4th Home Assignment – Safety Analysis

Evaluating an Adaptive Cruise Control

In this phase the team should focus on the safety of a crucial component of the autonomous vehicle: the Adaptive Cruise Control (ACC) component. Basic Cruise Control can maintain the speed set by the driver by accelerating or decelerating. ACC extends this functionality with sensing the distance of the vehicle in front of us and adjusting our speed to keep a safe following distance. As a first step, we will focus on only the speed information (and not the distance yet).

The speed is measured using two different methods simultaneously: *1)* an 3DM-GX5-10 *IMU* (Inertial Measurement Unit) provides movement data directly based on its internal sensors, while *2)* the movement of the four wheels are measured independently using *encoders*, one for each wheel. The data of the encoders is fed into a *microcontroller*, which runs an algorithm that provides usable speed data if at least half of the encoders are working correctly. The aggregated data from this microcontroller is then sent to the main logic unit. The system does not stop if the output of an encoder is detected to differ from the others, as this difference can easily be transient, and the different wheels can sometimes really move with different speeds.

The two speed measurements are processed by the *main logic unit* of the ACC, which is another microcontroller. The *maximum* of the two measurements is used to increase safety. The output is the command to accelerate, decelerate or maintain speed. In order to eliminate single points of failure, the *main logic unit* is duplicated.

A small *voter FPGA* is used as a shield to derive the final command or initiate a safe emergency shutdown of the system. Currently, if the commands from the two main logic units are the same, the final command will be that, otherwise the system shuts down. In any potential improved design, this FPGA may be used to implement more sophisticated comparisons and selection strategies (see tasks *d.* and *g.*). The voter FPGA is also duplicated, but their comparison is not realized by a separate voter – instead, they receive the output of the other FPGA and constantly check their own output against that to initiate safe emergency shutdown when they see any discrepancy.

In case of an emergency shutdown, as the problem might be transient, the system is automatically restarted. Note: The failure rates in this task include only permanent failures, but the functional logic of the system must consider the possibility of transient failures too. In the final implementation, this will be replaced with a better solutions to reliably detect permanent faults, but for the prototype, this will suffice. A practical consequence of this is that both voters can become faulty, in which case they might produce the *same erroneous output*.

In the whole system, communication through wires or channels is assumed to be either reliable or the probability of communication faults is already considered in the reliability metrics of the receiving component.

Each microcontroller's power is provided by the single *main power supply* through a dedicated *5V power regulator* (each microcontroller has its own regulator). If the corresponding regulator fails to provide the correct voltage, the microcontroller may start to operate outside of the rated working range, which leads to non-deterministic behavior. Every other component that needs external power is supplied from the main power supply without any additional regulator in the power chain. (Note: If voltage conversion would be logically needed for such components, you can treat it as already being included in the component's failure rate.) *The failure of the main power supply can be ignored in the current design phase.*

At the end of each day (treat a day as 24h, regardless of the actual duration of active operation), the vehicles return to the maintenance station for *full maintenance*. We can assume for now that this maintenance leads to full recovery, meaning that the vehicles start each day as if they were new.

The failure rates for some components are given in the following table:

Component type	**Failure rate (FIT)**
Encoder	580
FPGA	11
5V regulator	15

We do not know the exact failure rate of the microcontrollers, but they are *SIL2 certified*, and are used in a *continuous operation mode*. The reliability properties of the IMU can be found in its public *datasheet* (note that in practice, vendors often use MTBF instead of MTTF because repair time is negligible).

Tasks

Safety analysis

a. Draw a *fault tree model* of the ACC with the event "The ACC commands the vehicle to accelerate when it should not" as the top event.
b. *Qualitative analysis:* What are the minimal cut sets? Are there any SPoFs?
c. *Quantitative analysis:* Calculate the probability that the top event happens before the daily maintenance on a given day. To give a conservative estimate, assume that a faulty component has the worst possible behavior. Based on this, what is the probability that the top event happens in the 4-year mission time of a bus? *Hint: probabilities of the basic events should be computed based on the fault rates of the components, assuming an exponential distribution. Fault rates may have to be derived from other metrics.*
d. The target probability for the 4-year mission time is 10^{-6}. Introduce *new redundancies* to the system (as few as possible) so that the target is reached and provide a fault tree for the new architecture. *If the newly introduced components provide redundant data, describe how the final value is derived and how this affects the fault tree.*

Availability analysis

e. Draw a *fault tree model* of the original ACC design with the event "The ACC is unavailable for some reason" as the top event. Unlike in the previous part, this includes the safe error states. Unavailability here means that the system cannot perform its function as expected, not only when it does not answer at all. *Note: If the system is in an undetected dangerous state (as described in the previous task), it may seem available, but the chance of this is negligible. We suggest to ignore these cases while computing availability.*
f. *Quantitative analysis:* What is the *availability value* of the ACC? To give a conservative estimate, assume that a faulty component is not available.
g. (IMSc) The target unavailability of the ACC as derived from the overall target unavailability of the vehicle is 10^{-5}. Introduce new redundancies to the system (as few as possible) so that this target and the safety requirement in task d. are both satisfied - provide the two fault trees for the new architecture.

Extra Homework Conditions

Modeling and calculations may be performed in a suitable tool (e.g. Open Reliability Editor) or manually (e.g. in yEd or on paper) – the only requirement is that the solution must be thoroughly documented, including (in addition to the usual elements) the calculation of basic event probabilities, every fault tree that has been built, and either the calculations or the way the chosen tool was used to derive the final results.

In other words, the solution must be fully reproducible based on the documentation, as no other artifact is delivered this time.