FUNCTIONAL SAFETY CONCEPTS IN MOTOR CONTROL

Anura Fernando
Underwriters Laboratories Inc.
Northbrook, IL

Abstract – *The approach for addressing functional safety of embedded software dependent controls is becoming a growing concern as programmable components replace traditional electromechanical components in safety-related systems. While many debate the pedagogical concerns regarding software complexity and reliability, it is undeniable that the motor control industry is moving forward to capitalize upon the flexibility and low apparent cost of utilizing embedded software. When this embedded software impacts the functional safety of the embedded system, risk must be managed with due diligence and in accordance with current industry practice or the "state-of-the-art." Since consensus standards represent both current accepted practice and a baseline for state-of-the-art, they provide a good framework to direct risk management activities. While some fundamental philosophical differences may exist among such emerging standards, there are also many similarities that demonstrate the universality of the present approach to software verification and validation activities (V&V) with respect to functional safety of motor control systems.*

I. INTRODUCTION

While the word "risk" has many connotations depending upon its technical, legal, social, or philosophical context, it is a concept that we all address every day. For instance, when we negotiate our way through a furnished room we choose risk-avoiding behavior. We consider, by visual inspection (and often subconscious cognition), the hazards that may be encountered, and then we plan our route with assurance that we won't trip, slip, stub our toes, or topple an expensive vase. "Risk" is, for many, the motivating force behind conscious or subconscious decision-making. Understanding risk and the tools available for risk management can prove to be invaluable to engineers striving for reliable, robust, and cost-effective designs.

Risk is generally considered to be the combination of the severity and probability of an event (hazard) that has the potential to negatively impact people, assets, potential assets, or the environment [1]. While software risks can theoretically be segregated as technical risks and business risks [2], these two domains frequently overlap when considering issues such as delayed market entry due to manifestation of technical hazards (for example undetected software anomalies) during the development process. The scope of this discussion will focus on operational risk, with respect to the functional safety of embedded system software.

In this context, functional safety is defined in IEC 61508, the standard for *Functional Safety of Electrical / Electronic / Programmable Electronic Safety-Related Systems*, as "part of the overall safety relating to the EUC [Equipment Under Control] and the EUC control system which depends on the correct functioning of the E/E/PE [Electrical / Electronic / Programmable Electronic] safety-related systems, other technology safety-related system and external risk reduction facilities." UL/IEC 60730, the standard for *Automatic Electrical Controls for Household and Similar Use*, refers to this same concept through the

definition of a “protective control” as a “control, the operation of which is intended to prevent a hazardous situation during abnormal operation of the equipment.” This standard also requires that control functions be classified as either Type 1 or Type 2. Type 1 controls are those controls that perform functions where deviation and drift of control parameters will not introduce a hazard to either the component (control) or the system (end-product where the control is installed). By contrast, Type 2 controls would be those controls having control parameters with critical deviation and drift characteristics (ie. exceeding design constraints such as limits or tolerances could result in a risk).

Thus, when evaluating functional safety, proper identification of the critical control parameters and understanding the related component- and system-level failure modes is crucial. The existing end-product industry consensus standards may be used to help identify what these safety-related parameters are.

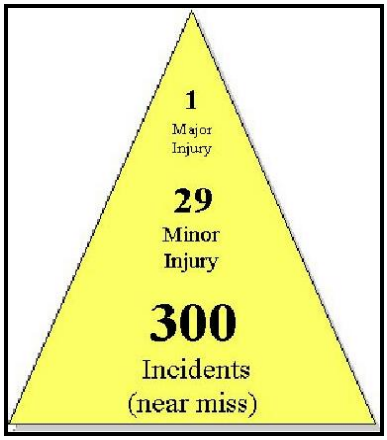
II. HISTORICAL PERSPECTIVE

Although programmable controllers did not appear in industrial application until the 1960’s [1], the earliest research on software faults appears to date back to the Electronic Numerical Integrator and Calculator (ENIAC), developed in 1946 by Dr. John Mauchly and J. Presper Eckert at the University of Pennsylvania [3]. They noted that excessive heat generated ‘faults’ in the over 18,000 valves (vacuum tubes) utilized for the execution of the manually wired “programs.” Since that time, extensive research has been conducted to understand the potential failure modes of software. The general conclusions of this research have been well encapsulated in the following excerpt from *Safeware* [4]:

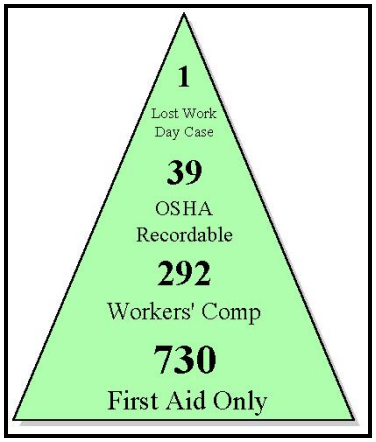
> *In control systems, the computer is usually simulating the behavior of an analog controller. Although the software may be implementing the same functions previously performed by the analog device, the translation of the function from analog to digital form may introduce inaccuracies and complications. Continuous functions can be difficult to translate to discrete functions, and the discrete functions may be much more complex to specify. In addition, the mathematics of continuous functions is well understood; mathematical analysis often can be used to predict the behavior of physical systems. The same type of analysis does not apply to discrete (software) systems. Software engineering has tried to use mathematical logic to replace continuous functions, but the large number of states and lack of regularity of most software result in extremely complex logical expressions. Moreover, factors such as time, finite-precision arithmetic, and concurrency are difficult to handle.*

Some of the earliest published standards relating to the functional safety of software in programmable systems appear to have emerged from the U.S. defense and aerospace industries as well as the process control industry. These standards rely on the concepts of ‘system safety’ engineering derived from systems theory, a discipline dating back to the early part of the last century.

After the development of the first U.S. electrical / fire safety standards, in the labs of William Henry Merrill in 1894, to later become Underwriters Laboratories in 1901 [5], the first American technical journal for accident prevention began publication in 1911. During this same period, the first U.S. safety standards addressing system-level risk management were established in 1914 as the "Universal Safety Standards" under the guidance of Carl M. Hansen. These standards prescribed the process of defining hazards and then eliminating the hazards by design. Also around this same time period, H.W. Heinrich conducted a study of 50,000 industrial accidents and published his findings in 1929. This led to a statistical basis for eliminating hazards using a model of his hypotheses known as Heinrich's pyramid [4]. This statistical relationship was again confirmed in a study conducted by the Proctor and Gamble Corporation many decades later in 1986 [6].



Heinrich's Triangle [6]



Proctor and Gamble Study [6]

Figure #1

This understanding of the potential impact of failures in protective functionality has been carried into modern risk-based control system standards such as IEC 61508 and UL/IEC 60730. It is important to remember, however, that the actual safety functions and the associated reliability metrics are driven by UL, IEC, and other harmonized industry end-product standards that capture the safety concerns relevant to the given application domain of the control system. This allows issues such as physical environment, electrical environment, user competency, and other application specific concerns or mitigation mechanisms to be considered when assessing the control system functional safety.

III. AVAILABILITY VS. RELIABILITY IN DESIGNING FOR SAFETY

One of the most significant differences between UL/IEC 60730 and IEC 61508 is the impact of whether the system may be designed to be "fail safe" or whether it must be "fail operational." UL/IEC 60730 allows for consideration of "fail safe" design, including the possibility of immediate cessation of operation, while IEC 61508 provides mechanisms for ensuring increasing levels of operational integrity of the safety

function, depending on the established Safety Integrity Level (SIL)[1]. IEC 61508 uses this SIL concept to prescribe software and electronic design considerations that may be used to satisfy the SIL requirements. UL/IEC 60730 is not typically a stand-alone document. It is made up of a "Part 1" with general requirements, which for many application domains, is referenced by 'Part 2's" or particular requirements for those specific application domains. With this structure, UL/IEC 60730 relies on the system requirements prescribed in the respective end-product standards to establish the operational characteristics (such as response times, functional availability, and tolerances) that are often under the control of the embedded software. Reliability of this software is typically one of the most important facets of assessing control system functional safety. Reliability can be generally considered as "the probability that a system or product will accomplish its designated mission in a satisfactory manner" [7]. Thus, while software can be validated at particular instants in time, the verification activities can help positively influence the stochastic aspects of the system's reliability over time.

Possibly the single most important software verification activity related to designing for functional safety is the risk analysis. Due to the highly subjective nature of what constitutes "acceptable levels of risk," neither UL/IEC 60730 nor IEC 61508 mandate specific methodologies for risk analysis. Five examples of possible risk analysis methodologies follow.

1. Fault Tree Analysis (FTA) - a means of identifying potential causes of hazards. It is a methodology developed at Bell Telephone Laboratories in 1964, initially for the aerospace, electronics, and nuclear industries. It is a top-down approach consisting of system definition, fault tree construction using logic gates and events, qualitative analysis, and quantitative analysis. Its drawbacks are that it requires foreknowledge of the system, and caution must be exercised to prevent oversight of critical paths due to oversimplification of system representation.

2. Event Trees - used extensively in business and economics. It provides advantage over the Fault Tree approach in that it breaks the overall complex system into smaller more manageable parts. An Event Tree is drawn from left to right, with the branches under the headings corresponding to the alternatives of successful performance of the safety function or failure of the safety function. Similarly to the Fault Tree, a probability can be assigned to each alternative and translated throughout the critical thread. Potential problems with timing issues and effects of common-cause failures on probability dependencies should be considered during the evaluation of this method.

3. Failure Modes, Effects, and Criticality Analysis (FMECA) - useful for the analysis of discrete failures. It has been used extensively in the reliability engineering community since it can be used to establish the overall probability that the product will operate without a failure for a specific length of time or for a specific length of time between failures. This variation supplements the traditional FMEA by introducing the concepts of "risk" and "residual risk" by additionally considering probability in the

[1] SIL – discrete level (one out of a possible four) for specifying the safety integrity requirements of the safety functions to be allocated to the E/E/PE safety-related systems, where safety integrity level 4 has the highest level of safety integrity and safety integrity level 1 has the lowest.

estimation of "criticality." This technique, although comprehensive, can be burdensome because of the need to exercise each failure mode of the device under evaluation.

4. Cause and Consequence Analysis (CCA) - starts with a "critical event" and uses a top-down approach or backward search to determine the cause of the event and the potential consequences of the event occurrence. The analysis begins with identification of a critical event. After this identification, factors are sought that constitute the critical event, and the potential effects of the event are propagated through the system. The interrelationships are then graphically represented using gates to describe relationships between cause events and vertices to describe relationships between consequences. CCA diagramming can become unwieldy due to the need for an individual diagram for each initiating event.

5. Hazard and Operability Analysis (HAZOP) - a qualitative technique used for the identification of deviation from expected operation and the hazards associated with such deviation. If conducted under ideal circumstances, this technique can identify and/or eliminate a great number of hazards. However, the "ideal circumstances" rely heavily on the experience and judgment of the engineers performing the analysis [4].

As previously mentioned, these methodologies are not specified in either UL/IEC 60730 or IEC 61508. They serve, instead, as examples of the types of risk analysis techniques that can be employed to help identify the safety functions that are to be addressed by these standards. It is the responsibility of the manufacturer to conduct such analyses prior to a third party assessment for conformity to either of these standards. While these standards leave the risk analysis approach up to the discretion of the manufacturer, they do provide guidance as to the generalized sources for failure modes of the electronic hardware and software that are to be addressed during the risk analysis process.

Many industry sectors have accepted, as evidenced by the requirements of their consensus standards, that software is susceptible to some "common cause" failures. Faults that can lead to such failures can arise from specification mistakes, implementation mistakes, external disturbances, and component defects. These faults can impact the system software as well as both the analog and digital hardware implemented in the "firmware." Such faults can be permanent, transient, or intermittent in duration. They can lead to either deterministic or non-deterministic states, depending on the design. Most importantly, they can have either local effects, isolated to subsystems, or they can globally impact the system functionality. While the global effects can have an immediate impact, it is the local effects that can be much more insidious, particularly if hidden within a safety function that is only called upon when abnormal conditions occur.

IV. EXAMPLE MOTOR CONTROL SAFETY FUNCTION

The following is an example of motor control safety functionality that is implemented in hardware whose functionality is defined by the embedded software. The basic safety concerns being considered are those of electric shock and fire. The design specification, standards profile, risk analysis, and system assumption are minimized, incomplete, and only used for purposes of illustration.

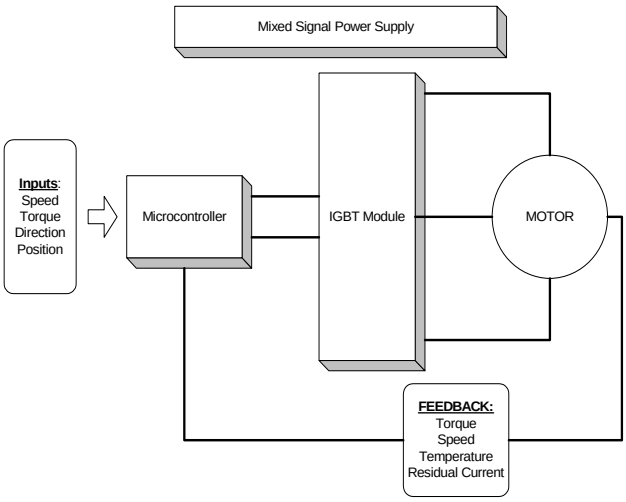


Figure #2 Example Motor Control Block Diagram [8]

For this example, we will assume that this control is intended for use in a residential appliance where a fail-safe state is defined. The second major assumption for this example is that the only required safety function for this system (per an end-product standard such as UL/IEC 60335) is “locked rotor” protection, and the intent of this requirement is to preclude over-temperature conditions resulting in either fire or insulation breakdown-related electric shock.

From a standards perspective, these assumptions point to the use of UL/IEC 60730 for the control assessment. Since UL/IEC 60730 is a harmonized functional safety standard that is part of the Certification Body (CB) Scheme, the test results that are generated by a Certification Body Test Lab (CBTL) such as UL would be accepted by other such testing organizations throughout the world. These assumptions would also point to the use of UL 2111, Overheating Protection for Motors, since the block diagram indicates that the motor is provided in combination with the control, and a locked rotor condition would lead to an over-temperature-related trip. Presently, no equivalent IEC standard is in place for the evaluation of residential use motor/control combinations; therefore such a combination would be tested in the end-appliance. These end-appliance standards vary with respect to requirements regarding the effects of loss-of-phase, locked-rotor, running-overload, or all three of these stress conditions, based on whether or not the appliance is remotely or automatically controlled.

The risk analysis for this control could be based on any of the aforementioned methodologies such as FMEA, FTA, CCA, etc. As an example, to analyze failures associated with the microcontroller specified in Figure 1, we will focus on a few of the common cause microelectronic faults (as described in UL/IEC 60730 Annex H Table H.11.12.7) that could be included in an FMEA:

Table #1 FMEA

Component	Failure Mode	Effect	Probability	Severity	Initial Risk	Mitigation	Final Risk
CPU Registers	stuck-at	Loss of temperature sense				Reciprocal comparison	
Volatile Memory	DC-Fault	Loss of temperature sense				Checker-board Test	
Clock	Harmonics	Loss of temperature sense				Time Slot Monitoring	
Etc...	Etc...	Etc...				Etc...	

In this example, the emphasis on “loss of temperature sense” is due to the declared safety function relative to thermal protection of the motor. Per UL/IEC 60730, this control would be declared as a Type 2 control, since deviation and drift associated with the temperature sensing and feedback could compromise the safety function of protecting the motor from thermally induced insulation degradation, which could result in fire or electric shock. While there are many obvious root-causes of loss of protective functionality, a thorough analysis would be required to uncover some of the even more insidious failures such as improper wave-shaping via high speed IGBT pulsing. This could also lead to a “locked rotor”-like condition wherein the windings are energized but perhaps the IGBT synchronization may not be correct (ie. loss of phase), resulting in the motor becoming essentially an inductive heater. Another possibility might be a “running overload”-like condition wherein the drive signal, due to a software fault, loses its critical characteristics relative to the impedance model of the motor, which would lead to thermal stress of the insulation system over time. The deviation and drift declarations could even be expanded to include deviation or drift from the prescribed drive signal waveform. Thus, when making declarations, consideration should be give to all critical parameters associated with the safety function(s), even if the “component” under consideration is an algorithm.

Such a control system would be subjected to the same general assessment approaches in UL/IEC 60730 as it would with any other Functional Safety standard such as IEC 61508 or IEC 61511: the safety functions would be defined, the safety lifecycle, design, and layers of protection analyzed, and both the hardware and software would be tested for robustness against physical, environmental, electrical, and electromagnetic stressors.

V. SUMMARY

Motor controls can fall into many different industry sector domains, each with its own regulatory considerations. A single motor control design could be considered for entry into many markets and sectors, which could include: Industrial, Process, Residential, Medical, Pharmaceutical, Commercial, etc. The requirements for each of these sectors could be further complicated by geography and politics, depending on whether they would be subject to the codes and regulations of North America, the European Union, the Far East, or a myriad of other possible jurisdictions. In addition to traditional conformity assessment services, UL offers many services to assist manufacturers in gaining global market access. Front-end consulting offered by UL can help manufacturers gain an understanding of the regulatory issues and related design constraints that they may face when entering multiple markets. Awareness of constraints early in the design process can enhance product safety, and minimize redesign and retesting...ultimately reducing cost and improving time to market.

VI. REFERENCES

[1] Gruhn, P, Safety Shutdown Systems: Design, Analysis, and Justification, Instrument Society of America, Research Triangle Park, North Carolina, 1998.

[2] Karolak, D.W, Software Engineering Risk Management, IEEE Computer Society Press, Los Alamitos, California, 1996.

[3] The Illustrated Science and Invention Encyclopedia, Volume 5, H.S. Stuttman Inc, publishers, Westport Connecticut, 1983

[4] Leveson, Nancy, Safeware- System Safety and Computers, Addison-Wesley Publishing, Inc., 1995

[5] Bezane, Norm, “This Inventive Century,” Underwriters Laboratories, Inc., Northbrook Illinois 1994.

[6] Downloaded from www.cbs.state.or.us/external/osha/ppt/100oh.ppt on April 18, 2003

[7] Blanchard, B.S, Systems Engineering and Analysis, Prentice Hall, Upper Saddle River, New Jersey, 1998.

[8] Downloaded from http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1483 on February 16, 2005

[9] International Electrotechnical Commission (IEC), IEC 61508 Parts 1-7, Functional Safety of Electrical / Electronic / Programmable Electronic Safety-Related Systems, First Edition, IEC, 3, Rue de Varembe, Geneva Switzerland, 1998.