Study of different load dependencies among shared redundant systems

Jan Galdun, Jean-Marc Thiriet, Jan Ligus

Study of different load dependencies among shared redundant systems

Ján Galdun*, **
* Laboratoire GIPSA-Lab (GIPSA-Lab UMR 5216 CNRS-INPG-UJF) BP 46, F-38402 Saint Martin d'Hères Cedex, France
Email: Jan.Galdun@tuke.sk

Jean-Marc Thiriet
Laboratoire GIPSA-Lab (GIPSA-Lab UMR 5216 CNRS-INPG-UJF) BP 46, F-38402 Saint Martin d'Hères Cedex, France
Email: Jean-Marc.Thiriet@ujf-grenoble.fr

Ján Liguš
** Department of Cybernetics and Artificial Intelligence, Technical University of Košice,
Letná 9, 04012 Košice, Slovakia
Email: Jan.Ligus@tuke.sk

***Abstract*—The paper presents features and implementation of a shared redundant approach to increase the reliability of networked control systems. Common approaches based on redundant components in control system use passive or active redundancy. We deal with quasi-redundant subsystems (shared redundancy) whereas basic features are introduced in the paper. This type of redundancy offers several important advantages such as minimizing the number of components as well as increasing the reliability. The example of a four-rotor mini-helicopter is presented in order to show reliability improving without using any additional redundant components. The main aim of this paper is to show the influence of the load increasing following different scenarios. The results could help to determine the applications where quasi-redundant subsystems are a good solution to remain in a significant reliability level even if critical failure appears.**

Keywords: Shared redundancy, Dependability, Networked control systems

I. Introduction

To be able to obtain relevant results of reliability evaluations for complex systems, it is necessary to describe the maximum of specific dependencies within the studied system and their influences on the system reliability. Different methods or approaches for control systems' reliability improvement are developed in order to be applied to specific subsystems or to deal with dependencies among subsystems. A classical technique consists in designing a fault-tolerant control [12] where the main aim is to propose a robust control algorithm. Guenab and others in [4] deal with this approach and reconfiguration strategy in complex systems, too.

On the other side is the design of reliable control architectures. Probably the most used technique is to consider the redundant components which enlarge the system structure and its complexity too. Active and passive redundancy is the simplest way how to improve dependability attributes of the systems such as reliability, maintainability, availability, etc [8]. However, as it was mentioned the control structure turns to be more complex due to an increasing number of components as well as the number of possible dependencies among components.

The paper introduces complex networked control architecture based on cascade control structure. The cascade structure was chosen purposely due to its advantages. This structure is widely used in industrial applications thanks to positive results for quality of control which are already described and generally known [2]. On the other side it offers some possibilities of system reliability improvement. There are potentially redundant components such as controllers (primary, secondary). If more than one network is implemented we could consider them as potentially redundant subsystems too. Finally if the physical system allows it, it is possible to take profit from sensors. The cascade structure and other features are introduced in more details in the third part.

The paper is organised as follows. After bringing closer the research background, the shared redundancy is introduced. The controllers and networks are presented in more details in order to show some dependencies which could be appeared when a shared redundancy approach is implemented. In the next part are presented networked topologies considered as cascade control (CC) structure of the 4-rotor mini-helicopter (drone) model [3]. Using Petri nets were prepared the models of the introduced quasi-redundant components as well as drone's control structure. A simple model of the two quasi-redundant subsystems is evaluated. Finally, are proposed the simulation results of the mentioned simple two components model as well as the model of the complex drone's structure with short conclusion.

II. Research Background

Control architecture design approach was taken into account by Wysocki, Debouk and Nouri [13]. They present shared redundancy as parts of systems (subsystems) which could replace another subsystem in case of its failure. This feature is conditioned with the same or similar function of the subsystem. Wysocki et al. introduce the shared

This work was not supported by any organization

redundant architecture in four different examples illustrated on "X-by-Wire" systems used in automotive applications. Presented results shown advantages of this approach in control architecture design.

The shared redundancy approach involves the problematic of a *Load Sharing* [1]. 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 [11]. Pozsgai and others analyze this type of systems and offer mathematical formalism for simple system 1-out-of-2 and 1-out-of-3. Also there are some mathematical studies [1] of several phenomena appeared on this field of research. Bebbington and others in [1] analyze several parameters of systems such as survival probability of load shared subsystems.

III. SHARED REDUNDANCY

Specific kind of redundant subsystems which have similar features such as active redundancy however gives us some additional advantages which will be 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 subsystem if it is urgently required. This type of redundancy is referred as *shared redundancy* [13] or *quasi-redundancy* [6]. 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.

As it was abovementioned, the *shared redundancy* (SR) mentioned by Wysocki and others in [13] 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 this function could be replaced by another part which follows the same or a 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 networked control systems, three kinds of quasi-redundant components (subsystems) could be considered:

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

Hence, a necessary but not sufficient condition is that a control structure where SR could be considered has to be composed 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 take profit 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 some combination of the physical values can not be measured by using one sensor due to inability to implement required functionality in one hardware component.*

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

There are several naturally suitable control structures which could implement the shared redundancy approach without other modifications such as cascade control structure (Fig. 1). This structure is often used in industrial applications thanks to its important features which improve the quality of control. With using cascade control structure there are several constraints [13]. The main condition requires that controlled system must contain subsystem (secondary subsystem FS(s) – Fig. 1) that directly affect to primary system FP(s). Thus, cascade structure composes of two independent controllers which could be used in order to implement the shared redundant approach.

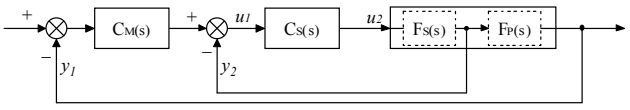

Fig. 1 Main structure of the cascade control

Usually for secondary subsystems there is a condition of faster dynamics than primary process. This condition must not be fulfilled [13] however, some modifications of conventional cascade structure (Fig. 1) and control laws must be provided.

A. Quasi-redundant controllers

In previous text, 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 shared redundant approach.

Suppose the networked cascade control system shown in figure 2. The system is composed of five main components (Sensor S_1, S_2, controllers C_1, C_2 and actuator A) and two networks. The communication flow among components is determined by its cascade control structure. Thus, sensor S_1 sends a measured value to 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.

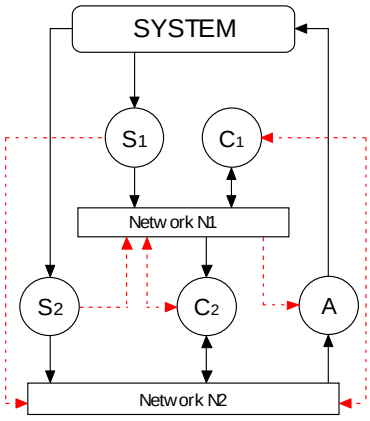

Fig. 2 NCCS with two networks and alternative network connections

Each part of the system (components and networks) presents independent subsystem. However, when quasi-redundant components are considered the system is not already composed of the 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 2 we could consider controllers C_1 and C_2 as the 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. However in case of failure of one of them, we could consider the second one as some kind of spare.

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 1).

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

The first one supposes that the controller is able to execute all the necessary tasks within the required sample periods (Fig. 3a). 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 failure of one of the components, the second takes care about its mission until its failure.

Figure 3b shows a second case when time to execute both necessary tasks is greater than the required sampling period. Thus, the controller will cause the delays which have significant influence to the system stability [5] [7]. Therefore, this delay could be known that allows its partially compensating by using several methods [10]. 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.

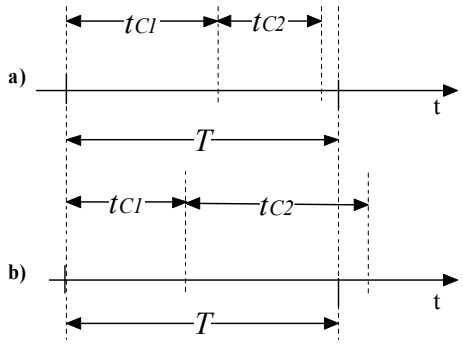

Fig. 3 Possible scenarios for quasi-redundant controllers

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.

B. Quasi-redundant networks

The second part of the NCS which could be taken into account as an SR subsystem are networks. Suppose a system with two networks (Fig. 2) 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 is 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 has failed the second one can take its load after a 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). This influence could be ignored when network performance parameters are sufficient. However, we can suppose that probability of network failure is increasing simultaneously with increasing network loading.

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.

C. Different scenarios in shared redundancy

When certain dependencies are ignored we could regard on the control system with QR components as control structure with 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. Thus, prevent from too pessimistic or too optimistic results of the reliability analysis of the considered systems. The dependencies could be distinguished as follows:

- active redundant dependency,
- single step change of the nominal failure rate $\lambda_n \in \langle 0;1 \rangle$ - 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. 4) 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 has failed and consequently the system is reconfigured and all nodes (components) start to communicate through the non-failed network which has 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 system failure. Mentioned examples with related equations are further discussed in more details.

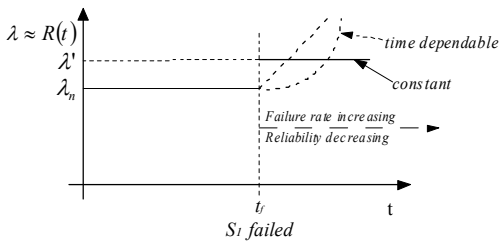

Fig. 4 Possible failure rate curves for subsystem S2 during its mission

Let's suppose that the reliability of the system $R(t)$, probability of the failure during time interval <0; t>, is characterized by a nominal failure rate $\lambda_n \in \langle 0;1 \rangle$. Let's suppose a system with two subsystems S_1 and S_2 (such as networks in abovementioned examples) whereas the subsystem S_1 will fail as first and then quasi-redundant subsystem S_2 will follow both missions (S_1 and S_2). In figure 4 are shown two above mentioned scenarios when the nominal failure rate λ_n of the subsystem is increased by a constant value or by value which could be described as linear or nonlinear function (functional dependencies).

At first increasing the failure rate λ_n one time by constant value (see Fig. 4) 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 simple formula

$$\lambda' = \lambda_n + d_R \qquad (1)$$

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 of its own mission and mission of the failed subsystem S_1.

The second case shown in figure 3 considers reliability reduction where the failure rate λ_n is increased during the working of the subsystem S_2 by a 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 following formula:

$$\lambda'(t) = \lambda_n + f_{dR}(t). \qquad (2)$$

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

$$\lambda'(t) = \lambda_n + d_R 10^{-3}(t + 1 - t_f) \qquad (3)$$

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

On the other side a nonlinear exponential function can be considered as follows:

$$\lambda'(t) = \lambda_n + e^{d_R(t-t_f)} \qquad (4)$$

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 on the increased failure rate.

D. Application to a mini-drone helicopter

The NCC structure is applied for the control of a four rotors mini-helicopter (Drone, Fig. 5). The proposed control structure for this real model is as follows. The NCC architecture is composed of one primary controller (Master) and one secondary controller (Slave), thirteen sensors, four actuators and two communication networks.

The Master is designed for attitude stabilization (control) through Slave controller for angular velocity control for each propeller. The aim of the control is to stabilize coordinates of the helicopter [11].

The controllers are used as quasi-redundant components within presented networked cascade control system (further only NCCS). They use the same control algorithm (propeller's angular velocity control) but with different input data (set point, system output, etc.)

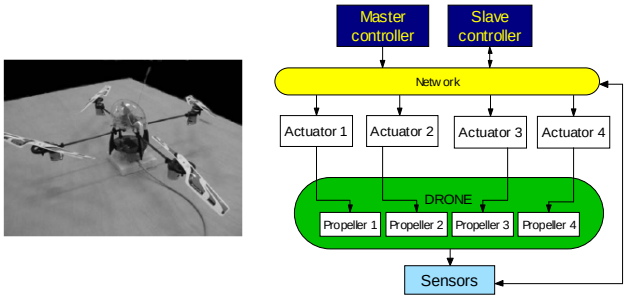

Fig. 5 Cascade control structure of mini-helicopter with one network

Hence, in case of failure one of them could retransmit all required data to another one, whereas pre-programmed control algorithm should compute the actuating value. Thus, failed controller is replaced by second one which start to compute actuating value.

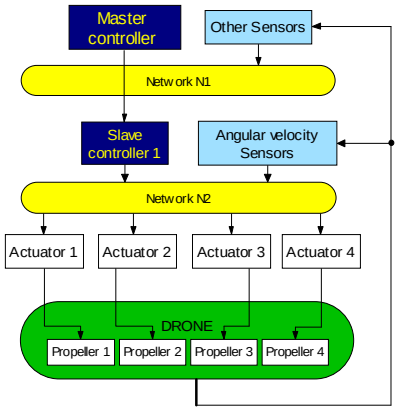

Fig. 6 Cascade control structure of mini-helicopter with two networks

Other quasi-redundant parts of this control structure are networks (Fig. 6). The same as in case of controllers, one of the networks can compensate another one after system reconfiguration. Usually, two networks are primary designed due to reduction amount of transmitted data. However, in case of network failure all data could be retransmitted through second one.

Described approach for subsystem's failure compensation by using the shared redundancy requires logical reconfiguration of the NCCS. Thus, in case of failure the hardware configuration is non-touched but communication ways must be changed in order to transmit the data to non-failed component or through non-failed network.

IV. Simulation And Results

All presented networked control architectures (Fig. 5, 6) were modelled by using Petri nets. This tool was chosen thanks to its ability to model different types of complex systems and dependencies within them. To provide the reliability analysis the Monte Carlo simulation (further only MCS) method was used. The multiple simulations of the modelled architecture [12] are provided to obtain the reliability behavior the basic two quasi-redundant components (for example two controllers in CCS structure).

Model of the system covers the simulation of the random events of the basic components of the system such as sensors, controllers and actuators as well as the network's random failures. Software used for model preparation is CPN Tools which allow multiple simulation of the model in order to obtain statistically representative sample of the necessary data to determine the reliability behaviour of the studied model.

As was mentioned, the simulation of the simple two quasi-redundant components with all considered changes of the failure rate (single, linear, nonlinear) was provided. Thus, new failure rate λ' of the non-failed component is computed by using equation (1), (3) and (4).

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 single step change of the failure rate by the specified decrease factor d_R to the reliability behaviour are shown in figure 7. We can see there are five curves. Two non-dashed curves show studied system as 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). These two curves determine borders where reliability of the studied system can be changed depending on value of the decrease factor d_R.

As we can see from figure 7, 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 λ' = 0.002 (d_R = 0.001) cause significant reduction of the reliability.

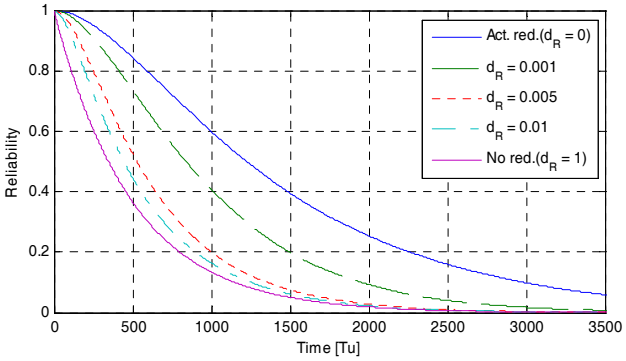

Fig. 7 Influence of the increased failure rate of the component by constant decrease factor d_R to reliability of the system composed of two quasi-redundant components

In tables I, are shown several values of life time (parameter MTTFF) for this studied system. Each table (Table I, II, III) shows the life time of the studied components as active redundant subsystems (d_R = 0) and as independent subsystems (d_R = 0.999). From value of the decrease factor d_R = 0.01 the life time of the system significantly improves (18% and more). The results of the linear and nonlinear failure rate increasing are shown in tables II and III. In all tables are noted the percentual value of the increased life time corresponding to the decrease factor.

Table IV shows the MTTFF parameters of both complex mini-helicopter structures. In the first drone structure (Fig. 5) two quasi-redundant controllers are considered. In the second structure (Fig. 6) two groups of quasi-redundant subsystems are considered and simulated – the controllers and the networks.

TABLE I.
MTTFF OF SIMULATED CONTROL STRUCTURES WITH DIFFERENT DECREASE FACTORS

Decrease factor – d_R	**MTTFF - Drone (Fig. 5)**	**MTTFF - Drone (Fig. 6)**
0	55 (+11%)	58 (+22%)
2.10^{-3}	54 (+9%)	56 (+17%)
10^{-2}	53 (+7%)	54 (+13%)
59.10^{-2}	50.5 (+2%)	49 (+3%)
0.999	49.6	47.6

In all simulated systems was observed the influence of the single step of the failure rate by a value specified by the decrease factor d_R. The same as in tables I – III, there are shown the life time of system corresponding to different decrease factors 2.10^{-3}, 10^{-2}, 59.10^{-3}. We can see that increasing the component's nominal failure rate λ_n by decrease factor equal to 59.10^{-3}, which represents approximately 59 times higher failure rate, has a significant influence to decreasing the life time of the system. The results are a little bit better than in the case of the system without redundant components (d_R = 0.999), but we could say that they are almost the same.

The drone's structure composes of twenty (twenty-one – structure with two networks) components – thirteen sensors (3 gyrometers, 3 magnetometers, 3 accelerometers, 4 rotors' angular velocity sensors), two controllers, four actuators and one (two) networks. Due to high ratio of the independent components and shared redundant components within drone's structure (18 independent and 2 quasi-redundant – Fig. 5) there is a difference between life times for minimal and maximal d_R is significantly smaller (about 11% and 22%) than in case of basic two components subsystem (Table I, II, III).

The Mean Time Before First system's Failure is significantly longer in case of basic two component subsystem than in drone's cases. As it was mentioned above this is caused by the difference in complexity between basic and drone's NCC architecture. In case of comparison between two drones structures (Fig. 5, 6) the results are better for architecture with two networks which is composed of two quasi-redundant subsystems – controllers (Master, Slave) and networks when the decrease factor is smaller than 59.10^{-3}. The increasing of the nominal failure rate by the decrease factor greater than 59.10^{-3} significantly decreases the life time of the drone. On the other side, even if the controller loading will change its failure rate approximatelly ten times ($d_R = 10^{-2}$) the system's life time is about 7% longer than in case of the system without shared redundant approach implementation.

V. CONCLUSION

The paper shows the influence of additional reliability decreasing of the quasi-redundant component to entire reliability of the studied system. Description of this dependency is getting closer to show the behavior of the system reliability when shared redundancy approach is implemented. The results shown in tables I – III could be very helpful in order to approximate the life time of the quasi-redundant subsystem under different conditions of the failure rate increasing. Presented cascade control architecture suitable for shared redundancy approach implementation could be applied to similar systems. For example, Steer-by-Wire control [9] of two front wheels in a car, etc. In addition the paper has shown the conventional cascade control structure within conditions of networked control systems as naturally suitable to profit from quasi-redundant subsystems as networks, controllers and potentially sensors if physical process allows it. Despite of some constraints for using this type of control, cascade architecture is widely used in industrial control applications. Hence, only the reconfiguration algorithm should be implemented to take profit from quasi-redundant subsystems.

The main advantages of the quasi-redundant components could be summarized as follows:

- The system is composed only of necessary

TABLE IV.
MTTFF OF THE TWO QUASI-REDUNDANT WITH SINGLE STEP CHANGE OF THE FAILURE RATE

$\lambda_n = 10^{-3}$	**Act. red. $d_R = 0$ ($\lambda' = 10^{-3}$)**	**$d_R = 0.001$ ($\lambda' = 0.002$)**	**$d_R = 0.005$ ($\lambda' = 0.006$)**	**$d_R = 0.01$ ($\lambda' = 0.011$)**	**$d_R = 0.1$ ($\lambda' = 0.101$)**	**No red. $d_R = 0.999$ ($\lambda' = 1$)**
MTTFF [*Tu*]	1503 (+300%)	1002 (+200%)	667 (+34%)	589 (+18%)	509 (+2%)	499

TABLE IV.
MTTFF OF THE TWO QUASI-REDUNDANT WITH LINEAR INCREASING OF THE FAILURE RATE

$\lambda_n = 10^{-3}$	**Active red. $d_R = 0$**	**$d_R = 10^{-3}$**	**$d_R = 10^{-2}$**	**$d_R = 10^{-1}$**	**No redundancy**
MTTFF [*Tu*]	1503 (+300%)	1153 (+231%)	812 (+63%)	611 (+22%)	499

TABLE IV.
MTTFF OF THE TWO QUASI-REDUNDANT WITH EXPONENTIAL INCREASING OF THE FAILURE RATE

$\lambda_n = 10^{-3}$	**Active red. $d_R = 0$**	**$d_R = 10^{-3}$**	**$d_R = 10^{-2}$**	**$d_R = 10^{-1}$**	**No redundancy**
MTTFF [Tu]	1503 (+300%)	902 (+80%)	676 (+35%)	537 (+8%)	499

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 significant.
- Prevention of the system's critical failure when QR subsystem has no sufficient hardware capacities.

VI. REFERENCES

[1] M. Bebbington, C-D. Lai, R. Zitikis, "Reliability of Modules with Load Sharing Components", *Journal of Aplied Mathematics and Decision Sciences*, 2007

[2] C. Brosilow, J. Babu, *Techniques of Model-Based Control*, Prentice Hall, 2002, ch. 10

[3] P. Castillo, A. Dzul, R. Lozano, "Real-Time Stabilisation and Tracking of a Four Rotor Mini-Rotorcraft", *IEEE Transaction on control systems technology*, Vol. 12, No. 4, 2004, pp. 510 – 516.

[4] F. Guenab, D. Theilliol, P. Weber, Y.,M. Zhang, D., "Sauter, Fault-tolerant control system design: A reconfiguration strategy based on reliability analysis under dynamic behaviour constraints", *6th IFAC Symposium on Fault Detection*, 2006, pp. : 1387-1392

[5] J. Galdun, R. Ghostine, J. M. Thiriet, J. Liguš, J. Sarnovský, "Definition and modelling of the communication architecture for the control of a helicopter-drone", *8th IFAC Symposium on Cost Oriented Automation*, 2007

[6] J. Galdun, J. Liguš, J-M. Thiriet, J. Sarnovský, "Reliability increasing through networked cascade control structure – consideration of quasi-redundant subsystems", *World IFAC Congress*, Seoul, South Korea, 2008

[7] J. Ligušová, J.M. Thiriet, J. Liguš, P. Barger, "Effect of Element's Initialization in Synchronous Network Control System to Control Quality", *RAMS/IEEE conference Annual Reliability and Maintainability Symposium*, 2004

[8] J. C. Laprie, , H. Kopetz, A. Avižienis, (1992). Dependability: Basic Concepts and Terminology, Chapter 1, Springer-Verlag / Wien, ISBN: 3-211-82296-8

[9] G. Leen, D. Heffernan, "Expanding Automotive Electronic Systems", *Computer IEEE*, Vol. 35, 2002, pp.: 88-93

[10] S.,I. Nicolescu,. *Stabilité systèmes à retard – Aspects qualitatifs sur la stabilité et la stabilisation*, Diderot multimedia, 1997

[11] P. Pozsgai, W. Neher, B. Bertsche, "Models to Consider Load-Sharing in reliability Calculation and Simulation of Systems Consisting of Mechanical Components", *IEEE – Proceedings annual reliability and maintainability symposium*, 2003, pp.: 493 – 499

[12] J. T. Spooner, K., M. Passino, "Fault-Tolerant Control for Automated Highway Systems", *IEEE Transactions on vehicular technology*, vol. 46, no. 3, 1997, pp. 770-785

[13] J. Wysocki, R. Debouk, K. Nouri, "Shared redundancy as a means of producing reliable mission critical systems", *2004 Annual Symposium – RAMS - Reliability and Maintainability*, 2004, pp.: 376-381