TECHNICAL UNIVERSITY KOŠICE
Faculty of Electrotechnical Engineering and Informatics
Department of Cybernetics and Artificial Intelligence

Ing. Ján GALDUN

Dependability Analysis of Networked Control Systems with Consideration of Shared Redundant Subsystems

Study program: Cybernetics
Study branch: 9.2.7 Cybernetics
Tutors: Prof. Jean-Marc THIRIET, Eng., Dr. Hab.,
(Université Joseph Fourier Grenoble, GIPSA-Lab)
Prof. Ing. Ján SARNOVSKÝ, CSc.,
Ing. Ján LIGUŠ, Phd.

Grenoble, 1.8.2008

Doctoral Thesis prepared in partnership (Eiffel Doctorate Grant) with GIPSA-Lab Grenoble (UMR 5216 CNRS-INPG-UJF-US)

Acronyms **4**

1. Networked Control Systems and reliability analysis **5**

1.1 Distributed control systems 5

1.1.1 Networked control systems 6
1.1.2 Communication networks in NCS 7
1.1.2.1 Networks with CSMA/CR (CSMA/AMP) access method 9
1.1.2.2 Networks with TDMA access method 10
1.1.3 NCS design 11
1.2 Dependability of the systems 14

1.2.1 Attributes and parameters associated to the dependability 16
1.2.2 SIL levels 17
1.2.3 Mathematical description 19
1.3 Different methods of dependability analysis 21

1.3.1 Markov models 22
1.3.2 Fault trees and reliability block diagrams 24
1.3.3 Bayesian networks 26
1.3.4 Petri nets and Monte Carlo simulation 27
1.3.5 Comparing the features of the reliability analysis methods 31
1.4 Conclusion 33

2. The aims of the doctoral thesis **35**

3. Modeling the system with various redundant strategies **36**

3.1 Common redundant subsystems 37

3.1.1 Passive redundant components 37
3.1.2 Active redundant components 38
3.1.3 Discussion on passive and active redundancies 39
3.2 Shared redundancy – quasi-redundant subsystems 41

3.2.1 An introduction to shared redundancy 41
3.2.1.1 Quasi-redundant controllers 44
3.2.1.2 Quasi-redundant networks 46
3.2.2 Description of specific dependencies given by SR application 47
3.2.3 Advantages of using shared redundancy approach 51
3.3 Modeling the QR structures 52

3.3.1 Coloured Petri Nets 52
3.3.2 Modeling in CPN Tools 53
3.3.2.1 Place 55
3.3.2.2 Transition 56
3.3.2.3 Arcs 57
3.3.2.4 Description of the several important functions of the prepared models 58
3.4 Conclusion 59

4. Case study of the shared redundant subsystems with consideration of different conditions for load sharing **61**

4.1 Models and Monte Carlo Simulation results verification 62
4.2 Two quasi-redundant components 65
4.2.1 Simulation results for a single step change of the failure rate 69
4.2.2 Linearly changed failure rate in 1-out-of-2 subsystems 71
4.2.3 Simulation results for exponentially changed failure rate 75
4.3 Triple Modular Redundancy with load sharing consideration 77
4.3.1 Single change of the component's failure rate 80
4.3.2 Simulation results for linearly changed failure rate 81
4.3.3 Simulation results for linearly changed of the failure rate each *k*-th time unit 83
4.3.4 Simulation results for exponentially changed failure rate 85
4.4 Conclusion 87
5. Reliability and MTTF parameter approximation for NCS with shared redundant parts 89
5.1 Reliability approximation 90
5.1.1 Reliability approximation of the two-component QR subsystem with dynamic FR behaviors 90
5.1.1.1 Single step failure rate change 91
5.1.1.2 Continuous linear failure rate change 92
5.1.1.3 Exponential failure rate change 92
5.1.2 Reliability approximation of the three-component QR subsystem with dynamic FR behaviors 93
5.2 MTTF parameter approximation 95
5.2.1 Two components QR subsystems 96
5.2.2 Three components QR subsystems 99
5.3 Study of the controlled subsystem loading with direct consequence to the reliability and life time 101
5.3.1 Scenarios simulation 103
5.3.2 Scenarios evaluation 104
5.3.3 Reconfiguration strategy 111
5.4 Conclusion 114
6. Application of the proposed method for dependability evaluation of the more complex systems 116
6.1.1 Reliability approximation of the cascade application with consideration of the SR subsystems 116
6.2 Mini-drone application 120
6.2.1 Structure introduction 120
6.2.2 Structure evaluation 123
6.3 Conclusion 128
7. Conclusion 130
A. Appendices 133

A.1 Several important distribution functions in reliability theory 133
A.1.1 Uniform exponential distribution 133
A.1.2 Exponential distribution 133
A.2 Description of the function used in the prepared Petri nets models in CPN Tools 134
A.2.1 File functions 134
A.2.2 Random distribution functions and type convert functions 135
List of figures 137
List of tables 140
References 143

Acronyms

BN	Bayesian Networks
BDMP	Boolean logic Driven Markov Process
CAN	Control Area Network
CPN	Coloured Petri Nets
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
CSMA/CR	Carrier Sense Multiple Access / Collision Resolution
CSMA/AMP	Carrier Sense Multiple Access / Arbitration by Message Priority
DCS	Distributed Control System
DFT	Dynamic Fault Trees
FDEP	Functional Dependency Gate
FDFH	Frequency of Dangerous Failure per Hour
FR	Failure rate
FT	Fault Trees
FTA	Fault Trees Analysis
eFT	Extended Fault Trees
GPN	Generalized Petri Nets
GSPN	Generalized Stochastic Petri Nets
IEC	International Electrotechnical Commission
MTBH	Mean Time Between Hazard
MTTF	Mean Time To Failure
MTTR	Mean Time To Repair
NDBA	Non Destructive Bitwise Arbitration
NCCS	Networked Cascade Control System
NCS	Networked Control System
PAND	Priority AND
PFD	Probability of Failure on Demand
PN	Petri Nets
QoC	Quality of Control
QR	Quasi-Redundant / Quasi-Redundancy
RBD	Reliability Block Diagrams
SEQ	Sequence Enforcing Gate
SFT	Static Fault Trees
SIL	Safety Integrity Level
SIF	Safety Integrity Function
SPN	Stochastic Petri Nets
SR	Shared Redundancy
TDMA	Time Division Multiple Access
TE	Top Event
WSP	Warm Spare Gate

1. Networked Control Systems and reliability analysis

1.1 Distributed control systems

The fast development of new technologies opens new opportunities for computing decentralization, which is the main idea in realization and practical application of distributed systems. This kind of system is represented by a decentralized computing structure, where an executed task is divided and executed by several specified subsystems.

In general, a distributed system is composed of several autonomous computing parts [Bit 06] (computers) interconnected by network and distributed middleware. The middleware allows coordinate executed task as well as sharing system's resources by each node of this system. For example, a distributed system could be represented as a group of computers connected to a network whereas each node is occupied by its own subtask which is part of a global task executed by the whole group. The similar approach for task executing is used by parallel distributed systems. Chen and Taylor in [Che 98] present a distributed system as a system which covers a platform with a network of resources as a cluster of workstations, cluster of personal computers or parallel machines.

Some approaches for tasks solving have been applied also on the field of automation control. Hence, modern control systems can be branded as distributed control systems (Fig. 1.1). In distributed control systems (DCS) [Zon 07], a common computing task is realized by controllers in predefined cooperation with sensors and actuators. Information exchanges among all cooperated autonomous subsystems are assigned to the communication network to which they are connected.

A DCS is defined [Pim 02] as a system which includes the set of control systems implemented in distributed fashion, whereas communication among each control subsystems is provided thanks to suitable communication protocol.

Networked control systems are examples of distributed control system which are further discussed.

1.1.1 Networked control systems

The definition of networked control systems presented in [Lia 01b] is formulated as follows:

„A networked control system is one type of distributed control system where sensors, actuators and controllers are interconnected by communication networks.“

As we can see from the definition of networked control system (NCS) (Fig. 1.1), the network is a very important part of the control structure. Thus, through the network is realized the exchange of all the required control data needed to hold the system's states between the required limits. The various control structure approaches as well as distributed control systems (or NCS) brings advantages but also some disadvantages.

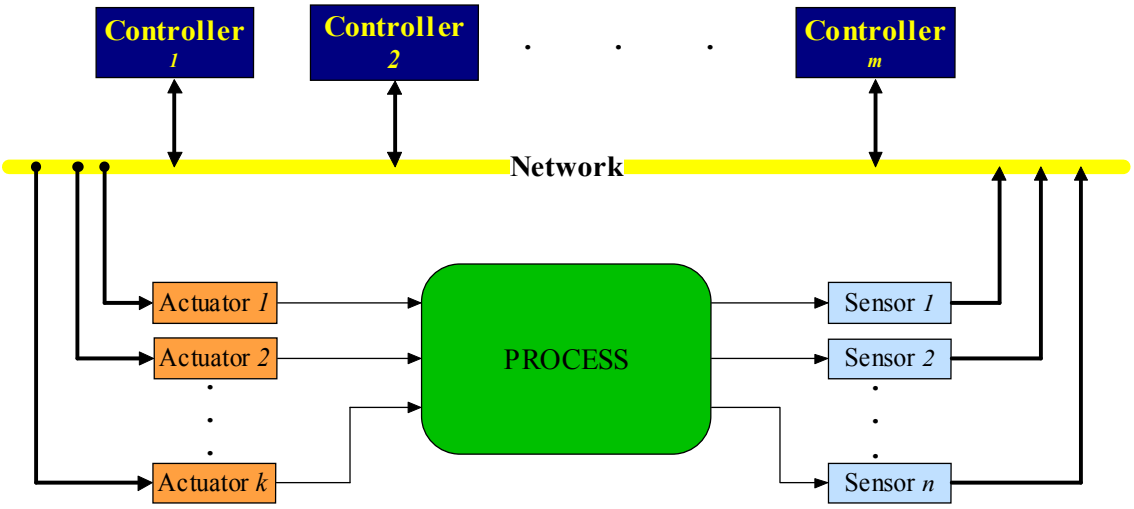


Fig. 1.1 Distributed control system

The main advantages of the NCS mentioned in [Wal 99] are as follows:

- low cost,
- reduced weight and power requirements,
- simple installation and maintenance.

In comparison with centralized control in NCS is significantly reduced the amount of used materials needed to interconnect all required parts of control system. This feature relates with important economic aspect as well as with possibility of more effective maintenance and diagnostic of the control system. More effective diagnostic and maintenance could be considered thanks to decreasing the level of complexity of interconnections among components in comparison with the same control system

realized in conventional (centralized) fashion. For example, X-by-Wire systems presented by Leen and Heffernan in [Lee 02] and Yih, Gerdes in [Yih 05], represent applications of NCS in the vehicular field for the control of brakes, fuel throat, etc.

On the other side, NCS brings new phenomena which are not described by classical control theory. One of the main problems of NCS are random delays caused by the network which could destabilize the controlled system [Zha 01]. There are other parameters related to network which have a direct influence on the stability or instability of the system:

- network bit rate,
- type of communication protocol,
- sample period of each of the nodes connected to the network (sensors, controllers, actuators – Fig. 1.1),
- number of connected nodes.

The mentioned parameters as well as others should be taken into account when a NCS is designed and realized for using in industrial applications. Several problems will be described in further text.

In conventional control systems (centralized control systems) measures, computing and actuating are strictly sequential. However, in DCS these three functions could be achieved in parallel. This feature brings problem of synchronization of all nodes connected to the network.

All the problems which could appear in NCS have an influence not only to the stability but also to other parameters for the quality of control (QoC) [Lig 04] as well as parameters and attributes of the system dependability (see chapter **1.2**) (maintainability, reliability, availability, etc). For example Sivencrona, Hedberg and Röklinger in [Siv 01] analyze the communication protocols used for networked data exchange among the nodes covered by NCS from the dependability point of view. In further text are introduced the necessary concepts to present the problematic of reliability analysis of networked control systems.

1.1.2 Communication networks in NCS

Networks used in NCS should be developed in order to be able to prevent critical failures as well as not to decrease the dependability of the whole system. Basic division of communication networks from utilization point of view is:

- clasical communication networks,

- technological control networks,

Other general division of the network could be as follows [Lig 03]:

- deterministic networks,
- non-deterministic networks.

In deterministic networks we can foreseen the time of transmission of the required amount of data as well as delay appeared on network. An example of deterministic network is WorldFIP. In case of non-deterministic network there are random delays and we can not determine the exact time of transmission.

The common communication network could be represented by Ethernet. Ethernet uses CSMA/CD (*Carrier Sense Multiple Access / Collision Detection*) [standard IEEE 802.3] [Lia 01a] mechanism for access on communication medium. This mechanism is realized as 2nd layer of ISO/OSI model [Par 99]. The main principle of this kind of access mechanism is collision detection by node when information is sent. When another signal is detected on the medium, then the transmission of the packet is interrupted and a *jam signal* is sent. At this moment, each node is waiting for a random time interval. The performance parameters are sufficient when the network loading is low. Increasing the amount of data which should be transmitted on the network (network loading) increases, in the same time, the number of collisions. Thus, the corrupted packets must be retransmitted and the network is gradually overloaded. Ethernet network with CSMA/CD mechanism has worse performance parameters in comparison with technological networks, when the network is high loaded. For example, with the same bit rate as in technological networks, the Ethernet network is not able to transmit required data within the required time intervals [Gal 07] [Tan 07]. Ethernet is a type of non-deterministic network. This feature is given by the inability of foreseen delays caused by the network due to its medium access method.

In order to prevent undesirable features and increase the dependability of the whole NCS, special networks were designed for industrial applications. There are several types of used control networks:

- CAN, CANopen, DeviceNet – represents the network with CSMA/CR [Par 99] or CSMA/AMP [Fre 95] medium access method,
- PROFIBUS, ControlNet, FlexRay – represents token passing networks with TDMA medium access method,

- Industrial Ethernet and other modifications.

The difference in approach is covered by different access mechanism for the nodes connected to the network. Generally, the most known technological networks apply CSMA/AMP (CAN) or TDMA (token passing bus / ring) medium access method. There are some modifications based on conventional communication network structure. Popularity of the classical Ethernet pushes it in industrial application. In order to be able to accomplish requirements given by its application within NCS, some features have to be modified.

The choice of a suitable type of control network (deterministic or non-deterministic) depends on network performance parameters given under different conditions. An important aspect is the dynamic of the controlled system as well as the complexity of the systems, number of components connected to the network and others which should not be ignored when the NCS is designed.

In further text, are presented two dominant different approaches developed for industrial networks:

- deterministic networks with TDMA access mechanism
- non-deterministic networks with CSMA access mechanism.

1.1.2.1 Networks with CSMA/CR (CSMA/AMP) access method

In order to design a suitable industrial network for control applications, the company Robert Bosh Gmbh in 1983 started the development of the CAN communication protocol (Control Area Network) [Par 99]. The main field of application of this type of network is the automotive industry. Specification “V 1.0” was presented in 1985. Further are briefly described CAN networks based on the information proposed by Dominique Paret in [Par 99] and [Par 05].

The CAN communication protocol is defined by ISO standard 11 898. CAN networks use CSMA/CR or /AMP (*Carrier Sense Multiple Access / Collision Resolution* [Par 99] or */Arbitration by Message Priority* [Fre 95]) medium access method. The signal on medium can be either “*dominant*” or “*recess*” state. There is no strictly defined that logical 1 or 0 is dominant state or contrary logical 0 or 1 is recess state. Usually, the dominant state is defined by logical 0. The networks with CAN protocol are also marked as *event triggered* networks. There exist some modifications of CAN protocol which are labeled as TTCAN – time triggered CAN (see next chapter - TDMA access method) [Par 05].

Arbitration mechanism is as follows. Each node could have a priority defined in order to determine its importance within the control system. The node listens to the communication. When the network is free, the node starts to send a frame which starts by its identification (priority number). In case when two nodes start the transmission at the same time, each node compares its state on medium with sent bit. When a dominant state is detected but the recess state was sent, then the node stops the transmission and the second one has won the access to the medium. In CAN specification, there are several different types of frames (data, remote, error, overload, etc.). When two nodes with the same priority (abnormal state) are sending *remote* and *data frame*, the remote frame has a higher priority (for more details see [Par 99] or [Par 05]).

As we can see from arbitration mechanism, there are no delays caused by collisions as in case of Ethernet network. This feature is very important in order to reduce the delays caused by networks as well as increase the dependability of the technological network.

CAN protocol are protected by error detection mechanism. Suppose that all global errors, local sender's errors and until five random errors sent in message are detected, then the probability of message corruption is less than 4.7×10^{-11}.

There are several networks based on CAN protocol such as CAL, CANopen, DeviceNet. For example, DeviceNet network implements CSMA/CD + NDBA (*Non Destructive Bitwise Arbitration*) arbitration mechanism. It is represented by bus topology with maximal bit rate determined to 500 kbits/s for 100m network length. It allows connecting 64 elements to a network.

1.1.2.2 Networks with TDMA access method

Another approach in control communication protocols design is proposed by TDMA (*Time Division Multiple Access*) medium access mechanism. This type of protocols is also called *time triggered* protocols. Using this type of protocol, it allows determining the maximal time between sending two sequential message frames within two token cycles (see further). Thus, networks with this type of protocol represent a group of deterministic networks.

The main principle of the TDMA mechanism is as follows [Lia 01a]. The network is organized as a logical ring. Each node knows the address of its predecessor and successor (ControlNet). Among all nodes connected to the network circulates a token in

the logical ring. The node with token transmits the data within a specified time interval. If the time limit has passed, the token is sent to the next node of the logical ring. If the token is not sent to the successor then the successor can generate a token and send data. If the node has no data to send, then the token is sent to its successor. As we can see the collisions can not occur because only one node can transmit data at a time. The nodes can dynamically access to the network as well as can send request to leave the ring. By using this type of protocol (TDMA), it is possible to guarantee a maximal time (delay) between two nodes connected to the network.

Some examples of control networks where communication is based on TDMA communication protocol are ControlNet, PROFIBUS or FlexRay. The bit rates can be increased up to 5 Mbit/s (ControlNet), 10 Mbit/s (FlexRay) and 12 Mbit/s (PROFIBUS). As we can see from bit rates values, all three control networks could be used in conditions of systems with fast dynamic.

1.1.3 NCS design

Developing networked control systems is a special complex part included in distributed control systems problematic. We can distinguish two basic aspects in NCS development. The first aspect considers control performance parameters i.e. quality of control [Lia 01b]. The Second aspect is related to design of NCS in order to obtain a sufficient reliability level.

The quality of control (QoC) depends on many parameters which have to be considered when a NCS is designed. Network's performance parameters have a significant influence to the final control performance parameters such as overshooting, rise time, etc. For example, combination of fast system dynamic and small network bit rate could cause network delays which could lead to the system destabilization or other non-desired consequences.

Lian in [Lia 01b] presents the dependence between sampling period and performance of NCS (see figure 1.2). In figure 1.2 there are three curves for continuous, digital and networked control as well as area of acceptable and unacceptable values of performance parameters. Continuous control curve is independent from sample period thus the performance is always in acceptable area. With digital (or discrete) control, we can see the change. With decreasing the sample period the discrete control curve is coming to an acceptable area ($\mathbf{P_A}$) and with further decreasing stays in this area whereas it is still closer to the continuous control curve. This behavior is described in the theory

of discrete systems. In case of networked control, there is an area defined by points **C** and **B**. With decreasing the sampling period, the curve appears in an acceptable area ($\mathbf{P_B}$). Suitable performance parameters are obtained until boundary value of sample period $\mathbf{P_C}$. By further decreasing the performance the curve is in an unacceptable area. Thus, the network is not able to transmit all required data within the required sampling period. It causes random network delays which destabilize the controlled value.

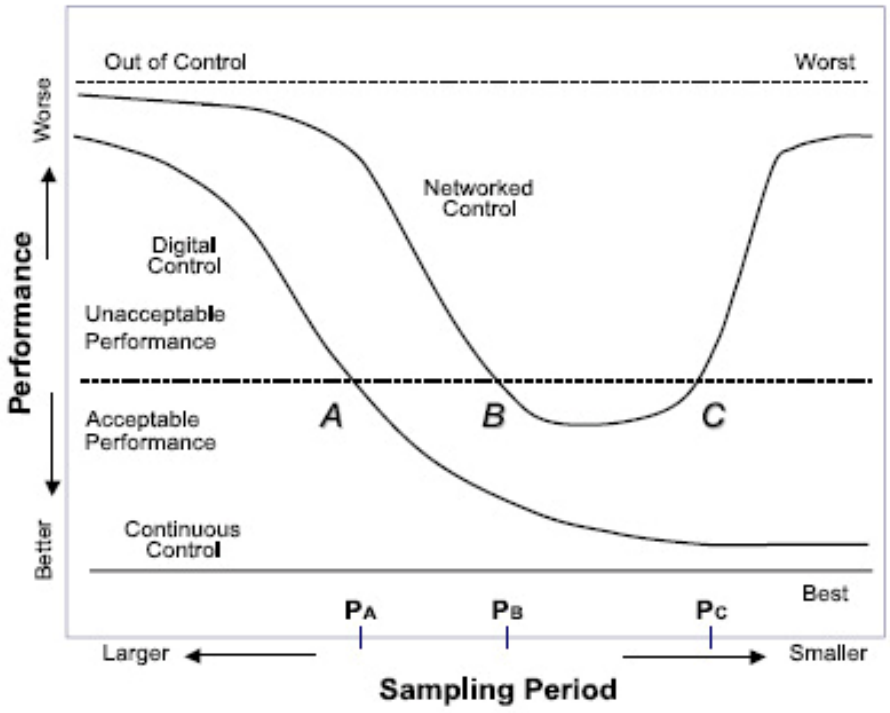


Fig. 1.2 Continuous, discrete and networked control

From figure 1.2 we can suppose several problems in networked control system design such as suitable network parameters in order to accomplish requirements given by system dynamics. There are different approaches in designing networked communication among all components of the NCS. Thus, communication among components could be provided as follows:

- time triggered communication (T),
- event-driven communication (E),
- components with defined deadbands (D).

Time triggered communication among sensors, controllers and actuators [Lig 04a] represents synchronous communication. Each component executes its task with a

defined sampling period *T*. Thus, values measured by sensors, actuating values computed by controllers and application of actuating value are synchronously sent and applied (Fig. 1.3). In this type of communication, there is a problem of synchronization of all components which are covered by the NCS. The problem of synchronization is difficult to solve and could cause undesirable delays. Thus, it could be another cause of delays together with wrong tuned sample period (Fig. 1.2).

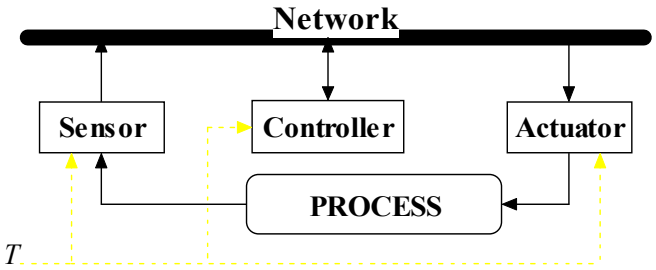


Fig. 1.3 Time triggered NCS – TTT

In order to prevent potential problem with components synchronization, it is possible to apply an event-driven communication technique (E). Normally, in event-driven NCS only the sensor runs periodically with the sampling period *T*. The sensor measures and sends its value with specified sampling period to the controller. The controller starts execute its task when receiving an event (the controller receives the measured value). The same behavior is implemented in the actuator. Thus, the actuator starts applying the actuating value when it is received from the controller. When network is overloaded, the time between two events is extended. Hence, some additional algorithm should be applied especially in actuator to prevent the system destabilization because of the application of old actuating values.

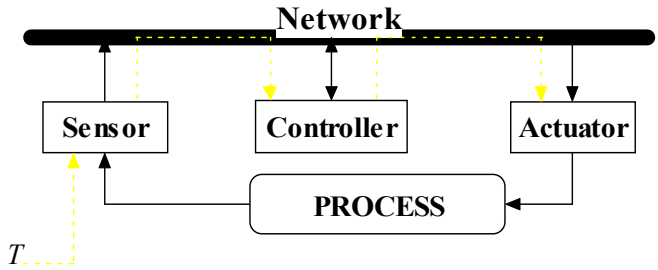


Fig. 1.4 Event-driven NCS - TEE

A third technique how to let communicate the components covered by NCS in order to prevent unnecessary communication among them is the implementation of the deadbands for sent and received values. The main principle of this communication is as follows. The sensor has specified limits for the measured value. These limits define whether a measured value has changed in comparison with the previous measured within defined limits or out of them. In case of smaller change (within limits), the value is not sent and the controller computes the actuating value with the last received value. In this type of communication, it is necessary to find suitable limits for all considered values.

1.2 Dependability of the systems

The term *dependability*, according to IEC 61069-5 [Aub 03], is a general term which covers the concepts of *availability* and *credibility* which covers *reliability*, *maintainability*, *integrity* and *security* term (Fig. 1.5).

A short definition of the dependability mentioned in [Kri 05] is as follows:

"The dependability is an ability of the system to perform properly."

Laprie in [Lap 92] proposes another definition of this term:

"Dependability is a trustworthiness of a computer system such that reliance can justifiably be placed on the service it delivers."

In figure 1.5 are shown all the relations among the mentioned concepts. As we can see, the reliability is covered by availability, which covers maintainability, too.

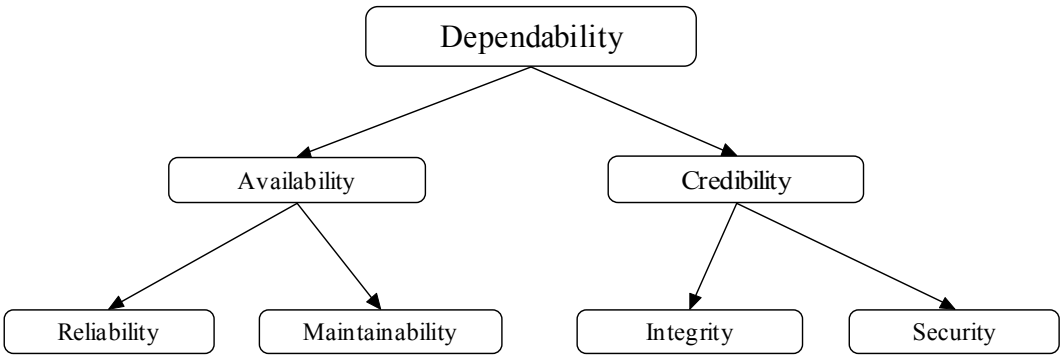


Fig. 1.5 Terms covered by dependability

Laprie in [Lap 92] introduces a larger tree of dependability which covers not only the attributes (availability, reliability, etc.) but also the means (the methods for the development of dependable systems as well as different types of impairments).

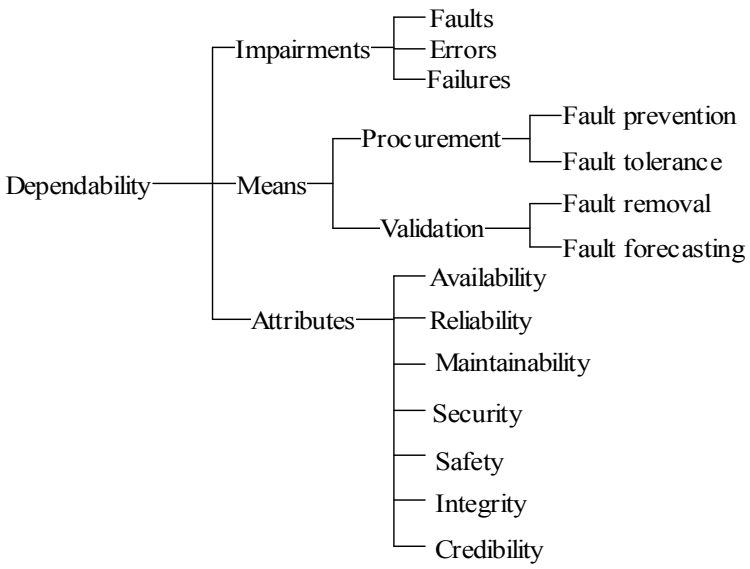


Fig. 1.6 Dependability tree proposed by Laprie in [Lap 92]

As we can see from figure 1.6 Laprie introduce three types of impairments – *faults*, *errors* and *failures* [Lap 92]. A system *failure* occurs when the system mission is not properly followed and the system behavior does not correspond to the expected function. *Error* could be represented by a system state which leads to a subsequent *failure*. And a *fault* is a hypothesized cause of an *error*. There are several divisions of the mentioned impairments such as internal / external faults given by the system boundaries or division depending on the failure perception such as consistent / inconsistent failures, etc.

Another sub-parameters introduced in figure 1.6 are means for dependability. There are two basic branches *procurement* and *validation*. The procurement covers *fault prevention* which includes the methods used in order to prevent fault occurrence. Another group of methods covered by procurement are *fault tolerance* methods which allows accomplishing a system mission when some faults have occurred. The second branch, validation, include the methods for decreasing the number of faults – *fault removal* as well as methods of faults prediction, estimation of number of faults and their consequences – *fault forecasting*.

Last part of the dependability tree introduces attributes of the dependability. In general, attributes correspond to the attributes in figure 1.5 and some of them will be defined in further text together with dependability parameters. However, Laprie in [Lap 92] presents four basic attributes (A*vailability*, *Reliability*, *Security*, *Safety*) and in addition presents other three attributes (*Maintainability*, *Integrity*, *Credibility*). In summary there is one additional attribute (*Safety*). This attribute is in newer literature [Aub 03] covered by *Security* attribute as it is shown in figure 1.5.

1.2.1 Attributes and parameters associated to the dependability

As it was abovementioned, the dependability covers the terms divided among several groups (Fig. 1.1). The definition some of the attributes as well as parameters are further provided. The *reliability* as first attribute of the dependability is defined as follows [Sta 98]:

The *Reliability* $R(t)$ of the system is the probability that a system is up and running correctly during the time interval $<0, t>$.

$$R(t) = P[\textit{system will not failed during the time interval } <0, t>]$$

The definition is proposed in standard IEC 50(191) too [Aub 03].

The second attribute is the *Availability* of the system $A(t)$ defined as the probability that a system is up and running correctly at time t.

$$A(t) = P[\textit{system will not failed at time } t].$$

The average availability [Aub 03] of the system within the time interval $<t_1, t_2>$ is defined as:

$$\overline{A}(t_1, t_2) = \frac{1}{t_2 - t_1} \int_{t1}^{t2} A(t)dt \qquad (1.1)$$

The complement to availability is the unavailability which is defined as:

$$U(t) = \mathrm{P}[\textit{system will failed at time } t\ ],$$

and holds

$$A(\mathrm{t}) + U(\mathrm{t}) = 1 \qquad (1.2)$$

The maintainability $M(t)$ is defined as a probability that the system will be restored within a specified period of time t.

$$M(t) = \mathrm{P}[\textit{system will be repaired during time } t]$$

In order to measure the attributes defined previously, specialists in dependability have defined some parameters: for example Mean Time To Failure (*MTTF)* and Mean Time To Repair (*MTTR)*.

MTTF is defined as:

$$MTTF = \int_0^{\infty} R(t)dt \tag{1.3}$$

where *R(t)* is the reliability of the system (detailed definition see in chapter **1.2.3**).

MTTR is defined as:

$$MTTR = \int_0^{\infty} [1 - M(t)]dt\,. \tag{1.4}$$

Other definitions of the concepts and terminology related to system dependability are presented in mentioned IEC standards.

1.2.2 SIL levels

Through the attribute of the dependability, we could divide the systems into several categories described by International standards IEC 61508 and 61511 which represent Safety Integrity Levels. According to [Mar 02], the SIL is:

"*The key design parameter specifying the amount of risk reduction that the safety equipment is required to achieve for a particular function in question.*"

There are four levels [Mar 02] divided by value Probability of Failure on Demand (PFD) or Frequency of Dangerous Failure per Hour (FDFH). All levels and values of PFD and FDFH are introduced in table **1**.

Safety Integrity Level (SIL)	Average PFD Range	FDFH Range
4	$10^{-4} \rightarrow 10^{-5}$	$10^{-8} \rightarrow 10^{-9}$
3	$10^{-3} \rightarrow 10^{-4}$	$10^{-7} \rightarrow 10^{-8}$
2	$10^{-2} \rightarrow 10^{-3}$	$10^{-6} \rightarrow 10^{-7}$
1	$10^{-1} \rightarrow 10^{-2}$	$10^{-5} \rightarrow 10^{-6}$

Table 1.1 Safety integrity levels and corresponding PFD and FDFH

Except other terms, standard IEC 61511 defines safety instrumented function (SIF) as an action which provides the controlled process or component in a safe state. These actions are realized in different modes of SIL assignment. Based on different types of values (PFD, FDFH), we can divide SIL assignment into two modes [Mar 02]:

- Demand mode (Low-demand) of operation – demands to activate safety instrumented function (SIF) are infrequent compared to the rest of the SIF.
- Continuous mode (High-demand) – demands are placed on the SIF much more frequently.

The first mentioned mode is most common in process industries. Based on average PFD value we can determine the SIL by using the followed formula:

$$\mathrm{SIL} = -\log_{10}\left(PFD_{avg}\right).$$

The second one is more common in the machine industry and avionics. As we can see, in this mode the SIL category is described by the number of dangerous failures per hour. The value represented by SIL 1 supposes approximately one dangerous failure every ten years.

In common DCS, there are different subsystems or group of subsystems with several levels of SIL. Then, the SIL of the entire system (DCS) has a value equal to the SIL of the smallest level: this is done under the assumption that there is no redundancy.

In order to determine the safety integrity level there are several different methods [Gru 98]. We can consider qualitative and quantitative approaches. Gruhn and Cheddie in [Gru 98] presents the followed qualitative methods based on:

- Evaluating the *frequency* of an event – five levels from *Improrable* (1^{st}) to *Frequent* (5^{th}) probability of an event.
- Evaluating the *severity* according to the different factors at risk (production, etc) – divided into five levels from *Negligible* (1^{st}) to *Catastrophic* (5^{th}) severity.
- Evaluating the *overall risk* – table scheme where combination of all levels of the frequency and severity is shown.
- Evaluating *effectiveness of additional layers* – three axes graph where with frequency and severity is added a third axe which should take into account additional safety layers.

The quantitative method is based on two features such as the target safety goal and demand rate (failure rate). As an example of target safety goal is given the Mean Time Between Hazard (*MTBH*). Base on this parameter the simple formula is proposed [Gru 98]:

Required Risk Reduction Factor = MTBH / Demand Rate

A variation of the quantitative methods is described in the AIChE CCPS text as the layers of protection analysis (LOPA) [Mar 02].

1.2.3 Mathematical description

Let's suppose that a system involves statistically independent and Boolean components [Bar 81], the state of the i^{th} component is random and determined by the probability:

$$P[s_i = 1] = p_i \text{ for i = 1, 2, ..., n.} \quad (1.5)$$

As well the state S of the system θ is represented by the states of each component, thus $S = (s_1, s_2, ..., s_n)$. Similarly, the reliability of the whole system θ is given by the equation:

$$P[\theta(S) = 1] = h, \quad (1.6)$$

where $h \in \langle 0;1 \rangle$.

Thus, it is possible to represent the reliability of the system as a reliability function of each component:

$$h = h(p), \quad (1.7)$$

where $p = (p_1, p_2, ..., p_n)$ and h(p) is the reliability function of the system at the fixed time. From the point of view of the components life lengths, the reliability function $R(t)$ can be defined as the probability that the system stays in the operating state without failure over the entire time interval $(0, t\rangle$.

A more precise definition of the reliability is as follows:

Let τ be the random variable that represents time to failure of the system under study, thus τ being a time quantity and defined only for non-negative values.

Hence the reliability is given by [Bar 81]:

$R(t) = P\{\tau > t\}$

and holds

$$R(0) = 1, \lim_{t \to \infty} R(t) = 0 \text{ and } R(t) \text{ is not increasing in } t.$$

Also the unreliability $Q(t)$ can be defined as the probability that the system fails at any time interval $(0, t\rangle$. Thus:

$$\mathrm{Q}(t) = \mathrm{P}\{\tau \le t\}. \quad (1.8)$$

Unreliability is the complement of $R(t)$:

$$\mathrm{Q}(t) = 1 - \mathrm{R}(t). \quad (1.9)$$

For completeness, further are provided the definitions of density of probability of failures and failure rate [Bar 81]. The density of probability of failures *f(t)* is given by:

$$f(t) = -\frac{dR(t)}{dt}, \tag{1.10}$$

and the failure rate $r(t)$ by:

$$r(t) = \frac{f(t)}{R(t)}. \tag{1.11}$$

By integrating both sides of the previous equation and its editing, we can obtain the representation of the reliability as a function of the failure rate.

Thus:

$$R(t) = \exp\left(-\int_0^t r(t)dt \right). \tag{1.12}$$

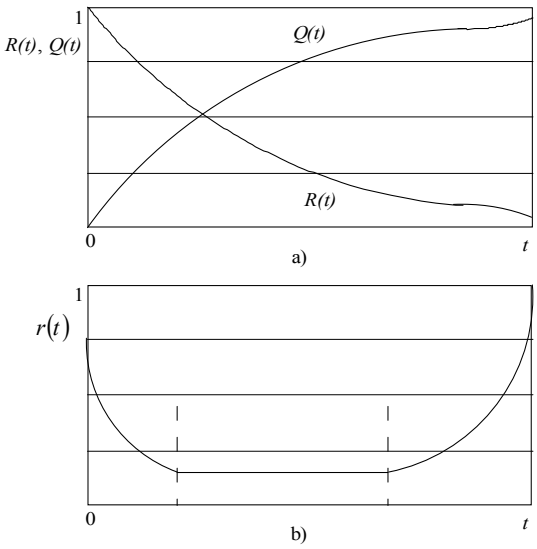


Fig. 1.7 Common shape of the curves R(t), Q(t) a r(t)

In figure 1.7 are shown standard shapes of the curves given by functions *R(t)*, *Q(t)*, and $r(t)$ [Sta 98]. The reliability $R(t)$ of the system (component) is decreasing with its length life whereas $Q(t)$ is consequently increasing (Fig. 1.7a). The failure rate curve $r(t)$ [Sta 98] is most suitable for component life description. In the figure 1.7b, the curve is divided into three parts. In the first part, the curve is decreasing, this life period is denoted as *period of frequent failures*. In the second part, failure rate is constant. Hence, this component life period is denoted as *period of constant rate failure*. Last part of the

failure rate curve is increasing and denoted as *period of the component failures given by its aging*.

Reliability can be divided [Cam 97] into several categories:

1. Operational reliability – is the result of the measuring (observations and analysis) the component during its operational conditions.
2. Extrapolated reliability – is extension of the operational reliability by extrapolation for a longer time interval.
3. Predicted reliability – is foreseen reliability for farther future.
4. Intrinsic reliability – is reliability obtained during the component test.

Roshandel in [Ros 04] distinguishes the reliability from the complexity point of view:

- *Local* reliability – reliability of one part (component) of the system.
- *Global* reliability – overall system reliability.

The process of reliability estimation could be described by two basic approaches, such as:

- Analytical,
- Simulations.

Analytical approach is based on the mathematical methods which are applied on the equations derived from prepared mathematical model of the system. These methods use exact or approximate mathematical techniques to solve the equations. Another approach uses the simulations. The model is prepared in order to be executed a certain number of times; this number must be determined in order that reliability estimation be statistically acceptable.

1.3 Different methods of dependability analysis

For dependability parameters estimation of the studied system, it is necessary to prepare a model of the system in order to evaluate formally the parameters or to estimate it from simulations of a model. Primary, we will focus on the reliability analysis. Thus the probability of system failure within a specified time interval could be determined by several different approaches for modeling and evaluation. Further will be described the followed methods:

- Markov models,
- Fault trees and reliability block diagrams,
- Binary decision trees,
- Bayesian networks,
- Petri nets and Monte Carlo simulation.

These five methods are chosen among various techniques, methods and their modifications determined for reliability analysis or prediction. These methods are widespread in reliability analysis. Rodrigues and others describe in [Rod 05] methods for the reliability analysis if the system uses scenarios. The method is composed of five basic steps:

1. annotation of the scenarios,
2. synthesis of the probabilistic labeled transmission system,
3. construction of the stochastic matrix,
4. system reliability prediction
5. implied scenario detection.

By accomplishing these steps, Rodrigues [Rod 05] and others present the possibility to estimate the reliability of the studied system.

In the following chapters are presented methods and tools suitable for modeling and evaluation of the reliability. The extensions of these methods are described too. For example, dynamic fault trees, dynamic Bayesian networks, etc. The last method – Petri nets is divided into two parts. The first part describes Petri nets as a tool suitable for reliability modeling whereas the second part of this chapter presents Monte Carlo simulation as a statistical method for the evaluation of the reliability as well as other attributes related to the dependability of the modeled system [Lap 92]. These two methods are used together when the reliability is analyzed.

1.3.1 Markov models

Markov models could be noted as the most known technique how to determine the reliability of a system. These models can be considered as functions of two random variables, state of the system and time. Models with discrete states and discrete time are known as Markov chains. Models with discrete states and continuous time are called Markov process.

The creation of the Markov model can be divided into the following steps:

- Definition of all the possible states of the system – initial states and final states [Sta 98]. States of the system which are composed of more components are determined by combination (which are possible) of the states of all components.
- On the basis of defined initial and final states are prepared differential equations which describe the probabilities of the transition from initial to final states during the time interval Δt. These equations are prepared on these two conditions:
    - Probability of the transition p_{ij} from state i to state j in time interval Δt is given by $p_{ij} = \lambda_{ij}(t)\Delta t$, where $\lambda_{ij}(t)$ is the probability rate of the transition between two states (i and j).
    - Probabilities of more than one transition in time interval Δt are insignificant.

The reliability of the modeled system can be determined by solving of the prepared differential equations [Sta 98] [Fri 97].

The size of the Markov model depends on the number of states of the modeled system. Each state corresponds to one differential equation. Thus, when the system is composed of n components with k possible states then the number of states of the system is $k*n$.

The advantage of the described technique is that it is simple using when a system with few states is modeled. When the system is large, Boudali and Dugan in [Bou 05a] [Buc 99a] present some disadvantages as the manual generating of the Markov models becomes a cumbersome, error-prone task; furthermore, the problem of the state space which grows exponentially with the number of components in the system can also be a difficulty.

Problems with results given by Markov models when the reliability of the redundant systems is evaluated are mentioned by Smith and Simpson in [Smi 04]. The results of the reliability analysis are more pessimistic than real results.

In order to compensate some of the disadvantages, Bouissou in [Bou 02] presents *Boolean logic Driven Markov Process* (BDMP). This formalism is presented as a combination of the Markov models and the method of *Fault trees* (FT) (see next chapter). BDMP includes new types of gates which use different semantics in comparison with FT. From a mathematical point of view, BDMP is a method for

definition of global Markov process based on few basic processes (gates) whose interact among themselves.

The main advantage presented by Bouissou is a significant decreasing of the combinatorial problems of the Markov models.

1.3.2 Fault trees and reliability block diagrams

Fault trees or *static Fault trees* (FT or SFT) are a symbolic analytical logic technique that can be applied to analyze system reliability. Fault trees are determined for the representation and analysis of the critical conditions which causes a specific event called Top event (TE). The analysis of the combination of elementary events leading to occurrence of the TE is called Fault trees analysis (FTA). FTA is commonly used by reliability engineers dealing with aircraft, chemical and nuclear systems, and it is also considered in the IEC-1025 standards. To introduce FT as a tool for reliability, it is necessary to define more precisely few basic terms.

TE is an event which occurrence may lead to unsafe operating conditions, dangerous failure, and so on. As it was mentioned, TE is caused by particular undesired events (input and output events) that are connected with gates coming from the traditional Boolean approach (AND, OR and extended as *k*-out-of-*n*, etc.). Reliability given by OR and AND gates is as follows:

$$R_{OR} = \prod_{i=1}^{n} R_i \,,\; R_{AND} = 1 - \prod_{i=1}^{n} (1 - R_i) \,. \tag{1.13}$$

The system k out of n is an extension of the parallel system. This system is working properly when k out of n components of the system are still working without failure (properly). Reliability of this system is given by following equation:

$$R_S(t) = \sum_{i=r}^{n} C_n^i R^i (1-R)^{n-i} \,. \tag{1.14}$$

The disadvantage of the classic static FT is its inability to model dynamic dependencies among the components covered by the studied system. To solve this problem, there are some extensions of classic FTs such as dynamic fault trees (DFTs) [Bou 05a] or extended FTs (eFTs) [Buc 99a] [Buc 99b] [Buc 00].

eFTs are developed as a combination of classic FTs with additional constructs and possibility of modelling the stochastic dependencies between basic events [Bob 03].

In the case of DFTs, the extension is realized by defining special gates to capture the component's sequential and functional dependencies. For example the defined dynamic gates are [Bou 05a] [Bou 06] [Bou 05b]:

- The functional dependency gate (FDEP),
- The cold, hot and warm spare gates (WSP),
- The priority AND gate (PAND),
- The sequence enforcing gate (SEQ).

As it was mentioned above, the extension of the static FT brings features which allow model the reliability of the systems with their dynamic interactions. All gates can be combined in order to prepare model of the system with all dependencies (among subsystems) which could be modeled by predefined gates (FDEP, etc.).

Similar tools as static fault trees are *reliability block diagrams* (RBD). These tools use blocks arranged in serial or parallel sequance (Fig. 1.8) and can be considered as adequate substitution to SFT. All standardized gates (OR, AND, *k*-out-of-*n*) included in SFT are realizable using the RBD.

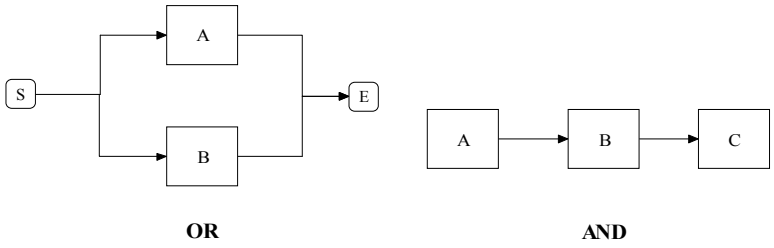


Fig. 1.8 RBD – equivalences to gates OR and AND

Advantages

Common advantages of both tools are the simplicity of utilization. Design of the reliability model is very intuitive from basic events through simple gates or blocks to final top event (in case of FT). However, important advantages are brought by different extension of the FT which allow modeling different dependencies appeared within dynamic systems.

Disadvantages

The main problem of the static FT is the inability to prepare a model of a system with dynamic dependencies. Even if this weakness is compensated by FT, there are

always some specific dependencies which have to be modeled by another more complex tool.

1.3.3 Bayesian networks

Another approach how to model the system reliability is using Bayesian networks (BN) which are a form of probabilistic graphical model.

A BN [Zha 01] [Bou 06] is a set of nodes (random variables) and directed edges between them. Random variables and directed edges form a directed acyclic graph and holds:

- each variable has a finite set of mutually exclusive states,
- to each variable A with parents $B_1,\ldots,B_n$ is attached a conditional probability table $P(A \mid B_1,\ldots,B_n)$.

Another definition is as follows. BN is a graph with one-way edges that connect nodes. The nodes represent assertions about relationships between them. Thus, a node A is related to a node B by an edge that represents that A is a cause of B, with a certain probability. Root nodes have marginal prior probability tables associated with them, and all other nodes have Conditional Probability Tables associated with them [Bou 05b].

For example, let A be a parent of B and C. Then the variable B is attached by a conditional probability $P(B \mid A)$ and variable C is attached by a conditional probability $P(C \mid A)$. Thus $P(B) = P(B \mid A)$ and $P(C) = P(C \mid A)$.

When a distributed control system is modeled, each node represents one element of the studied system. This element could represent a component or a subsystem which covers several components and interactions among them [Bou 04].

In order to take into account other properties, several extensions are proposed. The most known extensions are as follows:

- timed BN [Bou 04],
- dynamic BN [Mur 02] [Bur 05],
- continuous BN [Bou 06],
- discrete BN [Bou 05a].

These extensions of BN allow benefit from advantages which are brought by new implemented features. In [Bou 06] Boundali and Dugan describe some differences between continuous and discrete BN as well as some advantages. Neil and others in

[Nei 06] use hybrid BNs for systems modeling, as a combination of continuous and discrete BNs.

Advantages

Bayesian networks bring the advantages given by their extensions such as dynamic, timed, continuous BN, etc. The new features given by extensions extend field of application of the BN for reliability modeling of the systems as well as some specific dependencies within the systems [Bou 05b], [Nei 06].

Disadvantages

In case of systems with different dynamics within different subsystems, it is difficult to model this feature regardless of its extensions given by timed BN. The BN are more focused to the diagnostic and event prediction.

1.3.4 Petri nets and Monte Carlo simulation

Petri nets (PN) are a tool which can be widely utilizable for modeling different types of systems such as production processes, control systems, networks, system reliability estimation and others. In order to be able to model different systems and solve different tasks there are several types of PN. For example, for discrete control systems as well as for deterministic networks modeling are used timed PN [Dia 01] which allow modeling of the deterministic discrete time events. In the case of system reliability *stochastic Petri Nets* (SPN) can be used [Vol 04a]. SPN is a special case of timed PN where the transition firing times are considered as random variables.

Petri nets is a bipartite directed graph represented by a set (P, T, I, O) where:

- $\mathbf{P} = \{p_1, ..., p_n\}$ is a finite set of places.
- $\mathbf{T} = \{t_1, ..., t_m\}$ is a finite set of transitions.
- $\mathbf{I}(p, t)$ is a mapping $\mathbf{P} \times \mathbf{T} \rightarrow \{0,1\}$ corresponding to the set of directed arcs from places to transitions.
- $\mathbf{O}(t, p)$ is a mapping $\mathbf{T} \times \mathbf{P} \rightarrow \{0,1\}$ corresponding to the set of directed arcs from transitions to places.

Generalized Petri nets are represented by set ($\mathbf{P}$, $\mathbf{T}$, $\mathbf{I}$, $\mathbf{O}$, $\mathbf{M_0}$) where $\mathbf{M_0}$ is the initial marking of the PN.

The basic principle of functioning is as follows. Directed arcs connect places to a transition and transitions to a place (input and output arcs). To each place can be assigned a nonnegative number of tokens. The distribution of tokens in the PN is called the marking of the PN. A transition is enabled if each of its input places has a number of tokens which is required by arc (arc weight) which connect the place to a transition. When the transition is enabled, it can be fired immediately or after a specified time interval. Upon firing, a transition removes a token from its input places to its output places.

Stochastic Petri net [Dav 97] is an autonomous PN where each transition is associated with a random variable λ. The firing time for each transition is determined by this random variable. When firing time is exponentially distributed then the marking process is mapped into a continuous time Markov model [Bob 03]. Thus, in SPN transition wait for firing non-fixed time but time interval is determined by a random variable with an exponential distribution. An example of the system failure / repair is shown in figure 1.9. There is also shown the equivalent continuous time Markov model. The Stochastic Petri Nets are a dynamic modeling method (tool) [Ham 05] for system reliability modeling.

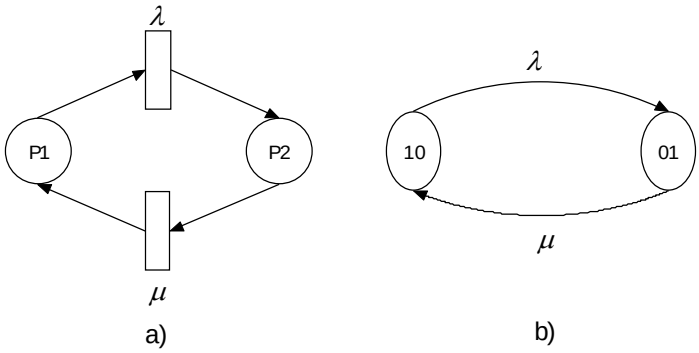


Fig. 1.9 SPN failure / repair model and its continuous time Markov model equivalent

As it was abovementioned, there are few types of SPN which include different extensions [Buc 99a] [Fri 98] [Ros 04], as *generalized* SPN, *Markov* SPN [Fri 98], *ESPN*, *Stochastic Reward nets*, *SPN with aging tokens* [Vol 04b], *Stochastic Activity Networks* with *hierarchical* extension [Azg 05] as well as *Stochastic Well-formed Nets* [Chi 93] [Bob 04] with *colored / parametric* extensions.

In generalized SPN, there are two types of transitions:

- Immediate transitions with zero firing time.
- Timed transitions with exponential firing time given by a random variable.

Immediate transitions are used for extremely short events modeling an immediate activation of a redundant component when a primary component has failed, for example.

Reliability modeling using PN presented by Pimentel in [Pim 02] (Fig. 1.10) could be provided by the following steps:

1. System model preparation – modeling each part of the system (components) and interconnections among them.
2. Testing the system – provide the test of the modeled system during the specified time interval which can be considered as statistically significant. The aim of the testing is to determine how many times the system has operated successfully and how many times the system was unable to operate in the required conditions (system failed).
3. The results of the test are two sets S and F, where S is the set of the system success and F is set of the system failures.
4. After then, the reliability of the modeled system is given by:

$$R = \frac{|S|}{|S| + |F|}, \tag{1.15}$$

In order to obtain relevant results from multiple simulations it is necessary use some appropriate method. Monte Carlo simulation is very useful statistical method which provides evaluating of dependability parameters and attributes.

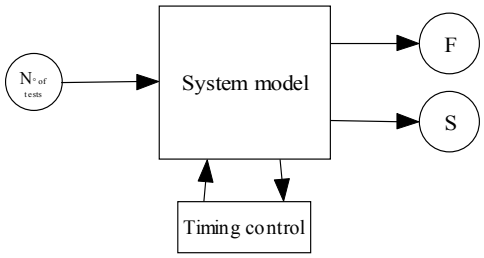


Fig. 1.10 Structure of a PN model for reliability evaluation

.

The principle of this method is to simulate several times a scenario in order to get statistically representative results. The exactness of this method depends on the duration

of each simulation and on the number of provided simulations. The number of simulation to obtain the results with precision ε can be given by [Bar 03]:

$$n = \left(\frac{u_{\alpha/2}}{\varepsilon} \right)^2, \tag{1.16}$$

where $\alpha = 0.1$ and $u_{\alpha/2} = 1.645$. Mentioned constants are table values.

In addition there are several methods which allow computing the error of MCS during the simulations. When error of the generated samples is within some specified interval then simulations are stopped [Gho 08].

In [Sta 98] is described the method of the static modeling which is one of the Monte Carlo methods. The procedure of dependability parameters computation by using this method can be summarized in the few following points:

1. Computing the random values of the time to repair and time to failure.
2. Computed random values are processed by preparing algorithms to compute time to failure and time to repair of the system.
3. The first and second simulation steps are repeated *n*-times.
4. The output values obtained in each 2nd step are used to compute required parameters of the system.

Error of the mentioned method is given by:

$$\varepsilon = \frac{3\sigma(X)}{\sqrt{n}}, \tag{1.17}$$

where X is random variable, $\sigma(X)$ is the standard deviation and n is the number of provided simulations.

In order to increase the exactness as obtain smaller standard deviation holds followed formula

$$n_1 = 2^i * n, \tag{1.18}$$

where n_1 is number of simulation to reduce standard deviation *i-th* times. Thus, decreasing the standard deviation to half value is necessary provide $2^2 * n$ simulations, hence holds

$$\varepsilon_1 = \frac{\varepsilon}{2} = \frac{\sigma(X)}{\sqrt{n_1}} = \frac{1}{2}\frac{\sigma(X)}{\sqrt{n}}, \text{ for i = 2.} \tag{1.19}$$

Advantages

Petri Nets are a really complex tool allowing to model different types of systems from simple sequence systems up to complex systems composed of many subsystems where different kind of dependencies could be modeled.

Disadvantages

The ability to model complex systems could be a disadvantage of this tool, particularly in the case of modeled system without any specific dependencies are considered. Thus, in case that some abovementioned tool with its extension is sufficient to prepare required model of the studied system. Hence, to model basic events with basic dependencies among subsystems it is of course better to use some simple tools such as fault trees.

1.3.5 Comparing the features of the reliability analysis methods

In order to complete the advantages and disadvantages or weak place to each described tools was prepared several criteria with regard on requirements of NCS. There are several conditions which should be considered in order to be able to prepare a model of the NCS and estimate its reliability as well.

The main requirement for the modeling tool is implementing the random events into a model of the studied system. A very important feature which has to be supported is the ability to model dynamic systems which encapsulates different dynamic dependencies. These dependencies could be divided between standard and non-described. Standard dependencies could be modeled by using extensions of the modeling tool. These extensions describe mathematically some possible dependencies among subsystems. For example, in fault trees we could use several specific gates. In case of non-described dependencies, it is necessary to find suitable tool which could be used for describing and modeling these dependencies. There are other features which should be considered and compared. In table 2 is a short comparison, of all tools according to the considered features.

The criteria for comparing presented methods are as follows:

1. Ability to model the failures appeared within the studied system – random events modeling.
2. Limits of method’s usability in conditions of networked control.

3. Ability to implement a mathematical model of the studied system within the model for reliability analysis.
4. Generating graphical or numerical representation of studied dynamic system behavior.
5. Ability to evaluate the reliability of the studied system.

Criteria / **Tools**	Markov models	Fault trees, RBD	Bayesian networks	Petri nets and simulation Monte Carlo
Random events modeling	Yes	Yes	Yes	Yes
Limitation of using in complex conditions of NCS	Yes(1)	Yes(2)	Yes	Yes
Implementation of the system state space within reliability model	No	No	No	Yes
Numeric / Graphical representation of the system behavior before / after failure	No	No	No	Yes
Ability to evaluate the reliability	Yes	Yes	Yes	Yes

Table 1.2 Comparison of the tools for reliability modeling

(1) In case of using BDMP.

(2) Using the extensions static fault trees such as DFT or eFT.

Based on table 2, we can summarize the possibilities of all abovementioned tools for reliability analysis. As we can see, a required but not sufficient condition is fulfilled by each mentioned tools from Markov models up to Petri nets.

In case of limits of using in condition of NCS, each method is more or less useful. The exceptions could be the Markov models and Fault Trees. In order to be able to model dynamic systems such as NCS by using Markov models, it is necessary to use some extensions such as BDMP. BDMP decrease the complexity of using in conditions of complex systems. Similarly, static Fault Trees with extension such as dynamic FT and extended FT allow model real described dependencies among components of non studied system. These extensions significantly enlarge the field of their using in analysis of dynamic systems.

As first condition of modeling, random events as well as condition of ability to reliability evaluation (estimation) is accomplished by all described tools. Each method uses different technique, however quantitative reliability level can be determined.

One exception among mentioned tools are Petri nets. This tool represents complex technique how to model different types of dynamic systems. This tool is one that allows implementation of mathematical model of the system in order to show its behavior before, during and after failure. However, most important is that numbers of possibilities given by its different specifications allow model specific dependencies whose influence to reliability of the studied system is not described. Thus, in case when extension of other tools prepared for standard dependencies can not be used then PN can be very useful due to their ability to various kind of the systems.

1.4 Conclusion

This chapter introduces the problematic of distributed systems on the field of the automatic control. This bibliographic review was prepared in order to show different features of the distributed control systems i.e. networked control systems. Implementation of the networks within control systems brings several new phenomena which are not considered in common continuous control systems. Networks and its influence to quality of the control represent a significant role. Hence, several different network protocol types were introduced (CSMA/CD, CSMA/AMP, etc.). Different approaches in NCS design shows different ways how to partially eliminate undesirable influence of the network delays, components' asynchronism, etc.

Other parts of the chapter cover dependability of the systems and methods for dependability evaluation primary focused on the reliability analysis. As it was shown, in order to characterize the security of the systems, several categories with different security levels, the SIL levels were developed. For estimation of the dependability parameters as well as its attributes several methods are briefly introduced.

The main difference among all methods is given by the ability to model dynamic systems. In context of NCS, it is necessary condition which has to be fulfilled in order to prepare a representative dependability model of the real NCS. Thus, static methods as SFT or RBD are unusable when the relevant result describing the real state should be obtained. Finally, the short summary of the necessary as well as useful features were presented in table 1.2 for a good selection of the suitable method. Each method offers

the extensions for modeling larger group of systems however there are always some limits which should be considered.

Based on the analysis of the mentioned tools and methods, Petri Nets based tool are a solution pertinent to prepare the dependability model of the further studied networked control systems. Petri Nets, in combination with Monte Carlo simulation represents a effective tool for modeling and the estimation of the dependability of the NCS as well as specific dependencies appeared. These dependencies appear with implementation of the different approaches in order to improve the dependability of the NCS. In addition, there are several doctoral dissertations which solve the problems of the NCS [Lig 04] as well as dependability evaluation of the NCS [Bar 03] by using the PN based tools.

2. The aims of the doctoral thesis

1. Description of the influence of the different failure rate behaviors to the final dependability attributes and parameters of an NCS with specified types of shared redundant subsystems – Application two and three components quasi-redundant subsystems.
2. Proposition of the effective method for dependability evaluation of the systems with quasi-redundant subsystems.
    a. Proposition of the reliability approximation method which cover the behavior of the systems with dynamically change failure rate during the system mission.
    b. Proposition of the MTTF (Means Time To Failure) approximation method for quasi-redundant subsystems with dynamically changed failure rate.
3. Generalization of the proposed effective method for defined structures of the shared redundant subsystems.
4. Definition and evaluation of reconfiguration strategy for networked control systems with implementation of the shared redundant approach.
5. Demonstration of the methodological technique in order to analyze the dependability of the NCS – demonstration and verification of the dependability evaluation by using for proposed method.

3. Modeling the system with various redundant strategies

In general the NCS as all system follows its mission which should be accomplished with guarantees. The system mission can be disturbed or interrupted by failures which occur with some probability.

To avoid or compensate faults occurred during the system mission several different techniques exist. The common way how to design more reliable control systems is using the fault tolerant approach. There are two basic different approaches how to eliminate the failures. The first one is software approach and deals with developing a numerical method [Sar 92] which allows tolerating some failures occurred during the system mission. There are several methods such as robust control design in combination with other techniques based on system model as estimation, prediction or diagnostic algorithms [Shu 08]. As was abovementioned this approach could be labeled as software approach.

The second approach is based on control system design where some additional redundant components are considered. Thus, except primary component as sensor or controller, there are secondary redundant components determined to save the system mission when a primary component will fail. Hence, this approach could be considered as a hardware approach. This hardware approach which is represented by using the redundant components will be considered further. Generally, there are two basic types of redundancy [Pim 02]:

- *Dynamic – Passive* replication (cold)
- *Static – Active* replication (warm/hot)

The combination of these two types of redundancy can be considered as *hybrid* redundancy. The principle of operation of both basic types of hardware components replication is described in more details in the further text. Except these two basic types another type of redundancy called *shared redundancy* can be considered.

The spares within a system create dynamic dependencies which have to be taken into account in dependability analysis in order to obtain relevant results. As it will be shown further, some of them, commonly known, are described by equations. In the case of *shared redundancy*, there are several specific dependencies, based on different possible scenarios, which have significant influence on the final reliability of the

system. The scenarios and description of the *shared redundancy* will be presented after bringing closer abovementioned two basic redundant techniques (*Dynamic* / *Static*).

3.1 Common redundant subsystems

3.1.1 Passive redundant components

Passive or *dynamic* redundancy is composed of the primary active component and the secondary passive component (*replication*), which starts operate when the primary component failed. Suppose a system with one primary C_P and several secondary components $\{C_{S1},\mathrm{K}\,,C_{Sn}\}$ represented by the block diagram shown in figure 3.1. Switching among is realized by using a simple switch *sw*. In the ideal case the shown switch has zero probability of failure. However the real probability of the failure should be significantly lower than the probability of the failure of the passive redundant systems as well as system(s) speared by them.

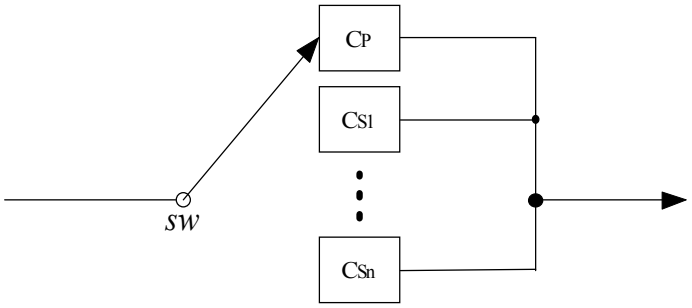


Fig. 3.1 System with passive redundancy

In the simplest case a failure-free operation of the switch *sw* is considered and each secondary component has a constant failure rate λ. The reliability of this system can be computed by the following equation [Bar 03]:

$$R_P(t) = e^{-\lambda t}\sum_{i=0}^{n}\frac{(\lambda t)^i}{i!}. \tag{3.1}$$

Figure 3.2a shows the life missions of the system composed of the primary component and one redundant component. The system mission is ensured by a primary subsystem (component) whereas the redundant component is in a non-operational state called the "*stand-by*" state. We can distinguish two kinds of passive redundant spares in a "*stand-by*" state:

- *Cold* spare.
- *Hot* spare.

A component labeled as *cold* spare is powered-off and this state is changed when it is needed. Hot spare is powered-on. Thus, this component should be able to take over the function of the primary redundant component at any time after its initialization.

The passive redundant component starts operate at the moment when the primary component failed. This redundancy is often conditioned by component initialization (in the worst case by the system restart). This condition could be a disadvantage of this redundant approach because some critical control systems can not be reinitialized or restarted when a primary component failure has occurred. In addition, the system which composes of passive redundant components has to encapsulate some *fault detection* technique (supervision) which allows determining fault state of the primary passive redundant component in order to activate (turn-on) its passive replication. Fault detection could be executed by the redundant component during its "*hot stand-by"* state. In case of "*cold stand-by*" state, some independent supervision module has to be used in order to activate cold spare. Thus, in all cases the *fault detection* mechanism has to be implemented.

3.1.2 Active redundant components

To prevent the problem of the system re-initialization, the active redundancy could be applied. Active redundant components are still in operational mode together with primary component. Figure 3.2b shows two cases of possible operational modes of the system with active redundant components. The first case shows the situation when the primary component failed whereas the redundant component continues in working. The second case shows the situation when the redundant component failed at first and the primary component is still in operational mode and accomplishes its mission. In order to be sure that one of the components failed, several techniques exist to identify the state of the components. Generally, diagnosis techniques are used to recognize failure states of the active redundant component.

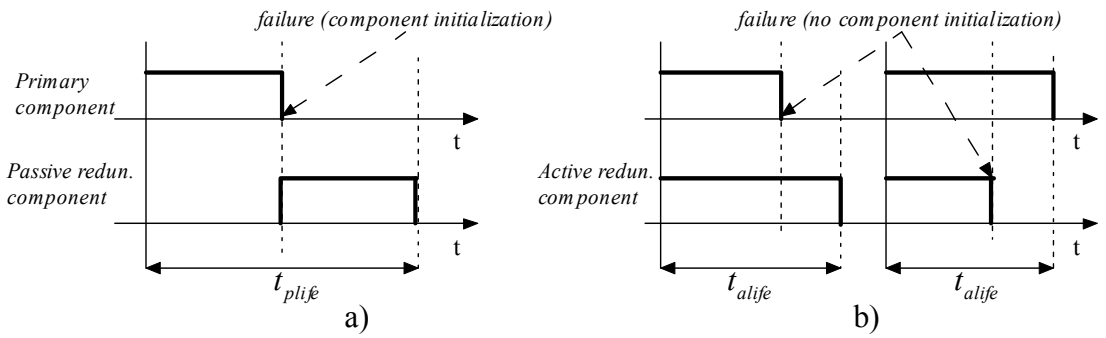


Fig. 3.2 Operating principle of the a) passive and b) active redundancy

For example, in the case of sensors, an often used technique consists in comparing the outputs given by three sensors (one primary and two active replications) in order to find the sensor with the largest error and determine its failure. This technique is called *Triple Modular Redundancy* or *N – Modular Redundancy* when more than three components are used.

3.1.3 Discussion on passive and active redundancies

As an example, a system composed of two components with the same constant failure rate λ=0.001 is simulated in order to compare the reliability curves of the system with passive and active redundant spare. In figure 3.3 both curves are compared with the curve of the same system without redundancy (long-dashed curve). Together with curves are shown *MTTF* (Means Time To First Failure) values defined in simulation *time units – Tu*. For example, the time unit *Tu* can represent one sampling period of the system or longer time intervals. We can see that life time of the system with passive redundant components (in figure 3.2 t_{plife}) is longer than life time of the system with active redundancy (in figure 3.2 t_{alife}). However, the mentioned problem with component initialization could decrease the number of application where passive redundancy could be implemented. *MTTF* of the simple system without redundancy is approximately three times less than system with active redundant component (t_{alife} = 1500 *Tu*) and four time less than system with passive redundant component (t_{plife} = 1994 *Tu*).

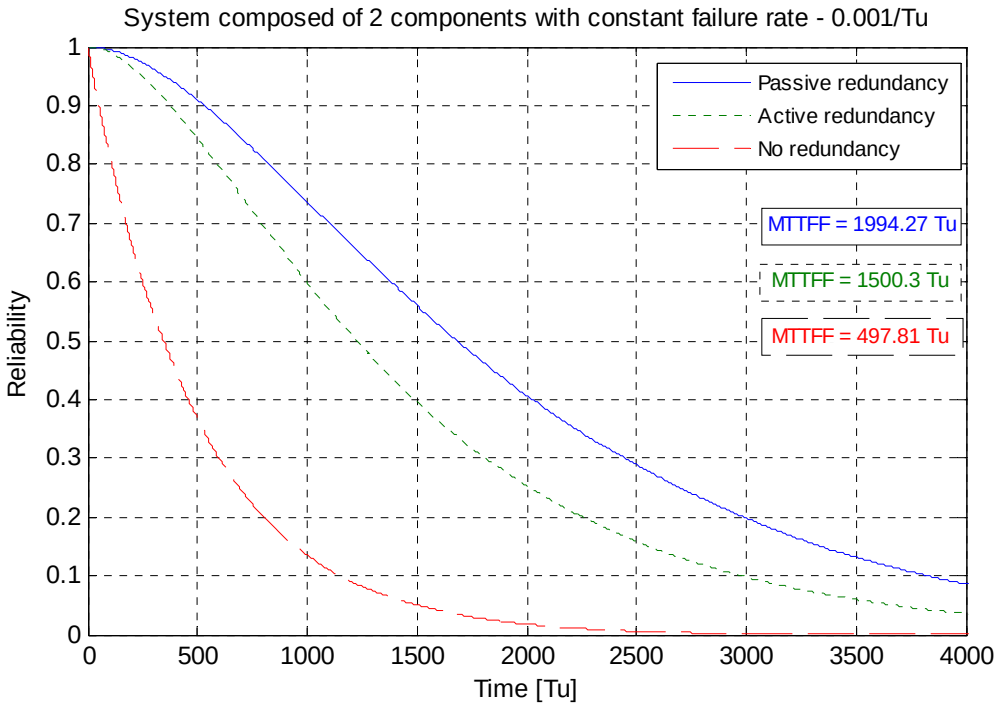


Fig. 3.3 Reliability curves with *MTTF* values of the two components system

As we can see system with two independent components (no redundancy) has a reliability curve with the common exponential shape. The curves which represent the system with redundant components have a different reliability behavior. As we can see the reliability at the beginning of the system mission much more decreases slower than in case of the systems without redundancy (red long-dashed curve). As it was mentioned above, there is a significant difference on behalf of passive redundancy, but implementing this type of redundancy could be unacceptable within the critical systems where component initialization or restart is needed.

With combination of both basic approaches based on components spares, it is possible to benefit from its advantages. In case of passive redundancy, there is a longer life time as well as there is no component's initialization problem in active redundancy. This approach is called *hybrid redundancy*. Of course, the implementation of this kind of redundant structure within the designed control system has to be allowed by its architecture and functionality. As it was abovementioned, in critical systems, passive spares can not be considered for its disadvantages. However, the system could be divided into several subsystems with different levels of criticality, in the similar meaning as SIL for defining level of the security of the components, subsystems and systems. Thus, to determine parts of the entire system where passive redundant

approach could be used and other parts of the system where using of this kind of spare is impossible or partially possible under some conditions.

3.2 Shared redundancy – quasi-redundant subsystems

Specific kind of redundant subsystems which have similar features as active redundancy however gives us some additional advantages which are introduced in further text. This kind of spares represents another type of redundant components which are not primary determined as redundant but they are able to replace some other subsystems if it is urgently required. This type of redundancy is referred as *shared redundancy* [Wys 04] or *quasi-redundancy* [Gal 08]. Due to its important advantages, it is useful to describe this kind of spares in order to show several non-considered and non-evaluated dependencies which could have an influence to the system reliability. Identification and description of this influence should not be ignored in order to obtain relevant results of the reliability estimation of the systems which involve this kind of spares.

The shared redundancy approach involves the problematic of a *Load Sharing* [Beb 07]. Thus, some of the components take part of the load of the failed components in order to let the system in functional mode. Consideration of the load sharing in mechanical components is presented by Pozsgai and others in [Poz 03]. Pozsgai and others analyze this type of systems and offer a mathematical formalism for simple systems: 1-out-of-2 and 1-out-of-3. Also there are some mathematical studies [Beb 07] of several phenomena appeared on this field of the research. Bebbington and others in [Beb 07] analyze several parameter of the system such as survival probability of the load shared subsystems.

3.2.1 An introduction to shared redundancy

As it was abovementioned, the *shared redundancy* (SR) mentioned by Wysocki and others in [Wys 04] is in further text taken into account in the same meaning as a *quasi-redundant* (QR) component. Thus, quasi-redundant components are the parts of the system which follow their primary mission when the entire system is in functional state. However, when some parts of the system fail then its function could be replaced by another part which follows the same or similar mission, thus by quasi-redundant part. The quasi-redundant components are not primary determined as active redundant subsystem because each one has its own mission which must be accomplished. Only in

case of failure it could be used. In NCS appears the question of logical reconfiguration of the system when the data flow must be changed in order to replace the functionality of a subsystem by another one. For example, some new node will lose the network connection and system has to avoid the state when packets are sent to node which does not exist. Thus, the main features of the shared redundancy could be summarized as follows:

"Quasi-redundant component is not considered as primary redundant component such as the active or the passive redundant components."

Generally in the networked control systems, three kinds of quasi-redundant components (subsystems) could be considered:

- QR controllers.
- QR networks.
- QR sensors.

Hence, a necessary condition but not sufficient condition is that a control structure where SR could be considered has to be composed of at least of two abovementioned subsystems (controllers, networks, actuators). The subsystems should have similar functionality or construction in order to be able to replace the mission of another component. In case of quasi-redundant components there are several limitations. In order to benefit of quasi-redundant networks, it is necessary to connect all nodes in all considered QR networks. Thus, in case of different networks the components should have implemented all necessary communication interfaces. In case of QR controllers, the hardware performance has to allow implementing more than one control task.

Third mentioned components are sensors. Consideration of the sensors as QR components has important physical limitations. In order to be able to replace a sensor for measuring a physical value X by another one for measuring Y it is necessary to use "multi-functional" smart sensors. We can suppose that for some combination of the measured physical values can not be constructed.

Other limitation is the distance between failed sensor and its QR sensor which could has significant influence to the possibility of its replacing. Generally, implementation of the QR sensors within control system structure could be more difficult than application of the SR approach on controllers or networks.

In order to offer a good description of the SR / quasi-redundant components several control structures will be briefly introduced.

There are naturally suitable control structures which could implement the shared redundancy approach without other modifications. Two well known control structures can be considered:

- *cascade* control structure (Fig. 3.4a)
- *feed-forward* control structure (Fig. 3.4b)

Both control structures (Fig. 3.4) are composed of two sensors, two controllers and one actuator [Bro 02]. Hence, we can consider at least two similar components which could be used as quasi-spare themselves. As we can see in figure 3.4 the control structures are different and the functionality from control point of view has a different philosophy. However from the dependability analysis point of view, their structures are similar according to the number of components as well as the functionality of the components. Thus, in further text we will deal with cascade control structure and its networked implementation as a networked cascade control system (NCCS). This structure was chosen because it is closer to real mini-helicopter application introduced and analyzed in chapter **5**. Hence, a brief description of the cascade control structure is further presented.

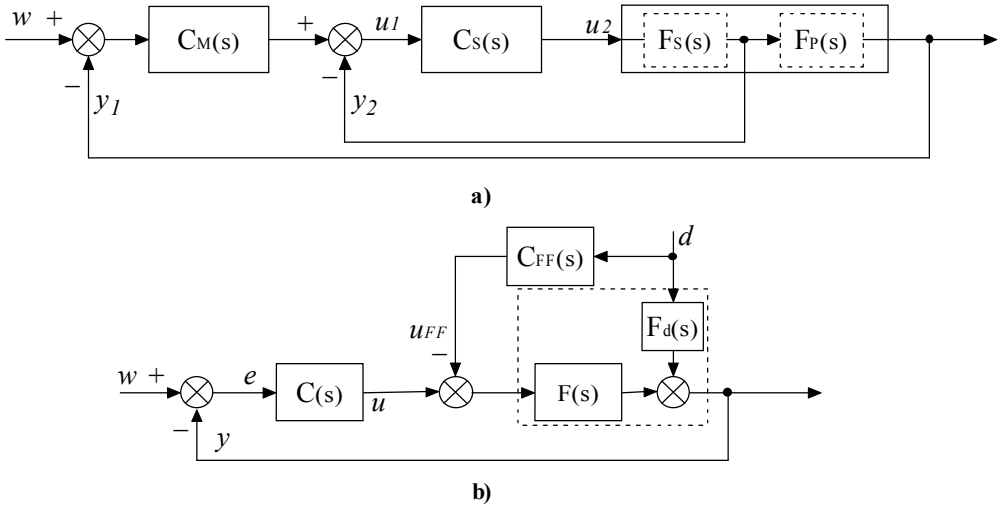


Fig. 3.4 a) Cascade control; b) Feed-forward control

Using the cascade control is possible in these cases [Bro 02]:

- Controlled system must contain a subsystem (secondary subsystem) that directly affect to a primary system.
- The gain of the secondary subsystem (encapsulated in the actuator) is non-linear.

- The disturbance is directly measured.

Usually, it is supposed that the secondary subsystem $F_S(s)$ has faster dynamics than the primary subsystem $F_P(s)$. This condition is not necessary [Bro 02], but some modifications of conventional cascade structure (Fig. 3.4a) and control laws must be provided if it is not fulfilled.

As we can see from the described conditions, the system has to contain the secondary subsystem $F_S(s)$ that affect directly the primary subsystem $F_P(s)$. In the cascade control structure there are inner and outer control loops. The inner loop is realized by a secondary controller (C_S – *Slave*) and the secondary subsystem $F_S(s)$. The main function of this control loop is to eliminate disturbances through the secondary measured state value with faster dynamics. The outer loop encapsulates inner loop with the primary controller (C_M – *Master*) and the primary process $F_P(s)$. The often example is heat exchanger where the temperature is controlled. There is the temperature as primary controlled value and a pressure as a secondary value with faster dynamics and direct influence to the primary controlled temperature.

3.2.1.1 Quasi-redundant controllers

In previous text, the several suitable control structures were briefly introduced. As was shown the controllers covered by these structures could be considered as quasi-redundant components by default. Thus, the hardware of both components could be shared in order to implement a shared redundant approach.

Suppose the networked cascade control system shown in figure 3.5. The system is composed of five main components (Sensor S_1, S_2, controllers C_1, C_2 and actuator A) and one network. The communication flow among components is determined by its cascade control structure. Thus, sensor S_1 sends a measured value to the controller C_1 (*Master*), the controller C_2 (*Slave*) receives the values from the sensor S_2 as well as the controller C_1 in order to compute an actuating value for the actuator A. Each part of the system (components and network) represents independent subsystems. However, when quasi-redundant components are considered the system is not already composed of independent components. Depending on the performance parameters of the used hardware equipment in the control loop, a specific influence on the system reliability should be taken into account. Thus some dependencies should not be ignored in the dependability analysis. In the NCCS shown in figure 3.5 we could consider controllers

C_1 and C_2 as quasi redundant subsystems (components). Both QR controllers have the primary mission which must be followed. Thus, controller C_1 controls outer control loop and controller C_2 stabilizes inner control loop.

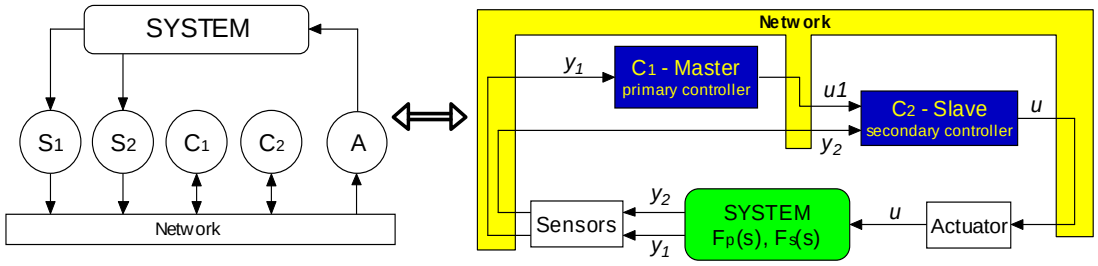


Fig. 3.5 NCCS with one network

As was abovementioned, the controllers follow their primary mission stabilization or performance optimization of the controlled system. Therefore, in regard to the similar hardware it allows sharing the computing capacity and executes different tasks. Thus, in order to implement SR approach both controllers *have to encapsulate both control tasks* for the outer and the inner control loop (see the cascade control structure in figure 3.4a).

In non-failure mode the primary task is executed in both controllers. However, in case of controller's failure (Master or Slave) the non-failed controller starts execute both tasks and computes actuating value for outer as well as the inner control loop. In this case we can suppose two scenarios (Fig. 3.6).

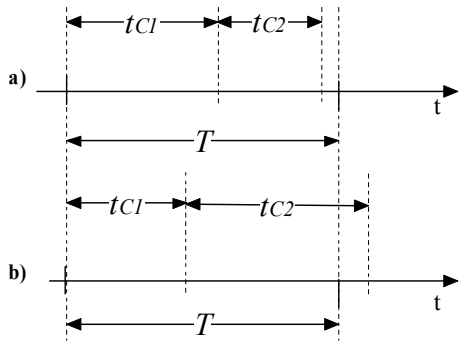


Fig. 3.6 Possible scenarios for quasi-redundant controllers

The first one supposes that the controller is able to execute all the necessary tasks within the required sample periods (Fig. 3.6a). Thus, no delays or other undesirable consequences are expected. In this case the behavior of the quasi-redundant component

is similar as in case of the active redundant components. Thus, in case of the failure of one of the components, the second takes care about its mission until its failure.

Figure 3.6b shows a second case when time to execute both necessary tasks is grater than the required sampling period. Thus, the controller will cause the delays which have significant influence to the system stability [Gal 07], [Lig 04]. Therefore, this delay could be known that allows its partially compensating by using several methods [Nic 97]. Thus, we can suppose that system destabilization will not occur immediately after the first delay and we are able to compensate it for some time interval. Thus, quasi-redundant controller does not fail immediately but its reliability decreased. There are several situations when this scenario could be considered. In critical systems where failure of an important component could cause undesired damage or other dangerous consequences the shared redundancy approach could helps to allocate some time interval in order to take the system in a safe state. Thus, SR approach can be a significant technique how to save the system before damage.

3.2.1.2 Quasi-redundant networks

The second part of the NCS which could be taken into account like SR subsystem are networks. Suppose the system with two networks (Fig. 3.7) where all components could communicate (connect) on these networks (N_1 and N_2) if is it needed. In this case we can apply SR approach on this system.

Considered functionality of the quasi redundant networks are as follows. Both networks transmit required data - network N_1 transmit data from S_1 to C_1 and from C_1 to C_2 such as network N_2 from S_2 to C_2 and from C_2 to A. Thus both networks are active and allocated during the system mission. The same as in case of QR controllers, when one network is failed the second one can take its load after system reconfiguration. Thus, all required data are sent through the second network. Hence, two similar scenarios as with controller task execution could be described. The amount of transmitted data on network with specified bit rate has logically influence on the probability of failure of the network (of course this depends on the network type and other parameters mentioned in chapter 1). This influence could be ignored when network performance parameters are sufficient. However, we can suppose that probability of network failure is decreasing simultaneously with increasing network loading.

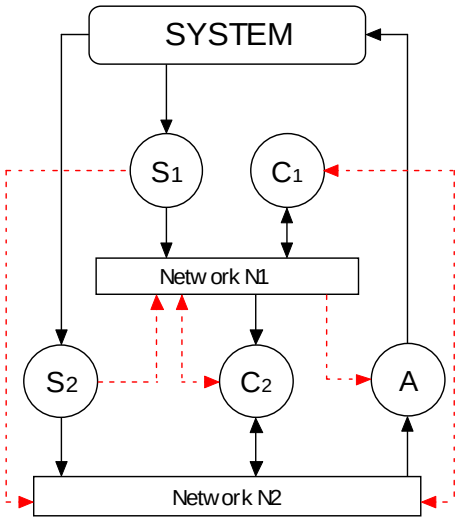


Fig. 3.7 NCCS with two *networks* and alternative network connections

The characteristic between network loading and its bit rate depends on the network type and have to be measured in real network conditions in order to determine the type of dependency – linear or nonlinear.

Not only the network bit rate can be important however other network limitations such as maximal number of nodes connected to network, etc. All limits of the QR subsystems can create dependencies with direct influence on the system reliability. Primary, we could consider these dependencies as undesirable but in case of critical failures this SR approach gives some time to save the system.

When NCS with SR approach are analyzed this characteristic should be included in prepared model and further evaluated in order to determine its influence to the reliability of the whole NCS.

3.2.2 Description of specific dependencies given by SR application

From the previous parts we can see that *shared redundancy* approach has several advantages in comparison with common redundant approaches. For example, the minimizing of the numbers of components – only the components needed for right system function are necessary. On the other side, some dependencies, which are not considered in common redundant systems (*active* or *passive*), appear. When certain dependencies are ignored we could regard on the control system with QR components as control structure with the active redundant components. However, there are several important scenarios when the reliability of the system could be decreased in order to prevent dangerous consequences or other undesirable events.

These scenarios could appear when some conditions could not be fulfilled (insufficient execution time or network bit rate) but the system need some time in order to take a safe state. Hence, it is necessary to identify and describe the influence of these dependencies which leads to more relevant results of the dependability analysis. Thus, in order to prevent from too pessimistic or too optimistic results of the dependability analysis of the considered systems, it is necessary to known the influence of the specific dependencies on the reliability behavior. The dependencies could be distinguished as follows:

- active redundant dependency,
- single step change of the nominal failure rate λ_n - increased once by constant value – step load change,
- time depend change of the nominal failure rate λ_n - functional dependency –the load of the subsystem is changed with time passed from speared subsystem failure,
    - linear,
    - nonlinear.

We suppose the presumption that destabilization of the system does not occur immediately after the first delay on the network caused by insufficient controller's hardware or network's parameters. Thus, quasi-redundant controller does not fail immediately but in this case its failure rate increases which correspond consequently to a decreased reliability.

Thus, in case of the active redundant dependency we suppose that quasi-redundant subsystem has sufficient capacities in order to follow its primary mission as well as the mission of the failed subsystem (or subsystems).

Single step change of the nominal failure rate of the subsystem is considered in case of subsystems where the failure rate of the quasi-redundant subsystem is changed (increased) once by constant value (Fig. 3.8) during its life time. Thus, the new increased failure rate λ' remains constant during further life time of the subsystem. For example, let's suppose a NCS with two Ethernet networks where one of them failed and consequently the system is reconfigured and all nodes (components) start to communicate through the non-failed network which has a sufficient bit rate capacity in order to transmit all required data. However, the amount of data has been increased

which consequently increases the probability of packets' collisions. Thus, probability of the failure (failure rate) has been increased up to new value λ'.

A third case considers the change of the nominal failure rate λ_n which depends on the time passed from the moment of the failure until current time of the working of the quasi-redundant subsystem which encapsulates the executing necessary tasks (own tasks as well as tasks of the failed subsystem). Thus, a functional dependency has to be considered. This dependency of the change of the failure rate λ_n could be described by linear or nonlinear dependency / function. We could take previous example of the system with two networks. However, in this case the bit rate of the second (non-failed) network is not sufficient. Consequently delays in data transmission as well as other consequential undesirable problems such as system destabilization might be caused. We can suppose that the non-failed network will fail in some time. Thus, the nominal failure rate λ_n of the second network is now time dependent and is linearly or nonlinearly increased until the system failure. Mentioned examples with related equations are further discussed in more details.

Suppose that reliability of the system R(t) is characterized by a nominal failure rate λn. Suppose a system with two subsystems S_1 and S_2 (such as networks in abovementioned examples) whereas the subsystem S1 will fail at first and then quasi-redundant subsystem S2 will follow both missions (S_1 and S_2). In figure 3.8 are shown two above mentioned scenarios when the nominal failure rate λn of the subsystem increased by a constant value or by a value which could be described as linear or nonlinear function (functional dependencies).

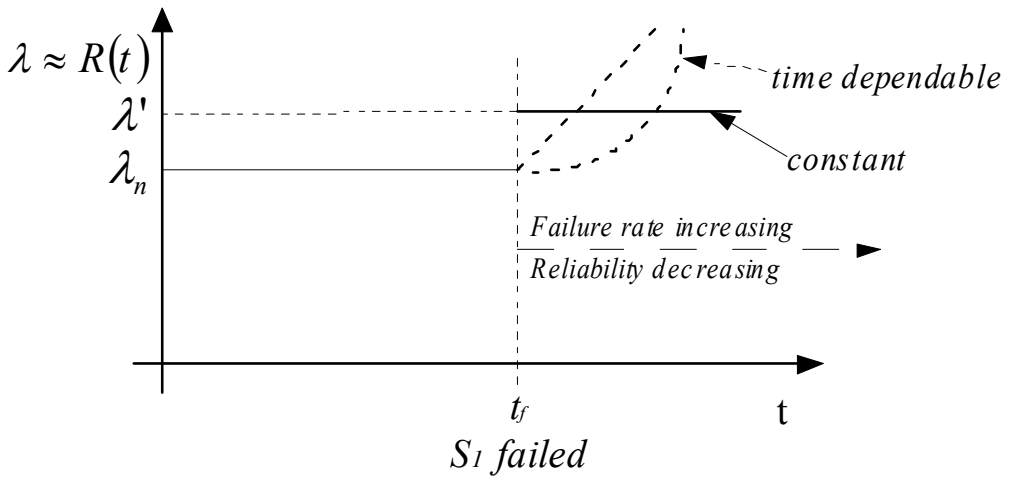


Fig. 3.8 Possible failure rate curves for subsystem S_2 during its mission

Single step change of the failure rate λ_n (only one time) by constant value (see Fig. 3.8) will be dealt. It corresponds with the reliability reduction of the quasi-redundant subsystem S_2 by increasing the failure rate, during its mission, from its nominal value λ_n up to new λ'. Consequently, the system will follow its primary mission thanks to QR subsystem S_2 but its failure rate is already increased and consequently the probability of failure of S_2 is higher. The difference between nominal λ_n and increased λ' failure rate will be called *decrease factor* d_R. Thus, mentioned constant value is characterized by decrease factor d_R of QR subsystem and new changed failure rate λ' at fail time t_f is given by followed equation:

$$\lambda' = \lambda_n + d_R . \tag{3.2}$$

Failure rate increases only one time by the specified value and QR subsystem S_2 with new constant failure rate λ' will follow both mission its own mission and mission of the failed subsystem S_1.

The second case shown in figure 3.8 considers reliability reduction where the failure rate λ_n is increased during the working of the subsystem S_2 by specified *decrease factor*. This change of the nominal failure rate depends on time whereas with time extending the failure rate of the S_2 is got near to 1 (system failed). Thus, a *decrease function* $f_{dr}(t)$ is represented by linear or nonlinear characteristic and depends on real subsystem which is considered as quasi-redundant. Thus, increased failure rate λ' of the subsystem S_2 depends on time t and is given by the following equation:

$$\lambda'(t_f) = \lambda_n + f_{dR}(t), \tag{3.3}$$

where t_f is the failure time of the subsystem when QR subsystem starts to execute another additional task.

As it was mentioned, the decrease function $f_{dr}(t)$ can be represented by a simple linear function, for example,

$$f_{dR}(t) = d_R * (t - t_f), \tag{3.4}$$

or nonlinear exponential function

$$f_{dR}(t) = e^{d_R(t-t_f)}, \tag{3.5}$$

or other types depending on the dependency of specified QR subsystems (networks , controllers etc.).

There are some subsystems where the failure rate could increase however decrease depending on some factors which have influence to this behavior. For example,

networks in NCS could be considered as a QR subsystem where the failure rate is changed in both directions (increased or decreased) depending on network loading or other factors. Thus, we can consider a *dynamic reliability*. To estimate this change of the reliability in real time without high time severity it is necessary to propose a relatively simple and effective method. Fast reliability estimation of the system whose parameters are changed during its mission is very important especially in the critical systems with high dynamics.

The influence of the constant and functional change (linear or nonlinear) to the final reliability of the quasi-redundant systems will be dealt in the next chapter. Based on parameters of the decrease function $f_{dr}(t)$ will be described different influences to reliability behavior of the studied NCS. Based on these result will be introduced an effective method for dependability analysis of the NCS which cover SR approach in order to avoid significant distortion of the obtained results in the positive as well as negative sense.

3.2.3 Advantages of using shared redundancy approach

Shared redundancy approach in networked control systems' design could have important influence to benefit from all its parts in order to achieve a maximal dependability level with a maximal efficiency, using hardware and software resources given by its basic control structure. As it was described above, the quasi-redundant components could be considered as active redundancy. On the other side there are several specific features which should be considered in further dependability analysis. Based on description given above we can summarize the most important advantages given by the implementation of this kind of redundancy within a designed networked control system as follows:

- The system is composed only of necessary components (parts) for following the primary mission of the system whereas higher system reliability is ensured without using any additional active redundant components.
- Following the first point we could suppose less number of components used for saving the control mission. Thus, economic aspect could be very significant.
- Prevention of the system's critical failure when QR subsystem has no sufficient hardware capacities.

The first point shows that only basic control structure is considered as necessary to follow a control function of the NCS. Thus, no additional components are used in order to increase the reliability level of the designed system. And in more complex NCS, its hardware complexity does not increased but it remains still the same.

Based on this presumption is logically supposed a lower economic severity. This aspect can be most important in case of dependability parameters of the critical systems, where each sampling period can help to save the system in critical moment when some important parts are failed.

3.3 Modeling the QR structures

As we discuss in the first chapter, it is difficult to represent the systems with dynamic dependencies such as NCCS with redundant subsystems, by simple common known static tools.

The classical approach for reliability evaluation of system with two redundant controllers shown in figure 3.5 could be represented by a simple reliability model created by using simple static tool (reliability block diagrams) shown in figure 3.9.

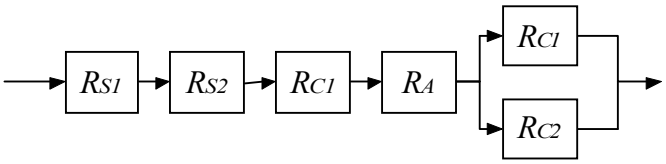


Fig. 3.9 Reliability model (using reliability bloc diagram) of the NCCS

Presented diagram suppose that all system components / parts are independent and controller C_2 is redundant (two parallel blocks – R_{C1} and R_{C2}). As was mentioned above, we can suppose that the exactness of this model is not sufficient in conditions of SR approach due to ignoring several important features of the SR systems. This presumption will be confirmed by the work presented in the next chapter.

As was mentioned, further models of the studied NCS and dependencies within them are prepared using Petri Nets tool whereas the dependability parameters and attributes are evaluated by Monte Carlo simulation (MCS) method. In order to be able to include all necessary dependencies, the Petri Nets were chosen.

3.3.1 Coloured Petri Nets

In the first chapter, several type of Petri Nets (PN) were mentioned and briefly presented. Such as Generalized PN (GPN), Coloured PN, Stochastic PN (SPN) etc. Coloured PN are a type of PN which can significantly reduce the complexity of the models created by using GPN.

The most used for modeling the dependability of dynamic systems are SPN. However, there are number of studies which use Coloured PN for dependability analysis [Bar 03], [Son 03].

This type of PN is supported by CPN Tools software which involves programming with ML language [Ull 98]. In combination with this programming language it offers a wide field of application in order to prepare needed models for dependability analysis.

3.3.2 Modeling in CPN Tools

Further presented PN models are prepared by using CPN Tools software. Hence, several features are presented in order to well explain the functionality of the basic parts of the PN implemented in CPN Tools. All further mentioned features are used in model developing.

The basic elements of the Coloured PN are places, transitions and arcs. In addition, CPN offers to declare different variables of several basic types (*colour sets*) such as *integer*, *boolean*, *string*, etc. By using basic types it is possible to create compounded colour sets such as *list*, *product*, *record*, etc.

CPN Tools have the possibility to create timed PN, too. In order to prepare timed model it is necessary to declare timed color set

<< **colset** name = *already_declared_colset* **timed**; >>.

All declaration and other features use the ML programming language syntax [Ull 98]. However, there are several constraints in CPN Tools in comparison with possibilities of the standard ML language presented by Ullman in [Ull 98]. For example, CPN Tools does not support *real* color set in order to declare variables which represent real number. On the other side, there are several basic supported functions such as function for different types of distributions (uniform, exponential) which work with *real* type parameters. Thus, real colour set can not be declared but real constant can be used as parameter of the function.

As was mentioned, there is possibility of declaring compounded types as well as variables of different types. In addition, different own function can be declared in order

to prepare more transparent and structured model. Several basic functions used in further models will be presented in further text.

Transparency of the model can be improved by preparing hierarchical models. The main principle is that a transition can be associated with a sub-page where the detailed functionality of the transition is prepared. For example a controller with an input and an output buffer which connect controller with the network. Thus, hierarchical model composes of root page (Fig. 1.10) and nested sub-pages.

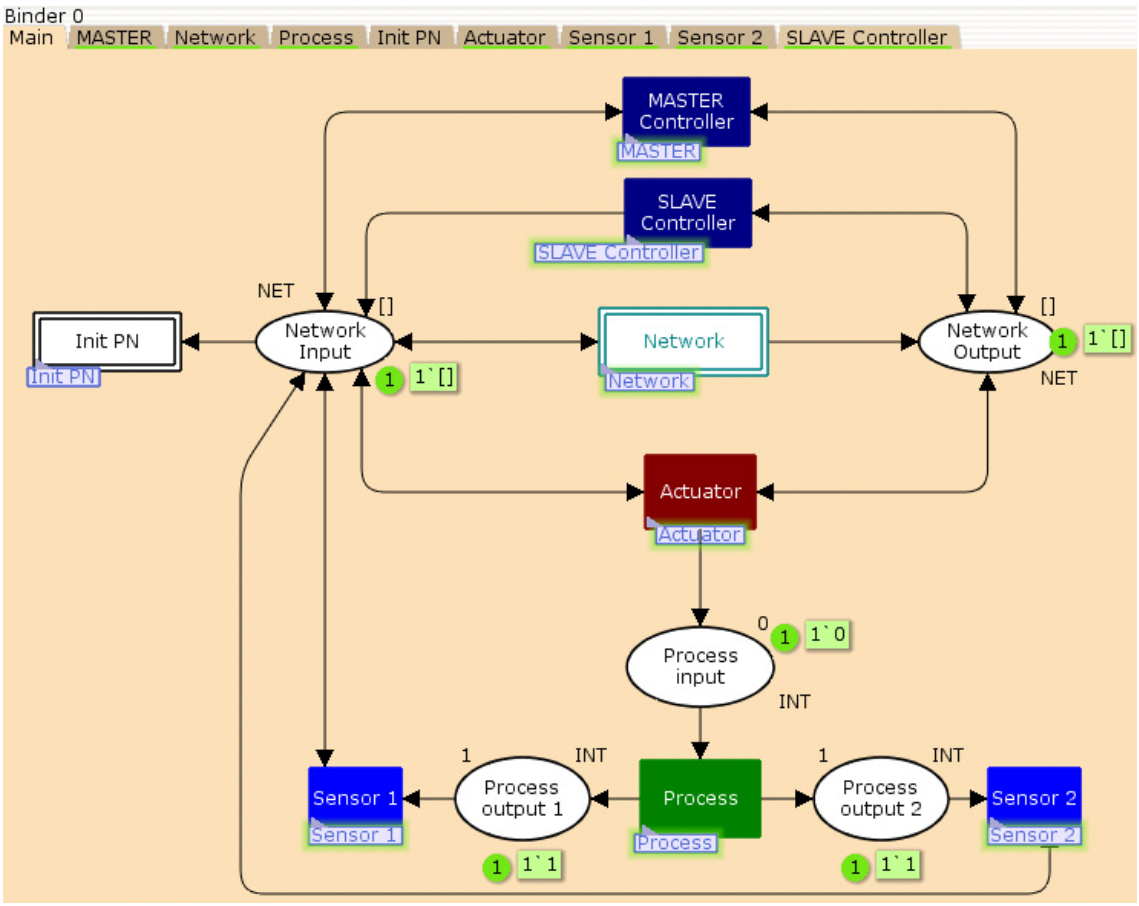


Fig. 3.10 Root page "Main" of the hierarchical cascade NCS model with one network

In addition performance analysis, state space evaluation, simulation replication and data collector monitors are implemented. Within performance analysis it is possible to obtain several statistical values (sum, average, variance, standard deviation, etc.) by using data collectors. The monitors help collect the data for example when multiple simulations are executed. Thus, during the simulation it extracts numerical data from a PN model. There is the possibility of creating the reports of the PN's variable evaluation during each simulation. For example, the reports about order of transition firing,

variables values in each step of the simulation. State space analysis is primarly determined for analyzing the states of the deterministic PN.

3.3.2.1 Place

Petri Nets are graphical tools hence a place in PN is represented by a circle which has three basic features as (see Fig 3.11):

- Name,
- Colour set.
- Initial marking.

Name is the identifier of the place in the entire model. Colour set defines the type of the tokens which can be received and stored in the place. Initial marking represent the initial state of the model. The name and the initial marking are unnecessary parameters. However, in order to generate a *graph of occurrence* the name of the place has to be specified.

If the place marking corresponds with place's colour set then marking is shown on the side of the place (in figure 3.10 see place "*Network Input*" with empty list *[]* as initial marking).

Additional feature which could be added to the place is possible in hierarchical models. There are recognized *input*, *output* and *input*/*output* places (Fig 3.11). Input place represents the place where sub-page receives the tokens from hierarchically higher page. On the contrary, the output place represents output point of the sub-page up to hierarchically higher page. The input/output place combines both functionalities.

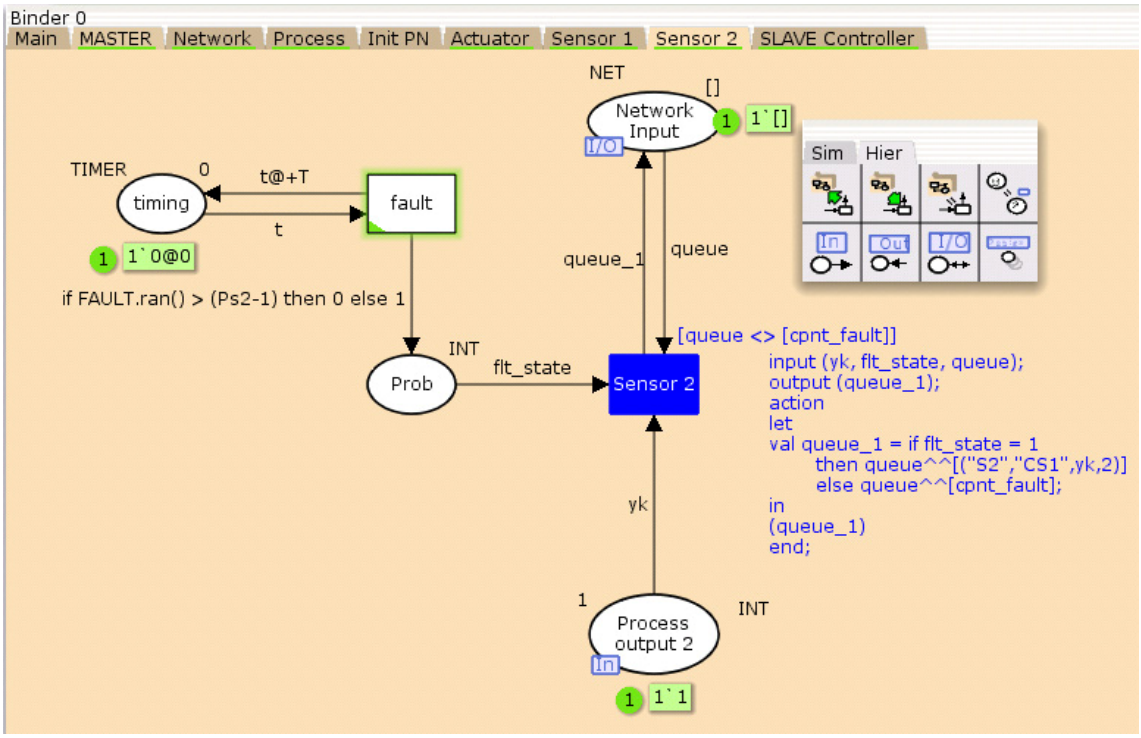


Fig. 3.11 Example of the different places, transition and arcs in CPN Tools

In figure 3.11 there are several examples of different places. The place "*timing*" is associated with the timed colour set *TIMER* and has initial marking *0*. The places "*Network Input*" and "Process output 2" represents *I/O* and *In* places which connect the sub-page "*Sensor2*" with "*Main*" page and they are the part of the root page "*Main*" of the hierarchical model. The places "*timing*" and "*Prob*" are parts of the sub-model "*Sensor2*" which is on "*Main*" page represented as transition "*Sensor2*". These places do not appear on the root page "*Main*".

3.3.2.2 Transition

The basic functionality of the transition is described by terms activation and firing. The transition is activated when the required number of tokens in input places is achieved. When the transition is fired the output places receive the required number of tokens. The transitions, represented by rectangles, have the followed features related:

- Name.
- Guard code.
- Code segment.
- Timing.

The same as in the case of places, the name is an identifier of the transition. Guard code of the transition is determined for defining some conditions of firing of the transition. The condition in the guard part bounded by brackets can be composed of several operators such as *andalso* or *orelse*. In figure 3.11 there is a simple guard condition *[queue <> [cpnt_fault]]* in transition "*Sensor 2*". This condition allows fire the transition only if the list of packets *queue* on the network is not equal to the list *[cpnt_fault]*, which represents component's failure.

To each transition can be associated a code segment where actions which have to be executed during transition firing are programmed. The basic construction of the code segment compose of line where all input variables are specified (input(…)), the line where all output variables are specified (output(…)), action part where all needed actions (computing) are written (programmed) and end part where all output variables which has been changed in action part have to be named.

Last feature of the transition offers to create the timing of the PN on transition. Timing has to be written in following syntax "@ + *T*" where *T* represents the number of simulation in time units to the next firing. There are two ways how to prepare timed PN in CPN Tools. First is using the transitions (already presented) and the second technique is using output arcs of the transition connected to the place which has associated timed colour set.

3.3.2.3 Arcs

The third element of the PN is arc. PN are defined as a cyclic bipartite graph hence the arc is graphically represented by line with the arrow on one or both sides. Arcs is usually associated with variable of the same type (colour set) as the colour set of the place which connects with the transition.

The arc has not to be associated only with simple variable. There are several possibilities. The output arc can be associated with a condition with the following syntax

"**if** <condition> **then** *value* **else** *value*".

The type of values has to correspond with the colour set of the output place which will receive this token (value). The values can be generated by different actions such as concatenation of two lists (the output value is list)

"**if** FAULT.ran() > (proba-1) **then** queue^^[CP1_fault] **else** queue",

or by using the functions, etc.

As was abovementioned the arcs can be used for preparing timed PN model. In this case the output arc of the transition have to be associated with timed variable as we can see in figure 1.11 between transition "*fault*" and place "*timing*" – "*t@ + T*" where *t* is the timed variable and *T* represents the number of simulation in the time units to the next firing. As we can see the place has an initial marking *0* (integer value) and its colour set is *TIMER* which represents *timed INT* colour set.

In literature we can find an inhibitor arc which has feature that transition is activated when the number of tokens in place is strictly lower than weight of the arc. This type of arc is not supported by CPN Tools as default. However there are several ways how to create this feature or condition of transition activation.

3.3.2.4 Description of the several important functions of the prepared models

Prepared PN models compose of several important functions which make them more transparent and simpler to modify as well as ensure some specific functionality which is required when multiple simulation have to be executed. For execution of multiple simulations the CPN Tools offer the simulation replication tool. However, for our needs few simple functions were prepared. This solution was chosen in order to be able to collect the data into the file in appropriate form for further analysis in Matlab software environment.

In order to be able to execute controlled multiple simulation the function *InitializePN* is used in all prepared models.

```
fun InitializePN() =
let
val _ = writeToFile(file_name,~1);
val n_of_sim = getNumOfSim(n_sim_file) - 1;
val file_id = TextIO.openAppend(Output.getModelDir()^"/"^file_name);
val _ = if n_of_sim = 0 then
      (
       TextIO.output(file_id,"Decrease="^Int.toString(decr_val)^"; ");
       TextIO.output(file_id,"Pc1="^Int.toString(Pc1)^"; ");
       TextIO.output(file_id,"Pc2="^Int.toString(Pc2)^"; ");
       CPN'Sim.reset_sim();
       CPN'Sim.init_state()
       )
else
     (
       setNumOfSim(n_sim_file, n_of_sim);
       InitNet()
```

```
      );
in
TextIO.closeOut(file_id)
end;

fun InitNet() =
let
val _ = CPN'Sim.reset_scheduler();
val _ = CPN'Sim.reset_sim();
val _ = CPN'Sim.init_state();
val _ = CPN'Sim.run();
in
()
end;
```

This function (*InitializePN*) is called when the modeled system is failed. The function writes the fail time of the system in the file and reads and decreases the number of simulations which have to be executed. A new decreased value of the number of simulation is written into the file. If the last simulation was finished then the model is initialized but simulation is not started. If the last simulation was not executed the model is reinitialized by calling the function *InitPN* and a next simulation is started.

3.4 Conclusion

This chapter brings closer principles and the features of redundant subsystems which represents the main point of this study. Redundant subsystems are a classical possibility how to improve the reliability as well as other dependability attributes and parameters of the systems. There are several most used approaches such as the warm or the cold spares presented in the first section of this chapter. However, implementation of these kinds of spares requires additional components which have to be covered by the system. This feature brings several disadvantages like increasing the complexity and price of the system. Hence, in the second part of the chapter is presented the shared redundancy approach which could significantly eliminate mentioned disadvantages.

Implementation of a shared redundancy approach in an NCS design can improve the dependability attributes and parameters of the systems without the need to increase the complexity and so the cost of the system. Hence, a significant part of this chapter deals with this kind of spares. The SR approach is brought closer in the context of NCS (networks, controllers). Shared redundancy necessarily involves problem of the load

sharing among quasi-redundant subsystems of the NCS, where several specific features are introduced which should be considered and taken into account. These features will be studied in further chapters in order to describe their influences on the dependability of the NCS as well as in order to propose a simple time thrift method for estimation of some attributes and parameters of the dependability (reliability, *MTTF*, …) of NCS.

In order to prepare the models of the systems with QR components, Petri Nets and appropriate software package used for modeling is presented. Several basic specifications of the Coloured PN in CPN Tools are introduced for a good understanding of further presented parts of the prepared models.

4. Case study of the shared redundant subsystems with consideration of different conditions for load sharing

It is necessary to describe the shared redundant subsystems in order to show several non-considered and non-evaluated dependencies which could have a significant influence on the system dependability parameters. Identification and description of this influence should not be ignored in order to obtain relevant results of the dependability estimation.

As was mentioned before, there are several different aspects which should be considered for the dependability evaluation. For example, when networks are considered as quasi redundant subsystems, it is not necessary to take into account the physical abrasion which appears in case of the sensors or actuators. However, other aspects appear such as *decrease factor* (chapter 3.2.2). Thus, suppose that the network bite rate after system reconfiguration is sufficient to transfer all the required information among all components of the reconfigured system then the question whether it is possible to use an equation for the reliability evaluation of the passive or active redundant systems in order to well describe its dependability behavior, appears. We can consider a negative response due to ignoring some specific dependencies. Hence proposing the equations in order to evaluate the system dependability without significant error is important aspect of this work.

The concept chosen in order to bring closer the problematic of the shared redundancy approach as well as evaluating and describing the influence to the dependability parameters and attributes is given by the *decrease factor*. The evaluation of the results is based on Monte Carlo simulation method whereas evaluated sample of data is given by models of the subsystems prepared using Petri nets.

To well explain the problematic of the shared redundancy implementation in the conditions of critical systems are further explained basic redundant systems such as *Triple Modular Redundancy* (see chapter 3) with shared redundancy approach (*Triple Modular Quasi-Redundancy*). Following the description of the basic QR subsystems, more complex structures with QR parts are introduced. Finally all presented influences of the SR approach in the critical conditions are summarized.

The results of the simulations are compared with the results of the same system without redundancy and with active redundancy in order to show error of evaluation by using classical commonly known methods which cover only basic dependencies such as passive redundancy, etc. The aim of the provided simulations is the identification of the influence of the various type of *decrease factor* to some of the dependability parameters behavior. As was abovementioned one of the aims of the comparison is to present whether quasi redundancy could be represented by active or passive redundant structure (Fig. 4.9) when dependability parameters are evaluated.

4.1 Models and Monte Carlo Simulation results verification

The main aim of the this section is show the accuracy of using the Petri Nets model in combination with Monte Carlo simulations in order to well evaluate the dependability of the redundant systems.

Monte Carlo simulation (MCS) is the statistical method. Hence the result given by this method should be verified in order to determine its exactness under several specified conditions. The main parameter which has A significant influence to the exactness of this statistical method is the number of simulation n (see equations 1.23 and 1.24 in the first chapter). The simulations are compared with result given by equation 4.1 mentioned in the third chapter.

In order to show the influence of this parameter to the final exactness of the results obtained by MCS the simple system is composed of two passive redundant components. The failure rates of each component are the same and are equal to 0.001. Thus, equation for this system is as follows:

$$R_P(t) = e^{-0.001t} \sum_{i=0}^{1} \frac{(0.001t)^i}{i!} \tag{4.1}$$

The results of the simulations are shown in figure 4.1 where on the left side are whole curves of the systems given by equation (4.1) as well as by MCS with different number of the simulations $n \in \{10000, 40000\}$ for a time interval $8000Tu$, where Tu represents simulation *time unit*. This *time unit* can be labeled as one hour of the component's life time. We can see that results are almost identical. There are only small differences caused by the exactness of the MCS.

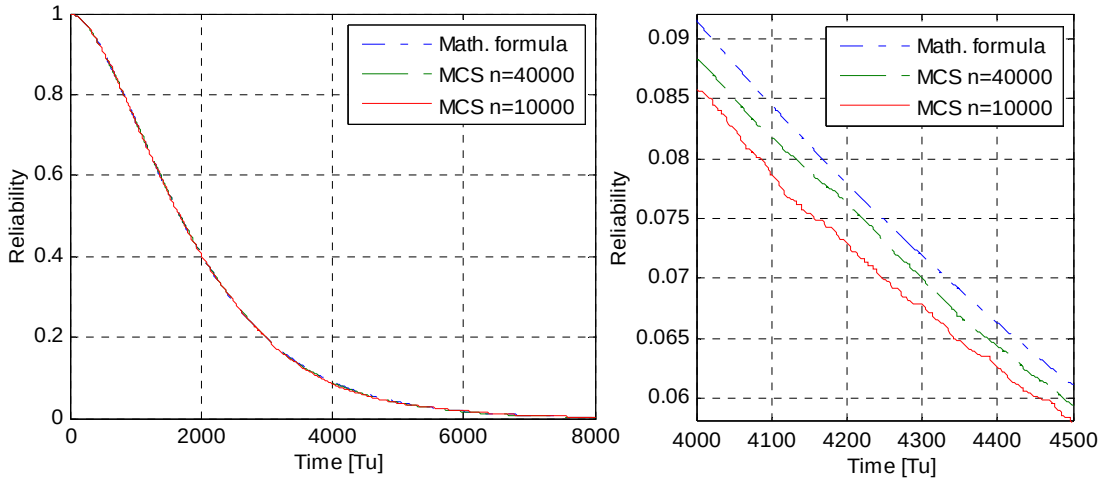


Fig. 4.1 Graphical comparison of the results given by MCS and equation for passive redundant system composed of two components with failure rate λ = 0.001

In the right sub-figure is shown the detail of the part where differences are most visible. It is within time interval from about 4000*Tu* up to 5000*Tu*. Thus, We can see that dashed curve which represent results of the MCS with 40000 simulations is closer to analytical results (point-dashed curve) given by mentioned formula (4.1) than continuous curve which represents the results of the MCS with 10000 simulations. Differences are only in values about 10^{-3} (see further evaluation of the different types of errors) but we can see that dashed curve (40000 sim.) is approximately two times closer to the reference curve given by the mathematical formula.

Several differences are shown in the table 4.1. There are different values of the reliability (MCS with 10000 and 40000 simulations and equation) of the evaluated system for the several specified time intervals from 10*Tu* up to 8000*Tu*.

Time (*Tu*)	10*Tu*	100*Tu*	500*Tu*	2000*Tu*	5000*Tu*	8000*Tu*
MCS 10 000 sim. – *R*(*t*)	0.9998	0.9944	0.9072	0.40744	0.0378	0.00292
MCS 40 000 sim. – *R*(*t*)	0.99995	0.99488	0.909	0.40598	0.0391	0.00293
Mathematical for. – *R*(*t*)	0.99995	0.99532	0.90979	0.406005	0.04043	0.00301

Table 4.1 Comparison of the reliability values given by MCS and mathematical equation

Also from the table 4.1 we can see that final reliability values obtained by MCS with higher number of simulations are closer to analytical values given by mathematical formula.

In order to propose another, more exact, comparison, the standard deviation and followed type of errors are evaluated [Spi 87]. All errors are computed for both number of the simulations $n \in \{10000,40000\}$.

Standard deviation of the obtained data by MCS is computed by following equation

$$\sigma(R) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(R_{MC\,i} - R_i\right)^2}\,, \tag{4.2}$$

where R_{MCi} is the prediction value of the MCS, R_i is the value given by equation (4.1) and n is the number of compared samples.

At first the *mean absolute error* (*MAE*) is proposed defined as

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|R_{MC\,i} - R_i\right|, \tag{4.3}$$

where, the same as in previous equation, R_{MCi} is the prediction value of the MCS and R_i is value given by mathematical equation (4.1) and n is the number of compared samples.

The *mean squared error (MSE*):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(R_{MC\,i} - R_i\right)^2\,, \tag{4.4}$$

where variables have the same mean as was mentioned above.

The *maximal absolute difference* (*MAD*) between the results of the MCS and mathematical formula is shown.

$$MAD = \max\left(\left|R_{MC\,i} - R_i\right|\right). \tag{4.5}$$

The results given by all mentioned error equations are summarized in table 4.2. We can check that error of the MCS with four time higher number of simulation gives more exact results. Thus, the results non-directly support the proposition mentioned in first chapter by formulas 1.23 and 1.24. This proposition shows that for decreasing the standard deviation of the results given by MCS to half it is necessary to increase number of the simulations four times. The relation between the standard deviation and the exactness of the result is evident which is confirmed by the table 4.2 too. As we can see from table all evaluated errors decrease significantly. The standard deviation is two times smaller which confirms the mentioned presumption given by equations 1.23 and 1.24.

Number of sim. n	n = 10 000	n = 40 000
$\sigma(R)$ (4.2)	$2*10^{-3}$	10^{-3}
MSE (4.3)	$5.3*10^{-6}$	$1.99*10^{-6}$
MAE (4.4)	$1.62*10^{-3}$	$1.18*10^{-3}$
MAD (4.5)	$6.4*10^{-3}$	$3.3*10^{-3}$

Table 4.2 Comparison of the errors of the MCS with different number of the simulation samples

Based on the result given by these simulations the further presented models are evaluated by using MCS with about 40000 simulations in order to obtain relevant results. This value could be considered as compromise between time for providing the simulation of the system necessary for obtain the results and final exactness of this simulations or evaluated results.

It is necessary to say that the number of simulation can be higher or lower depending on the failure rate of the subsystems. Let's suppose that all components of the mentioned system have smaller failure rate which leads to increase the mean time to first failure (*MTTF*) of the studied system. Thus, in order to obtain the same exactness it is necessary to use larger sample set of the failure times of the system. Otherwise, in case of higher failure rate, smaller number of the simulation could be sufficient in order to remain in the same level of exactness.

Hence, in order to be able stop the MCS with defined exactness of the obtained results the MCS with some stop criteria should be used. Ghostine in [Gho 08] introduce MCS method with defined the stop criteria through confidence interval and exactness parameter. This criterion takes into account the required exactness of the mean of the samples obtained during MCS. It is very useful in order to execute accurate number of the simulations.

Finally, we can confirm that correctness of the result given by MCS in combination with Petri Nets models is evident. Obtained results have only small differences from values obtained by analytical technique. Hence, as was abovementioned, further presented models were modeled and evaluated by Petri Nets and MCS under above mentioned condition for number of simulation n.

4.2 Two quasi-redundant components

As was mentioned above, in order to introduce a *shared redundancy* approach with its specific features, several basic systems will be further presented and evaluated. The simplest system composed of two components will be introduced in this chapter. The main reason of choosing this simple type of system is because two similar subsystems

which could be considered as quasi-redundant are often seen in many complex control systems. For example, both common used control structures mentioned in the third chapter (cascade, feed-forward) are naturally composed of at least one type of similar subsystems such as controllers. If we consider the structure shown in figure 4.7 we could find another couple of quasi-redundant subsystems such as networks.

All Petri nets (PN) models are hierarchical whereas the functionality of each component is modeled separately in a hierarchical made. In figure 4.2 is shown the main structure of the mentioned two component system. This system is composes of two transitions which represent two components (CP1 and CP2) as well as another transition which the main functionality is the initialization of the entire model after a system failure and execution of another simulation. Thus, repeating the simulation of the model until a specified number of simulations is obtained. This *Init* transition has associated the condition of firing which depends on the simulated system. Condition in this system is in fact related to its redundant feature. Thus, when both components failed then a system failure is occurred. This transition is part of each other model hence it will not be described further again. As was mentioned, only the condition of firing is different and depends on the type of modeled system.

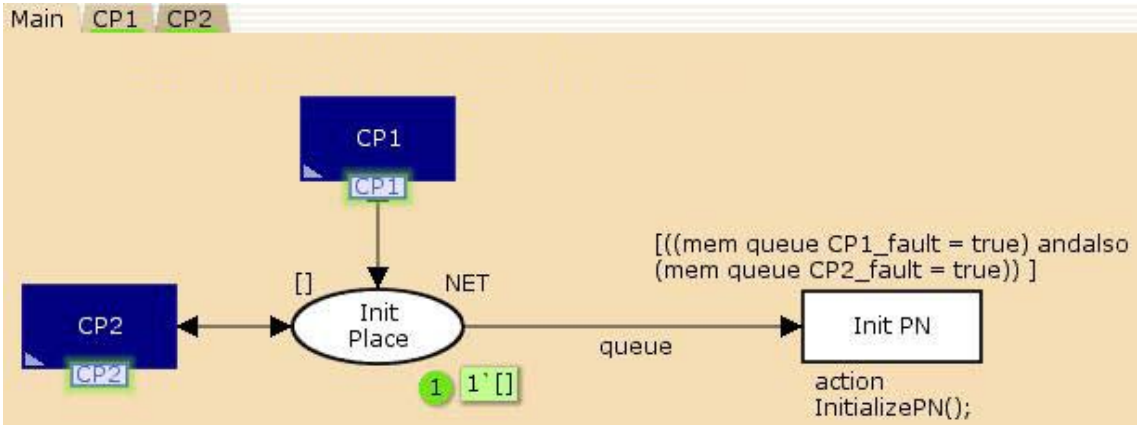


Fig. 4.2 Structure of the model composed of two quasi-redundant components CP1 and CP2

Figure 4.3 shows a detailed PN model of one of the components where the failure rate increasing follows the predefined function. As was abovementioned, it is a hierarchical model and the place "*Init Place*" represents the input/output place between global model (Fig. 4.2) and its component's sub-model (Fig. 4.3). the place "*proba*" is determined to save the value which corresponds with the probability of the component's failure which directly relates with its failure rate λ. The failure state or non-failure state

is generated by the function *decrease(...)* in transition code region. The failure state of the component is indicated by packet *CP1_fault* (one of the parameters of the function) which is sent from transition "*Component 1*" to the place "*Init Place*". All components encapsulated by this model are timed in order to be able to evaluate the dynamic behavior of the dependability parameters and attributes within a specified time interval. This timing is achieved by the output arc of the transition to the place "*timing*".

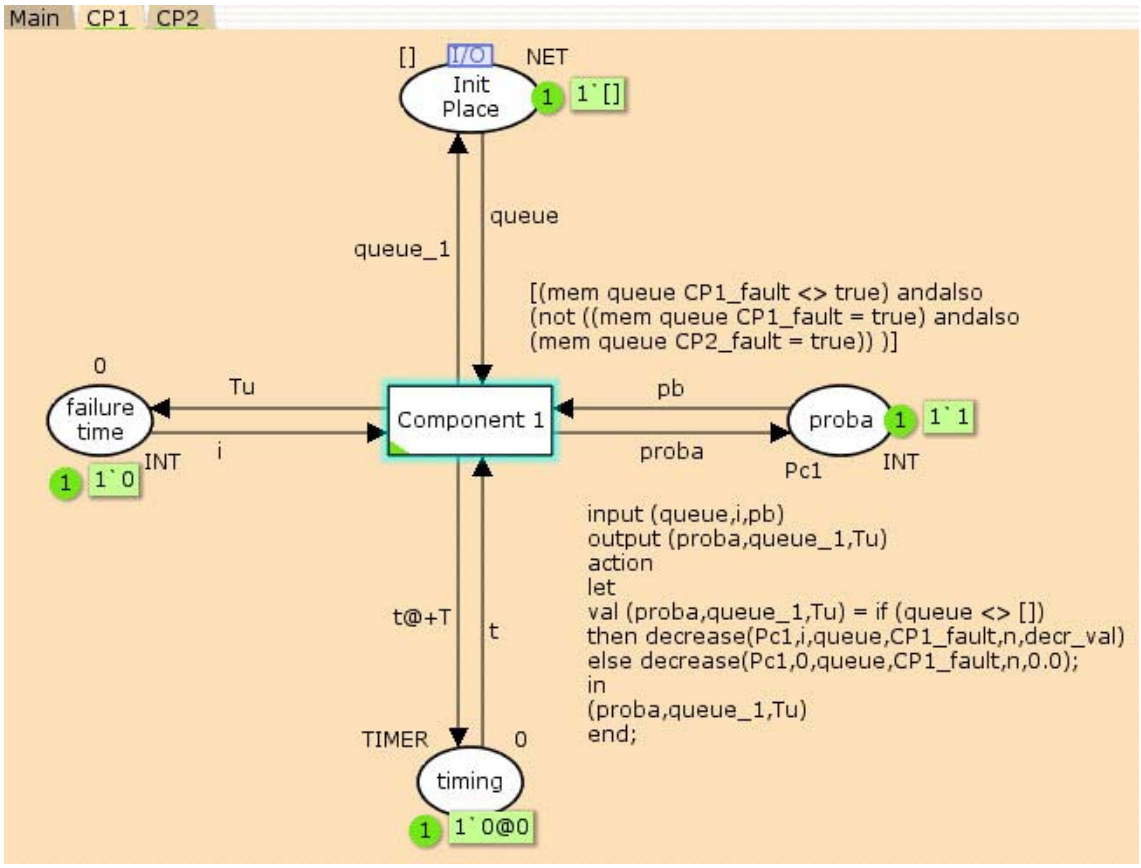


Fig. 4.3 Petri nets model of the component with time depending on the failure rate $\lambda(t)$

The functionality of the component which covers changing the failure rate λ by a a specified function composed of decrease factor parameter is encapsulated in the code region of the transition "*Component 1*". The function is defined using ML programming language as follows:

```
fun decrease(pb,i,queue,pack,n,d_val) =
let
val exp_val = Math.exp(d_val*Real.fromInt(i));
val p_inc = Real.fromInt(pb) + exp_val;
val r_n = Real.fromInt(n);
val rand_val = exponential(p_inc/r_n);
val queue_1 = if floor(rand_val) < 1
```

```
    then queue^^[pack] else queue;
val proba = round(p_inc);
in
(proba,queue_1,i+1)
end;
```

The function *decrease(...)* (Fig. 4.3) encapsulates generating the random events such as failure by using the exponential distribution of the life time of the component. The second functionality which is covered by this function is time depend failure rate increasing of the component which is defined by specified decrease factor (parameter of function *decr_val*).

In case of the single failure rate increasing (increasing from previous constant value to a new constant value just after changing the component's load) the function *decrease(...)* is a little bit modified because of any implementation of the functional increasing is considered.

In order to detect the failure state of the second component the transition in figure 4.3 has guard condition which allows exclude the failed components out of working. The PN model of the component composes of the fourth place "*failure time*", too. This place is used as counter of the time from the moment of the second component failure. It is necessary to know the time spent from the moment of failure of the second component (from failure time t_f of the CP2) in order to implement functionality where failure rate of the additionally loaded component is time dependent.

As an example of the presented model we can suppose SafePLC automat where two independent controllers are implemented connected by internal communication channel. Thus, in case of failure one of them the second one starts follow its control task each sample period. The model presented above takes into account the system where shared redundant approach. Thus, in context of SafePLC when one processor failed the second execute its task and some task in addition. Let's suppose that probability of the failure of the processor is defined as probability that processor will fail in one of n executed instruction cycles. Thus, with executing additional task by second non-failed processor the number of instruction cycles is increased which consequentially increase the probability of the failure (failure rate) of the non-failed processor.

In further text described two component QR system is presented under various conditions of the load sharing. Thus, different changes of the failure rates of the quasi-redundant components are studied. Several different cases are introduced from the

simple single change to more sophistical time dependent on failure rate changes. In general, several probable cases of the failure rate change are considered such as linear and exponential in several variations.

4.2.1 Simulation results for a single step change of the failure rate

At first the simulation of the mentioned simple system with single change of the failure rate of the QR component was provided. Thus, the condition of the component's failure rate changing is as follows. In case of failure of one of the components, the failure rate of the non-failed component λ_n changes (increased) by the value of the decrease factor d_R. Thus, a new failure rate λ' of the non-failed component is:

$$\lambda'(t) = \lambda_n + d_R \text{ , for } t \geq t_f \tag{4.6}$$

where t_f is failure time one of the quasi-redundant components.

This change could be called as single change because the component's failure rate is changed only once during QR component's life time. Both components has equal nominal failure rate $\lambda_n = 0.001$.

Few examples of the influence of the decrease factor d_R are shown in figure 4.4. Two non-dashed curves show studied system as a system with two active redundant components (thus, d_R is equal to zero – first curve from the top) and as system without redundant components (thus, system composes of two independent components without redundant relation – first curve from the bottom).

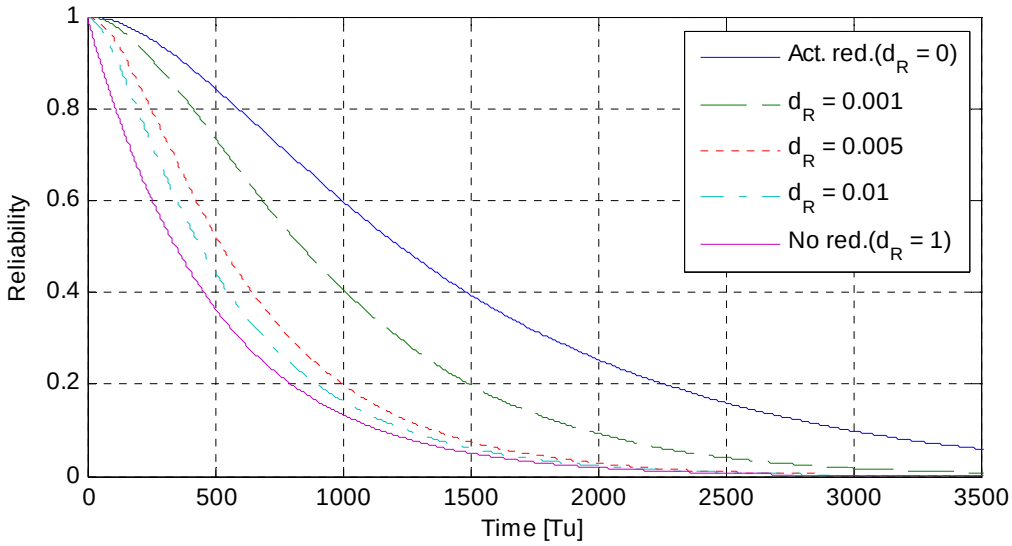


Fig. 4.4 Influence of the increased failure rate of the component by the constant decrease factor d_R to the reliability of the system composed of two quasi-redundant components

These two curves determine borders where reliability of the studied system can be changed depending on the value of the decrease factor d_R.

As we can see from figure 4.4, single increasing of the nominal failure rate λ_n of the non-failed components by the same value as was nominal failure rate λ_n up to $\lambda' = 0.002$ ($d_R = 0.001$) cause significant negative change of the reliability as well as fail time (*MTTF* – Table 4.1) of this studied system. With the value of the decrease factor $d_R = 0.01$, the curve is significantly closer to the curve of the non-redundant system. Thus values of the d_R about 0.1 we can consider as values which reduces the dependability of the two components quasi-redundant system significantly down near to the results of the system without consideration of the redundant components.

$\lambda_n = 10^{-3}$	$d_R = 0$ ($\lambda' = \lambda_n$)	$d_R = 0.001$ ($\lambda' = 2\lambda_n$)	$d_R = 0.005$ ($\lambda' = 6\lambda_n$)	$d_R = 0.01$ ($\lambda' = 11\lambda_n$)	$d_R = 0.1$ ($\lambda' = 101\lambda_n$)	$d_R = 1$ ($\lambda' = 1$)
MTTF [*Tu*]	1503 (+300%)	1002 (+200%)	667 (+34%)	589 (+18%)	509 (+2%)	499

Table 4.3 Mean Time to First Failure of the studied system depending on the decrease factor d_R

All curves shown in the figure 4.4 can be approximated by equation (4.7) which represents a similar approach as in case of the approximation the different type of curves by Taylor series approach. This equation is as follows:

$$R(t) = e^{-\lambda t}\left(1 + \sum_{i=1}^{\infty}(-1)^{i-1}\left(k_i \frac{(\lambda t)^i}{i!}\right)\right), \tag{4.7}$$

where λ is the common failure rate all of the components of the studied system and k_i is i-th coefficient of this expansion.

Thus, with approximation of the coefficients it is possible to describe the reliability of the studied system. As we can see from the result of the simulated system, the decrease factor has a direct influence on the coefficients of the mentioned expansion. For example, for the decrease factor equal to zero (active redundant system composed of two components) equation (4.8) can be used for evaluating the reliability of this system with two active redundant components with the same failure rate λ:

$$R_A(t) = e^{-\lambda t}\left(1 + \sum_{i=1}^{\infty}(-1)^{i-1}\frac{(\lambda t)^i}{i!}\right) \tag{4.8}$$

As we can see from equation 4.10, all coefficients k_i are equal to 1 for decrease factor equal to zero. Similarly we can suppose that it is possible to describe all obtained reliability curve by similar mathematical expansion with different coefficients.

4.2.2 Linearly changed failure rate in 1-out-of-2 subsystems

In previous text the influence of constant change of the nominal failure rate was analyzed. In this part of the work the study of the linear changing of the nominal failure rate λ_n of the component is brought closer. The change is based on the equation mentioned in the third chapter as well as further specified equations (4.9).

The failure rate is increased by the following linear function with different rising speed. The rising speed is characterized by the decrease factor which represents the constant value in the linear function such as:

$$\lambda'(t) = \lambda_n + \lambda_n d_R (t + 1 - t_f), \text{ for } t \geq t_f \qquad (4.9)$$

where t+1 allows change the nominal failure rate λ_n at the moment of the failure at time t_f.

By increasing the value of the decrease factor d_R the nominal failure rate raises faster. The influence of this linear change of the failure rate (Fig. 4.5b) is shown in figure 4.5a. There are shown five reliability curves of the studied two component system.

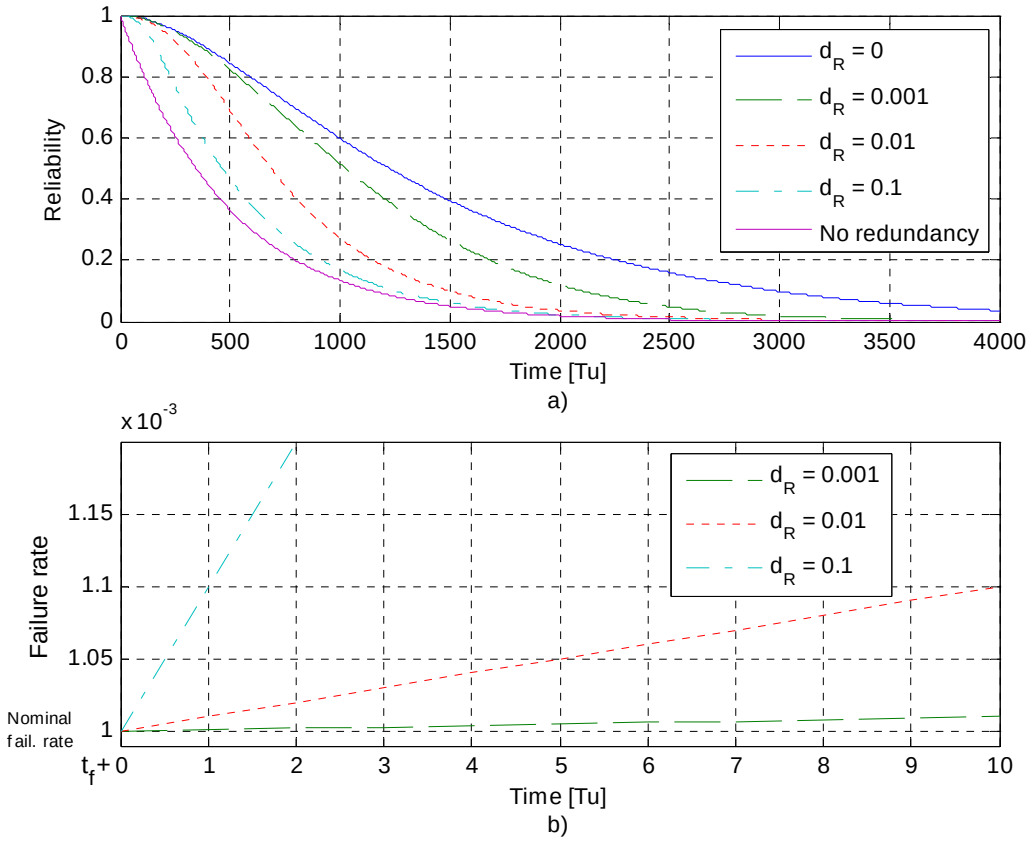


Fig. 4.5 Influence of the linearly increased failure rate by decrease factor d_R value

As we can see, there are three dashed curves in bound between the reliability curves of the studied system considered as active redundant (top curve $d_R = 0$) and studied system considered as two independent components (bottom curve). Only in this bound between two non-dashed reliability curves the results of the reliability analysis with different values of the d_R of the studied system (1-out-of-2) can be placed.

Rising speed of the nominal failure rate $\lambda_n = 10^{-3}$ is shown in the bottom figure (Fig. 4.5b). Each linear curve which characterizes the failure rate increasing corresponds with three dashed reliability curves (Fig. 4.5a). As we can see, the reliability of the studied system decreases from the nominal curve ($d_R = 0$) depending on the decrease factor. Its decreasing represents a slower rising speed of the nominal failure rate. Thus, the reliability curve of the system with the smallest decrease factor is closest to the top non-dashed curve which represents the studied system composed of two active redundant components ($d_R = 0$).

For a better illustration, the table 4.4 is prepared where *MTTF* parameters corresponding with considered decrease factors in figure 4.5 are shown. As we can see, the bound of the possible values of the *MTTF* for studied system is limited by highest value 1503*Tu* for a decrease factor equal to zero (active redundant components – top non-dashed curve in figure 4.5) and by the lowest value 499*Tu* which describes bottom reliability curve in figure 4.5.

Decrease factor	*Active redundancy* $d_R = 0$	$d_R = 10^{-3}$	$d_R = 10^{-2}$	$d_R = 10^{-1}$	*No redundancy*
MTTF [*Tu*]	1503	1153	812	611	499

Table 4.4 Examples of the Mean Time to First Failure parameter for linearly increased failure rate following d_R value

As we can see with relatively slow linear increasing of the nominal failure rate ($d_R = 10^{-3}$) the failure time of the system was reduced approximately 23%. With further increasing of the decrease factor the values closer to bottom limit is achieved. Based on these values it is possible to derive the curves which allow a fast approximation of the life time of this shared redundant system type. Obtained values and further approximations will be introduced in more details in the next chapter.

Previously the example of continuous linear increasing of the component's nominal failure rate λ_n in case of load sharing of the component was presented. Further, the

similar example of the failure rate changing is presented. However, the same simulations were provided but with a discrete (step) increasing of the failure rate λ_n each n-th time unit by a value specified by the decrease factor. This changing of the nominal failure rate is described by iterative equations (4.10 and 4.11) as follows:

$$\lambda'(T_f) = \lambda_n + d_R, \tag{4.10}$$

$$\lambda'(T_f + kTu) = \lambda'(T_f + (k-n)Tu) + d_R, \text{for } k = n, 2n, \ldots \tag{4.11}$$

where λ' is the changed failure rate of the component after a failure time, λ_n is the nominal failure rate of the component until failure time, T_f is the time unit when one of the components failed (failure time), n is the time delay ($n*Tu$) after which the failure rate has to be increased and d_R is the decrease factor which represents the size of the failure rate change.

The results of the simulations are shown in figure 4.6. There are three different sub-figures with different delays for failure rate increasing. In each sub-figure there are five reliability curves, the same as in case of figure 4.5. Two non-dashed curves represents bound where simulation results are expected.

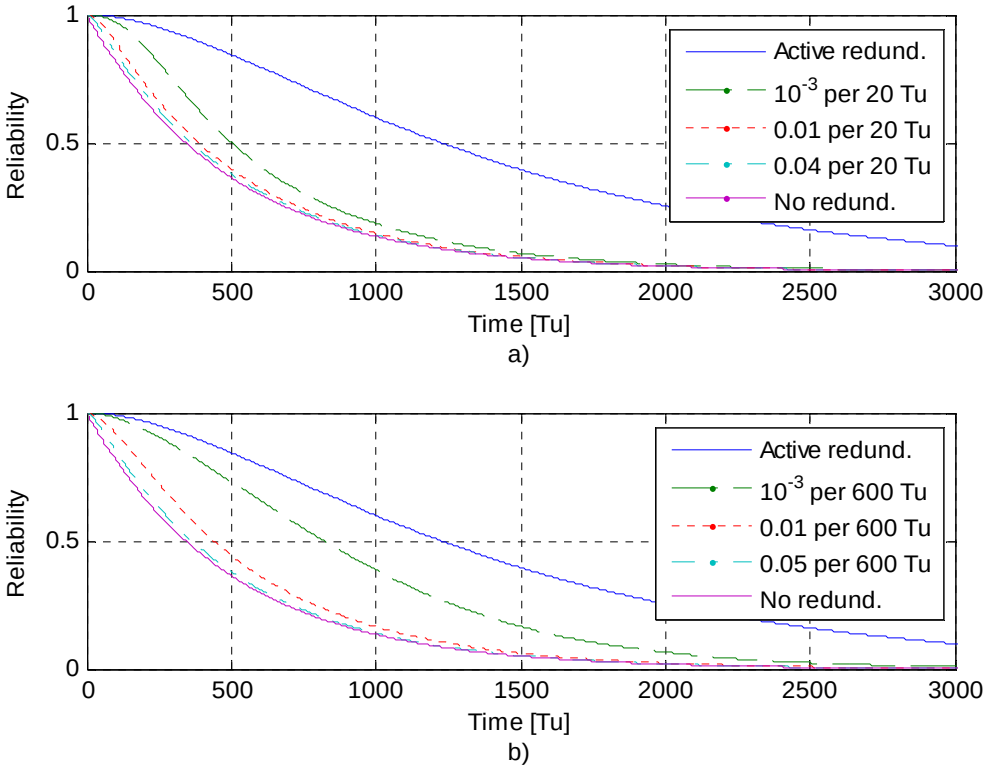


Fig. 4.6 Influence of the linearly increased failure rate by decrease factor d_R each 20^{th} and 600^{th} time unit to reliability of the two components redundant system

Other three curves correspond with three different values of the decrease factor shown in legend. In figure 4.6 we can see the decreasing of the nominal failure rate each 20 and 600 time units. These three examples were chosen in order to well demonstrate the reliability behavior of the studied system with the previous described rule of failure rate changing. Practically, the behavior described by equations 4.10 and 4.11 could by found in NCS where two quasi-redundant networks are considered. When one control network is failed, then the second (non-failed) network has to transmit new data sent by the new nodes which are connected in sequence to the network after the system reconfiguration. Thus, by connection of each node to the non-failed network the reliability is increased by a value of decrease factor.

From figure 4.6 that for delay of the value $n = 20$ (failure rate is increased each 20 time units – see Fig. 4.6a) the values of the system reliability as well as life time (*MTTF* parameter in table 4.5) go rapidly down.

As was abovementioned *MTTF* of the studied system without any decrease factor (equivalent of the two active redundant components) is 1503 time units (Table 4.3). Down limit of the system for the *MTTF* parameter is 499*Tu*. Thus, when we take the above mentioned example with two networks where load sharing with above defined behavior is considered, then the life time of the network with a nominal failure rate $\lambda_n = 10^{-3}$, where each 20 time units one node is connected cause the increasing of the failure rate of the network about the same value as λ_n ($d_R = 10^{-3}$), is reduced about 57% down to the value 649*Tu* (see table 4.5). With a higher decrease factor, the value is getting near to the down limit of the studied system.

Table 4.6 shows values of the *MTTF* of the studied systems for other delay 600*Tu* too. As we can see with delay 600*Tu* and decrease factor 10^{-3}, the reliability curve is much closer to the reliability of the system with two active redundant components. However, 600*Tu* could be a too long delay for the failure rate increasing especially in the case of systems with high dynamics. On the other side by following the reconfiguration strategy proposed and described in more details in the next chapter, the delay could be extended in order to obtain some time to save the system.

Per 20 *Tu*	$d_R = 10^{-3}$	$d_R = 10^{-2}$	$d_R = 4\ 10^{-2}$
MTTF [*Tu*]	649	543	517

Table 4.5 Examples of the Mean Time to First Failure for increasing the failure rate by d_R value each 20th time unit

Per 600 *Tu*	$d_R = 10^{-3}$	$d_R = 10^{-2}$	$d_R = 5\ 10^{-2}$
MTTF [*Tu*]	942	588	519

Table 4.6 Examples of the Mean Time to First Failure for increasing the failure rate by d_R value each 600th time unit

From the presented reliability curves (Fig. 4.5, 4.6) as well as values in tables 4.5 – 4.6 we can see that in case of control system reconfiguration where one network failed and the failure rate increases rapidly within short time intervals (for example decreasing each 20 time units) it is necessary to take into account relatively short (depending on the system dynamics) time interval in order to safety finish the system mission.

4.2.3 Simulation results for exponentially changed failure rate

The real systems are usually non-linear hence all analytical models are linearized for a specified bound of working [Sar 92]. So, we can suppose that increasing of the nominal failure rate of the quasi-redundant component could be non-linear with the increasing of the component's load. The increasing of the failure rate by following exponential behavior could help to describe usual non-linear FR change behavior within the real NCS. This type of failure rate change could be caused by non-controlled loading of the subsystem.

Thus, a simulated system has implemented the exponential failure rate increasing in case of component failure. Hence, the behavior of the change of the nominal component's failure rate is described by equation (4.12):

$$\lambda'(t) = \lambda_n + \lambda_n e^{d_R (t - t_f)}, \text{ for } t \geq t_f \qquad (4.12)$$

where λ' is the value of the increased failure rate, λ_n is the nominal failure rate of the component, t_f is the time of the failure of the component, d_R is the decrease factor which has a direct influence to the increased failure rate.

In the same way as in the previous section where linear change was considered and evaluated, there is a reliability behavior of the studied system shown in figure 4.7a. In figure 4.7b there is the change of the nominal failure rate $\lambda_n = 10^{-3}$ following the exponential function (equ. 4.12) where the speed of the raising is defined by the decrease factor which represents the exponent of the function. Following the

exponential function at the moment of the component's failure ($\lambda'(t_f) = \lambda_n + \lambda_n e^0$), the failure rate of another one is doubled up to value $\lambda' = 0.002$.

As we can see the decrease factor has a direct influence to the raising speed of the function from slowly raising with value about $d_R = 10^{-3}$ (and smaller) to values about 10^{-1} which represents the triple value of the nominal failure rate in the first seven time units.

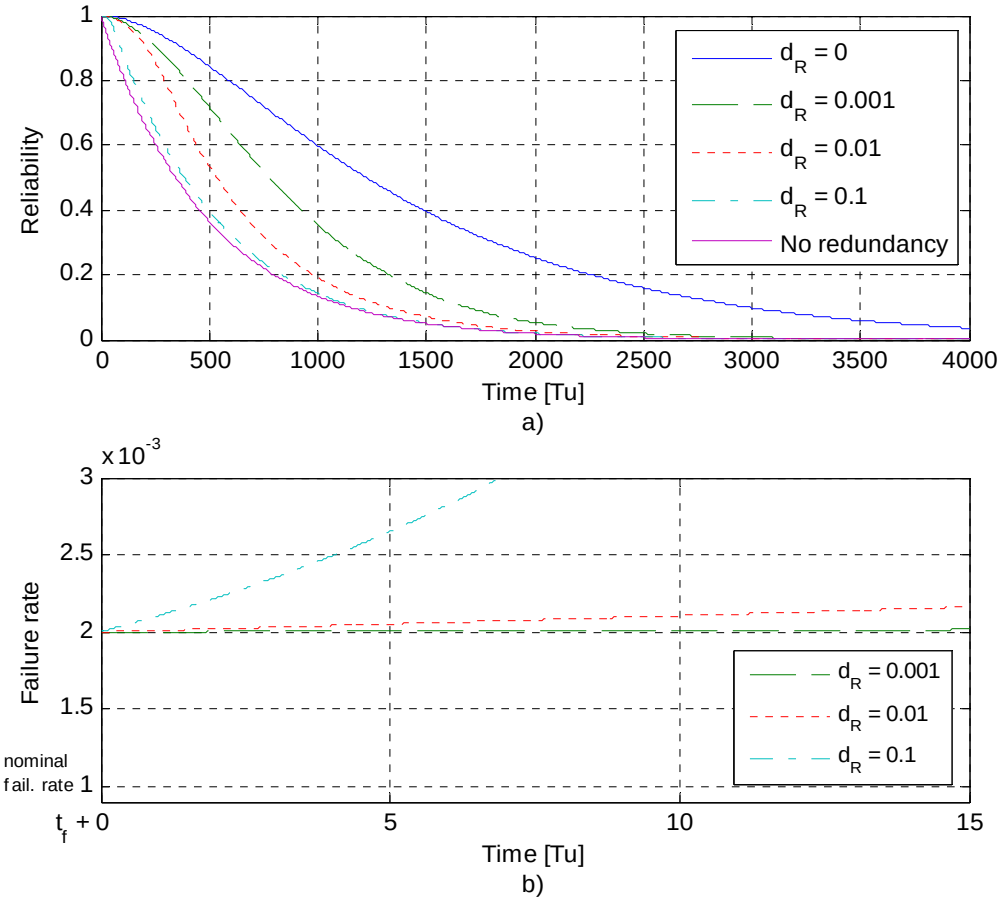


Fig. 4.7 Influence of the exponentially increased failure rate of the shared redundant component to the reliability behavior of the system 1-out-of-2

With exponent values about 10^{-1}, the reliability of the system is almost the same as in case of the system without redundant components (non-dashed curve on the bottom in figure 4.7a). This observation is confirmed by values shown in table 4.7 too. As in previous parts, the table with dependability parameter *MTTF* is prepared in order to well describe the life time to first failure of the studied system with the specified

dependency. We can see that time life of the non-redundant system is shorter only about 38 time units in comparison with simulation results for a decrease factor equal to 10^{-1}.

Decrease factor	*Active redundancy* $d_R = 0$	$d_R = 10^{-3}$	$d_R = 10^{-2}$	$d_R = 10^{-1}$	*No redundancy*
MTTF [*Tu*]	1503	902	676	537	499

Table 4.7 Several values of the dependability parameter Means Time to First Failure for exponentially increased failure rate following the equation 4.11

As we can see from table 4.7 with small decrease factor (values about 10^{-3}) the reliability of the system as well as its life time is getting near to the reference value of the system with two active redundant systems (1503*Tu*). For $d_R = 10^{-3}$ the *MTTF* parameter is 902*Tu* which is characterized by a 40% reduction from the higher possible value (1503*Tu*) within the limit of the bound <1503*Tu*; 499*Tu*>.

4.3 Triple Modular Redundancy with load sharing consideration

In the previous part of this chapter, the simple often used two component redundant subsystem was studied under different conditions of loading for non-failed components. In further text another frequently used redundant subsystem is studied. This subsystem is composed of three quasi-redundant components CP1 – CP3 (Fig. 4.8). The root page of the hierarchical PN model of this system is shown in figure 4.8. There are three sub-pages for each component as well as transition "*Init PN*" for PN initialization when condition of firing is accomplished. Thus, when simulated system is failed PN model is reinitialized by using the "*InitializePN()*" function (the same as in the previous presented model).

Global functionality of this system is as follows. In case of failure, one of the components *component fault* packet is sent to the "*Init Place*". When the first component which detects that one of the components was failed, then it takes load of the failed component and consequently increases its own failure rate λ_n. The failure rate is increased by following the law implemented by the *decrease(...)* function. This function is used in the sub-model of each component in code region of the transition (in figure 4.9 it is transition "*Component 2*").

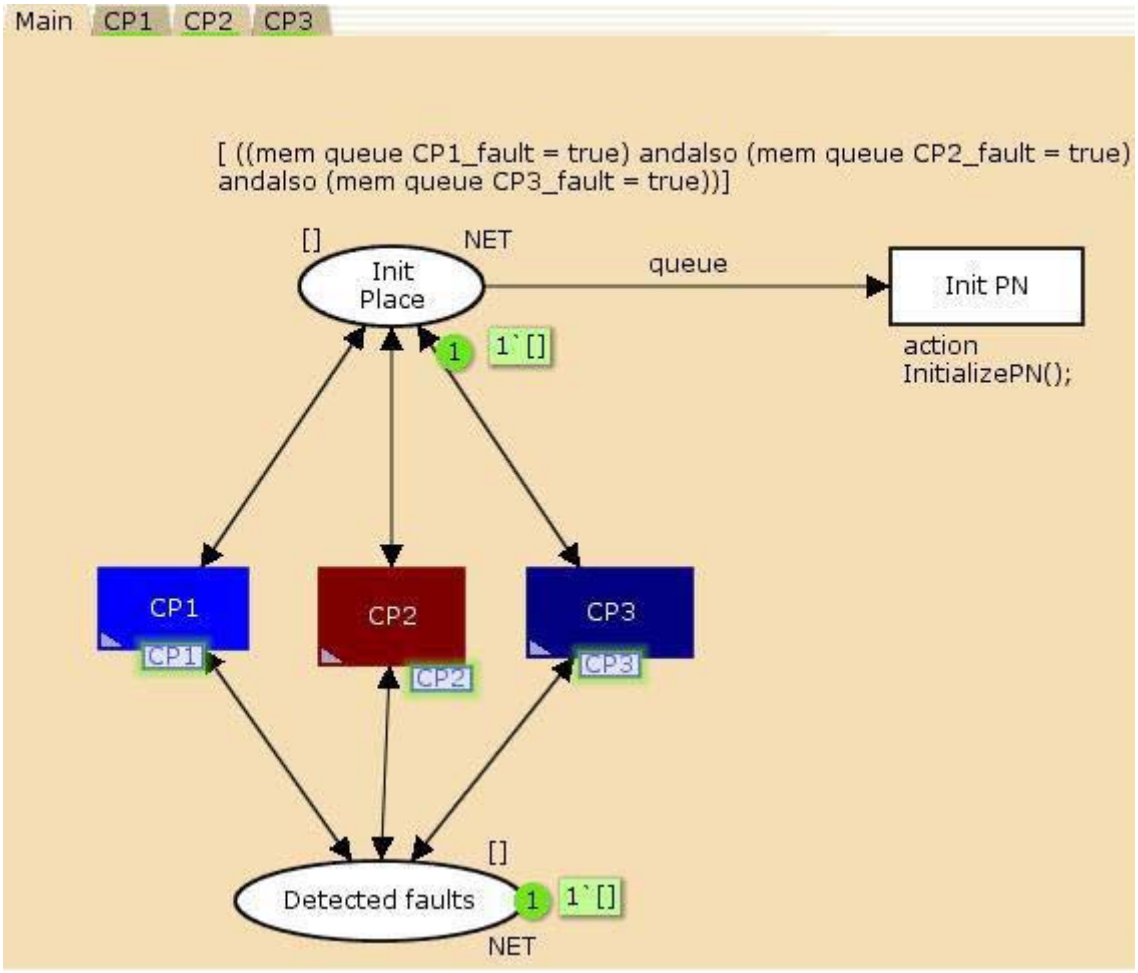


Fig. 4.8 Hierarchical PN model of the three quasi-redundant components

Each component composes of the places and a transition as we can see in figure 4.9. Four places are the same as in PN model of the two component-redundant system. The place "*Init Place*" is the input/output place from and to sub-page of the component. Timing of the model is realized on the output arc from transition to the place "*timing*". Place "*proba*" is used for storing the rounded number of the actual failure rate multiplied by thousand (no real type of the places is supported) and place "*failure time*" and counting the time passed from the component's failure.

In comparison with previous model (1-out-of-2), there are additional places "Detected faults". This place, the same as "*Init place*" is input/output (I/O) place which represents interconnection component's sub-page and model main page. Using variables *queue_1* and *queue_2* are associated to input and output arcs between transition and place, non-failed components are informed whether the failed component is supported by another one or not. Each non failed component compares the value in place "Detected faults" with values in "Init places". For example, when value *CP1_fault* is

found in both I/O places, then another non-failed component has to take load of the failed components and to replace by its own identifier (for example *CP2_fault*).

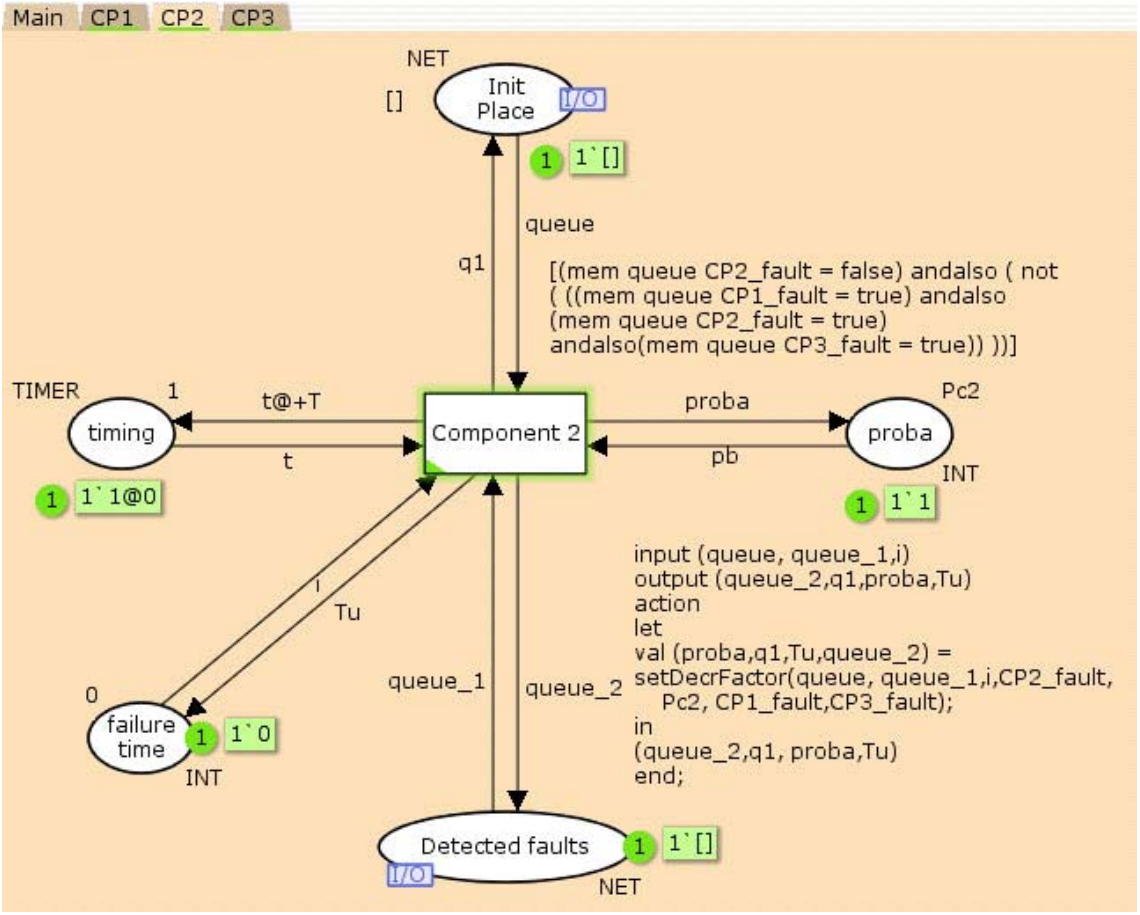


Fig. 4.9 Petri nets model of the component of TMR system with time dependent decrease factor $f_{dR}(t)$

Abovementioned function *decrease(...)* is encapsulated by function *setDecrFactor(...)* which is defined in ML programming language as follows:

```
fun setDecrFactor(queue, queue_1,i, pack,Pc, pack1, pack2) =
let
val pack_in = if queue_1 <> [] then (hd queue_1) else default_packet;
val (queue_1, d_val) = if mem queue pack_in
      then ([], decr_val) else (queue_1,decr_val);
val ((proba,queue_2,Tu),q_out) = if ((mem queue pack1 = true
      orelse  mem queue pack2 = true) andalso ((queue_1 =  []))
orelse (mem queue_1 pack)) )
      then (decrease(Pc,i,queue,pack,n,d_val), [pack])
      else (decrease(Pc,~1,queue,pack,n,0.0), queue_1);
in
(proba, queue_2, Tu,q_out)
end;
```

The *decrease(...)* function has a similar functionality as in the previous model. However in order to support more complicated logic of the fault detection, the function *setDecrFactor(...)* is proposed. This function encapsulates above described functionality of detection of the failed components and of change the values in the "*detected faults*" place.

4.3.1 Single change of the component's failure rate

As in the previous part where a system 1-out-of-2 was simulated, the simple single change of the failure rate after load sharing is introduced, too. We consider quasi-redundant system 1-out-of-3 where three quasi-redundant components are considered. Each component has the same failure rate value $\lambda_n = 10^{-3}$ which corresponds to the probability that the system will not failed within the time the interval specified by a time unit *Tu* equal to 0.999.

The single step change of the nominal failure rate by a value specified by the decrease factor d_R (see equation 4.6) is provided, just after when component detects and takes load of another failed component. The FR is increased by the same value (d_R) in both cases when one component takes the load of the one failed component as well as two failed components.

The reliability curves shown in figure 4.10 describe the reliability behavior of the five variants of the system composed of three subsystems. The first reliability curve (top non-dashed) represents the systems with three active redundant components whereas it is the high limit of the bound where the other simulation results should be placed.

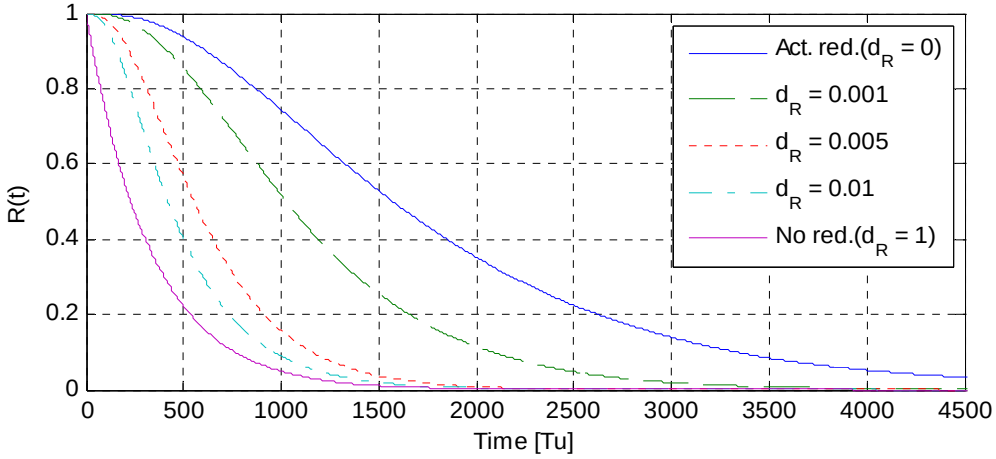


Fig. 4.10 Reliability behavior of the system composed of the three quasi-redundant subsystems with different values of the failure rate increasing

Bottom non-dashed curve represent the reliability of the system with three independent sub-systems. Thus, it is reliability down limit for the three components system. *MTTF* parameter of this studied system considered as active redundant equivalent ($d_R = 0$) is 1826*Tu* (table 4.8). The value of the *MTTF* for the same system but with three independent components it is 332*Tu*, which represents approximately 82% reduction.

In figure 4.10 there are three examples of the different decreasing of the failure rate. Long dashed curve which is closest to the top reliability curve represents the reliability behavior of the studied shared redundant system with component's failure rate increased by value $d_R = 10^{-3}$ to the $\lambda' = 2.10^{-3}$. This value of d_R causes reduction of the life time of the system about 36% (table 4.8) within the bound by two non-dashed reliability curves <332*Tu* ;1827*Tu*> (this interval is considered as 100% bound). As we can see with further increasing the dR value the reliability of the studied system with this simple load sharing dependency go rapidly down to *MTTF* value 332*Tu*.

$\lambda_n = 10^{-3}$	$d_R = 0$ ($\lambda' = \lambda_n$)	$d_R = 0.001$ ($\lambda' = 2\lambda_n$)	$d_R = 0.005$ ($\lambda' = 6\lambda_n$)	$d_R = 0.01$ ($\lambda' = 11\lambda_n$)	$d_R = 1$ ($\lambda' = 1$)
MTTF [*Tu*]	1827	1164	638	507	332

Table 4.8 Mean Time to First Failure of the studied three components system depend on the decrease factor d_R

Thus, in case of shared load which increase nominal value of the component's failure rate eleven times the life time of the system is rapidly reduced about 72% down to 507Tu. This dependency between increasing the failure rate and time life will be presented in more details in the next chapter.

4.3.2 Simulation results for linearly changed failure rate

Further the linear increasing of the failure rate is studied as another dependency among the shared redundant subsystems. This type of the change of the failure rate follows the same equation as in case of the two-component system (eqv. 4.9). Figure 4.11 there are shown several simulation results in order to demonstrate the influence of the considered dependency to the final reliability of the studied system. The results are shown in the similar fashion as in the figure before. Thus, there are the same two non-

dashed curves which represent mentioned limits for the reliability of the three components system.

As in the previous cases the study is focused to the dependence between parameter of the linear function (decrease factor d_R) and the final reliability as well as time life (parameter *MTTF*). In figure 4.11a are shown three examples of shared redundant system with linearly increased failure rate which corresponds to different decrease factors (10^{-3}, 10^{-2}, 10^{-1}).

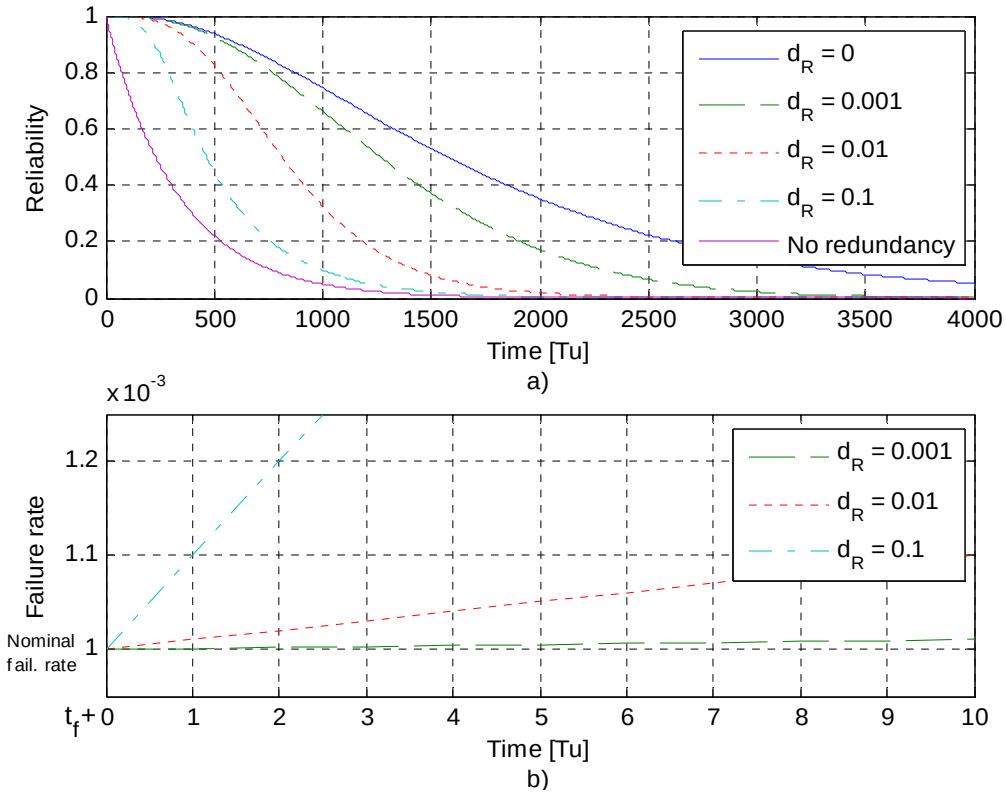


Fig. 4.11 Influence the constant decrease factor d_R to the reliability of the system composed of three quasi-redundant components

$\lambda_n = 10^{-3}$	$d_R = 0$ $(\lambda'(t_f) = \lambda_n)$	$d_R = 0.001$ $(\lambda'(t_f) = 2\lambda_n)$	$d_R = 0.01$ $(\lambda'(t_f) = 11\lambda_n)$	$d_R = 0.1$ $(\lambda'(t_f) = 101\lambda_n)$	$d_R = 1$ $(\lambda'(t_f) = 1)$
MTTF [*Tu*]	1827	1369	880	552	332

Table 4.9 Mean Time to First Failure of the studied system depending on the decrease factor d_R

As we can see, with values of the d_R about 10^{-1} (dashed and dotted curve) and more the reliability of the system is getting near to down limit (bottom non-dashed curve) represented by reliability curve of the three independent components. With this value of

the d_R, the raising of the failure rate is relatively fast whereas each ten time units (Fig. 4.11b) the failure rate is increased by the nominal value λ_n. In comparison with decrease factor 10^{-3}, the double value of the nominal failure rate λ_n is obtained after thousand time units. With a decrease factor about 10^{-2} the life time of the system is getting down to the approximately half value of the maximal life time (Table 4.9) of the three components system (active redundancy).

Similarly we can see the large difference between the life time values characterized by the decrease factors 10^{-3} and 10^{-1}. The time life (Table 4.9 – *MTTF* parameter) of the studied system with d_R equal to 10^{-3} is about 817*Tu* more than in case of the system where the decrease factor is equal to 10^{-1}. Thus, the reducing of the life time about 817*Tu* corresponds to the change of the failure rate each thousand time units about the value of λ_n to the same change but each ten time units. In comparison with the maximal value of the possible time life (1827*Tu* – Table 4.9) the reduction for d_R equal to 10^{-3} represents 25% and 70% for d_R equal to 10^{-1}.

Based on the values described above and shown in table 4.9, we can suppose that it is necessary to propose a strategy for the system reconfiguration in order to prevent the fast increasing of the nominal failure rate of the quasi-redundant component.

4.3.3 Simulation results for linearly changed of the failure rate each *k*-th time unit

This part considers the linear change of the failure rate in a discrete manner as it was presented in part 4.2.2. As in previous case-study (4.3.2), the failure rate increases linearly however discretely in steps each *k* time units (see the equations 4.10 and 4.11).

Figure 4.12 shows three different cases with the same value of the decrease factor $d_R = 10^{-3}$. This value is equal to the nominal failure rate λ_n of the components. Increasing the failure rate shown in figure 4.12 was provided each 20^{th}, 200^{th} and 600^{th} time unit. The difference between the failure rate change described in this part of the chapter and in the previous part is that failure rate increases by a value of λ_n at the time of component the failure T_f (eqv. 4.10) in comparison with the linear increasing (eqv. 4.12) where the same value of the failure rate ($2\lambda_n$) is obtained during 10^{th} time unit when decrease factor is equal to 10^{-1}. Thus, this part describes the step change of the failure rate whereas this change of the same size (defined by d_R) is provided ten time units later (in case of $d_R = 10^{-1}$ in equation 4.12).

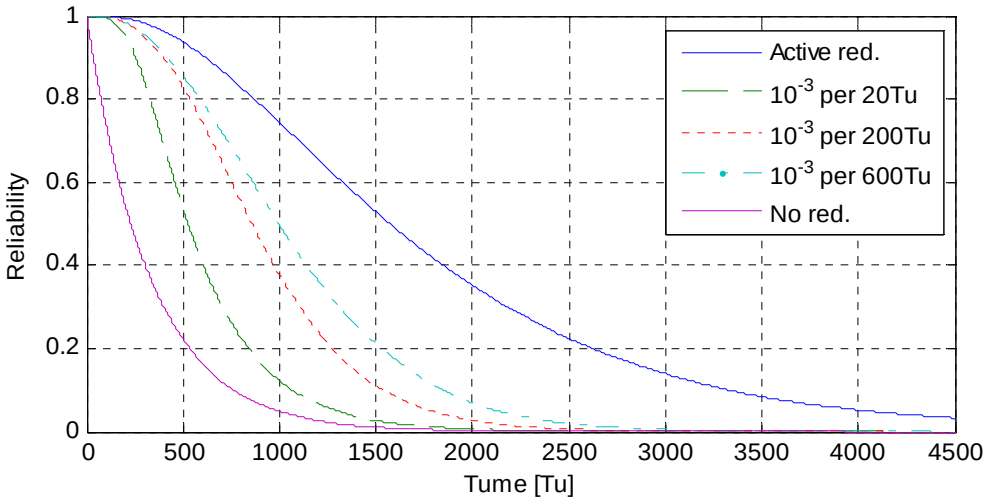


Fig. 4.12 Influence of the linearly increased failure rate by decrease factor d_R = 0.001 each *20th*, *200th* *and 600th* time unit to reliability of the three components redundant system

If we compare the values of the *MTTF* presented in table 4.10 we can see that the differences described few lines above between continuous linear and discrete linear increasing has a relatively minimal influence to the final time life of the studied system. The example is the value *MTTF* = 542*Tu* (Table 4.10) for the discrete linear increasing of the failure rate each ten time units which can be compared with the continuous increasing with parameter $d_R = 10^{-1}$ where the value of the *MTTF* is 552*Tu* (Table 4.9). The difference between the obtained values is minimal (also when we take into account possible error of the MCS).

MTTF	per 10 *Tu*	per 20 *Tu*	per 200 *Tu*	per 600 *Tu*
$d_R = 10^{-3}$	542 *Tu*	610 *Tu*	928 *Tu*	1082 *Tu*

Table 4.10 Values of the Means Time to First Failure parameter for increasing the failure rate by d_R value each *20th*, *200th* *and 600th* time unit

Thus, the step change of the failure rate on the value 2.10^{-3} at the moment of the failure of one of the components has a minimal influence in comparison with the system where the same value of the failure rate is obtained ten time units later. On the other side, in case of the critical systems, this few time unit could be very significant in order to take the system in the safe mode. Hence, this result could be helpful in order to

prepare some suitable system reconfiguration strategy when the subsystem is failed where all these mentioned differences should be considered.

The reliability curves figure 4.12 as well as the values of the *MTTF* parameter in table 4.10 show that the system should be reconfigured with the aim of extending the time where the overloading of the subsystem has not a critical level. Thus, guard the minimal change of the nominal failure rate of the subsystem for a maximal time.

4.3.4 Simulation results for exponentially changed failure rate

Last dependency which is dealt further is non-linear increasing the failure rate following the exponential function (eqv. 4.11). As previous studies with linear dependencies with fast failure rate increasing, the obtained reliability and *MTTF* values are getting closer to theirs down limit. The down limit is represented by non-dashed reliability curve on the bottom in figure 4.13a and in case of *MTTF* parameter, it is value 332 *Tu* shown in the table 4.11.

The fast raising of the failure rate corresponds with value of decrease factor (exponent in equation 4.11) equal to 10^{-1} (Fig. 4.13b). With decreasing this value the raising of the failure rate is slower and the reliability of the system is getting closer to the maximal possible values.

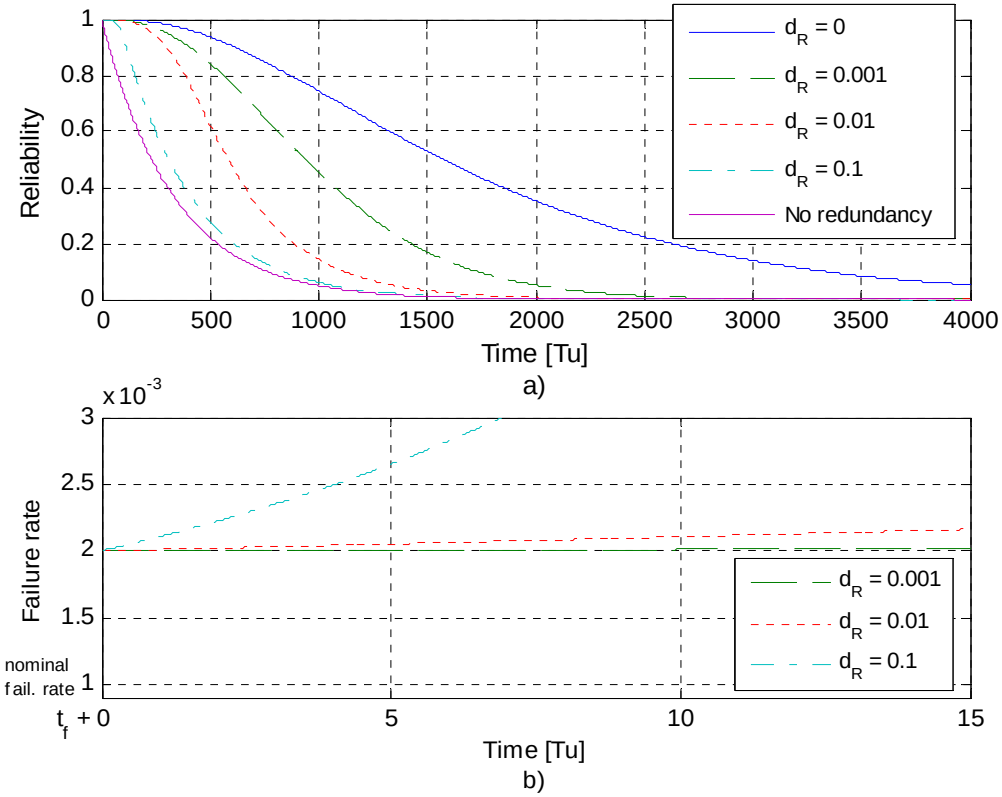


Fig. 4.13 Influence the exponentially increased failure rate of the shared redundant component to the reliability behavior of the 1-out-of-3 system

As was explained in part 4.2.2, the property of the failure rate increasing is characterized by its step change ($\lambda'(t_f) = \lambda_n + \lambda_n e^0$) at the failure time t_f thus at the moment when the load of the component increased. However as was shown in previous part of this chapter, the step change of the failure rate at the beginning could give us similar results as in the case of the system where the same level of the failure rate is obtained few time units later.

Decrease factor	*Active redundancy* $d_R = 0$	$d_R = 10^{-3}$	$d_R = 10^{-2}$	$d_R = 10^{-1}$	*No redundancy*
MTTF [*Tu*]	1827	1121	661	409	332

Table 4.11 Several values of the dependability parameter *MTTF* for exponentially increased failure rate of the studied three component system following the equation 4.11

The values presented in table 4.11 show decreasing of the life time of the studied system. As we can see with parameter $d_R = 10^{-1}$, the life time of the system is reduced to 409 *Tu* from the maximal value 1827 *Tu* what represents approximately 78% of reduction. In the case of $d_R = 10^{-1}$, he life time is reduced of 39% which represents a significant difference.

4.4 Conclusion

The chapter deals with the simulations of the different possible dependencies within the shared redundant systems whereas two basic and most used combinations of the redundant subsystem are chosen. The method used for the evaluation of the reliability as well as *MTTF* dependability parameter is Monte Carlo Simulation. Hence, the first part of the chapter is offered to this problematic. The presented example is confirmed that MCS can be used for reliability evaluation of the redundant systems whereas proposes a sufficient level of exactness for further works.

In order to bring closer the problematic of the quasi-redundant systems, two basic systems are evaluated. Several different dependencies which could be considered when load is shared among the quasi-redundant components are studied. This case study describes the reliability behavior of the quasi-redundant components not only under simple conditions as it is often presented (single change of the failure rate) however it brings closer more sophisticated changes of the failure rate.

By the description of the different dependencies and the differences in the failure increasing, several observations are presented in order to show the needs of the systems where shared redundant approach is applied. For example, the difference between immediately increased failure rate and delayed increasing the failure rate (caused by linear failure rate increasing).

Based on the observations of the reliability behavior of the studied two and three component shared redundant systems was briefly drafted out the way how to develop suitable reconfiguration strategy which could be fully applied in condition of the NCS where quasi-redundant networks as well as other sub-systems are used. The approximation of the influence of the presented reliability behavior of the basic systems under specific conditions to the reliability of the more complex systems such as NCS is proposed further.

5. Reliability and MTTF parameter approximation for NCS with shared redundant parts

The previous chapter presents two most used combinations of redundant subsystems was presented. At first the two component system was modeled and simulated in a shared redundant approach whereas three different failure rate behaviors are studied. The step change of the component's failure rate at the moment of the failure of another one is firstly introduced. However in the NCS there are several different situations where the failure rate can be changed depending on the time passed after the component's failure. Hence, general linear and non-linear (exponential) failure rate increasing is studied in order to propose an analytical description of the dependencies among the quasi-redundant components.

The triple modular redundant (TMR) system represents an often used redundant approach in order to increase the SIL level of the systems. Hence the second studied system is the TMR subsystem where a shared redundant approach is implemented. Thus, quasi-redundant subsystem composed of the three components where linear / nonlinear failure rate change is considered.

The aim of the previous study is to show the influence of the specific failure rate dependencies in order to derive an analytical method for estimation the dependability parameters of the more complex NCS. Hence, the first part of this chapter is to focus on the description of the dependencies and their identification in order to propose an analytical method which describes the reliability behavior of the shared redundant subsystems with dynamically changed failure rate. Thus, finally simple method for dependability analysis is proposed as an extension of the common known methods for dependability analysis which could not be used for NCS analysis. The limitation of the classical methods is determined by the inability to take into account some specific dependencies given by NCS environment.

The reconfiguration strategy of the NCS which covers quasi-redundant components is developed in the second part of the chapter. As it was described in the previous chapter, with consideration of the quasi-redundant parts of the NCS, several specific influences to failure rate appears. In consequence of this additional failure rate changes during the system mission, the reliability of the NCS decreases. Hence the second part

of this chapter focuses on the reconfiguration of the NCS in the case of a quasi-redundant component failure. It is necessary to define different approaches in order to reconfigure the NCS in case of different failure states. At least we should recognize the non-critical and critical failures and adapt the reconfiguration strategy to the actual states of the NCS.

5.1 Reliability approximation

Based on the MCS of the two most used redundant structures (two and three redundant components), the reliability approximation is provided. The main aim is to offer several simple equations which could be used in order to determine the reliability of the NCS with quasi-redundant subsystem (1-out-of-2, 1-out-of-3) where different types of failure rate changes are considered. At first the two-component QR subsystem is dealt whereas all FR behaviors mentioned in the fourth chapter are considered. Consequently the influence of the various decrease factors to the reliability of the three components QR subsystem is introduced.

The proposed method for reliability behavior approximation supposes that all quasi-redundant components have the same nominal failure rate or similar FR where differences are small and could be ignored. For example, the PCL controllers with the same parameters, etc.

5.1.1 Reliability approximation of the two-component QR subsystem with dynamic FR behaviors

As was abovementioned, the first considered two-component subsystem is further described. All mentioned FR behaviors (fourth chapter – single step, linear and exponential) are dealt whereas several reliability equations are proposed, in order to approximate the reliability behavior of the systems with time dependable failure rates.

Suppose two QR components with the nominal failure rate λ_n and defined the decrease factor d_R, then the reliability behavior of the QR subsystem composed of both components can be described as follows

$$R_{2qr}(t) = 1 - \prod_{i=1}^{2} \left(1 - e^{-(\lambda_n + k_i d_R)t}\right), \tag{5.1}$$

where k_i is the approximated coefficient in order to obtain the analytic results of the QR redundant subsystems which correspond with simulation results.

In the following text are described all mentioned dynamic behaviors of the FR change from single step change to non-linear exponential FR increasing. The tables with approximated coefficients are presented in order to propose a fast method for the reliability approximation of all considered QR subsystems.

5.1.1.1 Single step failure rate change

The first considered dynamic behavior of the nominal failure rate is a single step change described by equation 4.6 in the fourth chapter. The parameter decrease factor and approximated coefficients of equation 5.1 are shown in table 5.1. In each row of the table is shown the decrease factor with the corresponding value of the coefficients k_1 and k_2. The table shows several different values of the decrease factor whereas non-mentioned values can be easy approximated by using an appropriate method.

Decrease factor d_R	k_1	k_2
λ_n	0.44	0.52
$2\lambda_n$	0.39	0.395
$3\lambda_n$	0.28	0.393
$4\lambda_n$	0.198	0.434
$5\lambda_n$	0.154	0.46
$6\lambda_n$	0.13	0.4653
$7\lambda_n$	0.11	0.46
$8\lambda_n$	0.099	0.471
$9\lambda_n$	0.09	0.46
$10\lambda_n$	0.081	0.463
$20\lambda_n$	0.0445	0.38
$30\lambda_n$	0.0296	0.377
$40\lambda_n$	0.0225	0.385
$50\lambda_n$	0.0182	0.3518
$70\lambda_n$	0.0133	0.3284
$80\lambda_n$	0.011625	0.32475
$100\lambda_n$	0.0094	0.3332

Table 5.1 Parameters of equation 5.1 for a single step FR change

The maximal error of the approximation given by the parameters of the equation 5.1 is less than 1% of the maximal reliability value (thus less than 0.01). For the case when nominal FR is doubled, the following equation can be used in order to approximate the reliability behavior of the two-component QR subsystem

$$R_{2\lambda_n}(t) = 1 - \prod_{i=1}^{2}\left(1 - e^{-\left(\lambda_n + \frac{d_R}{2}\right)t}\right) \tag{5.2}$$

where d_R is the decrease factor and λ_n the nominal failure rate of the QR components. The exactness of this approximation is about 1%.

The error of all the approximations converge to the highest mentioned limits (1% for table's coefficients) in the bottom part of the reliability curves where the reliability of the system is smaller than 0.4.

5.1.1.2 Continuous linear failure rate change

The second considered dynamic FR behavior is a continuous linear FR change described by equation 4.9 (see chapter 4.2.2). In table 5.2 are shown the approximated coefficients for equation 5.1 corresponding to the decrease factor d_R.

Decrease factor d_R	k_1	k_2
0.0001	0	1
0,001	0.2	0.3
0,005	0.1	0.124
0,01	0.073	0.077
0,05	0.0238	0.0278
0,1	0.011	0.023
0,5	0.0018	0.013
1	0.0009	0.008

Table 5.2 Parameters of equation 5.1 for continuous linear FR increasing

The maximal approximation error is 5% in case of decrease factor $d_R = 10^{-2}$. This error corresponds to the approximated values corresponding with decrease factor $d_R = \{10^{-3},...,5.10^{-2}\}$ (shown in the table). For other mentioned decrease factors is maximal error less than 2% of the maximal reliability value.

5.1.1.3 Exponential failure rate change

The results of the reliability approximation of the QR subsystem with exponential FR change are shown in table 5.3. As in previous dynamic FR behaviors the coefficients k_1 and k_2 are approximated for the corresponding decrease factor considered in equation 4.12 mentioned in the fourth chapter (4.2.3).

Decrease factor d_R	k_1	k_2
0.00001	47.5	50
0.0001	5	5
0.001	0.61	0.62
0.01	0.1	0.14
0.1	0.009	0.086

Table 5.3 Parameters of equation 5.1 for exponential FR increasing

The maximal approximation error is less than 4% for $d_R = \{10^{-3}, 10^{-2}\}$. For other mentioned decrease factors the error is about 1%.

5.1.2 Reliability approximation of the three-component QR subsystem with dynamic FR behaviors

The second considered QR system composes of three subsystem whereas all mentioned dynamic FR behaviors presented above are evaluated too.

Suppose three QR components with the nominal failure rate λ_n and the defined decrease factor d_R, then the reliability behavior of the QR subsystem composed of both components can be described as follows

$$R_{3qr}(t) = 1 - \prod_{i=1}^{3} \left(1 - e^{-(\lambda_n + k_i d_R)t}\right), \tag{5.3}$$

where k_i is the approximated coefficient.

In the following text are described all mentioned dynamic behaviors of the FR change from single step change to non-linear exponential FR increasing. The tables with approximated coefficients are presented in order to propose a fast method for the reliability approximation of all considered dynamic FR behaviors of the three-component QR subsystems.

Table 5.4 shows the approximated coefficients for equation 5.3 for the considered QR subsystem with single step change of the nominal failure rate (eqv. 4.6). In tables 5.5 and 5.6 are shown the results of the approximation for the system with continuous linear FR increasing and exponential FR increasing described by equations 4.9 and 4.12.

Decrease factor d_R	k_1	k_2	k_3
λ_n	0.4	0.6	0.7
$2\lambda_n$	0.47	0.48	0.51
$3\lambda_n$	0.4	0.46	0.4633
$4\lambda_n$	0.405	0.38	0.4175
$5\lambda_n$	0.34	0.38	0.4
$6\lambda_n$	0.3166	0.333	0.4
$7\lambda_n$	0.2714	0.2857	0.4571
$8\lambda_n$	0.2375	0.2625	0.4875
$10\lambda_n$	0.16	0.36	0.41
$20\lambda_n$	0.14	0.21	0.215
$100\lambda_n$	0.0204	0.238	0.202

Table 5.4 Parameters of equation 5.3 for a single step FR change

The maximal approximation error in the case of the values shown in table 5.4 is less than 5% of the maximal reliability value 1.

Decrease factor d_R	k_1	k_2	k_3
0,001	0.11	0.4	0.4
0,01	0.09	0.096	0.104
0,1	0.0206	0.0236	0.0236

Table 5.5 Parameters of equation 5.3 for continuous linear FR increasing

In case of reliability approximation of the subsystem with continuous linear FR increasing (Table. 5.5), the maximal approximation error is about 7% of the maximal reliability value (thus 0.07).

Decrease factor d_R	k_1	k_2	k_3
0,00001	49	59	60
0,001	0.73	0.73	0.73
0,01	0.153	0.167	0.17
0,1	0.025	0.049	0.059

Table 5.6 Parameters of equation 5.3 for exponential FR increasing

The reliability approximation (Table 5.6) of the subsystem with non-linear exponential FR increasing gives the maximal error less than 6.5%.

As was above mentioned, the maximum differences between simulation results and approximated results are in the bottom part of the reliability curve where the reliability of the system is about 0.4 and less. In this part, the mean life time of the QR subsystems has already passed. Thus, we can suppose that the component will be replaced by a new

one in order to prevent the system failure and consequently this approximation differences have no significant influence to determine the final reliability of the system before the mean life time (*MTTF*) corresponding to a specified decrease factor. The life time of the simulated two an three components QR subsystems are studied in the following part of this chapter.

5.2 MTTF parameter approximation

Each quasi-redundant subsystem does not exceed the limits of the bound of the minimal ($MTTF_{min}$) and maximal time life ($MTTF_{max}$) of the quasi-redundant subsystem. The parameter $MTTF_{max}$ represents the maximal time life of the QR subsystem which could be obtained when the conditions are equal to the conditions of the subsystem with active redundant components. Thus, the nominal failure rate of the non-failed component is not changed after its load increasing – the case when the decrease factor is equal to zero. The lowest life time limit could be defined by parameter $MTTF_{min}$ which characterizes the subsystem composed of the independent components. Thus, when one of the components fails the system is considered as failed. In term of the decrease factor, it is equal to 1 or $(1-\lambda_n)$ for a single step FR change.

Let's suppose the system life time limited by the bound defined by *MTTF* parameter such as $\langle MTTF_{\min}; MTTF_{\max} \rangle$. These two parameters could be found by solving the simple equations (first chapter):

$$MTTF_{\min} = \int_0^{\infty} \prod_{i=1}^{n} R_i(t)dt \tag{5.4}$$

and

$$MTTF_{\max} = \int_0^{\infty} \left(1 - \prod_{i=1}^{n} (1 - R_i(t)) \right) dt \,, \tag{5.5}$$

where $R_i(t)$ is the reliability of each component (subsystem) of the NCS.

In the further text is described the life time increasing of the quasi-redundant subsystems with regard to the life time parameter $MTTF_{min}$ whereas various values of the decrease factor d_R are considered.

5.2.1 Two components QR subsystems

In order to propose simple and fast method for life time approximation, table 5.7 is prepared. The table shows the influence of the single step increasing of the nominal failure rate to the final life time of the two components QR system characterized by its *MTTF* parameter. In the first line, the failure rate of the non-failed QR component characterized by the multiple of the nominal failure rate λ_n. The second line shows the corresponding *MTTF* parameter percentage reduction within the limits defined by the abovementioned interval of the maximal and minimal life times (*MTTF*).

Single step change of the λ_n	$2\lambda_n$ $(d_R = \lambda_n)$	$3\lambda_n$ $(d_R = 2\lambda_n)$	$4\lambda_n$	$5\lambda_n$	$7\lambda_n$	$10\lambda_n$	$20\lambda_n$	$40\lambda_n$	$100\lambda_n$
Extended $MTTF_{min}$	50%	35%	25%	20%	15%	10%	5%	2%	1%

Table 5.7 Approximated values of the *MTTF* reduction of the two-component QR subsystem with different single step change of the nominal failure rate λ_n

The table could be used as a simple method in order to make fast the life time approximation of the quasi-redundant subsystems with single integer step change of the failure rate (chapter 4.2.1). In case of the complex NCS structures where a shared redundant approach is considered it can be very helpful in order to take decision among several different complex structures. The *MTTF* values introduced in table 5.7 are rounded hence the method error is about ±2 for the multiple of the nominal failure rate smaller or equal to $40\lambda_n$ (decrease factor $d_R < 40$). For higher value of the decrease factor, the approximated error is about ±1 of values shown in the table. Thus, in case of the very similar analysis result of considered complex structures it is necessary to prepare the exact model in order to obtain a more exact *MTTF* parameter reduction. This method could be used for the QR subsystems with the same failure rate or for the system when difference among the nominal failure rate λ_n of the components is very small and can be ignored. In case of the nominal FR smaller than 10^{-2} the increased value $100\lambda_n$ should represent approximately 0.1 whereas the error could be higher. Then, it could be useful that the value of nominal FR determined for a time interval T transform to the greater value for a shorter time interval (unit).

As was abovementioned, table 5.7 shows the approximated reduction of the *MTTF* parameter for a single step change of the nominal failure rate λ_n for non-failed QR

component. In the further text the approximation of the life time for a linear and non-linear FR increasing is introduced. Presented values can be used in order to achieve a faster approximation method.

Table 5.8 shows the life time reduction of the two-component quasi-redundant subsystems where the failure rate behavior describes by equation 4.9 (see part 4.2.2 – continuous linear FR increasing). As we can see the time life is increased non-linearly with regard to the parameter of the linear function (decrease factor d_R). As in previously presented results, they are rounded hence the error of this approximation could be about ±1 of values shown in the table.

Linear change of the λ_n – decrease factor d_R	$d_R = 10^{-4}$	$d_R = 10^{-3}$	$d_R = 5.10^{-3}$	$d_R = 10^{-2}$
Extended $MTTF_{min}$	92%	65%	40%	31%
Linear change of the λ_n – decrease factor d_R	$d_R = 5.10^{-2}$	$d_R = 10^{-1}$	$d_R = 5.10^{-1}$	$d_R = 1$
Extended $MTTF_{min}$	26%	11%	5%	4%

Table 5.8 Approximated values of the *MTTF* reduction of the two-component QR subsystem with linear change of the nominal failure rate λ_n defined by the decrease factor d_R

As we can see, with a relatively fast increasing of the FR ($d_R = 1$ – the nominal FR is constantly increased each time unit *Tu* about value of nominal FR λ_n) the system can benefit from an extended life time about approximately 4%. In case of the critical state of the system, it could need a significant time in order to set the system into the safe state and prevent if from undesired damage.

There is only a small difference between the results with continuous linear FR increasing and a linear step change of the FR. Hence, in order to show the life time behavior in figure 5.1, is presented the dependency of the *MTTF* parameter on the decrease factor which is considered as parameter of the linear step change equation (eqv. 4.10, 4.11). The considered redundant subsystem composes of two QR components with a nominal failure rate $\lambda_n = 10^{-3}$. As was mentioned, considered increasing of the FR follows the linear step behavior described in fourth chapter (eqv. 4.10, 4.11). Figure 5.1 shows the dependency between the decrease factor d_R and the life time of the system (*MTTF*).

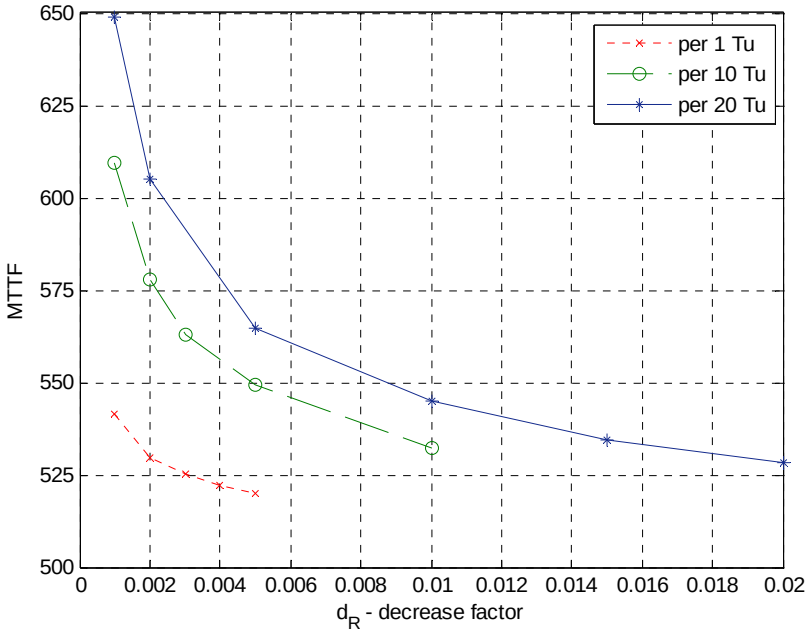


Fig. 5.1 Dependency between decrease function and life time (*MTTF*) of the two QR components system with a nominal failure rate $\lambda_n = 10^{-3}$

Several values shown in the figure 5.1 are mentioned in the fourth chapter (4.2.2). As in the case of the continuous linear increasing, the life time increases non-linearly depending on the specified decrease factor.

Exponential change of the λ_n – decrease factor d_R	$d_R \leq 10^{-5}$	$d_R = 10^{-4}$	$d_R = 10^{-3}$	$d_R = 10^{-2}$	$d_R = 10^{-1}$
Extended $MTTF_{min}$	50%	48%	40%	18%	4%

Table 5.9 Approximated values of the *MTTF* reduction of the two-component QR subsystem with exponential change of the nominal failure rate λ_n defined by the decrease factor d_R

Table 5.9 shows the life time increasing of the two-component QR subsystem where the FR behavior is described by the exponential equation 4.12. As we can see the system life time is increased about 4% however the decrease factor is ten times smaller than in the case of linear FR increasing. With value of d_R equal or smaller than 10^{-5} the life time of the system converges to the same value as in the case of the single step FR change with $d_R = \lambda_n$. This result confirms the correctness of the provided simulation because the value of changed FR at the moment of the failure is $2\lambda_n$ (see eqv. 4.12).

Thus, the $MTTF_{min}$ parameter can not be increased of more than in the case of a single step FR change (50%).

In the case of the FR change by following different exponential behavior defined as follows

$$\lambda'(t) = \lambda_n + d_R e^{(t-t_f)}, \tag{5.6}$$

the decrease factor values have to be about 10^{-6} in order to increasing of the $MTTF_{min}$ parameter only about 2%. This behavior is given by the fast exponential increasing where values about 10^6 are obtained after 14 time units. Thus the nominal failure rate is doubled after this time and the increasing is still faster. Several results of the $MTTF_{min}$ increasing are shown in table 5.10.

Exponential change of the λ_n (eqv. 5.4) – decrease factor d_R	$d_R = 10^{-25}$	$d_R = 10^{-20}$	$d_R = 10^{-10}$	$d_R = 10^{-6}$	$d_R = 10^{-3}$
Extended $MTTF_{min}$	6%	5%	3%	2%	1%

Table 5.10 Approximated values of the MTTF reduction of the two-component QR subsystem with an exponential change of the nominal failure rate λ_n following equation 5.4

We can suppose the similar results in case of the three-component QR subsystems hence this behavior will not be further considered.

5.2.2 Three components QR subsystems

In the following text, the approximation of the *MTTF* parameter of the three-component quasi-redundant subsystem is introduced. Table 5.11 shows the influence of the shared redundant approach for three-component system to the increasing of its time life. As we can see, even if the failure rate increases about eleven times, the system life time is extended about 10%. Approximately 25% more than in the case of the system FR is increased about five times.

Single step change of the λ_n	$2\lambda_n$ $(d_R = \lambda_n)$	$3\lambda_n$ $(d_R = 2\lambda_n)$	$4\lambda_n$	$5\lambda_n$	$6\lambda_n$	$8\lambda_n$	$11\lambda_n$	$25\lambda_n$	$100\lambda_n$
Extended $MTTF_{min}$	55%	40%	30%	25%	20%	15%	10%	5%	1%

Table 5.11 Approximated values of the *MTTF* reduction of the three-component QR subsystem with different single step change of the nominal failure rate λ_n

The same as in the case of a two-component system the increasing the nominal FR the hundred times cause only a minimal extension of the life time of the subsystem (approximately 1%). Table 5.12 shows simulation results for a continuous linear FR increasing (eqv. 4.9). The time life extension to the decrease factor around 10^{-4} is more than 90%. On the other side with increasing the value of the decrease factor (d_R=1 and more) the life time is extended to less 10%.

Linear change of the λ_n – decrease factor d_R	$d_R = 10^{-4}$	$d_R = 10^{-3}$	$d_R = 5.10^{-3}$	$d_R = 10^{-2}$
Extended $MTTF_{min}$	93%	70%	45%	36%
Linear change of the λ_n – decrease factor d_R	$d_R = 5.10^{-2}$	$d_R = 10^{-1}$	$d_R = 5.10^{-1}$	$d_R = 1$
Extended $MTTF_{min}$	31%	15%	10%	8%

Table 5.12 Approximated values of the *MTTF* reduction of the three-component QR subsystem with a linear change of the nominal failure rate λ_n defined by decrease factor d_R

Exponential change of the λ_n – decrease factor d_R	$d_R \leq 10^{-5}$	$d_R = 10^{-3}$	$d_R = 10^{-2}$	$d_R = 10^{-1}$
Extended $MTTF_{min}$	55%	46%	22%	5%

Table 5.13 Approximated values of the *MTTF* reduction of the three-component QR subsystem with an exponential change of the nominal failure rate λ_n defined by the decrease factor d_R

Third considered time dependable behavior of the FR of the non-failed quasi-redundant component is non-linear exponential FR increasing. This behavior is described by equation 4.12 mentioned in fourth chapter (Table 5.13). As in the case of the two-component quasi-redundant subsystem, the maximal life time extension is equal to 55% which corresponds to the value obtained with single step change of the FR when the nominal FR doubles. In case of an exponential increasing, this value corresponds to a decrease factor smaller or equal than 10^{-5}.

The presented results show the influence of the different types of the FR behaviors on the life time extension of the two most used redundant combinations such as two and three component redundant subsystems. The described results propose a simple and fast method in order to determine the influence of the time dependable FR to the final life time of the subsystem. This approach can be very useful especially in the case of the more complex structures where the fast decision among several different NCS structures

should be taken. Thus, the prediction of the time interval needed to take the system into a safe state can be quickly achieved for the mentioned quasi-redundant structures.

5.3 Study of the controlled subsystem loading with direct consequence to the reliability and life time

Presented studies and results take into account spontaneous loading of the quasi-redundant subsystem S_k just after the subsystem S_n failure. However, the networked control systems compose of some parts where loading could be controlled. If the NCS allows benefiting from quasi-redundant networks then the load control approach could be applied.

A primary requirement in order to benefit from quasi-redundant networks is to prepare some reconfiguration mechanism which is executed just after the network failure, when the communication of some of the components has to be retransmitted through other non-failed networks. The reconfiguration mechanism should take into account all important physical parameters of the network such as bit rate, maximal number of connected components, etc. From a physical point of view, it is necessary that all potentially reconfigured components should be connected to both (all) networks which are considered as quasi-redundant. As it is shown in figure 5.2 the components S_1, S_2, C_1, C_2, A have to be connected in both networks if the shared redundancy can be applied on this networked control structure.

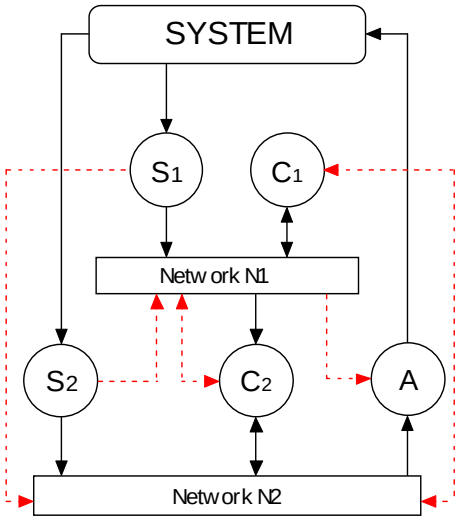


Fig. 5.2 NCCS with connections after network failure

As we can see from the description of the quasi-redundant components (chapter 3.3), the alternative physical connection of the components could be considered even if the bit rate of the shared network is insufficient. In this case there are two possible scenarios. At first we can suppose that, even if the network capacity is insufficient, the communication will run for a short time interval which allows switch the system into the safe mode. A second scenario is to solve the problem of the insufficient bit rate by a controlled reconfiguration mechanism.

The first scenario could be hazardous because the mentioned short interval could have randomly long duration whereas it can disallow a guarantee that the system will have enough time in order to remain in the safe mode. On the other hand, the controlled reconfiguration can specify a time interval which can be taken into account in order to protect the system against the damage.

Hence in further text several different scenarios are studied in order to propose a reconfiguration strategy which minimizes the reliability reduction of the shared redundant components. Studied scenarios are as follows:

- Step change of the failure rate up to a specified value for a time interval.
- Gradually increased failure rate during a specified time interval until the maximal value of the failure rate.
- Periodically failure rate increasing and decreasing between its nominal and maximal values.

In the case of the first scenario, the failure rate of the quasi-redundant subsystem is increased by step up to the maximal value whereas after a specified time interval the loading of the components is eliminated and the failure rate is returned to its nominal value. This controlled loading of the subsystem is periodically repeated until the system failure.

The next situation which is studied is the gradual increasing of the failure rate until specified value. When the maximal limit is obtained then the failure rate decreases down to the nominal value of the failure rate for a defined time.

The last mentioned scenario supposes a continuous load increasing and decreasing with a defined periodicity similarly like sine or cosine functions. The two different approaches are compared. In the first, the step subsystem loading is supposed whereas loading is continuously decreased down to a limit which is characterized by the nominal failure rate (cosine function). The second one supposes continuously loading up to the

high limit of the failure rate, then the loading is continuously decreased down to the value of the nominal failure rate. As it was already mentioned, all changes of the failure rate of the quasi-redundant subsystems are periodical until the system failure.

Finally, the results of all the mentioned scenarios are compared in order to propose the best strategy for reconfiguration of the NCS based on the principle of the load control.

5.3.1 Scenarios simulation

This section is dedicated to the example of two quasi-redundant subsystems (networks N_1, N_2). The prepared PN model of the studied system corresponds to the two-component model described in the previous chapter (4.2). In order to implement the required functionality of the periodical scenario repetition until the system failure, several modifications are proposed.

Figure 5.3 shows the model of the subsystem S_1. In order to modify the functionality of the subsystem the function *changeFR(...)* for a periodical change of the failure rate is prepared. Similarly, the subsystem S_2 is modified. The function with different parameters is called when the transition "*Subsystem 1*" is fired. The detailed implementation of the function for a gradual failure rate increasing (Fig 5.4d) is as follows:

```
fun changeFR(pb,i,queue,pack,n,d_val) =
let
val (Tu,exp_val) = if i > (Tfr_n+Tfr_i-1) then (1, decr_val)
   else if i > (Tfr_i-1) then (i+1,0.0)
       else if round(d_val*1000.0) = 0 then (0,0.0)
            else (i+1,Real.fromInt(i+1)*decr_val);
val p_inc = Real.fromInt(pb)+exp_val;
val r_n = Real.fromInt(n);
val rand_val = exponential(p_inc/r_n);
val queue_1 = if floor(rand_val) < 1
then queue^^[pack] else queue;
val proba = round(p_inc);
in
(proba,queue_1,Tu)
end;
```

The input parameters of the function are *pb* which represents the parameter of the exponential distribution function and defines the actual failure rate of the subsystem. Parameter *i* is a counter used for a periodical execution of the simulated scenario. Variable *queue* store the information about subsystems' failure where the identifier of

the subsystem failure is stored in the input variable *pack*. Parameters *n* and *d_val* are used for generating the FR due to a simulated scenario.

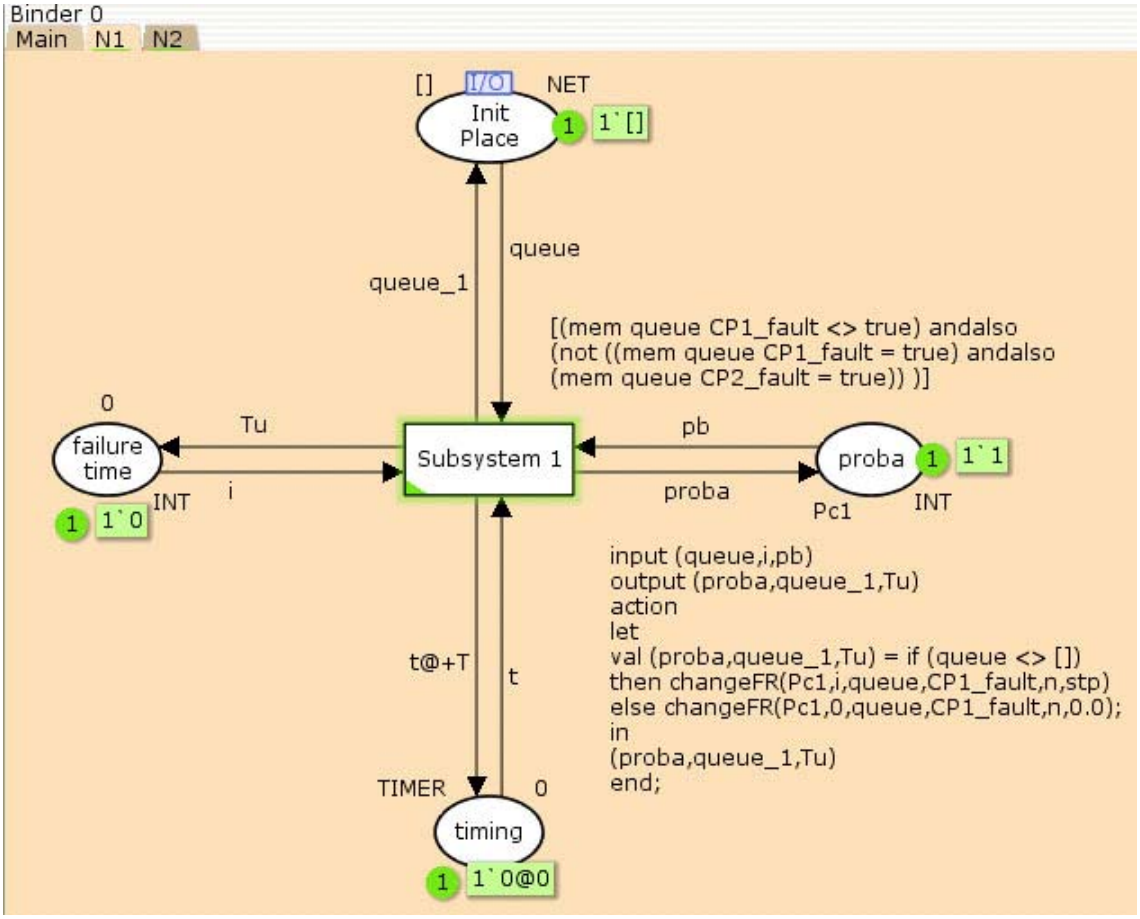


Fig. 5.3 Modified model of the subsystem

Described function gradually increases FR (see Fig 5.4d) during the time interval specified by variable *Tfr_i*. Time interval when FR of the system has nominal valie is defined by the variable *Tfr_n*. In case of the failure rate increasing by following the sine and the cosine function only this function is added.

5.3.2 Scenarios evaluation

In further study the presumptions are based on the example of two shared redundant networks (subsystems N_1 and N_2). As was mentioned three supposed scenarios were simulated where the FR is periodically changed following these behaviors:

- Step change - rectangle curve (Fig 5.4b, c and 5.5b) – Scenario I.
- Gradual increasing and step reduction (Fig 5.4d and 5.5c) – Scenario II.

- Gradual increasing and decreasing – sine and cosine functions (Fig 5.6b) – Scenario III.

The periodicity of all the simulated scenarios is ten time units. This time interval is chosen due to its sufficient length in order to demonstrate all scenarios. Following the simulated scenario the FR of the components are increased within the interval $\lambda \in <10^{-3}, 6.10^{-3}>$. Thus, the maximal decrease factor d_R is 0.005 (as equivalent to single FR change) what represents increasing the load of the network which increases the probably of the failure of the network five times.

The FR curve which corresponds to the studied scenario is defined by two parameters which specify the time interval T_i when the network load increases and the time interval T_n when the network load is nominal (nominal FR $\lambda_n = 10^{-3}$). Depending on the simulated scenario the FR increases gradually or by step during the time interval T_i. The time intervals T_i and T_n have to be determined based on the parameters of the controlled system such as dynamics, etc.

Figure 5.4a shows the comparison of the reliability curves for two different scenarios - step change (Fig 5.4b, c) and gradual increasing (Fig 5.4d) of the FR. As we can see the scenario shown in figure 5.4c supposes the increasing of the network loading up to the maximal limit ($\lambda_{max} = 6.10^{-3}$) during nine time units and for one *Tu* the network load decreases back to the nominal value ($\lambda_n = 10^{-3}$). This scenario covers the situation when the control network fails and the components connected to this network immediately start transmitting the required data through a non-failed network. After a specified number of the sampling periods (time units) several nodes are excluded from the communication in order to decrease the failure rate to the nominal value. This communication blocking some of the nodes is held during the time interval T_n.

In order to develop the mentioned behavior of component communication through the network, it is necessary to implement a network scheduling technique: for example, the (m,k)-firm [Lin 06] approach which is used in order to guarantee the minimum level of the quality of service (QoS) could be used. This technique defines the constraints for the data transition in order to guarantee the transmission of more important data.

In further text and figures two different situations are considered:

- The communication of all components actually connected to the non-failed network has a longer duration than the time of blocking of some of the components ($T_i > T_n$).

- Opposite to the first case

Thus, in the first situation, the network failure rate is higher for a longer time interval than the nominal FR. In the second case, the network loading is increased for a shorter time interval in comparison with the time when the network is normally loaded (nominal FR).

Figure 5.4 shows the two considered scenarios which suppose step and gradual change of the failure rate up to the maximal limit ($\lambda_{max} = 6.10^{-3}$) for nine time units. For the step FR change, we can use the equivalent example when the newly connected nodes to the network starts sending the data immediately and it causes the step change of the network loading up to a limit which corresponds to a failure rate level $\lambda_{max} = 6.10^{-3}$. After nine time units several components stop the communication in order to decrease the network load and reduce the FR to the nominal value ($\lambda_n = 10^{-3}$). After one time when this communication interruption continues the scenario is repeated until the system failure. Thus, time intervals ratio T_i/T_n is 9/1. As in previously presented figures with reliability curves, there are shown highest and lowest reliability values which are equivalent to a system with classical active redundant systems (top long-dashed curve – $MTTF = 1503Tu$) and system without redundancy (bottom long-dashed curve – $MTTF = 498Tu$).

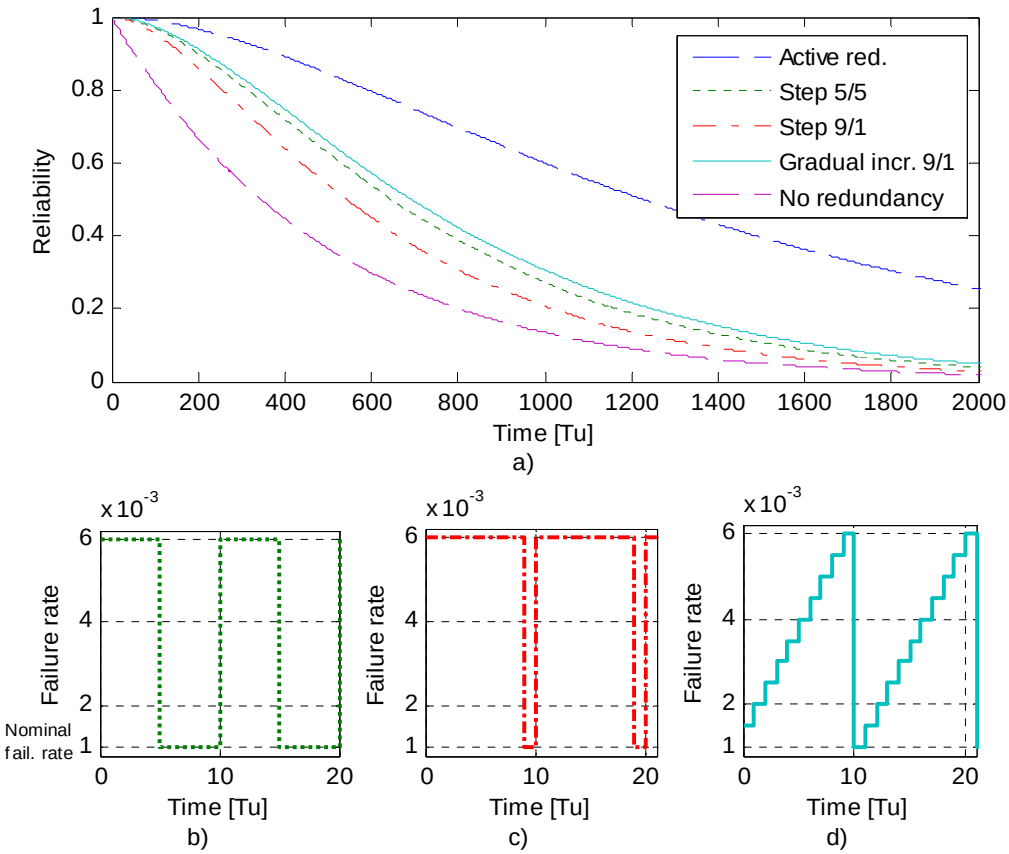


Fig. 5.4 a) Comparison of two different scenario of the subsystem loading, b) step FR change with time ratio T_i/T_n = 9/1 c) step FR change - T_i/T_n = 5/5 d) gradual FR increasing T_i/T_n = 9/1

As we can see from figure 5.4a, the best reliability result is obtained by the implementation of the scenario of gradual FR increasing (non-dashed reliability curve) where it is supposed that components are gradually connected to the non-failed network whereas consequentially the network load is increased together with network probability of failure defined by the failure rate (Fig. 5.4d).

Step scenario with the same parameters (T_i/T_n = 9/1) gives significantly worse results what can be also seen in the high difference between the life times of the system (Table 5.14). As we can see the difference is 155 time units whose could be very useful especially in case of critical failure.

FR change	Step change 5/5	Step change 9/1	Gradual incr. 9/1
MTTF [*Tu*]	783 *Tu*	678 *Tu*	833 *Tu*

Table 5.14 System life time by following scenarios shown in figure 5.4

By step scenario could be obtained the same result as in the case of gradual scenario however the parameters have to be changed. In figure 5.4b there is an example of the step scenario where the time intervals ratio is 5/5. Thus, the time of higher network loading is shorter whereas the time of nominal loading was extended. As we can see from table 5.14 the life time has been increased however it still does not obtain the level of the gradual scenario with the worse time interval ratio (9/1). Thus, in order to obtain equivalent reliability results as well as life time, the time T_i when the network loading increases should be shorter than half the periodic time interval given by both values T_i and T_n.

In next figure (Fig. 5.5), the same scenarios are compared however with a shorter period of increased FR T_i. Figure 5.5a shows four reliability curves. Two dashed curves correspond to the scenarios shown in figures 5.5b,c. As we can see, the gradual FR increasing gives a higher reliability level during the system life time than its ratio equivalent (2/8) with the step changed FR.

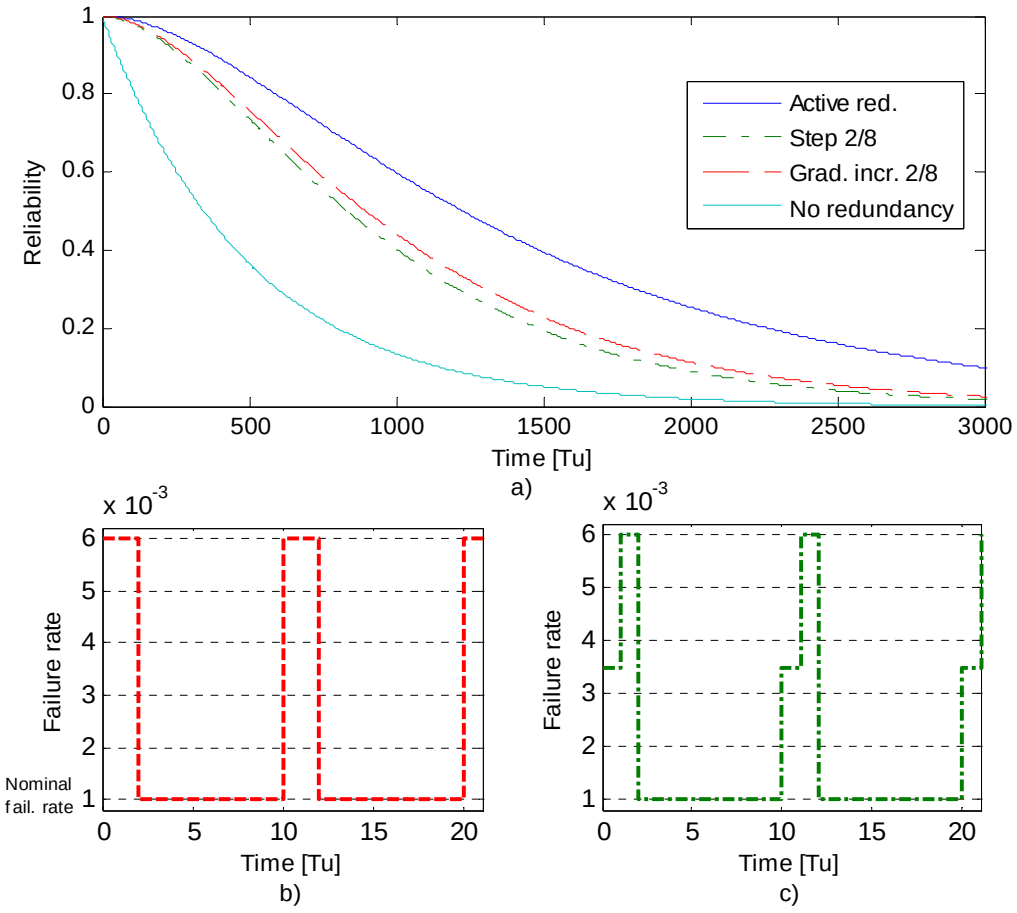


Fig. 5.5 a) Comparison of two different scenarios of the subsystem loading, b) step FR change with time ratio T_i/T_n = 2/8 c) gradual FR increasing T_i/T_n = 2/8

The scenarios shown in figure 5.5 have time intervals ratio 2/8 whereas additional value is added into the table 5.15 in order to shown equivalent of step FR change scenario (2/8). As we can see from table the gradual increasing of the FR during the two time units (Fig. 5.5b) corresponds to a life time (*MTTF*) 1070 *Tu* which represents more than 7% longer life time than in case of step increased FR with duration two time units (Fig. 5.5b). Similar result as in the case of "step change 2/8" is obtained by the gradual increasing with a time intervals ratio 3/7 (T_i/T_n). Thus, the network load can be increased during longer time interval (3*Tu*) than the network with immediately connected components even if the communication is shorter (2*Tu*). The simulation result for a *MTTF* parameter are 995*Tu* for step FR change with ratio 2/8 and 998*Tu* for gradual FR change with ratio 3/7.

FR change	Step change 2/8	Gradual incr.2/8	Gradual incr. 3/7
MTTF [*Tu*]	995 *Tu*	1070 *Tu*	998 *Tu*

Table 5.15 System life time by following scenarios shown in figure 5.5

The last simulated scenario is a gradual increasing and decreasing of the failure rate where two variants are considered:

- With step FR change at the moment of the network failure and system reconfiguration – cosine function.
- With gradual FR change at the moment of the network failure and system reconfiguration – sine function.

Thus, the failure rate behavior follows the sine and cosine function whereas the period of scenario repetition is the same as in previous examples ten time units (Fig. 5.6b).

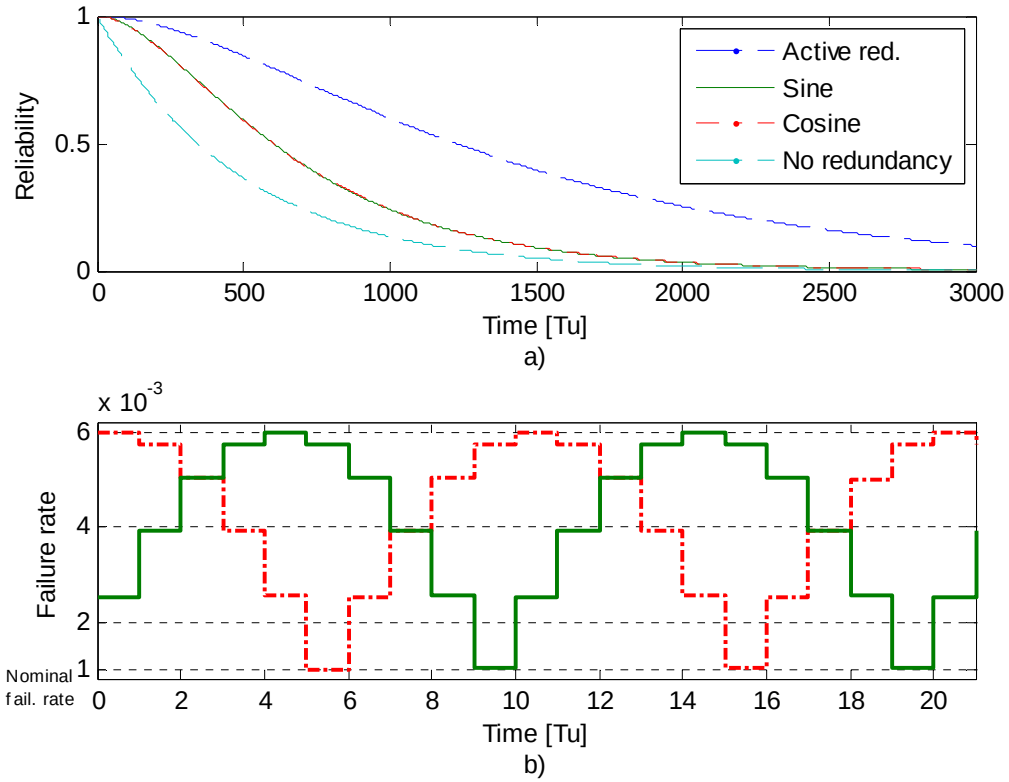


Fig. 5.6 Gradual increasing and decreasing scenario – following sine and cosine behavior

As we can see, there is a minimal difference between reliability behaviors of the simulated scenarios. From table 5.16 we can see that difference between life times is smaller than one time unit. Thus as we can see FR step change at the beginning of the subsystem loading has not influence to the life time of the studied system for relatively small change of the FR such as in studied example ($\lambda_n = 10^{-3}$, $\lambda_{max} = 6.10^{-3}$). Also in case of enormous FR step change (hundred times higher FR than nominal FR) at the moment of first loading (cosine) the influence to the life time of the system is minimal. Increasing the FR hundred times practically means that second network is failed almost immediately after the first one fail. The simulation results (*MTTF*) for $\lambda_{max} = 10^{-2}$ are shown in the table 5.16. The lowest limit for the life time of the studied system is 499 *Tu* (corresponding with bottom dashed reliability curve in figure 5.6a – system without consideration of the redundancy) whereas life time of the system with mentioned scenario where $\lambda_{max} = 10^{-2}$ is only about 15*Tu* longer.

FR change	Sine $\lambda_{max} = 6.10^{-3}$	Cosine $\lambda_{max} = 6.10^{-3}$	Sine $\lambda_{max} = 10^{-2}$	Cosine $\lambda_{max} = 10^{-2}$
MTTF [*Tu*]	737.9 *Tu*	737.32 *Tu*	513.3 *Tu*	514.5 *Tu*

Table 5.16 System life time by following scenarios shown in figure 5.6

The difference between both scenario results for $\lambda_{max} = 10^{-2}$ is minimal (about one time unit). Thus, we can see that there is no difference between sine and cosine scenarios even if FR rate of the subsystem is hugely increased.

In comparison with previous simulated scenarios the results could be compared with step FR change scenario with ratio 5/5 (783 *Tu*) because the difference between the life time is lowest in comparison with other scenarios results. On the other side the results of sine / cosine reconfiguration strategy are better than in the case of ratio 9/1 (step FR change).

As we can see from presented results the best way in order to prolong the life time of the system is to control the increasing of the network load gradually until the maximal level and than decrease the FR down to the nominal FR for a several time units defined by parameters of the controlled system. When this scenario can not be applied there are other possibilities which kind of reconfiguration strategy should be chosen.

5.3.3 Reconfiguration strategy

The main aim of the reconfiguration strategy is to reduce the influence of the increased failure rate due to the implementation of a shared redundancy approach within an NCS. In order to apply the reconfiguration strategy, it is necessary to implement several features:

- To connect all components in each network which is considered as quasi-redundant.
- To implement the network scheduling.

If quasi-redundant networks are considered, then multi-connection of all necessary components is required in order to implement the shared redundancy approach and reconfiguration strategy as well. The reconfiguration strategy requires a scheduling of the NCS's components. Thus, some level of intelligence should be considered. Hence, a multi-agent [Woo 95] approach could be very useful in order to design NCS with the SR approach and a suitable reconfiguration strategy.

In order to realize the reconfiguration strategy with gradual FR increasing, the priority of the connected nodes should be specified. This reconfiguration strategy could

be very useful in case of the critical failures when the network can be overloaded. Then a gradual connection and step disconnection of the nodes can help the control of the network loading and consequently extend the system life time.

Figure 5.7 shows a diagram in order to choose a suitable strategy where several conditions are considered. When more than one strategy can be considered, the order of the strategy scenarios is shown from the highest to the lowest priority. As we can see before the reconfiguration strategy implementation it is necessary to determine abovementioned time intervals specified by a variables T_n and T_i. This time intervals are derived from information about maximal time extension between two sent samples by specific node which is considered as potentially connected and disconnected. Thus how many samples could be ignored whereas the system remains in a stable state? If the T_i interval is not greater than one time unit Tu, then only the first scenario can be provided (Scenario I) where the step load (FR) change is considered.

Generally two basic states should be recognized. If we consider that the failure rate of the network is defined as a failure per n transmitted bits, then only a single step FR change can be considered (transmitted higher number of bits within the specified time interval). Thus, the load of the network is increased but network parameters are sufficient to transfer all required data. In this case all reconfiguration strategies can be implemented.

However, when the network load significantly increases and consequently, the network is overloaded, then the strategy with the step change of the load is not very useful. Hence, as we can see from scenario simulation results, the gradual load (FR) increasing is most suitable in order to extend the time life of the NCS. The third scenario with a gradual increasing and a gradual decreasing of the load can be considered as well.

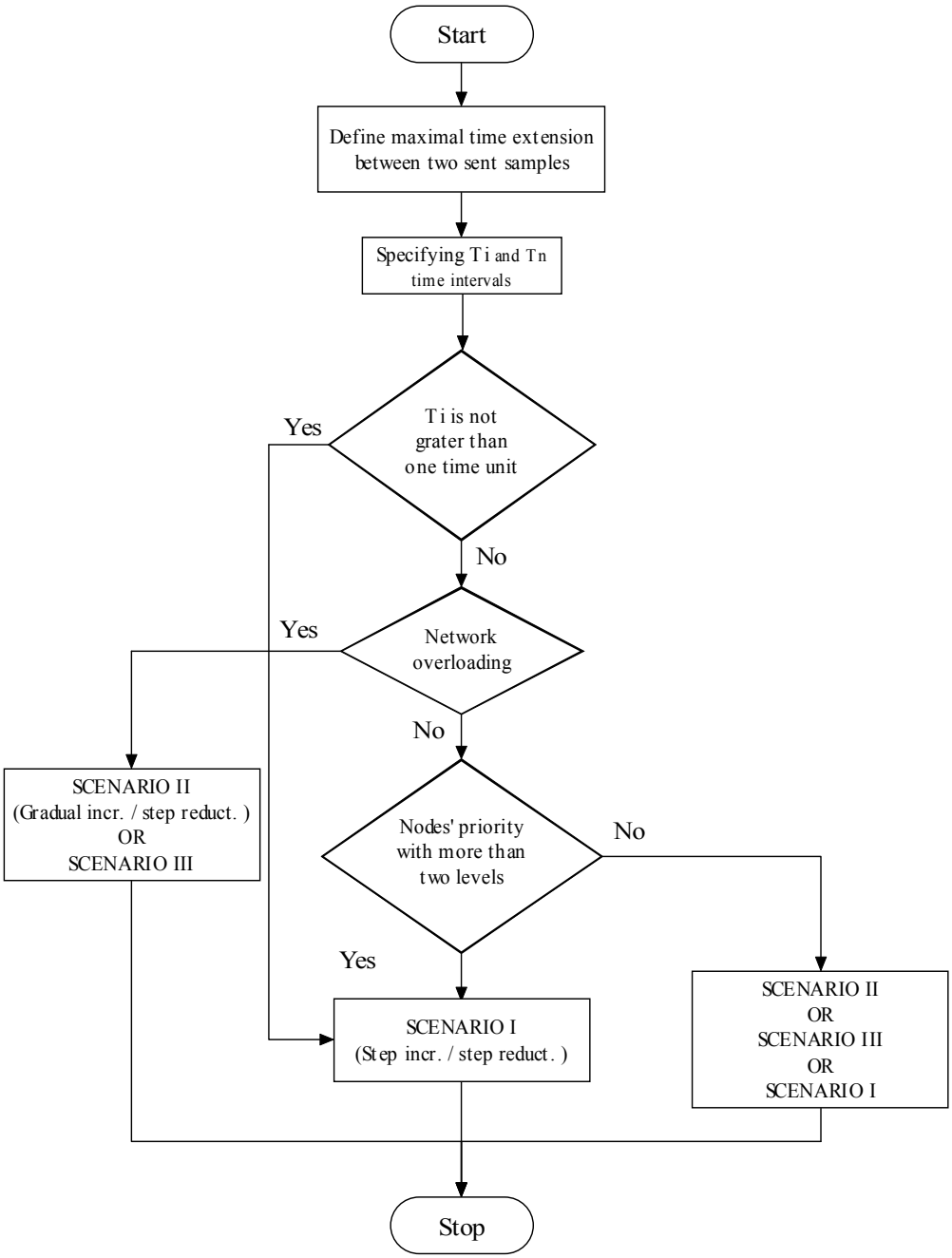


Fig. 5.7 The reconfiguration strategy diagram

The property which should be taken into account is the implementation of some level of the intelligence in order to implement some kind of the reconfiguration strategy. If we consider only the system where it is difficult to implement additional reconfiguration scheduling in order to obtain one of the gradual load increasing scenarios, then only one step load change could be implemented. For this scenario it is not necessary to define the nodes with different priorities. It is sufficient to divide the

components into two groups where one of them is periodically disconnected without any additional higher scheduling process.

In case of a gradual load increasing scenarios the nodes with different priorities have to be considered which could require additional logic implemented within the NCS. In some kind of NCS where SR approach with the reconfiguration strategy has to be implemented, it could be difficult to implement through several hardware as well as software constraints.

5.4 Conclusion

The fifth chapter focuses on the approximation of the dependability of the previously studied two and three components quasi-redundant subsystems. In the first part of the chapter, the reliability approximation is proposed as a simple method for the reliability evaluation of the QR subsystems with considered dynamic failure rate behaviors such as a single step FR change, a continuous linear FR change and a nonlinear exponential FR change.

The method proposed in this chapter consider reliability equations with different coefficients corresponding with a decrease factor which represents the parameter of the equation (4.6, 4.9, 4.12) described in the fourth chapter. The presented tables compose of necessary coefficients for equation 5.1 and 5.3 which allow approximating the reliability behavior of the mentioned quasi-redundant systems. Non considered values of the decrease factor and corresponding coefficients can be easily derived from other values shown in the tables. Proposed reliability approximation method has several limitations of its using. The main required condition is that the failure rates of the quasi-redundant subsystems are identical or similar whereas differences could be ignored. This condition of the same nominal failure rate could be accomplished within different NCS because we can suppose that quasi-redundant subsystems have similar or identical hardware. As we can see, the reliability behavior of the two and three component QR subsystems can be approximated. In case of system with different FR behavior specified by decrease factor which is not included by tables of coefficients it is necessary to approximate the coefficients.

Similarly, the method for the life time approximation of the studied subsystems is proposed in the second part of the chapter. The life time is characterized by a *MTTF* parameter whereas percentage extension of the life time of the non redundant subsystems is presented.

The shared redundancy approach can significantly increase the reliability and extend the life time of the system as well. However this advantage can be additionally improved in order to obtain better reliability results when a suitable reconfiguration strategy is used. Hence, the last part of this chapter focuses on the reconfiguration strategy of the NCS. Several strategies are compared and evaluated for different cases of the system abilities (consideration of nodes' priority, etc.).

Thus, the implementation of a shared redundant approach in combination with a suitable reconfiguration strategy allows significantly improving of the dependability attributes and parameters.

6. Application of the proposed method for dependability evaluation of the more complex systems

Based on the study presented in the fourth chapter, several different methods and methodological technique for dependability analysis are proposed in the fifth chapter. The analysis focuses on the reliability and life time (*MTTF* parameter) approximation. Previously introduced and described techniques are demonstrated in this chapter. Several examples of more complex networked control structures are proposed in order to verify the results obtained by using the proposed method for the dependability analysis.

Thus, following the decrease factor, the parameter of failure rate function, the reliability reduction as well as the *MTTF* parameter can be approximated. With further application of the proposed reconfiguration strategy, this reduction could be partially eliminated.

At first the basic cascade control structure is evaluated. Proposed cascade structure is considered as networked cascade control system (NCCS). The NCCS composes of two quasi-redundant controllers (Fig. 6.1) whereas further the structures with two control networks are studied. Thus, two different structures with different numbers of quasi-redundant subsystems are analyzed. In the first case the quasi-redundant controllers are considered. The second structure composes of two different groups of components such as two networks and two controllers.

In the second part of the chapter more complex structure of the mini-helicopter control structure is evaluated. This structure was chosen due to the high complexity of the control structure given by the number of components and different types of quasi-redundant components. As was mentioned, two different control structures of the mini-helicopter are evaluated – with one network and two control networks. Each example is evaluated by using Petri Nets model whereas the results of the Monte Carlo simulation are compared with results given by developed methods for dependability analysis.

6.1.1 Reliability approximation of the cascade application with consideration of the SR subsystems

Based on the simulation results of the basic quasi-redundant structures introduced in the fourth chapter as well as the generalized equation presented in the fifth chapter, the

dependability analysis of the networked cascade control structure is provided. This control structure is discussed in the third chapter (3.2.1.1). It composes of five components and one network. This control structure was chosen for its wide application in industrial applications. Hence the results of the simulation are determined finally in order to verify the reliability approximation method mentioned in fifth chapter. In addition, they can be useful when shared redundant approach in common networked cascade control structure is implemented.

As it was above mentioned, the first example is considered simple networked cascade control structure with one network shown in figure 6.1. In this control structure there are two quasi-redundant controllers are considered – C_1 and C_2. All three previously presented failure rate behaviors is tested and evaluated – single step, linear and exponential failure rate change. The simulation results are shown in figure 6.2. Each dashed curve corresponds with the decrease factor which represents failure rate change. Decrease factor of the single step FR change is equal to 10^{-3} (long-dashed reliability curve). In case of the linear (short-dashed reliability curve) and exponential FR increasing the decrease factors are determined to value 10^{-1} (dashed and dotted reliability curve).

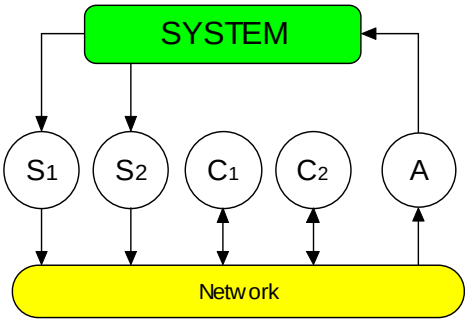


Fig. 6.1 NCCS with one network

All simulated FR behaviors and their influence to the final reliability of the simulated system are used for verification of the proposed method (fifth chapter) for reliability approximation of the NCS with shared redundant approach implementation. Thus, in order to verify proposed coefficients and equation, the reliability curve shown in the figure 6.2 are further approximated. In the simulated example is considered the same nominal failure rate $\lambda_n = 10^{-3}$ for all components of the NCCS.

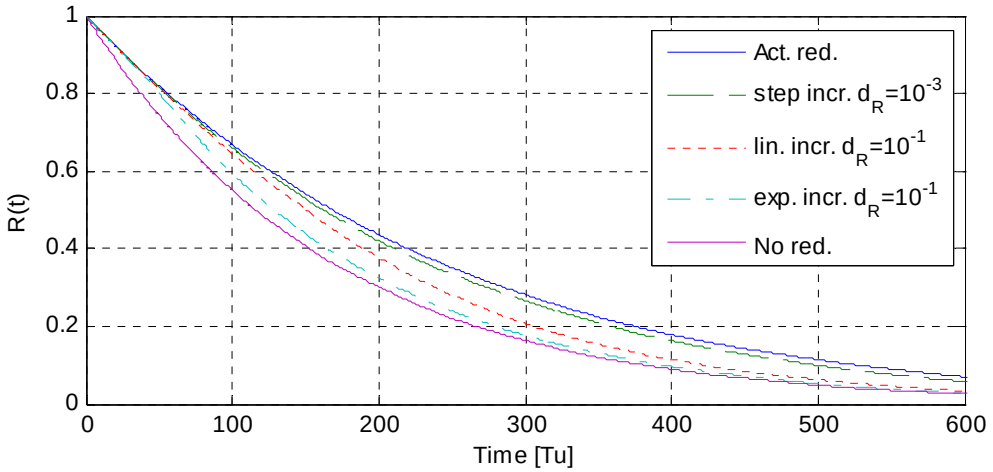


Fig. 6.2 Reliability curves of the NCCS with one network with different failure rate behaviors

As was abovementioned, the reliability approximation is provided by using the proposed coefficients (see tables in chapter 5.1) in presented equations (eqv. 5.1). In case of the single step change of the failure rate specified by the decrease factor d_R=10^{-3} the table coefficients are k_1 = 0.44 and k_2 = 0.52. Then, the reliability of this system composed of four independent subsystems (sensors S_1, S_2, actuator A and network N) and two quasi redundant controllers is as follows:

$$R_{NCCS} = R_{S1}(t)R_{S2}(t)R_A(t)R_N(t)R_{2qr}(t), \tag{6.1}$$

where $R_{2qr}(t)$ is equation 5.1 which describes the reliability of the two quasi redundant subsystems. Hence,

$$R_{NCCS} = e^{-4\lambda_n t}\left(1-\prod_{i=1}^{2}\left(1-e^{-(\lambda_n+k_i d_R)t}\right)\right). \tag{6.2}$$

With adding the exact values of the failure rates and table coefficients (k_1 = 0.44 and k_2 = 0.52) to the equation 5.2, the final reliability of the system is represented by following equation:

$$R_{NCCS} = e^{-410^{-3}t}\left(1-\left(1-e^{-\left(10^{-3}+0.44*10^{-3}\right)t}\right)\left(1-e^{-\left(10^{-3}+0.52*10^{-3}\right)t}\right)\right). \tag{6.3}$$

The approximation results given by equation 6.3 are shown in figure 6.3a. As we can see the results are almost identical whereas maximal absolute difference between the Monte Carlo simulation results and proposed equation 6.3 is only 3.89 10^{-3} (see table 6.2).

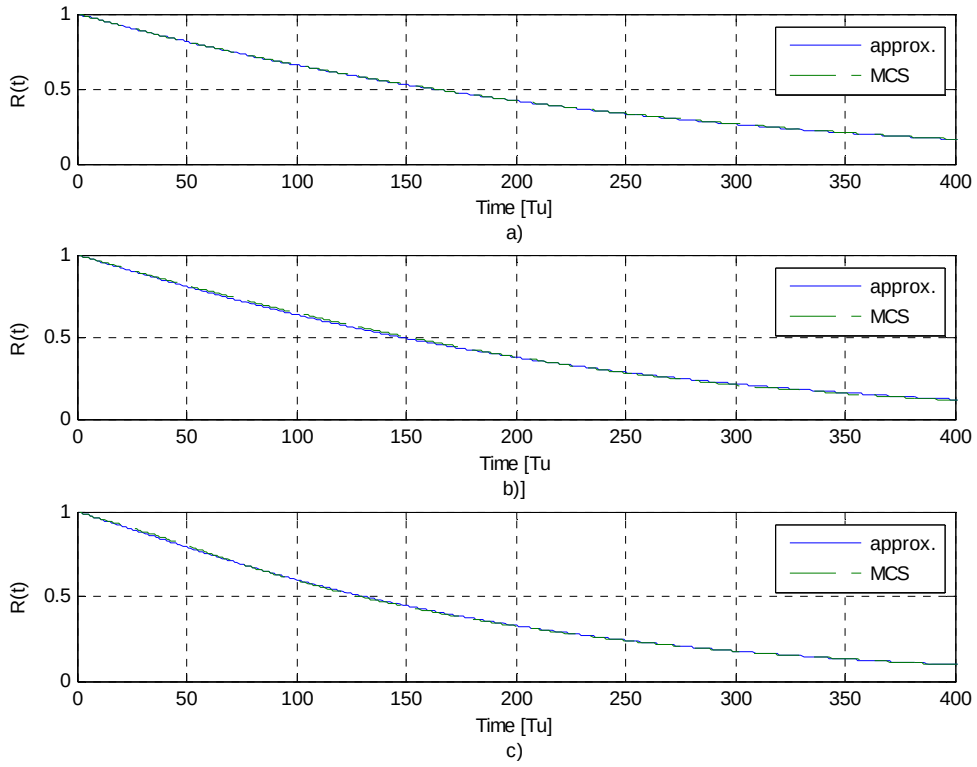


Fig. 6.3 Approximated reliability curves of the NCCS with one network for a different FR change a) single step change dR = 10^{-3} b) linear FR change dR = 10^{-1} c) exp. FR change dR = 10^{-1}

The similar technique is used for approximation the linear and exponential FR change by using specified decrease factor $d_R = 10^{-1}$. Similarly, corresponding coefficients k_1 and k_2 with decrease factor are chosen from tables 5.2 and 5.3. The figure 6.2b, c shows the approximation results of the reliability of the considered NCCS. The error (maximal absolute difference – *MAD*) is shown in table 6.2. Table 6.1 shows the life times of the considered system characterized by *MTTF* dependability parameter. Parameter *MTTF* represents life time given by MCS which is compared with $MTTF_{app}$ parameter given by approximated reliability results.

Decrease factor d_R	Active red.	Step incr. $d_R = 10^{-3}$	Linear incr. $d_R = 10^{-1}$	Exponential incr. $d_R = 10^{-1}$	No red.
$MTTF\ [Tu]$	232.7	221.4	194.6	177.5	165.9
$MTTF_{app}\ [Tu]$	232.8	220.8	194	178	166.1

Table 6.1 Table of the *MTTF* parameters related with reliability curves of the NCCS with one network

As we can see there are minimal differences among the values of the *MTTF* and $MTTF_{app}$ parameter. These results confirm the minimal errors given by proposed approximation method.

Decrease factor d_R	Step incr. $d_R = 10^{-3}$	Linear incr. $d_R = 10^{-1}$	Exponential incr. $d_R = 10^{-1}$
MAD	$3.89\ 10^{-3}$	$1.2\ 10^{-2}$	$1.17\ 10^{-2}$

Table 6.2 Maximal absolute difference between approximated results and results of the MCS

Due to minimal differences (almost invisible – Fig. 6.3) between reliability curves given by MCS and analytical approximation, in further presented examples only the *MAD* error will be presented in order to show obtained exactness of the reliability approximation.

6.2 Mini-drone application

6.2.1 Structure introduction

The embedded networked control structure of the mini-helicopter model is composed of thirteen sensors, four actuators and three controllers. One of the controllers executes the observer task. Thus, two controllers are directly determined for control the helicopter model. The helicopter model is composed of the following parts:

Sensors:

- Three magnetometers – measure the intensity of the local magnetic field.
- Three gyrometers – measure angular speed of the helicopter.
- Three accelerometers – acceleration measuring.
- Sensors for measuring the angular velocity of each rotor.

Controllers:

- Micro controller – ASIC-Silicon
- Micro controller – MPC555
- Microcontroller for an observer task

Actuators:

- Four motors.

Thirteen mentioned sensors are used in order to stabilize the helicopter altitude. The magnetometers measure the intensity of the local magnetic field which allows determine the helicopter's space orientation. There are three gyrometers in order to measure the angular speed of the helicopter and three accelerometers for acceleration of the helicopter's turning as well.

The orientation of the helicopter in the space is described by Euler angles [Cas 04], [Cas 05], [Gal 06] where generalized coordinates are

$$q = (x, y, z, \psi, \theta, \phi), \tag{6.4}$$

where x, y and z denote the position of the centre of mass of the rotorcraft and angles ψ, θ and ϕ represent Euler angles (yaw, pitch and roll angles) which denote the orientation of the rotorcraft.

The second possibility how to describe the orientation of the drone is the quaternion [Gue 05], [Tan 06]

$$q = [q_0, q_1, q_2, q_3]. \tag{6.5}$$

The quaternion represents the attitude of the model which describes helicopter's orientation in the space and it is computed by using the values from mentioned nine sensors. The main aim of the control is to obtain required system state q = [1, 0, 0, 0] which represents a stable position of the system.

The control can be represented in the cascade mode where the primary controlled value is helicopter's attitude and secondary, more dynamic, is angular velocity of the motors. Thus, the primary controller "*Master*" (MPC555) computes the angular velocity of the motors and the controller "Slave" compute the actuation value (tension) of the motors.

In order to be able to follow a flight mission other six ultrasonic sensors are presupposed. These sensors should be used to measure the distance from obstacles around the mini-helicopter. However, to stabilize the helicopter's attitude, it is sufficient to measure the values with mentioned gyro/magneto/accelerometers and four angular velocity sensors. Hence in further presented control structures as well as simulated structures, these thirteen basic sensors are considered.

There are other additional parts such as four propellers, four branches arranged in a cross and electronic placed in the middle of the cross (Fig. 6.4).

Mini-Helicopter's part / distances	**Weight / size**
Helicopter (excluding batteries)	320 g
Batteries	200 g
Maximum length	74 cm
Rotors diameter	29 cm
Height	11 cm
Distance between motor and center of gravity	20.5 cm

Table 6.3 Basic physical characteristic of the Mini-Helicopter laboratory model

Table 6.3 shows detailed basic physical characteristics of the mini-helicopter model (Fig. 6.4). The table describes weight and size of the helicopter's parts and the distances between them.

The mathematical model of the mentioned mini-helicopter proposed by Castillo and others in [Cas 04] is derived as follows. As was mentioned the Euler angles (yaw, pitch and roll angles) denote the orientation of the rotorcraft. These coordinates are divided into translational and rotational coordinates:

$$\xi = (x, y, z),\ \eta = (\psi, \theta, \phi). \tag{6.6}$$

The translational kinetic energy of the rotorcraft is:

$$T_{trans} \cong \frac{m}{2} \dot{\xi}^T \dot{\xi}, \tag{6.7}$$

where m denotes the mass of the rotorcraft. The rotational kinetic energy is:

$$T_{rot} \cong \frac{1}{2} \dot{\eta}^T J \dot{\eta}, \tag{6.8}$$

where matrix *J* represents the inertia matrix. And potential energy is given by:

$$U = mgz. \tag{6.9}$$

Thus the Lagrangian is:

$$\begin{aligned} L(q, \dot{q}) &= T_{trans} + T_{rot} - U \\ &= \frac{m}{2} \dot{\xi}^T J \dot{\xi} + \frac{1}{2} \dot{\eta}^T J \dot{\eta} - mgz \end{aligned}. \tag{6.10}$$

The model of the rotorcraft dynamics is obtained from the Euler-Lagrange equations with external generalized force:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = F, \tag{6.11}$$

where $F = (F\xi,\ \tau)$ and $F\xi$ is the translational force applied to rotorcraft due to the control inputs. Small body forces are ignored and thus:

$$\hat{F} = \begin{pmatrix} 0 \\ 0 \\ u \end{pmatrix}, \tag{6.12}$$

where control input $u = f_1 + f_2 + f_3 + f_4$ and $f_1 = k_i \omega_i$ where $k_i > 0$ is constant and ω_i is the angular speed of the i-th motor. Then

$$F_\xi = R\hat{F}, \tag{6.13}$$

where R is the transformation matrix representing the orientation of the helicopter (for cosine is used c and for sine is used s).

$$R = \begin{pmatrix} c_\psi c_\theta & s_\psi s_\theta & -s_\theta \\ c_\psi s_\theta s_\phi - s_\psi c_\phi & s_\psi s_\theta s_\phi + c_\psi s_\phi & c_\theta s_\phi \\ c_\psi s_\theta c_\phi + s_\psi s_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi & c_\theta c_\phi \end{pmatrix} \tag{6.14}$$

With defining generalized moments τ on the η variables

$$\tau \cong \begin{pmatrix} \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{pmatrix}, \tag{6.15}$$

the Coriolis-Centripetal vector

$$\overline{V}(\eta, \dot{\eta}) = \left(\dot{J} - \frac{1}{2} \frac{\partial}{\partial \eta} \left(\dot{\eta}^T J \right) \right) \dot{\eta} = -C(\eta, \dot{\eta}) \dot{\eta}, \tag{6.16}$$

and with other mathematical modifications was obtained this model:

$$m\ddot{\xi} = u \begin{pmatrix} -\sin\theta \\ \cos\theta \sin\phi \\ \cos\theta \cos\phi \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix} \tag{6.17}$$

$$J\ddot{\eta} = -C(\eta, \dot{\eta})\dot{\eta} + \tau \tag{6.18}$$

6.2.2 Structure evaluation

The structure of the helicopter's cascade control with one network is shown in figure 6.4. The structure composes of primary "*Master*" and secondary "*Slave*" controller and

group of the needed sensors for mini-helicopter attitude stabilization and four actuators (motors) as well. The communication among the components is realized through a CAN network.

Primary, in drone's control structure we consider thirteen sensors (three gyrometers, three accelerometers, three magnetometers and four angular velocity sensors). In addition, there are two controllers, four actuators and one communication network (Fig. 6.4).

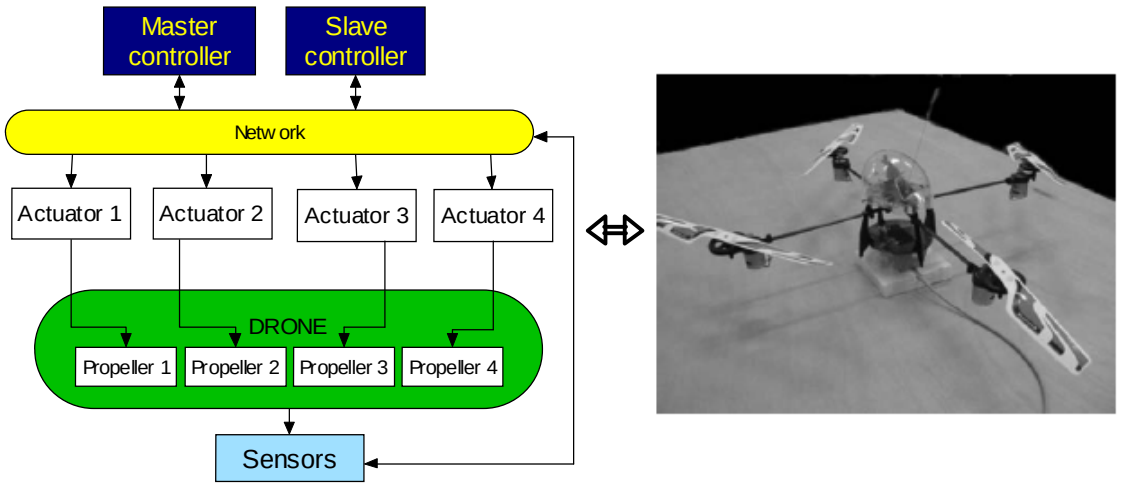


Fig. 6.4 Basic NCS structure of the mini-helicopter model

Considered common nominal failure rate is $\lambda_n = 10^{-3}$. In order to approximate the reliability of the drone's one network structure the mentioned table coefficients are used (tables 5.1 – 5.3). The same as in the previous examples the table with *MTTF* parameter is shown where simulated (*MTTF*) and approximated ($MTTF_{app}$) value are compared (Table 6.5). The values correspond with the structure where quasi-redundant controllers are considered as active redundant (no FR change) as well as with the structure without any redundant approach. As we can see, the differences are minimal what is confirmed by small maximal differences between simulated and approximated reliability values, shown in the table 6.6. The reliability curves of the considered control structure with different decrease factors are shown in figure 6.8.

Decrease factor d_R	Active red.	Step incr. $d_R = 10^{-3}$	Linear incr. $d_R = 10^{-1}$	Exponential incr. $d_R = 10^{-1}$	No red.
$MTTF$ $[Tu]$	54.9	54	53.6	52	49.6
$MTTF_{app}$ $[Tu]$	54.7	54.4	53.4	52.1	49.5

Table 6.4 *MTTF* parameters related with reliability curves of the drone's one network structure

The parameters of the decrease factors as well as nominal failures are identical with previous examples (see tables 5.1, 5.2). In order to evaluate the system under different conditions the parameters in further evaluated NCS example of the drone are changed.

Decrease factor d_R	Step incr. $d_R = 10^{-3}$	Linear incr. $d_R = 10^{-1}$	Exponential incr. $d_R = 10^{-1}$
MAD	$3.1\ 10^{-3}$	$1.08\ 10^{-2}$	$1.13\ 10^{-2}$

Table 6.5 Maximal absolute difference between approximated results and results of the MCS of the drone's one network control structure

Figure 6.7 shows an alternative networked control structure of the mini-helicopter model with two communication networks. This structure offers two groups of quasi-redundant components such as controllers ("*Master*","*Slave*") and networks (N_1, N_2). Thus, in comparison with the previous mentioned helicopter's control structure, this alternative allows benefit from controllers' reconfiguration and networks' reconfiguration as well.

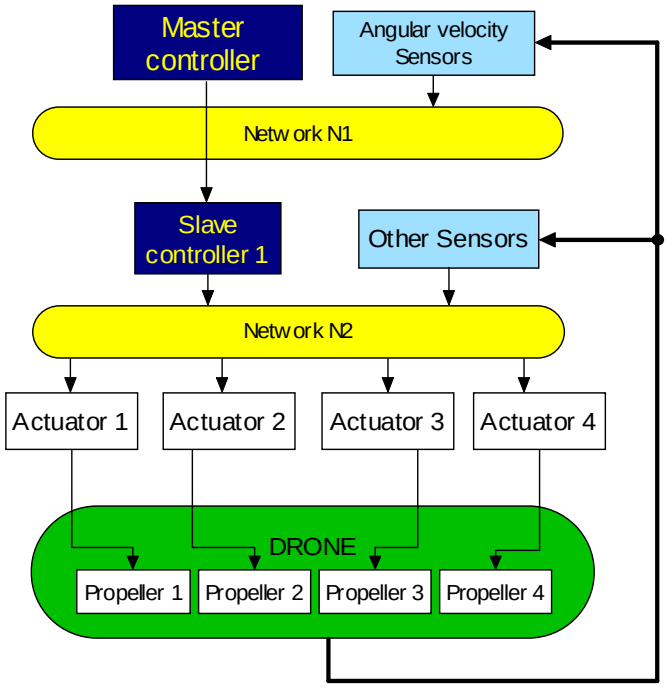


Fig. 6.5 Alternative structure of the mini-drone networked control system

The reliability curves of both drone control structures (Fig. 6.6 and 6.7) are shown in figure 6.8. Considered values of the decrease factors are shown in the legend of each subfigure (Fig. 6.8a, b). The corresponding simulated and approximated values of the

drone's two network structure are shown in table 6.8. The maximal evaluated errors for each approximation are shown in the table 6.9.

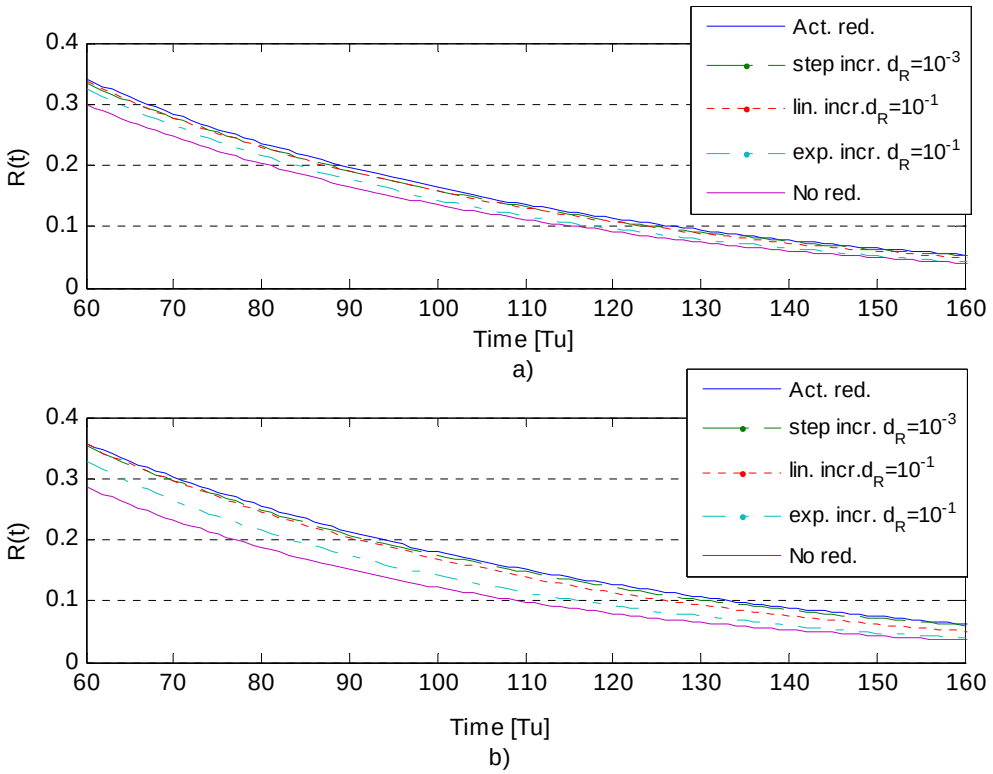


Fig. 6.6 Reliability curves of the drone control structure with a) one network, b) two networks

Decrease factor d_R	Active red.	Step incr. $d_R = 2.10^{-3}$	Linear incr. $d_R = 10^{-4}$	Exponential incr. $d_R = 10^{-4}$	No red.
$MTTF\ [Tu]$	57.5	56.7	55.5	52	47.1
$MTTF_{app}\ [Tu]$	57.6	56.9	54.9	52	47.1

Table 6.6 *MTTF* parameters related with reliability curves of the drone two network structure

Decrease factor d_R	Step incr. $d_R = 2.10^{-3}$	Linear incr. $d_R = 10^{-4}$	Exponential incr. $d_R = 10^{-4}$
MAD	$3.1\ 10^{-3}$	$1.08\ 10^{-2}$	$1.13\ 10^{-2}$

Table 6.7 Maximal absolute difference between approximated results and results of the MCS of the drone's two networks control structure

The reliability is approximated by using the following equation whereas approximed coeficients are used (tables 5.1 - 5.3). Thus general realiability equation is as follows

$$R_{Drone2N} = R_{S1-13}(t)R_{A1-4}(t)R_{2Cqr}(t)R_{2Nqr}(t), \qquad (6.19)$$

where $R_{S1\text{-}13}(t)$ is the reliability of the thirteen independent sensors, $R_{A1\text{-}4}(t)$ is the reliability of the four drone's actuators, $R_{2Cqr}(t)$ is the reliability of the two quasi

redundant controllers given by equation 5.1 with proposed table coefficients and $R_{2Nqr}(t)$ is the reliability of the two quasi-redundant networks (eqv. 5.1).

As was mentioned, the nominal failure rates of all the components are the same (in order to simplify the reliability computing through presented equations), hence the final reliability of the considered structure is:

$$R_{Drone2N} = e^{-17\lambda_n t}\left(1-\prod_{i=1}^{2}\left(1-e^{-(\lambda_{nC}+k_i d_R)t}\right)\right)\left(1-\prod_{i=1}^{2}\left(1-e^{-(\lambda_{nN}+k_i d_R)t}\right)\right), \quad (6.20)$$

where λ_{nC} is the nominal failure of the QR controllers and λ_{nN} is the nominal failure of the QR networks.

With adding the exact values of the failure rates and table coefficients to the equation 6.20, the final reliability of the system with exponential failure rate change with decrease factor $d_R = 10^{-1}$ is represented by following equation:

$$R_{NCCS} = e^{-17*10^{-3}t}\left(1-\left(1-e^{-(10^{-3}+0.009*10^{-3})t}\right)\left(1-e^{-(10^{-3}+0.086*10^{-3})t}\right)\right)^2. \quad (6.21)$$

In order to present the approximation results of the more complex NCS structures the drone's two network structure is evaluated with different parameters. The nominal failure rate of all components is $\lambda_n = 2.10^{-3}$ whereas the decrease factors for all three types of the FR change (single step, continuous linear and exponential) are as follows. The decrease factor for single step change of the controllers' and the networks' failure rate is 2.10^{-2}. In case of the continuous linear FR is $d_R = 10^{-4}$ and the same for exponential FR change the decrease factor is $d_R = 10^{-4}$. All values are shown in the table 6.8, similarly as in previous evaluated examples.

Decrease factor d_R	Active red.	Step incr. $d_R = 2.10^{-3}$	Linear incr. $d_R = 10^{-4}$	Exponential incr. $d_R = 10^{-4}$	No red.
$MTTF$ $[Tu]$	-	28.2	28.55	28.38	-
$MTTF_{app}$ $[Tu]$	28.5	28.2	28.54	28.4	23.3

Table 6.8 *MTTF* parameters related with reliability curves of the drone two network structure

Decrease factor d_R	Step incr. $d_R = 2.10^{-3}$	Linear incr. $d_R = 10^{-4}$	Exponential incr. $d_R = 10^{-4}$
MAD	$3.86\ 10^{-3}$	$4.06\ 10^{-3}$	$4.59\ 10^{-3}$

Table 6.9 Maximal absolute difference between approximated results and results of the MCS of the drone's two networks control structure

The errors of the approximation are shown in the table 6.9. The error in not higher than 0.5% of the maximal obtained reliability value. For completness the reliability

equation for all three cases of the FR change are presented. The reliability of the drone's two network control structure with the single step change of the controllers' and the networks' FR is as follows:

$$R_{sDrone2N} = e^{-17*2*10^{-3}t}\left(1-\left(1-e^{-\left(2*10^{-3}+0.44*10^{-3}\right)t}\right)\left(1-e^{-\left(2*10^{-3}+0.52*2*10^{-3}\right)t}\right)\right)^2, \quad (6.22)$$

where coefficients given by proposed method are $k_1 = 0.44$ and $k_2 = 0.52$ (Table 5.1) for the decrease factor $d_R = 2.10^{-3}$.

By using other coefficients and equation proposed in fifth chapter the reliability behavior of the considered system with the mentioned parameters for the continuous linear FR change with the decrease factor $d_R = 10^{-4}$ is as follows:

$$R_{lDrone2N} = e^{-17*2*10^{-3}t}\left(1-\left(1-e^{-\left(2*10^{-3}+0*10^{-3}\right)t}\right)\left(1-e^{-\left(2*10^{-3}+1*10^{-3}\right)t}\right)\right)^2, \quad (6.23)$$

where coefficients given by proposed method are $k_1 = 0$ and $k_2 = 1$ (Table 5.2) for the decrease factor $d_R = 10^{-4}$.

Third reliability curve where exponential FR change with the decrease factor $d_R = 10^{-4}$ is considered can be approximated by following equation:

$$R_{eDrone2N} = e^{-17*2*10^{-3}t}\left(1-\left(1-e^{-\left(2*10^{-3}+5*10^{-3}\right)t}\right)\left(1-e^{-\left(2*10^{-3}+5*10^{-3}\right)t}\right)\right)^2, \quad (6.24)$$

where coefficients given by proposed method are $k_1 = 5$ and $k_2 = 5$ (Table 5.3) for the decrease factor $d_R = 10^{-4}$.

6.3 Conclusion

The chapter presents using the proposed method for dependability analysis of the networked control systems where a shared redundant approach is implemented. Several networked control structures are simulated in order to verify the proposed reliability approximation method.

At first, the simple common known cascade control structure is chosen in order to demonstrate the rightness of the results of the reliability approximation based on the approximated coefficients presented in the tables 5.1 – 5.3 in the fifth chapter. As we can see, approximated results give the minimal difference in comparison to the results obtained by MCS. Of course, the time severity of the proposed analytical method is much lower in comparison with MCS.

In order to evaluate more complex control structure the mini-helicopter networked cascade control structure is chosen in two different variants. There is a drone with one network where twenty subsystems are considered whereas two quasi-redundant

controllers are implemented. The second proposed alternative is a drone networked control structure with two networks in order to cover not only the controllers' redundancy however benefit from the quasi-redundant networks too. The results of the reliability approximation as well as the life time parameter approximation (*MTTF*, $MTTF_{app}$) confirm the rightness and the effectiveness of the proposed method for dependability analysis.

The last presented drone control structure is evaluated with different parameters in order to demonstrate the method usability. The changed nominal failure rates of the components as well as different failure rate behavior defined by different decrease factors are considered. For the presented fast reliability approximation method the exactness of the obtained results can be considered as very sufficient. In case of higher exactness of the approximated results, the detailed PN model of the studied system has to be prepared.

7. Conclusion

Networked control systems are complex control structures with various types of dependencies among constituting subsystems. Several new phenomena such as random delays or problematic of asynchronism appear among the subsystems communicating through the network, which are not solved by using the classical control theory. Because of the complexity of NCS, different dependencies appear which have influence to the final system behavior; the ignorance of the inter-subsystem influences could lead to a non-presumed system behavior. The complexity of the NCS is also an advantage due to the existence of distributed processing which allow the decision algorithms to be implemented onto several processing units (microcontrollers, processors…): this property can be used in order to improve the dependability of the global system.

With the fast development of more and more sophisticated NCS, the methods of dependability evaluation have to cover these newer phenomena. The standards impose critical or safe systems to be evaluated in regards to the SIL (Safety Integrated Level). Thus, the question which should be posed just after asking for SIL level of the developed system is: “How did we determine the level of SIL ?”. The question is whether the method which has been used for determining the SIL level of each subsystem as well as the entire system is able to take into account all the indispensable influences within the studied system.

One of the main aims of this work is to describe and evaluate the influence of the implementation of the shared-redundant approach within NCS to the final dependability of the system. The work focuses on the implementation of the shared redundant or quasi-redundant subsystems within complex networked control structures. As was introduced in the third chapter, there are common redundant approaches (active, passive, hybrid redundancy) which are often used in order to increase the dependability of the systems. The disadvantage of these approaches is the requirement of additional components or subsystems which have to be added into the basic NCS structure. This leads to increasing the system complexity by the number of components as well as the cost increasing of the designed NCS. Hence, if the structure of the NCS allows implementing the shared redundant approach, it could be very helpful in order to improve the dependability parameter of the NCS.

The present work focuses on the description of the dependencies existing within the NCS with the SR approach. The two most used redundant combinations are studied: two and three components redundant subsystems. The use of the SR approach leads to the dynamic change of the failure rate. Depending on the different types of quasi-redundant subsystems (controllers, networks, …), as well as different states of the subsystems (simple increasing of the load, overloading, …), several different failure rate behavior are supposed. Three types of subsystems failure rate changes are considered such as a single step FR change, a continuous linear FR step change and an exponential FR change.

In order to identify the influence of the mentioned FR behavior to the final reliability of quasi-redundant subsystems, Petri Nets models are prepared whereas the reliability behavior is evaluated by using the Monte Carlo simulation method. In order to avoid preparing the model of the NCS with SR approach by using complex tool such as Petri nets, the effective reliability approximation method is proposed. The method is based on the simulation results of the mentioned most used redundant subsystems combinations with different FR behaviors given by MCS. By using the simple reliability equation and approximated coefficients presented in the fifth chapter, the reliability evaluation of the NCS where mentioned quasi-redundant subsystems are covered can be very effective and faster in comparison with reliability analysis by using PN in combination with MCS method.

Consequently, a simple technique based on the simple tables values, for a fast life time approximation, is proposed. Both methods, for reliability approximation as well as for time approximations are very effective with regard to the time needed for the reliability determination of the NCS where the groups of two or three SR components are considered. The proposed reliability approximation method can be considered as an extension of the classical methods for the reliability analysis such as fault trees or reliability block diagrams. This analytical extension of the classical approach for reliability analysis allows cover the specific type of the dynamic failure rate behavior such as linear and exponential FR change. It is necessary to mention that there are several limitations for using the proposed reliability approximation method. The main is that the method can be applied on the systems where failure rate of the quasi-redundant subsystems is identical or similar whereas differences could be ignored. However we can suppose that quasi-redundant subsystems have similar or identical hardware. Thus,

the condition of the same nominal failure rate could be accomplished within different NCS.

Together with proposition of the simple effective methods for reliability and life time approximation of the SR subsystems, different reconfiguration strategies are evaluated and proposed in order to additionally improve the dependability of the NCS with the implemented SR approach. Thus, the aim of the developed reconfiguration strategy for NCS is the partial elimination of the reliability reduction (as well as life time) of the system with SR approach which depends on the decrease factor associated to the studied quasi-redundant subsystems.

The correctness of the proposed method for the reliability approximation of the quasi-redundant subsystems is further verified in the final sixth chapter. Two more complex networked control structures are considered such as a networked cascade control system and a NCC structure of the mini-helicopter (drone) model. In case of the drone's structure, two alternatives are evaluated – with one group of the QR components (two controllers) and two groups (two controllers and two networks). The results confirm the possibilities of using the proposed methods for faster analytical reliability approximation of the more complex NCS with shared redundant subsystems.

A. Appendices

A.1 Several important distribution functions in reliability theory

Several distribution functions exist for a pertinent representation of the life length of simulated systems as well as for generating random component's failures. Depending on the type of simulated systems as well as their functionality, different distribution functions can be used. First introduced are *uniform* and *exponential* distributions which are the main distribution functions used in prepared and simulated models described in previous chapters. However, in order to show another type of system which could be considered in the reliability analysis of the NCS the Poisson distribution is also presented.

A.1.1 Uniform exponential distribution

The *uniform* distribution is a widespread used probability distribution in order to generate random number from the specified interval. This interval is defined by two parameters minimal value *a* and maximal value *b*.

The mean of this distribution is given by formula $(a+b)/2$ and the probability density function is

$$f(x)=\frac{1}{b-a}, \tag{A.1}$$

for $a \le x \le b$, in other cases has density function zero value.

The *continuous uniform* distribution function is

$$F(x)=\begin{cases} 0 & \text{for } x < a \\ \dfrac{x-a}{b-a} & \text{for } a \le x < b \\ 1 & \text{for } x \ge b \end{cases}. \tag{A.2}$$

A.1.2 Exponential distribution

Barlow and Proshan in [Bar 81] give theorems for exponential distribution as follows. Suppose that a component has the constant failure rate $r(t)$:

$$r(t)=\frac{f(t)}{R(t)}=\frac{\lambda e^{-\lambda t}}{e^{-\lambda t}}=\lambda, \tag{A.3}$$

then the *exponential* distribution is the only life distribution with constant failure rate function $r(t)$.

The *exponential* distribution $G_\lambda(t)$ [Bar 81] is given by

$$G_\lambda(t) = 1 - e^{-\lambda t} \text{ for } t \geq 0. \quad \text{(A.4)}$$

An important property of the exponential distribution is called "*memoryless*" property which means that *the remaining life of a used component is independent of its initial age*.

There is a relation between *uniform* and *exponential* distribution. In order to generate an exponential random variable it is sufficient to generate a uniform random variable and then to form it due to the followed theorem [Bar 81]:

Let U be a uniform random variable on [0, 1], then $Y = -\log U$ has an exponential distribution $1 - e^{-t}$.

Thus, the probability of failure of a component can be equally generated by a uniform random number generator as well as an exponential random number generator.

In the models where number of events during a specified time interval has to be generated a *Poisson* distribution is used. For example the modeling of the network loading where all connected nodes send the required information each sampling period T whereas some nodes could send packets more or less frequently. For more details the *Poisson* distribution is discussed in [Bar 81].

A.2 Description of the function used in the prepared Petri nets models in CPN Tools

In the described Petri nets models there are several often used functions in order to cover the work with files as well as correct functionality of the reliability models and theirs simulation of the random events by following suitable distribution function.

A.2.1 File functions

For reading and writing the data from files are following three functions used.

```
fun writeToFile(file_name, int_value) =
let
val file_id = TextIO.openAppend(Output.getModelDir()^"/"^file_name)
val _ =TextIO.output(file_id,IntInf.toString(time()))
val _ =TextIO.output(file_id,"\n")
in
TextIO.closeOut(file_id)
end;
```

```
fun getNumOfSim(fl_name) =
let
val file_id = TextIO.openIn(Output.getModelDir()^"/"^fl_name)
val n_of_sim = Option.valOf(Int.fromString(TextIO.inputLine file_id))
val _ = TextIO.closeIn(file_id)
in
(n_of_sim)
end;

fun setNumOfSim(file_name, n_of_sim) =
let
val file_id = TextIO.openOut(Output.getModelDir()^"/"^file_name)
val _ =TextIO.output(file_id,Int.toString(n_of_sim))
in
TextIO.closeOut(file_id)
end;
```

Function *writeToFile* writes the failure time of the modeled system. Function *getNumOfSim* reads number of simulations which have to be executed and function *setNumOfSim* writes new (decreased) number of simulation to the file.

The function *TextIO.openAppend*, *TextIO.openIn*, *TextIO.output*, *Option.valOf*, *TextIO.closeOut*, are internal function of the CPN Tools in order to be able to work with files.

A.2.2 Random distribution functions and type convert functions

CPN Tools offers several random distribution functions for modeling different random processes. In this study mainly the *uniform* and *exponential* random distribution functions was used.

The uniform function follows the conditions described by equations A.1 – A.3 and the syntax is as follows:

uniform(a:real, b:real) : real

Thus, uniform function has two real arguments which specify the interval for generating the real random numbers.

The second random distribution function is exponential function described by equation 7.4 where syntax in CPN Tools is as follows:

exponential(r:real) : real

where one real argument is required which represents the failure rate of component.

In order to be able to work with real types in the CPN Tools software several function for convert the types from and to integer and real type are necessary to use. Function for converting the type from integer to real has following syntax:

Real.fromInt(i:int) : real

For backward conversion following function should be used:

Real.toInt(r:real) : int

Conversion between real and string type should be provided by function

Real.toString(s:string) : real

In addition several mathematical round function are used such as floor (produces largest integer not larger than *r*), ceil (produces the smallest integer not less than *r*) and round (yields the integer nearest to *r*) where syntax is as follows:

floor (r:real) : int
ceil (r:real) : int
round (r:real) : int

Others functions covered by ML language are described in [Ull 98].

List of figures

Fig. 1.1 Distributed control system..... 6
Fig. 1.2 Continuous, discrete and networked control..... 12
Fig. 1.3 Time triggered NCS – TTT 13
Fig. 1.4 Event-driven NCS - TEE 13
Fig. 1.5 Terms covered by dependability..... 14
Fig. 1.6 Dependability tree proposed by Laprie in [Lap 92]..... 15
Fig. 1.7 Common shape of the curves R(t), Q(t) a r(t)..... 20
Fig. 1.8 RBD – equivalences to gates OR and AND..... 25
Fig. 1.9 SPN failure / repair model and its continuous time Markov model equivalent..... 28
Fig. 1.10 Structure of a PN model for reliability evaluation..... 29
Fig. 3.1 System with passive redundancy 37
Fig. 3.2 Operating principle of the a) passive and b) active redundancy 39
Fig. 3.3 Reliability curves with *MTTF* values of the two components system..... 40
Fig. 3.4 a) Cascade control; b) Feed-forward control 43
Fig. 3.5 NCCS with one network 45
Fig. 3.6 Possible scenarios for quasi-redundant controllers 45
Fig. 3.7 NCCS with two *networks* and alternative network connections 47
Fig. 3.8 Possible failure rate curves for subsystem S_2 during its mission 49
Fig. 3.9 Reliability model (using reliability bloc diagram) of the NCCS 52
Fig. 3.10 Root page "Main" of the hierarchical cascade NCS model with one network 54
Fig. 3.11 Example of the different places, transition and arcs in CPN Tools..... 56
Fig. 4.1 Graphical comparison of the results given by MCS and equation for passive redundant system composed of two components with failure rate $\lambda = 0.001$ 63
Fig. 4.2 Structure of the model composed of two quasi-redundant components CP1 and CP2..... 66
Fig. 4.3 Petri nets model of the component with time depending on the failure rate $\lambda(t)$..... 67

Fig. 4.4 Influence of the increased failure rate of the component by the constant decrease factor d_R to the reliability of the system composed of two quasi-redundant components 69

Fig. 4.5 Influence of the linearly increased failure rate by decrease factor d_R value 71

Fig. 4.6 Influence of the linearly increased failure rate by decrease factor d_R each 20^{th} and 600^{th} time unit to reliability of the two components redundant system 73

Fig. 4.7 Influence of the exponentially increased failure rate of the shared redundant component to the reliability behavior of the system 1-out-of-2 76

Fig. 4.8 Hierarchical PN model of the three quasi-redundant components 78

Fig. 4.9 Petri nets model of the component of TMR system with time dependent decrease factor $f_{dR}(t)$ 79

Fig. 4.10 Reliability behavior of the system composed of the three quasi-redundant subsystems with different values of the failure rate increasing 80

Fig. 4.11 Influence the constant decrease factor d_R to the reliability of the system composed of three quasi-redundant components 82

Fig. 4.12 Influence of the linearly increased failure rate by decrease factor d_R = 0.001 each 20^{th}, 200^{th} and 600^{th} time unit to reliability of the three components redundant system 84

Fig. 4.13 Influence the exponentially increased failure rate of the shared redundant component to the reliability behavior of the 1-out-of-3 system 86

Fig. 5.1 Dependency between decrease function and life time (*MTTF*) of the two QR components system with a nominal failure rate $\lambda_n = 10^{-3}$ 98

Fig. 5.2 NCCS with connections after network failure 101

Fig. 5.3 Modified model of the subsystem 104

Fig. 5.4 a) Comparison of two different scenario of the subsystem loading, b) step FR change with time ratio T_i/T_n = 9/1 c) step FR change - T_i/T_n = 5/5 d) gradual FR increasing T_i/T_n = 9/1 107

Fig. 5.5 a) Comparison of two different scenarios of the subsystem loading, b) step FR change with time ratio T_i/T_n = 2/8 c) gradual FR increasing T_i/T_n = 2/8 108

Fig. 5.6 Gradual increasing and decreasing scenario – following sine and cosine behavior 110

Fig. 5.7 The reconfiguration strategy diagram 113

Fig. 6.1 NCCS with one network 117

Fig. 6.2 Reliability curves of the NCCS with one network with different failure rate behaviors 118

Fig. 6.3 Approximated reliability curves of the NCCS with one network for a different FR change a) single step change dR = 10^{-3} b) linear FR change dR = 10^{-1} c) exp. FR change dR = 10^{-1} 119

Fig. 6.4 Basic NCS structure of the mini-helicopter model 124

Fig. 6.5 Alternative structure of the mini-drone networked control system 125

Fig. 6.6 Reliability curves of the drone control structure with a) one network, b) two networks 126

List of tables

Table 1.1 Safety integrity levels and corresponding PFD and FDFH 17
Table 1.2 Comparison of the tools for reliability modeling 32
Table 4.1 Comparison of the reliability values given by MCS and mathematical equation 63
Table 4.2 Comparison of the errors of the MCS with different number of the simulation samples 65
Table 4.3 Mean Time to First Failure of the studied system depending on the decrease factor d_R 70
Table 4.4 Examples of the Mean Time to First Failure parameter for linearly increased failure rate following d_R value 72
Table 4.5 Examples of the Mean Time to First Failure for increasing the failure rate by d_R value each 20^{th} time unit 75
Table 4.6 Examples of the Mean Time to First Failure for increasing the failure rate by d_R value each 600^{th} time unit 75
Table 4.7 Several values of the dependability parameter Means Time to First Failure for exponentially increased failure rate following the equation 4.11 77
Table 4.8 Mean Time to First Failure of the studied three components system depend on the decrease factor d_R 81
Table 4.9 Mean Time to First Failure of the studied system depending on the decrease factor d_R 82
Table 4.10 Values of the Means Time to First Failure parameter for increasing the failure rate by d_R value each 20^{th}, 200^{th} and 600^{th} time unit 84
Table 4.11 Several values of the dependability parameter *MTTF* for exponentially increased failure rate of the studied three component system following the equation 4.11 86
Table 5.1 Parameters of equation 5.1 for a single step FR change 91
Table 5.2 Parameters of equation 5.1 for continuous linear FR increasing 92
Table 5.3 Parameters of equation 5.1 for exponential FR increasing 93
Table 5.4 Parameters of equation 5.3 for a single step FR change 94
Table 5.5 Parameters of equation 5.3 for continuous linear FR increasing 94
Table 5.6 Parameters of equation 5.3 for exponential FR increasing 94

Table 5.7 Approximated values of the *MTTF* reduction of the two-component QR subsystem with different single step change of the nominal failure rate λ_n 96
Table 5.8 Approximated values of the *MTTF* reduction of the two-component QR subsystem with linear change of the nominal failure rate λ_n defined by the decrease factor d_R 97
Table 5.9 Approximated values of the *MTTF* reduction of the two-component QR subsystem with exponential change of the nominal failure rate λ_n defined by the decrease factor d_R 98
Table 5.10 Approximated values of the MTTF reduction of the two-component QR subsystem with an exponential change of the nominal failure rate λ_n following equation 5.4 99
Table 5.11 Approximated values of the *MTTF* reduction of the three-component QR subsystem with different single step change of the nominal failure rate λ_n 99
Table 5.12 Approximated values of the *MTTF* reduction of the three-component QR subsystem with a linear change of the nominal failure rate λ_n defined by decrease factor d_R 100
Table 5.13 Approximated values of the *MTTF* reduction of the three-component QR subsystem with an exponential change of the nominal failure rate λ_n defined by the decrease factor d_R 100
Table 5.14 System life time by following scenarios shown in figure 5.4 107
Table 5.15 System life time by following scenarios shown in figure 5.5 109
Table 5.16 System life time by following scenarios shown in figure 5.6 111
Table 6.1 Table of the *MTTF* parameters related with reliability curves of the NCCS with one network 119
Table 6.2 Maximal absolute difference between approximated results and results of the MCS 120
Table 6.3 Basic physical characteristic of the Mini-Helicopter laboratory model 121
Table 6.4 *MTTF* parameters related with reliability curves of the drone‘s one network structure 124
Table 6.5 Maximal absolute difference between approximated results and results of the MCS of the drone’s one network control structure 125
Table 6.6 *MTTF* parameters related with reliability curves of the drone two network structure 126

Table 6.7 Maximal absolute difference between approximated results and results of the MCS of the drone‘s two networks control structure 126
Table 6.8 *MTTF* parameters related with reliability curves of the drone two network structure 127
Table 6.9 Maximal absolute difference between approximated results and results of the MCS of the drone‘s two networks control structure 127

References

[Aub 03] Aubry, J.-F., "Sûreté de fonctionnement" (in French), Support de cours, INPL, Nancy, 2003

[Azg 05] Azgomi, A. M., Movaghar, A., "Hierarchical Stochastic Activity Networks: Formal Definitions and Behavior", International Journal of Simulation, Systems, Science and Technology 6(1-2), UK Simulation Society, p. 56-66, 2005

[Bar 81] Barlow,R., E., Proschan, F., "Statistical Theory of Reliability and Life Testing – Probability models", McArdle Press, Inc., Silver Spring, 1981

[Bar 03] Barger, P., "Evaluation et validation de la fiabilité des systemes d'automatisation à intelligence distribuée en phase dyamique" (in French), Doctoral dissertation, Université Henri Poincaré, Nancy, 2003

[Beb 07] Bebbington, M., Lai, C-D., Zitikis, R., "Reliability of Modules with Load Sharing Components", Journal of Aplied Mathematics and Decision Sciences, 2007

[Bit 06] Bitam, M., Alla, H., "Performance evaluation of communication networks for distributed systems", Int. J. Computer Applications in Technology, Vol. 25, No. 4, 2006

[Bob 03] Bobbio, A., Trivedi, K., "System Modeling with Petri Nets", Duke University, 2003

[Bob 04] Bobbio, A., Raiteri, D.,C., "Parametric Fault Trees with Dynamic Gates and Repairs Boxes", Reliability and Maintainability, 2004 Annual Symposium – RAMS, p. 459-465, ISBN: 0-7803-8215-3, 2004

[Bou 02] Bouissou, M., "Boolean logic driven Markov processes: A powerful new formalism for specifying and solving very large Markov models", Proceedings of the 6th International Conference on Probabilistic Safety Assessment and Management, San Juan, Puerto Rico, USA, Jún, 2002.

[Bou 04] Boudali, H., Dugan, B.J., "A Temporal Bayesian Network Reliability Framework", Proceedings of the 4th international conference on Mathematical Methods in Reliability, 2004

[Bou 05] Boudali, H., Dugan., J. B., "A new Bayesian network approach to solve dynamic fault trees", IEEE Reliability and Maintainability Symposium, pages: 451-456, 2005

[Bou 05a] Boudali, H., Dugan., J. B., "A discrete-time Bayesian network reliability modeling and analysis framework", Reliability Engineering and System Safety journal, vol. 87, No.3, pages: 337-349, 2005

[Bou 06] Boudali, H., Dugan., J. B., "A continuous-time Bayesian network reliability modeling and analysis framework", IEEE Transaction on Reliability, vol. 55, No.1, page 86-97, March 2006.

[Bro 02] Brosilow, C., Babu, J., Techniques of Model-Based Control, Prentice Hall, ISBN: 978-0-13-028078-7, 2002

[Buc 99a] Buchacker, K., "Combining fault trees and Petri-nets to model safety-critical Systems", High Performance Computing, 1999

[Buc 99b] Buchacker, K., "Analyzing Safety-Critical Systems Using Extendet Fault Trees", ARCS '99: Architektur von Rechensystemen, Jena, 1999

[Buc 00] Buchacker, K., "Modeling with extended fault trees", Proc. 5th Int. IEEE High-Assurance Systems Engineering Symposium, Albuquerque, pages: 238-246, 2000

[Bur 05b] Burge, J., Lane, T., "Learning Class-Discriminative Dynamic Bayesian Networks", Proceedings of the 22th International Conference on Machine Learning, Bonn, Germany, 2005

[Cam 97] Campelo, J.C., Yuste, P., Rodriguez, F., Gill, P.J., Serrano, J. J., "Dependability Evaluation of Fault Tolerant Distributed Industrial Systems", Valencia, 1997

[Che 98] Chen, J., Taylor, V. "Mesh Partitioning for Distributed Systems", The 7th International Symposium on High Performance Distributed Computing, p. 292-300, July 1998

[Chi 93] Chiola, G., Dutheillet, C., Franceschinis, G., Haddad, S., "Stochastic well-formed colored nets and symmetric modeling applications", Vol. 42, IEEE Transactions on Computers, p. 1343-1360, 1993

[Dav 97] David, R., Alla, H., "Du grafcet aux réseaux de Petri" (in French), Institut National Polytechnique de Grenoble, Editions HERMES, ISBN: 2-86601-325-5, Paris, 1997

[Dia 01] Diaz, M., "Les reseaux de Petri" (in French), HERMES Science Publications, ISBN 2-7462-0250-6, Paris, 2001

[Fre 95] Fredriksson, L.-B., "A CAN Kingdom – Rev 3.01", KVASER AB, Kinnahult, Sweden, 1995

[Fri 97] Fricks, R.M., Trivedi, K.S., "Modelling failure dependencies in reliability analysis using Stochastic Petri Nets", 11th European Simulation Multiconference, Istanbul, 1997

[Fri 98] Fricks, R.M., Pulia, A., Telek, M., Trivedi, K., "Applications of non-Markovian Stochastic Petri Nets", Performance Evaluation Review, Vol.26, No2, 1998

[Gal 07] Galdun, J., Ghostine, R., Thiriet, J. M., Liguš, J., Sarnovský J., "Definition and modelling of the communication architecture for the control of a helicopter-drone", 8th IFAC Symposium on Cost Oriented Automation, Cuba, February 2007

[Gal 08] Galdun, J., Liguš, J., Thiriet, J-M., Sarnovský, J., "Reliability increasing through networked cascade control structure – consideration of quasi-redundant subsystems", World IFAC Congress, Seoul, South Korea, July 2008

[Gho 08] Ghostine, R., "Influence des fautes transitoires sur la fiabilité d'un système contrôlé en réseau" (in French), Institut National Polytechnique de Lorraine, Doctoral dissertation, Nancy, 2008

[Gru 98] Gruhn, P., Cheddie, H. L., "Safety Shutdown System: Design, Analysis and Justification", ISA – The Instrumentation, Systems, and Automation Society, ISBN: 1-55617-665-1, 1998

[Gue 06] Guenab, F., Theilliol, D., Weber, P., Zhang, Y.,M., Sauter, D., "Fault-tolerant control system design: A reconfiguration strategy based on reliability analysis under dynamic behaviour constraints", p.: 1387-1392, 6th IFAC Symposium on Fault Detection, 2006

[Ham 05] Hamidi, K., "Contribution à un modèle d'évaluation quantitative des performances fiabilistes de fonctions électroniques et programmables dédiées à la sécurité", Thesis (in French), Institut National Polytechnique de Lorraine, Nancy, 2005

[Kri 05] Krings, A., W., "Fault-Tolerant Systems", CS449/549, 2005

[Lap 02] Laprie, J. C., "Dependability: Basic Concepts and Terminology", Springer Verlag Wien-New York, volume 5, 1992

[Lee 02] Leen, G., Heffernan, D., "Expanding Automotive Electronic Systems", Computer IEEE, Vol. 35, pages: 88-93, January 2002

[Lia 01a] Lian, F-L., Moyne, J. R., Tilbury, D. M., "Performance evaluation of control networks: Ethernet, ControlNet,and DeviceNet", IEEE Control Systems Magazine, Vol. 21, p. 66 – 83, February 2001

[Lia 01b] Lian, F-L., "Analysis, Design, Modeling and Control of Networked Control Systems", Doctoral dissertation, The University of Michigan, 2001

[Lin 06] Lin, J., Cheng, A. M. K. "Maximizing Guaranteed QoS in (m,k)-firm Real-time Systems", Proceedings of the 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications - RTSA'06, p. 402-410, Huston (USA), August 2006

[Lig 04a] Ligušová, J., Thiriet, J.-M., Liguš, J., Barger, P., "Effect of Element Initialization in Synchronous Networked control System to Control Quality", Reliability and Maintainability Annual Symposium – RAMS, p. 135-140, January 2004

[Lig 04b] Ligušová, J., "Metódy riadenia eliminujúce vplyv oneskorení spôsobených komunikačnou sieťou v sieťových riadiacich systémoch" (in Slovak), Doctoral dissertation, Košice, 2004

[Mar 02] Marshal, E., Scharpf, E., "Safety Integrity Level Selection – Systematic Methods Including Layer of Protection Analysis", ISA – The Instrumentation, Systems, and Automation Society, ISBN: 1-55617-777-1, 2002

[Mur 02] Murphy, K., "Dynamic Bayesian Networks: Representation, Inference and Learning", Doctoral thesis, Berkeley, University of California, Computer Science Division, 2002

[Nei 06] Neil, M., Tailor, M., Marquez, D., Fenton, N., Hearty, P., "Modeling Dependable Systems using Hybrid Bayesian Networks", The First International Conference on Availability, Reliability and Security, ARES, p. 5, ISBN: 0-7695-2567-9, April 2006

[Par 99] Paret, D., "Le Bus CAN Aplications CAN, CANopen, DeviceNet, OSEK, SDS..." (in French), ISBN: 2 10 0003659 9, Dunod, Paris, 1999

[Par 05] Paret, D., "Réseax Multiplexés pour Systèmes Embarqués – CAN, LIN, FlexRay, Safe-by-Wire…" (in French), ISBN: 2 10 005267 5, Dunod, Paris, 2005

[Pim 02] Pimentel, J.R, Salazar, M., "Dependability of Distributed Control System Fault Tolerant Units", The 28th Annual Conference of the IEEE Industrial Electronics Society, p. 3164- 3169, vol. 4, IECON '02, ISBN: 0-7803-7474-6, November 2002

[Poz 03] Pozsgai, P., Neher, W., Bertsche, B., "Models to Consider Load-Sharing in reliability Calculation and Simulation of Systems Consisting of Mechanical Components", IEEE – Proceedings annual reliability and maintainability symposium, p.: 493 – 499, 2003

[Rod 05] Rodrigues, G. N., Rosenblum, D. S., Uchitel, S., "Sensitivity Analysis for a Scenario Based Reliability Prediction Model", Workshop on Algorithms and Data Structures, 2005

[Ros 04] Roshandel, R., "Calculating Architectural Reliability via Modeling and Analysis", Proceedings of the 26th International Conference on Software engineering, 2004

[Sar 92] Sarnovský, J., "Riadenie zložitých systémov"(in Slovak), ISBN 80-05-00945-3, Alfa, Bratislava, 1992

[Shu 08] Shumsky, A., "Algebraic Approach to the Problem of Fault Accommodation in Nonlinear Systems", Proceedings of the 17th World Congress, p. 1884-1889, Seoul, Korea, July 2008

[Siv 01] Sivencrona, H., Hedberg, J., Röklinger, H., "Comparative Analysis of Dependability Properties of Communication Protocols in Distributed Control Systems", PALBUS Validation of Dependable Bus Systems, 2001

[Smi 04] Smith, D. J., Simpson, K. G. L., "Functional Safety – A straightforward guide to applying IEC 61508 and related standards", Oxford, ISBN 0 7506 6269 7, 2004

[Spi 87] Spiegel, M. R., "Theory and problems of statistics", McGraw-Hill Inc., New York, ISBN: 0-07-060227-1, 1972

[Son 03] Son, S. H., Seong, P. H., "Development of a safety critical software requirements verification method with combined CPN and PVS: a nuclear power plant protection system application", Reliability Engineering and System Safety, p.: 19-32, 2003

[Sta 98] Starý, I., "Spolehlivost systému" (in Chzech), Vydavatelstvý ČVUT, Praha, ISBN 80-01-01756-7, 1998

[Tan 07] Tanwani, A., Galdun, J., Thiriet, J-M., Lesecq, S., Gentil, S., "Experimental Networked Embedded Mini Drone - Part I. Consideration of Faults", European Control Conference 2007, Kos, Greece, p.: 4332-4337, ISBN: 978-960-89028-5-5, July 2007

[Ull 98] Ullman, J., D., "Elements of ML programming, ML97 Edition", Prentice Hall, ISBN 0-13-790387-1, New Jersey, 1998

[Vol 04a] Volovoi, V., "Modelling of the System Reliability with Stochastic Petri Nets", Annual Reliability and Maintainability Symposium, 2004

[Vol 04b] Volovoi, V., "Modeling of System Reliability Using Petri Nets with Aging Tokens", Reliability engineering and system safety, Vol. 84, pages: 149 – 161, 2004

[Wal 99] Walsh, G.C., Hong Ye, Bushnell, L.G., "Stability analysis of networked control systems", Proceedings American Control Conference, San Diego, CA, 2867-2880, June 1999.

[Woo 95] Wooldridge M., Jennings N. R., "Intelligent Agents Theory and Practice", Knowledge Engineering Review, 1995

[Wys 04] Wysocki, J., Debouk, R., Nouri, K., "Shared redundancy as a means of producing reliable mission critical systems", 2004 Annual Symposium – RAMS - Reliability and Maintainability, p.: 376-381, 2004

[Yih 05] Yih, P., Gerdes, J.C., "Modification of Vehicle Handling Characteristics via Steer-by-Wire", IEEE Transactions on Control Systems Technology, Vol. 13, p. 965-976, November 2005

[Zha 01] Zhang, W.B., Branicky, M.S., Philips, S.M., "Stability of networked control systems", IEEE Control System Magazine, pages: 84-99, 2001

[Zon 07] Zong, Q., Lin, W., Sun, L., « «Fault diagnosis of distributed networked systems , Procedings of the IEEE, International Conference on Automation and Logistics, p. 2334-2338, August 2007