Dependability for Autonomous Control with a Probability Approach

by Lan Anh Trinh, Baran Cürüklü and Mikael Ekström

For the last decade, dependability – the ability to offer a service that can be trusted - has been the focus of much research, and is of particular interest when designing and building systems. We are developing a dependable framework for an autonomous system and its control.

A shift from automatic to autonomous control has emerged in the development of robots. Autonomous control allows the robot to have freedom of movement as well the ability to directly interact with humans as well as other robots in a collaborative environment. Having a dependable platform for autonomous control becomes absolutely crucial when building such a system.

The concept of dependability originally derives from software development and can be defined as ‘the ability to deliver service that can justifiably be trusted’ (Avizienis et al. [1]). The dependability of a system is evaluated by one, several or all of the following attributes: availability, reliability, safety, integrity, and maintainability. The implementation of dependability starts with an understanding of the threats to the system’s dependability, which may include failures, errors, and faults. Therefore, four means have been developed to protect system dependability: fault prevention, fault removal, fault forecasting, and fault tolerance.

Mälardalen University is hosting a long term project, DPAC – Dependable Platforms for Autonomous system and Control, with the main aim of implementing dependability for different platforms for autonomous systems and their control. Within the profile, three main fields are covered that include hardware with heterogeneous system architecture (HAS), software design and autonomous control. In this project, the dependability of autonomous control system plays a vital role.

A fault is the root of every failure occurring inside or outside of the system. Nevertheless, the most pressing challenge is how to predict the frequency of faults and at what moment a fault occurs, thus fault analysis is presented to minimise the probability of a fault and to estimate when faults will happen in the system. Thereafter, other means are developed to protect the dependability with respect to the analysis of faults.

Various approaches to fault analysis, such as Petri Net (PN), fault tree analysis (FTA), failure modes effects and criticality analysis (FMECA), and hazard operability (HAZOP) are introduced in the work of Bernardi et al. [2]. However, our research focusses on PN, since the PN framework provides not only a probability approach for fault analysis and fault prevention in both development and operational stages of designing a system but also for mitigation of the implementation progress. For instance, as an extension of PN, a stochastic Petri net can be combined with Markovian models to evaluate the probability of current state and the probability of future fault events for a fault prognosis process. In our study [3], a coloured time PN is utilised for fault tolerance analysis of multi-agents in a complex and collaborative context.

In our current research, three different stages are considered to be implemented, which relate to three means: fault prediction, fault prevention and fault tolerance for autonomous control. For fault prediction, the probability model is established to estimate the time fails of the system. The time that a fail could happen is modelled by an exponential distribution. The design of the autonomous control system can be presented by a network of nodes like a Petri net and the fails are propagated from one node to others. Each type of failure is computed by separated variables and the hierarchy structure may be taken into account for a complicated agent interaction. The probability of a successful task depends on the parameters of the system. Note that there are two types of parameters: those from the environment that cannot be changed i.e., non-configurable, and the others that are configurable. Correspondingly, the probability of failures could be estimated based on the defined values of those parameters and the architecture of autonomous systems.

After designing a mathematics model of fault prediction with respect to configurable parameters, expectation maximisation (EM) is applied to minimise failures while taking into account hidden parameters from the environment. This stage is actually the implementation of fault prevention as we attempt to find the best model for the system to avoid further failures in future.

Finally, the fault tolerance allows the system to continue to work even as fails happen. To do so, there are different algorithms and modules running in parallel in the system. Each module will use the fault prevention design as described above. When one algorithm or module fails, it is immediately replaced by another. It is important to note that the decision must be done in advance i.e., before the fails happen, otherwise it could be too late. Again the fault analysis combined with an artificial intelligence algorithm are used for decision making.

Overall, this architecture allows all necessary means to be implemented to preserve the dependability of the system. Using a graphical probabilistic model, the means could be implemented in a real robot to facilitate dependable autonomous control.

References:

[1] A. Avizienis, et al.: "Basic concepts and taxonomy of dependable and secure computing", IEEE Transactions on Dependable and Secure Computing, 2004.

[2] S. Bernardi, J. Merseguer, D.C. Petriu: "Model-Driven Dependability Assessment of Software Systems", Springer, 2013.

[3] T. Lan Anh, C. Baran, E. Mikael: "Fault Tolerance Analysis for Dependable Autonomous Agents using Colored Time Petri Net", 9th International Conference on Agents and Artificial Intelligence, ICAART 2017.

Please contact:

Lan Anh Trinh

Mälardalen University, Sweden

anh.lan@mdh.se

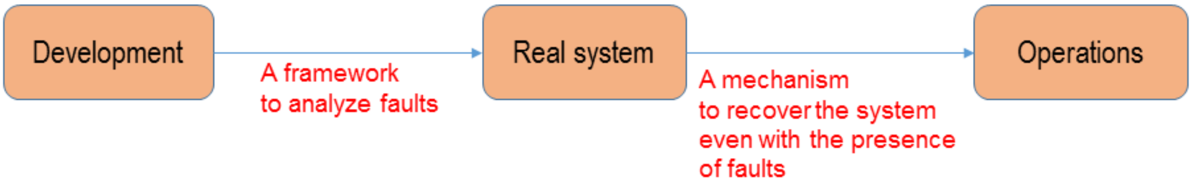

Figure 1: The development and operational stages in building a dependable control system.