Acronyme	**FEANICSES**	**Type of research**	Industrial research
Proposal title	**F**ormal and **E**xhaustive **A**nalyses of **N**umerical **I**ntensive **C**ontrol **S**oftware for **E**mbedded **S**ystems		
Funding call	Appel à projets générique 2017 Call JCJC (Jeunes Chercheuses, Jeunes Chercheurs)		
Theme(s)	DEFI 7 – Société de l'information et de la communication Axe 3 : Sciences et technologies logicielles Axe 4 : Interaction, robotique		
Requested funding	300 k€	**Duration**	48 months

Contents

1 Executive summary

1.1 Abstract

As computer power and memory continue to be commoditized, the pressure towards developing more complex, embedded, safety-critical software keeps growing. However, the resulting exponential growth of software verification and validation (V&V) and its certification are significant obstacles; It is often said that half the development cost of a complex, safety-critical system such as a commercial aircraft is currently absorbed by software certification. This cost becomes increasingly unbearable by industry, and may constitute a show-stopper for emerging systems, such as commercial unmanned aerial systems and autonomous cars. Many software V&V challenges can be traced to "intrinsic complexity", which makes certain advanced, e.g. autonomous, software-borne technologies out of reach for safety-critical applications.

The FEANICSES project is articulated around the identified need to support the analysis of system-level properties such as stability, robustness and performance, at all stages of the system

development including code level. The underlying predicate justifying FEANICSES is that all these properties can be expressed as numerical invariants over the system states, e.g. using Lyapunov functions. The approach proposed combines the expression of such properties and the definition of non linear static analysis techniques.

This interdisciplinary proposal will impact the evolution of industry practices supporting the making of safety-critical, software-enabled functions, and yielding faster convergence towards a consensus about the quality of the software, and its eventual certification. Associated collaborators include researchers from control and optimization in addition to computer scientists.

FEANICSES results will include an integrated analysis toolchain to analyze complex cyberphysical systems. The toolchain will support the end-to-end analysis of a controlled system included its system-level properties. The toolchain's final goal is to reduce software certification time and demonstrate this on several representative use cases.

1.2 Involved participants summary

Affiliation	Name	Position	Commitment (man-month)	Activities & Responsibilities
Onera	Pierre-Loïc Garoche	Research scientist (CS)	40.8 (85%)	Coordinator of FEANICSES
Onera	Rémi Delmas	Research scientist (CS)	9.6 (20%)	Participation to WP2, WP4 (challenges C2, C5)
Onera	Pierre Roux	Research scientist (CS)	9.6 (20%)	Participation to WP3 (challenges C3, C4)
ENSTA Paritech	Alexandre Chapoutot	Associate Prof. (CS)	9.6 (20%)	WP2 (challenges C3, C4), co-advising PhD2
LAAS – CNRS	Didier Henrion	DR CNRS (CT, Opt)	9.6 (20%)	WP1, WP2 (challenges C1, C3), co-adivisng Postdoc, already co-advising two PhD students
Univerty of Washington	Behçet Açıkmeşe	Associate Prof. (CT)	9.6 (20%)	WP2, WP4 (challenges C1, C5), co-advising PostDoc
Georgia Tech	Eric Féron	Professor (CT)	9.6 (20%)	WP1, co-advising Postdoc, already co-advising one PhD student
NASA Ames	Guillaume Brat	Researcher (CS)	–	WP2, WP4 to be co-advisor of H. Bourbouh thesis
Raphaël Cohen		PhD student (Onera-Georgia Tech)		WP1, WP4 (challenges C1, C5) co-advised with E. Féron
Guillaume Davy		PhD student (ENS Cachan funding)		WP1, WP4 (challenges C1, C5), co-advised with D. Henrion
Hamza Bourbouh		PhD student (NASA funding)		WP1, WP4 (challenges C1, C2, C5), NASA employee, to be co-advised with G. Brat
PhD1		PhD student (FEANICSES – Onera)		WP1, WP3 (challenge C4) to be co-advised with A. Chapoutot
PhD2		PhD student (FEANICSES – UW fund.)		WP2, WP4 (challenges C1, C2, C5), to be co-advised with B. Açıkmeşe
PostDoc1		PostDoc (FEANICSES funding)		WP2, (challenges C1, C3, C5), to be co-advised with E. Féron and D. Henrion

Fields: CS: Computer Science, CT: Control Theory, Opt: Optimization

1.3 Evaluation of full proposal content with respect to initial preproposal

A few modifications have been made with respect to the submitted preproposal, following the feedback received from the reviewers.

- **Evolution of budget, total costs and mission budget.** Both reviewers question the percentage of mission cost with respect to project cost. Let us first note the FEANICSES project is covered at marginal costs by ANR while Onera employee salaries are not directly covered by the project. Onera has therefore to support the commitment of three researchers to that important project for a total of 60 months about 545 k€. Furthermore the mission budget is meant to cover the related trips of associated partners mentioned in Sect.1.2 as well as the yearly workshop.

  To answer reviewer concerns, we reduced the mission budget to 75 k€ (instead of 81k€). To balance the participation to different work packages, we increase the Postdoc involvement to 18 months (instead of 12). We also reduced to 7k€ the material budget for student laptop while adding the cost of 1 small UAVs and 1 robot (about 2k each) on which to illustrate the toolchain at Onera, since the full test-bench by Price Induction is much more expensive.

  Overall, with all those modifications, the travel budget represents now 25% of requested funding. Considering the 100 permanent man months (60 at Onera, 40 at GT, UW, ENSTA and LAAS[1]) and the 162 non-permanent man.months (existing PhD thesis plus FEANICSES funded students), it corresponds to 286€ per involved man-month. The workshop budget is about 10k per year, the remaining travel budget for project participants is 35k€, ie. 133€ per man.month and 11% of requested funding.

- **Workshop organization and researcher invitations.** The mission budget will largely be used to support the organization of yearly workshops in Toulouse. In the past ANR ASTRID VORACE project, we had a great success and outcomes in the organization of four of these yearly workshops. Those workshops were fully covered by the project funding including the mission costs of all participants. These invitations were targeting major scientists from their field and generated fruitful discussions and new collaborations. We are motivated to initiate a similar synergy around FEANICSES project topics.

- **About the industrial impact of project results**, the current interest by some industrial in the toolchain we developed with NASA[2], as well as the commitment to the project by the French SME Price Induction, outlines the possible uses of FEANICSES methods and tools by more general industrials.

- **Regarding project organization, risk management and relationships between funded PhD theses and project**, all these elements have been clarified in each work-packages description.

- **External collaborators** were reinforced with Guillaume Brat, a senior Computer Science research scientist and head of the Robust Software Engineering group at NASA Ames. Our existing collaboration with him and his group (about two visits a year since 2012) will mainly be focused on WP2 and WP4, co-developping open-source prototype implementing the analyses. We also decided to co-advise the PhD of Hamza Bourbouh, a previous student of mine, now an engineer at NASA, as part of the FEANICSES project.

2 Context, positioning and objectives of the proposal

This project takes place in the context of critical embedded systems development and verification. We focus on reactive systems typically found in control applications. Contrary to transformational systems, whose executions terminate and produce a value, reactive systems sustain a permanent interaction with their environment and their execution never terminates. For synchronous systems,

Algorithm 1 Infinite Reactive Loop

```
do_init();
while true do
  read_input();
  compute_reaction();
  write_output();
end while
```

the underlying algorithm can be sketched as:

In the aircraft industry, over the last 30 years, an increasing amount of functionality has been implemented in software. In their earliest and simplest form, critical software were limited to local actuator control or to purely logical tasks of a system. Nowadays, a typical reactive software tightly combines numerically intensive computations with discrete and temporal logic, and is in charge of advanced fault diagnosis, system reconfiguration, flight mode management, communication, etc. Let us consider the evolution of Airbus airliners code:

Between the A310 (1975) and A380 (2005) programs, the amount of embedded software rose from a few kilo-bytes to more than a 100 mega-bytes, while the CPU power has been multiplied by a factor 100. This rapid evolution raises technical issues with respect to design and verification processes, as it is now impossible to guarantee by testing alone that the final software actually fulfills its specification and is safe to operate in all possible conditions.

Moreover, the design of reactive software is strongly correlated with a model of the physical environment, mainly because it is based on techniques coming from control theory. In this theory, the software exists in the context of a bigger system, and its purpose is to restrict and control the behaviors of this bigger system. An abstract model of such a system is given in Figure 1. The software takes its input from *sensors* reading the state of the physical environment and controls the physical environment through *actuators*. The modification of the state of the physical environment decided by the software depends on a particular input named *reference* (modeling the desired system behavior) and *external input* (modeling the possible interaction with other subsystems).

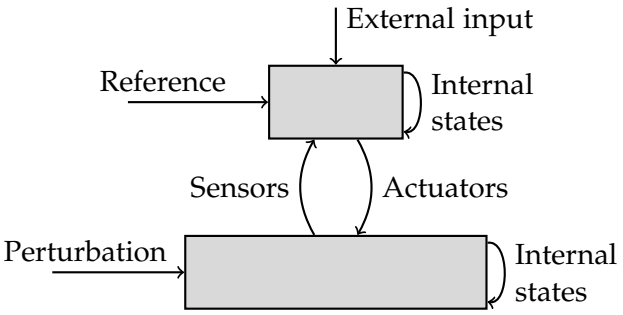


Figure 1: *General schema of control systems.*

Figure 1 represents the next challenge of applying formal methods in the development cycle of embedded software. Indeed such software is specified in terms of the physical environment's behavior. So verifying reactive software should be also done while taking into account the physical environment's behavior. However, modeling the physical environment's behavior increases the complexity of systems on which formal methods have to be applied, because software is based on discrete-time semantics while physical environment is based on continuous-time semantics. Another major challenge is to enable the analysis of advances controllers such as Model-Predictive Controllers (MPC) that rely on advanced algorithms, typically convex optimization, to perform the control computation.

2.1 Context, social and economic issues

The recent evolution of cyber-physical systems is marked by several trends that make their insertion into safety-critical environments increasingly problematic.

- a trend towards a 'software everything, everywhere' environment, where software is consid-

[1]Note that no mission could be paid to USA federal employees, that includes our NASA partner G. Brat (but not the PhD student H. Bourbouh.)

[2]Eg. cf. https://ti.arc.nasa.gov/news/RSE-Develops-Tools-With-GE/

ered to be a flexible means to enable new system functions at a relatively low cost.

- a trend towards increasing the complexity of the functions managed by software. What used to be relatively procedural, deterministic and easily understood software routines have now often become full-fledged problem-solving capabilities where it becomes very difficult for the engineer, operator and regulatory body to understand the details of what is going on. This trend has accelerated with the introduction of increasingly smart autonomous systems, e.g. robots, UAVs, Cube/MicroSat, autonomous cars, where autonomous behavior is generated by implementing advanced search algorithms and executing them on-line.

Cyberphysical systems that leverage these capabilities are, for the most part, still confined to the laboratory or to other controlled environments. One exception is the aerospace industry, which has systematically explored and implemented advanced functions onboard air and space vehicles for the purpose of reducing human exposure to danger (military drones), increase system safety (commercial aviation), or simply enable exploration currently unthinkable for humans (planetary exploration).

The aerospace's safety culture has also led to the development of increasingly demanding software assurance verification techniques. If the standards used by aerospace systems are to be used throughout the safety-critical cyber-physical system industry, the prospects are somewhat daunting: The German Aerospace Research Establishment (DLR: Deutsches Zentrum für Luft- und Raumfahrt) is often quoted for its devastating productivity estimate of **0.6 lines of code per hour per software engineer** for its mission-critical space applications [3].

The trend towards ever increasing software development and needs for verification is not likely to change anytime soon. The current safety standards for commercial aviation are among industry's highest. Yet they may not be sufficient for the future. The current requirement against catastrophic, lethal events requires system failure rates to be under 10^{-9} per flight hour. With the current traffic and the current safety requirements, that is about one incident per month world-wide. With a traffic growth of about 5% per year, traffic will double in 15 years and triple in 25 years. Today's safety requirements will therefore allow one accident to occur every other week in 15 years, and one accident to occur every 10 days in 25 years, and these figures do not account for the increased collision rate due to increased traffic density. Such figures of one catastrophe per ten days is unlikely to be tolerated by the public, as illustrated by the Malaysian Airlines 370 accident, and much of the needed **improved aircraft reliability** is likely to be **realized**, in part, by ever **more sophisticated, yet certified, software functions**.

Institutional and industrial response to the mounting software assurance challenge has been slow, but steadily oriented towards system safety. Essential documents, such as RTCA's DO-178B, can be credited not only for providing software development guidance to the aerospace industry, but also for being influential for all the other safety-critical industries (nuclear, automotive, medical, ...).

RTCA's DO-178B was, however, developed at a time when software size and functionality was considerably smaller than what it has become today. FEANICSES' ambition is to take advantage of its recent update, RTCA DO-178C [4], together with its supplements DO-333 [5], DO-331 [6], to tackle the complexity of today's embedded software development, most notably advanced command and control software. In particular, our view is that RTCA DO-178C (and its supplements), because it authorizes the **use of advanced mathematical techniques to support system verification** and validation, offers the potential to unleash many computer-aided mathematical analysis techniques without which we would not be able to decipher and support the complex semantics of today's and tomorrow's advanced software. In these respects it paves the ground for the safety standards of other industry sectors, so focusing on DO-178C with its strict verification requirements allows us to develop solutions that will also apply to the requirements of future safety standards in other domains.

[3] A typical modern aircraft flight controller is about 300K lines of code while NASA Mars Exploration Rover mission flight software is about 650K lines of code.

[4] Software Considerations in Airborne Systems and Equipment Certification

[5] Formal Methods Supplement to DO-178C and DO-278A

[6] DO-331 Model-Based Development and Verification Supplement to DO-178C and DO-278A

Despite the growing interest of industrial users, formal verification has not yet reached its full deployment potential. Costly human intervention is still needed in order to obtain the results that are necessary to certify real world applications. In fact, **industry often finds that current formal verification tools do not exactly fit their certification practices**, and end up repeating verification tasks performed by other methods, resulting in *increased* certification costs. Furthermore, the **current state of the art of formal verification is mainly focused on low level software properties while the high level properties of the systems are rarely validated** all along the development chain.

Concerning the new markets of civil UAVs and Cube/Microsatellites[7], newcomers do not necessarily have a strong background in terms of processes to develop their critical systems. Therefore the cost associated to these developments could be a show-stopper. **It is mandatory to propose new development processes that integrate formal methods seamlessly, achieving the highest level of certification in an automatic fashion for a reduced cost.**

The last year or months have seen the attempts of SpaceX and BlueOrigin to pinpoint land rockets. Not so long ago, we saw the successful planetary landing of NASA Curiosity rover. In the US, and soon in EU, the NextGen collision advoidance algorithm ACAS-X from MIT Lincoln Lab will be used to replace T-CAS in civil aircrafts. Last, fuel efficiency or better use of aircraft or rocket engine is mainly dependent on the complexity of the FADEC (Full Authority Digital Engine Control), the software controlling the engine. **Enabling the use of numerical intensive control software requires new methods to validate their uses and enabling their certification.**

(a) Masten Xombie Rocket

(b) PriceInduction DGEN380 Engine

Figure 2: Systems requiring numerical intensive controllers

The FEANICSES project is therefore about pushing an evolution of the software development process through the design of a **formal methods toolchain** and its components, allowing industry to tackle **more capable** software systems **faster, that is, at a lower cost**. We believe one of the keys to FEANICSES' success will be to revisit the current paradigm based on separation of concerns, whereby systems engineers specify the software system, software engineers and programmers implement the system, and communication across the two groups is kept at a minimum. Instead, we target an **integrated process**, where software traceability is not only achieved with documents and processes, but also with formal **semantics preservation all along the process, that is including system level properties**. We believe that the creation of this process will help formal verification methods reach their full deployment potential, and **avoid the costly human intervention** that is still necessary to obtain results of significance to the real world.

The FEANICSES team will work on the design, implementation and evaluation of innovative cooperations of formal techniques, with the double objective of **increasing the level of automation, scalability** and **robustness** of formal verification in the one hand, and improving the **automatic generation of faithful implementations** from models on the other hand. FEANICSES will also contribute to moving the focus of formal methods away from low-level, nonfunctional software properties, towards higher-level, functional properties.

[7] According to market analyses, UAVs market will be worth +$100B in 2023, MicroSat market will reach $2B in 2019.

The de facto *content-based* traceability resulting from FEANICSES will be key to enabling a true interaction between the actors of the development process: properties addressed at one level will be also evaluated or their preservation checked in the later phases, feedback will be provided either through invariants, properties or, when a property is broken, via counter examples. This will enable short loop feedback closer to spiral development of embedded systems rather than traditional, and expensive "Vee" development structure common to many complex systems development.

The FEANICSES project will expand the knowledge and practices of system-level property analysis at model and code level, providing a backbone for next-generation process development of critical CPS, with highly integrated formal methods.

2.2 Position of the project

The projects listed in this section illustrate that the field of research addressed by this project, the combination of formal methods, is very active today, both at national and international levels.

These projects can be categorized with respect to their relative positioning as identified in the ANR document, one of *past*, *concurrent* or *complementary*, represented by labels *P*, *Conc* and *Comp*, respectively. Complementary projects are projects involving participants of the FEANICSES project. Project ending in 2016, or early 2017 are systematically labeled as past projects.

ANR INS CAFEIN$^{[Comp,P]}$ I coordinated this ANR project focused on the Combination of Analyses for the Study of Numerical Invariants. This project ended in early 2017. With seven partners, and international collaboration with the NSF Project CrAVES, the project has been largely successful. We co-authored about 40 publications, developed analysis prototypes and applied them on representative examples. FEANICSES project is the natural continuation of CAFEIN focusing more on non linear invariants and system level properties.

NSF CPS CrAVES (USA)$^{[Comp,P]}$ CrAVES stands for Credible Autocoding and Verification of Embedded Software. This project gathered people from control system theory at Georgia Tech (Eric Féron) and static analysis at NASA/CMU (Arnaud Venet). It aims at easing the development and validation of critical control software by automatizing the generation of the code while ensuring that the system properties are preserved.

Prof. Féron is a collaborator of FEANICSES project. Furthermore CrAVES was associated to the ANR project CAFEIN I led, and its result will naturally be used as starting points for FEANICSES research activities.

ASTRID VORACE$^{[Comp,P]}$ The ASTRID project VORACE focused on two main goals: build an interdisciplinary research community gathering formal verification, control and convex optimization, and study the verification of convex optimization algorithms to enable their use in real-time embedded systems. I was the PI of this project for Onera. This project also ended in early 2017.

Through the VORACE project I met with lots of top-tier research in the field of optimization and control and identified interesting target systems or verification methods that worth to be transferred or analyzed with formal methods. For example this enabled us to rely on convex optimization to synthesize non linear invariants, providing that we check *a posteriori* the soundness, ie the feasibility, of the computed invariants.

NSF CPS SORTIES (USA)$^{[Comp]}$ As for the the relationship between CAFEIN and CrAVES, the SORTIES project was the NSF CPS counterpart of the VORACE project. Because of the consortium composition the balance was more on the control side but the motivation is similar: understand more finely the proof of optimization algorithms to be able to embed them in critical applications such as planetary landing or collision avoidance algorithms.

I was affiliated with the project and participated to their yearly meeting. I also exchanged students with their teams (co-advising of PhD, sending and hosting PhD students for a couple of months). Both Eric Féron and Behcet Açıkmeşe were PI of that project.

ERC STATOR[Conc] STATic analysis with ORiginal methods is an ERC project (2012–2017) led by David Monniaux that focuses on proposing new static analysis methods. This project uses a similar approaches than we do on relying on convex optimization to replace the classical Kleene iterative fixpoint computation of the abstract interpretation approach. However the target is different: the motivation is to compute simple linear invariants but to scale to large general programs, while, in FEANICSES projet we focus on complex non linear numerical properties as emerging in controllers.

ANR ARPEGE ASOPT[P,Comp] *Analyse Statique et OPTimisation / Static Analysis and Optimization* (2008–2012) was a project funded by ANR , whose aim was to improve the precision and the scalability of static analysis techniques using methods coming from the field of optimization. It included the development of new numerical abstract domains and fixpoint computation techniques, and their implementation into open-source libraries (Fixpoint[8], Interproc[9], and APRON[10]).

While I did not directly participated to this project, I collaborated with them during CAFEIN project or host a member of this project, Assalé Adjé for 18 months.

NASA CoCo[P,Comp] *CoCo project focuses on Safety Analysis of Flight Critical Systems* (2014–2017). This project between CMU and U. of Iowa developed specification means for dataflow languages as well as model-checking tools to verify these specification. It initiated as well the development of the open-source CoCoSim toolchain that will be used as the backbone on which we will integrate our analyses.

ANR INS VACSIM[P,Conc] was a project[11] (2011-2015) aiming at combining numerical simulation and formal methods to validate control-command software. This project approach is considered as concurrent to ours because its goal was similar but it considered different methods than we do, for example the use of tests and simulation to evaluate a system's properties.

2.3 Objectives, originality and novelty of the project

FEANICSES important challenge is the formal verification of system-level properties at all stages of a controlled system development.
Nowadays these properties are only evaluated at early stages when the controller is designed, and at the final stages, eg. when performing flight tests for an aircraft. Analyzing them formally and exhaustively all along the development chain will shorten the development process and leverage the quality of the final product, increasing the global safety of such systems.

To realize this ambitious and ground-breaking objective, I listed the following specific new concepts and new methods that the project will develop within its execution:

- Enable the **expression of system level properties**, including stability, robustness, performance as well as safety and reliability of the final system, **at all stages of its development**, including the final source code.
- Develop new methods to **address system properties verification** on simple to complex systems, **including combination with a fault tolerant architecture**.
- Develop the **CoCoSim toolchain integrating formal specification and formal verification** to support the development of critical embedded CPS while **analyzing system-level properties all along the chain**.
- The first attempt at **addressing all verification activities through formal methods applied to industrial examples**, including a Full Authority Digital Engine Control system (FADEC) for jet engines, already provided by our partner Price Induction, and an Unmanned Aerial Vehicle (UAV) avionics system, already provided by our partner NASA (see reference letters in Appendix).

[8] http://pop-art.inrialpes.fr/people/bjeannet/bjeannet-forge/fixpoint/index.html
[9] http://pop-art.inrialpes.fr/interproc/interprocweb.cgi
[10] http://apron.cri.ensmp.fr/library/
[11] http://vacsim.inria.fr/

The FEANICSES project will sustain a groundbreaking move from process based certification to product-based verification, providing the end-to-end formal verification of a system.

2.3.1 State of the Art and Outstanding scientific challenges.

FEANICSES objectives could be refined in terms of extensions of the State of the Art. The **ambition is to bridge the gap between** (i) **control level properties** expressed over controlled systems, (ii) **formal analyses** performed on models or code, and (iii) their implementation in a **usable toolchain**. Our project addresses these issues by targeting the following challenges:

C1: Express system level properties as numerical invariants. In current industry practices, most of the analyses used are based on frequency response analyses; analyzing the Fourier Transform of the system dynamics and evaluating its stability and robustness. Concerning performances properties, usual approaches amount to run a set of tests, illustrating the behavior of the controlled system for some inputs (eg. a step, or a ramp).

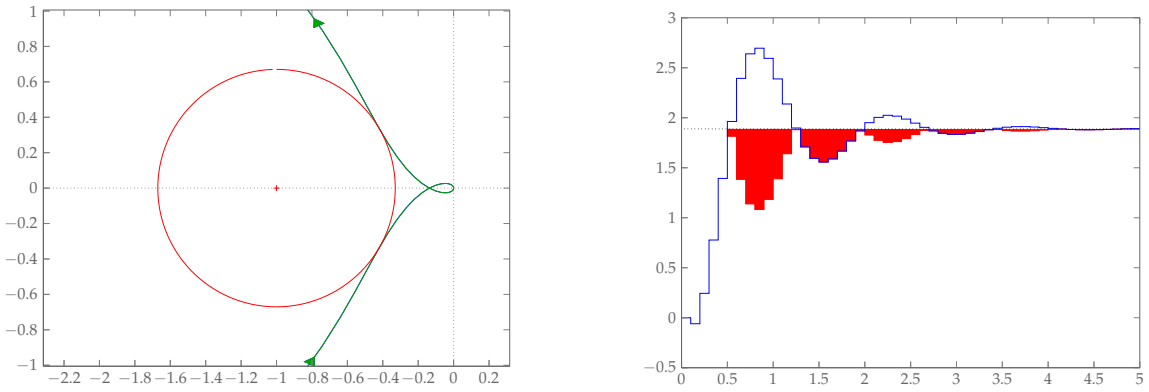
Motivated by the work of Féron [12], we proposed new automatic approaches [21, 22], based on Lyapunov functions principles, allowing to address the same properties using measures of the system energy. System-level properties can be then expressed as sub-levels of numerical functions.

Figure 3 represents two system properties that are not typically analyzed with Lyapunov functions but can be rephrased as such. The first one shows a vector margin computation, alternative to the classical phase and gain margins. The second one specifies the speed of convergence in terms of L^1-norm of the deviation between the signal and the command. Both can be computed using numerical optimization.

Today no existing method, applicable to realistic complete systems, performs the analysis of system-level properties such as performance in an exhaustive fashion.

As a general principle, our hypothesis here is to rephrase all system-level properties (stability, robustness, bounded overshoot, time to settle, etc) as a sublevel-set invariant, for example based on the H_∞ norm of a system or using Rantzer and Megretski's discrete Integral Quadratic Constraints (IQC) [18]. The first challenge of the FEANICSES project is to **express all system-level properties using numerical invariants**, allowing the computation of the latter through numerical optimization.

C2: Address the analysis of complete controllers in their safety environment. For critical devices, once the controller has been designed, it is embedded in a more complex system implementing a safety architecture: inputs are replicated and their value is consolidated; even the core controller can be replicated. On the control side, saturations or anti-windup constructs allow to avoid unexpected high values for variables. Another classical construct is to combine simpler (linear) controllers through linear interpolation. However the delay or inaccuracy introduced by these mechanisms will impact all high level properties mentioned earlier.



(a) Vector margin using Lyapunov functions.

(b) Expressing speed of convergence.

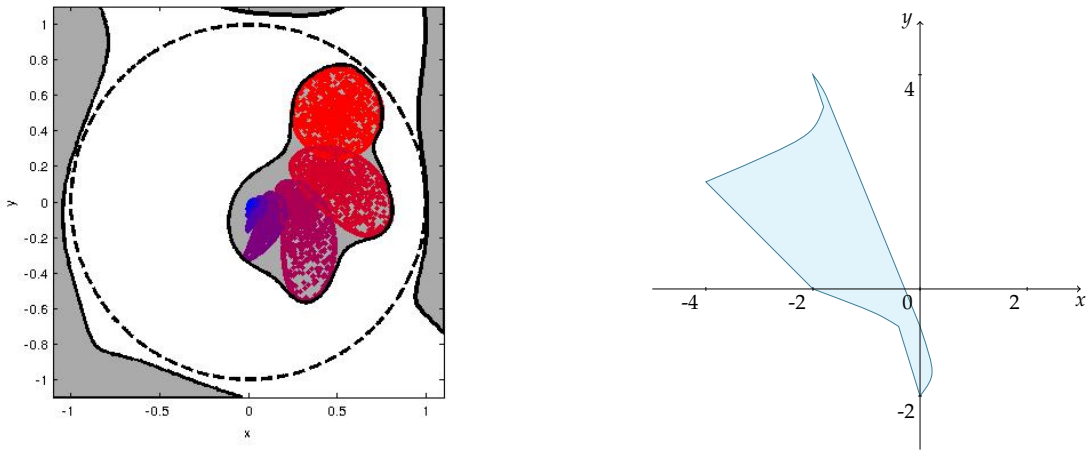
Figure 3: Expressing high level system properties as inductive numerical invariants.

However, when considering system level properties, their formal analysis in such environments is currently not feasible. **Today, no static analysis is able to address the high-level property analysis of a complete system combining control with fault-tolerant architecture.** As sketched in [8], I will address this important challenge by considering combination of numerical methods for the controller parts and both numerical and combinatorial methods for the safety architecture. This will enable the formal analysis of complete systems.

C3: Provide new means to synthesize non linear invariants. **Except for my work and a few others such as [4, 11], the state of the art of software formal analysis of code is mainly based on linear abstractions**: eg. intervals, polyhedra, or octagons. However the Lyapunov functions used to capture the behavior of controllers require at least quadratic invariants [2, 22]. It is therefore of the utmost importance to develop new methods able to synthesize non linear invariants.

Figure 4 presents two approaches we started to investigate: the use of Sum-of-Squares optimization to over-approximate reachable states of a non linear system [15]; and the iterative computation of inductive Parametric Quadratic Curves [3].

FEANICSES project will address the **design of new scalable automatic analyses capable of synthesizing non linear invariants**. In particular these analyses will rely on numerical optimization solvers (Sum-of-Squares, Semi-definite Programming) or on algebraic structures such as polynomial arithmetics [3] to compute such properties.



(a) Polynomial approximation of reachable states using numerical optimization.

(b) Parametric Quadratic Curves.

Figure 4: New means to compute non linear properties.

C4: Adapt formal analysis methods to handle floating point soundness. As mentioned above, the gap between control and formal verification communities is pregnant, especially with respect to floating point arithmetic. In the control community, floating point soundness is usually neglected and more or less covered by the robustness properties of the system. In the formal verification community, floating point arithmetic issues are tackled more often and in more depth (see e.g. [13]).

I will fill this gap by **developing static analyses of high level properties considering numerical imprecision due to floating point semantics**. In particular, I will **handle float-real mixed systems** where the variables of the software will be analyzed with a floating point semantics while the ones modeling the system dynamics will remain in the real field. Moreover analyzers results obtained with floating point computation will be checked.

C5: Develop a sound and scalable framework specific to the verification of controllers. A last yet major challenge is to provide practical tools to support the analysis of real systems. We will define such a sound and scalable framework, enabling the analysis of complete systems.

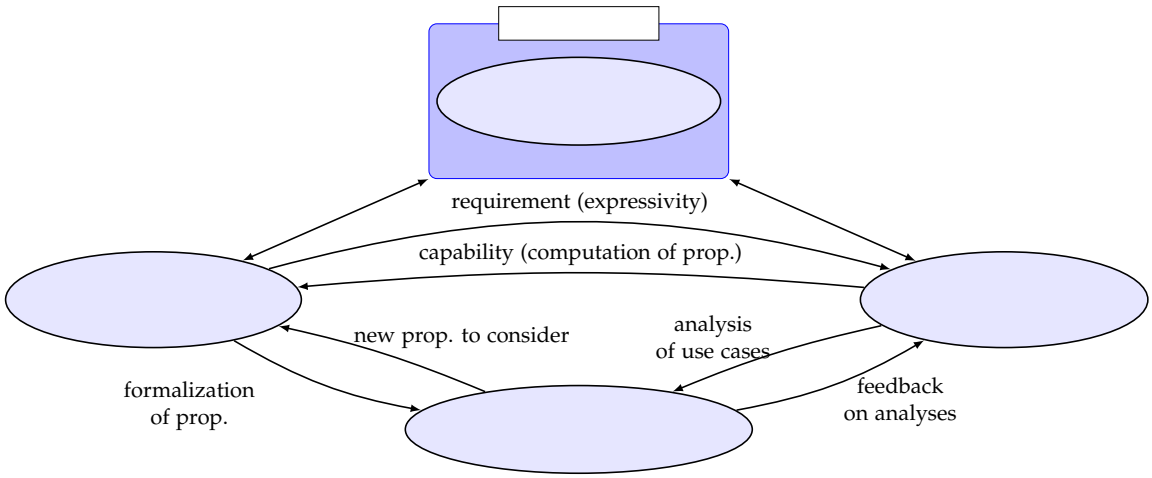


Figure 5: PERT: Relationship between WPs.

FEANICSES project will **release an innovative open-source toolchain** that computes these high level properties on a controlled system including its safety architecture. This framework will be **applicable to a wide range of targets including the new markets of UAVs and Cube/Mini Satellites**.

2.3.2 Evaluation of FEANICSES results

The evaluation of the project contributions will be based on challenge problems and benchmarks already available at Onera, Georgia Tech (Prof. Féron) and UW (Prof Açıkmeşe), as well as ones provided by NASA and our industrial partner Price Induction.

In particular, we will evaluate: 1. the applicability of the toolchain 2. the efficient and usefulness of synthesized invariants 3. measure the impact of floating point computation to the global systems.

Considered systems will include: 1. realistic control command systems for aircraft; 2. realistic attitude and orbital control systems; 3. FADEC (engine control for aircraft); 4. trajectory planning for pinpoint planetary landing; 5. collision avoidance systems.

3 Scientific and technical program, Project organization

The current sections presents the organization of the research activity in a project setting: tasks are identified with inputs, outputs, theoretical contributions and implementation, dependencies among tasks. Then we present the team participating to this young researcher (JCJC) project: internal participants and external collaborators. Last a justification of requested ressources is provided.

3.1 Scientific program, project structure

As mentioned in the previous section, the project is structured around the formal verification of properties of numerical controllers models and implementations. All work packages of the project will hence share this common description and formal semantics of the system under analysis, and share the closely related goals of either specifying properties or proving them on the formal description.

We detail here the four work packages that will drive the FEANICSES project. For each work package, we analyze separately the required theoretical contributions, implementation issues and experimental validation. We also describe the staff that will be in charge of each task. This information will be summarized in Section 3.3. Last we identify deliverables and release schedule as well as risks and mitigation.

Figure 5 presents the breakdown of the project in work packages and sketches the interactions or dependencies between WPs. An overview of the purpose and structure of each work package will be given in the following paragraphs.

3.1.1 Work Package overview.

WP1: High Level System Properties. This work-package addresses parts of the issues mentioned in challenges C1 and C2. It aims at formalizing high level system requirements in a form that supports formal analysis on the software level. A well known example is the closed and open-loop stability verification using quadratic Lyapunov functions.

WP2: Non Linear Analyses. Once the properties expressed as numerical invariants, we are looking for their automatic computation on a controlled system. It is obvious that linear properties would not be sufficient in general to characterize these properties. The goal of this work-package is to develop richer analyses able to manipulate the required constructs identified in the WP1. This will address the challenges C2 and C3.

WP3: Floating point soundness. While floating point soundness is common for the formal verification community, it seems to us that floating point soundness is not an explicit concern in the control community. It may seem reasonable in most cases, since the systems usually have robustness properties: they are resilient to floating point rounding errors, supposedly negligible with respect to sensor errors for example. However when it comes to proving properties about software implementing these systems, the use of floating point arithmetic cannot be overlooked. This WP will address challenge C4 while focusing on C1 and C3.

WP4: Applications and use cases. This WP is major to the project success and impact. It will summarize the implementation effort and produce a usable toolchain addressing challenge C5.

3.1.2 Work Packages interactions

The Figure 5 illustrates the interactions between work packages. The backbone of the project resides between WP1 and WP2: WP1 has to express requirements that WP2 has to satisfy. In a reciprocal manner, analysis capabilities in WP2 may open new approaches to formalize properties in WP1. Once WP1 and WP2 produces early results, their floating point issues will be explored in WP3. Last WP4 enable the integration of the proposed approaches in a unified toolchain applicable on realistic examples. WP4 will also inject feedback on intermediate results and propose challenges and research directions depending on the difficulties encountered during experiments.

3.2 Project organization.

WP1: High Level System Properties (challenges C1, C2)

Theory. Expressing system level properties often requires to express the system semantics, also known as the plant semantics. Multiple choices are possible: the use of ODEs (Ordinary Differential Equations), a discretized version of it, or even a linearization of the discrete semantics around some well chosen points of the state space.

The classical properties capturing the behavior of a controlled system are 1. stability; 2. robustness; and 3. performances. While it is now common to rely on Lyapunov functions, ie. functions defined over the system state, the robustness is systematically analyzed by considering the Fourier transform of the system signals. These frequency response based analyses are not suitable for their evaluation at code level. Finally when it comes to performances, the only analyses available consist in performing a set of tests and observing the obtained signal results. While it may be understandable for purely linear systems, it is not trustable once the considered system is combined with saturations, interpolations, or redundancy patterns that makes it non linear. **We will rephrase all kinds of system level properties as numerical predicates over the system states**. This would first **enable their analysis at the code level** and second **deliver exhaustive results** that test cannot provide.

Implementation. The work package will start at the beginning of the project and last the first three years. While this task incurs high risks, it is extremely important to obtain results. In [21, 22], we have already addressed stability analysis for discrete open and closed loop systems. As illustrated in Figure 3a, we are able to compute the vector margins of a linear system using the computation of a Lyapunov function as the solution of a Linear Matrix Inequality (LMI) using

a Semi Definite Programming (SDP) solver[12]. **In terms of performances, classical notions of overshoot, rising time, settling time, could be theoretically exhaustively analyzed and bounded on reachable states.**

In order to obtain results while mitigating the high risk/high value aspect, the **WP implementation will progress in two orthogonal directions:** 1. **the semantics used to describe the system** – from linear discrete to ODE, and 2. **the controller – with increased complexity**: linear, saturation, piece-wise linear, interpolated, non linear.

In the latest stages of the WP, we also plan to evaluate other kinds of properties which are less common in the industry practice. Following techniques should be able to address discrete systems, continuous ones, and hybrid ones.

1. Contraction, developed by J.J. Slotine [17] is a kind of stronger stability property with compositional feature. It has been developed in a large number of settings but never evaluated on large realistic control systems software, ie. with a discrete part; 2. Viability theory, developed by J.P. Aubin [5] is a theoretical approach to the computation of reachable states, named viability kernel. While most works are highly theoretical and hardly computable, we have some insight to perform viability kernel computation for restricted classes of programs and systems (see e.g. [10]).

Work package results will consist in new characterizations of system level properties that will fuel all the other work-packages.

Experimental evaluation. The expression of these high level properties as system state functions, will be evaluated on existing controllers. At Onera, through past and current collaborations, we built and have access to a set of examples ranging from prototypes or toy examples – eg. controllers with a few state variables like a spring mass damper, a 3-DOF helicopter, inverse pendulums, typical safety constructs – to industry level systems, with hundreds or thousands of blocks – like the complete specification of a PriceInduction aircraft turbofan engine FADEC (Full Authority Digital Engine Control), a NASA open-source example of a Transport Class Model (TCM) Aircraft Simulation, or projections of control system of large aircrafts, provided by our industrial partners.

The evaluation will compare classical methods with the ones developed in the FEANICSES projects, addressing first the simpler systems. The goal being to compute meaningful results on large examples.

Organization. This important work-package will be crucial for the project success. Therefore we will allocate an important part of the staff to it. Two existing PhD students (G. Davy and R. Cohen) co-advised with E. Féron and D. Henrion, respectively, are already assigned to it, while a third PhD thesis (PhD1), co-advised with A. Chapoutot, will be half funded by the project. This work will also benefit from contributions by invited researchers during the organized workshops.

Deliverables.

D1.1	Stability and robustness for interpolated controllers	T_0 +12
D1.2	Performance and convergence properties formalization	T_0 +24
D1.3	System-level properties formalization	T_0 +36

Risks/Mitigation. Failing in being able to express high-level properties in a formal way may impact the success of the project. Stability using Lyapunov functions is classical and we proposed in [23] an encoding of robustness for linear systems using H_∞ norm. This notion should be extensible to more general settings. Regarding performance the definition of these notions in a formal way is not obvious. The IQC approach [18] seems promising. However, in case of failure in finding an acceptable formal and general definition of these, we will define our own and build upon that. In that case the acceptance by the control community as a viable alternative will be reduced.

WP2: Non Linear Analyses (challenges C2, C3)

Theory. In this WP, we focus more specifically on the computation of semi-algebraic constraints,

[12]It amounts to maximizing the possible perturbations while keeping the system stable.

that is a property defined as a finite union of finite conjunctions of polynomial inequalities: $\bigvee \bigwedge (p_i(x) \leq 0)$ where x denotes a system state.

We will focus on two different approaches. A first one consists in manipulating, iteratively, these properties while analyzing the program. This is typically performed while **computing a Kleene fixpoint in the abstract interpretation setting**. A second one amounts to first **synthesize an appropriate set of polynomials** $p_i(\widetilde{x})$ - the templates. Then, in a second phase find the smallest bounds b_i such that $\bigwedge p_i(\widetilde{x}) \leq b_i$ is an inductive invariant. This phase is known as policy iteration or strategy iteration. It relies on the **use of numerical optimization to precisely bound these templates**.

Implementation. The WP implementation will be divided in the following sub-tasks:

A. Iterative methods. In [3], we adapted, to the static analysis setting, the quadratic extension of affine arithmetic that was proposed by Messine [19] in a global optimization context. It allows to iteratively compute a non convex subset of $\mathbb{R}^n$, using the same approach as the zonotopic domain [14] in the affine setting. Fig. 4b represents such a quadratic invariant. **We will further develop the use of a polynomial extension of affine arithmetic to iteratively and algebraically synthesize non linear and non convex properties**. We will specifically **evaluate the usability of this approach for controller analysis**.

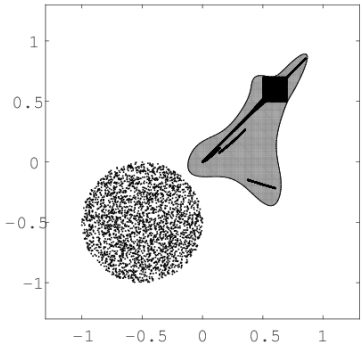


Figure 6: Property-directed template

B. Non linear template synthesis. This task will investigate multiple directions. In [2], we already proposed a piece-wise construction of templates. It would be meaningful to **combine this synthesis with property-directed templates**, eg. see Figure 6. Another approach, based on the computation of the controllability region of a dynamical system, **solving an optimization problem over Borel measures [15], will be defined to address the computation of reachable states.**

C. Policy iteration extension. This method [4] is still confidential in the formal methods community and requires work to be more applicable. Among the extensions that will be considered in this project, we want to extend the set of systems analyzable as well as improve the use of templates, eg. the **dynamic injection of templates** during the computation and the use of *disjunctive templates*.

D. Synthesizing invariants from Set-based simulation. In [16] the authors presented a method allowing to use simulation data to drive the computation of a template-based invariant. We will extend this approach by considering set-based simulation [7]: exhaustive test for finite traces based on zonotopic abstractions.

More generally, **we plan to develop new non linear analyses and study their expressiveness, and applicability, in the light of the considered systems.**

Experimental evaluation. At the early stages of the FEANICSES project, the first evaluation will consist in analyzing the wide variety of systems mentioned in WP1 and computing their reachable states.

Once richer properties will be characterized by WP1, we will experiment their automatic computation using property-driven templates.

This WP is also a high risk/high gain one since its success could widely impact the verification community providing more expressive analyses. While the definition of these new analyses may remain solely theoretical, the real measure of success is the applicability of the proposed analyses on real systems. It would require the identification of a very restricted subset of variables for the program or state variables for the system that will be specifically analyzed with these costly analyses.

Organization. As for WP1, the criticality of WP2 is major for the project success. We will allocate both PhD2 on the topic, co-advised with B. Açıkmeşe as well as the 18 months Postdoc co-advised with E. Féron and D. Henrion.

Deliverables.

D2.1	Non linear template synthesis	T_0 +12
D2.2	Disjunctive analyses	T_0 +24
D2.3	Property-based analyses	T_0 +36
D2.4	Non linear invariants	T_0 +48

Risks/Mitigation. Since 2012 we have produced numerous papers [1, 2, 3, 20, 22] proposing non linear analyses. All of them are implemented into prototypes. The main risk here lies in their applicability on larger programs. This risk can be mitigated be considering iteratively larger components or splitting the considered examples in sub-blocks.

WP3: Floating point soundness (challenges C1, C3, C4)

Theory. Floating point arithmetic appears both in the program we are analyzing – controller software implemented with floats – and in the tools we are using to analyze the system. This WP will revisit WP1 and WP2 solutions to address their floating point soundness. There will be no new contributions with respect to floating point soundness itself, but **the theoretical contribution will lie in the sound versions of the considered algorithms and properties.**

Implementation. The activities in this WP will cover these two aspects:

A. Analyzing floating point arithmetic. Both the properties formalization characterized in WP1 and the methods developed in WP2 will have to be extended to manipulate floating point semantics. We will specifically address the following issues:

A first approach will model floating point errors using either additional error terms, or representing values with safe intervals. In both case, it amounts to **instrument the analysis with an explicit representation of floating point errors**. This is typically the approach chosen in [9, 13].

A second line of work consists in performing the usual over-approximation using variables with real semantics. Then, once the result obtained, perform an ***a posteriori* checking of the floating point soundness** which is expected to work when errors due to floating point rounding are orders of magnitude smaller than the over-approximation previously performed. For example, in [22] we checked the semi-positive definiteness using an interval-arithmetic sound implementation of a Cholesky decomposition algorithm.

In this task **we will evaluate both choices for each property and verification method proposed, in order to identify the most appropriate one in terms of precision and scalability**. The floating point extension of these should also deal with mixed settings where part of the system description – the plant – has real semantics while another part – the controller software – has floating point semantics.

B. Working with floating point arithmetic. Another important issue is to be able to trust analysis tools despite their use of floating point arithmetic. An important part of WP2 methods will rely on numerical optimization solvers such as linear programming (LP) solvers, semidefinite programming (SDP) solvers or their Sum of Squares (SOS) extension[13]. In the last cases, the implementation typically relies on a primal-dual interior point method. These tools, while extremely efficient, may produce a solution which is near the optimal but without being a feasible solution: it does not satisfy the constraints of the solved problem. **This task will focus on the definition of a sub-optimal yet efficient algorithm, based on the interior-point method, but computing a sound and feasible solution.**

Experimental evaluation. As mentioned above, the techniques proposed will be evaluated on WP1 and WP2 results. Since floating point theorems could be easily error-prone, WP3 solutions will, as much as possible, be formalized in proof assistants like Coq.

Organization. This WP will start at Y2, once early results of WP1 and WP2 will be made available. Pierre Roux, a permanent researcher of Onera will mainly participate to this WP. During his postdoc

[13] Technically the relationship between SOS and SDP can be seen both as SOS included in the SDP code with additional equality constraints, and SDP as the SOS cone constrained to polynomials of degree at most 2.

at LRI/INRIA Saclay, he proved in Coq all his PhD results with respect to floating point soundness. Both existing PhD students (G. Davy and R. Cohen) will be specifically allocated on task B targeting sound implementation of the interior point method in the SDP cone, and analysis of the ellipsoid method, respectively. PhD1 co-advised with A. Chapoutot will develop dedicated static analyses.

Deliverables.

D3.1	Floating-point variants of system-level properties	T_0 +24
D3.2	Precise over-approximation of non linear floating point computations	T_0 +36
D3.3	Enhancing numerical precision	T_0 +48

Risks/Mitigation. Floating point analysis do not present any specific risks. The issue is mainly about precision. In case of difficulties on the proof process, a proved approach based on Coq could be replaced by static analyses, bounding the rounding errors on a specific example.

WP4: Applications and use cases (challenge C5) **Theory.** This WP does not involve any theoretical contribution. It will however propose a new development process based on other WPs theoretical contributions.

Implementation. The implementation tasks will consist in the development of an integrated toolchain in which the controlled system considered could be fully developed and verified. This toolchain will combine existing open-source compilers we have developed, based on the modular compilation scheme of [6], with the formal verification methods proposed in WP2.

The FEANICSES project demonstrator will be capable of analyzing and compiling a controller defined in a language such as Matlab Simulink, and produce a verified C code. It will integrate seamlessly the expression of the closed-loop control properties, the expression of the system dynamics, as well as the definition of the safety architecture (See challenge C2).

While each WP will produce tools and methods since their beginning, their integration as a global toolchain in the WP4 will start at Y3. All proposed tools will be open-source and applicable to realistic systems.

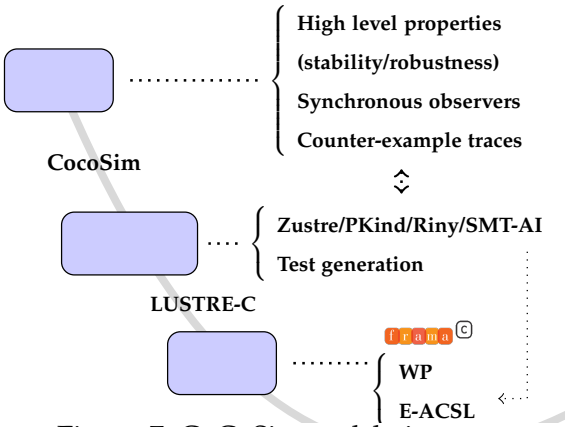


Figure 7: CoCoSim toolchain

Since 2012, we developed with NASA Ames and INPT/IRIT a toolchain capable of compiling Simulink models into C code while specifying their expected behavior. This toolchain is, for the moment, focused on combinatorics properties, as expressed in challenge C2, and, thanks to both its efficiency and its open-source licence, has attracted interest of major industrials such as General Electric or Lockheed Martin. We plan to extend this toolchain and integrate more advanced properties in both the specification language and the associated analyzers. Figure 7 presents the activities of WP2.

Experimental evaluation. The key evaluation of this WP is the capability for the developed toolchain to address the analysis of representative examples. As mentioned in the introduction, we will apply the demonstrator to the PriceInduction FADEC system as well as the NASA TCM controller. We will also use our academic and industrial relationships to obtain access to representative examples of UAV and Cube/Micro satellite controllers, planetary landing algorithms, and NextGen collision avoidance systems (ACAS-X)..

Organization. This WP will be mainly supported by the PI and the other permanent staff. However PhD students and Postdocs involved in the project will participate to the integration of their contributions in this demonstrator. The specific commitment of PhD2, co-advised with B. Açıkmeşe,

will apply the framework to UAV trajectory planning. Similarly, existing PhD student R. Cohen will focus on Price Induction FADEC.

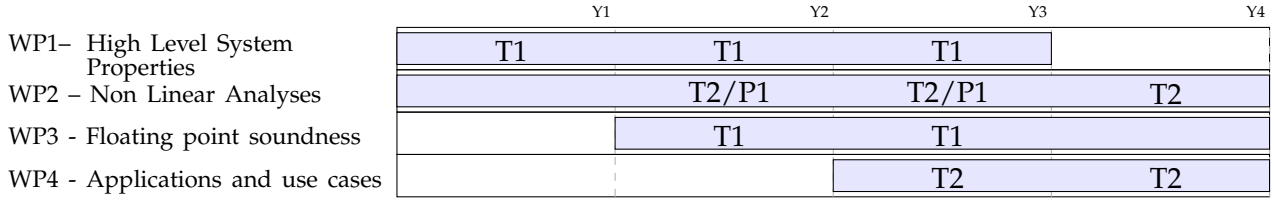
Deliverables.

D4.1	Integrated toolchain (first results)	T_0 +36
D4.2	Uses cases evaluation (first results)	T_0 +36
D4.3	Integrated toolchain (consolidation)	T_0 +48
D4.4	Uses cases evaluation (consolidation)	T_0 +48

Risks/Mitigation. Main risk is the lack of applicability of the toolchain to considered examples. We can mitigate these risks by considering multiple yet valuable examples in order to show that the approach support the verification of some of them. We have good preliminary results on regular controllers and the NASA controller has been partially analyzed with an early version of the toolchain. Anti-collision systems are more prospective.

3.2.1 Scheduling of work packages.

The diagram of Fig. 8 presents the Gantt chart of the project. WP1 and WP2 will start at Y0 and last 3 and 4 years. WP3 and WP4 will start at Y1 and last 3 years. Last, WP5 will start at Y2 and last 2 years.



Tx stands for PhDx and Px for PostDocx.

Figure 8: Scheduling of work packages.

3.3 Involved resources.

Resources will involve both local support from Onera to provide 60 man-month (estimated cost of about 500 k€) shared between Onera participants (85% for the PI, 20% for P. Roux, R. Delmas, each), support for the four associated partners (20% each, ie. 38 man-month) and ANR funding to support 2 PhD (half-funded), 18-months of Postdoc as well as workshop organization and missions for project meetings between partners and attending conferences. We obtained official commitments from external partners, Alexandre Chapoutot, Didier Henrion, Behcet Açımeşe and Eric Féron to be associated with this FEANICSES project proposal. PhD1 will start at T_0 and be half funded by Onera. It will focus on analyzing system level behavior (WP1) and numerical imprecision (WP3). PhD2 will start at $T_0 + 12$, half funded by Univ. of Washington, and be focused on non linear analyses (WP2) and application on UAV controller. A 18 months postdoc will also start at $T_0 + 12$, focused on non linear analysis (WP2). The mentoring of 2 PhD students and the postdoc will involve all participants of the project, both internal and external ones.

The project also involve already funded PhD students Guillaume Davy and Raphaël Cohen as mentioned in Section 1.2. This totalizes 226 man.months for the project duration. A specific funding (4×10k€)is requested to support the organization and the invitation of participants to attend yearly workshops in Toulouse. The approximate funding request is 300 k€ for 48 months. It is split as such: 2 half-funded PhD (58k each), 18 months of Postdoc (76k), missions for all partners to attend conferences, and project meeting (35k, about 154€ per man.month) and organize the workshops (40k), budget to acquire devices on which to illustrate the analyses (7k: 1 UAV and 1 robot, laptops for students), and Onera computing resources (4k).

D1.1	Stability and robustness for interpolated controllers	T_0 +12
D2.1	Non linear template synthesis	T_0 +12
D1.2	Performance and convergence properties formalization	T_0 +24
D2.2	Disjunctive analyses	T_0 +24
D3.1	Floating-point variants of system-level properties	T_0 +24
D4.1	System-level properties as probabilistic measures over reachable states	T_0 +24
D1.3	System-level properties formalization	T_0 +36
D2.3	Property-based analyses	T_0 +36
D3.2	Precise over-approximation of non linear floating point computations	T_0 +36
D4.2	Sensitivity value-based analysis	T_0 +36
D5.1	Integrated toolchain (first results)	T_0 +36
D5.2	Uses cases evaluation (first results)	T_0 +36
D2.4	Non linear invariants	T_0 +48
D3.3	Enhancing numerical precision	T_0 +48
D4.3	Formal analysis of probabilistic properties for control systems	T_0 +48
D5.3	Integrated toolchain (consolidation)	T_0 +48
D5.4	Uses cases evaluation (consolidation)	T_0 +48

Table 1: Summary of deliverables

3.4 Risk Evaluation, Expected Impact and Interdisciplinary nature of the project.

Classical risks analysis first evaluates what is required to perform the planned tasks. The basic risk inherent to most research projects is the access to input data: it is here mitigated since we already have small to large examples on which to apply our analyses.The main risks here lie in the feasibility of the proposed research and the scientific choices to address it. Through my early experiments and my recent works [2, 21, 22], **I have shown the feasibility of the approach** for open and closed loop stability, considering floating point semantics and analyzing code. In terms of methodology to reach the objectives, **energy-based analyses, *à la* Lyapunov, using semi-algebraic sets, appear to be the only solution to express and analyze system-level properties at code level**. During the project course, while characterizing new properties and proposing new non linear static analyses, we will evaluate our solutions incrementally, considering more complex properties and bigger, more realistic controlled systems.

Regarding industrial impact, the CoCoSim toolchain we developed with NASA Ames is currently considered by General Electric [14] and the integration of open-source analyzer integrated in the toolchain will ease their use. I am confident that **FEANICSES approach is applicable, the expected results reachable, and will impact the industry practices widely**.

Analyzing system-level properties on complete systems combining multiple controllers, non linearities and fault-tolerance architecture is an extremely high risk/high pioneering objective. Showing the feasibility of the approach, supported by tools will **widely impact the industry practices** (aircrafts, UAVs, MiniSat, medical devices, etc).
The FEANICSES project will **extended CoCoSim, a new integrated development process** for critical controlled system, **based on formal verification** and semantics preservation. It will **expand the knowledge and practices** of system-level property analysis at model and code level, **providing a backbone for next-generation process development of critical CPS, with highly integrated formal methods**.

The groundbreaking nature of the proposed framework have also sparked the interest of aerospace industrials seeing an opportunity to leverage the properties analyzed while minimiz-

[14] https://ti.arc.nasa.gov/RSE-Develops-Tools-With-GE/

ing their V&V costs. In terms of impact, the support of ANR to this project could really develop and sustain this major interdisciplinary topic. For the moment, the USA, through various State Agencies (NSF, NASA, AFOSR) has shown its interest for the subject, **expecting a large market growth for these controlled systems**.

I believe that this JCJC grant could greatly contribute to move the barycenter of this emerging interdisciplinary community to France. While Europe and France are still the leaders in terms of formal methods and have strong teams in control design, this subject is not yet clearly addressed by a specific team.

It is important for France to have a team focused on formal verification of numerical controlled systems implementation, a team I intend to create within the first two years of the FEANICSES project.

4 Impact, Dissemination and exploitation of results

Dissemination of FEANICSES results will mainly be achieved through usual academic channels. But they will also impact the maturity level of existing open-source software as well as new software and prototypes created during the project. We present here the possible mid and long term impact of FEANICSES project.

FEANICSES results will contribute to Défi 7 / Axe 3 of the ANR Plan d'Action: combining design tools and methodology with formal methods to validate software and systems as early as possible in the development cycle. Moreover, focusing on control command system and advanced numerical algorithms for critical embedded systems, the outcome of the project also apply to the Défi 7 / Axe 4 on robotics.

4.1 Autonomy – Group development

One of the goal of JCJC projects is to sustain the development of a research group and community around the coordinator, providing him/her scientific autonomy.

Thanks to the organization of the yearly meeting, the strong collaboration with participants of the projects at LAAS CNRS, ENSTA Paritech, Univ. of Washington and Georgia Tech, the support of NASA and French SME Price Induction, as well as the invitation of field experts along the project duration, I believe that a strong team will be created, with numerous collaborations. In addition, the co-advising of 2 PhD students and postdoc funded by the project will also sustain that goal.

4.2 Academic dissemination

The usual academic dissemination will be done through publications, such as journal papers, conferences proceedings and presentations at national and international levels.

We plan to organize a dedicated workshop that addresses the FEANICSES project scientific challenge. This workshop will be hosted yearly in Toulouse, involving world wide experts of the field, invited on FEANICSES budget.

FEANICSES project results will also be presented to other projects or related research groups abroad. In addition to the collaborators mentioned in the proposal, we will especially communicate with the following groups:

- Prof. Ilya Kolmanovsky (design and proof of extended reference governor controllers), University of Michigan (MI, USA);
- Prof. John Hauser (non linear control), University of Boulder (CO, USA);
- Prof. Anders Rantzer (Integral Quadratic Constraints and system level properties analysis), Lunds Universitet (Sweden);
- Prof. Cesare Tinelli (founder and leader of the SMT-lib project), Iowa University, (IA, USA).

4.3 Open-source Strategy and Dissemination

The different work packages of the FEANICSES projects are based on several existing static analysis tools. Some of these are FLOSS (free/libre and open source software) such as kind (developed by Cesare Tinelli at Univ. of Iowa), OSDP, the LustreC toolchain, CoCoSim (developed with NASA Ames/ CMU). These different software, prototypes and library will directly take benefit from the FEANICSES project results both in terms of functionality and robustness.

Furthermore, these tools will be integrated in a more general toolchain, as presented in challenge C5.

4.4 Long and middle term impact

Certification norms and authorities FEANICSES project results will be transferred as technical solutions proposals compliant with the latest certification norms. These norms, such as the DO178C, advocate for a wider use of formal methods to validate software and do not just focus on validation by test anymore. They also push forward verification means early in the development cycle, in particular at model level.

The impact of the project results will then be widely applicable to industrial contexts, such as civil aircraft, or aerospace, which all rely on similar kinds of modeling frameworks for reactive systems.

Another dissemination channel will target directly certification authorities. Through its involvement in the DO process and its relationship with DGA-TA (DGA Techniques aéronautiques, ex CEAT), the French authorities for civilian aircraft certification, Onera will also present these new techniques to certification authorities; for example through a sequence of seminars. These presentations will expose the use, the applicability and validity of these formal approaches of verification at model level.

Industrial impact As a young research project, with only a single partner, as required by the call, the relationship with industrial is not required. As presented in the support letters in the Annex, the properties analyzed and the methods developed could largely benefit to the certification processes of both PriceInduction and future planetary landing missions as developed by Univ. of Washington.

Economic impact A more long term impact of the project results will be the development of new expertise linked to the combination of analyses. With the evolution of the norms and the scalability of the formal approaches, especially through FEANICSES results, a new service activity can emerge at mid or long term, as a support of the verification and validation phases of software development.

FEANICSES Project impact and outcome

- With respect to the ANR call and more specifically the "Défi 7/Axe 3 and Axe 4": the **extension of the state of the art on basic research for non linear analyses**, the **application of proposed methods on realistic aerospace examples** and the **development of a demonstrator** to illustrate the applicability of the approach. All those elements will sustain the impact of the FEANICSES project at the society level, **addressing major identified goals**.
- FEANICSES project, through both the support of co-supervision of PhD students and postdoc, and with the funding to organize workshops and seminars on this topic, will **support the existence of a group focused on the formal analysis of numerical intensive control software**.
- Project results will be presented in **major conferences** of the field from academic conference (SAS, NFM, HSCC, ACC/ECC) to industrial ones (ERTS, DASC, SAE AeroTech). We will also organize a **yearly event in Toulouse** gathering world class researchers from both the formal methods community and the control systems design. We will also **write a textbook** addressing the issues behind the development of critical CPS and the use of formal methods.
- Project outputs will be **CoCoSim, an open-source demonstrator as well as examples** easing collaborations. Two **French SMEs**, **Price Induction** – an aircraft engine manufacturer – and

Numalis – focused on numerical accuracy – will be privileged targets to **transfer FEANICSES project results**. Furthermore the techniques developed could **support the certification of the software of a next pinpoint mission to Mars**, thanks to the collaboration with Prof. Behçet Açıkmeşe.

5 References.

[1] A. Adjé, P. Garoche, and V. Magron. "Property-based Polynomial Invariant Generation Using Sums-of-Squares Optimization". In: *SAS 2015, Saint-Malo*. 2015, pp. 235–251.

[2] A. Adjé and P.-L. Garoche. "Automatic Synthesis of Piecewise Linear Quadratic Invariants for Programs". In: *VMCAI 2015, Mumbai, India*. 2015, pp. 99–116.

[3] A. Adjé, P.-L. Garoche, and A. Werey. "Quadratic Zonotopes: An extension of Zonotopes to Quadratic Arithmetics". In: *APLAS'15, Pohang, Korea*. 2015.

[4] A. Adjé, S. Gaubert, and E. Goubault. "Coupling policy iteration with semi-definite relaxation to compute accurate numerical invariants in static analysis". In: *Logical Methods in Computer Science* 8.1 (2012).

[5] J. Aubin. *Viability Theory*. Modern Birkhäuser Classics. Birkhäuser Boston, 2009.

[6] D. Biernacki, J. Colaço, G. Hamon, and M. Pouzet. "Clock-directed modular code generation for synchronous data-flow languages". In: *LCTES'08, Tucson, AZ, USA*. 2008, pp. 121–130.

[7] O. Bouissou, A. Chapoutot, and A. Djoudi. "Enclosing Temporal Evolution of Dynamical Systems Using Numerical Methods". In: *NFM 2013, Moffett Field, CA, USA, May 14-16, 2013. Proceedings*. 2013, pp. 108–123.

[8] A. Champion, R. Delmas, M. Dierkes, P.-L. Garoche, R. Jobredeaux, and P. Roux. "Formal Methods for the Analysis of Critical Control Systems Models: Combining Non-Linear and Linear Analyses". In: *SAE International Journal of Aerospace* 6.1 (2013), pp. 150–160.

[9] P. Cousot, R. Cousot, J. Feret, L. Mauborgne, A. Miné, D. Monniaux, and X. Rival. "The ASTREÉ Analyzer". In: *ESOP 2005, part of ETAPS 2005, Edinburgh, UK, April 4-8, 2005, Proceedings*. 2005, pp. 21–30.

[10] E. Crück. "Problèmes de cible sous contraintes d'état pour des systèmes non linéaires avec sauts d'état". In: *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics* 333.5 (2001), pp. 403 –408.

[11] J. Feret. "Static Analysis of Digital Filters". In: *ESOP 2004, Barcelona, Spain, March 29 - April 2, 2004, Proceedings*. 2004, pp. 33–48.

[12] E. Feron. "From Control Systems to Control Software". In: *IEEE Control Systems Magazine* 30.6 (Dec. 2010), pp. 50–71.

[13] E. Goubault. "Static Analysis by Abstract Interpretation of Numerical Programs and Systems, and FLUCTUAT". In: *SAS 2013, Seattle, WA, USA*. 2013, pp. 1–3.

[14] E. Goubault, S. Putot, and F. Védrine. "Modular Static Analysis with Zonotopes". In: *SAS 2012, Deauville, France, September 11-13, 2012. Proceedings*. 2012, pp. 24–40.

[15] D. Henrion and M. Korda. "Convex Computation of the Region of Attraction of Polynomial Control Systems". In: *IEEE Transactions on Automatic Control* 59.2 (2014), pp. 297–312.

[16] J. Kapinski, J. V. Deshmukh, S. Sankaranarayanan, and N. Arechiga. "Simulation-guided lyapunov analysis for hybrid dynamical systems". In: *HSCC'14, Berlin, Germany, April 15-17, 2014*. 2014, pp. 133–142.

[17] W. Lohmiller and J.-J. Slotine. "On Contraction Analysis for Non-linear Systems". In: *Automatica* 34.6 (1998), pp. 683 –696.

[18] A. Megretski and A. Rantzer. "System analysis via integral quadratic constraints". In: *IEEE Transactions on Automatic Control* 42.6 (1997), pp. 819–830.

[19] F. Messine and A. Touhami. "A General Reliable Quadratic Form: An Extension of Affine Arithmetic". In: *Reliable Computing* 12.3 (2006), pp. 171–192.

[20] P. Roux and P.-L. Garoche. "Computing Quadratic Invariants with Min- and Max-Policy Iterations: A Practical Comparison". In: *FM 2014: Formal Methods - 19th International Symposium, Singapore, May 12-16, 2014. Proceedings*. 2014, pp. 563–578.

[21] P. Roux, R. Jobredeaux, and P.-L. Garoche. "Closed Loop Analysis of Control Command Software". In: *HSCC'15, Seattle, Washington, USA*. 2015.

[22] P. Roux, R. Jobredeaux, P.-L. Garoche, and E. Feron. "A generic ellipsoid abstract domain for linear time invariant systems". In: *HSCC'12, Beijing, China*. 2012, pp. 105–114.

[23] T. Wang, P.-L. Garoche, P. Roux, R. Jobredeaux, and E. Feron. "Formal Analysis of Robustness at Model and Code Level". In: *HSCC'16, Vienna, Austria, to appear*. 2016.