Data Intensive Engineering course

TOPIC x

Title: Reliability evaluation of distributed embedded systems

Contact Person: Alberto Ballesteros (alberto.ballesteros@uib.es)

Julian Proenza (julian.proenza@uib.es)

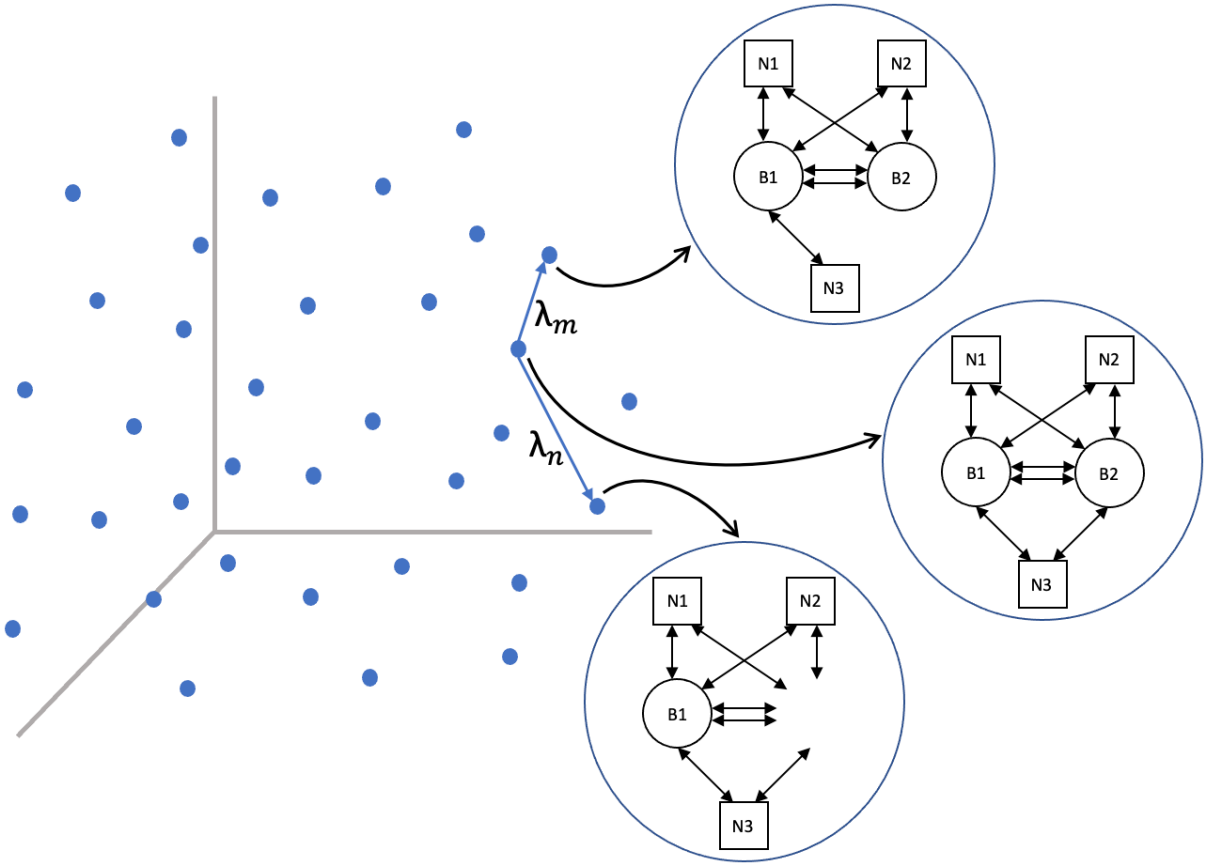
Project objectives: Distributed embedded systems that need to exhibit high reliability must be designed using specific techniques to ensure a high probability that they provide a correct service in a continuous manner. When focusing on the design of the architecture, to provide high reliability a very common technique is **fault tolerance**. Fault tolerance roughly consists in designing the system to perform the computation in a redundant manner to provide correct service even in the presence of faults.

When designing distributed systems, the network is key for the system to carry out a correct operation. Thus, for the overall system to be highly reliable, the network must exhibit high reliability too. Furthermore, to **guarantee** that the reliability of the system is sufficient, once the system is designed, it must be **modelled,** and proper reliability analyses must be executed. This process is usually iterative, i.e., the results of the reliability analysis lead to changes in the design of the system, requiring to build a new model and evaluate the reliability again.

Unfortunately, designing and modelling highly reliable systems is costly in terms of time (from weeks to months). Furthermore, it is highly dependent on the expertise of the designer. For this reason, we have proposed a modelling technique that allows to explore the design space of a distributed embedded systemin an automatic manner.

This technique consists in modelling the system architecture using **colored graphs**. Specifically, the designer should provide the description of the system (which types of components it contains, how they are connected and how they fail) and the restrictions of the architecture (maximum number of components it can contain, and minimum number of components required by the system to provide a correct service).

We have developed a tool that generates the graph of the architecture with the maximum number of components (max architecture), and then generates all the subsets of architectures. Each one of the generated architectures can be seen as the result of the failure of one component. Figure 1 depicts this process. The blue dots represent all the architectures generated, while on the right side we see examples of architectures. The rightmost architecture in the Figure shows an example of max architecture, while the other two architectures are the result of eliminating a component from the max architecture. Finally, we see that these architectures are connected using probabilities, which represent the probability of the eliminated component failing.



In this way the design space is modelled using a single graph that is then converted to a Continuous Time Markov Chain (CTMC) that can be fed to modelling tools to calculate the reliability of each one of the architectures.

The objective of this work is to use machine learning techniques to automatise and speed-up the reliability evaluation after the failure of some components or after changes in the system's requirements. In this way, this work would constitute an essential part of novel applications that allowed, among other things, the following:

- Trigger a warning when the reliability of the system goes below a certain threshold in supervisory systems for the monitoring of systems, e.g., SCADA systems.
- Re-evaluate the reliability of the system when new components are added, e.g., when a new energy production component is added to a microgrid.
- Adapt the fault tolerance capacity of the system when the reliability requirements change, e.g., reduce the redundancy used to safe energy when the reliability requirements decrease.

Ultimately it opens room for the automation of the design of systems by exploiting the advantages of the exploration of the design space.

Schedule and milestones:

- Part 1 – Produce the training dataset
  1. Select a max architecture and generate the CTMC using the tool provided.
  2. Calculate the reliability of each generated architecture using CTMC evaluation tools.

- Part 2 – Select the adequate ML algorithm
    1. Select the most adequate technique to carry out the estimation of the reliability of an architecture based on the results obtained.

- Part 3 – Build the model
    1. Use the dataset obtained from part 1 to train the algorithm selected in part 2.