一个 RS485 多机通信系统的可靠性评估的建模和仿真

组号：26　　李澄曜 5140219288　　林特 5140219432

摘要：本次实验以一个 RS485 多机通信系统为对象，研究其可靠性问题。将该系统的物理结构和故障类型进行数学建模，提出相应仿真算法，得到系统在不同条件下的人工修理次数、系统（重大）故障发生次数，平均连续无（重大）故障运行时间等重要参数。同时针对几种改进措施进行建模，得到相应条件下的仿真结果，从而为系统改进提出合理化的建议。

关键词：RS485，主从多机，可靠性，数学建模

Modeling and Simulation of Evaluation of Reliability of a RS485 Multi-machine Communication System

Abstract： This article focuses on a RS485 multi-machine communication system and studies its reliability. We try to model the physical structure and different kinds of breakdown of the system, put forward a simulation algorithm and obtain a series of important parameters. Meanwhile we try to model a few improvements and get the corresponding results, in order to give more reasonable advice.

Key words： RS485, Master-slave Communication System, reliability, modeling

1 模型建构

1.1 RS485 系统模型

接口电路、控制电路、集线器均由特定个数的元件有特定的组合方式（串联组合、并联组合、k-out-of-n 组合）组合而成。具体如下：

(1) 控制电路板可看作两个部件组成:“控制硬部件”对应硬件电路部分,“控制软部件”对应软件程序部分。

(2) 控制硬部件可看作由 41 个统计特性独立元件构成，并适用串联组合。

(3) 控制软部件可看作单元件构成。

(4) 未使用措施 2 前,接口电路板可看作由 19 个统计特性独立元件构成的“接口部件”,并适用串联组合；使用措施 2 后，接口电路板可看作由 27 个统计特性独立元件构成的“接口部件”，并适用串联组合。

(5) 集线器可看作由 11 个统计特性独立元件构成的“集线器部件”，并适用串联组合。

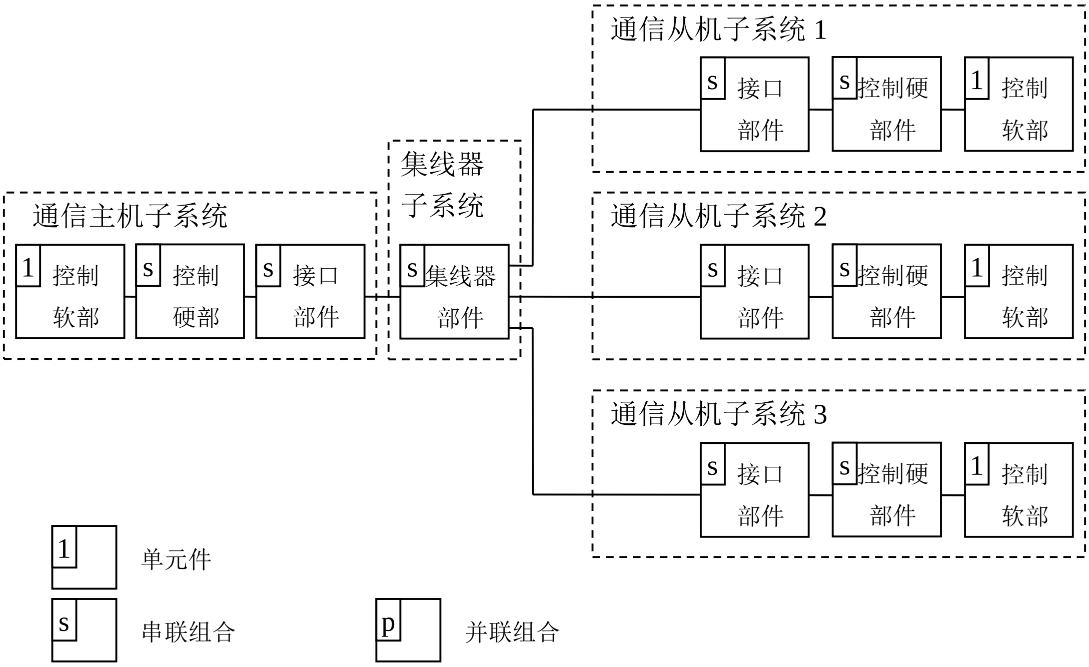


图 1-1 RS485 系统完整模型

1.2 故障模型

1.2.1 故障参数设定

所有元件连续无故障运行时间t的概率密度分布都遵从负指数分布

$$f_T(t)=\lambda e^{-\lambda t} \tag{1-1}$$

其中λ为常数，对不同的元件可以取不同的值。

构成控制硬部件、接口部件、集线器部件的所有元件特性满足独立同分布，对应参数$1/\lambda_1=9.3\times10^4 hour$。构成控制软部件的单元件，对应分布参数$1/\lambda_2=5.7\times10^3 hour$。对故障进行人工修理，排除故障所需花费的时间$t'$是随机变量，其概率密度分布遵从负指数分布$f_{T'}(t')=\lambda_1' e^{-\lambda' t'}$，其中$1/\lambda_1'=9.5hour$。控制软部件由 Watchdog 重启从故障状态中恢复，所需花费的时间是随机变量，其概率密度分布遵从负指数分布，参数$1/\lambda_2'=1hour$

1.2.2 故障结果模型

(1) 控制硬部件、集线器部件、接口部件组成元件的故障需要依靠人工修复。未采取措施 1 时，控制软部件组成元件的故障需要依靠人工修复；采取措施 1 以后，控制软部件组成元件的故障中有 98.2%可以通过自动重启恢复，还有 1.8%的故障需要依靠人工修复。

(2)未采取措施 2 时，接口部件组成元件的故障中有 35%会引发总线阻塞；采取措施 2 以后，可完全避免引发总线阻塞（但没有改变单个元件故障发生率）。

(3) 未采取措施 3 时，通信主机子系统一旦有故障即刻完全失效。采取措施 3 以后，在互为备份的双机同时失效时，子系统失效，即双机适用并联组合。不过这不是严格意义上的并联组合，因为如果未同时采取措施 2，双机热备配置中的一台若接口电路硬件故障引发总线阻塞，仍会造成系统失效。

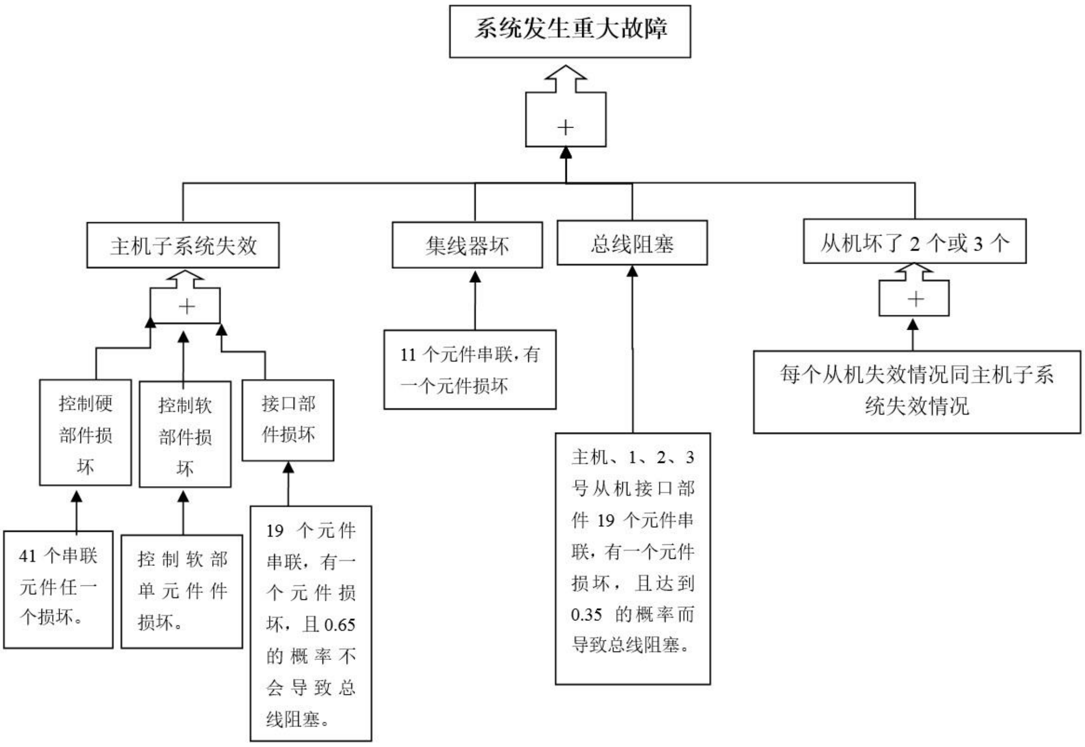

系统发生重大故障
+
主机子系统失效
集线器坏
总线阻塞
从机坏了 2 个或 3 个
+
+
控制硬部件损坏
控制软部件损坏
接口部件损坏
11 个元件串联，有一个元件损坏
每个从机失效情况同主机子系统失效情况
41 个串联元件任一个损坏。
控制软部单元件损坏。
19 个元件串联，有一个元件损坏，且 0.65 的概率不会导致总线阻塞。
主机、1、2、3 号从机接口部件 19 个元件串联，有一个元件损坏，且达到 0.35 的概率而导致总线阻塞。

图 1-2 基本配置下系统故障树

1.3 解决方案模型

(1) watchdog“看门狗”

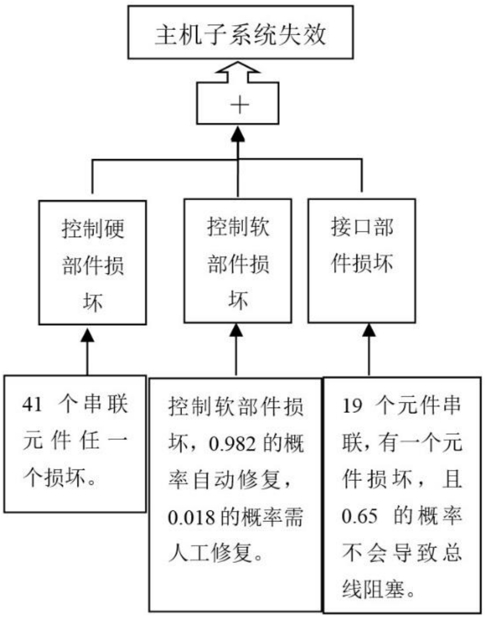
改进后主机部分子系统失效的故障树见图 1-3。

(2) 防总线阻塞电路

增加 8 个接口部件，可以完全避免总线阻塞发生。

(3) 主机双机热备

改进后减少了因主机失效导致的故障。故障树见图 1-4。


主机子系统失效
+
控制硬部件损坏
控制软部件损坏
接口部件损坏
41 个串联元件任一个损坏。
控制软部件损坏，0.982 的概率自动修复，0.018 的概率需人工修复。
19 个元件串联，有一个元件损坏，且 0.65 的概率不会导致总线阻塞。

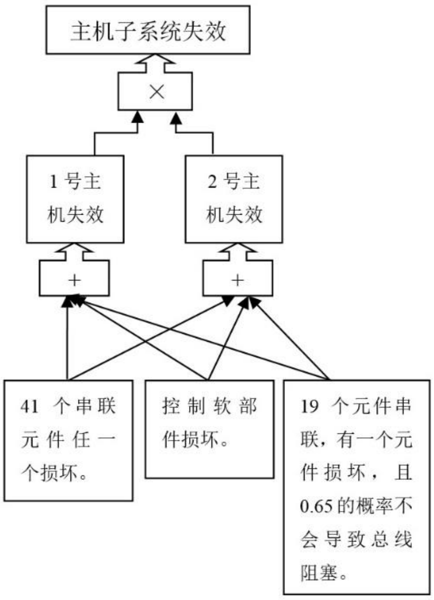


图 1-3 采用措施一后主机子系统故障树　　图 1-4 采用双机设备主机子系统故障树

2 算法设计

2.1 算法思路

我们采取定步长思路求解这一问题。其核心思路为：

(1) 分别为软部件、硬部件、接口部件、集线器部件分配状态矩阵，为接口部件单独分配总线阻塞状态矩阵，初始状态均置零；

(2) 按照软部件、硬部件、接口部件的顺序遍历所有通信机中的元件，根据不同状态转移概率修改状态矩阵以及总线阻塞状态矩阵的值。当由正常状态转移至故障状态时，修理次数加一。之后遍历集线器中的元件，以同样方式修改集线器部件状态矩阵的值；

(3) 设前一小时系统故障标志位，此时刻系统故障标志位。同样地，设前一小时系统严重故障标志位以及此时刻系统重大故障标志位。根据此时刻是否有坏元件判断主机、从机子系统是否故障以及系统是否故障，如出现故障则对应故障次数加一。根据子系统状态、主机状态、总线阻塞状态及集线器状态判断系统是否重大故障，如出现系统重大故障则重大故障次数加一；

(4) 根据前一小时故障标志位及此时刻故障标志位判断各时间指标是否加一步长；

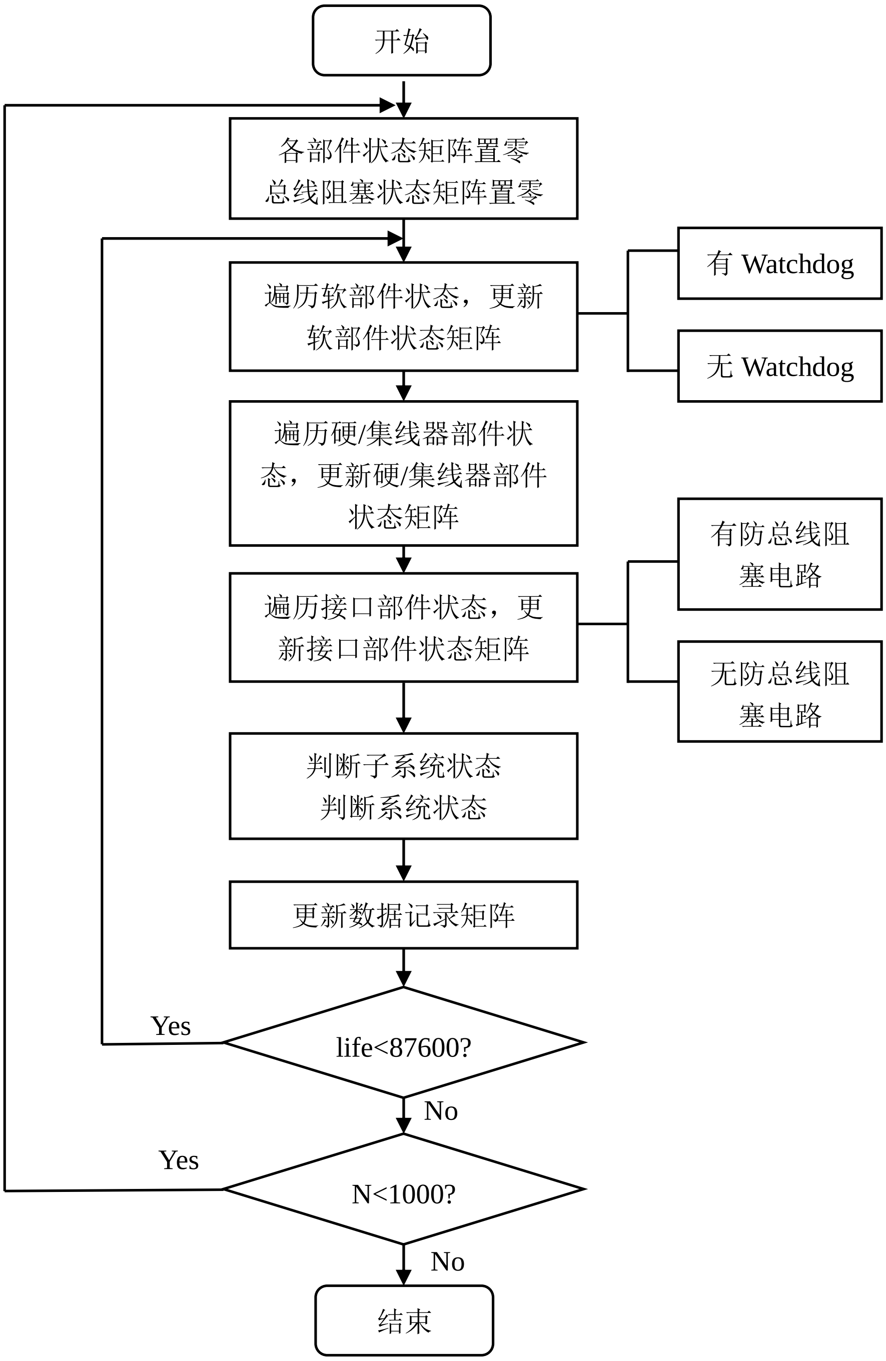
(5) 回到步骤(2)，开始下一小时的仿真。

(6) 当仿真时间达到设置值时，将此次仿真所得各参数填入结果记录数组，回到步骤(1)开始下一次仿真。

2.2 算法讨论

在程序编写的过程中发现，将不同部件的循环分开可大大提高代码的可重复利用性。在不同措施组合时，可直接将对应循环进行组合，修改小部分参数（如主机数量等）以及结果判断语句（在增加主机数量时），即可合成出对应代码。这样既提高效率，也提高了程序的可读性。

2.2 程序框图



3 仿真结果

我们针对每一种可能的措施组合进行了仿真，数据记录如下：

表 3-1 仿真结果数据记录

采取的措施	无	措施 1	措施 2	措施 3	措施12	措施13	措施23	三种措施
平均修理次数	297	237	328	368	265	292	407	336
平均故障次数	287	289	315	351	315	356	388	390
平均重大故障次数	102	101	93	44	90	45	16	15
平均无故障运行时间(小时)	84680	85162	84377	83980	84898	84608	84185	84184
平均连续无故障运行时间(小时)	294	294	266	237	268	237	215	215
平均连续无重大故障运行时间(小时)	850	859	933	1976	967	1997	6195	6195

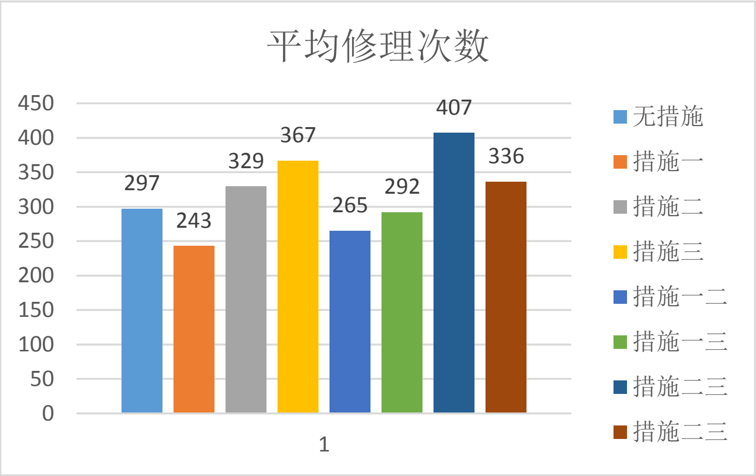


图 3-1 平均修理次数记录

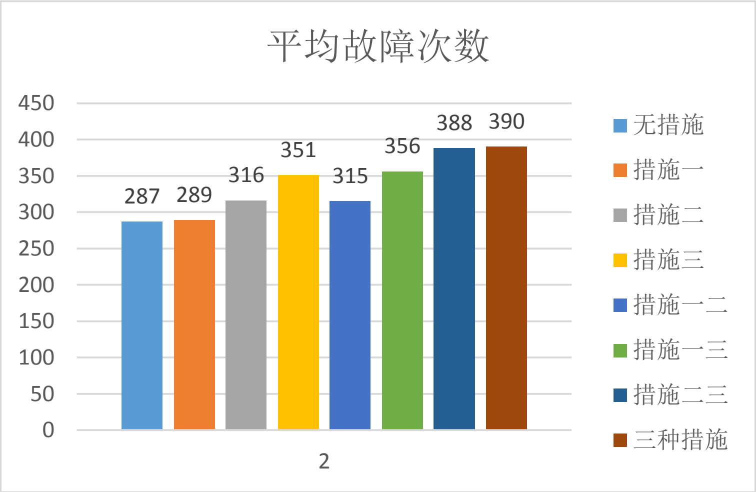


图 3-2 平均故障次数记录

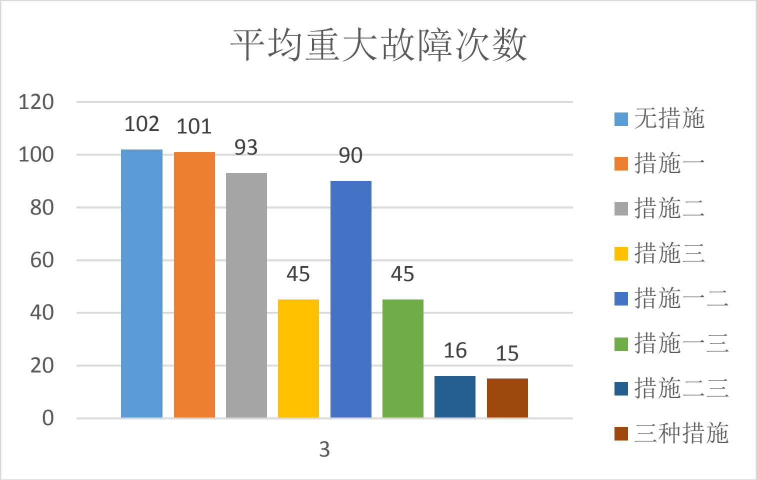


图 3-3 平均重大故障次数记录

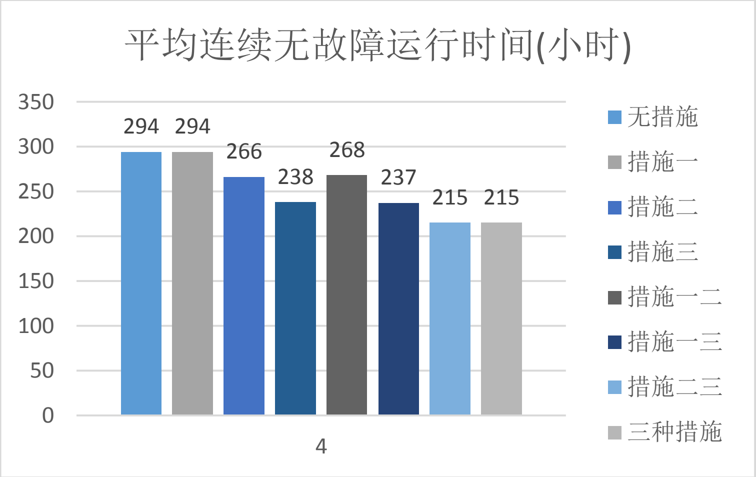


图 3-4 平均连续无故障运行时间记录

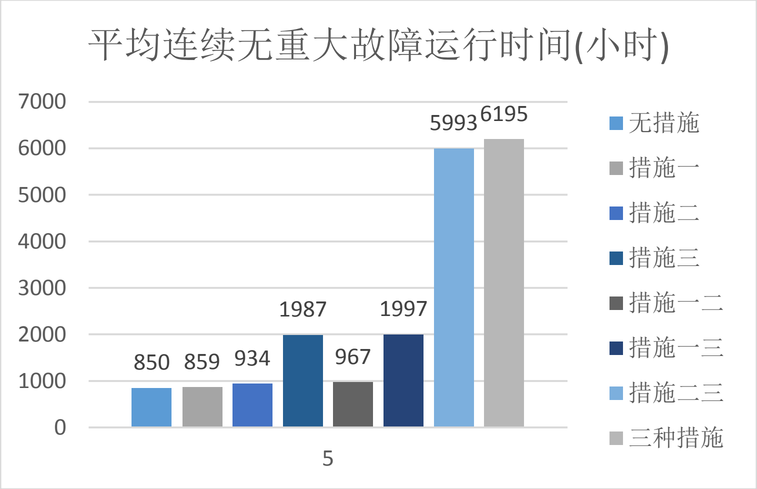


图 3-5 平均连续无重大故障运行时间记录

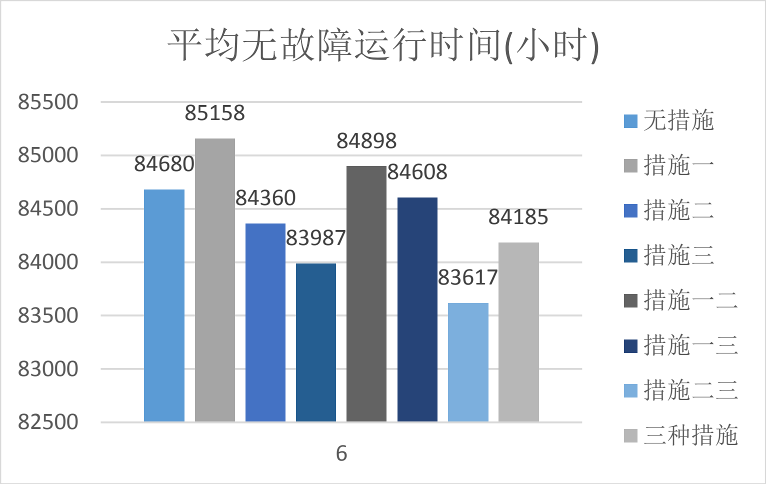


图 3-6 平均无故障运行时间记录

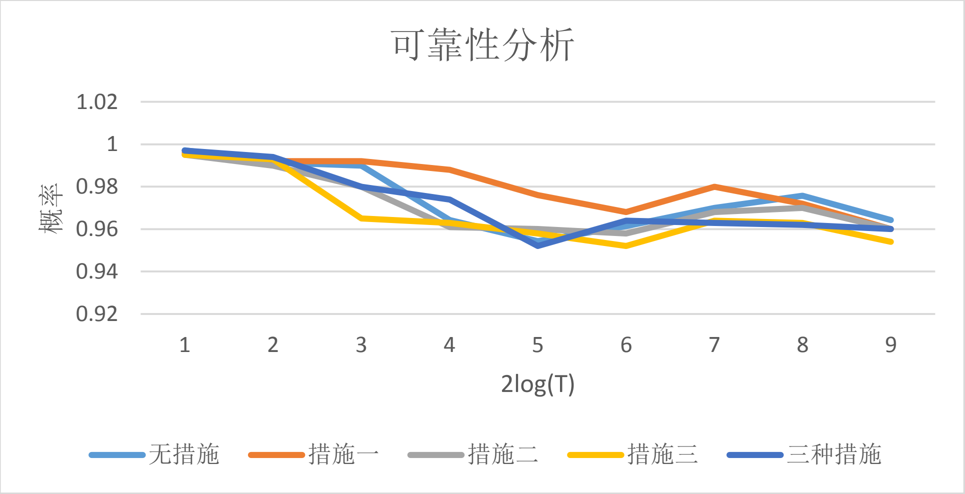


图 3-7 系统可靠性记录

4 结论及建议

4.1结论

(1)采用 watchdog 可以显著减少人工修理次数，但由于 watchdog 只是一种自动修复的机制，它只能做到尽快自动修复系统，不能减少发生故障次数。但是可显著提高平均无故障运行时间。

(2)已知的所有措施均无法降低平均故障次数。相反，引入新的接口元件及主机均使系统元件总数上升，在其他条件没有改变的情况下会大幅提高故障次数。

(3)增加双机热备对降低重大故障次数有较大帮助，而增加接口部件尽管从理论上杜绝了发生总线阻塞的可能性，反映在实际中对重大故障次数的影响却并不大。猜测其原因是原本接口部件坏只有 35%的概率发生总线接口阻塞，采用增加元件改进接口的方式，虽然没有阻塞系统阻塞的情况，但是增加了 8 个元件。这对于减小发生总线阻塞这种重大故障的概率并不明显，反而由于引入新的原件，导致发生故障概率大幅增加。因此，对于这个系统单独采取增加元件改进接口的方式来提高系统稳定性是不明智的。

4.2 合理化建议

（1） 若想尽可能减少系统重大故障，采取措施二和措施三或同时采取三种措施均可。

（2） 如果人工修理比较麻烦，费用较高，希望这个系统尽量减小人工修理费用，可以考

虑采用 watchdog 方式改进原来系统。

(3) 为了满足实际情况中的不同需要，可以选择不同的改进方式。

5 拓展探究

从实验结果可以看出，我们所给出的仿真模型绝大多数系统指标与老师所给结果接近甚至相同。但是其中“平均连续无重大故障运行时间”稍有差别。因此，我们有必要分析造成这种差别的原因：是由仿真过程中的偶然性造成的，还是由算法逻辑的差别即必然性造成的。考虑到老师所使用的思路及算法具有一定的权威性，我们将老师所给出的“平均连续无重大故障运行时间”作为总体均值，并对我们所得到的结果做假设检验。

5.1 理论基础

5.1.1 独立同分布中心极限定理

设$\{X_n\}$为独立的随机序列，且$\mathrm{E}\{X_i\}=\mu$，$\mathrm{D}\{X_i\}=\sigma^2>0$，$(i=1,2,\cdots)$，记

$$Y_n^*=\frac{\sum_{k=1}^n X_k-n\mu}{\sigma\sqrt{n}}$$

则对任意实数y，有

$$\lim_{n\to\infty}P(Y_n^*\le y)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{y}e^{-\frac{x^2}{2}}dx=\Phi(y)$$

由本定理可知，不管$\{X_n\}$服从什么分布，是离散型的还是连续型的，只要$\{X_n\}$独立同分布并且方差存在，那么当n很大时$(n\ge 50)$，$\sum_{i=1}^n X_i$的近似分布为正态分布。

同样的，可以推出

$$Z_n^*=\frac{\frac{1}{n}\sum_{i=1}^n X_i-\frac{1}{n}\sum_{i=1}^n E(X_i)}{\sqrt{D(\frac{1}{n}\sum_{i=1}^n X_i)}}$$

即Z_n^*的极限分布为标准正态分布，$\frac{1}{n}\sum_{i=1}^n X_i$的近似分布也为正态分布。

本仿真模型中，马尔科夫链状态转移机模型以及判断总线阻塞可能性时均使用了均匀分布随机数函数 *rand()*。该函数决定了每次仿真所得到的“平均连续无重大故障运行时间”是一个随机变量，记为$\{Y_n\}$。设该随机变量的总体均值$E(X_i)$为μ，总体方差$D(X_i)$为σ^2。同时，每次仿真代码相同且互不影响，可以认为该随机变量独立同分布。根据独立同分布中心极限定理，取$n=50$为一组计算均值，$\frac{1}{n}\sum_{i=1}^n Y_i$服从正态分布$\mathrm{N}(\mu,\frac{\sigma^2}{n})$。

5.1.2 假设检验

设$\mathrm{Z}=\frac{1}{n}\sum_{i=1}^n Y_i$，$(Z_1,\cdots Z_k)$为来自总体 Z 的一个样本，$\mathrm{Z}\sim\mathrm{N}(\mu,\frac{\sigma^2}{n})$，给定显著性水平$\alpha(0<\alpha<1)$，构造检验假设为

$$H_0:\mu=\mu_0，H_1:\mu\ne\mu_0（\mu_0为已知）$$

由于本模型中难以从理论推导出总体方差$D(X_i)$，故采用方差未知的情形下计算。

构造估计量

$$\mathrm{T}=\frac{\bar{Z}-\mu_0}{S/\sqrt{k}}$$

当$H_0:\mu=\mu_0$为真时，统计量$\mathrm{T}\sim\mathrm{t}(\mathrm{n}-1)$。当给定水平为$\alpha$时，拒绝域为$\mathrm{W}=\Big(-\infty,$

$-t_{\frac{\alpha}{2}}(n-1)\Big] \cup \Big[t_{\frac{\alpha}{2}}(n-1)，+\infty)$。

5.2 数据计算

从“不采取任何措施”模型中的“平均连续无重大故障运行时间”数据结果中取 1000 个数据，取$n = 50$为一组计算均值，得到 20 组均值$(Z_1,\cdots Z_{20})$ =(857,835,859,838,865,843,852,849,851,857,853,851,850,860,859,842,859,842,845,853)。

5.3 检验结果

总体均值取老师所给结果$\mu_0 = 837$。取显著性水平$\alpha = 0.05$。查表得拒绝域

$$\mathrm{W} = (-\infty，\ -2.0390) \cup (2.0390，\ +\infty)$$

计算得 10 组均值的平均值$\bar{Z}$=851.7，S=62.29。可得

$$\mathrm{T} = 7.93 \in \mathrm{W}$$

拒绝原假设，故“平均连续无重大故障运行时间”并非由仿真过程中的偶然性造成。

5.4 检验结果分析

我们猜测该检验结果与程序中“平均连续无重大故障运行时间”的计算方法有关。为了尽快求得这一参数我们直接用“总无重大故障运行时间”除以“系统重大故障次数”。从实验结果中我们发现，这一结果随“系统重大故障次数”的变化上下跳变剧烈，很有可能向这一变量中添加了新的不稳定因素而没有清楚的认识，导致了“平均连续无重大故障运行时间”这一参数与老师参考答案稍有不同的结果。

附录

所使用的实验代码如下：

无措施代码：

```
life_soft = 5700;%软部件寿命
life_hard = 93000;%硬部件寿命
lamda_soft = 1/life_soft;
lamda_hard = 1/life_hard;
time_repair = 9.5;%修理时间期望
lamda_repair = 1/time_repair;

samples = 100;%仿真次数
record = zeros(samples,6);%数据记录矩阵
avail = zeros(samples,9);

prb_hard = 1 - exp(-lamda_hard);%硬部件状态转移概率
prb_soft = 1 - exp(-lamda_soft);%软部件状态转移概率
prb_repair = 1 - exp(-lamda_repair);%修理状态转移概率
tic;

for m = 1:samples
    time_record = zeros(1,6);

    soft_state = zeros(4,1);
    hard_state = zeros(4,41);
```

```
inter_state = zeros(4,19);
col_state = zeros(1,11);

sys_state = zeros(1,5);
fail_flag_old = 0;
fail_flag_new = 0;
srs_fail_flag_old = 0;
srs_fail_flag_new = 0;
stk_flag = zeros(4,19);

time_flag = 0;

for life = 1:87600
   for i = 1:4
      if (soft_state(i,1) == 0)
         if (rand(1) < prb_soft)
            soft_state(i,1) = 1;
            time_record(1) = time_record(1) + 1;
         %else soft_state(i,1) = 0;
         end
      else
         if (rand(1) < prb_repair)
            soft_state(i,1) = 0;
         %else soft_state(i,1) = 1;
         end
      end
   end

   for i = 1:4
      for j = 1:41
         if (hard_state(i,j) == 0)
            if (rand(1) < prb_hard)
               hard_state(i,j) = 1;
               time_record(1) = time_record(1) + 1;
            %else hard_state(i,j) = 0;
            end
         else
            if (rand(1) < prb_repair)
               hard_state(i,j) = 0;
            %else hard_state(i,j) = 1;
            end
         end
      end
   end
```

```
for i = 1:4
    for j = 1:19
        if (inter_state(i,j) == 0)
            if (rand(1) < prb_hard)
                inter_state(i,j) = 1;
                time_record(1) = time_record(1) + 1;
                if (rand(1)<0.35)
                    stk_flag(i,j) = 1;
                else stk_flag(i,j) = 0;
                end
            else
                %inter_state(i,j) = 0;
                stk_flag(i,j) = 0;
            end
        else
            if (rand(1) < prb_repair)
                inter_state(i,j) = 0;
                stk_flag(i,j) = 0;
            %else inter_state(i,j) = 1;
            end
        end
    end
end

for i = 1:11
    if (col_state(i) == 0)
        if (rand(1) < prb_hard)
            col_state(i) = 1;
            time_record(1) = time_record(1) + 1;
        %else col_state(i) = 0;
        end
    else
        if (rand(1) < prb_repair)
            col_state(i) = 0;
        %else col_state(i) = 1;
        end
    end
end

fail_flag_old = fail_flag_new;
srs_fail_flag_old = srs_fail_flag_new;
```

```
        sum_comp = sum(soft_state(:)) + sum(hard_state(:)) +
sum(inter_state(:)) + sum(col_state(:));

        if (sum_comp == 0)
            fail_flag_new = 0;
            time_flag = time_flag + 1;
            time_record(6) = time_record(6) + 1;
        else
            fail_flag_new = 1;
            time_flag = 0;
        end

        for i = 1:4
            if (sum(soft_state(i,:)) + sum(hard_state(i,:)) +
sum(inter_state(i,:)) > 0)
                sys_state(i) = 1;
            else sys_state(i) = 0;
            end
        end

        if (sum(col_state(:))>0)
            sys_state(5) = 1;
        else sys_state(5) = 0;
        end

        sum_state = sys_state(2) + sys_state(3) + sys_state(4);

        if (sys_state(1) || sys_state(5) || sum(stk_flag(:)) > 0 ||
sum_state >= 2)
            srs_fail_flag_new = 1;
        else srs_fail_flag_new = 0;
        end

        if (fail_flag_old > fail_flag_new)
            time_record(2) = time_record(2) + 1;
        end

        if (fail_flag_new + fail_flag_old == 0)
            time_record(4) = time_record(4) + 1;
        end

        if (srs_fail_flag_old > srs_fail_flag_new)
            time_record(3) = time_record(3) + 1;
        end
```

```
if (srs_fail_flag_new + srs_fail_flag_old == 0)
   time_record(5) = time_record(5) + 1;
end

if (life == 87600)
   time_record(4) = time_record(4) - time_flag;
   time_record(5) = time_record(5) - time_flag;
end

if (life == 1)
   if (fail_flag_new == 0)
      avail(m,1) = 1;
   end
end

if (life == 2)
   if (fail_flag_new == 0)
      avail(m,2) = 1;
   end
end

if (life == 8)
   if (fail_flag_new == 0)
      avail(m,3) = 1;
   end
end

if (life == 20)
   if (fail_flag_new == 0)
      avail(m,4) = 1;
   end
end

if (life == 100)
   if (fail_flag_new == 0)
      avail(m,5) = 1;
   end
end

if (life == 1000)
   if (fail_flag_new == 0)
      avail(m,6) = 1;
   end
```

```
        end
        
        if (life == 8760)
            if (fail_flag_new == 0)
                avail(m,7) = 1;
            end
        end
        
        if (life == 43800)
            if (fail_flag_new == 0)
                avail(m,8) = 1;
            end
        end
        
        if (life == 87600)
            if (fail_flag_new == 0)
                avail(m,9) = 1;
            end
        end
        
    end
    record(m,1)=time_record(1);
    record(m,2)=time_record(2);
    record(m,3)=time_record(3);
    record(m,4)=time_record(4)/time_record(2);
    record(m,5)=time_record(5)/time_record(3);
    record(m,6)=time_record(6);
    
    m
    t = toc
end
```

措施一代码：

```
life_soft = 5700;%软部件寿命
life_hard = 93000;%硬部件寿命
lamda_soft = 1/life_soft;
lamda_hard = 1/life_hard;
time_repair = 9.5;%修理时间期望
lamda_repair = 1/time_repair;
time_restart = 1;
lamda_restart = 1/time_restart;

samples = 100;%仿真次数
record = zeros(samples,6);%数据记录矩阵
```

```
avail = zeros(samples,9);

prb_hard = 1 - exp(-lamda_hard);%硬部件状态转移概率
prb_soft = 1 - exp(-lamda_soft);%软部件状态转移概率
prb_repair = 1 - exp(-lamda_repair);%修理状态转移概率
prb_restart = 1 - exp(-lamda_restart);
tic;

for m = 1:samples
    time_record = zeros(1,6);

    soft_state = zeros(4,1);
    hard_state = zeros(4,41);
    inter_state = zeros(4,19);
    col_state = zeros(1,11);

    sys_state = zeros(1,5);
    fail_flag_old = 0;
    fail_flag_new = 0;
    srs_fail_flag_old = 0;
    srs_fail_flag_new = 0;
    stk_flag = zeros(4,19);

    time_flag = 0;

    for life = 1:87600
        for i = 1:4
            if (soft_state(i,1) == 0)
                if (rand(1) < prb_soft)
                    soft_state(i,1) = 1;
                    %time_record(1) = time_record(1) + 1;
                %else soft_state(i,1) = 0;
                end
            else
                if (rand(1) < 0.982)
                    if (rand(1) < prb_restart)
                        soft_state(i,1) = 0;
                    else soft_state(i,1) = 1;
                    end
                else
                    if (rand(1) < prb_repair)
                        soft_state(i,1) = 0;
                        time_record(1) = time_record(1) + 1;
                    else soft_state(i,1) = 1;
                    end
                end
            end
        end

        for i = 1:4
            for j = 1:41
                if (hard_state(i,j) == 0)
                    if (rand(1) < prb_hard)
                        hard_state(i,j) = 1;
                        time_record(1) = time_record(1) + 1;
                    %else hard_state(i,j) = 0;
                    end
                else
                    if (rand(1) < prb_repair)
                        hard_state(i,j) = 0;
                    %else hard_state(i,j) = 1;
                    end
                end
            end
        end

        for i = 1:4
            for j = 1:19
                if (inter_state(i,j) == 0)
                    if (rand(1) < prb_hard)
                        inter_state(i,j) = 1;
                        time_record(1) = time_record(1) + 1;
                        if (rand(1)<0.35)
                            stk_flag(i,j) = 1;
                        else stk_flag(i,j) = 0;
                        end
                    else
                        %inter_state(i,j) = 0;
                        stk_flag(i,j) = 0;
                    end
                else
                    if (rand(1) < prb_repair)
                        inter_state(i,j) = 0;
                        stk_flag(i,j) = 0;
                    %else inter_state(i,j) = 1;
                    end
                end
            end
```

```
        end

        for i = 1:11
            if (col_state(i) == 0)
                if (rand(1) < prb_hard)
                    col_state(i) = 1;
                    time_record(1) = time_record(1) + 1;
                %else col_state(i) = 0;
                end
            else
                if (rand(1) < prb_repair)
                    col_state(i) = 0;
                %else col_state(i) = 1;
                end
            end
        end

        fail_flag_old = fail_flag_new;
        srs_fail_flag_old = srs_fail_flag_new;

        sum_comp = sum(soft_state(:)) + sum(hard_state(:)) +
sum(inter_state(:)) + sum(col_state(:));

        if (sum_comp == 0)
            fail_flag_new = 0;
            time_flag = time_flag + 1;
            time_record(6) = time_record(6) + 1;
        else
            fail_flag_new = 1;
            time_flag = 0;
        end

        for i = 1:4
            if (sum(soft_state(i,:)) + sum(hard_state(i,:)) +
sum(inter_state(i,:)) > 0)
                sys_state(i) = 1;
            else sys_state(i) = 0;
            end
        end

        if (sum(col_state(:))>0)
            sys_state(5) = 1;
        else sys_state(5) = 0;
        end
```

```
      sum_state = sys_state(2) + sys_state(3) + sys_state(4);

      if (sys_state(1) || sys_state(5) || sum(stk_flag(:)) > 0 ||
sum_state >= 2)
          srs_fail_flag_new = 1;
      else srs_fail_flag_new = 0;
      end

      if (fail_flag_old > fail_flag_new)
          time_record(2) = time_record(2) + 1;
      end

      if (fail_flag_new + fail_flag_old == 0)
          time_record(4) = time_record(4) + 1;
      end

      if (srs_fail_flag_old > srs_fail_flag_new)
          time_record(3) = time_record(3) + 1;
      end

      if (srs_fail_flag_new + srs_fail_flag_old == 0)
          time_record(5) = time_record(5) + 1;
      end

      if (life == 87600)
          time_record(4) = time_record(4) - time_flag;
          time_record(5) = time_record(5) - time_flag;
      end

      if (life == 1)
          if (fail_flag_new == 0)
              avail(m,1) = 1;
          end
      end

      if (life == 2)
          if (fail_flag_new == 0)
              avail(m,2) = 1;
          end
      end

      if (life == 8)
          if (fail_flag_new == 0)
```

```
            avail(m,3) = 1;
        end
    end

    if (life == 20)
        if (fail_flag_new == 0)
            avail(m,4) = 1;
        end
    end

    if (life == 100)
        if (fail_flag_new == 0)
            avail(m,5) = 1;
        end
    end

    if (life == 1000)
        if (fail_flag_new == 0)
            avail(m,6) = 1;
        end
    end

    if (life == 8760)
        if (fail_flag_new == 0)
            avail(m,7) = 1;
        end
    end

    if (life == 43800)
        if (fail_flag_new == 0)
            avail(m,8) = 1;
        end
    end

    if (life == 87600)
        if (fail_flag_new == 0)
            avail(m,9) = 1;
        end
    end

end
record(m,1)=time_record(1);
record(m,2)=time_record(2);
record(m,3)=time_record(3);
```

```
    record(m,4)=time_record(4)/time_record(2);
    record(m,5)=time_record(5)/time_record(3);
    record(m,6)=time_record(6);

    m
    t = toc
end
```

措施二代码：

```
life_soft = 5700;%软部件寿命
life_hard = 93000;%硬部件寿命
lamda_soft = 1/life_soft;
lamda_hard = 1/life_hard;
time_repair = 9.5;%修理时间期望
lamda_repair = 1/time_repair;

samples = 100;%仿真次数
record = zeros(samples,6);%数据记录矩阵
avail = zeros(samples,9);

prb_hard = 1 - exp(-lamda_hard);%硬部件状态转移概率
prb_soft = 1 - exp(-lamda_soft);%软部件状态转移概率
prb_repair = 1 - exp(-lamda_repair);%修理状态转移概率
tic;

for m = 1:samples
    time_record = zeros(1,6);

    soft_state = zeros(4,1);
    hard_state = zeros(4,41);
    inter_state = zeros(4,27);
    col_state = zeros(1,11);

    sys_state = zeros(1,5);
    fail_flag_old = 0;
    fail_flag_new = 0;
    srs_fail_flag_old = 0;
    srs_fail_flag_new = 0;

    time_flag = 0;

    for life = 1:87600
        for i = 1:4
            if (soft_state(i,1) == 0)
        if (rand(1) < prb_soft)
            soft_state(i,1) = 1;
            time_record(1) = time_record(1) + 1;
        %else soft_state(i,1) = 0;
        end
    else
        if (rand(1) < prb_repair)
            soft_state(i,1) = 0;
        %else soft_state(i,1) = 1;
        end
    end
end

for i = 1:4
    for j = 1:41
        if (hard_state(i,j) == 0)
            if (rand(1) < prb_hard)
                hard_state(i,j) = 1;
                time_record(1) = time_record(1) + 1;
            %else hard_state(i,j) = 0;
            end
        else
            if (rand(1) < prb_repair)
                hard_state(i,j) = 0;
            %else hard_state(i,j) = 1;
            end
        end
    end
end

for i = 1:4
    for j = 1:27
        if (inter_state(i,j) == 0)
            if (rand(1) < prb_hard)
                inter_state(i,j) = 1;
                time_record(1) = time_record(1) + 1;
            %else inter_state(i,j) = 0;
            end
        else
            if (rand(1) < prb_repair)
                inter_state(i,j) = 0;
            %else inter_state(i,j) = 1;
            end
        end
```

```
            end
        end

        for i = 1:11
            if (col_state(i) == 0)
                if (rand(1) < prb_hard)
                    col_state(i) = 1;
                    time_record(1) = time_record(1) + 1;
                %else col_state(i) = 0;
                end
            else
                if (rand(1) < prb_repair)
                    col_state(i) = 0;
                %else col_state(i) = 1;
                end
            end
        end

        fail_flag_old = fail_flag_new;
        srs_fail_flag_old = srs_fail_flag_new;

        sum_comp = sum(soft_state(:)) + sum(hard_state(:)) +
sum(inter_state(:)) + sum(col_state(:));

        if (sum_comp == 0)
            fail_flag_new = 0;
            time_flag = time_flag + 1;
            time_record(6) = time_record(6) + 1;
        else
            fail_flag_new = 1;
            time_flag = 0;
        end

        for i = 1:4
            if (sum(soft_state(i,:)) + sum(hard_state(i,:)) +
sum(inter_state(i,:)) > 0)
                sys_state(i) = 1;
            else sys_state(i) = 0;
            end
        end

        if (sum(col_state(:))>0)
            sys_state(5) = 1;
        else sys_state(5) = 0;
```

```
end

sum_state = sys_state(2) + sys_state(3) + sys_state(4);

if (sys_state(1) || sys_state(5) || sum_state >= 2)
    srs_fail_flag_new = 1;
else srs_fail_flag_new = 0;
end

if (fail_flag_old > fail_flag_new)
    time_record(2) = time_record(2) + 1;
end

if (fail_flag_new + fail_flag_old == 0)
    time_record(4) = time_record(4) + 1;
end

if (srs_fail_flag_old > srs_fail_flag_new)
    time_record(3) = time_record(3) + 1;
end

if (srs_fail_flag_new + srs_fail_flag_old == 0)
    time_record(5) = time_record(5) + 1;
end

if (life == 87600)
    time_record(4) = time_record(4) - time_flag;
    time_record(5) = time_record(5) - time_flag;
end

if (life == 1)
    if (fail_flag_new == 0)
        avail(m,1) = 1;
    end
end

if (life == 2)
    if (fail_flag_new == 0)
        avail(m,2) = 1;
    end
end

if (life == 8)
    if (fail_flag_new == 0)
```

```
            avail(m,3) = 1;
        end
    end

    if (life == 20)
        if (fail_flag_new == 0)
            avail(m,4) = 1;
        end
    end

    if (life == 100)
        if (fail_flag_new == 0)
            avail(m,5) = 1;
        end
    end

    if (life == 1000)
        if (fail_flag_new == 0)
            avail(m,6) = 1;
        end
    end

    if (life == 8760)
        if (fail_flag_new == 0)
            avail(m,7) = 1;
        end
    end

    if (life == 43800)
        if (fail_flag_new == 0)
            avail(m,8) = 1;
        end
    end

    if (life == 87600)
        if (fail_flag_new == 0)
            avail(m,9) = 1;
        end
    end

end
record(m,1)=time_record(1);
record(m,2)=time_record(2);
record(m,3)=time_record(3);
```

```
    record(m,4)=time_record(4)/time_record(2);
    record(m,5)=time_record(5)/time_record(3);
    record(m,6)=time_record(6);

    m
    t = toc
end
```

措施三代码：

```
life_soft = 5700;%软部件寿命
life_hard = 93000;%硬部件寿命
lamda_soft = 1/life_soft;
lamda_hard = 1/life_hard;
time_repair = 9.5;%修理时间期望
lamda_repair = 1/time_repair;

samples = 100;%仿真次数
record = zeros(samples,6);%数据记录矩阵
avail = zeros(samples,9);

prb_hard = 1 - exp(-lamda_hard);%硬部件状态转移概率
prb_soft = 1 - exp(-lamda_soft);%软部件状态转移概率
prb_repair = 1 - exp(-lamda_repair);%修理状态转移概率
tic;

for m = 1:samples
    time_record = zeros(1,6);

    soft_state = zeros(5,1);
    hard_state = zeros(5,41);
    inter_state = zeros(5,19);
    col_state = zeros(1,11);

    sys_state = zeros(1,6);
    fail_flag_old = 0;
    fail_flag_new = 0;
    srs_fail_flag_old = 0;
    srs_fail_flag_new = 0;
    stk_flag = zeros(5,19);

    time_flag = 0;

    for life = 1:87600
        for i = 1:5
    if (soft_state(i,1) == 0)
        if (rand(1) < prb_soft)
            soft_state(i,1) = 1;
            time_record(1) = time_record(1) + 1;
        %else soft_state(i,1) = 0;
        end
    else
        if (rand(1) < prb_repair)
            soft_state(i,1) = 0;
        %else soft_state(i,1) = 1;
        end
    end
end

for i = 1:5
    for j = 1:41
        if (hard_state(i,j) == 0)
            if (rand(1) < prb_hard)
                hard_state(i,j) = 1;
                time_record(1) = time_record(1) + 1;
            %else hard_state(i,j) = 0;
            end
        else
            if (rand(1) < prb_repair)
                hard_state(i,j) = 0;
            %else hard_state(i,j) = 1;
            end
        end
    end
end

for i = 1:5
    for j = 1:19
        if (inter_state(i,j) == 0)
            if (rand(1) < prb_hard)
                inter_state(i,j) = 1;
                time_record(1) = time_record(1) + 1;
                if (rand(1)<0.35)
                    stk_flag(i,j) = 1;
                else stk_flag(i,j) = 0;
                end
            else
                %inter_state(i,j) = 0;
                stk_flag(i,j) = 0;
```

```
                    end
                else
                    if (rand(1) < prb_repair)
                        inter_state(i,j) = 0;
                        stk_flag(i,j) = 0;
                    %else inter_state(i,j) = 1;
                    end
                end
            end
        end

        for i = 1:11
            if (col_state(i) == 0)
                if (rand(1) < prb_hard)
                    col_state(i) = 1;
                    time_record(1) = time_record(1) + 1;
                %else col_state(i) = 0;
                end
            else
                if (rand(1) < prb_repair)
                    col_state(i) = 0;
                %else col_state(i) = 1;
                end
            end
        end

        fail_flag_old = fail_flag_new;
        srs_fail_flag_old = srs_fail_flag_new;

        for i = 1:5
            if (sum(soft_state(i,:)) + sum(hard_state(i,:)) +
sum(inter_state(i,:)) > 0)
                sys_state(i) = 1;
            else sys_state(i) = 0;
            end
        end

        if (sum(col_state(:))>0)
            sys_state(6) = 1;
        else sys_state(6) = 0;
        end

        sum_comp1 = sys_state(1) + sys_state(3) + sys_state(4) +
sys_state(5) + sys_state(6);
```

```
    sum_comp2 = sys_state(2) + sys_state(3) + sys_state(4) +
sys_state(5) + sys_state(6);

    if (sum_comp1 + sum(stk_flag(2,:)) == 0 && sum_comp2 +
sum(stk_flag(1,:)) == 0)
      fail_flag_new = 0;
      time_flag = time_flag + 1;
      time_record(6) = time_record(6) + 1;
    else
      fail_flag_new = 1;
      time_flag = 0;
    end

    sum_state = sys_state(3) + sys_state(4) + sys_state(5);

    if (sys_state(1) + sys_state(2) >= 2 || sys_state(6) ||
sum(stk_flag(:)) > 0 || sum_state >= 2)
      srs_fail_flag_new = 1;
    else srs_fail_flag_new = 0;
    end

    if (fail_flag_old > fail_flag_new)
      time_record(2) = time_record(2) + 1;
    end

    if (fail_flag_new + fail_flag_old == 0)
      time_record(4) = time_record(4) + 1;
    end

    if (srs_fail_flag_old > srs_fail_flag_new)
      time_record(3) = time_record(3) + 1;
    end

    if (srs_fail_flag_new + srs_fail_flag_old == 0)
      time_record(5) = time_record(5) + 1;
    end

    if (life == 87600)
      time_record(4) = time_record(4) - time_flag;
      time_record(5) = time_record(5) - time_flag;
    end

    if (life == 1)
      if (fail_flag_new == 0)
```

```
        avail(m,1) = 1;
    end
end

if (life == 2)
    if (fail_flag_new == 0)
        avail(m,2) = 1;
    end
end

if (life == 8)
    if (fail_flag_new == 0)
        avail(m,3) = 1;
    end
end

if (life == 20)
    if (fail_flag_new == 0)
        avail(m,4) = 1;
    end
end

if (life == 100)
    if (fail_flag_new == 0)
        avail(m,5) = 1;
    end
end

if (life == 1000)
    if (fail_flag_new == 0)
        avail(m,6) = 1;
    end
end

if (life == 8760)
    if (fail_flag_new == 0)
        avail(m,7) = 1;
    end
end

if (life == 43800)
    if (fail_flag_new == 0)
        avail(m,8) = 1;
    end
```

```
        end

        if (life == 87600)
            if (fail_flag_new == 0)
                avail(m,9) = 1;
            end
        end

    end
    record(m,1)=time_record(1);
    record(m,2)=time_record(2);
    record(m,3)=time_record(3);
    record(m,4)=time_record(4)/time_record(2);
    record(m,5)=time_record(5)/time_record(3);
    record(m,6)=time_record(6);

    m
    t = toc
end
```

措施一、二代码：

```
life_soft = 5700;%软部件寿命
life_hard = 93000;%硬部件寿命
lamda_soft = 1/life_soft;
lamda_hard = 1/life_hard;
time_repair = 9.5;%修理时间期望
lamda_repair = 1/time_repair;
time_restart = 1;
lamda_restart = 1/time_restart;

samples = 100;%仿真次数
record = zeros(samples,6);%数据记录矩阵
avail = zeros(samples,9);

prb_hard = 1 - exp(-lamda_hard);%硬部件状态转移概率
prb_soft = 1 - exp(-lamda_soft);%软部件状态转移概率
prb_repair = 1 - exp(-lamda_repair);%修理状态转移概率
prb_restart = 1 - exp(-lamda_restart);
tic;

for m = 1:samples
    time_record = zeros(1,6);

    soft_state = zeros(4,1);
```

```
hard_state = zeros(4,41);
inter_state = zeros(4,27);
col_state = zeros(1,11);

sys_state = zeros(1,5);
fail_flag_old = 0;
fail_flag_new = 0;
srs_fail_flag_old = 0;
srs_fail_flag_new = 0;

time_flag = 0;

for life = 1:87600
    for i = 1:4
        if (soft_state(i,1) == 0)
            if (rand(1) < prb_soft)
                soft_state(i,1) = 1;
                %time_record(1) = time_record(1) + 1;
            %else soft_state(i,1) = 0;
            end
        else
            if (rand(1) < 0.982)
                if (rand(1) < prb_restart)
                    soft_state(i,1) = 0;
                else soft_state(i,1) = 1;
                end
            else
                if (rand(1) < prb_repair)
                    soft_state(i,1) = 0;
                    time_record(1) = time_record(1) + 1;
                else soft_state(i,1) = 1;
                end
            end
        end
    end

    for i = 1:4
        for j = 1:41
            if (hard_state(i,j) == 0)
                if (rand(1) < prb_hard)
                    hard_state(i,j) = 1;
                    time_record(1) = time_record(1) + 1;
                %else hard_state(i,j) = 0;
                end
            else
                if (rand(1) < prb_repair)
                    hard_state(i,j) = 0;
                %else hard_state(i,j) = 1;
                end
            end
        end
    end

    for i = 1:4
        for j = 1:27
            if (inter_state(i,j) == 0)
                if (rand(1) < prb_hard)
                    inter_state(i,j) = 1;
                    time_record(1) = time_record(1) + 1;
                %else inter_state(i,j) = 0;
                end
            else
                if (rand(1) < prb_repair)
                    inter_state(i,j) = 0;
                %else inter_state(i,j) = 1;
                end
            end
        end
    end

    for i = 1:11
        if (col_state(i) == 0)
            if (rand(1) < prb_hard)
                col_state(i) = 1;
                time_record(1) = time_record(1) + 1;
            %else col_state(i) = 0;
            end
        else
            if (rand(1) < prb_repair)
                col_state(i) = 0;
            %else col_state(i) = 1;
            end
        end
    end

    fail_flag_old = fail_flag_new;
    srs_fail_flag_old = srs_fail_flag_new;
```

```
        sum_comp = sum(soft_state(:)) + sum(hard_state(:)) +
sum(inter_state(:)) + sum(col_state(:));

        if (sum_comp == 0)
            fail_flag_new = 0;
            time_flag = time_flag + 1;
            time_record(6) = time_record(6) + 1;
        else
            fail_flag_new = 1;
            time_flag = 0;
        end

        for i = 1:4
            if (sum(soft_state(i,:)) + sum(hard_state(i,:)) +
sum(inter_state(i,:)) > 0)
                sys_state(i) = 1;
            else sys_state(i) = 0;
            end
        end

        if (sum(col_state(:))>0)
            sys_state(5) = 1;
        else sys_state(5) = 0;
        end

        sum_state = sys_state(2) + sys_state(3) + sys_state(4);

        if (sys_state(1) || sys_state(5) || sum_state >= 2)
            srs_fail_flag_new = 1;
        else srs_fail_flag_new = 0;
        end

        if (fail_flag_old > fail_flag_new)
            time_record(2) = time_record(2) + 1;
        end

        if (fail_flag_new + fail_flag_old == 0)
            time_record(4) = time_record(4) + 1;
        end

        if (srs_fail_flag_old > srs_fail_flag_new)
            time_record(3) = time_record(3) + 1;
        end
```

```
if (srs_fail_flag_new + srs_fail_flag_old == 0)
   time_record(5) = time_record(5) + 1;
end

if (life == 87600)
   time_record(4) = time_record(4) - time_flag;
   time_record(5) = time_record(5) - time_flag;
end

if (life == 1)
   if (fail_flag_new == 0)
      avail(m,1) = 1;
   end
end

if (life == 2)
   if (fail_flag_new == 0)
      avail(m,2) = 1;
   end
end

if (life == 8)
   if (fail_flag_new == 0)
      avail(m,3) = 1;
   end
end

if (life == 20)
   if (fail_flag_new == 0)
      avail(m,4) = 1;
   end
end

if (life == 100)
   if (fail_flag_new == 0)
      avail(m,5) = 1;
   end
end

if (life == 1000)
   if (fail_flag_new == 0)
      avail(m,6) = 1;
   end
end
```

```
        if (life == 8760)
            if (fail_flag_new == 0)
                avail(m,7) = 1;
            end
        end

        if (life == 43800)
            if (fail_flag_new == 0)
                avail(m,8) = 1;
            end
        end

        if (life == 87600)
            if (fail_flag_new == 0)
                avail(m,9) = 1;
            end
        end

    end
    record(m,1)=time_record(1);
    record(m,2)=time_record(2);
    record(m,3)=time_record(3);
    record(m,4)=time_record(4)/time_record(2);
    record(m,5)=time_record(5)/time_record(3);
    record(m,6)=time_record(6);

    m
    t = toc
end
```

措施二、三代码:

```
life_soft = 5700;%软部件寿命
life_hard = 93000;%硬部件寿命
lamda_soft = 1/life_soft;
lamda_hard = 1/life_hard;
time_repair = 9.5;%修理时间期望
lamda_repair = 1/time_repair;

samples = 100;%仿真次数
record = zeros(samples,6);%数据记录矩阵
avail = zeros(samples,9);

prb_hard = 1 - exp(-lamda_hard);%硬部件状态转移概率
```

```
prb_soft = 1 - exp(-lamda_soft);%软部件状态转移概率
prb_repair = 1 - exp(-lamda_repair);%修理状态转移概率
tic;

for m = 1:samples
    time_record = zeros(1,6);

    soft_state = zeros(5,1);
    hard_state = zeros(5,41);
    inter_state = zeros(5,27);
    col_state = zeros(1,11);

    sys_state = zeros(1,6);
    fail_flag_old = 0;
    fail_flag_new = 0;
    srs_fail_flag_old = 0;
    srs_fail_flag_new = 0;

    time_flag = 0;

    for life = 1:87600
        for i = 1:5
            if (soft_state(i,1) == 0)
                if (rand(1) < prb_soft)
                    soft_state(i,1) = 1;
                    time_record(1) = time_record(1) + 1;
                %else soft_state(i,1) = 0;
                end
            else
                if (rand(1) < prb_repair)
                    soft_state(i,1) = 0;
                %else soft_state(i,1) = 1;
                end
            end
        end

        for i = 1:5
            for j = 1:41
                if (hard_state(i,j) == 0)
                    if (rand(1) < prb_hard)
                        hard_state(i,j) = 1;
                        time_record(1) = time_record(1) + 1;
                    %else hard_state(i,j) = 0;
                    end
                else
                    if (rand(1) < prb_repair)
                        hard_state(i,j) = 0;
                    %else hard_state(i,j) = 1;
                    end
                end
            end
        end

        for i = 1:5
            for j = 1:27
                if (inter_state(i,j) == 0)
                    if (rand(1) < prb_hard)
                        inter_state(i,j) = 1;
                        time_record(1) = time_record(1) + 1;
                    %else inter_state(i,j) = 0;
                    end
                else
                    if (rand(1) < prb_repair)
                        inter_state(i,j) = 0;
                    %else inter_state(i,j) = 1;
                    end
                end
            end
        endd

        for i = 1:11
            if (col_state(i) == 0)
                if (rand(1) < prb_hard)
                    col_state(i) = 1;
                    time_record(1) = time_record(1) + 1;
                %else col_state(i) = 0;
                end
            else
                if (rand(1) < prb_repair)
                    col_state(i) = 0;
                %else col_state(i) = 1;
                end
            end
        end

        fail_flag_old = fail_flag_new;
        srs_fail_flag_old = srs_fail_flag_new;
```

```
    for i = 1:5
      if (sum(soft_state(i,:)) + sum(hard_state(i,:)) +
sum(inter_state(i,:)) > 0)
        sys_state(i) = 1;
      else sys_state(i) = 0;
      end
    end

    if (sum(col_state(:))>0)
      sys_state(6) = 1;
    else sys_state(6) = 0;
    end

    sum_comp1 = sys_state(1) + sys_state(3) + sys_state(4) +
sys_state(5) + sys_state(6);
    sum_comp2 = sys_state(2) + sys_state(3) + sys_state(4) +
sys_state(5) + sys_state(6);

    if (sum_comp1 == 0 && sum_comp2 == 0)
      fail_flag_new = 0;
      time_flag = time_flag + 1;
      time_record(6) = time_record(6) + 1;
    else
      fail_flag_new = 1;
      time_flag = 0;
    end

    sum_state = sys_state(3) + sys_state(4) + sys_state(5);

    if (sys_state(1) + sys_state(2) >= 2 || sys_state(6) ||
sum_state >= 2)
      srs_fail_flag_new = 1;
    else srs_fail_flag_new = 0;
    end

    if (fail_flag_old > fail_flag_new)
      time_record(2) = time_record(2) + 1;
    end

    if (fail_flag_new + fail_flag_old == 0)
      time_record(4) = time_record(4) + 1;
    end

    if (srs_fail_flag_old > srs_fail_flag_new)
    time_record(3) = time_record(3) + 1;
end

if (srs_fail_flag_new + srs_fail_flag_old == 0)
    time_record(5) = time_record(5) + 1;
end

if (life == 87600)
    time_record(4) = time_record(4) - time_flag;
    time_record(5) = time_record(5) - time_flag;
end

if (life == 1)
    if (fail_flag_new == 0)
        avail(m,1) = 1;
    end
end

if (life == 2)
    if (fail_flag_new == 0)
        avail(m,2) = 1;
    end
end

if (life == 8)
    if (fail_flag_new == 0)
        avail(m,3) = 1;
    end
end

if (life == 20)
    if (fail_flag_new == 0)
        avail(m,4) = 1;
    end
end

if (life == 100)
    if (fail_flag_new == 0)
        avail(m,5) = 1;
    end
end

if (life == 1000)
    if (fail_flag_new == 0)
```

```
                avail(m,6) = 1;
            end
        end

        if (life == 8760)
            if (fail_flag_new == 0)
                avail(m,7) = 1;
            end
        end

        if (life == 43800)
            if (fail_flag_new == 0)
                avail(m,8) = 1;
            end
        end

        if (life == 87600)
            if (fail_flag_new == 0)
                avail(m,9) = 1;
            end
        end

    end
    record(m,1)=time_record(1);
    record(m,2)=time_record(2);
    record(m,3)=time_record(3);
    record(m,4)=time_record(4)/time_record(2);
    record(m,5)=time_record(5)/time_record(3);
    record(m,6)=time_record(6);

    m
    t = toc
end
```

措施一、三代码：

```
life_soft = 5700;%软部件寿命
life_hard = 93000;%硬部件寿命
lamda_soft = 1/life_soft;
lamda_hard = 1/life_hard;
time_repair = 9.5;%修理时间期望
lamda_repair = 1/time_repair;
time_restart = 1;
lamda_restart = 1/time_restart;
```

```
samples = 100;%仿真次数
record = zeros(samples,6);%数据记录矩阵
avail = zeros(samples,9);

prb_hard = 1 - exp(-lamda_hard);%硬部件状态转移概率
prb_soft = 1 - exp(-lamda_soft);%软部件状态转移概率
prb_repair = 1 - exp(-lamda_repair);%修理状态转移概率
prb_restart = 1 - exp(-lamda_restart);
tic;

for m = 1:samples
    time_record = zeros(1,6);

    soft_state = zeros(5,1);
    hard_state = zeros(5,41);
    inter_state = zeros(5,19);
    col_state = zeros(1,11);

    sys_state = zeros(1,6);
    fail_flag_old = 0;
    fail_flag_new = 0;
    srs_fail_flag_old = 0;
    srs_fail_flag_new = 0;
    stk_flag = zeros(5,19);

    time_flag = 0;

    for life = 1:87600
        for i = 1:5
            if (soft_state(i,1) == 0)
                if (rand(1) < prb_soft)
                    soft_state(i,1) = 1;
                    %time_record(1) = time_record(1) + 1;
                %else soft_state(i,1) = 0;
                end
            else
                if (rand(1) < 0.982)
                    if (rand(1) < prb_restart)
                        soft_state(i,1) = 0;
                    else soft_state(i,1) = 1;
                    end
                else
                    if (rand(1) < prb_repair)
                        soft_state(i,1) = 0;
                        time_record(1) = time_record(1) + 1;
                    else soft_state(i,1) = 1;
                    end
                end
            end
        end

        for i = 1:5
            for j = 1:41
                if (hard_state(i,j) == 0)
                    if (rand(1) < prb_hard)
                        hard_state(i,j) = 1;
                        time_record(1) = time_record(1) + 1;
                    %else hard_state(i,j) = 0;
                    end
                else
                    if (rand(1) < prb_repair)
                        hard_state(i,j) = 0;
                    %else hard_state(i,j) = 1;
                    end
                end
            end
        end

        for i = 1:5
            for j = 1:19
                if (inter_state(i,j) == 0)
                    if (rand(1) < prb_hard)
                        inter_state(i,j) = 1;
                        time_record(1) = time_record(1) + 1;
                        if (rand(1)<0.35)
                            stk_flag(i,j) = 1;
                        else stk_flag(i,j) = 0;
                        end
                    else
                        %inter_state(i,j) = 0;
                        stk_flag(i,j) = 0;
                    end
                else
                    if (rand(1) < prb_repair)
                        inter_state(i,j) = 0;
                        stk_flag(i,j) = 0;
                    %else inter_state(i,j) = 1;
                    end
                end
              end
            end

            for i = 1:11
              if (col_state(i) == 0)
                if (rand(1) < prb_hard)
                  col_state(i) = 1;
                  time_record(1) = time_record(1) + 1;
                %else col_state(i) = 0;
                end
              else
                if (rand(1) < prb_repair)
                  col_state(i) = 0;
                %else col_state(i) = 1;
                end
              end
            end

            fail_flag_old = fail_flag_new;
            srs_fail_flag_old = srs_fail_flag_new;

            for i = 1:5
              if (sum(soft_state(i,:)) + sum(hard_state(i,:)) +
        sum(inter_state(i,:)) > 0)
                sys_state(i) = 1;
              else sys_state(i) = 0;
              end
            end

            if (sum(col_state(:))>0)
              sys_state(6) = 1;
            else sys_state(6) = 0;
            end

            sum_comp1 = sys_state(1) + sys_state(3) + sys_state(4) +
        sys_state(5) + sys_state(6);
            sum_comp2 = sys_state(2) + sys_state(3) + sys_state(4) +
        sys_state(5) + sys_state(6);

            if (sum_comp1 + sum(stk_flag(2,:)) == 0 && sum_comp2 +
        sum(stk_flag(1,:)) == 0)
              fail_flag_new = 0;
              time_flag = time_flag + 1;
```

```
            time_record(6) = time_record(6) + 1;
        else
            fail_flag_new = 1;
            time_flag = 0;
        end

        sum_state = sys_state(3) + sys_state(4) + sys_state(5);

        if (sys_state(1) + sys_state(2) >= 2 || sys_state(6) ||
sum(stk_flag(:)) > 0 || sum_state >= 2)
            srs_fail_flag_new = 1;
        else srs_fail_flag_new = 0;
        end

        if (fail_flag_old > fail_flag_new)
            time_record(2) = time_record(2) + 1;
        end

        if (fail_flag_new + fail_flag_old == 0)
            time_record(4) = time_record(4) + 1;
        end

        if (srs_fail_flag_old > srs_fail_flag_new)
            time_record(3) = time_record(3) + 1;
        end

        if (srs_fail_flag_new + srs_fail_flag_old == 0)
            time_record(5) = time_record(5) + 1;
        end

        if (life == 87600)
            time_record(4) = time_record(4) - time_flag;
            time_record(5) = time_record(5) - time_flag;
        end

        if (life == 1)
            if (fail_flag_new == 0)
                avail(m,1) = 1;
            end
        end

        if (life == 2)
            if (fail_flag_new == 0)
                avail(m,2) = 1;
```

```
    end
end

if (life == 8)
    if (fail_flag_new == 0)
        avail(m,3) = 1;
    end
end

if (life == 20)
    if (fail_flag_new == 0)
        avail(m,4) = 1;
    end
end

if (life == 100)
    if (fail_flag_new == 0)
        avail(m,5) = 1;
    end
end

if (life == 1000)
    if (fail_flag_new == 0)
        avail(m,6) = 1;
    end
end

if (life == 8760)
    if (fail_flag_new == 0)
        avail(m,7) = 1;
    end
end

if (life == 43800)
    if (fail_flag_new == 0)
        avail(m,8) = 1;
    end
end

if (life == 87600)
    if (fail_flag_new == 0)
        avail(m,9) = 1;
    end
end
```

```
    end
    record(m,1)=time_record(1);
    record(m,2)=time_record(2);
    record(m,3)=time_record(3);
    record(m,4)=time_record(4)/time_record(2);
    record(m,5)=time_record(5)/time_record(3);
    record(m,6)=time_record(6);

    m
    t = toc
end
```

全部措施代码：

```
life_soft = 5700;%软部件寿命
life_hard = 93000;%硬部件寿命
lamda_soft = 1/life_soft;
lamda_hard = 1/life_hard;
time_repair = 9.5;%修理时间期望
lamda_repair = 1/time_repair;
time_restart = 1;
lamda_restart = 1/time_restart;

samples = 100;%仿真次数
record = zeros(samples,6);%数据记录矩阵
avail = zeros(samples,9);

prb_hard = 1 - exp(-lamda_hard);%硬部件状态转移概率
prb_soft = 1 - exp(-lamda_soft);%软部件状态转移概率
prb_repair = 1 - exp(-lamda_repair);%修理状态转移概率
prb_restart = 1 - exp(-lamda_restart);
tic;

for m = 1:samples
    time_record = zeros(1,6);

    soft_state = zeros(5,1);
    hard_state = zeros(5,41);
    inter_state = zeros(5,19);
    col_state = zeros(1,11);

    sys_state = zeros(1,6);
    fail_flag_old = 0;
    fail_flag_new = 0;
```

```
srs_fail_flag_old = 0;
srs_fail_flag_new = 0;

time_flag = 0;

for life = 1:87600
   for i = 1:5
      if (soft_state(i,1) == 0)
         if (rand(1) < prb_soft)
            soft_state(i,1) = 1;
            %time_record(1) = time_record(1) + 1;
         %else soft_state(i,1) = 0;
         end
      else
         if (rand(1) < 0.982)
            if (rand(1) < prb_restart)
               soft_state(i,1) = 0;
            else soft_state(i,1) = 1;
            end
         else
            if (rand(1) < prb_repair)
               soft_state(i,1) = 0;
               time_record(1) = time_record(1) + 1;
            else soft_state(i,1) = 1;
            end
         end
      end
   end

   for i = 1:5
      for j = 1:41
         if (hard_state(i,j) == 0)
            if (rand(1) < prb_hard)
               hard_state(i,j) = 1;
               time_record(1) = time_record(1) + 1;
            %else hard_state(i,j) = 0;
            end
         else
            if (rand(1) < prb_repair)
               hard_state(i,j) = 0;
            %else hard_state(i,j) = 1;
            end
         end
      end
```

```
        end

        for i = 1:5
            for j = 1:27
                if (inter_state(i,j) == 0)
                    if (rand(1) < prb_hard)
                        inter_state(i,j) = 1;
                        time_record(1) = time_record(1) + 1;
                    %else inter_state(i,j) = 0;
                    end
                else
                    if (rand(1) < prb_repair)
                        inter_state(i,j) = 0;
                    %else inter_state(i,j) = 1;
                    end
                end
            end
        end

        for i = 1:11
            if (col_state(i) == 0)
                if (rand(1) < prb_hard)
                    col_state(i) = 1;
                    time_record(1) = time_record(1) + 1;
                %else col_state(i) = 0;
                end
            else
                if (rand(1) < prb_repair)
                    col_state(i) = 0;
                %else col_state(i) = 1;
                end
            end
        end

        fail_flag_old = fail_flag_new;
        srs_fail_flag_old = srs_fail_flag_new;

        for i = 1:5
            if (sum(soft_state(i,:)) + sum(hard_state(i,:)) +
sum(inter_state(i,:)) > 0)
                sys_state(i) = 1;
            else sys_state(i) = 0;
            end
        end
        if (sum(col_state(:))>0)
            sys_state(6) = 1;
        else sys_state(6) = 0;
        end

        sum_comp1 = sys_state(1) + sys_state(3) + sys_state(4) +
sys_state(5) + sys_state(6);
        sum_comp2 = sys_state(2) + sys_state(3) + sys_state(4) +
sys_state(5) + sys_state(6);

        if (sum_comp1 == 0 && sum_comp2 == 0)
            fail_flag_new = 0;
            time_flag = time_flag + 1;
            time_record(6) = time_record(6) + 1;
        else
            fail_flag_new = 1;
            time_flag = 0;
        end

        sum_state = sys_state(3) + sys_state(4) + sys_state(5);

        if (sys_state(1) + sys_state(2) >= 2 || sys_state(6) ||
sum_state >= 2)
            srs_fail_flag_new = 1;
        else srs_fail_flag_new = 0;
        end

        if (fail_flag_old > fail_flag_new)
            time_record(2) = time_record(2) + 1;
        end

        if (fail_flag_new + fail_flag_old == 0)
            time_record(4) = time_record(4) + 1;
        end

        if (srs_fail_flag_old > srs_fail_flag_new)
            time_record(3) = time_record(3) + 1;
        end

        if (srs_fail_flag_new + srs_fail_flag_old == 0)
            time_record(5) = time_record(5) + 1;
        end
```

```
if (life == 87600)
    time_record(4) = time_record(4) - time_flag;
    time_record(5) = time_record(5) - time_flag;
end

if (life == 1)
    if (fail_flag_new == 0)
        avail(m,1) = 1;
    end
end

if (life == 2)
    if (fail_flag_new == 0)
        avail(m,2) = 1;
    end
end

if (life == 8)
    if (fail_flag_new == 0)
        avail(m,3) = 1;
    end
end

if (life == 20)
    if (fail_flag_new == 0)
        avail(m,4) = 1;
    end
end

if (life == 100)
    if (fail_flag_new == 0)
        avail(m,5) = 1;
    end
end

if (life == 1000)
    if (fail_flag_new == 0)
        avail(m,6) = 1;
    end
end

if (life == 8760)
    if (fail_flag_new == 0)
        avail(m,7) = 1;
```

```
            end
        end

        if (life == 43800)
            if (fail_flag_new == 0)
                avail(m,8) = 1;
            end
        end

        if (life == 87600)
            if (fail_flag_new == 0)
                avail(m,9) = 1;
            end
        end

    end
    record(m,1)=time_record(1);
    record(m,2)=time_record(2);
    record(m,3)=time_record(3);
    record(m,4)=time_record(4)/time_record(2);
    record(m,5)=time_record(5)/time_record(3);
    record(m,6)=time_record(6);

    m
    t = toc
end

%dlmwrite('data0.txt',record);
```