# Communication integrity for slow-dynamic critical embedded systems

Amira Zammali, Agnan de Bonneval, Yves Crouzet

# Communication integrity for slow-dynamic critical embedded systems

Amira Zammali [(1),(2)]
[(1)] CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
[(2)] Univ of Toulouse, UPS, LAAS, F-31400, Toulouse, France
Email: zammali@laas.fr

Agnan de BONNEVAL [(1),(2)]
[(1)] CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
[(2)] Univ of Toulouse, UPS, LAAS, F-31400, Toulouse, France
Email: agnan@laas.fr

Yves CROUZET [(1),(2)]
[(1)] CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
[(2)] Univ of Toulouse, LAAS, F-31400, Toulouse, France
Email: crouzet@laas.fr

***Abstract*—We present, in this paper, challenges and works in progress for a new communication integrity approach that is based on error detection codes and targets slow-dynamic critical embedded systems. The novelty of this approach lies in the fact that it takes profit of the fault tolerance criterion of slow-dynamic systems. Thus, it does not focus on each exchanged message but rather on a set of messages (which number is being be set according to the safety requirement of the targeted system). This approach relies on a set of control functions whose error detection capabilities and coverage are complementary, which improves the resulting detection capability compared to the usual use of one unique control function.**

***Keywords—slow-dynamic systems, critical embedded systems, fault tolerance, safety, communication integrity, error detection codes.***

## I. Introduction And Problematic

Nowadays, critical embedded systems are based on complex networks including active intermediate nodes. This increases the occurrence of erroneous messages and introduces new types of errors, even though the occurrence of undetected erroneous messages can lead to catastrophic events (e.g. airplane crash). Thus, ensuring the communication integrity in such systems is crucial. Traditionally, integrity policies aim at avoiding the occurrence of one undetected erroneous message. So they use heavy error detection codes in order to obtain an efficient detection power per each exchanged message. Yet, previous works [1] in our research team revealed that, for some kinds of systems, to meet the safety requirement, there is no need to focus very strongly on the integrity of each message. In fact, avoiding the occurrence of a number X (X>1) of undetected erroneous messages among N messages is sufficient for these systems. These previous works have defined a cumulative error detection policy consisting of a set of complementary control functions. This policy was based solely on CRCs codes and targeted Flight Control Systems. These works open horizons to us in order to dig deeper and propose a more complete and generic approach adopting the complementary property of used functions. Section II describes the targeted systems: slow-dynamic critical embedded systems. Section III presents the context communication integrity approach to be adopted in these targeted systems and section IV is devoted to present our works in progress.

## II. Slow Dynamic Critical Embedded Systems

The class of systems we target in our works is the class of slow-dynamic critical embedded systems. The critical property induces high safety requirements. It means the system is low fault tolerant because of some kinds of failures may lead to catastrophic events (loss of goods and even lives): typically, failure rate must be less than $10^{-9}$ failure/hour. "Embedded" means that such systems do not dispose of a huge of resources (memories, processors, etc.) and communications are based on short messages (e.g. 100 bits for Flight Control Systems).

The novelty, here, is the "slow-dynamic" property of the system (first defined in [1]). In fact systems can be classified into two classes: i) fast-dynamic systems; ii) slow-dynamic systems. "Fast-dynamic systems" are defined by a duration of their significant changes very close to the duration of the refresh cycle of their changes command computation. This enables to send one unique message (command) during the duration of significant change. Thus, an undetected erroneous message may lead to a catastrophic event. While the so-called "slow-dynamic systems" are defined by a duration of significant changes is much larger than the refresh cycle duration. This enables to send several messages (commands) during this duration (see Fig.1). Thus, a catastrophic event cannot result from one undetected erroneous message, but only from a set of undetected erroneous messages whose number exceeds a threshold being set according to the case study.

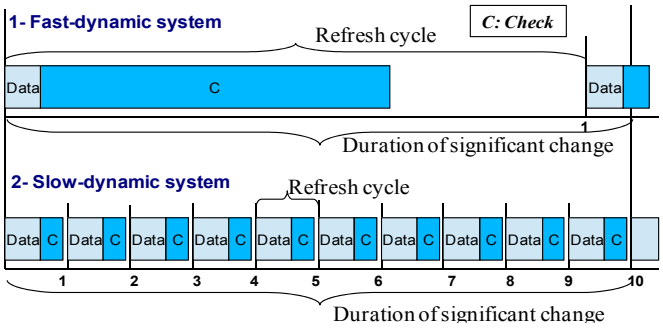


Fig. 1. Slow-dynamic systems compared to fast-dynamic systems

An example of slow-dynamic system is the flight control system on commercial airplanes. Computers exchange control-command messages with the actuators governing the flight

control surfaces. These surfaces are designed to move slowly (about 50°/s). The refresh cycle duration (about 10 ms) is much smaller than the duration of a significant change. Thus, many control-command messages are exchanged and the system can tolerate several undetected erroneous messages.

This slow-dynamic property makes possible to deal with the problem of communication integrity in a different way.

## III. Communication Integrity Approach

As described before, for slow-dynamic critical embedded systems, the integrity policy does not focus on each exchanged message. Our goal is rather to avoid the occurrence of more than X undetected erroneous messages among N transmitted ones. So instead of using one unique control function, we rely on a set of complementary ones, which means they have complementary (therefore cumulative) detection capabilities and coverage. So, this policy is more efficient, as described in Fig. 2, where we assume: all messages are erroneous, X=3, N=10, three complementary functions F1, F2 and F3 (and D: Detected, ND: NonDetected).

In our approach, we target the application layer and we aim at ensuring the end- to-end integrity. We consider the following assumptions:1) the key safety requirement of considered systems is the tolerance of less than $10^{-9}$ failures per hour [2] [3]; 2) the X undetected erroneous messages among N can be considered either as consecutive, nor as not consecutive; 3) intermediate nodes are active (with memories and treatment capabilities); 4) communication channels are binary and symmetric; 5) messages size is around 100 bits; 6) refresh cycle is around some ms; 7) targeted errors are random independent errors, burst errors and particularly repetitive errors; 8) the redundancy must be as low as possible in terms of check bits, networks components and channels.

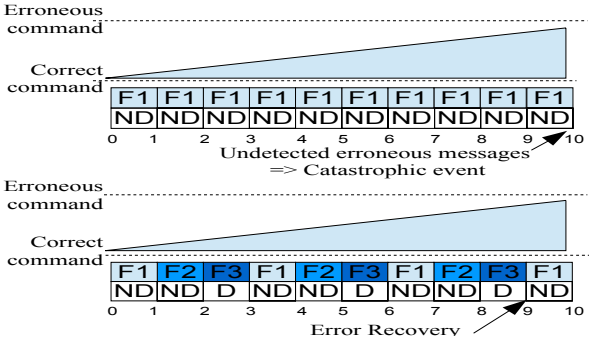


Fig. 2. Error detection policy : one function versus multi functions policy

## IV. Challenges And Works In Progress

Several challenges are arising with the specificities of our error detection policy. The first one is to find theoretical proofs and/or validate by experimentations the complementarity property of nonCRC codes. Previous works [1] have proved that two different CRC generator polynomials are complementary only if they share a minimum of common factors. Now, our goal is to find other complementary codes. An other side of this challenge is to use complementary codes belonging to different families in the same detection policy. The second challenge is to limit the resources consumption in terms of calculation time and memory since we consider embedded systems. The third challenge is to decrease the redundancy in terms of check bits in order to respect the short messages property we consider. The forth challenge is to ensure the synchronization between the network nodes (source, sink and intermediate nodes) in order to be sure that they use all, at every refresh cycle, the same control function. The last challenge is to extend the application domain of the detection policy and explore other slow-dynamic critical embedded systems and not only be limited to flight control systems.

To take up these challenges, our approach is based on an optimal error detection codes selection. In fact, for CRC codes, it was proven [4] that conventional polynomials are not necessarily the best choice to make. Moreover, we are exploring lightweight codes like Adler and Fast CRCs that have efficient capabilities with a lower complexity than CRCs. Besides, we are working on automotive systems in order to study the possibility to apply our policy to it. To validate our theoretical solution, we will rely on simulations via the Matlab-Simulink platform which provides tools to model and simulate communications. Experimentations will consist of Monte Carlo simulations. We have started modelling our experimentations. We are working on three models of errors injection: i) exhaustive injection (considering all possible erroneous messages); ii) selective injection (considering a kind or a subset of erroneous messages) and iii) random injection. To accelerate simulations, we are exploring the "Parallel computing" tool, a Matlab tool that we are working on it. It permits to make a set of parallel simulations while providing the synchronization between inputs, outputs and parameters.

## V. Conclusion

In this paper, we have first presented how the dynamic of a system can impact the way of considering communication integrity in critical embedded systems. For the class of slow-dynamic systems, we have reminded an innovative solution based on complementary error detection functions. In this context, one of the most important challenge we deal with, is to find (by theory or simulation) other error detection codes (than CRCs codes previously used), that would consume less time and memory, while having the property of complementary detection capabilities and coverage. And we seek to extend this approach to other domains than only aeronautic.

## References


[1] A. Youssef, Y. Crouzet, A. de Bonneval, J. Arlat, J. J. Aubert and P. Brot, "Communication integrity in networks for critical control systems". The European Dependable Computing Conference (EDCC), Coimbra, Portugal, 18-20 Oct. 2006, pp.23-34

[2] M. Paulitsch, J. Morris, B. Hall, K. Driscoll, E. Latronico and Ph. Koopman, "Coverage and the use of cyclic redundancy codes in ultra-dependable systems". The international Conference on Dependable Systems and Networks (DSN), Yokohama, Japan, 28 June-1 July 2005, pp.346-355

[3] Federal Aviation Administration, "System Safety Handbook, chapter 3:Principles of System Safety", 30 december 2000, 19 p.

[4] Ph. Koopman and T. Chakravarty, "Cyclic redundancy code (CRC) polynomial selection for embedded networks". The international Conference on Dependable Systems and Networks (DSN), Florence, Italy, 28 June-1 July 2004, pp.145-154.