# 2025 *Activity Report*

RESEARCH CENTRE: **Inria Centre at Rennes University**

## Project-Team

# HYCOMES

## Modélisation hybride & conception par contrats pour les systèmes embarqués multi-physiques

*In collaboration with* Institut de recherche en informatique et systèmes aléatoires (IRISA)

# Project-Team HYCOMES

*Creation of the Project-Team: 2016 September 01*

Each year, Inria research teams publish an Activity Report presenting their work and results over the reporting period. These reports follow a common structure, with some optional sections depending on the specific team. They typically begin by outlining the overall objectives and research programme, including the main research themes, goals, and methodological approaches. They also describe the application domains targeted by the team, highlighting the scientific or societal contexts in which their work is situated. The reports then present the highlights of the year, covering major scientific achievements, software developments, or teaching contributions. When relevant, they include sections on software, platforms, and open data, detailing the tools developed and how they are shared. A substantial part is dedicated to new results, where scientific contributions are described in detail, often with subsections specifying participants and associated keywords. Finally, the Activity Report addresses funding, contracts, partnerships, and collaborations at various levels, from industrial agreements to international cooperations. It also covers dissemination and teaching activities, such as participation in scientific events, outreach, and supervision. The document concludes with a presentation of scientific production, including major publications and those produced during the year.

# Keywords

## Computer sciences and digital sciences

A2.1. – Programming Languages

A2.1.1. – Semantics of programming languages

A2.1.9. – Synchronous languages

A2.1.10. – Domain-specific languages

A2.2. – Compilation

A2.2.1. – Static analysis

A2.2.8. – Code generation

A2.3. – Embedded and cyber-physical systems

A2.3.1. – Embedded systems

A2.3.2. – Cyber-physical systems

A2.3.3. – Real-time systems

A4.5. – Formal method for verification, reliability, certification

A4.5.1. – Static analysis

A4.5.3. – Program proof

A6. – Modeling, simulation and control

A6.1. – Methods in mathematical modeling

A6.1.1. – Continuous Modeling (PDE, ODE)

A6.1.5. – Multiphysics modeling

A6.4.3. – Observability and Controlability

A6.4.4. – Stability and Stabilization

A6.4.5. – Control of distributed parameter systems

A6.5. – Mathematical modeling for physical sciences

A7.2.1. – Decision procedures

A7.2.2. – Automated Theorem Proving

A7.2.4. – Mechanized Formalization of Mathematics

A8. – Mathematics of computing

A8.4. – Computer Algebra

## Other research topics and application domains

B4. – Energy

B4.4. – Energy delivery

B4.4.1. – Smart grids

B5. – Industry of the future

B5.1. – Factory of the future

B5.2. – Design and manufacturing

B5.9. – Industrial maintenance

B8. – Smart Cities and Territories

B8.1. – Smart building/home

B8.1.1. – Energy for smart buildings

B8.2. – Connected city

B8.3. – Urbanism and urban planning

# Contents

# 1 Team members, visitors, external collaborators

## Research Scientists

- Benoit Caillaud [Team leader, INRIA, Senior Researcher, HDR]
- Albert Benveniste [INRIA, Emeritus, HDR]
- Yahao Chen [INRIA, ISFP]
- Khalil Ghorbal [INRIA, Researcher]

## PhD Student

- Maxime Bridoux [INRIA, until Aug 2025]

## Technical Staff

- Mathias Malandain [INRIA, Engineer]

## Interns and Apprentices

- Damien Lejosne [ENS Rennes, Intern, from May 2025 until Jul 2025]

## Administrative Assistant

- Armelle Mozziconacci [CNRS]

# 2 Overall objectives

The Hycomes team is focused on two topics in cyber-physical systems design:

- Hybrid systems modeling, with an emphasis on the design of modeling languages in which software systems, in interaction with a complex physical environment, can be modelled, simulated and verified. A special attention is paid to the mathematical rigorous semantics of these languages, and to the correctness (wrt. such semantics) of the simulations and of the static analyses that must be performed during compilation. The Modelica language is the main application field. The team aims at contributing language extensions facilitating the modeling of physical domains which are poorly supported by the Modelica language. The Hycomes team is also designing new structural analysis methods for hybrid (aka. multi-mode) Modelica models. New simulation and verification techniques for large Modelica models are also in the scope of the team.
- Contract-based design and interface theories, with applications to requirements engineering in the context of safety-critical systems design. The objective of our research is to bridge the gap between system-level requirements, often expressed in natural, constrained or semi-formal languages and formal models, that can be simulated and verified.

# 3 Research program

## 3.1 Hybrid Systems Modeling

Systems industries today make extensive use of mathematical modeling tools to design computer controlled physical systems. This class of tools addresses the modeling of physical systems with models that are simpler than usual scientific computing problems by using only Ordinary Differential Equations (ODE) and Difference Equations but not Partial Differential Equations (PDE). This family of tools first emerged in the

1980's with SystemBuild by MatrixX (now distributed by National Instruments) followed soon by Simulink by Mathworks, with an impressive subsequent development.

In the early 90's control scientists from the University of Lund (Sweden) realized that the above approach did not support component based modeling of physical systems with reuse [1]. For instance, it was not easy to draw an electrical or hydraulic circuit by assembling component models of the various devices. The development of the Omola language by Hilding Elmqvist was a first attempt to bridge this gap by supporting some form of Differential Algebraic Equations (DAE) in the models. Modelica quickly emerged from this first attempt and became in the 2000's a major international concerted effort with the Modelica Consortium. A wider set of tools, both industrial and academic, now exists in this segment [2]. In the Electronic Design Automation (EDA) sector, VHDL-AMS was developed as a standard [57] and also enables the use of differential algebraic equations. Several domain-specific languages and tools for mechanical systems or electronic circuits also support some restricted classes of differential algebraic equations. Spice is the historic and most striking instance of these domain-specific languages/tools [3]. The main difference is that equations are hidden and the fixed structure of the differential algebraic results from the physical domain covered by these languages.

Despite the fact that these tools are now widely used by a number of engineers, they raise a number of technical difficulties. The meaning of some programs, their mathematical semantics, is indeed ambiguous. A main source of difficulty is the correct simulation of continuous-time dynamics, interacting with discrete-time dynamics: How the propagation of mode switchings should be handled? How to avoid artifacts due to the use of a global ODE solver causing unwanted coupling between seemingly non interacting subsystems? Also, the mixed use of an equational style for the continuous dynamics with an imperative style for the mode changes and resets, is a source of difficulty when handling parallel composition. It is therefore not uncommon that tools return complex warnings for programs with many different suggested hints for fixing them. Yet, these "pathological" programs can still be executed, if wanted so, giving surprising results — See for instance the Simulink examples in [26], [20] and [21].

Indeed this area suffers from the same difficulties that led to the development of the theory of synchronous languages as an effort to fix obscure compilation schemes for discrete time equation based languages in the 1980's. Our vision is that hybrid systems modeling tools deserve similar efforts in theory as synchronous languages did for the programming of embedded systems.

## 3.2 Background on non-standard analysis

Non-Standard analysis plays a central role in our research on hybrid systems modeling [20, 26, 22, 21, 24, 3]. The following text provides a brief summary of this theory and gives some hints on its usefulness in the context of hybrid systems modeling. This presentation is based on our paper [2], a chapter of Simon Bliudze's PhD thesis [32], and a recent presentation of non-standard analysis, not axiomatic in style, due to the mathematician Lindström [64].

Non-standard numbers allowed us to reconsider the semantics of hybrid systems and propose a radical alternative to the *super-dense time semantics* developed by Edward Lee and his team as part of the Ptolemy II project, where cascades of successive instants can occur in zero time by using $\mathbb{R}_+ \times \mathbb{N}$ as a time index. In the non-standard semantics, the time index is defined as a set $\mathbb{T} = \{n\partial \mid n \in {}^*\mathbb{N}\}$, where $\partial$ is an *infinitesimal* and ${}^*\mathbb{N}$ is the set of *non-standard integers*. Remark that (1) $\mathbb{T}$ is dense in $\mathbb{R}_+$, making it "continuous", and (2) every $t \in \mathbb{T}$ has a predecessor in $\mathbb{T}$ and a successor in $\mathbb{T}$, making it "discrete". Although it is not effective from a computability point of view, the *non-standard semantics* provides a framework that is familiar to the computer scientist and at the same time efficient as a symbolic abstraction. This makes it an excellent candidate for the development of provably correct compilation schemes and type systems for hybrid systems modeling languages.

Non-standard analysis was proposed by Abraham Robinson in the 1960s to allow the explicit manipulation of "infinitesimals" in analysis [73, 48, 44]. Robinson's approach is axiomatic; he proposes adding three new axioms to the basic Zermelo-Fraenkel (ZFC) framework. While the need for non-standard analysis (in addition to the usual or standard analysis) has long agitated the mathematical community, it is not our

---

[1] Origins of Equation-Based Modeling

[2] SimScape by Mathworks, Amesim by LMS International, now Siemens PLM, and more.

[3] Such as the Spice3 electronic circuit simulator.

purpose to debate such aspects. The important thing for us is that non-standard analysis allows the use of the non-standard discretization of continuous dynamics “as if” it was operational.

Not surprisingly, such an idea is not novel. Iwasaki et al. [59] first proposed using non-standard analysis to discuss the nature of time in hybrid systems. Bliudze and Krob [33, 32] have also used non-standard analysis as a mathematical support for defining a system theory for hybrid systems. They discuss in detail the notion of “system” and investigate computability issues. The formalization they propose closely follows that of Turing machines, with a memory tape and a control mechanism.

## 3.3 Structural Analysis of DAE Systems

The Modelica language is based on Differential Algebraic Equations (DAE). The general form of a DAE is given by:

$$F(t,x,x',x'',\dots) \tag{1}$$

where $F$ is a system of $n_e$ equations $\{f_1,\dots,f_{n_e}\}$ and $x$ is a finite list of $n_v$ independent real-valued, smooth enough, functions $\{x_1,\dots,x_{n_v}\}$ of the independent variable $t$. We use $x'$ as a shorthand for the list of first-order time derivatives of $x_j$, $j=1,\dots,n_v$. High-order derivatives are recursively defined as usual, and $x^{(k)}$ denotes the list formed by the $k$-th derivatives of the functions $x_j$. Each $f_i$ depends on the scalar $t$ and some of the functions $x_j$ as well as a finite number of their derivatives.

Let $\sigma_{i,j}$ denote the highest differentiation order of variable $x_j$ effectively appearing in equation $f_i$, or $-\infty$ if $x_j$ does not appear in $f_i$. The *leading variables* of $F$ are the variables in the set

$$\left\{x_j^{(\sigma_j)} \mid \sigma_j = \max_i \sigma_{i,j}\right\}$$

The *state variables* of $F$ are the variables in the set

$$\left\{x_j^{(\nu_j)} \mid 0 \le \nu_j < \max_i \sigma_{i,j}\right\}$$

A leading variable $x_j^{(\sigma_j)}$ is said to be *algebraic* if $\sigma_j = 0$ (in which case, neither $x_j$ nor any of its derivatives are state variables). In the sequel, $v$ and $u$ denote the leading and state variables of $F$, respectively.

DAE are a strict generalization of *ordinary differential equations* (*ODE*), in the sense that it may not be immediate to rewrite a DAE as an explicit ODE of the form $v = G(u)$. The reason is that this transformation relies on the Implicit Function Theorem, requiring that the Jacobian matrix $\frac{\partial F}{\partial v}$ to be full rank. This is, in general, not the case for a DAE. Simple examples, like the two-dimensional fixed-length pendulum in Cartesian coordinates [70], exhibit this behaviour.

For a square DAE of dimension $n$ (i.e., we now assume $n_e = n_v = n$) to be solved in the neighborhood of some $(v^*, u^*)$, one needs to find a set of non-negative integers $C = \{c_1,\dots,c_n\}$ such that system

$$F^{(C)} = \{f_1^{(c_1)},\dots,f_n^{(c_n)}\}$$

can locally be made explicit, i.e., the Jacobian matrix of $F^{(C)}$ with respect to its leading variables, evaluated at $(v^*, u^*)$, is nonsingular. The smallest possible value of $\max_i c_i$ for a set $C$ that satisfies this property is the *differentiation index* [38] of $F$, that is, the minimal number of time differentiations of all or part of the equations $f_i$ required to get an ODE.

In practice, the problem of automatically finding a minimal solution $C$ to this problem quickly becomes intractable. Moreover, the differentiation index may depend on the value of $(v^*, u^*)$. This is why, in lieu of numerical nonsingularity, one is interested in the *structural nonsingularity* of the Jacobian matrix, i.e., its almost certain nonsingularity when its nonzero entries vary over some neighborhood. In this framework, the *structural analysis* (*SA*) of a DAE returns, when successful, values of the $c_i$ that are independent from a given value of $(v^*, u^*)$.

A renowned method for the SA of DAE is the *Pantelides method*; however, Pryce's $\Sigma$-*method* is introduced also in what follows, as it is a crucial tool for our works.

### 3.3.1 Pantelides method

In 1988, Pantelides proposed what is probably the most well-known SA method for DAE [70]. The main idea of his work is that the structural representation of a DAE can be condensed into a bipartite graph whose left nodes (resp. right nodes) represent the equations (resp. the variables), and in which an edge exists if and only if the variable occurs in the equation.

By detecting specific subsets of the nodes, called *Minimally Structurally Singular* (*MSS*) subsets, the Pantelides method iteratively differentiates part of the equations until a perfect matching between the equations and the leading variables is found. One can easily prove that this is a necessary and sufficient condition for the structural nonsingularity of the system.

The main reason why the Pantelides method is not used in our work is that it cannot efficiently be adapted to multimode DAE (mDAE). As a matter of fact, the adjacency graph of a mDAE has both its nodes and edges parametrized by the subset of modes in which they are active; this, in turn, requires that a parametrized Pantelides method must branch every time no mode-independent MSS is found, ultimately resulting, in the worst case, in the enumeration of modes.

### 3.3.2 Pryce's Sigma-method

Albeit less renowned that the Pantelides method, Pryce's $\Sigma$-method [71] is an efficient SA method for DAE, whose equivalence to the Pantelides method has already been established. This method consists in solving two successive problems, denoted by primal and dual, relying on the $\Sigma$-*matrix*, or *signature matrix*, of the DAE $F$.

This matrix is given by:

$$\Sigma = (\sigma_{ij})_{1\leq i,j\leq n} \tag{2}$$

where $\sigma_{ij}$ is equal to the greatest integer $k$ such that $x_j^{(k)}$ appears in $f_i$, or $-\infty$ if variable $x_j$ does not appear in $f_i$. It is the adjacency matrix of a weighted bipartite graph, with structure similar to the graph considered in the Pantelides method, but whose edges are weighted by the highest differentiation orders. The $-\infty$ entries denote non-existent edges.

The *primal problem* consists in finding a *maximum-weight perfect matching* (*MWPM*) in the weighted adjacency graph. This is actually an assignment problem for which several standard algorithms exist, such as the push-relabel algorithm [55] or the Edmonds-Karp algorithm [50] to only give a few. However, none of these algorithms are easily parametrizable, even for applications to mDAE systems with a fixed number of variables.

The *dual problem* consists in finding the component-wise minimal solution $(C, D)$ where $C = \{c_1, \ldots, c_n\}$ and $D = \{d_1, \ldots, d_n\}$) to a given linear programming problem, defined as the dual of the aforementioned assignment problem. This is performed by means of a *fixpoint iteration* (*FPI*) that makes use of the MWPM found as a solution to the primal problem, described by the set of tuples $\{(i, j_i)\}_{i\in\{1,\ldots,n\}}$:

1. Initialize $\{c_1, \ldots, c_n\}$ to the zero vector.
2. For every $j \in \{1, \ldots, n\}$,
$$d_j \leftarrow \max_i(\sigma_{ij} + c_i)$$
3. For every $i \in \{1, \ldots, n\}$,
$$c_i \leftarrow d_{j_i} - \sigma_{i,j_i}$$
4. Repeat Steps 2 and 3 until convergence is reached.

From the results proved by Pryce in [71], it is known that the above algorithm terminates if and only if it is provided a MWPM, and that the values it returns are independent of the choice of a MWPM whenever there exist several such matchings. In particular, a direct corollary is that the $\Sigma$-method succeeds as long as a perfect matching can be found between equations and variables.

Another important result is that, if the Pantelides method succeeds for a given DAE $F$, then the $\Sigma$-method also succeeds for $F$ and the values it returns for $C$ are exactly the differentiation indices for the equations that are returned by the Pantelides method. As for the values of the $d_j$, being given by $d_j = \max_i(\sigma_{ij} + c_i)$, they are the differentiation indices of the leading variables in $F^{(C)}$.

Working with this method is natural for our works, since the algorithm for solving the dual problem is easily parametrizable for dealing with multimode systems, as shown in our recent paper [37].

### 3.3.3 Block triangular decomposition

Once structural analysis has been performed, system $F^{(C)}$ can be regarded, for the needs of numerical solving, as an algebraic system with unknowns $x_j^{(d_j)}$, $j = 1 \dots n$. As such, (inter)dependencies between its equations must be taken into account in order to put it into block triangular form (BTF). Three steps are required:

1. the *dependency graph* of system $F^{(C)}$ is generated, by taking into account the perfect matching between equations $f_i^{(c_i)}$ and unknowns $x_j^{(d_j)}$;

2. the *strongly connected components* (*SCC*) in this graph are determined: these will be the *equation blocks* that have to be solved;

3. the *block dependency graph* is constructed as the condensation of the dependency graph, from the knowledge of the SCC; a BTF of system $F^{(C)}$ can be made explicit from this graph.

## 3.4 Contract-Based Design, Interfaces Theories, and Requirements Engineering

System companies such as automotive and aeronautic companies are facing significant difficulties due to the exponentially raising complexity of their products coupled with increasingly tight demands on functionality, correctness, and time-to-market. The cost of being late to market or of imperfections in the products is staggering as witnessed by the recent recalls and delivery delays that many major car and airplane manufacturers had to bear in the recent years. The root causes of these design problems are complex and relate to a number of issues ranging from design processes and relationships with different departments of the same company and with suppliers, to incomplete requirement specification and testing.

We believe the most promising means to address the challenges in systems engineering is to employ formal design methodologies that seamlessly and coherently combine the various viewpoints of the design space (behavior, time, energy, reliability, ...), that provide the appropriate abstractions to manage the inherent complexity, and that can provide correct-by-construction implementations. The following issues must be addressed when developing new approaches to the design of complex systems:

- The overall design flows for heterogeneous systems and the associated use of models across traditional boundaries are not well developed and understood. Relationships between different teams inside a same company, or between different stake-holders in the supplier chain, are not supported by precise mathematical specifications of the components each party is expected to deliver.

- System requirements capture and analysis is in large part a heuristic process, where informal text and natural language-based techniques in use today are facing significant challenges [60]. Formal requirements engineering is in its infancy: mathematical models, formal analysis techniques and links to system implementation must be developed.

- Dealing with variability, uncertainty, and life-cycle issues, such as extensibility of a product family, are not well-addressed using available systems engineering methodologies and tools.

The challenge is to address the entire process and not to consider only local solutions of methodology, tools, and models that ease part of the design.

*Contract-based design* has been proposed as a new approach to the system design problem that is rigorous and effective in dealing with the problems and challenges described before, and that, at the same time, does not require a radical change in the way industrial designers carry out their task as it cuts across design flows of different types. Indeed, contracts can be used almost everywhere and at nearly all stages of system design, from early requirements capture, to embedded computing infrastructure and detailed design involving circuits and other hardware. Intuitively, a contract captures two properties, respectively representing the assumptions on the environment and the guarantees of the system under these assumptions. Hence, a contract can be defined as a pair $C = (A, G)$ of assumptions and guarantees characterizing in a formal way 1) under which context the design is assumed to operate, and 2) what its obligations are. Assume/Guarantee reasoning has been known for a long time, and has been used mostly in software engineering [68]. However, contract-based design is not limited to types and values in a piece of software. It can also be used to capture its performances (time, memory consumption, energy) and reliability. This amounts to enrich a component's interface with, on one hand, formal specifications of the behavior of the environment in which the component may be instantiated and, on the other hand, of the expected behavior of the component itself. To leverage contract-based reasoning as a technique of choice for system engineers, we aim to develop:

- mathematical foundations of contracts, that enable the design of formal verification frameworks;
- System engineering methodologies and tools, that focus on requirements modeling, contract specification and verification, at multiple abstraction levels.

A detailed bibliography on contract and interface theories for embedded system design can be found in [5]. In a nutshell, contract and interface theories fall into two main categories:

**Assume/guarantee contracts.** By explicitly relying on the notions of assumptions and guarantees, A/G-contracts are intuitive. This makes them appealing for the engineer. In A/G-contracts, assumptions and guarantees are just properties regarding the behavior of a component and of its environment. The typical case is when these properties are formal languages or sets of traces. This includes the class of safety properties [61, 42, 67, 19, 43]. Contract theories were initially developed as specification formalisms able to refuse some inputs from the environment [49]. A/G-contracts were advocated in [23] and are is still a very active research topic, with several contributions dealing with the timed [30] and probabilistic [35, 36] viewpoints in system design, and even hybrid systems design [69].

**Automata theoretic interfaces.** Interfaces combine assumptions and guarantees in a single, automata theoretic specification. Most interface theories are based on Lynch's Input/Output Automata [66, 65]. Interface Automata [15, 14, 16, 40] focus primarily on parallel composition and compatibility: two interfaces are compatible if there exists at least one environment where they can work together. The idea is that the resulting composition exposes as an interface the needed information to ensure that incompatible pairs of states cannot be reached. This can be achieved by using the possibility, for an Interface Automaton, to refuse some inputs from the environment in a given state. This amounts to the implicit assumption that the environment will never produce any of the refused inputs, when the interface is in this state. Modal Interfaces [72] inherit from both Interface Automata and the originally unrelated notion of Modal Transition System [63, 18, 34, 62]. Modal Interfaces are strictly more expressive than Interface Automata by decoupling the I/O orientation of an event and its deontic modalities (mandatory, allowed or forbidden). Informally, a *must* transition is offered in every component that realizes the modal interface, while a *may* transition is optional. Research on interface theories is still very active. For instance, timed [17, 27, 29, 46, 45, 28], probabilistic [35, 47] and energy-aware [41] interface theories have been proposed recently.

Requirements Engineering is one of the major concerns in large systems industries today, particularly so in sectors where certification prevails [74]. Most requirements engineering tools offer a poor structuring of the requirements and cannot be considered as formal modeling frameworks today. They are nothing less, but nothing more than an informal structured documentation enriched with hyperlinks.

We see Contract-Based Design and Interfaces Theories as innovative tools in support of Requirements Engineering. The Software Engineering community has extensively covered several aspects of Requirements Engineering, in particular:

- the development and use of large and rich *ontologies*; and
- the use of Model Driven Engineering technology for the structural aspects of requirements and resulting hyperlinks (to tests, documentation, PLM, architecture, and so on).

Behavioral models and properties, however, are not properly encompassed by the above approaches. This is the cause of a remaining gap between this phase of systems design and later phases where formal model based methods involving behavior have become prevalent. We believe that our work on contract-based design and interface theories is best suited to bridge this gap.

## 3.5 Effective Differential Algebra

A *limit cycle* of a vector field is a generalization of a point attractor towards which the states represented by nearby points ultimately converge. At the end of the 19th century, Poincaré observed that dynamical systems may exhibit more sophisticated attractors like closed curves for planar vector fields. Such objects are very important to understand and analyze the behavior of parametrized vector fields. One typically wants to know their number (if finite) as well as their robustness under slight perturbations of the involved parameters thereby avoiding or characterizing potential bifurcations.

These questions appeared as part of the 16th problem of the famous list of mathematical problems posed by David Hilbert at the International Congress of Mathematicians in 1900. 125 years later, despite the very interesting related developments, the problem remains essentially unsolved. For planar vector fields described by polynomials of degree at most $n$, as of today, no uniform upper bound $H(n)$ is known for the number of limit cycles. Partial results showing that limit cycles are finite, have been oscillating between conjecture and theorem since the first attempt by Henri Dulac in 1923. A recent paper by Melvin Yeung (January 2025) shows that there is a gap in the proof of Ilyashenko (published in 1991 and supposedly fixing Dulac's proof) fueling the dramatic history of the problem.

If one restricts attention to algebraic limit cycles, we already know that, in the planar case, there exists an upper bound on their degree. However such upper bound is not constructive. Only its mere existence is stated. That is to say, the link with the degree of the polynomials describing the vector field remains unknown. For higher dimensions, the situation is worse. All we currently know is that such a bound doesn't exist in general, which is to say almost nothing (even when restricting to purely algebraic limit cycles).

For historical reasons, the angles of attacks and toolboxes so far used revolve around geometrical intuitions (singularity theory) and analysis tools (special analytic functions). The algorithmic treatment of such problems, reminiscent of constructive mathematics, hasn't been explored to its full extent yet. Thanks to the ongoing exploratory action (AEx) `Backbone` (October 2022–March 2026), our recent contributions (over the past two years) to the problem have shown their immense potential and attracted attention within the communities of differential and computer algebra. In the upcoming years, we plan to push forward our work using effective generalized concepts of polynomials towards shedding more light on counting algebraic limit cycles for polynomial vector fields. More specifically:

- Current generation methods are essentially enumerative: they exhaustively enumerate all algebraic invariant hypersurfaces up to a fixed degree. Such approach is limited to low degrees as its computational complexity is prohibitive. We believe we have means to reduce the search space drastically. This is the topic of a research internship that will start in February 2026.

- Singularity theory has been already used to locally analyze the behavior of solutions around particular points and eventually state the existence of invariant hypersurfaces. This approach is geometric in nature and it would be very relevant to build the appropriate bridge with the algebraic approach we've been working on so far.

The algebraic version of Hilbert's 16th problem will serve as a far reaching goal bringing together mathematicians, physicist and computer scientists in a multidisciplinary effort to tackle this long standing challenge. In all cases, we see the road itself as valuable as fruitful since we expect to develop the necessary theory, data structures, concepts and approaches for effective differential algebra that we hope to impact other disciplines well beyond ours. We have in mind at least two communities that may benefit from such developments: (1) controllability in control theory where one needs to synthesize a controller for a particular region to be either reachable or invariant, and (2) formal verification of dynamical and hybrid (i.e. combining discrete and continuous time) systems where one wants to prove the correctness of the whole system.

# 4 Application domains

The Hycomes team contributes to the design of mathematical modeling languages and tools, to be used for the design of cyberphysical systems. In a nutshell, two major applications can be clearly identified: (i) our work on the structural analysis of multimode DAE systems has a sizeable impact on the techniques to be used in Modelica tools; (ii) our work on the verification of dynamical systems has an impact on the design methodology for safety-critical cyberphysical systems. These two applications are detailed below.

## 4.1 Modelica

Mathematical modeling tools are a considerable business, with major actors such as MathWorks, with Matlab/Simulink, or Wolfram, with Mathematica. However, none of these prominent tools are suitable for the engineering of large systems. The Modelica language has been designed with this objective in mind, making the best of the advantages of DAEs to support a component-based approach. Several industries in the energy sector have adopted Modelica as their main systems engineering language.

Although multimode features have been introduced in version 3.3 of the language [51], proper tool support of multimode models is still lagging behind. The reason is not a lack of interest from tool vendors and academia, but rather that multimode DAE systems poses several fundamental difficulties, such as a proper definition of a concept of solutions for multimode DAEs, how to handle mode switchings that trigger a change of system structure, or how impulsive variables should be handled. Our work on multimode DAEs focuses on these crucial issues [3].

Thanks to our IsamDAE software [37, 4], a larger class of Modelica models are expected to be compiled and simulated correctly. This should enable industrial users to have cleaner and simpler multimode Modelica models, with dynamically changing structure of cyberphysical systems. On the longer term, our ambition is to provide efficient code-generation techniques for the Modelica language, supporting, in full generality, multimode DAE systems, with dynamically changing differentiation index, structure and dimension.

The Hycomes team also focuses on scalability problems related to the compilation and simulation of large Modelica models. Digital twins developed by industrial Modelica users in the energy sector tend to be extremely large models, with up to $10^6$ equations. State-of-the-art Modelica compilers can not handle such models and users are forced to partition their model into smaller parts and use complex co-simulation techniques to produce executable digital twins. This puts a heavy burden on digital twin developers, since both the partitioning and the implementation of cosimulation methods are manual, finely tailored to the model, and require a high degree of expertise.

The Hycomes team is working on a new generation of algorithms for the compilation of the Modelica language, that can scale up to large models. The key contributations are modular index-reduction [25, 10] and block-triangular equation sorting algorithms, that can be applied to incomplete (rectangular) DAE systems.

## 4.2 Dynamical Systems Verification

In addition to well-defined operational semantics for hybrid systems, one often needs to provide formal guarantees about the behavior of some critical components of the system, or at least its main underlying logic. To do so, we are actively developing new techniques to automatically verify whether a hybrid system complies with its specifications, and/or to infer automatically the envelope within which the system behaves safely. The approaches we developed have been already successfully used to formally verify the intricate logic of the ACAS X, a mid-air collision avoidance system that advises the pilot to go upward or downward to avoid a nearby airplane which requires mixing the continuous motion of the aircraft with the discrete decisions to resolve the potential conflict [7]. This challenging example is nothing but an instance of the kind of systems we are targeting: autonomous smart systems that are designed to perform sophisticated tasks with an internal tricky logic. What is even more interesting perhaps is that such techniques can be often "reverted" to actually synthesize missing components so that some property holds, effectively helping the design of such complex systems.

# 5 Social and environmental responsibility

## 5.1 Impact of research results

The expected impact of our research is to allow both better designs and more efficient exploitation of energy production units and distribution networks, enabling large-scale energy savings. At least, this is what we could observe in the context of the FUI ModeliScale collaborative project (2018–2021), focused on electric grids, urban heat networks, and building thermal modeling.

The rationale is as follows: system engineering models are meant to assess the correctness, safety and optimality of a system under design. However, system models are still useful after the system has been put in operation. This is especially true in the energy sector, where systems have an extremely long lifespan (for instance, more than 50 years for some nuclear power plants) and are upgraded periodically, to integrate new technologies. Exactly like in software engineering, where a software and its model co-evolve throughout the lifespan of the software, a co-evolution of the system and its physical models has to be maintained. This is required in order to maintain the safety of the system, but also its optimality.

Moreover, physical models can be instrumental to the optimal exploitation of a system. A typical example are model-predictive control (MPC) techniques, where the model is simulated, during the exploitation of the system, in order to predict system trajectories up to a bounded-time horizon. Optimal control inputs can then be computed by mathematical programming methods, possibly using multiple simulation results. This has been proved to be a practical solution [53], whenever classical optimal control methods are ineffective, for instance, when the system is non-linear or discontinuous. However, this requires the generation of high-performance simulation code, capable of simulating a system much faster than real-time.

The structural analysis techniques implemented in IsamDAE [37] generate a conditional block dependency graph, that can be used to generate high-performance simulation code : static code can be generated for each block of equations, and a scheduling of these blocks can be computed, at runtime, at each mode switching, thanks to an inexpensive topological sort algorithm. In contrast to other approaches (such as [52]), no structural analysis, block-triangular decompositions, or automatic differentiation has to be performed at runtime.

# 6 Highlights of the year

Dynamical systems can be used to model the time dependence of phenomena acting according to some law (such as physical laws). Such systems however do not generally have closed-form solutions (Liouville's Theorem). These systems can still be (at least partially) solved when they possess first integrals, which are functions that are constant on any solution of the system. Several important classes of first integrals can be constructed by combining sufficiently many polynomials, called Darboux polynomials. Showing the non-existence of such first integrals requires exhaustive enumeration of Darboux polynomials, which can only be done up to a certain bound of their degree. In his thesis, Maxime Bridoux (2022-2025), presented and implemented algorithms (7.1.6) that can generate proofs that a given system does not possess any Darboux polynomials. This technique provides for instance a new, entirely automated, proof that the Van der Pol oscillator does not have any Darboux polynomials. The approach is not limited by the dimension of the system. It was applied to show that the Shimizu-Morioka system, of dimension 3, does not have any Darboux polynomial for all valuation of its parameters, which answers an open conjecture.

# 7 Latest software developments, platforms, open data

## 7.1 Latest software developments

### 7.1.1 IsamDAE

**Name:** Implicit Structural Analysis of Multimode DAE systems

**Keywords:** Structural analysis, Differential algebraic equations, Multimode, Scheduling, Consistent initialization, Code generation

**Scientific Description:** Modeling languages and tools based on Differential Algebraic Equations (DAE) bring several specific issues that do not exist with modeling languages based on Ordinary Differential Equations. The main problem is the determination of the differentiation index and latent equations. Prior to generating simulation code and calling solvers, the compilation of a model requires a structural analysis step, which reduces the differentiation index to a level acceptable by numerical solvers.

The Modelica language, among others, allows hybrid models with multiple modes, mode-dependent dynamics and state-dependent mode switching. These Multimode DAE (mDAE) systems are much harder to deal with. The main difficulties are (i) the combinatorial explosion of the number of modes, and (ii) the correct handling of mode switchings.

The IsamDAE software aims at providing a compilation chain for mDAE-based modeling languages that make it possible to efficiently generate correct simulation code for multimode models. Novel structural analysis methods for mDAE systems were designed and implemented, based on an implicit representation of the varying structure of such systems. Several standard algorithms, such as J. Pryce's Sigma-method and the Dulmage-Mendelsohn decomposition, were adapted to the multimode case, using Binary Decision Diagrams (BDD) to represent the mode-dependent structure of an mDAE system.

IsamDAE determines, as a function of the mode, the set of latent equations, the leading variables and the state vector. This is then used to compute a conditional dependency graph (CDG) of the system, that can be used to generate simulation code with a mode-dependent scheduling of the blocks of equations. The software is also fit for generating simulation code for the hybrid dynamical system simulation tool Siconos, as well as handling the structural analysis of the multimode consistent initialization problem associated with an mDAE system.

**Functional Description:** IsamDAE (Implicit Structural Analysis of Multimode DAE systems) is a software library implementing new structural analysis methods for multimode DAE systems, based on an implicit representation of incidence graphs, matchings between equations and variables, and block decompositions. The input of the software is a variable dimension multimode DAE system consisting in a set of guarded equations and guarded variable declarations. It computes a mode-dependent structural index reduction of the multimode system and is able to produce a mode-dependent graph for the scheduling of blocks of equations in long modes, check the structural nonsingularity of the associated consistent initialization problem, or generate simulation code for the nonsmooth dynamical system simulation tool Siconos.

IsamDAE is coded in OCaml, and uses the following packages: GuaCaml by Joan Thibault, MLBDD by Arlen Cox, Menhir by François Pottier and Yann Régis-Gianas, Pprint by François Pottier, Snowflake by Joan Thibault, XML-Light by Nicolas Cannasse and Jacques Garrigue.

**Release Contributions:** New features:

* XML representations of the structure of a multimode DAE model are accepted as inputs by the IsamDAE tool, in order to enable weak coupling with tools based on existing DAE-based languages. IsamDAE distinguishes between MEL and XML inputs based on the extension of the input file (.mel versus .mdae.xml).

Bug fixes:

* A better handling of the model structure for consistent initialization prevents subtle bugs that were observed for a few models and initial events. Specific error messages are returned when initial equations involve variables that are not active in the corresponding modes.

Performance improvement:

* Better handling of sets of equations/variables labeled with propositional formulas, thanks to an adapted data structure.

Various:

* Verbosity option -v now takes as a parameter an integer ranging from 0 ("quiet") to 5 ("deep debug"). The detailed output of CoSTreD is only available in "deep debug" mode.

**URL:** `https://team.inria.fr/hycomes/software/isamdae/`

**Publications:** hal-03768331, hal-02572879, hal-03320499, hal-02476541

**Contact:** Benoit Caillaud

**Participants:** Benoit Caillaud, Mathias Malandain, 3 anonymous participants

### 7.1.2 modularSigma

**Name:** A modular Sigma-method for the structural analysis of large DAE systems

**Keywords:** Differential algebraic equations, Modularity

**Scientific Description:** A key feature of the Modelica language is its object-oriented nature: components are instances of classes and they can aggregate other components, so that extremely large models can be efficiently designed as "trees of components". However, the structural analysis of Modelica models, a necessary step for generating simulation code, often relies on the flattening of this hierarchical structure, which undermines the scalability of the language and results in widely-used Modelica tools not being able to compile and simulate such large models. This software implements a new algorithm for the modular structural analysis of Modelica models. An adaptation of Pryce's Sigma-method for non-square DAE systems, along with a carefully crafted notion of component interface, make it possible to fully exploit the object tree structure of a model. The structural analysis of a component class can be performed once and for all, only requiring the information provided by the interface of its child components. The resulting method alleviates the exponential computation costs that can be yielded by model flattening, hence, its scalability makes it ideally suited for the modeling and simulation of large cyber-physical systems.

Algorithms implemented in modularSigma are based on the Sigma-method, which reduces the DAE structural index-reduction problem to two complementary linear programs: the primal problem amounts to the computation of a maximal-weight perfect matching of the equation-variable incidence graph of the DAE, while the dual problem consists in the computation of the minimal solution of a difference bound matrix (DBM). Modularity is achieved thanks to a decomposition of both problems, using dynamic programming principles (akin to message passing techniques, that are often used in statistical estimation) and memoization of the intermediate results.

**Functional Description:** The software performs the index reduction and the bloc-triangular decomposition of large DAE systems, defined as the composition, hiding and renaming of incomplete (rectangular) DAE systems.

**Release Contributions:** This release implements the block triangular decomposition of the reduce-index system. The benchmarks have been enriched with a model of a district heating system. This model is parameterized and its size can be adjusted up to several millions of equations.

**News of the Year:** In 2025, the block triangular decomposition of the reduce-index system has been implemented and benchmarked. The benchmarks have been enriched with a model of a district heating system. This model is parameterized and can scale up to several millions of equations.

**Publications:** hal-05257001, hal-04295096

**Contact:** Benoit Caillaud

**Participant:** Benoit Caillaud

### 7.1.3 PythonFMUGenerator

**Keywords:** Cosimulation, FMI, Cyber-physical systems

**Scientific Description:** PythonFMUGenerator is a tool for the automatic encapsulation of Python code into C++-based standardized cosimulation units (FMUs). It only relies on a Python source file and a JSON description of the properties of the generated FMU. This makes it possible to integrate on-demand FMU generation to a system model assembly and simulation pipeline, contrary to existing tools that create templates to be populated by hand before compilation and FMU generation.

**Functional Description:** FMI is a fast-growing standard for the cosimulation of large mutli- and cyberphysical system models. It relies on the encapsulation of source code, written in various tools and languages, into cosimulation units called FMUs that share a common interface. However, the encapsulation of Python code into an FMU is still a technical challenge that very few tools try to address.

PythonFMUGenerator is a tool for the automatic encapsulation of Python code into C++-based FMUs for cosimulation. It only relies on a Python source file and a JSON description of the properties of the generated FMU. This makes it possible to integrate on-demand FMU generation to a system model assembly and simulation pipeline, contrary to existing tools that create templates to be populated by hand before compilation and FMU generation.

PythonFMUGenerator relies on the Spycic library (from the same author), that acts as a wrapper around the C/Python API so as to considerably simplify Python function calls from C or C++ code. It is based on FMICodeGenerator, a tool developed by Andreas Nicolai (ghorwin) and coworkers, itself under a BSD3 license.

**News of the Year:** Passage au standard FMI3 (API intégrale + logiques d'exécution pour l'initialisation, l'avancement temporel et l'entrée/sortie de valeurs de tous les types gérés par le standard), gestion des vecteurs en entrées/sorties des FMU, pipeline CI pour de nombreux tests de l'intégralité de la chaîne de génération des FMU.

**Contact:** Mathias Malandain

**Participants:** Benoit Caillaud, Mathias Malandain, Thibaud Toullier

### 7.1.4 Spycic

**Name:** Spycic library

**Keywords:** Python, C++, Binding

**Scientific Description:** Spycic is a header-only C++ library for fetching and calling Python functions from C++ code. Designed as a wrapper of the C/Python API, Spycic strongly relies on variadic templates to make it possible to call in a simple way Python functions with different signatures and an arbitrary number of arguments. GIL handling, exception handling and type casting are performed under the hood, so as to make use as simple as possible.

**Functional Description:** The Spycic (Simple Python Calls In C++) header-only library is a wrapper around the C/Python API that provides a handful of functions allowing for simple calls to Python functions from a C++ code. Python functions to be used may be declared in several *independent* source files.

Spycic provides the following functions:

* 'fetchFunction(const char* functionName, const char* sourceCode)' is used to fetch the Python function called 'functionName' from a given Python code 'sourceCode' provided as a C-style string. The function is returned as a 'PyObject*'. * 'fetchFunction(const char* functionName, std::string& sourceCode)' is used to fetch the Python function called 'functionName' from a given Python code 'sourceCode' provided as a 'std::string'. The function is returned as a 'PyObject*'. * 'runFunction<returnType>(PyObject* func, Values... values)' is used to call a Python function (imported as a 'PyObject*') with an arbitrary number of arguments. The return type has to be specified as a template argument, for example, 'runFunction<double>(f, arg1, arg2)'. The return type can be 'void', as in 'runFunction<void>(f, arg1, arg2)'. All necessary operations (GIL handling, formatting, type castings, etc.) are performed under the hood. The arguments are provided as C++ POD (Plain Old Data) values and/or 'std::vector' containers, and the output (if any) is a C++ POD or vector as well.

A fresh exception class called 'PythonError' is also defined in order to handle errors that occur during calls to functions provided by the C/Python API itself.

Client code must still call functions 'Py_Initialize()' and 'Py_Finalize()' to be able to use the Python interpreter.

**Release Contributions:** The handling of 'std::vector' objects (containing floating point numbers or integers) as both function inputs and function outputs, and the handling of functions returning 'void', were added since version 0.1.

**News of the Year:** Regroupement sous la forme d'une bibliothèque header-only, appels transparents de fonctions ne retournant aucune valeur, vérification des overflows lors du casting de valeurs de retour (Python vers C), mémoisation pour l'import des modules, pipeline CI, passage en REUSE-compliant en vue d'une ouverture open source.

**Contact:** Mathias Malandain

**Participant:** Mathias Malandain

### 7.1.5 mmDM

**Name:** Multi-Mode Dulmage-Mendelsohn

**Keywords:** Structural analysis, Differential algebraic equations, Multimode, Model-based diagnosis, Dulmage-Mendelsohn decomposition

**Scientific Description:** Model-based diagnosis for the health monitoring of single-mode, smooth physical systems is a well-established field, supported by a large body of literature covering various approaches. In particular, structural fault detectability and isolability is a graph-based method to evaluate diagnosability properties on DAEs (Differential Algebraic Equations). It is based on the Dulmage-Mendelsohn (DM) decomposition, a building block of the structural analysis of equation systems.

However, the modeling of non-smooth physical systems typically yields switched DAEs, also known as multimode DAEs (mDAE), which combine continuous behaviors, defined as solutions of a set of smooth DAE systems, with discrete mode changes. Direct application of traditional fault diagnosis methods to all possible configurations of mDAEs quickly becomes intractable, as the number of modes tends to be exponential in the size of the system.

mmDM implements a novel multimode DM decomposition algorithm, based on an implicit representation of the varying structure of multimode systems. Under the hood, Binary Decision Diagrams (BDD) are used to represent the mode-dependent structure of an mDAE system, and BDD computations make it possible to compute the DM decomposition of a multimode system in all modes "at once", instead of having to enumerate them.

mmDM outputs a compact description of the DM decomposition of the input mDAE, as well as the maximum equation-variable matchings that were used for computing this decomposition in each mode. It was designed as a foundation stone for structural diagnosis methods for mDAEs, and the implementation of new features for diagnosability and fault isolability analysis is planned in the near future.

**Functional Description:** mmDM (Multi-Mode Dulmage-Mendelsohn) is a small tool that implements the Dulmage-Mendelsohn decomposition for multimode DAE systems. It is based on an implicit representation of incidence graphs and matchings between equations and variables. The input of the software is a variable dimension multimode DAE system consisting in a set of guarded equations and guarded variable declarations. The output is a description of the Dulmage-Mendelsohn decomposition of the system in all its modes, in this output, three Boolean propositions are associated to each equation and variable, describing the set of modes in which this equation/variable is part of the underdetermined, square and overdetermined parts of the decomposition. This output also details the maximum matchings (one per mode) that were used to compute the decomposition by listing, for each equation, which variables are matched to it in which sets of modes.

mmDM is based on IsamDAE (Implicit Structural Analysis of Multimode DAE systems), a software library developed at the Inria Center at Rennes University that implements new structural analysis methods for multimode DAE systems (see https://team.inria.fr/hycomes/software/isamdae/ for more information).

mmDM is coded in OCaml, and uses the following packages: GuaCaml by Joan Thibault, MLBDD by Arlen Cox, Menhir by François Pottier and Yann Régis-Gianas, Pprint by François Pottier, Snowflake by Joan Thibault, XML-Light by Nicolas Cannasse and Jacques Garrigue.

**Release Contributions:** * New "atmostone" n-ary operator in the input language. * Enumerations can mix lists and iterators, such as in:

if exists { { i in 1..3 : p[i] }, b, c } then ... end,

**News of the Year:** A multimode extension of fault detectability and isolability decision procedures have been implemented in mmDM. This includes new data-structures, algorithmic building blocks and tests. A CD pipeline has been configured to automatically release the software as Opem bundles.

**Publication:** hal-04803147

**Contact:** Benoit Caillaud

**Participants:** Benoit Caillaud, Mathias Malandain

### 7.1.6 DarbouxCertification

**Keywords:** Dynamical system, Symbolic computation, Differential algebraic equations

**Functional Description:** The package provides necessary conditions for an ansatz to be a Darboux polynomials, that is to say an algebraic special invariant for a given system of polynomial ordinary differential equations. When it succeeds in proving their non-existence, the package issues a formal certificate (that can be check by an independent theorem prover). For dynamical systems with parameters, the package interacts with the user to get additional assumptions on the involved parameters. For instance, it might provide a non-existence proof only under certain extra assumptions on the parameters.

**URL:** `https://gitlab.com/maximebridoux/darbouxcertification`

**Publication:** hal-04818282

**Contact:** Khalil Ghorbal

**Participants:** Khalil Ghorbal, Maxime Bridoux

# 8 New results

## 8.1 Modular Structural Analysis of DAE Systems

**Participants:** Albert Benveniste, Benoît Caillaud, Mathias Malandain.

In [25], a new modular structural analysis algorithm has been proposed that takes full advantage of the object tree structure of a DAE model. The bedrock of this method is a novel concept of *structural analysis-aware interface* for components. The essence of a component interface is to capture the necessary information about a Modelica class that needs to be exposed, in order to perform the structural analysis of a component comprising instances of the former class, while hiding away useless information regarding the equations and all *protected* features it may contain.

In order to compute a component interface, one has to be able to perform the structural analysis of the possibly non-square DAE system that this component encapsulates, and to use the interfaces of the components it aggregates in this analysis. We base our algorithm on Pryce's $\Sigma$-method for index reduction [71], which essentially consists in the successive solving of two dual linear integer programs. The striking difference with Pryce's algorithm is that these problems are solved by parts, in a scalable manner.

Putting all of this together, it is then possible to perform a *modular structural analysis*, performed at the class level, and the results can then be instantiated for each component of the system model, knowing its context. Hence, structural information at the system level is derived from composing the result of component-level analysis. Modular structural analysis yields huge gains in terms of memory usage and computational costs, as the analysis of a single large-scale DAE is replaced with that of multiple smaller subsystems.

In 2025, the modular structural analysis algorihtm has been fully implemented (see Section 7.1.2) and extended to the computation of the block-triangular decomposition of the reduced-index system of equations [10]. This implementation has been tested against several scalability benchmarks from the literature [39]. A parameterized model of a district heating system has also been developed, with the objective of providing a more relevant test-case. The size of the model can be made arbitrarily large, and experiments confirm that the algorithm has an empirical complexity that is sublinear in the size of the model. This is made possible thanks to the sparsity and low treewidth of the model, as it is often the case for energy network infrastructures.

## 8.2 Fault Diagnosability Analysis of Multi-Mode Systems

**Participants:** Benoît Caillaud, Mathias Malandain.

This work has been conducted in collaboration with the University of Linköping (Sweden) on the topic of system diagnosis, based on multimode DAE systems.

Fault detection and diagnosis are important for the health monitoring of physical systems. Model-based approaches for single-mode, smooth, systems are a well-established field, supported by a large body of literature covering various approaches like structural methods [31], parity space techniques, and observer-based methods [58].

While single-mode systems are often described using differential algebraic equations (DAEs), the modeling of non-smooth physical systems yields switched DAEs, also known as multimode DAEs (mDAEs), which combine continuous behaviors, defined as solutions of a set of DAE systems, with discrete mode changes [76, 3]. Direct application of traditional fault diagnosis methods to all possible configurations of multi-mode systems quickly becomes intractable, as the number of modes tends to be exponential in the size of the system.

Structural fault detectability and isolability are a graph-based method to evaluate diagnosability properties on DAEs [54]. It is based on the Dulmage-Mendelsohn decomposition (DM), a building block of the structural analysis of equation systems. In [56], we show how its extension to multimode systems, introduced in [4], can be applied in the context of structural fault detectability and isolability of multimode systems.

In 2025, these algorithms have been fully implemented in the mmDM software 7.1.5 and benchmarked on a model of an automotive electric battery pack.

## 8.3 Automated Reasoning For The Existence Of Darboux Polynomials

**Participants:** Maxime Bridoux, Khalil Ghorbal.

*Darboux polynomials* are particular algebraic invariants that play an important role in the integrability theory of ordinary differential equations (ODEs). Computation of Darboux polynomials is a central problem in the Prelle-Singer procedure for computing elementary first integrals of planar systems of polynomial ODEs, which yields a systematic method for computing elementary closed-form solutions (whenever these exist) to an important class of ordinary differential equations. Owing to this important application, algorithms for generating Darboux polynomials have received considerable attention in computer algebra. More recently, Darboux polynomials have found application in the area of formal safety verification of cyber-physical systems, where the problem of their automatic generation is encountered in the broader context of searching for invariant (and positively invariant) sets.

Darboux generation algorithms are semi-decision procedures enumerating all Darboux polynomials up to a certain fixed bound on the total degree. The bound is eventually increased until finding a (not necessarily irreducible) Darboux polynomial or reaching memory and/or time limits. Theoretically, the existence of a bound on the total degree of irreducible Darboux polynomials is, as of today, an open problem when $n \geq 3$. Even when such theoretical bound exists, it is easily seen that it depends non trivially not only on the total degrees of the polynomials defining the ODE but also on their coefficients, making the task of estimating an upper bound even harder.

Given a polynomial ordinary differential equation (ODE), we devise *generic* polynomial reduction algorithms to automatically investigate the intertwined relationship between the total degree of (nontrivial) Darboux polynomials and the polynomials defining the ODE. By generic we mean that both the coefficients and the multidegree of the involved polynomials are symbolic. We use Newton polytopes as a light-weight abstraction to select optimal weight monomial orders improving the efficiency of the involved computations. The method works by inferring necessary conditions on both the coefficients and the multidegree for the polynomial to be Darboux [6]. These conditions are then used, via constants' propagation, to restrict the shape of the generic candidate, pinpointing which monomials ought to be preserved by removing the superfluous ones. In some relevant cases, we are able to automatically prove the nonexistence of (nontrivial) Darboux polynomials providing a new toolbox (7.1.6) to prove and formally certify that some limit cycles are not algebraic.

## 8.4 On Covering Euclidean Spaces with Q-arrangements of Cones

**Participants:** Khalil Ghorbal.

The Linear Complementarity Problem (LCP) provides a unifying framework for both linear and quadratic programming (respectively optimizing a linear and a quadratic function over a convex polyhedron) as well as the bimatrix game problem. Several algorithms to solve these largely used problems are fundamentally based on efficient LCP solvers. Despite important recent advances and improvements on solving instances of the LCP (for a fixed input vector), several fundamental questions remain open. The *solvability* of the LCP is one of these open problems. Formally, given a vector $q \in \mathbb{R}^n$ and an $n \times n$ matrix $M$ over the reals, the *linear complementarity problem*, $\mathrm{LCP}(q, M)$, asks whether there exists a pair $w, z \in \mathbb{R}^n$ satisfying $w - Mz = q$, $w, z \geq 0$, and $w.z = 0$, where $w, z \geq 0$ means that $w$ and $z$ belong to $\mathbb{R}^n_+$, the nonnegative orthant of $\mathbb{R}^n$, and $w.z$ is the scalar product of $w$ and $z$. When $\mathrm{LCP}(q, M)$ admits a solution, it is said to be *solvable*.

The class of Q-matrices is related to the solvability concept: when $\mathrm{LCP}(q, M)$ is solvable for all $q$, $M$ is called a *Q-matrix*. Characterizing Q-matrices (without extra degeneracy or structural assumptions on the matrix $M$) is an open problem since the sixties. It was only known for $n \leq 2$. This work [12] provides a full algebraic characterization for Q-matrices for $n = 3$ and shows that the problem reduces to checking sign conditions on the minors of the involved matrix for dimensions less than 4.

The originality of our approach comes from using a topological insight on a geometrical reformulation of the problem to locate the regions for which the LCP *does not have a solution*. This dual standpoint was instrumental to reduce the problem to a local similar problem around the original vectors. In addition, the (symbolic) computational perspective was novel. In fact, the characterization is a program enumerating all sign conditions on the minors of the matrix $M$ that are satisfied if and only if $M$ is a Q-matrix. We encountered two theoretical difficulties: (i) reformulating and formalizing our topological intuitions in an appropriate language for the optimization community (we opted for convex analysis), and (ii) to “tame” the underlying inherent combinatorial explosion to make it comprehensive and checkable by a (knowledgeable) human reader.

## 8.5 Torque observation of WRSM with model uncertainties for EV applications

**Participants:** Yahao Chen.

This work has been conducted in collaboration with Centrale Nantes and Renault on the topic of system diagnosis with applications on electric vehicles and electric motors.

We propose a torque observation method based on a linear parameter varying (LPV) approach for a wound rotor synchronous machine (WRSM) used in Electric Vehicles (EVs), specifically for the Renault ZOE. The novelty of our approach lies in its ability to handle a wide range of uncertainties and parameter variations, such as speed fluctuations and model uncertainties in both magnetic flux and resistance. This enables more accurate and robust torque estimation, which is crucial for the demanding performance requirements of EV applications. We present a comprehensive observation methodology, which includes a state and unknown input observability study, robust LPV observer design, and a convergence analysis. The effectiveness of this approach is demonstrated through both simulations and experimental tests conducted on the BEMEVE real-power test bench. To highlight its merits, the performance of the LPV observer is compared to different types of observers [9].

## 8.6 Strong structural controllability analysis of structured networks with identical nodes

**Participants:** Yahao Chen.

This research was carried out in collaboration with China University of Mining and Technology, focusing on structural analysis and networked systems.

In recent decades, the controllability of large-scale networks has become a major topic of interest within the systems and control community. Analyzing network controllability is often challenging due to incomplete knowledge about both the interconnection structure of the network and the dynamic behavior of its individual nodes. To address these limitations, many researchers employ structural analysis techniques, which provide a framework for modeling such partial information and thereby facilitate controllability studies. Specifically, by representing the network information using zero/nonzero/arbitrary pattern matrices, the controllability problem for a single network is generalized to a family of networks, referred to as structured networks.

In this study, we establish necessary and sufficient rank conditions to assess the strong structural controllability of structured networks composed of identical nodes. For networks with multi-input multi-output (MIMO) nodes, we introduce an assumption termed strong structural input-state observability, under which the strong structural controllability of the network depends solely on its topology. In the single-input single-output (SISO) case, this assumption can be relaxed by incorporating specific controllability and observability conditions at the nodal level. Finally, the proposed rank conditions are validated using a recently developed graph-theoretical approach for structured systems [13].

# 9 Partnerships and cooperations

## 9.1 International research visitors

### 9.1.1 Visits of international scientists

**Other international visits to the team**

**Stephan Trenn**

**Status:** Associate Professor

**Institution of origin:** University of Groningen

**Country:** the Netherlands

**Dates:** 27/08/2025-27/08/2025

**Context of the visit:** Prof. Trenn is a leading expert in the field of switched DAEs. This visit follows up on his visit to HYCOMES in 2024. The discussions focus on the initialization of the ANR JCJC project (PI: Yahao Chen) of the team and his recent results on solutions of switched DAEs in discrete time.

**Mobility program/type of mobility:** Research Stay

**Bart Bessilink**

**Status:** Associate Professor

**Institution of origin:** University of Groningen

**Country:** the Netherlands

**Dates:** 6/10/2025-11/10/2025

**Context of the visit:** Prof. Besselink is a renowned expert in control analysis and large-scale system design and was recently awarded an ERC Consolidator Grant for his work on Contracts for Control System Design. The HYCOMES team has long focused on contract-based design, with particular emphasis on the use of contract algebras to model system behaviors and interactions. In contrast, Prof. Besselink's work centers on contract-based control of large-scale systems.Our teams share a common interest in understanding how contracts can govern the behavior of complex systems, albeit with different emphases: HYCOMES focuses on the algebraic structures underlying contracts, while Prof. Besselink investigates their application in control systems. During his visit, Prof. Besselink delivered a 68NQRT seminar presenting his recent work on geometric control and behavioral theory for interpreting contracts in control systems. Several potential directions for collaboration were explored, including extending the notion of hypercontracts and contract algebras to control systems; defining contracts for linear systems with inequality constraints as a generalization of existing frameworks; and using differential-algebraic equations (DAEs) to represent contracts.

**Mobility program/type of mobility:** Research Stay

## 9.2 National initiatives

### Generalized Filippov solutions for discontinuous DAEs: Control and Simulations

**Participants:** Yahao Chen(PI), Benoît Caillaud, Khalil Ghorbal, Albert Benveniste, Mathias Malandain.

**Project:** ANR-25-CE48-4916:GFdDAE

**Description:** For control systems based on discontinuous ODEs, Filippov solutions play an important role in the broad field of switched systems and also form the theoretical foundation for the so called sliding mode control method. The goal of this proposal is to extend the concept of Filippov solutions to discontinuous differential-algebraic equations (DAEs) of the form

$$E(x)\dot{x} = f(x) \tag{3}$$

where $E : \mathbb{R}^n \to \mathbb{R}^{l\times n}$ and $f : \mathbb{R}^n \to \mathbb{R}^l$ are possiblely discontinuous. The matrix $E(x)$ may not be invertible or even not be square, which introduces algebraic constraints to the systems. At present, there is no established theoretical foundation for studying discontinuous DAEs of the form (1). Without such a foundation, subsequent investigations such as numerical simulations, stability analysis, and controller design lack rigorous justification.

A central challenge in defining solutions for (1) lies in the absence of a precise notion of state-dependent jumps. As a defining characteristic of DAEs, induced jumps—namely, instantaneous state changes arising from consistent initialization—play a crucial role in the analysis of switched DAEs. While a comprehensive solution theory exists for time-dependent switching DAEs, a direct extension of the corresponding jump rules to the state-dependent case is generally not feasible. In particular, the resulting initialization may conflict with the active regions determined by the state-dependent switching signal. An additional mathematical difficulty arises from the presence of Dirac impulses (i.e., derivatives of jumps) that may occur in response to state jumps.

Embedding the proposed solution concept into simulation software, especially DAE-based languages such as Modelica, poses further significant challenges. Multi-mode DAEs frequently give rise to issues including compilation failures, modeling inconsistencies, and inadequate handling of abrupt transitions. The core difficulty lies in faithfully translating the theoretical framework into practical simulation tools while ensuring accurate and efficient treatment of both continuous dynamics and discrete jumps at switching boundaries. Addressing this challenge requires a robust simulation framework capable of handling complex state-dependent switching mechanisms and transitions.

**CityVal Line B**

**Participants:** Benoît Caillaud, Mathias Malandain.

The Hycomes team is currently involved in the development of a digital twin of part of Line B of the Rennes subway. These works are carried out in collaboration with the I4S team (Inria, Rennes and Gustave Eiffel University, Nantes, France).

Focusing on a portion of the viaduct for line B, the considered subsystem consists of the concrete rollways, the electric supply and guiding rails, the defrosting/heating elements embedded in the rollways, and the electric control stations. Its digital twin is designed to predict the heating requirements for defrosting the outdoor subway track and send energy-optimal heating commands to the control stations. It is made of a finite element model of heat balance (diffusion, convection and radiation coupling) in the viaduct, interacting with a 3D environmental model (for drop shadows), measurements (real-time data from thermocouples and weather stations), weather forecasts, and a model of the heating control system. Physical modeling and software architecture are carried out by Benoît Caillaud and Mathias Malandain, as well as Jean Dumoulin and Thibaud Toullier (Gustave Eiffel University), with the involvement of Siemens Mobility (designer and manufacturer) and Keolis (operator).

The purpose of this digital twin, seen as a research artifact, is to be used as a testbed for model-based optimization and health monitoring techniques to be developed jointly between the I4S and Hycomes teams.

## 9.3 Regional initiatives

**Participants:** Yahao Chen(PI), Benoît Caillaud, Albert Benveniste.

**Project:** Rennes metropole allocation d'installation scientifique (AIS)

**Title:** Contracts theory based on DAEs

**Description:** As an existing research axis of the HYCOMES team, contract-based design is a modular methodology that enables independent component development while ensuring correct systemwide integration. The aim of this project is to extend contract theory to DAEs, for example, of the form

$$E\dot{z} = Az.$$

In recent years, the design and analysis of large-scale control systems have become increasingly challenging. To address this, contract-based design has been introduced into the control systems domain. For example, for linear time-invariant (LTI) control systems:

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du. \end{aligned}$$

the classical behavioral theory introduced by Jan Willems is used to formalize key contract-theoretic notions—such as assumptions, guarantees, refinementand composition. Moreover, geometric control theory is employed to define simulation relations between two control systems, providing a foundation for implementing assume-guarantee contracts.

The reasons for choosing DAEs over control systems is that DAEs offer several potential advantages for contract-based analysis: (i) System interconnections can be naturally expressed as algebraic equations, supporting a compositional framework. (ii) DAEs treat all variables uniformly—states, inputs, and outputs—aligning well with the behavioral approach. (iii) The geometric analysis of DAEs is well established, providing effective tools for describing relations between systems and specifications.

# 10 Dissemination

## 10.1 Promoting scientific activities

### 10.1.1 Scientific events: selection

**Chair of conference program committees**

- Benoît Caillaud is Program Committee Chair of the EOOLT 2026 workshop on equation-based object-oriented modeling languages and tools, that will take place in Bielefeld, Germany, in June 2026.

**Member of the conference program committees**

- Benoît Caillaud has served on the program committe of the 16th International Modelica & FMI Conference, Luzern, Switzerland, September 2025.

**Reviewer**

- Benoît Caillaud has reviewed a paper for the HSCC & ICPS 2026 conference.
- In 2025, Yahao Chen served as a reviewer for the CDC, ECC and HSCC conferences.

### 10.1.2 Journal

**Member of the editorial boards**

- Benoît Caillaud has served on the editorial board of the Cambridge University Press open access journal *Research Directions: Cyber-Physical Systems*. The journal was terminated in 2025 because of its poor financial profitability.

**Reviewer - reviewing activities**

- Khalil Ghorbal served as a reviwer for the Journal of Computational Algebra (Open access journal edited by Elsevier)
- Yahao Chen served as a reviewer for the IEEE Transactions on Automatic Control and IEEE Control System Letters.

### 10.1.3 Invited talks

- Yahao Chen was invited to the CSE-SYNOBS-2025, Paris Mine. January 21 2025. "Robust LPV torque observation for WRSM with model uncertainties".
- Yahao Chen was invited to the MINGUS seminar, Rennes. March 3rd 2025. "From differential-algebraic equations to control systems and back".
- In May 2025, Khalil Ghorbal was invited to the 13th edition of Differential Algebra and Related Topics (DART) which is a series of *invitation-based* workshops gathering mathematicians and computer scientists. During this event a short lecture about *generalized polynomials* and how to manipulate them effectively was provided.
- Khalil Ghorbal was invited to the 30th edition of Applications of Computer Algebra (ACA) which is a conference series with parallel sessions devoted to promoting all kinds of computer algebra applications as well as promoting interactions between computer scientists, engineers, educators, and mathematicians. (July 2025, AADIOS Session). During this event, we showed how our method can be used to *solve an open conjecture* about the existence of Darboux polynomials for a well known physical model for turbulent convection (slides).

### 10.1.4 Leadership within the scientific community

- In july 2025 Khalil Ghorbal was elected as ‘co-responsable du groupe de travail Calcul formel (GT-CF) du groupement de recherche Informatique fondamentale et ses mathématiques (GdR IFM). (website).

### 10.1.5 Scientific expertise

- Albert Benveniste has directed the publication of the public report *Les jumeaux numériques – Un atout incontournable pour l’industrie 4.0 et le développement des secteurs du vivant* by the French Academy of Technologies.

## 10.2 Teaching - Supervision - Juries - Educational and pedagogical outreach

### 10.2.1 Teaching

- Yahao Chen, Advanced control of electric propulsion systems, 30 TD, Eramus E-pico Master program, Master 2, Ecole Centrale Nantes.

### 10.2.2 Supervision

- Khalil Ghorbal supervised Maxime Bridoux. Agrégé informatique. PhD Student in Computer Science (2022-2025). Thesis entitled *Inferring and exploiting necessary conditions for the existence of Darboux polynomials*. Defended November 28th 2025.
- Khalil Ghorbal supervised Damien Lejosne, student at ENS Rennes, during his L3 internship, 19/05/2025–05/07/2025, on the formal verification of electric power systems.
- Benoît Caillaud and Khalil Ghorbal jointly supervise the M1 research project of Damien Lejosne, student at ENS Rennes, on the formal verification of electric power networks.

### 10.2.3 Juries

- Benoît Caillaud has served on the PhD jury of Jon Tinnerholm, doctoral student at the University of Linköping, Sweden [75]. He has also been external examiner and jury member for the PhD of Moustafa Said Hawchar, University of Nantes.

## 10.3 Popularization

### 10.3.1 Others science outreach relevant activities

- On Saturday March 15th, INRIA Rennes welcomed 36 participants of different ages (from 5 to 60 years-old) for the semi-finals of the 39th edition of the *championnat des jeux mathématiques et logiques*. The event was organized by Khalil Ghorbal.
- Khalil Ghorbal presented the job of researcher for secondary schoolers (collège Bourgchevreuil, Cesson-Sévigné, Saturday February 1st 2025).

# 11 Scientific production

## 11.1 Major publications

[1] A. Benveniste, T. Bourke, B. Caillaud, J.-L. Colaço, C. Pasteur and M. Pouzet. ‘Building a Hybrid Systems Modeler on Synchronous Languages Principles’. In: *Proceedings of the IEEE*. Design Automation for Cyber-Physical Systems 106.9 (Sept. 2018), pp. 1568–1592. DOI: `10.1109/JPROC.2018.2858016`. URL: `https://hal.inria.fr/hal-01879026`.

[2] A. Benveniste, T. Bourke, B. Caillaud and M. Pouzet. ‘Non-standard semantics of hybrid systems modelers’. English. In: *Journal of Computer and System Sciences* 78.3 (2012). This work was supported by the SYNCHRONICS large scale initiative of INRIA, pp. 877–910. DOI: `10.1016/j.jcss.2011.08.009`. URL: `http://hal.inria.fr/hal-00766726` (cit. on p. 7).

[3] A. Benveniste, B. Caillaud and M. Malandain. ‘The mathematical foundations of physical systems modeling languages’. In: *Annual Reviews in Control* 50 (2020), pp. 72–118. DOI: `10.1016/j.arcontrol.2020.08.001`. URL: `https://hal.inria.fr/hal-03045498` (cit. on pp. 7, 13, 20).

[4] A. Benveniste, B. Caillaud, M. Malandain and J. Thibault. ‘Algorithms for the Structural Analysis of Multimode Modelica Models’. In: *Electronics* 11.17 (1st Sept. 2022), pp. 1–63. DOI: `10.3390/electronics11172755`. URL: `https://inria.hal.science/hal-03768331` (cit. on pp. 13, 20).

[5] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. Henzinger and K. G. Larsen. ‘Contracts for System Design’. In: *Foundations and Trends in Electronic Design Automation* 12.2-3 (2018), pp. 124–400. DOI: `10.1561/1000000053`. URL: `https://hal.inria.fr/hal-01971429` (cit. on p. 11).

[6] M. Bridoux and K. Ghorbal. *A Mathematica Package for Certifying the Nonexistence of Darboux Polynomials*. 16th July 2024. DOI: `10.1145/3712023.3712027`. URL: `https://inria.hal.science/hal-04818282` (cit. on p. 20).

[7] J.-B. Jeannin, K. Ghorbal, Y. Kouskoulas, A. Schmidt, R. Gardner, S. Mitsch and A. Platzer. ‘A Formally Verified Hybrid System for Safe Advisories in the Next-Generation Airborne Collision Avoidance System’. In: *International Journal on Software Tools for Technology Transfer* 19.6 (Nov. 2017), pp. 717–741. DOI: `10.1007/s10009-016-0434-1`. URL: `https://hal.archives-ouvertes.fr/hal-01232365` (cit. on p. 13).

[8] A. Sogokon, K. Ghorbal and T. T. Johnson. ‘Operational Models for Piecewise-Smooth Systems’. In: *ACM Transactions on Embedded Computing Systems (TECS)* 16.5s (Oct. 2017), 185:1–185:19. DOI: `10.1145/3126506`. URL: `https://hal.inria.fr/hal-01658196`.

## 11.2 Publications of the year

### International journals

[9] Y. Chen, M. Ghanes, A. Fekik and A. Maloum. ‘Torque Observation of WRSM With Model Uncertainties for EV Applications’. In: *IEEE Transactions on Control Systems Technology* (2025), pp. 1–13. DOI: `10.1109/tcst.2025.3609396`. URL: `https://hal.science/hal-05273039` (cit. on p. 21).

### International peer-reviewed conferences

[10] B. Caillaud, A. Benveniste and M. Malandain. ‘Benchmarking the Modular Structural Analysis Algorithm’. In: 2025 - 16th International Modelica & FMI Conference. Lucerne, Switzerland: Linköping University Press, 9th Sept. 2025, pp. 1–14. DOI: `10.3384/ecp218175`. URL: `https://inria.hal.science/hal-05257001` (cit. on pp. 13, 19).

### Reports & preprints

[11] A. Benveniste, B. Caillaud, Y. Chen, K. Ghorbal and M. Malandain. *Structural Methods for handling mode changes in multimode DAE systems*. RR-9603. Inria, 8th Jan. 2026, p. 61. URL: `https://inria.hal.science/hal-05385651`.

[12] K. Ghorbal and C. Kozaily. *On Covering Euclidean Space with Q-arrangements of Cones*. 25th June 2025. DOI: `10.1007/s10107-025-02252-x`. URL: `https://inria.hal.science/hal-04444572` (cit. on p. 21).

[13] G. Zhang, J. Jia, J. Jiao and Y. Chen. *Strong Structural Controllability Analysis of Structured Networks with Identical Nodes*. 1st Dec. 2025. URL: `https://hal.science/hal-04707907` (cit. on p. 22).

## 11.3 Cited publications

[14] L. de Alfaro. 'Game Models for Open Systems'. In: *Verification: Theory and Practice*. Vol. 2772. Lecture Notes in Computer Science. Springer, 2003, pp. 269–289. DOI: `10.1007/978-3-540-39910-0_12` (cit. on p. 11).

[15] L. de Alfaro and T. A. Henzinger. 'Interface automata'. In: *Proc. of the 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE'01)*. ACM Press, 2001, pp. 109–120. DOI: `10.1145/503271.503226` (cit. on p. 11).

[16] L. de Alfaro and T. A. Henzinger. 'Interface-based design'. In: *In Engineering Theories of Software Intensive Systems, proceedings of the Marktoberdorf Summer School*. Kluwer, 2004. DOI: `10.1007/1-4020-3532-2_3` (cit. on p. 11).

[17] L. de Alfaro, T. A. Henzinger and M. Stoelinga. 'Timed Interfaces'. In: *Proc. of the 2nd International Workshop on Embedded Software (EMSOFT'02)*. Vol. 2491. Lecture Notes in Computer Science. Springer, 2002, pp. 108–122. DOI: `10.1007/3-540-45828-X_9` (cit. on p. 11).

[18] A. Antonik, M. Huth, K. G. Larsen, U. Nyman and A. Wasowski. '20 Years of Modal and Mixed Specifications'. In: *Bulletin of European Association of Theoretical Computer Science* 1.94 (2008). URL: `https://dblp.org/rec/journals/eatcs/AntonikHLNW08.bib` (cit. on p. 11).

[19] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, Cambridge, 2008. URL: `https://mitpress.mit.edu/9780262026499/principles-of-model-checking/` (cit. on p. 11).

[20] A. Benveniste, T. Bourke, B. Caillaud, J.-L. Colaço, C. Pasteur and M. Pouzet. 'Building a Hybrid Systems Modeler on Synchronous Languages Principles'. In: *Proceedings of the IEEE*. Design Automation for Cyber-Physical Systems 106.9 (Sept. 2018), pp. 1568–1592. DOI: `10.1109/JPROC.2018.2858016`. URL: `https://inria.hal.science/hal-01879026` (cit. on p. 7).

[21] A. Benveniste, T. Bourke, B. Caillaud, B. Pagano and M. Pouzet. *A Type-Based Analysis of Causality Loops In Hybrid Systems Modelers*. Deliverable D3.1_1 v 1.0 of the Sys2soft collaborative project "Physics Aware Software". Dec. 2013. URL: `https://inria.hal.science/hal-00938866` (cit. on p. 7).

[22] A. Benveniste, T. Bourke, B. Caillaud and M. Pouzet. *Semantics of multi-mode DAE systems*. Deliverable D.4.1.1 of the ITEA2 Modrio collaborative project. Aug. 2013. URL: `https://inria.hal.science/hal-00938891` (cit. on p. 7).

[23] A. Benveniste, B. Caillaud, A. Ferrari, L. Mangeruca, R. Passerone and C. Sofronis. 'Multiple viewpoint contract-based specification and design'. In: *Proceedings of the Software Technology Concertation on Formal Methods for Components and Objects (FMCO'07)*. Vol. 5382. Revised Lectures, Lecture Notes in Computer Science. Amsterdam, The Netherlands: Springer, Oct. 2008. DOI: `10.1007/978-3-540-92188-2_9` (cit. on p. 11).

[24] A. Benveniste, B. Caillaud and M. Malandain. *Structural Analysis of Multimode DAE Systems: summary of results*. Research Report RR-9387. Inria Rennes – Bretagne Atlantique, Jan. 2021, p. 27. URL: `https://inria.hal.science/hal-03104030` (cit. on p. 7).

[25] A. Benveniste, B. Caillaud, M. Malandain and J. Thibault. 'Towards the separate compilation of Modelica: modularity and interfaces for the index reduction of incomplete DAE systems'. In: *Linköping Electronic Conference Proceedings*. Vol. 204. Aachen, Germany, Oct. 2023, p. 10. DOI: `10.3384/ecp204`. URL: `https://inria.hal.science/hal-04295096` (cit. on pp. 13, 19).

[26] A. Benveniste, B. Caillaud, B. Pagano and M. Pouzet. 'A Type-Based Analysis of Causality Loops in Hybrid Modelers'. In: *Proceedings of the 17th international conference on Hybrid systems: computation and control (HSCC 2014)*. Proceedings of the 17th international conference on Hybrid systems: computation and control (HSCC '14). Berlin, Germany: ACM Press, Apr. 2014, p. 13. DOI: `10.1145/2562059.2562125`. URL: `https://inria.hal.science/hal-01093388` (cit. on p. 7).

[27] N. Bertrand, A. Legay, S. Pinchinat and J.-B. Raclet. 'A Compositional Approach on Modal Specifications for Timed Systems.' In: *LNCS*. Vol. 5885. LNCS. Rio de Janeiro, Brazil: Springer, Dec. 2009, pp. 679–697. URL: `https://inria.hal.science/inria-00424356` (cit. on p. 11).

[28] N. Bertrand, A. Legay, S. Pinchinat and J.-B. Raclet. 'Modal event-clock specifications for timed component-based design'. In: *Science of Computer Programming* 77 (2012), pp. 1212–1234. DOI: 10.1016/j.scico.2011.01.007. URL: https://inria.hal.science/hal-00752449 (cit. on p. 11).

[29] N. Bertrand, S. Pinchinat and J.-B. Raclet. 'Refinement and Consistency of Timed Modal Specifications.' In: *LNCS*. Vol. 5457. LNCS. Tarragona, Spain: Springer, Apr. 2009, pp. 152–163. DOI: 10.1007/978-3-642-00982-2_13. URL: https://inria.hal.science/inria-00424283 (cit. on p. 11).

[30] P. Bhaduri and I. Stierand. 'A proposal for real-time interfaces in SPEEDS'. In: *Design, Automation and Test in Europe (DATE'10)*. IEEE, 2010, pp. 441–446. DOI: 10.1109/DATE.2010.5457163 (cit. on p. 11).

[31] M. Blanke, M. Kinnaert, J. Lunze and M. Staroswiecki. 'Diagnosis and Fault-Tolerant Control'. In: Springer Berlin, Heidelberg, Sept. 2006, pp. 109–188. DOI: https://doi.org/10.1007/978-3-540-35653-0 (cit. on p. 20).

[32] S. Bliudze. 'Un cadre formel pour l'étude des systèmes industriels complexes: un exemple basé sur l'infrastructure de l'UMTS'. PhD thesis. Ecole Polytechnique, 2006 (cit. on pp. 7, 8).

[33] S. Bliudze and D. Krob. 'Modelling of Complex Systems: Systems as Dataflow Machines'. In: *Fundamenta Informaticae* 91.2 (2009), pp. 251–274. DOI: 10.3233/FI-2009-0043. URL: https://hal.science/hal-02561099 (cit. on p. 8).

[34] G. Boudol and K. G. Larsen. 'Graphical versus logical specifications'. In: *Theoretical Computer Science* 106.1 (1992), pp. 3–20. DOI: https://doi.org/10.1016/0304-3975(92)90276-L. URL: https://www.sciencedirect.com/science/article/pii/030439759290276L (cit. on p. 11).

[35] B. Caillaud, B. Delahaye, K. G. Larsen, A. Legay, M. L. Pedersen and A. Wasowski. 'Compositional design methodology with constraint Markov chains'. In: *QEST 2010*. Williamsburg, Virginia, United States, Sept. 2010. DOI: 10.1109/QEST.2010.23. URL: https://inria.hal.science/inria-00591578 (cit. on p. 11).

[36] B. Caillaud, B. Delahaye, K. G. Larsen, A. Legay, M. L. Pedersen and A. Wasowski. 'Constraint Markov Chains'. In: *Theoretical Computer Science* 412.34 (May 2011), pp. 4373–4404. DOI: 10.1016/j.tcs.2011.05.010. URL: https://inria.hal.science/hal-00654003 (cit. on p. 11).

[37] B. Caillaud, M. Malandain and J. Thibault. 'Implicit structural analysis of multimode DAE systems'. In: *HSCC 2020 - 23rd ACM International Conference on Hybrid Systems: Computation and Control*. Sydney New South Wales Australia, France: ACM, Apr. 2020, pp. 1–11. DOI: 10.1145/3365365.3382201. URL: https://inria.hal.science/hal-02572879 (cit. on pp. 9, 13, 14).

[38] S. L. Campbell and C. W. Gear. 'The index of general nonlinear DAEs'. In: *Numerische Mathematik* 72.2 (Dec. 1995), pp. 173–196. DOI: 10.1007/s002110050165. URL: http://dx.doi.org/10.1007/s002110050165 (cit. on p. 8).

[39] F. Casella and A. Guironnet. 'ScalableTestGrids - An Open-Source and Flexible Benchmark Suite to Assess Modelica Tool Performance on Large-Scale Power System Test Cases'. In: *Proceedings of the 14th International Modelica Conference*. Ed. by M. Sjölund, L. Buffoni, A. Pop and L. Ochel. Linköping Electronic Conference Proceedings 181. Linköping, Sweden: Modelica Association and Linköping University Electronic Press, Sept. 2021, pp. 351–358. DOI: 10.3384/ecp21181351 (cit. on p. 19).

[40] A. Chakrabarti. 'A Framework for Compositional Design and Analysis of Systems'. PhD thesis. EECS Department, University of California, Berkeley, Dec. 2007. URL: http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-174.html (cit. on p. 11).

[41] A. Chakrabarti, L. de Alfaro, T. A. Henzinger and M. Stoelinga. 'Resource Interfaces'. In: *Embedded Software, Third International Conference, EMSOFT 2003, Philadelphia, PA, USA, October 13-15, 2003, Proceedings*. Vol. 2855. Lecture Notes in Computer Science. Springer, 2003, pp. 117–133. DOI: 10.1007/978-3-540-45212-6_9 (cit. on p. 11).

[42] E. Y. Chang, Z. Manna and A. Pnueli. ‘Characterization of temporal property classes’. In: *ICALP*. Vol. 623. Lecture Notes in Computer Science. Springer, 1992, pp. 474–486. DOI: 10.1007/3-540-55719-9_97 (cit. on p. 11).

[43] E. Clarke, O. Grumberg and D. Peled. *Model Checking*. MIT Press, 1999. URL: https://mitpress.mit.edu/9780262038836/model-checking/ (cit. on p. 11).

[44] N. J. Cutland, ed. *Nonstandard Analysis and its Applications*. Cambridge Univ. Press, 1988. DOI: 10.1017/CBO9781139172110 (cit. on p. 7).

[45] A. David, K. G. Larsen, A. Legay, U. Nyman and A. Wasowski. ‘ECDAR: An Environment for Compositional Design and Analysis of Real Time Systems’. In: *Automated Technology for Verification and Analysis - 8th International Symposium, ATVA 2010, Singapore, September 21-24, 2010. Proceedings*. 2010, pp. 365–370. DOI: 10.1007/978-3-642-15643-4_29 (cit. on p. 11).

[46] A. David, K. G. Larsen, A. Legay, U. Nyman and A. Wasowski. ‘Timed I/O automata: a complete specification theory for real-time systems’. In: *Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2010, Stockholm, Sweden, April 12-15, 2010*. 2010, pp. 91–100. DOI: 10.1145/1755952.1755967 (cit. on p. 11).

[47] B. Delahaye, J.-P. Katoen, K. G. Larsen, A. Legay, M. L. Pedersen, F. Sher and A. Wasowski. ‘Abstract Probabilistic Automata’. In: *Verification, Model Checking, and Abstract Interpretation - 12th International Conference, VMCAI 2011, Austin, TX, USA, January 23-25, 2011. Proceedings*. Vol. 6538. Lecture Notes in Computer Science. 2011, pp. 324–339. DOI: 10.1007/978-3-642-18275-4_23 (cit. on p. 11).

[48] F. Diener and G. Reeb. *Analyse non standard*. Hermann, 1989. URL: https://www.editions-hermann.fr/livre/analyse-non-standard-francine-diener (cit. on p. 7).

[49] D. L. Dill. *Trace Theory for Automatic Hierarchical Verification of Speed-Independent Circuits*. ACM Distinguished Dissertations. MIT Press, 1989. DOI: 10.7551/mitpress/6874.001.0001 (cit. on p. 11).

[50] J. Edmonds and R. M. Karp. ‘Theoretical improvements in algorithmic efficiency for network flow problems’. In: *Journal of the ACM* 19.2 (1972), pp. 248–264. DOI: 10.1145/321694.321699. URL: http://dx.doi.org/10.1145/321694.321699 (cit. on p. 9).

[51] H. Elmqvist, S. E. Mattsson and M. Otter. ‘Modelica extensions for Multi-Mode DAE Systems’. In: *Proceedings of the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden*. Linköping University Electronic Press, Mar. 2014. DOI: 10.3384/ecp14096183 (cit. on p. 13).

[52] H. Elmqvist, A. Neumayr and M. Otter. ‘Modia-dynamic modeling and simulation with julia’. In: *Juliacon’18*. University College London, UK, Aug. 2018. URL: https://elib.dlr.de/124133/ (cit. on p. 14).

[53] H. J. Ferreau, S. Almér, H. Peyrl, J. L. Jerez and A. Domahidi. ‘Survey of industrial applications of embedded model predictive control’. In: *2016 European Control Conference (ECC)*. 2016, pp. 601–601. DOI: 10.1109/ECC.2016.7810351 (cit. on p. 14).

[54] E. Frisk, A. Bregon, J. Aslund, M. Krysander, B. Pulido and G. Biswas. ‘Diagnosability analysis considering causal interpretations for differential constraints’. In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 42.5 (2012), pp. 1216–1229 (cit. on p. 20).

[55] A. V. Goldberg and R. E. Tarjan. ‘A new approach to the maximum flow problem’. In: *Proceedings of the eighteenth annual ACM symposium on Theory of computing (STOC’86)*. 1986. DOI: 10.1145/12130.12144. URL: http://dx.doi.org/10.1145/12130.12144 (cit. on p. 9).

[56] F. Hashemniya, B. Caillaud, E. Frisk, M. Krysander and M. Malandain. ‘Fault Diagnosability Analysis of Multi-Mode Systems’. In: *SAFEPROCESS 2024 - 12th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*. Vol. 58. 4. IFAC. Ferrara, Italy: Elsevier, June 2024, pp. 210–215. DOI: 10.1016/j.ifacol.2024.07.219. URL: https://inria.hal.science/hal-04803147 (cit. on p. 20).

[57] *IEEE Standard VHDL Analog and Mixed-Signal Extensions, Std 1076.1-1999*. 1999. DOI: 10.1109/IEEESTD.1999.90578. URL: http://dx.doi.org/10.1109/IEEESTD.1999.90578 (cit. on p. 7).

[58] R. Isermann. *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer Berlin Heidelberg, 2006 (cit. on p. 20).

[59] Y. Iwasaki, A. Farquhar, V. Saraswat, D. Bobrow and V. Gupta. 'Modeling time in hybrid Systems: How fast is "instantaneous"?' In: *IJCAI*. 1995, pp. 1773–1781. URL: `https://www.ijcai.org/Proceedings/95-2/Papers/097.pdf` (cit. on p. 8).

[60] A. Lamercerie. 'Principe de transduction sémantique pour l'application de théories d'interfaces sur des documents de spécification'. Theses. Université de Rennes ; Rennes 1, Apr. 2021. URL: `https://theses.hal.science/tel-03366457` (cit. on p. 10).

[61] L. Lamport. 'Proving the correctness of multiprocess programs'. In: *IEEE Trans. Software Eng.* 3.2 (1977), pp. 125–143. DOI: `10.1109/TSE.1977.229904` (cit. on p. 11).

[62] K. G. Larsen, U. Nyman and A. Wasowski. 'On Modal Refinement and Consistency'. In: *Proc. of the 18th International Conference on Concurrency Theory (CONCUR'07)*. Springer, 2007, pp. 105–119. DOI: `10.1007/978-3-540-74407-8_8` (cit. on p. 11).

[63] K. G. Larsen and B. Thomsen. 'A Modal Process Logic'. In: *Proceedings of the Third Annual Symposium on Logic in Computer Science (LICS'88)*. IEEE, 1988, pp. 203–210. DOI: `10.1109/LICS.1988.5119` (cit. on p. 11).

[64] T. Lindstrøm. 'An invitation to nonstandard analysis'. In: *Nonstandard Analysis and its Applications*. Ed. by N. J. Cutland. Cambridge Univ. Press, 1988, pp. 1–105. DOI: `10.1017/CBO9781139172110.002` (cit. on p. 7).

[65] N. A. Lynch. 'Input/Output Automata: Basic, Timed, Hybrid, Probabilistic and Dynamic'. In: *CONCUR 2003 - Concurrency Theory, 14th International Conference, Marseille, France, September 3-5, 2003, Proceedings*. Vol. 2761. Lecture Notes in Computer Science. Springer, 2003, pp. 187–188. DOI: `10.1007/978-3-540-45187-7_12` (cit. on p. 11).

[66] N. A. Lynch and E. W. Stark. 'A Proof of the Kahn Principle for Input/Output Automata'. In: *Inf. Comput.* 82.1 (1989), pp. 81–92. DOI: `10.1016/0890-5401(89)90066-7` (cit. on p. 11).

[67] Z. Manna and A. Pnueli. *Temporal verification of reactive systems: Safety*. Springer, 1995. DOI: `10.1007/978-1-4612-4222-2` (cit. on p. 11).

[68] B. Meyer. 'Applying "Design by Contract"'. In: *Computer* 25.10 (Oct. 1992), pp. 40–51. DOI: `10.1109/2.161279`. URL: `http://dx.doi.org/10.1109/2.161279` (cit. on p. 10).

[69] P. Nuzzo, A. L. Sangiovanni-Vincentelli, X. Sun and A. Puggelli. 'Methodology for the Design of Analog Integrated Interfaces Using Contracts'. In: *IEEE Sensors Journal* 12.12 (Dec. 2012), pp. 3329–3345. DOI: `10.1109/JSEN.2012.2211098` (cit. on p. 11).

[70] C. Pantelides. 'The consistent initialization of differential-algebraic systems'. In: *SIAM J. Sci. Stat. Comput.* 9.2 (1988), pp. 213–231. DOI: `10.1137/0909014` (cit. on pp. 8, 9).

[71] J. D. Pryce. 'A Simple Structural Analysis Method for DAEs'. In: *BIT Numerical Mathematics* 41.2 (Mar. 2001), pp. 364–394. DOI: `10.1023/a:1021998624799`. URL: `http://dx.doi.org/10.1023/a:1021998624799` (cit. on pp. 9, 19).

[72] J.-B. Raclet, E. Badouel, A. Benveniste, B. Caillaud, A. Legay and R. Passerone. 'A Modal Interface Theory for Component-based Design'. In: *Fundamenta Informaticae* 108.1-2 (2011). To appear, pp. 119–149. DOI: `10.3233/FI-2011-416`. URL: `https://inria.hal.science/inria-00554283` (cit. on p. 11).

[73] A. Robinson. *Non-Standard Analysis*. Princeton Landmarks in Mathematics, 1996. URL: `https://press.princeton.edu/books/paperback/9780691044903/non-standard-analysis` (cit. on p. 7).

[74] E. Sikora, B. Tenbergen and K. Pohl. 'Industry needs and research directions in requirements engineering for embedded systems'. In: *Requirements Engineering* 17 (2012), pp. 57–78. DOI: `10.1007/s00766-011-0144-x`. URL: `http://link.springer.com/article/10.1007/s00766-011-0144-x` (cit. on p. 11).

[75] J. Tinnerholm. *Dynamic and Variable-Structure System Modeling for Equation-Based Languages : Applications, Methods and Tools*. Linköping, Sweden: Linköping University Electronic Press, 2025. DOI: `10.3384/9789181183375` (cit. on p. 26).

[76] S. Trenn. 'Switched Differential Algebraic Equations'. In: *Dynamics and Control of Switched Electronic Systems: Advanced Perspectives for Modeling, Simulation and Control of Power Converters*. Ed. by F. Vasca and L. Iannelli. London: Springer London, 2012, pp. 189–216. DOI: `10.1007/978-1-4471-2885-4` (cit. on p. 20).