类型为 X 的条件概率
$Q_x(t)$	t 时刻处于 x 状态的节点个数
S	用于仿真的系统个数
T_f	系统工作寿命，即首次失效时间（单位：小时）
$R(w)$	系统可靠性，系统工作寿命超过某一定值 w 的概率

4 理论基础

4.1 指数分布

设随机变量 X 服从参数为 λ 的指数分布，记作 $X \sim E(\lambda)$，其概率密度函数 [2]

$$f(x)=\begin{cases}\lambda e^{-\lambda x}, & x \geqslant 0,\\ 0, & x<0,\end{cases} \tag{1}$$

则

$$\begin{aligned} E(X)=&\int_0^{\infty} x\lambda e^{-\lambda x}dx=-\int_0^{\infty} x de^{-\lambda x}\\ =&-xe^{-\lambda x}\mid_0^{+\infty}+\frac{1}{\lambda}\int_0^{\infty}\lambda e^{-\lambda x}dx\\ =&\frac{1}{\lambda}\end{aligned} \tag{2}$$

式2说明一个电子元件的使用寿命如果服从参数为 λ 的指数分布，则平均使用寿命为参数的倒数。参数越大，平均能使用寿命越短。

对于任意 $0<a<b$，

$$\begin{aligned} P(a<X\leqslant b)=&\int_a^b \lambda e^{-\lambda x}dx\\ =&e^{-\lambda a}-e^{-\lambda b}\end{aligned} \tag{3}$$

有

$$\begin{aligned} P(t<X\leqslant t+\triangle t\mid X>t)=&\frac{P(t<X\leqslant t+\triangle t)}{P(X>t)}\\ =&P(0<X\leqslant\triangle t)(\triangle t>0)\end{aligned} \tag{4}$$

由式3、式4可知元件使用了 t 小时后，至少使用 $(s+t)$ 小时的概率与从开始使用至少使用 s 小时的概率相等。

4.2 马尔科夫链

马尔科夫过程是一个可以仅基于其当前状态对未来结果进行预测的过程，而且最重要的是，此类预测与了解过程的完整历史一样可以做出预测。换句话说，以系统的当前状态为条件，其未来状态和过去状态是独立的。马尔科夫链是一种具有离散状态空间或离散索引集 (通常表示时间) 的马尔科夫过程，移动到下一个状态的概率仅取决于当前状态，而不取决于前一个状态 [3]。

在本问题中，不可靠元件的寿命服从指数分布，根据指数函数的性质可得到系统当前状态与过去独立，符合马尔科夫过程的条件。因此本问题可以使用马尔科夫链描述元件工作状态变化，进而对系统工作状态进行仿真。

5 解决方案

5.1 模型细节

5.1.1 切换器状态

已知切换器 A、B 使用寿命均服从指数分布，以小时为单位指数分布的参数分别为 $\lambda_A=\dfrac{1}{2.72\times10^4}$、$\lambda_B=\dfrac{1}{3.31\times10^5}$，则在取时间离散间隔为一小时情况下，每隔一小时进行一次判断，其损坏的概率分别为 $(1-e^{-\lambda_A})$、$(1-e^{-\lambda_B})$。

同时已知 A 有三种故障形式、B 有两种故障形式，其条件概率如表2所示。

表 2: 各故障类型条件概率

P_{EA1}	P_{EA2}	P_{EA3}	P_{EB1}	P_{EB2}
0.30	0.30	0.40	0.33	0.67

元件寿命和故障类型均依照概率通过随机数进行模拟，模拟的最大时常为系统寿命的最大值 90000 小时。

5.1.2 节点状态

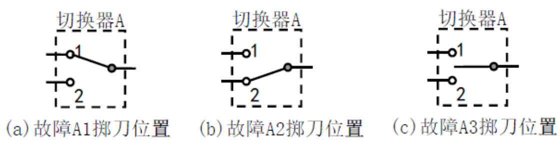

图 2: 切换器 A 的故障类型

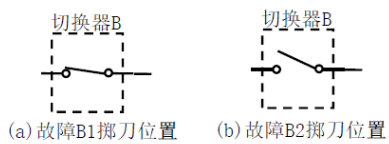

图 3: 切换器 B 的故障类型

如图2、3所示，切换器 A 有四种状态 (正常和三类故障)，B 有三种状态 (正常和两类故障)，共产生 12 种组合，其中一种正常、十一种故障。节点的工作模式有主模式和从模式两种，发生某些特定故障的节点仍然可以在某种模式下正常工作。

根据故障对节点功能的影响可分为六类：1PF(性能完好)、2SO(只能作为从节点)、3DM(主节点或失效节点)、4MO(只能作为主节点)、5DN(失效、不阻塞总线)、6FB(阻塞总线)。

与 A、B 状态具体对应方式如表3所示 (状态 0 表示切换器无故障)。

表 3: 各故障类型于切换器状态对应关系

序号	节点状态	切换器状态 (A,B)
1	PF	(0,0)
2	SO	(0,2) (1,0) (1,2)
3	DM	(2,0)
4	MO	(0,1) (2,1)
5	DN	(2,2) (3,0) (3,1) (3,2)
6	BF	(1,1)

5.1.3 系统状态

系统中共有 n 个节点，每个节点有六种可能的状态。节点功能可分为主节点和从节点两种，正常工作条件下要求主节点有且仅有一个，有效从节点个数不少于 $(k-1)$。根据这一条件，系统状态可分为四类：一定有效工作、一定不能有效工作、恰好有效工作、恰好不能有效工作。

仿真求解的是系统失效的时间，本文仅关注“一定不能有效工作”和“恰好不能有效工作”两种状态，其他各状态的判断条件在课程讲义中有详细介绍。

一定不能有效工作判断条件：

1. 系统中出现阻塞节点，即 $G_{FB}(t) > 0$；
2. 有多个主节点，即 $G_{MO}(t) > 1$；
3. 无主节点，即 $G_{MO}(t) + G_{PF}(t) + G_{DM}(t) = 0$；

4. 从节点不足，即 $G_{PF}(t)+G_{SO}(t)+1(G_{MO}(t)+G_{DM}(t))<k$.

以上条件只要满足其一，系统不能正常工作，记录此时的时间 t，跳出循环进行下一轮的判断。

恰好不能有效工作判断条件：

1. 无节点处于阻塞或只能作为主节点状态，即 $G_{FB}(t)+G_{MO}(t)=0$；
2. 有完全正常节点，即 $G_{PF}>0$;
3. 正常节点和只能作为从节点的节点个数恰为 $(k-1)$，即 $G_{SO}(t)+G_{PF}(t)=k-1$;
4. 有节点处于主节点或失效状态，即 $G_{DM}(t)>0$.

若以上条件全部满足，则系统处于主节点缺失状态，需要从 DM 和 PF 状态的节点中选择其一作为主节点，若选择的节点为 PF 状态的节点 (随机选中的概率为 $\frac{G_{PF}(t)}{G_{PF}(t)+G_{DM}(t)}$)，会造成有效从节点不足，记录此时的时间 t，跳出循环进行下一轮的判断。

5.1.4 可靠性指标

首次失效时间 T_f 即系统从初始时间到首次满足失效条件的时间，最大值不超过 90000 小时。平均首次失效时间 $E(T_f)$ 为给定 n 时仿真的 S 个系统的首次失效时间平均值。

可靠性 $R(w)$ 表示在 $0<t\leqslant w$ 期间系统一直处于有效工作状态的概率，有

$$R(w)=P(T_f\geqslant w)=\sum_{i=1}^{S}1(T_f^i\geqslant w)/S$$

。

5.2 模型优化

根据层次由低到高的切换器-节点-系统的顺序依次判断状态能够完成本问题的求解，但是求解时间较长。因此对模型进行优化，使其在能完成首次失效问题的基础上减少不必要的存储和计算。

5.2.1 状态判断优化

本问题研究的问题仅需关注系统首次失效时间，根据切换器故障不可逆这一假设，可以认为每个切换器在整个系统有效时间内至多发生一次变化，则节点状态至多发生两次变化，从而整个系统至多发生 $2n$ 次变化，我们仅需记录并检查这 $2n$ 个时间节点的切换器、节点和系统的状态即可。由一小时的恒定步长优化为根据故障的变步长方式进行仿真。

5.2.2 元件寿命求解优化

由于仅需关注节点发生变化的时刻，可直接使用 MATLAB 内置的 exprnd 函数直接计算每个切换器的寿命；依旧使用随机数确定故障类型。将状态变化的时间节点和故障类型作为参数传递给状态判断函数即可完成求解。

优化后的程序无需存储、判断每个小时的各级部件状态，大大节省了空间和时间，在仿真过程中 S 取 10^5 时，n 从 5 取到 20 共 16 轮仅需一分钟左右。得到的平均首次失效时间和 $w=25000$ 时的可靠性如表4所示。

表 4: 平均首次失效时间与可靠性 ($w = 25000h, S = 5 \times 10^5$)

n	5	6	7	8	9	10	11	12
$E(T_f)(h)$	9584	19000	28051	36373	43651	49895	55025	59091
$R(w)$	0.07575	0.23385	0.41876	0.58436	0.70954	0.79700	0.85244	0.88339
n	13	14	15	16	17	18	19	20
$E(T_f)(h)$	62199	64418	66014	67016	67501	67715	67586	67260
$R(w)$	0.89946	0.90535	0.90546	0.90365	0.89836	0.89275	0.88641	0.87962

6 拓展探究

6.1 可靠性的理论值计算

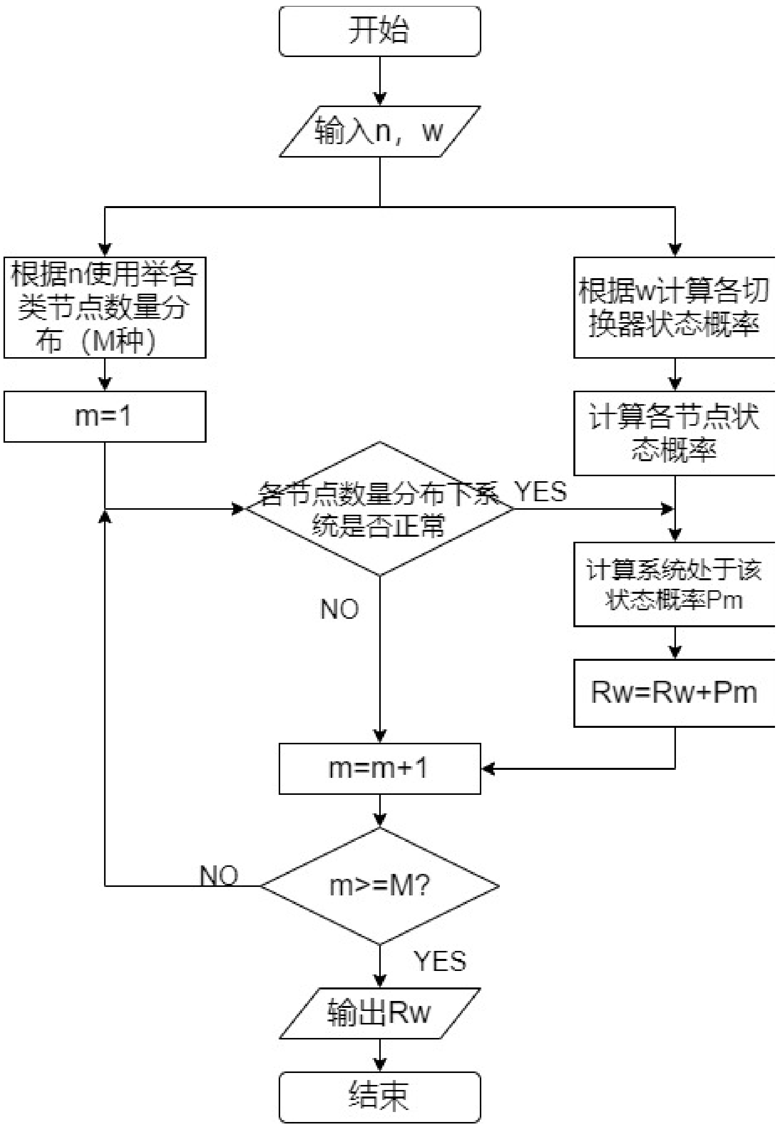

图 4: 可靠性 R_w 的理论值计算流程

由于切换器 A、B 的寿命分布已知，可根据4.2中的式3 、式4得到在使用 w 小时后切换器 A、B 损坏的概率分别为 $(1 - e^{-\lambda_A w})$、$(1 - e^{-\lambda_B w})$，根据表2中的概率条件可以计算表3中出现每种节点状态的概率。因此在已知各类节点状态数量时可以准确计算出系统处于该状态的概率。

各类节点的数量是一个参数已知、进行 n 次试验的 6 项分布问题，可采用枚举法列举所有的分布可能性，判断各种可能的组合下系统是否失效并计算在 w 小时后仍然能正常工作的概率即为节点数为 n 时的系统可靠性。

可靠性理论值计算流程图如图4所示，理论计算结果如表5所示。

表 5: 可靠性理论值 ($w = 25000h$)

n	5	6	7	8	9	10	11	12
$R(w)$	0.090519	0.25734	0.44364	0.60662	0.72927	0.81218	0.86347	0.89247
n	13	14	15	16	17	18	19	20
$R(w)$	0.90696	0.9125	0.91277	0.91	0.90552	0.9001	0.89414	0.88788

将表5中的理论值与表4中的仿真值对比可知理论值较仿真值偏大，这是由于仿真过程中用于计算可靠性的时间是首次失效时间，而系统本身可能会出现”复活”, 即系统因为某故障失效后因为切换器发生新故障导致节点状态改变，系统再次可以正常工作，处于”复活”状态的节点会被纳入理论值计算中，因此理论值较仿真值偏高。

6.2　故障原因分析

在系统状态判断过程中共有四类系统失效原因：出现阻塞节点、多主节点、无主节点、从节点不足，根据假设可知有部分系统寿命可以达到规定的最大使用寿命 90000 小时，将达到最大使用寿命作为系统失效的第五类原因，统计仿真过程中 n 取不同数值时系统故障原因（每个 n 值仿真 100000 套系统），得到如图5所示曲线。

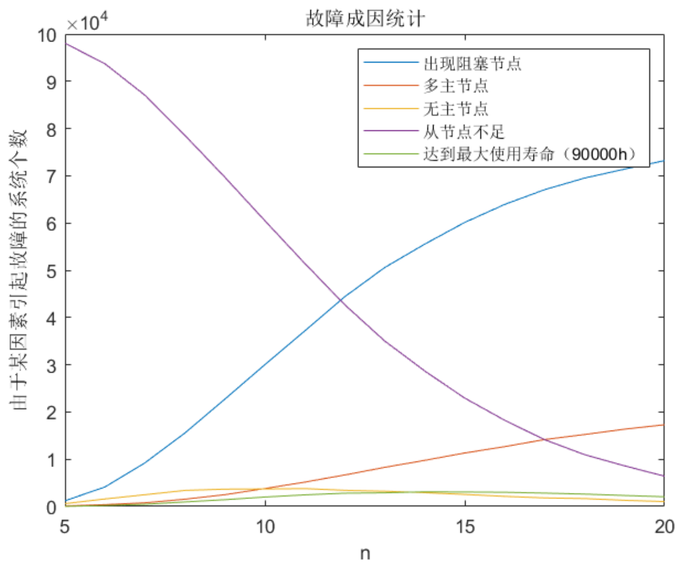

图 5: 故障原因统计 (S=100000)

由图5可知随着 n 增大，因出现阻塞节点而故障的系统越来越多，由系统仿真（表4）得到的可靠性最大和平均工作寿命最大的 n 值下（n=15、18），阻塞节点造成故障占比均超过百分之五十，这为系统优化提供了思路——降低阻塞节点出现的概率。

7　结论

通过随机模拟运行实验，研究 $n = 5, 6, ..., 20$ 时的一系列指标，得到如下结论

1. 在节点总数 n 为 15 时系统可靠性 $R(w)|_{w=25000h}$ 最大，最大值为 0.90546 ;
2. 在节点总数 n 为 18 时系统平均工作寿命 $E(T_f)$ 最大，最大值为 67715 小时。
3. 随着节点总数 n 的增加由于阻塞节点造成的故障占比大幅上升，系统可靠性最大和平均工作寿命最大时因阻塞节点造成的故障占比达到百分之五十以上，在进行系统优化时可优先考虑减少阻塞节点的出现。

8 致谢

非常感谢上海交通大学工程建模与仿真课程组设计的案例，非常感谢袁焱、李安琪老师课堂上的讲解和答疑。

参考文献

[1] 上海交通大学电子工程系, 一个多节点声纳系统中同步时钟机制的可靠性评估和系统优化问题 *[OL]*, ftp://202.120.39.248, 2020.

[2] 上海交通大学数学系, 概率论与数理统计 *(第二版)[M].* 上海交通大学出版社, 2011.

[3] "Markov chain," Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Markov_chain#cite_note-:3-11

附录

A 求解代码

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%初始化
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tic
S=100000;%系统套数，建议大于10^5
%n=5;%每套系统中的节点总数5，6，7，8，……，19，20
k=5;%固定为5
max=90000;%最大使用寿命
Mu_1=2.72*10^4;%A寿命期望
Mu_2=3.32*10^5;%B寿命期望
A=[0.3 0.6];
B=0.33;
T=zeros(1,S);%记录每套系统第一次出现故障的时间
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%切换器状态（A4种B3种）
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for n=5:20
    t=zeros(2,n);%记录一套系统中n个节点的寿命,第一行A，第二行B
    for s=1:S
        t(1,:)=exprnd(Mu_1,[1,n]);
        t(2,:)=exprnd(Mu_2,[1,n]);
        t=round(t);
        t(t<1)=1;
        t(t>max)=max;
        tA=rand(1,n);
        type_A=ones(1,n);
        type_A(tA>=A(1)&tA<A(2))=2;
        type_A(tA>=A(2))=3;
        tB=rand(1,n);
        type_B=ones(1,n);
        type_B(tB>=B)=2;
        %%%%%%%%%%%%%%%%%%%%%%%%
        for i=sort(reshape(t,[1,2*n]))
            count=zeros(6);
            for j =1:n
                nnn=1;
                if t(1,j)<=i
                    if t(2,j)<=i
                        nnn=node(type_A(j),type_B(j));
                    else
```

```
                    nnn=node(type_A(j),0);
                end
            elseif t(2,j)<=i
                nnn=node(0,type_B(j));
            end
            count(nnn)=count(nnn)+1;
        end
        if system(count,k)
            T(s)=i;
            break;
        end
    end
end
T_f=mean(T);
R=find(T>25000);
R_w=length(R)/S;
disp(['n=',num2str(n)]);
disp(['T_f=',num2str(T_f)]);
disp(['R_w=',num2str(R_w)]);
end
toc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%节点状态判断
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function output=node(a,b)
if a+b==0
    output=1;
elseif a==3
    output=5;
elseif a==1&&b==2
    output=2;
elseif a==2&&b==1
    output=4;
elseif a==2&&b==2
    output=5;
elseif a*b==1
    output=6;
elseif a==0&&b==2
    output=2;
elseif a==1&&b==0
    output=2;
elseif a==2&&b==0
    output=3;
elseif a==0&&b==1
```

```
    output=4;
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%系统状态判断
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function out=system(count,k)
out=0;
if count(6)      %C1
    out=1;
elseif count(4)>=2 %C2
    out=1;
elseif count(1)+count(3)+count(4)==0 %C3
    out=1;
elseif count(1)+count(2)+((count(3)+count(4))>0)<k %C4
    out=1;
elseif (count(4)==0)&&count(1)&&count(3)&&(count(1)+count(2)==(k-1))%C8&&C9
    Pr=count(3)/(count(3)+count(1));
    if rand(1)>Pr
        out=1;
    end
end
end
```

```
tic
%%%%%%理论计算
w=25000;
k=5;
lambda_1=1/(2.72*10^4);%A
lambda_2=1/(3.31*10^5);%B
PA=1-exp(-lambda_1*w);%A损坏的概率
PB=1-exp(-lambda_2*w);%B损坏的概率
A0=exp(-lambda_1*w);
B0=exp(-lambda_2*w);
A1=PA*0.3;
A2=A1;
A3=PA*0.4;
B1=PB*0.33;
B2=PB*0.67;

PF=A0*B0;
SO=A0*B2+A1*B0+A1*B2;
DM=A2*B0;
MO=A0*B1+A2*B1;
DN=A2*B2+A3;
```

```
BF=A1*B1;

for n=5:20
    p=0;
    for n_1=0:n
        for n_2=0:n
            for n_3=0:n
                for n_4=0:n
                    for n_5=0:n
                        for n_6=0:n
                            if n_1+n_2+n_3+n_4+n_5+n_6==n
                                ooo=S(n_1,n_2,n_3,n_4,n_5,n_6,k);
                                if ooo
                                    p=p+ooo*nchoosek(n,n_1)*PF^n_1*nchoosek(n-n_1,n_2)*SO^n_2*nchoosek(n-n_1-n_2,n_3)*DM^n_3*nchoosek(n-n_1-n_2-n_3,n_4)*MO^n_4*nchoosek(n-n_1-n_2-n_3-n_4,n_5)*DN^n_5*BF^n_6;
                                end
                            end
                        end
                    end
                end
            end
        end
    end
    disp(['n=',num2str(n)]);
    disp(['R_w=',num2str(1-p)]);
end
toc
function out=S(n1,n2,n3,n4,~,n6,k)
out=0;
if n6        %C1
    out=1;
elseif n4>=2 %C2
    out=1;
elseif n1+n3+n4==0   %C3
    out=1;
elseif n1+n2+((n3+n4)>0)<k %C4
    out=1;
elseif (n4==0)&&n1&&n3&&(n1+n2==(k-1))%C8&&C9
    Pr=n1/(n3+n1);
    out=Pr;
    end
end
```