

RESEARCH CENTRE

Paris

2021

ACTIVITY REPORT

Project-Team

KOPERNIC

**Keeping worst case reasoning for different  
criticalities**

**DOMAIN**

**Algorithmics, Programming, Software  
and Architecture**

**THEME**

**Embedded and Real-time Systems**

# Contents

<b>Project-Team KOPERNIC</b>	<b>1</b>
<b>1 Team members, visitors, external collaborators</b>	<b>2</b>
<b>2 Overall objectives</b>	<b>3</b>
<b>3 Research program</b>	<b>5</b>
3.1 Worst case execution time estimation of a program	5
3.2 Building measurement-based benchmarks	6
3.3 Scheduling of graph tasks on different resources within an energy budget	7
<b>4 Application domains</b>	<b>7</b>
4.1 Avionics	7
4.2 Railway	8
4.3 Autonomous cars	8
4.4 Drones	8
<b>5 Social and environmental responsibility</b>	<b>8</b>
5.1 Impact of research results	8
<b>6 Highlights of the year</b>	<b>8</b>
6.1 Awards	8
<b>7 New software and platforms</b>	<b>8</b>
7.1 New software	8
7.1.1 SynDEx	8
7.1.2 EVT Kopernic	9
<b>8 New results</b>	<b>10</b>
8.1 Worst case execution time estimation of a program	10
8.2 Building measurement-based benchmarks	12
<b>9 Bilateral contracts and grants with industry</b>	<b>13</b>
9.1 CIFRE Grant funded by StatInf	13
<b>10 Partnerships and cooperations</b>	<b>14</b>
10.1 International initiatives	14
10.1.1 Associate Teams in the framework of an Inria International Lab or in the framework of an Inria International Program	14
10.2 International research visitors	14
10.3 National initiatives	14
10.3.1 FUJ	14
10.3.2 PSpC	14
<b>11 Dissemination</b>	<b>14</b>
11.1 Promoting scientific activities	14
11.1.1 Scientific events: selection	14
11.1.2 Journal	15
11.1.3 Leadership within the scientific community	15
11.1.4 Scientific expertise	15
11.1.5 Research administration	15
11.2 Teaching - Supervision - Juries	16
11.2.1 Teaching	16
11.2.2 Supervision	16
11.2.3 Juries	16
11.3 Popularization	17

11.3.1 Articles and contents	17
11.3.2 Interventions	17
<b>12 Scientific production</b>	<b>17</b>
12.1 Major publications	17
12.2 Publications of the year	17
12.3 Cited publications	18

# Project-Team KOPERNIC

*Creation of the Project-Team: 2021 October 01*

## Keywords

### Computer sciences and digital sciences

- A1.1.1. – Multicore, Manycore
- A1.5. – Complex systems
  - A1.5.1. – Systems of systems
  - A1.5.2. – Communicating systems
- A2.3. – Embedded and cyber-physical systems
  - A2.3.1. – Embedded systems
  - A2.3.2. – Cyber-physical systems
  - A2.3.3. – Real-time systems
- A2.4.1. – Analysis

### Other research topics and application domains

- B5.2. – Design and manufacturing
  - B5.2.1. – Road vehicles
  - B5.2.2. – Railway
  - B5.2.3. – Aviation
  - B5.2.4. – Aerospace
- B6.6. – Embedded systems

## 1 Team members, visitors, external collaborators

### Research Scientists

- Liliana Cucu [Team leader, Inria, Researcher, HDR]
- Yasmina Abdeddaïm [Université Gustave Eiffel, ESIEE Paris, LIGM, Researcher, from Feb 2021 until Aug 2021]
- Yves Sorel [Inria, Senior Researcher, until Feb 2021]

### Faculty Member

- Avner Bar-Hen [CNAM, Professor]

### Post-Doctoral Fellow

- Evariste Ntaryamira [Université de Nanterre, from Sep 2021]

### PhD Students

- Mohamed Amine Khelassi [Université Gustave Eiffel, from Oct 2021]
- Evariste Ntaryamira [CY Paris Université, until Aug 2021]
- Marwan Wehaiba El Khazen [StatInf, CIFRE, From 01/10/2020 to 30/09/2023]
- Kevin Zagalo [Inria]

### Technical Staff

- Rihab Bennour [Inria, Engineer, from Apr 2021]
- Hadrien Clarke [Inria, Engineer, from Apr 2021]

### Interns and Apprentices

- Marc Antoine Auvray [Inria, Apprentice, from Sep 2021]
- Mohamed Amine Khelassi [Inria, from Mar 2021 until Sep 2021]
- Assie Koffi [Inria, from Apr 2021 until Aug 2021]

### Administrative Assistants

- Christine Anocq [Inria]
- Nelly Maloisel [Inria]

### External Collaborators

- Yasmina Abdeddaïm [Université Gustave Eiffel, ESIEE Paris, LIGM, Since September 2021]
- Slim Ben Amor [StatInf, from January 2021]
- Rihab Bennour [StatInf, until Feb 2021]
- Adriana Gogonel [StatInf]
- Kossivi Koumblenou [StatInf, from January 2021]
- Yves Sorel [External collaborator, from Feb 2021]

## 2 Overall objectives

The Kopernic members are focusing their research on studying **time for embedded communicating systems**, also known as cyber-physical systems. More precisely, the team proposes a **system-oriented solution** to the problem of studying time properties of the cyber components of a CPS. The solution is expected to be obtained by composing probabilistic and non-probabilistic approaches for CPSs. Moreover, statistical approaches are expected to validate existing hypotheses or propose new ones for the models considered by probabilistic analyses.

*The term cyber-physical systems refers to a new generation of systems with integrated computational and physical capabilities that can interact with humans through many new modalities* [14]. A defibrillator, a mobile phone, an autonomous car or an aircraft, they all are CPSs. Beside constraints like power consumption, security, size and weight, CPSs may have cyber components required to fulfill their functions within a limited time interval (a.k.a. time dependability), often imposed by the environment, e.g., a physical process controlled by some cyber components. The appearance of communication channels between cyber-physical components, easing the CPS utilization within larger systems, forces cyber components with high criticality to interact with lower criticality cyber components. This interaction is completed by external events from the environment that has a time impact on the CPS. Moreover, some programs of the cyber components may be executed on time predictable processors and other programs on less time predictable processors.

Different research communities study separately the three design phases of these systems: the modeling, the design and the analysis of CPSs [27]. These phases are repeated iteratively until an appropriate solution is found. During the first phase, the behavior of a system is often described using model-based methods. Other methods exist, but model-driven approaches are widely used by both the research and the industry communities. A solution described by a model is proved (functionally) correct usually by a formal verification method used during the analysis phase (third phase described below).

During the second phase of the design, the physical components (e.g., sensors and actuators) and the cyber components (e.g., programs, messages and embedded processors) are chosen often among those available on the market. However, due to the ever increasing pressure of smartphone market, the microprocessor industry provides general purpose processors based on multicore and, in a near future, based on manycore processors. These processors have complex architectures that are not time predictable due to features like multiple levels of caches and pipelines, speculative branching, communicating through shared memory or/and through a network on chip, internet, etc. Due to the time unpredictability of some processors, nowadays the CPS industry is facing the great challenge of estimating worst case execution times (WCETs) of programs executed on these processors. Indeed, the current complexity of both processors and programs does not allow to propose reasonable worst case bounds. Then, the phase of design ends with the implementation of the cyber components on such processors, where the models are transformed in programs (or messages for the communication channels) manually or by code generation techniques [17].

During the third phase of analysis, the correctness of the cyber components is verified at program level where the functions of the cyber component are implemented. The execution times of programs are estimated either by static analysis, by measurements or by a combination of both approaches [36].

These WCETs are then used as inputs to scheduling problems [29], the highest level of formalization for verifying the time properties of a CPS. The programs are provided a start time within the schedule together with an assignment of resources (processor, memory, communication, etc.). Verifying that a schedule and an associated assignment are a solution for a scheduling problem is known as a schedulability analysis.

The current CPS design, exploiting formal description of the models and their transformation into physical and cyber parts of the CPS, ensures that the functional behavior of the CPS is correct. Unfortunately, there is no formal description guaranteeing today that the execution times of the generated programs is smaller than a given bound. Clearly all communities working on CPS design are aware that **computing takes time** [26], but there is **no CPS solution guaranteeing time predictability** of these systems as **the processors appear late within the design phase** (see Figure 1). Indeed, the choice of the processor is made at the end of the CPS design process, after writing or generating the programs.

Since the processor appears late within the CPS design process, the CPS designer in charge of estimating the worst case execution time of a program or analyzing the schedulability of a set of programs

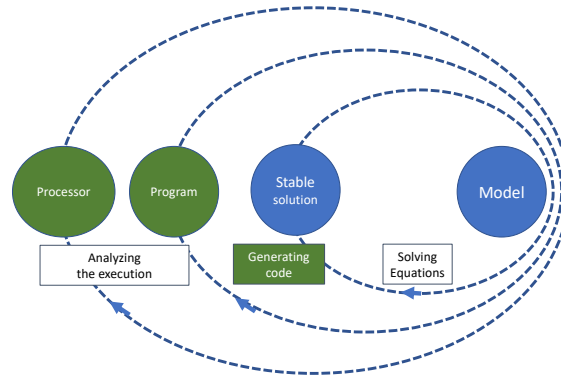


Figure 1: The CPSs design: from models towards analyzing the time properties of the cyber components

inherits a difficult problem. The Kopernic main purpose is the proposition of compositional rules with respect to the time behaviour of a CPS, allowing to restrain the CPS design to analyzable instances of the WCET estimation problem and of the schedulability analysis problem.

With respect to the WCET estimation problem, we say that a rule  $\circ$  is compositional for any two sets of measured execution times  $\mathcal{C}_1$  and  $\mathcal{C}_2$  of a program  $A$ , and a WCET statistical estimator  $p$ , if we obtain a safe WCET estimation for  $A$  from  $p(\mathcal{C}_1 \circ \mathcal{C}_2)$ . For instance,  $\mathcal{C}_1$  may be the set of measured execution times of the program  $A$  while all processor features except the local cache L1 are deactivated, while  $\mathcal{C}_2$  is obtained, similarly, with a shared L2 cache activated. We consider that the variation of all input variables of the program  $A$  follows the same sequence of values, when measuring the execution time of the program  $p$ . With respect to the schedulability analysis problem, we are interested in analyzing graphs of communicating programs. A program  $A$  communicates with a program  $B$  if input variables of the program  $B$  are among output variables of the program  $A$ . A graph of communicating programs is a direct acyclic graph with programs as vertices. An edge from a program  $A$  to program  $B$  is defined if  $A$  communicates with  $B$ . The end to end response time of such graph is the longest path from any source vertex to any sink vertex of the graph, if there is, at least one path between these two vertices. A rule  $\odot$  is compositional for any set of measured response times  $\mathcal{R}_A$  of program  $A$ , any set of measured response times  $\mathcal{R}_B$  and a schedulability analysis  $S$  if we obtain a safe schedulability analysis from  $S(\mathcal{R}_A \odot \mathcal{R}_B)$ .

Before enumerating our scientific objectives, we introduce the concept of variability factors. More precisely, the time properties of a cyber component are subject to variability factors. We understand by variability the distance between the smallest value and the largest value of a time property. With respect to the time properties of a CPS, the factors may be classified in three main classes:

- program structure: for instance, the execution time of a program that has two main branches is obtained, if appropriate composition principles apply, as the maximum between the largest execution time of each branch. In this case the branch is a variability factor on the execution time of the program;
- processor structure: for instance, the execution time of a program on a less predictable processor (e.g., one core, two levels of cache memory and one main memory) will have a larger variability than the execution time of the same program executed on a more predictable processor (e.g., one core, one main memory). In this case the cache memory is a variability factor on the execution time of the program;
- execution environment: for instance, the appearance of a pedestrian in front of a car triggers the execution of the program corresponding to the brakes in an autonomous car. In this case the pedestrian is a variability factor for the triggering of the program.

We identify three main scientific objectives to validate our research hypothesis. The three objectives are presented from program level, where we use statistical approaches, to the level of communicating programs, where we use probabilistic and non-probabilistic approaches.

The Kopernic scientific objectives are:

- **[O1] worst case execution time estimation of a program** - modern processors induce an increased variability of the execution time of programs, making difficult (or even impossible) a complete static analysis to estimate such worst case. Our objective is to propose a solution composing probabilistic and non-probabilistic approaches based both on static and on statistical analyses by answering the following **scientific challenges**:
  1. **a classification of the variability of execution times** of a program with respect to the processor features. The difficulty of this challenge is related to the definition of an element belonging to the set of variability factors and its mapping to the execution time of the program.
  2. **a compositional rule** of statistical models associated to each variability factor. The difficulty of this challenge comes from the fact that a global maximum of a multicore processor cannot be obtained by upper bounding the local maxima on each core.
- **[O2] deciding the schedulability of all programs running within the same cyber component, given an energy budget** - in this case the programs may have different time criticalities, but they share the same processor, possibly multicore<sup>1</sup>. Our objective is to propose a solution composing probabilistic and non-probabilistic approaches based on answers to the following **scientific challenges**:
  1. **scheduling algorithms taking into account the interaction between different variability factors**. The existence of time parameters described by probability distributions imposes to answer to the challenge of revisiting scheduling algorithms that lose their optimality even in the case of an uncore processor [30]. Moreover, the multicore partitioning problem is recognized difficult for the non-probabilistic case [34];
  2. **schedulability analyses** based on the algorithms proposed previously. In the case of predictable processors, the schedulability analyses accounting for operating systems costs increase the time dependability of CPSs [32]. Moreover, in presence of variability factors, the composition property of non-probabilistic approaches is lost and new principles are required.
- **[O3] deciding the schedulability of all programs communicating through predictable and non-predictable networks, given an energy budget** - in this case the programs of the same cyber component execute on the same processor and they may communicate with the programs of other cyber components through networks that may be predictable (network on chip) or non-predictable (internet, telecommunications). Our objective is to propose a solution to this challenge by analysing schedulability of programs, for which existing (worst case) probabilistic solutions exist [31], communicating through networks, for which probabilistic worst-case solutions [18] and average solutions exist [28].

### 3 Research program

The research program for reaching these three objectives is organized according three main research axes

- Worst case execution time estimation of a program, detailed in Section 3.1;
- Building measurement-based benchmarks, detailed in Section 3.2;
- Scheduling of graph tasks on different resources within an energy budget in Section 3.3.

#### 3.1 Worst case execution time estimation of a program

The temporal study of real-time systems is based on the estimation of the bounds for their temporal parameters and more precisely the WCET of a program executed on a given processor. The main analyses for estimating WCETs are static analyses [36], dynamic analyses [19], also called measurement-based analyses, and finally hybrid analyses that combine the two previous ones [36].

<sup>1</sup>This case is referred as a mixed criticality approach.



The Kopernic approach for solving the WCET estimation problem is based on (i) the identification of the impact of variability factors on the execution of a program on a processor and (ii) the proposition of compositional rules allowing to integrate the impact of each factor within a WCET estimation. Historically, the real-time community had illustrated the distribution of execution times for programs as heavy-tailed ones as intuitively the large values of execution times of programs are agreed to have a low probability of appearance. For instance Tia et al. are the first underlining this intuition within a paper introducing execution times described by probability distributions within a single core schedulability analysis [35]. Since [35], a low probability is associated to large values of execution times of a program executed on a single core processor. It is, finally, in 2000 that the group of Alan Burns, within the thesis of Stewart Edgar [20], formalizes this property as a conjecture indicating that a maximal bound on the execution times of a program may be estimated by the Extreme Value Theory [23]. No mathematical definition of what represents this bound for the execution time of a program has been proposed at that time. Two years later, a first attempt to define this bound has been done by Bernat et al. [16], but the proposed definition is extending the static WCET understanding as a combination of execution times of basic blocks of a program. Extremely pessimistic, the definition remains intuitive, without associating a mathematical description. After 2013, several publications from Liliana Cucu-Grosjean group at Inria Nancy introduce a mathematical definition of a probabilistic worst-case execution time, respectively, probabilistic worst-case response time, as an appropriate formalization for a correct application of the Extreme Value Theory to the real-time problems.

We identify the following open research problems related to the first research axis:

1. the generalization of modes analysis to multi-dimensional, each dimension representing a program when several programs cooperate;
2. the proposition of a rules set for building programs that are time predictable for the internal architecture of a given single core and, then, of a multicore processor;
3. modeling the impact of processor features on the energy consumption to better consider both worst case execution time and schedulability analyses considered within the third research axis of this proposal.

### 3.2 Building measurement-based benchmarks

The real-time community is facing the lack of benchmarks adapted to measurement-based analyses. Existing benchmarks for the estimation of WCET [33, 24, 21] have been used to estimate WCETs mainly for static analyses. They contain very simple programs and are not accompanied by a measurement protocol. They do not take into account functional dependencies between programs, mainly due to shared global variables which, of course, influence their execution times. Furthermore, current benchmarks do not take into account interferences due to the competition for resources, e.g., the memory shared by the different cores in a multicore. On the other hand, measurement-based analyses require execution times measured while executing programs on embedded processors, similar to those used in the embedded systems industry. For example, the mobile phone industry uses multicore based on non predictable cores with complex internal architecture, such as those of the ARM Cortex-A family. In a near future, these multicore will be found in critical embedded systems found in application domains such as avionics, autonomous cars, railway, etc., in which the team is deeply involved. This increases dramatically the complexity of measurement-based analyses compared to analyses performed on general purpose personal computers as they are currently performed.

We understand by measurement-based benchmarks a 3-uple composed by a program, a processor and a measurement protocol. The associated measurement protocols should detail the variation of the input variables (associated to sensors) of these benchmarks and their impact on the output variables (associated to actuators), as well as the variation of the processor states.

Proposing reproducibility and representativity properties that measurement-based benchmarks should follow is the strength of this research axis. We understand by the reproducibility, the property of a measurement protocol to provide the same ordered set of execution times for a fixed pair (program, processor). We understand by the representativity, the existence of a (sufficiently small) number of values

for the input variables allowing a measurement protocol to provide an ordered set of execution times that ensure a convergence for the Extreme Value Index estimators.

Within this research axis we identify the following open problems:

1. proving reproducibility and representativity properties while extending current benchmarks from predictable uniprocessors (e.g., ARM Cortex-M4) to non predictable ones (e.g., ARM Cortex-A53 or Cortex-A7);
2. proving reproducibility and representativity properties while extending uniprocessor benchmarks to multicore processors. In this context, we face the supplementary difficulty of defining the principles that an operating system should satisfy in order to ensure a real-time behaviour.

### 3.3 Scheduling of graph tasks on different resources within an energy budget

As stressed in the previous sections, the utilisation of multicore processors is the current trend of the CPS industry. On the other hand, following the model-driven approach, the functional description of the cyber part of the CPS, is performed as a graph of dependent functions, e.g., a block diagram of functions in Simulink, the most widely used modeling/simulation tool in industry. Of course, a program is associated to every function. Since the graph of dependent programs becomes a set of dependent tasks when real-time constraints must be taken into account, we are facing the problem of verifying the schedulability of such dependent task sets when it is executed on a multicore processor.

Directed Acyclic Graphs (DAG) are widely used to model different types of dependent task sets. The typical model consists of a set of independent tasks where every task is described by a DAG of dependent sub-tasks with the same period inherited from the period of each task [15]. In such DAG, the sub-tasks are vertices and edges are dependencies between sub-tasks. This model is well suited to represent, for example, the engine controller of a car described with Simulink. The multicore schedulability analysis may be of two types, global or partitioned. To reduce interference and interactions between sub-tasks, we focus on partitioned scheduling where each sub-task is assigned to a given core [22].

In order to propose a general DAG task model, we identify the following open research problems:

1. solving the schedulability problem where probabilistic DAG tasks are executed on predictable and non predictable processors, and such that some tasks communicate through predictable networks, e.g., inside a multicore or a manycore processor, and non-predictable networks, e.g., between these processors through internet. Within this general schedulability problem; we consider five main classes of scheduling algorithms that we adapt to solve probabilistic DAG task scheduling problems. We compare the new algorithms with respect to their energy-consumption in order to propose new versions with a decreased energy consumption by integrating variation of frequencies for processor features like CPU or memory accesses.
2. the validation of the proposed framework on our multicore drone case study. To answer to the challenging objective of proposing time predictable platforms for drones, we currently migrate the PX4-RT programs on heterogeneous architectures. This includes an implementation of the scheduling algorithms detailed within this research axis within current operating system, NuttX<sup>2</sup>.

## 4 Application domains

### 4.1 Avionics

The time critical solutions in this context are based on temporal and spatial isolation of the programs and the understanding of multicore interferences is crucial. Our contributions belong mainly to the solutions space for the objective [O1] identified previously.

---

<sup>2</sup>For more details, see the NuttX [webpage](#)

## 4.2 Railway

The time critical solutions in this context concern both the proposition of an appropriate scheduler and associated schedulability analyses. Our contributions belong to the solutions space of problems dealt within objectives [O1] and [O2] identified previously.

## 4.3 Autonomous cars

Autonomous cars - the time critical solutions in this context concern the interaction between programs executed on multicore processors and messages transmitted through wireless communication channels. Our contributions belong to the solutions space of all three classes of problems dealt within all three Kopernic objectives identified previously.

## 4.4 Drones

As it is the case of autonomous cars, there is an interaction between programs and messages, suggesting that our contributions in this context belong to the solutions space of all three classes of problems dealt within the objectives identified previously.

# 5 Social and environmental responsibility

## 5.1 Impact of research results

The Kopernic members provide theoretical means to improve the processor utilization. Such gain is estimated within 30% to 60% utilization gain for existing architectures or energy consumption for new architectures by decreasing the number of necessary cores.

# 6 Highlights of the year

## 6.1 Awards

Liliana Cucu-Grosjean and Yasmina Abdeddaim have been awarded a Paris Region PhD 2021 with the support of StatInf. Liliana Cucu-Grosjean and Adriana Gogonel have been awarded a Trophée des technologies by Embedded France for the RocqStat software which is the industrial version of EVT\_Kopernic, an Kopernic technology transferred to StatInf under an exclusive licence.

# 7 New software and platforms

The Kopernic members made contributions to two software, SynDEx and EVT\_Kopernic, described below.

## 7.1 New software

### 7.1.1 SynDEx

**Keywords:** Distributed, Optimization, Real time, Embedded systems, Scheduling analyses

**Scientific Description:** SynDEx is a system level CAD software implementing the AAA methodology for rapid prototyping and for optimizing distributed real-time embedded applications. It is developed in OCAML.

Architectures are represented as graphical block diagrams composed of programmable (processors) and non-programmable (ASIC, FPGA) computing components, interconnected by communication media (shared memories, links and busses for message passing). In order to deal with heterogeneous architectures it may feature several components of the same kind but with different characteristics.

Two types of non-functional properties can be specified for each task of the algorithm graph. First, a period that does not depend on the hardware architecture. Second, real-time features that depend on the different types of hardware components, ranging amongst execution and data transfer time, memory, etc.. Requirements are generally constraints on deadline equal to period, latency between any pair of tasks in the algorithm graph, dependence between tasks, etc.

Exploration of alternative allocations of the algorithm onto the architecture may be performed manually and/or automatically. The latter is achieved by performing real-time multiprocessor schedulability analyses and optimization heuristics based on the minimization of temporal or resource criteria. For example while satisfying deadline and latency constraints they can minimize the total execution time (makespan) of the application onto the given architecture, as well as the amount of memory. The results of each exploration is visualized as timing diagrams simulating the distributed real-time implementation.

Finally, real-time distributed embedded code can be automatically generated for dedicated distributed real-time executives, possibly calling services of resident real-time operating systems such as Linux/RTAI or Osek for instance. These executives are deadlock-free, based on off-line scheduling policies. Dedicated executives induce minimal overhead, and are built from processor-dependent executive kernels. To this date, executive kernels are provided for: TMS320C40, PIC18F2680, i80386, MC68332, MPC555, i80C196 and Unix/Linux workstations. Executive kernels for other processors can be achieved at reasonable cost following these examples as patterns.

**Functional Description:** Software for optimising the implementation of embedded distributed real-time applications and generating efficient and correct by construction code

**News of the Year:** We improved the distribution and scheduling heuristics to take into account the needs of co-simulation.

**URL:** <http://www.syndex.org>

**Contact:** Yves Sorel

**Participant:** Yves Sorel

### 7.1.2 EVT Kopernic

**Keywords:** Embedded systems, Worst Case Execution Time, Real-time application, Statistics

**Scientific Description:** The EVT-Kopernic tool is an implementation of the Extreme Value Theory (EVT) for the problem of the statistical estimation of worst-case bounds for the execution time of a program on a processor. Our implementation uses the two versions of EVT - GEV and GPD - to propose two independent methods of estimation. Their results are compared and only results that are sufficiently close allow to validate an estimation. Our tool is proved predictable by its unique choice of block (GEV) and threshold (GPD) while proposing reproducible estimations.

**Functional Description:** EVT-Kopernic is tool proposing a statistical estimation for bounds on worst-case execution time of a program on a processor. The estimator takes into account dependences between execution times by learning from the history of execution, while dealing also with cases of small variability of the execution times.

**News of the Year:** Any statistical estimator should come with an representative measurement protocole based on the processus of composition, proved correct. We propose the first such principle of composition while using a Bayesian modeling taking into account iteratively different measurement models. The composition model has been described in a patent submitted this year with a scientific publication under preparation.

**URL:** <http://www.stainf.fr>

**Contact:** Adriana Gogonel

**Participants:** Adriana Gogonel, Liliana Cucu

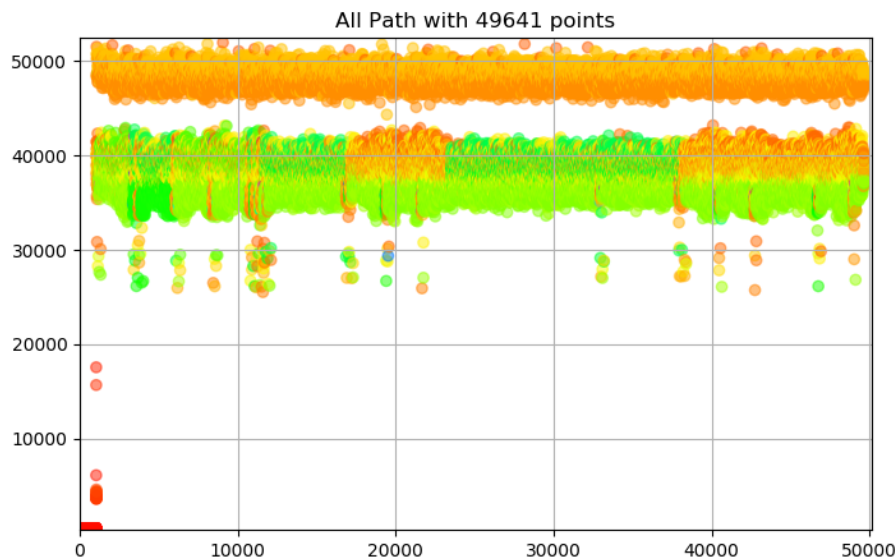


Figure 2: We indicate execution times obtained by measuring the same path by the same color

## 8 New results

We present below our new results covering our first two research axes and partially the third axis. Concerning the third axis, we have dedicated our effort in 2021 preparing the experimental platforms within the PX4-RT use case, as well as the STARTREC use case for the validation of results proposed in [9], on real applications. By real applications, we understand set of programs fulfilling functions for an embedded real-time system.

### 8.1 Worst case execution time estimation of a program

**Participants:** Avner Bar-Hen, Slim Ben Amor, Yasmina Abdeddaïm, Hadrien Clarke, Liliana Cucu-Grosjean, Adriana Gogonel, Kossivi Kougblenou, Yves Sorel, Kevin Zagalo, Marwan Wehaiba El Khazen.

We consider WCET statistical estimators that are based on the utilization of the Extreme Value Theory [23]. Compared to existing methods [36], our results require the execution of the program under study on the targeted processor or at least a cycle-accurate simulator of this processor. The originality of considering such WCET statistical estimators consists in the proposition of a black box solution with respect to the program structure. Such solution is obtained by (i) comparing different Generalized Extreme Values estimators [25] and (ii) separating the impact of the processor features from those of the program structure and of the execution environment, as variability factors for the CPS time properties.

Understanding the impact of the processor features and of the program structure is widely studied in the static analysis literature. Thanks to our new results [11, 8], our first objective has been refined by separating the impact of the program structure from the execution environment. Indeed, when the execution time of a program varies as a new path of that program is executed, then one may study the impact of the execution environment by comparing measurements from the same path, while varying input variables [12] or processor features states. For instance, in Figure 2 we notice that the green colour appears within the different clouds of execution time indicating that another factor than the path variation plays a role within the variation of this execution time.

Thus, we consider now the problem of identifying the impact of the execution environment on

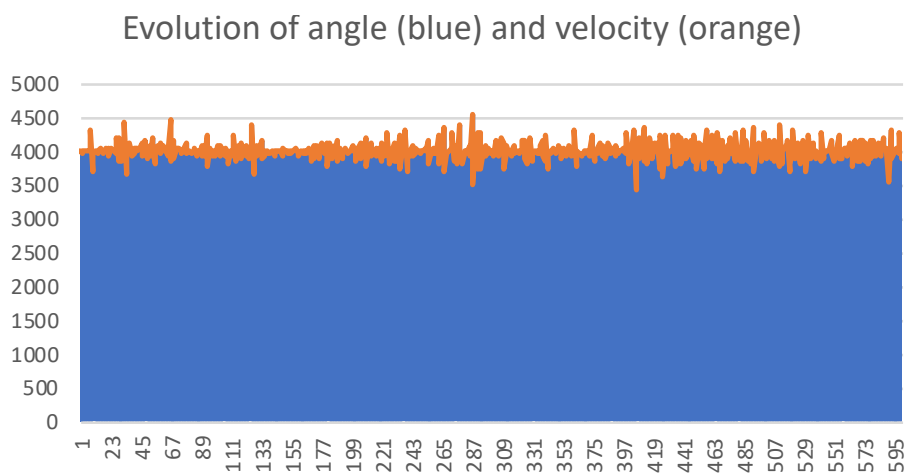


Figure 3: The evolution of inputs collected at  $T = 4ms$  for *Sensors*

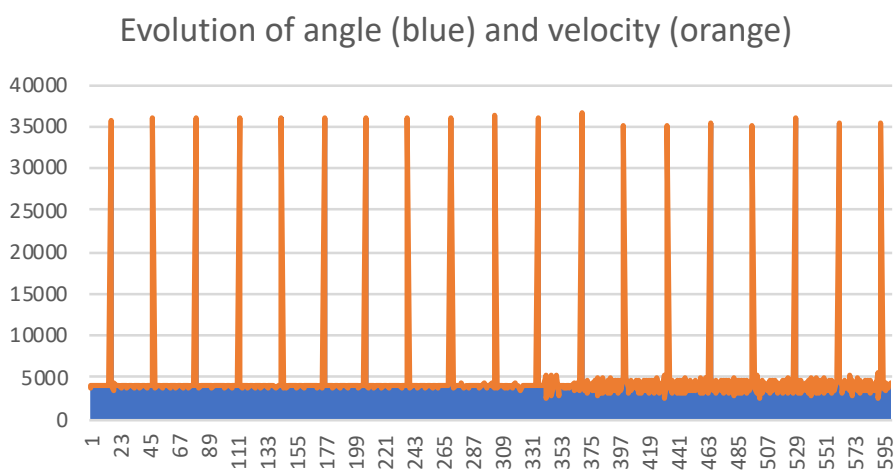


Figure 4: The evolution of inputs collected at  $T = 5ms$  for *Sensors*

the execution of a set of programs. A first step towards solving this problem is the identification of a possible relation between the period of updating the information from sensors through input variables of programs and the execution time of those programs [10]. For instance, in Figures 3 and 4 we may notice the important difference between two input variables for two flights of the autopilot of a drone<sup>3</sup>, while the trajectory and external conditions of these two flights are identical. More precisely, in Figures 3 and 4 the vertical axis describes the values for two sensors (*angle* in blue and *velocity* in orange), while the horizontal axis describes the order of appearance of these values during two flights with trajectories and external conditions that are as close as possible.

One difficult problem when advancing within an analysis of the relation between the input variables and sets of programs is **the identification of an appropriate study interval**. During this year, we have provided **an answer to this very difficult problem** by proving that for real-time systems with a mean utilization lower than 1, there always exists a finite instant from which a single-core system becomes schedulable under a fixed priority policy. We describe the system workload as a time-continuous random variable and provide an explicit formula for the time instant when the system enters a stable schedule in terms of the maximum deadline miss probability and the mean system utilization. We also provide the resulting distribution of response times. A paper detailing these results is currently under preparation for submission in IEEE Transactions on Computers.

<sup>3</sup>This autopilot has been proposed while fulfilling the objective [O2] of our research program

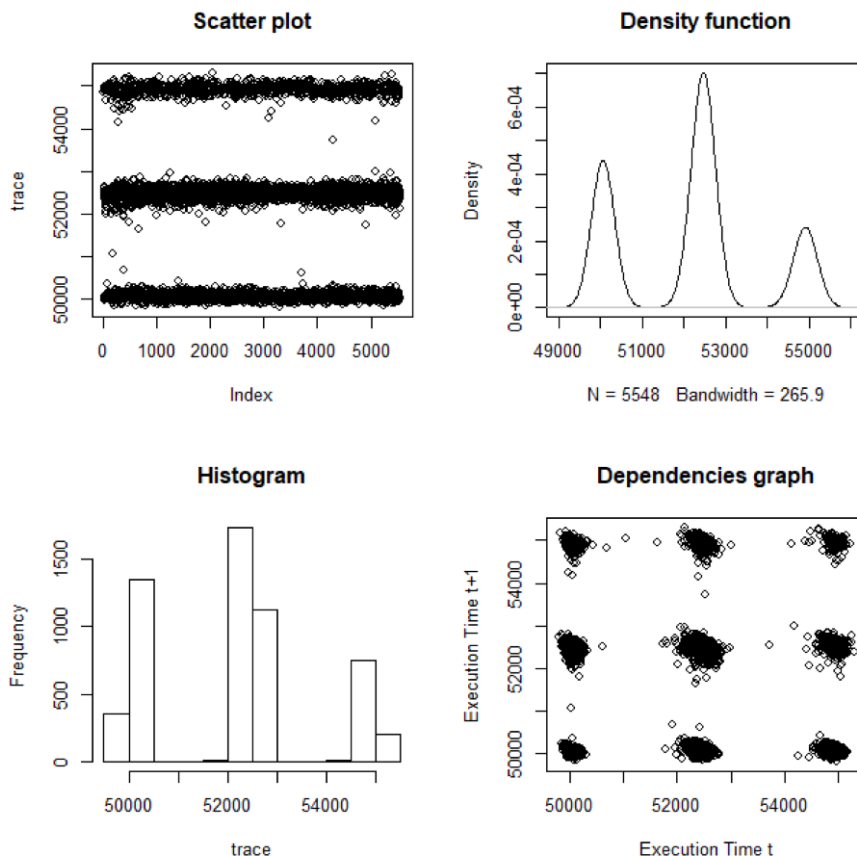


Figure 5: Different views of the execution times distribution of the Navigator program (PX4-RT)

## 8.2 Building measurement-based benchmarks

**Participants:** Avner Bar-Hen, Slim Ben Amor, Rihab Bennour, Hadrien Clarke, Liliana Cucu-Grosjean, Evariste Ntaryamira, Kevin Zagalo, Yves Sorel, Marwan Wehaiba El Khazen.

Our results are obtained while studying open-source programs of the autopilot **PX4** designed to control a wide range of air, land, sea and underwater drones, we identify the existence of execution modes, as well as strong dependencies with respect to the execution environment. Previously, we have underlined the existence of execution modes. In Figure 5 the graph on the left (top side) reveals three main execution modes, while the dependencies graph underlines the existence of correlations between the execution times at  $t$ , respectively, at  $t + 1$ .

More precisely, the studied programs are executed on a standard Pixhawk drone board based on a predictable single core processor ARM Cortex-M4 and during the CEOS project we have transformed this set of programs into a set of dependent tasks that satisfies real-time constraints, leading to a new version of PX4, called PX4-RT. As usual, the set of dependent real-time tasks is defined by a data dependency graph. An interested reader may refer to Figure 6 which details real-time tasks corresponding to the various functions of the autopilot: sensors processing, Kalman filter (EKF2) estimating position and attitude, position control, attitude control, navigator for long term navigation, commander for modes control, actuators (electric motors) processing. This PX4-RT version is available on the [Gitlab](#) website of our team.

During 2021, we have fulfilled several important steps towards the proposition of an open source

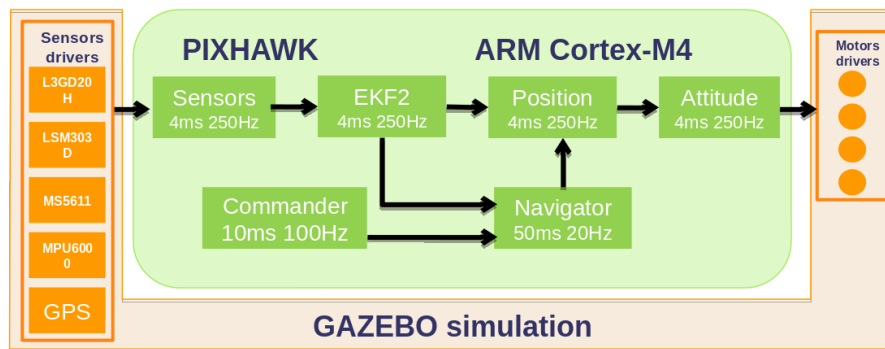


Figure 6: Dependent real-time tasks graph of the PX4-RT autopilot

measurement-based benchmarks set:

- the migration of the PX4-RT from the version 1.9 of PX4 to the PX4 version 1.12.
- for each hardware in the loop simulated flight, the data extraction of execution times from main programs (see the green boxes depicted in Figure 6) and other associated execution traces is now automatized while each recorded data is timestamped. Moreover, all scheduling points like release times, start times of the programs execution, preemption points are monitored and presented within a graphical interface. This graphical interface depicts the schedule of the PX4-RT programs (see Figure 7).
- a remote access to the hardware and software platform is now available to all Kopernic members as a first step preparing the reproducibility of the measurements made by different benchmarks users on the exact same hardware and software platform.
- the study of  $\mu$ ORB data communication framework on the real-time constraints [13].



Figure 7: A schedule of the PX4-RT programs

## 9 Bilateral contracts and grants with industry

### 9.1 CIFRE Grant funded by StatInf

**Participants:** Liliana Cucu-Grosjean, Adriana Gogonel, Marwan Wehaiba El Khazen.

A CIFRE agreement between the Kopernic team and the start-up StatInf has started on October 1st, 2020. Its funding is related to study of WCET models taking into account the energy consumption according to Kopernic research objectives.



## 10 Partnerships and cooperations

**Participants:** Kopernic members.

### 10.1 International initiatives

#### 10.1.1 Associate Teams in the framework of an Inria International Lab or in the framework of an Inria International Program

##### KEPLER

**Title:** Probabilistic foundations for time, a key concept for the certification of cyber-physical systems

**Duration:** 2020 ->

**Coordinator:** George Lima (gmlima@ufba.br)

##### Partners:

- Universidade Federal da Bahia

**Inria contact:** Liliana Cucu

**Summary:** The Kepler research concerns the proposition of new timing analyses and of new scheduling multicore algorithms.

### 10.2 International research visitors

Joël Goossens (Université Libre de Bruxelles) and Enrico Bini (University of Turin) have visited the Kopernic team in December 2021.

### 10.3 National initiatives

#### 10.3.1 FUI

**CEOS** The CEOS project has started on May 2017 and it has ended on February 2021. Partners of the project are: ADCIS, ALERION, Aeroport de Caen, EDF, ENEDIS, RTaW, EDF, Thales Communications and Security, ESIEE engineering school and Lorraine University. The CEOS project delivers a reliable and secure system of inspections of pieces of works using professional mini-drone for Operators of Vital Importance coupled with their Geographical Information System. These inspections are carried out automatically at a lower cost than current solutions employing helicopters or off-road vehicles. Several software applications proposed by the industrial partners, are developed and integrated in the drone, within an innovative mixed-criticality approach using multi-core platforms.

#### 10.3.2 PSPC

**STARTREC** The STARTREC project is funded by the PSPC call. Its partners are Easymile, StatInf, Trustinsoft, Inria and CEA. Its objective is the proposition of ISO26262 compliant arguments for the autonomous driving.

## 11 Dissemination

### 11.1 Promoting scientific activities

#### 11.1.1 Scientific events: selection

##### Chair of conference program committees

- Liliana Cucu-Grosjean has been the E2 topic chair at DATE2021.

### **Member of the conference program committees**

- The Kopernic members are regular PC members at events like RTSS, RTAS, RTNS, RTCSA, ICESSE, ETFA, VECOS, MSR, DASIP and WFCS.
- Avner Bar-Hen has been member of the scientific committee of LOD 2021 (the 7th Online and Onsite International Conference on Machine Learning, Optimization and Data Science).

#### **11.1.2 Journal**

##### **Guest editor**

- Liliana Cucu-Grosjean has been guest co-editor at Real-Time Systems Journal for a Special Issue dedicated to the best paper awards of IEEE RTSS2019 [7].

##### **Reviewer**

- All members of the team are regularly serving as reviewers for the main journals of our domain: Journal of Real-Time Systems, Information Processing Letter, Journal of Heuristics, Journal of Systems Architecture, Journal of Signal Processing Systems, Leibniz Transactions on Embedded Systems, IEEE Transactions on Industrial Informatics, etc.

#### **11.1.3 Leadership within the scientific community**

- Liliana Cucu-Grosjean is a IEEE TCRTS member (2016-2023).

#### **11.1.4 Scientific expertise**

- Yasmina Abdeddaïm is an elected member of the evaluation commission of ESIEE Paris.
- Yasmina Abdeddaïm has been member of the selection committee a the MCF contest (the 27 section) at Ecole Centrale de Nantes.
- Avner Bar-Hen is a member of the conseil scientifique du Haut Conseil aux Biotechnologies, of the scientific board of CESP, of the ANR committee on very high performance 2020, of the scientific board of EpiNano (Santé publique France).
- Liliana Cucu-Grosjean has been member of the selection committee a the MCF contest (the 27 section) at University of Lorraine.
- Yves Sorel is a member of the Steering Committee of System Design and Development Tools Group of Systematic Paris-Region Cluster.
- Yves Sorel is a member of the Steering Committee of Technologies and Tools Program of SystemX Institute for Technological Research (IRT).

#### **11.1.5 Research administration**

- Yasmina Abdeddaïm is in charge of the ESIEE Master on Artificial Intelligence and Cybersecurity.
- Liliana Cucu-Grosjean has been the co-chair of the Equal Opportunities and Gender Equality committee of Inria until October 2021.
- Liliana Cucu-Grosjean is an elected member of Inria scientific board(CS)and Inria CRCN disciplinary commision (until October 2021).
- Liliana Cucu-Grosjean is a nominated member of the CLHSCT Paris commission.
- Yves Sorel is the chair of the CUMI Paris center commission.
- Yves Sorel is member of the CDT Paris center commission.

## 11.2 Teaching - Supervision - Juries

### 11.2.1 Teaching

- Yasmina Abdeddaïm, Graphs and Algorithmes, the 2nd year of Engineering school, ESIEE Paris.
- Yasmina Abdeddaïm, Safe design of reactive systems, the 3rd year of Engineering school, ESIEE Paris.
- Yasmina Abdeddaïm, Embedded Real-Time Software, the 3rd year of Engineering school, ESIEE Paris.
- Avner Bar-Hen, Spatial statistics spatiales, ENSAI
- Amine Mohamed Khelassi, Introduction to algorithms, ESIEE Paris.
- Yves Sorel, Optimization of distributed real-time embedded systems, M2, University of Paris Sud, France.
- Yves Sorel, Safe design of reactive systems, M2, ESIEE Paris.
- Kevin Zagalo, Probabilities, ESIEE Paris.
- Kevin Zagalo, Machine learning, ESIEE Paris.
- Marwan Wehaiba El Khazen, Machine learning, ESIEE Paris.
- Marwan Wehaiba El Khazen, Statistics, ESIEE Paris.
- Marwan Wehaiba El Khazen, Optimization, ESIEE Paris.

### 11.2.2 Supervision

- Amine Mohamed Khelassi, Using statistical methods to model and estimate the time variability of programs executed on multicore architectures, Gustave Eiffel university, started on October 2021, supervised by Yasmina Abdeddaïm and Eva Dokladalova (ESIEE) and Liliana Cucu-Grosjean.
- Marwan Wehaiba El Khazen, Statistical models for optimizing the energy consumption of cyber-physical systems, Sorbonne university, started on October 2020, supervised by Liliana Cucu-Grosjean and Adriana Gogonel (StatInf).
- Kevin Zagalo, Statistical predictability of cyber-physical systems, Sorbonne university, started on January 2020, supervised by Liliana Cucu-Grosjean and Prof. Avner Bar-Hen (CNAM).
- Evariste Ntaryamira, A synchronous generalize method for preserving the quality of data for embedded real-time systems: the autopilot PX4-RT case, Sorbonne university, defended in May 2021, supervised by Liliana Cucu-Grosjean and Cristian Maxim (IRT System-X)

### 11.2.3 Juries

- Liliana Cucu-Grosjean has been member of the habilitation jury of Audrey Queudet, University of Nantes.
- Liliana Cucu-Grosjean has been a reviewer for the following PhD theses: Tristan Fautrel (LIGM), Louis Viard (University of Lorraine), Hai-Dang Vu (University of Nantes) and Matteo Bertolino (IPP Paris).
- Avner Bar-Hen has been a reviewer for the PhD thesis of Nathanaël Randriamihamison (University of Toulouse).

## 11.3 Popularization

### 11.3.1 Articles and contents

- Quentin Lazzarotto and Avner Bar-Hen, *Dingue de maths*, Hachette, 2021.
- Avner Bar-Hen, *Et si les mathématiques nous aidaient à gagner à l'Euromillions?*, The Conversation, 2021
- Liliana Cucu-Grosjean, *Un cadeau pour le 8 mars*, Le Binaire, 2021.
- Liliana Cucu-Grosjean, *Pourquoi accélérer le fonctionnement d'un ordinateur peut augmenter les risques d'erreur*, The Conversation, 2021

### 11.3.2 Interventions

- Avner Bar-Hen has been a speaker at Méthode Scientifique (France Culture), 2021
- Avner Bar-Hen has been a speaker at the event Science and You, 2021
- Liliana Cucu-Grosjean has been a speaker at the 5th edition of Sciences et média organized by BNE, 2021.
- Liliana Cucu-Grosjean has been a speaker at the Vivatech event Mega Women and Girls in Tech, 2021.
- Liliana Cucu-Grosjean has been a speaker at the event mediation event (Science and You), 2021

## 12 Scientific production

### 12.1 Major publications

- [1] L. Cucu-Grosjean and A. Gogonel. 'Simulation Device'. FR2016/050504 (France). Mar. 2016. URL: <https://hal.archives-ouvertes.fr/hal-01666599>.
- [2] L. Cucu-Grosjean, L. Santinelli, M. Houston, C. Lo, T. Vardanega, L. Kosmidis, J. Abella, E. Mezzetti, E. Quiñones and F. J. Cazorla. 'Measurement-Based Probabilistic Timing Analysis for Multi-path Programs'. In: *the 24th Euromicro Conference on Real-Time Systems, ECRTS*. 2012, pp. 91–101.
- [3] A. Gogonel and L. Cucu-Grosjean. 'Dispositif de caractérisation et/ou de modélisation de temps d'exécution pire-cas'. 1000408053 (France). June 2017. URL: <https://hal.archives-ouvertes.fr/hal-01666535>.
- [4] T. Kloda, A. Bertout and Y. Sorel. 'Latency analysis for data chains of real-time periodic tasks'. In: *the 23rd IEEE International Conference on Emerging Technologies and Factory Automation, ETFA'18*. Sept. 2018.
- [5] C. Maxim, A. Gogonel, I. M. Asavae, M. Asavae and L. Cucu-Grosjean. 'Reproducibility and representativity: mandatory properties for the compositionality of measurement-based WCET estimation approaches'. In: *SIGBED Review* 14.3 (2017), pp. 24–31.
- [6] S. E. Saidi, N. Pernet and Y. Sorel. 'Scheduling Real-time HiL Co-simulation of Cyber-Physical Systems on Multi-core Architectures'. In: *the 24th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*. Aug. 2018.

### 12.2 Publications of the year

#### International journals

- [7] L. Cucu-Grosjean and X. S. Hu. 'Guest editorial: Special issue on Real-Time Systems Symposium (RTSS)'. In: *Real-Time Systems* 57.3 (July 2021), pp. 265–267. DOI: [10.1007/s11241-021-09371-6](https://doi.org/10.1007/s11241-021-09371-6). URL: <https://hal.inria.fr/hal-03537530>.

### International peer-reviewed conferences

- [8] T. N. C. Andrade, G. Lima, V. M. C. Lima, Y. Abdeddaim and L. Cucu-Grosjean. ‘On the Selection of Relevant Hardware Events for Explaining Execution Time Behavior’. In: SBESC 2021 - XI Brazilian Symposium on Computing Systems Engineering. Florianopolis, Brazil: IEEE, 2021, pp. 1–8. DOI: [10.1109/SBESC53686.2021.9628374](https://doi.org/10.1109/SBESC53686.2021.9628374). URL: <https://hal.inria.fr/hal-03537584>.
- [9] S. Ben Amor and L. Cucu-Grosjean. ‘Graph reductions and partitioning heuristics for multicore DAG scheduling’. In: ICESS 2021 - 17th International Conference Embedded Software and Systems. Online, France, Dec. 2021. URL: <https://hal.inria.fr/hal-03538102>.
- [10] L. Cucu-Grosjean, A. Bar-Hen, Y. Sorel and H. Clarke. ‘Work-in-Progress Abstract: The impact of the period variation on execution time distributions of programs’. In: RTCSA 2021 - IEEE 27th International Conference on Embedded and Real-Time Computing Systems and Applications. Online, France: IEEE, 18th Aug. 2021, pp. 204–206. DOI: [10.1109/RTCSA52859.2021.00033](https://doi.org/10.1109/RTCSA52859.2021.00033). URL: <https://hal.inria.fr/hal-03537567>.
- [11] L. Cucu-Grosjean, M. W. E. Khazen, A. Gogonel, H. Clarke and Y. Sorel. ‘Work-in-Progress Abstract: WKS, a local unsupervised statistical algorithm for the detection of transitions in timing analysis’. In: RTCSA 2021 - IEEE 27th International Conference on Embedded and Real-Time Computing Systems and Applications. Online, France: IEEE, 18th Aug. 2021, pp. 201–203. DOI: [10.1109/RTCSA52859.2021.00032](https://doi.org/10.1109/RTCSA52859.2021.00032). URL: <https://hal.inria.fr/hal-03537554>.
- [12] A. Pavao, I. Guyon, N. Stéphane, F. Lebeau, M. Ghienne, L. Platon, T. Barbagelata, P. Escamilla, S. Mzali, M. Liao, S. Lassonde, A. Braun, S. B. Amor, L. Cucu-Grosjean, M. Wehaiba, A. Bar-Hen, A. Gogonel, A. B. Cheikh, M. Duda, J. Laugel, M. Marauri, M. Souissi, T. Lecerf, M. Elion, S. Tabti, J. Budynek, P. Le Bouteiller, A. Penon, R.-D. Lasserri, J. Ripoché and T. Epalle. ‘Aircraft Numerical "Twin": A Time Series Regression Competition’. In: ICMLA 2021 - 20th IEEE International Conference on Machine Learning and Applications. Pasadena / Virtual, United States, 13th Dec. 2021. URL: <https://hal.inria.fr/hal-03463307>.

### Doctoral dissertations and habilitation theses

- [13] E. Ntaryamira. ‘A synchronous generalize method for preserving the quality of data for embedded real-time systems: the autopilot PX4-RT case’. Sorbonne Universites, UPMC University of Paris 6, 7th May 2021. URL: <https://hal.inria.fr/tel-03544703>.

## 12.3 Cited publications

- [14] R. Baheti and H. Gill. *Cyber-physical systems*. IEEE, 2011.
- [15] S. Baruah, V. Bonifaci, A. Marchetti-Spaccamela, L. Stougie and A. Wiese. ‘A Generalized Parallel Task Model for Recurrent Real-time Processes’. In: *2012 IEEE 33rd Real-Time Systems Symposium (RTSS)*. 2012, pp. 63–72.
- [16] G. Bernat, A. Colin and S. M. Petters. ‘WCET Analysis of Probabilistic Hard Real-Time System’. In: *Proceedings of the 23rd IEEE Real-Time Systems Symposium (RTSS'02)*. IEEE Computer Society, 2002, pp. 279–288.
- [17] T. Bourke, J.-L. Colaço, B. Pagano, C. Pasteur and M. Pouzet. ‘A Synchronous-Based Code Generator for Explicit Hybrid Systems Languages’. In: *Compiler Construction - 24th International Conference, CC, Joint with ETAPS*. 2015, pp. 69–88.
- [18] L. Cucu. ‘Preliminary results for introducing dependent random variables in stochastic feasibility analysis on CAN’. In: *the WIP session of the 7th IEEE International Workshop on Factory Communication Systems (WFCS)*. 2008.
- [19] R. I. Davis and L. Cucu-Grosjean. ‘A Survey of Probabilistic Timing Analysis Techniques for Real-Time Systems’. In: *LITES 6.1* (2019), 03:1–03:60.
- [20] S. Edgar and A. Burns. ‘Statistical Analysis of WCET for Scheduling’. In: *the 22nd IEEE Real-Time Systems Symposium (RTSS)*. 2001, pp. 215–225.

- [21] H. Falk, S. Altmeyer, P. Hellinckx, B. Lisper, W. Puffitsch, C. Rochange, M. Schoeberl, R. B. Sorensen, P. Wägemann and S. Wegener. 'TACLeBench: A Benchmark Collection to Support Worst-Case Execution Time Research'. In: *16th International Workshop on Worst-Case Execution Time Analysis (WCET)*. Vol. 55. OASICS. 2016, 2:1–2:10.
- [22] J. Fonseca, G. Nelissen, V. Nelis and L. Pinho. 'Response time analysis of sporadic DAG tasks under partitioned scheduling'. In: *11th IEEE Symposium on Industrial Embedded Systems (SIES)*. May 2016, pp. 1–10.
- [23] S. J. Gil, I. Bate, G. Lima, L. Santinelli, A. Gogonel and L. Cucu-Grosjean. 'Open Challenges for Probabilistic Measurement-Based Worst-Case Execution Time'. In: *Embedded Systems Letters* 9.3 (2017), pp. 69–72.
- [24] J. Gustafsson, A. Betts, A. Ermedahl and B. Lisper. 'The Mälardalen WCET Benchmarks: Past, Present And Future'. In: *10th International Workshop on Worst-Case Execution Time Analysis (WCET)*. Vol. 15. OASICS. 2010, pp. 136–146.
- [25] M. W. el Khazen, A. Gogonel and L. Cucu-Grosjean. 'Work-in-Progress: Lessons learnt from creating an Extreme Value Library in Python'. In: *41st IEEE Real-Time Systems Symposium, RTSS*. IEEE, 2020, pp. 415–418.
- [26] E. Lee. 'Computing Needs Time'. In: *Communications of ACM* 52.5 (2009).
- [27] E. Lee and S. Seshia. *Introduction to embedded systems - a cyber-physical systems approach*. MIT Press, 2017.
- [28] J. Lehoczky. 'Real-Time Queueing Theory'. In: *the 10th IEEE Real-Time Systems Symposium (RTSS)*. 1996.
- [29] S. B. M. Bertogna and G. Buttazzo. *Multiprocessor Scheduling for Real-Time Systems*. Springer, 2015.
- [30] D. Maxim, O. Buffet, L. Santinelli, L. Cucu-Grosjean and R. I. Davis. 'Optimal Priority Assignment Algorithms for Probabilistic Real-Time Systems'. In: *the 19th International Conference on Real-Time and Network Systems (RTNS)*. 2011.
- [31] D. Maxim and L. Cucu-Grosjean. 'Response Time Analysis for Fixed-Priority Tasks with Multiple Probabilistic Parameters'. In: *the IEEE Real-Time Systems Symposium (RTSS)*. 2013.
- [32] F. Ndoye and Y. Sorel. 'Monoprocessor Real-Time Scheduling of Data Dependent Tasks with Exact Preemption Cost for Embedded Systems'. In: *the 16th IEEE International Conference on Computational Science and Engineering (CSE)*. 2013.
- [33] F. Nemer, H. Cassé, P. Sainrat, J. P. Bahsoun and M. D. Michiel. 'PapaBench: a Free Real-Time Benchmark'. In: *6th Intl. Workshop on Worst-Case Execution Time (WCET) Analysis*. Vol. 4. OASICS. 2006.
- [34] S. E. Saidi, N. Pernet and Y. Sorel. 'Automatic Parallelization of Multi-Rate FMI-based Co-Simulation On Multi-core'. In: *the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*. 2017.
- [35] T. Tia, Z. Deng, M. Shankar, M. Storch, J. Sun, L. Wu and J. Liu. 'Probabilistic Performance Guarantee for Real-Time Tasks with Varying Computation Times'. In: *IEEE Real-Time and Embedded Technology and Applications Symposium*. 1995.
- [36] R. Wilhelm, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, D. Whalley, G. Bernat, C. Ferdinand, R. Heckmann, T. Mitra, F. Mueller, I. Puaut, P. Puschner, G. Staschulat and P. Stenström. 'The worst-case execution time problem: overview of methods and survey of tools'. In: *Trans. on Embedded Computing Systems* 7.3 (2008), pp. 1–53.