

Modularisation of Safety & Control for X-By-Wire Multiapplication-Platforms

M. Armbruster, E. Zimmer, M. Lehmann, R. Reichel and J. Klaubert

Universität Stuttgart, Institute of Aircraft Systems

Copyright © 2008 SAE International

ABSTRACT

This paper shows a realisation approach for functional safety considerations for X-by-Wire vehicles. The E/E architecture has been built up as a redundant highly-safety-critical multiapplication platform with access to all sensors and actuators (so called satellites) ensuring different safety integrity levels besides reliability. Such it will be possible to develop high-level control functions (HCFs) independently of the overall vehicle E/E architecture, redundancy and even data-flow. This means that herein an approach is shown of how to split the responsibility of safety and the development of intelligent vehicle control-functionality. The feasibility has been shown within the EC-funded project SPARC (www.sparc-eu.net). The paper focuses on the identification of functions being implemented in software to ensure safety. The detailed design of the mechanisms themselves will not be part of this paper.

INTRODUCTION

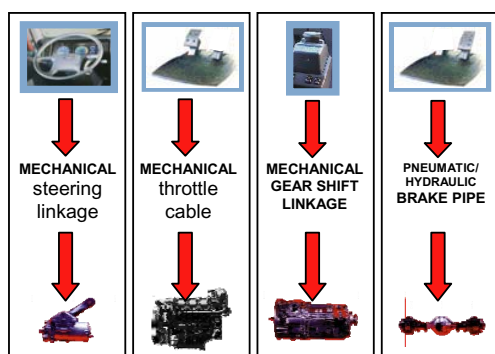


Figure 1 – traditionally vertically set-up systems [19]

Driven by the aim of halving the number of road accident victims in the EU by 2010 [17], research activities go towards increasing safety by improving the quality and reliability of information used by advanced driver assistance systems (ADAS) and allowing the implementation of advanced safety applications [16].

This directly leads to very close coupling (Figure 2) of all traditionally vertically set-up systems (Figure 1).

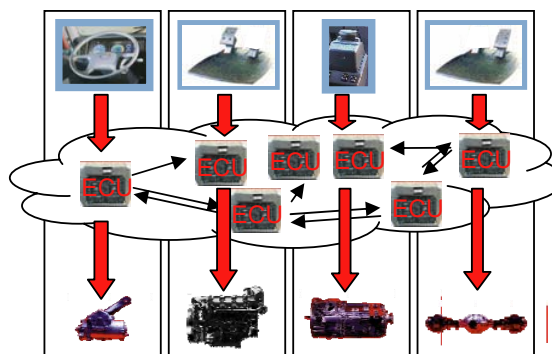


Figure 2 - close coupling of vertically separated control-functions based on X-by-Wire technology [19]

The idea shown here is to set up a completely new concept of vehicle control based on X-by-Wire technology (feasibility has been shown within the EC-funded project SPARC). The vehicle's functionality is no more built up out of separated systems but of functional-layers being geared to the functional active chain of vehicle control as shown in Figure 3: input, processing and output.

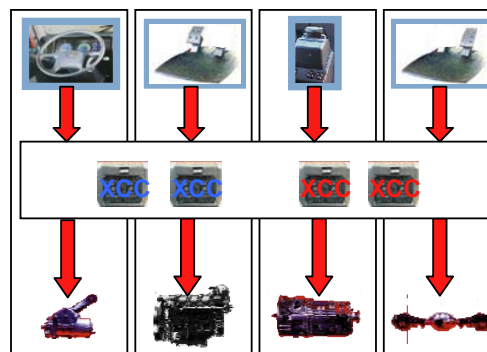


Figure 3 - horizontal separation of vehicle control-functionality: input, processing and output [19]

The realisation thereby is based on the idea to split up the responsibility of safety and vehicle control. High-level control functionality (HCF) as advanced electronic Stability Program with steering intervention (aESP), Antilock Braking System (ABS), emergency brake-assistant or co-pilot functionality shall be developed independently of any vehicle architecture, redundancy or even data-flow. The vehicle (will be called system from now on) gets abstracted to a safely consolidated data-base – from a functional point of view (see figure 4).

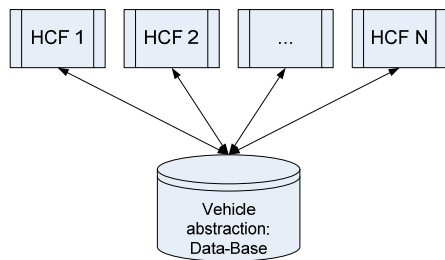


Figure 4 - vehicle abstraction to data-base

The problem of this modularisation of safety and control comes from a HW resp. SW-design point-of view: how to characterize safety and how to transfer quantitative requirements to functional-ones within the system, HW and SW- codesign.

QUANTIFICATION OF SAFETY

As shown in [2] the quantification of safety can be done based on a failure-condition classification. Each class (no safety-effect, minor, major, hazardous, catastrophic) gets assigned an allowed maximum average probability for the occurrence of the corresponding failure-effect (not given, $1E-03$, $1E-05$, $1E-07$, $1E-09$). Anyhow it must be taken into account, that the quantitative meaning of “catastrophic” must be determined as an equivalent level of safety compared to state of the art technology. As summarized in [12] “catastrophic” failure conditions are “those which would prevent continued safe flight and landing and include loss of function, malfunction, and unintended function”. Catastrophic failure conditions thereby must be extremely improbable. That is, they must be “so unlikely that they are not anticipated to occur during the entire operational life of all airplanes of one type” [8], paragraph 9.e(3). Transferred to X-by-Wire vehicles the max. tolerable numerical probabilities for safety-critical vehicle control-functions as steering or braking must be determined.

A possible determination has been shown in [1] which also leads to a tolerable failure-rate/hour for catastrophic events in the order of magnitude $1E-09$. As this calculation is based on national data, an alternative probabilistic estimation for catastrophic hazards over the whole fleet of a vehicle is given in table 1. The results shown in here compared to the results shown in [1] can be summarized to the following conclusion:

- The max. expected tolerable failure-rate for catastrophic events for the realisation-approach of the aspired data-base abstraction of a vehicle is in the order of magnitude $1E-09$.
- A worldwide expected tolerable max. failure-rate is expected to be less than $1E-11$.
- The realisation of X-by-Wire approach shown here must be able to handle
 - highest degrees of integrity and
 - highest degrees of reliability with respect to scalability (relevant variations are expected from OEM to OEM and depending on the vehicle being considered).

number vehicles sold worldwide	2E06
number of hours per vehicle and year	200
lifetime per vehicle in years	10
systems, considered with hazardous failure-conditions (steering, braking)	2
number of catastrophic conditions per system	10
magnitude of order for resulting tolerable failure-rate for catastrophic events / h	$< 1E-11$

Table 1 – rough estimation of the acceptable failure-rate for catastrophic events

CHARACTERIZING THE ADVERSARIES OF SAFETY

As adversaries of safety of an electronic control-system either errors or faults can be identified. Whereas errors are basically treated within a development-process oriented approach, faults are handled by a functional approach during runtime by corresponding detection and treatment-methods within the redundancy-management.

Those functions basically handle the detection of failures, the passivation of faults and finally the treatment which reallocates a valid functional item-cutset to execute the HCFs. Therefore the system must be split up to separated regions to which failures can be assigned (granulates). Granulates (comparable to fault-containment regions - [10]. The misbehaviour resp. failure can either be caused by an error or a fault – based on the given definition of error and fault) can be defined as a single or a set of items with the following characteristics:

- it is possible to assign faults resp. errors to a granulate,
- it is possible to partition the overall system to a set of granulates,
- granulates can be passivated which means they can be removed out of the functional chain of the overall system so that they can not harm any other granulate and thus the execution of the HCFs.

Fault-treatment can be set up in different forms. I.e. dynamic reconfiguration leads to a full masking of faulty granulates in case that non-faulty ones can take over the

corresponding functionality. If this was not possible, functional degradation helps to reduce functionality to a minimum required with the help of remaining non-faulty granulates. In some special cases a retry-strategy can help to reintegrate formerly incorrect granulates (especially CPU-based units) to reestablish availability.

The characteristics of failure-effects can be classified to scope, duration and domain. The scope is split up to global or local failures¹, means failures which can not be seen consistently over the whole system. Those failures are also known as byzantine failures as shown in [18]. The duration of failures can be permanent, transient or intermittent and finally the domain can be separated to the time, value-domain and space-domain. This means, that data is not received at all resp. in time, invalid data is sent/ generated or – very important to prevent fault/ error-propagation within among partitions – SW-services do not terminate in time or they try to access invalid memory-areas. Each failure-effect characterisation requires its corresponding detection mechanisms.

A single point-failure describes a failure which directly leads to the event “Loss of system” without any backup. Such a failure shall not happen independently of any probability and must be avoided by design (development-oriented approach).

The appearance of a failure can occur just once, several times, stay forever or toggle around. For the detection and isolation of failures with respect to the duration, it is important to react just with the corresponding isolation: a transient failure shall not be isolated permanently. This does also cover the transient event “loss of data”, which may occur due to distributed power-interrupts or even during the start-up-process of the system. The differentiation between time and value-domain is important because both domains require totally different detection mechanisms.

SYSTEM

System requirements basically can be split into functional requirements, safety requirements and architectural/technological requirements. Functionality thereby does not only include high-level control functionality. In fact it does include basic operational control-flow handling including power-up/down-handling, setting up communication, power-distribution or task/operational-state handling of high-level control-functions resp. satellites as sensors or actuators. This responsibility will be summarized herein as the control-flow management sw-domain. High-level control-functionality represents all vehicle control-functionalities as for instance aESP, ABS,... and will not be considered within this paper, as the redundancy-management functionality is independent of HCFs and can be configured flexibly. Safety will be characterized herein by integrity and reliability. The quantification of safety has

¹ dormant failures are not treated herein, because they are not treated by online-monitoring mechanisms.

been shown in the section “Quantification of Safety”. Thereby it is known that developing a multiapplication platform for X-by-Wire application requires highest integrity. Within the centralized platform, approach integrity will be ensured by both self-checking pair hardware/ duplex-computers and by the corresponding monitoring mechanisms realized by software.

Reliability shall be ensured via multiplicity and simplicity. Satellites (sensors and actuators around the central platform-computer) shall be set up as simple as possible to ensure a very low failure-rate. The self-checking pair-computers (XCCs) instead are multiplied whereas the multiplicity itself can be derived as a function of mission reliability. Such it will even be possible to set up the platform-core as a fit-and-forget system. This means, the core can be set up such that within a given period of operational service-time less than a given percentage of all vehicles may require an unplanned maintenance.

Besides the operational service-time between planned maintenance-actions, lets first do an safety-assessment with respect to a mission-time of one hour. Assuming a given failure-rate per Lane λ_{Lane} the probability for the event “Loss of all XCCs” $P\{E\}$ as a function of the given failure-rate per Lane λ_{Lane} over the number of used XCCs can be calculated. For a very conservative assessment λ_{Lane} has been set to 1E-04 and the allowed Probability $P\{E\}$ shall be less then 1E-11 for a mission-time of one hour. With that it can be seen, that the central platform-computer shall be equipped at least with three XCCs.

The shown scenario strengthens the necessity of a quite flexible platform-core not only with respect to functionality but also with respect to redundancy-architectures due to different mission-szenario definitions and mission-parameters. The quad-duplex architecture being realized within the SPARC project shows one possible configuration of a scalable redundancy-approach.

The reconfiguration of functional item cutset is based on the detection of failures. This means, that redundancy can just be used if failures can be detected and a switchover to an operational granulate is possible. At system-level these boundaries must be known so that the corresponding multiplicity of items can be implemented [6]. By means of dynamic functional allocation, it is even possible to use generic hardware as the XCCs of the Central Platform-Computer (CPC) in a very effective way: the amount of redundancy can be determined independently of the functional allocation and such just with respect to the minimum hardware required to fulfil probabilistic requirements for the corresponding functional execution levels. Thus, it is possible to minimize the costs of the CPC.

The schematic vehicle architecture approach is shown in Figure 12 (appendix). The overall vehicle is generally split up to two sides to be resistant against single-point