Recent Innovations in Mechatronics (RIiM) Vol. 2. (2015). No. 1-2.
DOI: 10.17667/riim.2015.1-2/19.

Fault-tolerant Mechatronic Systems Development: a Biologically-inspired Approach

Szász Csaba
Department of Electrical Machines and Drives
Technical University of Cluj
Csaba.Szasz@emd.utcluj.ro

Abstract— **Modern mechatronics embeds sophisticated control systems to meet increased performance and safety requirements. Timely fault detection is a critical requirement especially in safety-critical mechatronic applications, where a minor fault can evolve to catastrophic situations. In such cases it looks a high demand for more reliable, safety and fault-tolerant mechatronic systems development. The alternative to overcome all these bottlenecks was inspired from the biological world. By adapting the remarkable surviving and self-healing abilities of living entities it is possible to develop novel hardware systems suitable to fulfill in all the most demanding high reliability operation criteria's and requirements. The paper presents a biologically-inspired computing system based on a Field Programmable Gate Array (FPGA) network developed for high reliability mechatronic applications. By choosing a design strategy relying on a multi-cellular concept which outlines the versatility of biologically inspired technologies, task allocation or reliability problems can be solved with high efficiency. Real-time simulations prove that by implementing methods that imitate biological processes, high performance fault-tolerant and self-healing hardware architectures can be experimented and tested. The benefits of this approach are also confirmed by experiments performed on a laboratory-prototype hardware platform. The results underline that techniques which imitate bio-inspired strategies can offer viable solutions in high reliability mechatronic systems development.**

Keywords— fault-tolerant, mechatronic system, reliability, bio-inspired, FPGA network;

I. Introduction

By integrating mechanics with digital electronics and information processing complex mechatronic systems embeds a wide range of components: electromechanical actuators, sensors, electronic devices, mechanical components, hardware architectures, or software toolkits. As it is well known, modern mechatronics also relay on sophisticated control systems to meet increased performance and safety requirements. Timely fault detection abilities or system monitoring are critical requirements of many mechatronic systems as well [1, 2]. These features have been of utmost importance in safety-critical industrial applications such as nuclear power plants, space- and aircrafts, manufacturing processes, chemical plants, medical applications, energetic plants and grids, or military systems. Malfunctions in sensors, actuators, or other components can generate unsatisfactory operation, or even an accentuated instability of the entire system. Conscious that the consequences of a minor fault system can evolve to catastrophic situations, the demand on reliability, safety and fault-tolerance in such mechatronic systems is generally high. Therefore, it is necessary to design and implement control systems being capable of tolerate potential faults and improve the reliability while providing a desirable performance [3, 4].

However, it is no doubt that conventional feedback control design approaches in modern mechatronic systems may result in unsatisfactory performances in the event of malfunctions of system's components. Beside the above, the application of traditional fault detection and elimination methodologies in all these systems seems to be very inefficient and expensive, as well [5].To overcome such bottlenecks or weakness it is necessary to design high reliability mechatronic systems which are capable of tolerating potential faults, by keeping on adequate level the entire system parameters while providing a desirable global performance. In other words, the endeavor is to maintain the system stability properties by implementing control strategies with abilities to accommodate component failures automatically. This type of mecatronic systems are referred as "fault-tolerant control systems" (FTCSs) [1, 6, 7].

Over the past two decades, the increasing demand for more safety and reliable mechatronic systems has drawn more attention in the scientific community involved, the topic emerging as one of the most challenging research area for engineers specialized in mechatronics. The huge amount of effort has led to the development of a large scale of fault detection and diagnosis methods. There usually three main issues are followed: fault detection (the occurrence of a fault), fault isolation (the determination of the location and type of the fault), and fault identification (the determination of the fault magnitude) [1, 4, 5]. Of course there it is not enough room to enter in specific details. Therefore, without the claim to explore the entire topic it is mentioned only that the FTCSs can be classified into two main types: active or passive. Known in references as reliable systems (or control systems with integrity), the passive FTCSs embeds digital controllers designed to be robust against a class of presumed faults [1]. In contrast with the above, active FTCSs have embedded self-repairing and reconfigurable abilities; therefore they can react to the system component failures in active manner. In this way the stability and desired performance of the entire system are possible to be maintained. At the same time, in many cases active FTCSs can manage fault diagnosis (fault identification), fault detection, handles faults functionality, or take actions to compensate for the impacts of faults by selecting pre-computed control strategy under real-time processing constraints [8, 9, 10]. Howsoever, it is no doubt that the fault-tolerant approach implies a lot of challenging tasks in actual level high reliability mechatronic systems design, development and implementation issues.

Recent Innovations in Mechatronics (RIiM) Vol. 2. (2015). No. 1-2.
DOI: 10.17667/riim.2015.1-2/19.

Somewhat surprising, the alternative to overcome all these challenges was inspired by natural phenomena from biological word. It is well known, that during a long evolution process living entities has been enhanced with remarkable adaptation and surviving abilities, expressing powerful self-healing and surviving properties. By adapting and mimicking such mechanisms and capabilities from nature, it is possible to develop complex novel hardware systems suitable to fulfill in all the most demanding high reliability operation criteria's and requirements. Therefore, it is not surprising fact that a large amount of bio-inspired techniques are now frequently used to design and develop high reliability decentralized and distributed computing hardware systems for a wide range of safety-critical mechatronic applications.

II. Bio-inspired Fault-Tolerant Model Development

As it is well known, in the biological world of living entities the life starts when two gamete cells join it by forming a fertilized cell named the zygote (or mother cell). In multi-cellular organisms the zygote represents the earliest developmental stage of the embryo and constitutes the first stage in the growing process (via successively cell replications) of a unique biological organism. Such mother cell contain DNA (Deoxyribonucleic Acid) derived from both parents and provides all the genetic information necessary to form a new individual. With the main purpose to imitate the above mentioned biological process, let's consider a specially developed state-machine (or self-replicating automata) mimicking cell division properties has been designed and introduced as follows. It is given an arbitrarily large and shape two-dimensional (2D) finite space with borders marked by green elements, as shown in figure 1.

Similarly with the Langton's loops universal constructor operation, an "artificial cell" fulfilling the role of a zygote enters inside border of the 2D finite space. Then it self-replicates into daughter cells both to the right (x direction) and down (y direction) if the cell does not reach the green border. It is self-understood that a full self-replication process towards the two cardinal directions constitutes one "iteration". Of course, theoretically this state-machine is capable to grow in an infinite space as well, by executing an infinite number of replication iterations. However, biological organisms are finite structures with well established shapes encoded within their biological cells gene set. Therefore, the size and shape of the 2D space from figure 1 will be also encoded into the artificial cell's gene set.

As it is well known, cell differentiation defines the role and functions of each cell of an organism. Each cell has a copy of all the genes that constitute the genome of the species and depending on the cell's position inside the organism each one is defined by its particular pattern of regulated gene expression. In this model, a gene is expressed by a specific program task, and the set of genes forms the artificial DNA. In order to avoid a complex presentation, let's assume the simple case in which the artificial DNA is composed of a set of only 5 genes (a double helix made of 5 pieces) labeled here *A, B, C, D,* and *E,* as shown in figure 3.

In order to handle the artificial cell differentiation problem, the state-machine can set the dividing cells according to the rule expressed in figure 3. Within it, in addition to healthy cells spare cells are also considered.

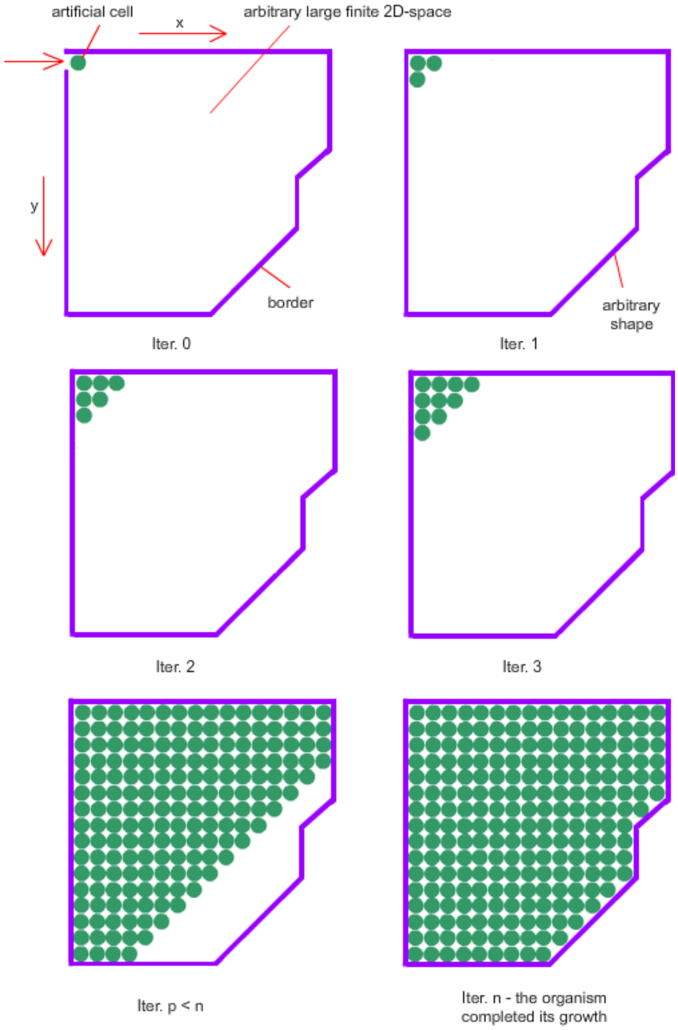
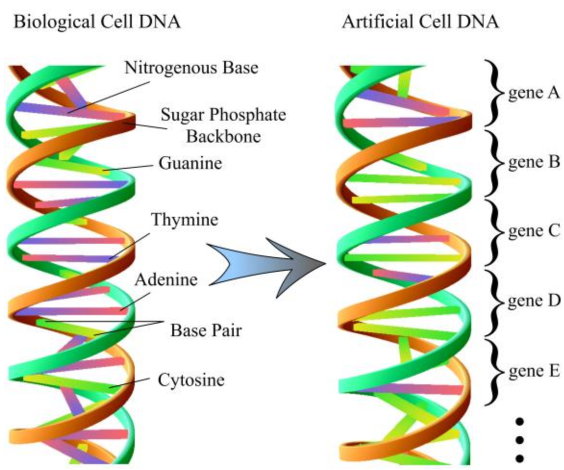


Fig. 1: The 2D state-machine model and its evolution after an arbitrary



DOI: 10.17667/riim.2015.1-2/19.

number of self-replicating iterations.

Fig. 2: The artificial DNA implementation strategy.

This is because a biological organism can easily accommodate imperfections due to the large quantity of redundancy inside of them, but artificial structures require some inherent fault tolerance. If many faults are found within the system, there will also be an adequate amount of spare cells which can handle the physical errors and damage which have occurred. Of course, with only 5 genes considered in this example, artificial organisms with 5 different gene expression cells will be born in the network. These, surrounded by 4 additional spare cells, constitute a group of 9 artificial cells which expresses an organism configuration ready to tolerate multiple faults.

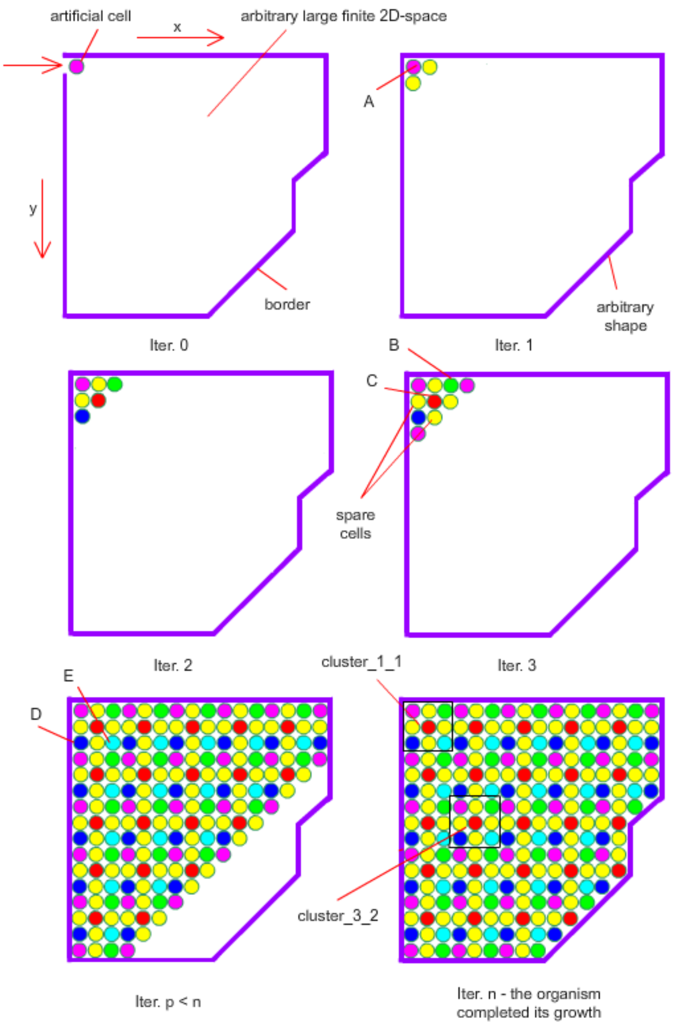


Fig. 3: The cell differentiation process

In the *3x3* dimension array of the 9 cells, they are differentiated according to the following rule (figure 3) [11]:

if coordinates (1_1) then express gene A *light (red);*
if coordinates (1_3) then express gene B *(green);*
if coordinates (1_2, 2_1, 2_3, 3_2) then cell is spare *(yellow);*
if coordinates (2_2) then express *gene* C *(red);*
if coordinates (3_1) then express *gene* D *(dark blue);*
if coordinates (3_3) then express *gene* E *(light blue);*

The model does not at all limit the number of considered genes rather for a higher number of artificial genes the differentiation rule could become more complicated.

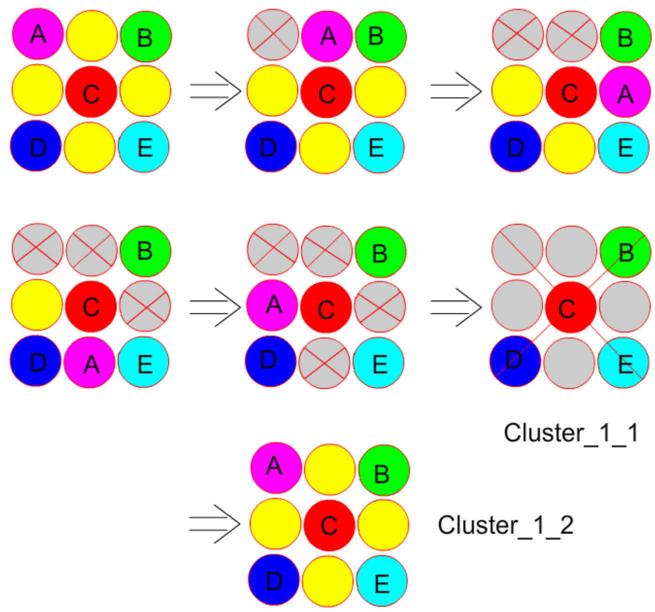


Fig. 4: The fault-tolerant abilities implementation

The entire fault-tolerance mechanism of the artificial organism is explained through the simple algorithm expressed in figure 4. There it is considered an arbitrary cell cluster (for example Cluster_1_1) with 5 active cells expressing the genes *A*, *B*, *C*, *D*, *E*, and 4 spare cells. Let's consider in this cluster the fault of the active cell *A*. The neighbors are able to detect that the communication with cell *A* it fails and thus the faulty cell is instantaneously isolated in the network. At the same time a spare cell immediately becomes active expressing the same gene *A* as the faulted ones. If another fault of the cell with gene *A* occurs, the above cell-replacement process it will be repeated. In the case of multiple faults, this replacement sequence can be repeated until there are free spare cells in the cluster. Moreover, if a cluster dies another instantaneously takes over its functionality (in figure 4 the cluster labeled *1_2*). The most important aspect of the entire self-repairing (or self-healing) process is that in any moment the organism is able to express the same gene set, even in case of multiple cell-faults. This means that the whole multi-cellular system is highly fault-tolerant, tolerating a huge amount of cell faults or damages [12].

Recent Innovations in Mechatronics (RIiM) Vol. 2. (2015). No. 1-2.
DOI: 10.17667/riim.2015.1-2/19.

III. The Bio-Inspired Fault-Tolerant Model Hardware Implementation

The bio-inspired fault-tolerant model introduced in the previous paragraph has been experimented by designing and developing a specially conceived laboratory setup. Being conscious of that at current technological levels the Field Programmable Gate Arrays (FPGAs) represents the perfect solution solution to implement highly concurrent interaction maps like biological cell arrays, the reconfigurable technology has been used for development. In fact, this type of processor is a Very Large Scale Integrated (VLSI) chip built inherently with massively parallel structures with a powerful hierarchy of customer reconfigurable interconnections network. Therefore, FPGAs can be configured by the customer arbitrarily wiring together a huge number of different hardware configurations. The main advantage of FPGA circuits is that they are ideally suited to implement highly parallel architectures and therefore are very suitable to build digital computing systems that emulates logical model of biological networks.

To validate the model by experimental development *Spartan-3E Starter Kit* boards have been used for implementation. The main hardware resources of the *Spartan-3E Starter Kit* built upon the *XC3S500E FPGA* are as follows: *4 Mbit Platform Flash Memory*, *16 Mbit SPI Flash*, *16 MByte NOR Flash*, *64 MByte DDR SDRAM Memory*, powerful *VGA* display port and incorporated *LCD* monitoring, up to 110 user-programmed I/O digital lines, 4 analog output channels, and 2 analog input channels [13]. These powerful resources meet our expectations excellently regarding the artificial cell's processing power. Thus, the high amount of *4 Mbit Platform Flash Memory*, *16 Mbit SPI Flash*, *16 MByte NOR Flash*, and *64 MByte DDR SDRAM Memory* will be used both for program tasks execution and data managing.

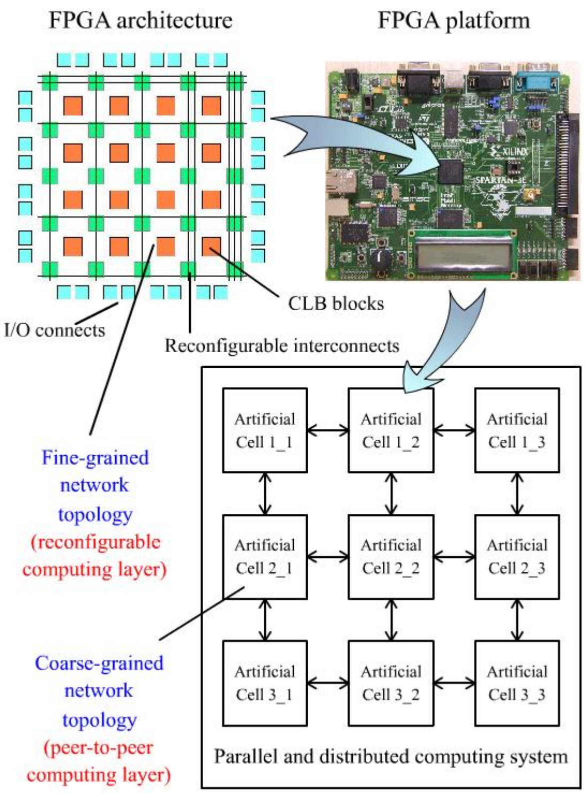


Fig. 5: The bio-inspired fault-tolerant system implementation strategy

The *LCD* display is a very appropriate device for the tasks execution and cells own state (e.g. dead or alive) visualization. The inter-cellular communication signals and data flow between cells can be conveniently monitored by interconnecting with the *VGA* port. Furthermore, the *110 I/O* digital lines have been shared for communication busses on the four lattices of an artificial cell. At least, the 2 analog input lines are suitable for performing the cell's sensing of its environment, and the 4 analog outputs to deliver signals [13].

During the laboratory setup development has been followed the implementation strategy expressed in figure 5. There a tissue-topology coarse-grained network of FPGA-based development boards it is considered, assembled into a peer-to-peer computing layer. Each computing unit of this layer (named generically "artificial cell") covers a powerful FPGA-based processing system. 9 of such identical modules (with the same hardware configuration) constitute a computing cluster with the main function to execute a complex processing task. On this layer there is no a supervisor unit or central infrastructure to provide a service or manage the network resources, all responsibilities being uniformly divided among all processing units. This layer can be extended then arbitrarily in the *X* and *Y* directions, in accordance with the designer's options and needs. However, this peer-to-peer computing layer imitates an inter-cellular layer of a biological organism, where flows inter-cellular communication processes and cell-multiplication, embryonic growth, faults accommodation, or self-healing processes can be modeled and tracked. The intra-cellular layer of the proposed model will be implemented on a powerful fine-grained network topology, using the reconfigurable technology embedded into the FPGA processors.

The previously presented hardware system has been interconnected to a servomotor actuator-based mechatronic system (figure 6). There the servomotor is a two-phase hybrid-stepping motor driven by a current-source asynchronous PWM inverter. The inverter embeds an SGS Thomson L6506 current controller IC, respectively one L298 and one L6210 chips for the power electronic module implementation.

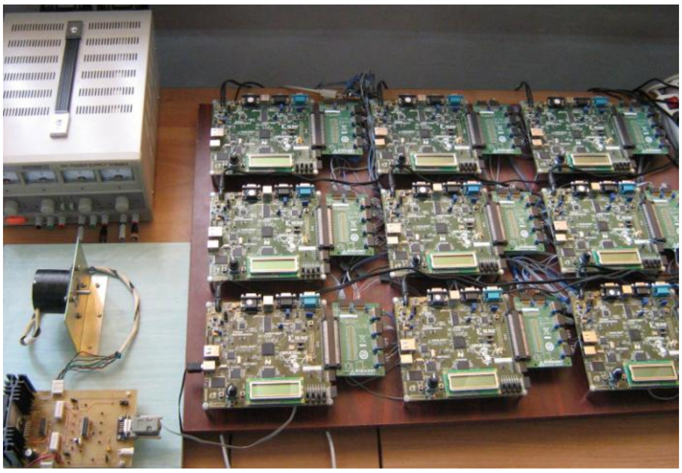


Fig. 6. Laboratory test bench for bio-inspired fault-tolerant mechatronic system development [14]

Supposing that it is required a high reliability fault-tolerant mechatronic system implementation the main control algorithm has been divided it into 5 smaller tasks, as shown next in figure 7. These tasks have been distributed then between the available computing units of a cluster being processed in parallel by these. For example, *Cell1_1* execute only the *task A*, which means the generation of the control signals to the electrical drive system inputs. In the same way, *Cell1_3* can execute the *task B* (the motor angular position acquisition), *Cell2_2 task C* (the motor phase currents acquisition), *Cell3_1 task D* (processing the motor control strategy), and *Cell3_3* the *task E* (PWM inverter control) [14].

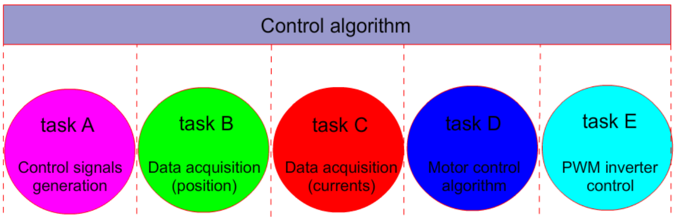


Fig. 7: Task allocation map of the computing system [12]

It is not difficult to deduce that parallel and distributed execution of the motor control task will provide a much higher processing speed and the real-time implementation will be significantly facilitated. At the same time, the entire mechatronic system is highly fault tolerant. Moreover, the introduced hardware system is well suitable to control several similar applications as well, or even a few different types of other industrial processes.

IV. Conclusion

This paper highlights that biologically-inspired techniques that imitate living entities remarkable surviving and self-healing abilities can offer viable solutions for fault-tolerant mechatronic systems implementation. A short insight has been introduced about how to develop a bio-inspired fault-tolerant model capable of embedding biological organism's properties in silicon technology. By adapting a multi-cellular development strategy system reliability problems can be solved with high efficiency. Experimental results prove that the adopted fault-tolerant model is well suitable for high reliability mechatronic systems design and development.

References

[1] Y. Zhang, j. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems", *Annual Reviews in Control*, Vol. 32, issue 2, pp. 229-252, doi: 10.1016/j.arcontrol.2008.03.008, 2008.

[2] S. Dash, V. Venkatasubramanian, "Challenges in the industrial applications of fault diagnostic systems", *Computers and Chemical Engineering*, 24 (2), pp. 785–791, 2000.

[3] D. Dochain, W. Marquardt, S.C. Won, O. Malik, M. Kinnaert, "Monitoring and control of process and power systems: Towards new paradigms: Status report prepared by the IFAC coordinating committee on process and power systems", *Annual Reviews in Control*, 30 (1), pp. 69–79, 2006.

[4] P.M. Frank, S.X. Ding, T. Marcu, "Model-based fault diagnosis in technical processes", *Transactions of the Institute of Measurement and Control*, 22 (1), pp. 57–101, 2000.

[5] R. Isermann, "Model-based fault-detection and diagnosis—Status and applications", *Annual Reviews in Control*, 29 (1), pp. 71–85, 2005.

[6] R. Isermann, R. Schwarz, S. Stolzl, "Fault-tolerant drive-by-wire systems", *IEEE Control Systems Magazine*, 22 (5), pp. 64–81, 2002.

[7] V. Venkatasubramanian, R. Rengaswamy, K. Yin, S.N. Kavuri, "A review of process fault detection and diagnosis. Part I. Quantitative model-based methods", *Computers and Chemical Engineering*, 27 (3), pp. 293–311, 2003.

[8] M. Zhong, H. Fang, H. Ye, "Fault diagnosis of networked control systems", *Annual Reviews in Control*, 31 (1), pp. 55–68, 2007.

[9] R. Isermann, "Fault-diagnosis systems: An introduction from fault detection to fault tolerance", *Springer*, Berlin, Germany, 2006.

[10] M. Witczak, "Modelling and estimation strategies for fault diagnosis of non-linear systems: From analytical to soft computing approaches", Lecture notes in control and information sciences, Vol. 354, *Springer*, Berlin, Germany, 2007.

[11] Cs. Szász, V. Chindriş (2009) – *Self-healing and Fault-tolerance Abilities Development in Embryonic systems implemented with FPGA-based hardware*, IEEE 13th International Conference on Intelligent Engineering Systems, April 16-18, 2009 Barbados, IEEE Catalog Number: CFP09IES-CDR, ISBN: 978-1-4244-4113-6, Library of the Congress: 2009901330, pp. 215-220.

[12] V. Chindriş, Cs.Szász Cs (2011) – *Bio-inspired Parallel Computing Structures for High Reliability Servomotor Control Applications*, Proceedings of the 10th International Symposium on Parallel and Distributed Computing (ISPDC'2011), Cluj, pp. 270-273, 2011.

[13] Xilinx Inc., *Spartan-3E Families User's Guide* (www.xilinx.com).

[14] Cs. Szász Cs., A. Cioloca (2013), "Two-layer Coarse-fine-grid Network Model for Bio-inspired Computing Systems Development", *17th International Conference on System Theory, Control and Computing – ICSTCC 2013*, October 11-13, Sinaia, ISBN: 978-1-4799-2228-4/13$31.00.