Work-in-Progress Abstract: The impact of the period variation on execution time distributions of programs

Liliana Cucu-Grosjean
liliana.cucu@inria.fr

Avner Bar-Hen
avner@cnam.fr

Yves Sorel
yves.sorel@inria.fr

Hadrien Clarke
hadrien.clarke@inria.fr

Abstract—Designers of embedded real-time systems derive, in general, their time parameters such as activation periods from those of sensors or actuators. By designers, we mean the team in charge of conceiving embedded real-time systems. This team includes Control Theory designers and Computer Science designers. Within this paper we present the point of view of Computer Science designers, while the periods proposed by Control Theory designers are supposed robust with respect to the physical behavior of the system.

The execution times are, then, estimated by studying statically the programs structure or dynamically the programs execution. In some cases, both activation periods and execution times depend on a sensor information. For instance, they depend on the angular speed of wheels within an automotive embedded real-time system and such systems follow a rate-dependent model.

Elastic tasks is another model, where one may consider execution time variation depending on the selected period. Within this paper, we are interested in describing statistically the relationship between activation periods and execution times of programs. More precisely, we study the impact of the period variation on the distributions of the execution times. To illustrate our preliminary results, we consider, as case study, the set of programs executing the autopilot of an open-source PX4 drone.

Index Terms—real-time systems, execution time distributions

I. OUR MOTIVATION

The design of embedded real-time systems is characterized by the verification of both functional and timing correctness. The timing verification is, often, done with respect to some sensors that allow the system to receive information from the execution environment. While the activation period of programs implementing embedded real-time systems are inherited from the sensors frequency, designers may decide to vary these periods.

For instance, in Figure 1 we consider the graph of communications between programs of the PX4 autopilot 1. The Sensors program has an activation period of 4ms which is also the period of EKF2, mc_position_cont, control_attitude and output driver programs. We may notice, also, that the GPS program with an activation period of 200ms has a smaller activation period than the period of the EKF2 program.

The relationship between the activation period and the execution time has received an increased interest from the real-time community since the keynote of Buttle at ECRTS 2012. An interested reader may find more details on the rate-dependent tasks within a dedicated survey [1] presenting the existing schedulability results.

Fig. 1. The graph of a PX4 autopilot describing the dependencies between programs

This research is partially funded by the FR PSPC STARTEC and the Inria ADT KDBench projects.

1See more details at https://px4.io.

II. THE PROBLEM DESCRIPTION AND PRELIMINARY RESULTS ON A DRONE PX4-RT CASE STUDY

We consider a set $\tau$ of $n$ periodic implicit deadline tasks $\{\tau_1, \tau_2, \ldots, \tau_n\}$ scheduled according to a fixed-priority preemptive scheduling policy on a single core processor. A task $\tau_i$ is described by $(C_i, T_i, D_i)$, $\forall 1 \leq i \leq n$ with the period $T_i$ of a task $\tau_i$ that may have $m$ several possible values $\{T_i^1, T_i^2, \ldots, T_i^m\}$ and the deadline $D_i = T_i$. The activation period does not vary during the execution, but it may be modified from one execution scenario to another. $C_i^j$ is the probabilistic worst-case execution time, which is an upper bound on any distribution of execution times obtained for an execution scenario $S_j$ with a period $T_j^j$, $\forall 1 \leq j \leq m$. Moreover, we associate to the set of tasks $\tau$, a DAG $G$ defined by $(V, E)$, where the set of vertices $V$ is equal to $\{\tau_i\}_{1 \leq i \leq n}$.
and the set of edges $E_i$ describes the communication between tasks. We say that $(\tau_i, \tau_j) \in E$, if a task $\tau_i$ communicates with a task $\tau_j$, i.e., the input variables of the program $\tau_j$ are among output variables of the program $\tau_i$. Nevertheless, the existence of these edges do not impose any order within a feasible schedule, nor cause a priority inversion.

Existing probabilistic schedulability analyses [3] are, often, built on an independence hypothesis between the periods and the execution times of a task, a commonly accepted hypothesis while dealing with unprobabilistic description of the time parameters. Within this paper, we present preliminary results underlining the existence of a dependence between the two time parameters $T_i$ and $C_i$ and with a future objective of a sensitivity analysis allowing to formalize it.

To illustrate our preliminary results, we consider the PX4 autopilot of a drone, an open source flight control software [4] designed by the KTH Zurich, which interacts, through sensors, e.g., IMU and actuators (motors), with the physical environment. The latter is mainly composed of air in motion around objects like trees, power lines or buildings. The autopilot controls the position and attitude of the drone while following a given trajectory. The trajectory may be generated by a human using a radio controller for a manual mission or stored in the SD card of the drone, as a set of GPS points, for an autonomous mission. In this case the drone, itself, has to comprehend with its sensors the physical process in order to achieve the expected mission.

Our numerical results are obtained by executing tasks on an ARM Cortex-M4 unicore microcontroller included in a Pixhawk board. The PX4 tasks run on top of NuttX, a Unix-like OS. The tasks are executed according a preemptive fixed priority scheduler. We consider two flights following exactly the same trajectory. The periods are the same for all tasks, except for the Sensors task, which calibrates the sensors and has an activation period of $4\text{ms}$ for the first flight and a period of $5\text{ms}$ for the second one (see the list of all activation periods in Table I).

In Figures 2 and 3, the horizontal axis describes the order of appearance of the execution times, while the vertical axis represents the values of the execution times in ms. From these figures, we notice that the execution times of the Sensors task for $T = 4\text{ms}$ and $T = 5\text{ms}$ are different. Indeed, there are several execution modes with larger values for the activation period $T = 5\text{ms}$. By execution mode, we mean a cluster of points that are close to each other. This difference between the two clusters may be explained by the variation of the values of the input variables of the Sensors task and the existence of several paths within this task.

### III. A Statistical Sensitivity Analysis

In order to study the impact of the input variables on the execution times and describe the dependence between periods and execution times, we monitor the evolution of the inputs. For instance, in Figures 4 and 5 we notice the important difference between the values for two sensors (angle in blue and velocity in orange) during two flights with trajectories and external conditions that are as close as possible. These sensors provide values for input variables of the Sensors task.

We believe a statistical sensitivity analysis may describe the existing dependencies. Such analysis studies the uncertainties within the inputs and the outputs of a task, while considering the task as a black box. Our plan is to gradually introduce information on the task structure (control graph describing the paths, for instance).
REFERENCES


