

Reliability Analysis for AFTI-F16 SRFCS Using ASSIST and SURE¹

N. Eva Wu

Department of Electrical Engineering, Binghamton University, Binghamton, NY 13902-6000
Tel: 607-777-4375, Fax: 607-777-4464, Email: evawu@binghamton.edu

Abstract

This paper reports the results of a study on reliability analysis of an AFTI-16 Self-Repairing Flight Control System (SRFCS) using software tools SURE (Semi-Markov Unreliability Range Evaluator) and ASSIST (Abstract Semi-Markov Specification Interface to the SURE Tool). The purpose of the study is to investigate the potential utility of the software tools in the ongoing effort of the NASA Aviation Safety Program, where the class of systems must be extended beyond the originally intended serving class of electronic digital processors. The study concludes that SURE and ASSIST are applicable to reliability analysis of flight control systems. They are especially efficient for sensitivity analysis that quantifies the dependence of system reliability on model parameters. The study also confirms an earlier finding on the dominant role of a parameter called a failure coverage. The paper will remark on issues related to the improvement of coverage and the optimization of redundancy level.

1 Introduction

In 1997, the Clinton administration issued a national goal to reduce the aircraft fatal accident rate by 80% in 10 years. In response to this challenge, NASA established the Aviation Safety Program. The single aircraft accident prevention project is one of the six project areas under this program. Technologies being developed under the single aircraft accident prevention project include vehicle health management technologies, control upset prevention & recovery technologies, and system validation & verification (V&V) technologies. This paper addresses the reliability analysis aspect of the validation & verification technologies. Other aspects of the technologies include simulation-based and experimental V&V methods. Methods for system validation and verification are essential to commercialization of new safety enhancement technologies due to certifica-

tion requirements. The reader is referred to Belcastro and Belcastro¹ for an overview of the single aircraft accident prevention project and relevant technologies.

In addition to providing tools for system validation and verification, reliability analysis also helps identify needs and specify goals in aviation safety. It serves to offer guidelines for integrated system design as well.

SURE³ was developed at NASA Langley in response to growing size and complexity of fault-tolerant digital systems and the resulting intractable reliability analysis. The original version of SURE runs on VMS and UNIX operating systems. A more recent update allows the program to run on Windows. In principle, the reliability of any fault tolerant system with a failure process described by a Markov model⁵ can be computed using SURE. However, the process of delineating all of the states and transitions can be devastatingly tedious and error-prone. ASSIST² was then developed as an abstract model definition language which specifies a set of rules for generating large Markov models automatically with a small number of statements. This tool has also been modified recently to run on Windows.

SURE is based on a mathematical theorem developed by White⁷ for computing the reliability of a fault tolerant system. Two characteristics of a fault tolerant system have made the task of reliability assessment difficult³. They are that the use of sophisticated reconfiguration strategies has resulted in complex models, and that system recovery is many orders of magnitude faster than the fault arrival process, which can cause rapid growth in the error terms in numerical integration errors. The mathematical theorem for SURE overcomes both difficulties. This makes it a strong candidate for being used as a tool for the aviation safety program where the required system failure probability is in the order of 10^{-9} .

A key to enable the application of SURE to a fault tolerant control system is the coverage modeling⁹. A formal definition of coverage was introduced in [4] for use as a parameter to reflect the ability of digital processors to automatically recover from the occurrence of a

¹This work was supported by NASA under Cooperative Agreement # NCC-1-336. The author would like to thank Ricky Butler and Allan White of NASA Langley for sharing their expertise in reliability.

coverage \equiv *probability(system recovers|fault occurs)*.

Schemes used for managing the analytic redundancy in a complex control system involve considerable more risks than schemes used for managing the direct redundancy such as majority voting. This is because decision making is often based on residual signals formed by the differences between noisy measurements and calculated values of output variables based on inaccurate models. Decision errors can be associated with uncertainties on whether there is a subsystem failure, which subsystem has failed, how severe is its effect, whether it is necessary to take a corrective action, which actions to take. In addition, the question on whether there is adequate control relevant redundancy and authority to allow recovery from the effect the failure becomes more difficult to answer. The dynamic and closed-loop nature, common to all control systems, is the source for additional difficulties, such as temporary mask of the effect of subsystem failures, the vagueness in the definition of a system level failure in the context of control performance, and the sometimes significant processing requirement in supporting the redundancy management. Coverage in this context has been shown to be highly scenario dependent, highly time dependent, and difficult to model^{9,10}

The paper is organized as follows. The reliability model of a self-repairing flight control system for an AFTI-16 aircraft is described in Section 2. Results of reliability and sensitivity analysis are presented in Section 3. Conclusions drawn from the analysis are summarized in Section 4. Section 5 lists the references. Section 6 provides a sample program using in ASSIST for Markov model building.

The upper block diagram in Fig.1 shows the dependency of functional modules in a AFTI-16 SRFCS. The

Figure 1 illustrates the functional architecture of the flight control system. The system is composed of several key components and their interconnections:

- Flight critical processors**: The initial processing stage.
- I/O control modules**: Modules for input/output control.
- Pilot command sensors**: Sensors for pilot commands.
- Aircraft state sensors**: Sensors for aircraft state.
- Pitch and roll effectors**: Effectors for pitch and roll control.
- yaw effectors**: Effectors for yaw control.

The **Pitch and roll effectors** are connected to a **Dual tandem power actuator**, which in turn connects to a **Computer-effector interface**. This interface is linked to a **micro-controller** and a **Surface**. The **micro-controller** manages the **Left Slab**, **Right Slab**, **Left Canard**, and **Right Canard**. The **Surface** manages the **Left Flap**, **Right Flap**, **Left Rudder**, and **Right Rudder**.

The reliability indicator used in the following discussion is the probability of loss of control denoted by P_{LOC} . P_{LOC} estimates the system compliance with applicable safety-of-flight criterion and provides an indication of the impact of added or reduced hardware redundancy as well as the flight control system reconfiguration capability. Each small box in Fig.1 represents a subsystem. The symbols λ_i ($i = 1, 2, 3$), shown in the small boxes are the subsystem failure rates in terms of failures per hour. Under the assumption of low subsystem failure rates and short mission time, constant failure rates (exponential distribution) are appropriate. The safety requirement for the inner layer parallel configuration (the n-plex computer/effector interface subsystem) is 1-out-of- n (fail-operational/fail-operational/loss-of-control, for example, in the 3-plex

case). The safety requirement for the outer layer parallel configuration in the pitch-roll effector channels is 3-out-of-4 (fail-operational/loss-of-control). This means that the three remaining effector channels in this block must work in concert to accommodate a failure in one effector channel. The safety requirement for the outer layer parallel configuration in the yaw effector channels is 1-out-of-2 (fail-operational/loss-of-control).

The redundancy architecture shown in Fig.1 does not truly reflect how effector channel hardware is configured. It must be understood as an effective redundancy configuration which assumes that any anomaly in an effector channel serious enough to warrant a control adaptation or reconfiguration action for failure accommodation can do so promptly and successfully. In reality, however, due to uncertainties in the model of the system to be controlled, uncertainties in the models of signals exerted on the system, and the limited processing capability, considerable risks exist in making a decision on the corrective action. These decision risks must be taken into consideration in reliability assessment. The risks encountered may include overly slow or severe transients, false alarm, missed detection, false identification, false reconfiguration, and lack or exhaustion of redundancy. The notion of coverage is now used to account for such risks. It represents an attempt to separate the handling of failures from the occurrence of failures. Once a decision is made however, the process of removing a subsystem or reconfiguring the system is generally involved. This process, though fast in comparison with a failure process, still takes time, and has been shown to be generally non-exponentially distributed³. Including this process in a reliability model implies the creation of a numerically stiff problem. SURE is designed to specifically solve the stiff problem.

We now present the data used for the reliability analysis. Table 1 gives the values of coverage, and Table 2 gives the values of failure rates, recovery rates and variances, and mission time.

| | |
|------------------------------|-------------------------|
| Redundancy management | Coverage |
| Voting (4-plex C/E) | $c_{1v} = 0.99992$ |
| Voting (3-plex C/E) | $c_{1v} = 0.9999$ |
| Comparing (2-plex C/E) | $c_{1c} = 0.99$ |
| Self-monitoring (1-plex C/E) | $c_{1m} = 0.89$ |
| Actuator self-monitoring | $c_2 = 0.99$ |
| Surface self-monitoring | $c_3 = 0.89 \sim 0.995$ |

Table 1

| | | |
|-----------------------------|------------------------------------|--------------------|
| C/E subsystem failure rate | $\lambda_1 = 70 \times 10^{-6}$ | hour ⁻¹ |
| Actuator failure rate | $\lambda_2 = 0.5 \times 10^{-6}$ | hour ⁻¹ |
| Surface failure rate | $\lambda_3 = 10^{-6} \sim 10^{-4}$ | hour ⁻¹ |
| Mean time to recover | $\mu_j = 0 \sim 10^{-4}$ | hours |
| Variance of time to recover | $\sigma_j = 0 \sim 10^{-4}$ | hours |
| Mission time | $T = 10^0 \sim 10^1$ | hours |

Table 2

The above table reflects the two common characteristics of highly reliable fault tolerant systems: details due to small failure probabilities cannot be arbitrarily ignored, and recovery process is much faster than failure process.

3 Reliability and sensitivity analysis

This section performs reliability analysis for the AFTI-16 SRFCs using ASSIST and SURE. Since SURE allows a parameter of the model to vary over a range of values, it therefore easily provides results of sensitivity with respect to the parameter. The largest data uncertainties are the mission time, the recovery time distribution, the failure rate and the coverage values associated with the control surfaces. These will be selected as variables or parameters in the sensitivity analysis. Also of interest is the level of redundancy in the computer/effector interface which would impact hardware addition/reduction involving servo electronics, DCM coils, LVDTs, power supplies, etc.

One of the most important original goals⁸ of reliability analysis for AFTI-16 SRFCs is to determine whether a probability of loss of control at 10^{-7} can be achieved with the use of aerodynamically redundant surfaces without added hardware. The main goal here however is focused on applicability of SURE and ASSIST to systems that are targeted for enhanced reliability in the aviation safety program. Therefore, no special effort will be made to justify the numerical numbers of the data based on which our computation is carried out, though the numbers used are reasonably close to the values in practice.

The major assumptions used in the following failure probability computation are as follows:

- the failure probability of any given subsystem is $1 - e^{-\lambda t}$ where λ is the constant failure rate of that subsystem;
- a failure in any subsystem is independent of that in all other subsystems;
- redundancy management restores the system operation with a certain coverage following a subsystem failure (caused by decision errors, delays in redundancy management, and the exhaustion of redundancy as a special case);
- an uncovered subsystem failure leads to the system failure;
- a covered subsystem failure obeys a recovery time distribution with mean time μ and variance σ^2 (caused by transients following a restructure, such as the removal of a failed component, or the reconfiguration of a control law);
- all rates of recovery are orders of magnitude faster than rates of subsystem failures.

The rationales for the assumptions are now given.

Assumption (a) is appropriate for highly reliable subsystems performing short missions based on the argument that a process with a nonconstant failure rate can be approximated by a process with a piecewise constant rate to desired accuracy in any given finite interval³. A reliability assessment task becomes more tractable with a constant rate Markov model.

Assumption (b) is made based on the fact that a reliability model that contains common-mode failures can be, in general, recasted with altered redundancy configuration and altered reliability requirements into one containing only independent subsystems.

Assumption (c) concerns the use of coverage. Since redundancy management decisions are made based on processed measurements, coverage is usually an increasing function of time, and therefore a dynamic parameter¹⁰. Coverage used for reliability assessment is the value at the onset of a corrective action that follows a redundancy management decision. This value is called a static coverage value. Its effect on the overall system reliability can be examined through numerical means by varying the static coverage value. The case of exhaustion of redundancy can be regarded as a special case of zero coverage. Assumption (d) below is where this case is reflected.

Corresponding to each failure scenario, there is a critical clearance time t_c at which a corrective action must be taken. Critical clearance time is described in Khalil⁶ for the scenario of a short circuit in a nonlinear electrical network to introduce the concept of region of attraction. The scenario is retold here to rationalize the need to separately list assumptions (d) and (e). Suppose the short circuit has caused a subsystem failure that resulted in a departure of the network states' trajectory from the pre-failure asymptotically stable equilibrium. Suppose the system is recoverable through control reconfiguration and based on the post-failure dynamics a new equilibrium can be established. Then critical clearance time is the maximum period allowed between the occurrence of the failure and the establishment of the new equilibrium, during which the departing trajectory from the old equilibrium still stays within the region of attraction of the new equilibrium. In this case, it is a matter of time the new equilibrium will be reached. Assumption (d) above addresses failures as consequences of a prolonged control reconfiguration decision beyond the critical clearance time, or the establishment of an incorrect new equilibrium that is out of reach of the current state. While assumption (e) addresses the recovery process where the trajectory is already within the new region of attraction but is still racing with other failure processes to reach its final destination—the new equilibrium.

There are two commonly shared features of highly reliable systems. One is the use of complex redundancy management strategies, especially when analytic redundancy is involved. The second feature is stated in

assumption (f), i.e., average rate of system recovery through redundancy management is nonzero but many orders of magnitude faster than the average rate of fault arrival. Because of the much faster recovery rate, constant rate assumption as given in (a) is no longer appropriate. Theory for accurate reliability prediction has been developed⁷ to address the issue that uses only the means and the variances of recovery times, which can be acquired empirically.

The SRFCS under consideration has six cascaded blocks. If the i th block has a failure probability $P_{LOC}^i(t)$, the composite failure probability is given by $1 - \prod_{i=1}^6 \{1 - P_{LOC}^i(t)\}$. Since the first four blocks are simple quadruplex blocks for which no major uncertainties are present and no further configuration changes are being considered, only the result of composite failure probability of the two effector blocks is presented.

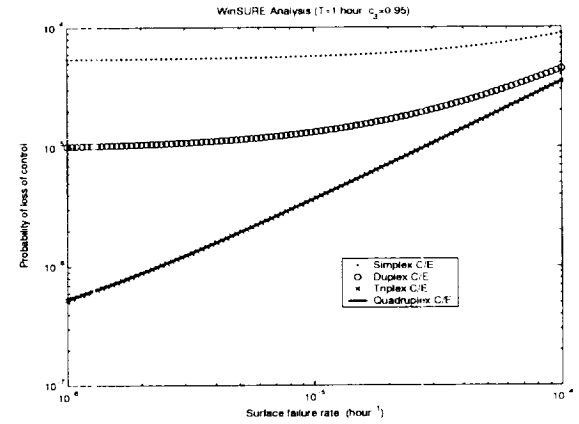


Fig.2 AFTI-16 SRFCS P_{LOC} v.s. λ_3 with varying C/E redundancy level

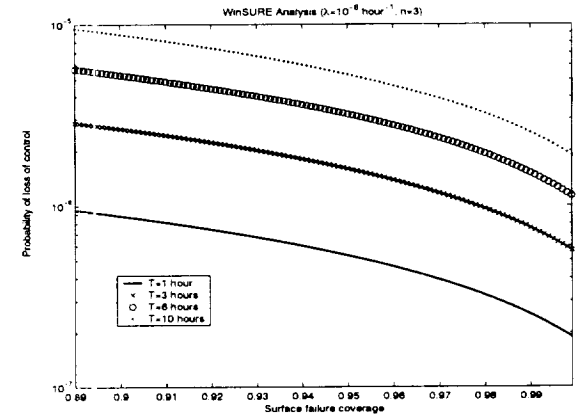


Fig.3 AFTI-16 SRFCS P_{LOC} v.s. c_3 with varying mission time

The two figures above depict the failure probability of the effector blocks as well as its sensitivity with respect to variations of four parameters, which are mis-

sion time, surface damage rate, surface damage coverage, and redundancy level of the computer/effector interface subsystem. Most observations from these two figures are consistent with our intuitions. There are, however, exceptions. The most notable from Fig.2 is that the P_{LOC} corresponding to a quadruplex C/E configuration is not any better than that corresponding to a triplex configuration. This seemingly elusive property can be shown to attribute to the imperfection of coverage (< 1). Moreover, the effect of variation of recovery rates is not observable in the computation for the prescribed range, which, together with the elusive property regarding the redundancy level, have been recently confirmed by theory⁹. Also from the vastly different dynamic range between Fig.3 and Fig.2, it can be observed that P_{LOC} is much more sensitive to coverage than to other variables. In particular, improvement in coverage, even by a small percentage, say 1% (from .99 to .999), could reduce the system failure probability by an order of magnitude.

The ASSIST program used to generate the SURE program (Markov model) for the failure probability calculation in the yaw-axis block is given in the Appendix. This sample program ignores the recovery time. The generated SURE program contains many thousands of states and therefore is too large to be included in the paper.

4 Summary of results

ASSIST and SURE are used to evaluate the probability of loss of control of a AFTI-16 SRFCS, as well as the sensitivity of the probability with respect to some of the model parameters. For the prescribed data range, the following observations have been made.

1. P_{LOC} increases approximately linearly as mission time increases.
2. P_{LOC} increases approximately linearly as surface failure rate increases. This calls for more reliable subsystems.
3. P_{LOC} increases approximately linearly as the complementary coverage ($1 - c$) of a surface damage increases. Since the value of coverage is generally very close to one, the effect of coverage on P_{LOC} is significant. This calls for a focused effort on the development more optimized decision and control algorithms for better redundancy management.
4. P_{LOC} is minimized at an appropriate redundancy level, more specifically in this case study, at the triplex level. This is due to the fact that coverage of failures is not perfect.
5. P_{LOC} is unaffected by the recovery rate for the range specified in Table 2. Therefore, the mean

recovery time can be set to zero in the probability evaluation. This results in a significant simplification of the problem—from a semi-Markov model to a homogeneous Markov model.

It can be shown analytically that the approximate linear relationship between P_{LOC} and the parameters, as well as the independence of P_{LOC} from the recovery rates begin to falter when $1 - c \gg \lambda T$ no longer holds⁹. On the other hand, as long as $1 - c \gg \lambda T$ holds, one can obtain very accurate system failure probability estimation using some approximations that reveal in analytic forms the above observations.

In summary, SURE is capable of handling complex re-configuration strategies with simple reliability models. It provides sufficient accuracy for disparate failure and recovery processes encountered in fault tolerant control systems in aviation. It has the flexibility to allow incorporation of decision risk factors. It therefore is a suitable reliability analysis tool for the aviation safety program. However, the use of ASSIST and SURE requires a thorough understanding of failure and recovery processes, and therefore adequate background in reliability theory and Markov process is needed for the potential user. This is when accurate and efficient tools such as SURE become absolutely indispensable.

The major challenge in reliability analysis of flight critical systems lies with the fact that test data crucial to reliability study but sensitive from market-competition and liability viewpoints are difficult to obtain, while rare accident data alone are not statistically significant. It is important to understand that most data required by SURE are obtained through statistical means. Therefore, our reliability estimates can only infer from failure data to the general population and say very little concerning an individual system. A reliability model so developed displays a pattern only over a large number of failures. Nevertheless, such reliability analysis provides very important information on whether subsystems are well designed and properly configured into an overall system.

5 References

1. Belcastro, C., and Belcastro, C., Application of failure detection, identification, and accommodation methods for improved aircraft safety, *Proc. American Control Conference*, 2001.
2. R.W.Butler, An abstract language for specifying Markov reliability models, *IEEE Trans. on Reliability*, vol.R-35, pp 595-601, 1986.
3. R.W.Butler, The SURE approach to reliability analysis, *IEEE Trans. Reliability*, vol.41, pp 210-218, 1992.
4. Dugan, and Trivedi, Coverage modeling for dependability analysis of fault tolerant systems.

IEEE Trans. Computers, vol.38, pp 775-787, 1989.

5. Howard, R.A., *Dynamic probabilistic systems: V1, Markov Models, V2, Semi-Markov and Decision Processes*, Wiley, 1971.
6. Khalil, H., *Nonlinear Systems*, 2nd Edition, Prentice Hall, 1996.
7. White, A.L., Reliability estimation for reconfigurable systems with fast recovery, *Microelectronics Reliability*, vol.26, pp 1111-1120, 1986.
8. Wu, N. E., and T.J. Chen, Reliability prediction for self-repairing flight control systems. *Proc. 35th IEEE Conference on Decision and Control*, 1996.
9. Wu, N. E., Reliability of fault tolerant control systems: Part I, *Proc. 40th IEEE Conference on Decision and Control*, 2001.
10. Wu, N. E., Reliability of fault tolerant control systems: Part II, *Proc. 40th IEEE Conference on Decision and Control*, 2001.

6 Appendix

A sample ASSIST program is given here, which is used to build the Markov model for failure probability calculation in the yaw block.

```
(* Markov model generation for AFTI-16 SRFCS yaw axes*)
(* Failure rates and coverages *)
LI=7.0E-5; (* subsystem failure rate for EO interface block (3-plex)*)
LA=0.5E-6; (* subsystem failure rate for actuator block (1-plex block) *)
LS=1E-6; (* subsystem failure rate for surface block (1-plex block)*)
CI01=0.99992; (* coverage for the 1st failure in block I *)
CI12=0.9999; (* coverage for the 2nd failure in block I *)
CI23=0.99; (* coverage for the 3rd failure in block I *)
CI34=0.89; (* coverage for the 4th failure in block I *)
CA01=0.99; (* coverage for the failure in block A *)

(* Input to SURE for coverage variation *)
"DELTA = 0.0 TO+ 1.0;" (* Delta times the coverage range = step size *)
"POINTS = 11;"
"CS01=0.89+DELTA*(0.995-0.89);" (* CB01 ranges from 0.99 to 1.0 *)

(* State space definition. (Array of four identical channels)*)
SPACE=(NWI: ARRAY[1..3] OF 0..3, (* NWI: Number of operative subsystems in block I *)
NFI: ARRAY[1..3] OF 0..3, (* NFI: Number of inoperative subsystems in block I *)
NUI: ARRAY[1..3] OF 0..1, (* NUI: Flag uncovered failures in block A when NUI=1 *)
NWA: ARRAY[1..3] OF 0..1, (* NWA: Number of operative subsystems in block A *)
NFA: ARRAY[1..3] OF 0..1, (* NFA: Number of inoperative subsystems in block A *)
NWS: ARRAY[1..3] OF 0..1, (* NWS: Number of operative subsystems in block S *)
NFS: ARRAY[1..3] OF 0..1); (* NFS: Number of inoperative subsystems in block S *)

(* Initial state definition *)
START = (3 OF 3, 3 OF 0, 3 OF 0, 3 OF 1, 3 OF 0, 3 OF 1, 3 OF 0);
(* NWI[I]=3, NFI[I]=0, NUI[I]=0, NWA[I]=1, NFA[I]=0, NWS[I]=1, NFS[I]=0, I=1,2,3 *)
```

```
(* Death state definition by defining minimum cut sets *)
DEATHIF (NUI[1]+NUI[2]+NUI[3]>=1) (* any uncovered failures *)
OR (NFI[1]+NFI[3]>5)
OR (NFI[2]+NFI[3]>5)
OR (NFA[1]+NFI[3]>3)
OR (NFA[2]+NFI[3]>3)
OR (NFS[1]+NFI[3]>3)
OR (NFS[2]+NFI[3]>3);
DEATHIF (NFI[1]+NFS[3]>3)
OR (NFI[1]+NFA[3]>3)
OR (NFI[2]+NFS[3]>3)
OR (NFI[2]+NFA[3]>3)
OR (NFS[1]+NFA[3]>1)
OR (NFA[1]+NFA[3]>1);
DEATHIF (NFA[1]+NFS[3]>1)
OR (NFS[2]+NFA[3]>1)
OR (NFA[2]+NFA[3]>1)
OR (NFA[2]+NFS[3]>1)
OR (NFS[1]+NFS[3]>1)
OR (NFS[2]+NFS[3]>1);

(* State transitions in channel I, I=1,2,3 *)
FOR I IN [1..3];
IF (NFA[I]=0) AND (NFS[I]=0) AND (NFI[I]=0) THEN (* 1st failure in block I *)
  TRANS TO NWI[I]=NWI[I]-1 , NFI[I]=NFI[I]+1 , NUI[I]=0 BY NWI[I]*LI*CI01; (* covered *)
  TRANS TO NWI[I]=NWI[I]-1 , NFI[I]=NFI[I]+1 , NUI[I]=1 BY NWI[I]*LI*(1-CI01); (* uncovered *)
ENDIF;
IF (NFA[I]=0) AND (NFS[I]=0) AND (NFI[I]=1) THEN (* 2nd failure in block I *)
  TRANS TO NWI[I]=NWI[I]-1 , NFI[I]=NFI[I]+1 , NUI[I]=0 BY NWI[I]*LI*CI12; (* covered *)
  TRANS TO NWI[I]=NWI[I]-1 , NFI[I]=NFI[I]+1 , NUI[I]=1 BY NWI[I]*LI*(1-CI12); (* uncovered *)
ENDIF;
IF (NFA[I]=0) AND (NFS[I]=0) AND (NFI[I]=2) THEN (* 3rd failure in block A *)
  TRANS TO NWI[I]=NWI[I]-1 , NFI[I]=NFI[I]+1 , NUI[I]=0 BY NWI[I]*LI*CI23; (* covered *)
  TRANS TO NWI[I]=NWI[I]-1 , NFI[I]=NFI[I]+1 , NUI[I]=1 BY NWI[I]*LI*(1-CI23); (* uncovered *)
ENDIF;
IF (NWA[I]=1) AND (NWS[I]>0) AND (NWI[I]>0) THEN (* Failure in block A *)
  TRANS TO NUI[I]=0, NWA[I]=NWA[I]-1 , NFA[I]=NFA[I]+1 BY NWA[I]*LA*CA01; (* covered *)
  TRANS TO NUI[I]=1, NWA[I]=NWA[I]-1 , NFA[I]=NFA[I]+1 BY NWA[I]*LA*(1-CA01); (* uncovered *)
ENDIF;
IF (NWA[I]>0) AND (NWS[I]=1) AND (NWI[I]>0) THEN (* Failure in block S *)
  TRANS TO NUI[I]=0, NWS[I]=NWS[I]-1 , NFS[I]=NFS[I]+1 BY NWS[I]*LS*CS01; (* covered *)
  TRANS TO NUI[I]=1, NWS[I]=NWS[I]-1 , NFS[I]=NFS[I]+1 BY NWS[I]*LS*(1-CS01); (* uncovered *)
ENDIF;
ENDFOR;
```