UNIVERSIDADE FEDERAL DE SANTA CATARINA

MANCHESTER 1824 The University of Manchester

# Safe UAV Continuous-Control Architecture Design

**Leandro Buss Becker**[*,¥], Fernando Silvano Gonçalves[#], Elton Ferreira Broering[*], Henrique Amaral Misson[*], Lucas Cordeiro[¥]

[*] Federal University of Santa Catarina – Florianópolis - Brazil

[*] Federal University of Santa Catarina Federal Institute – Tubarão - Brazil

[¥] The University of Manchester - UK

3rd ADEPT Workshop

June / 2024

PROVANT UFSC UFMG

# Outline

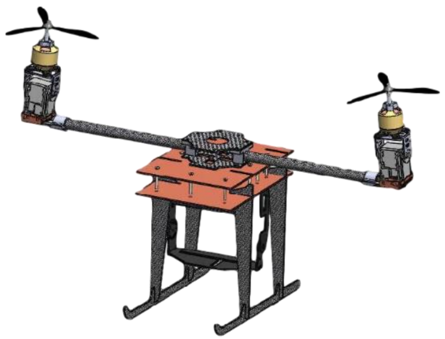
- Introduction: problem and related tools
- Design Method Activities & Artifacts
- Design of UAV's continuous control architecture
- Conclusions and Future Works

# Outline

- **Introduction: problem and related tools**
- Design Method Activities & Artifacts
- Design of UAV's continuous control architecture
- Conclusions and Future Works

# Projeto VANT - ProVANT

- Started in 2012 at UFSC; now partnership with UFMG + Seville
- **Objective:** Design autonomous UAVs as research platform
- Research topics include:
  - Critical Embedded Systems;
  - Wireless Communication;
  - Artificial Inteligence;
  - Control Systems.


UAV 1.0


UAV 2.0


UAV 3.0


UAV 4.0

# UAV Design Process



- Multidisciplinary process involving different teams
- Adequate methods and Tools are of utmost importance to support and guide the design process
- Due to the system's complexity, strong analysis are required
- MDE techniques are envisioned
- Formal verification becomes a natural candidate to be part this process

# Paper aim: Formally verify the design with distinct MC tools



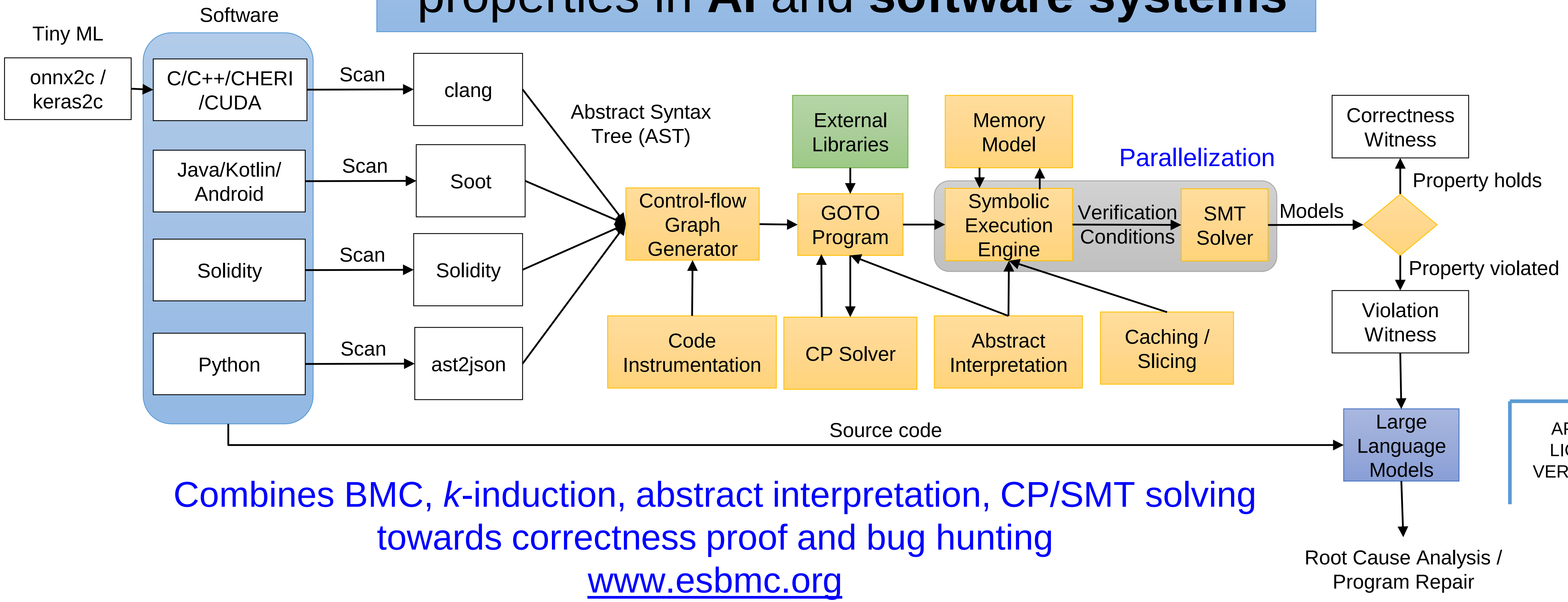
- Integrate the **Model Checking** technique on the **UAV design process**
- Utilize a **model transformation** process from **AADL** to generate Timed automata to the **UPPAAL** Tool
  - Allow the **timing** properties evaluation
- Utilize **bounded model checking** to verify correctness of the **C source code**

# ESBMC: Logic-based Verification Platf.

Universidade Federal de Santa Catarina

MANCHESTER 1824 The University of Manchester

**Logic-based automated verification** for checking **safety** and **liveness** properties in **AI** and **software systems**




APACHE LICENSE VERSION 2.0

Combines BMC, *k*-induction, abstract interpretation, CP/SMT solving towards correctness proof and bug hunting

www.esbmc.org

Universidade Federal de Santa Catarina

- Env
- Co






UFSC UFMG

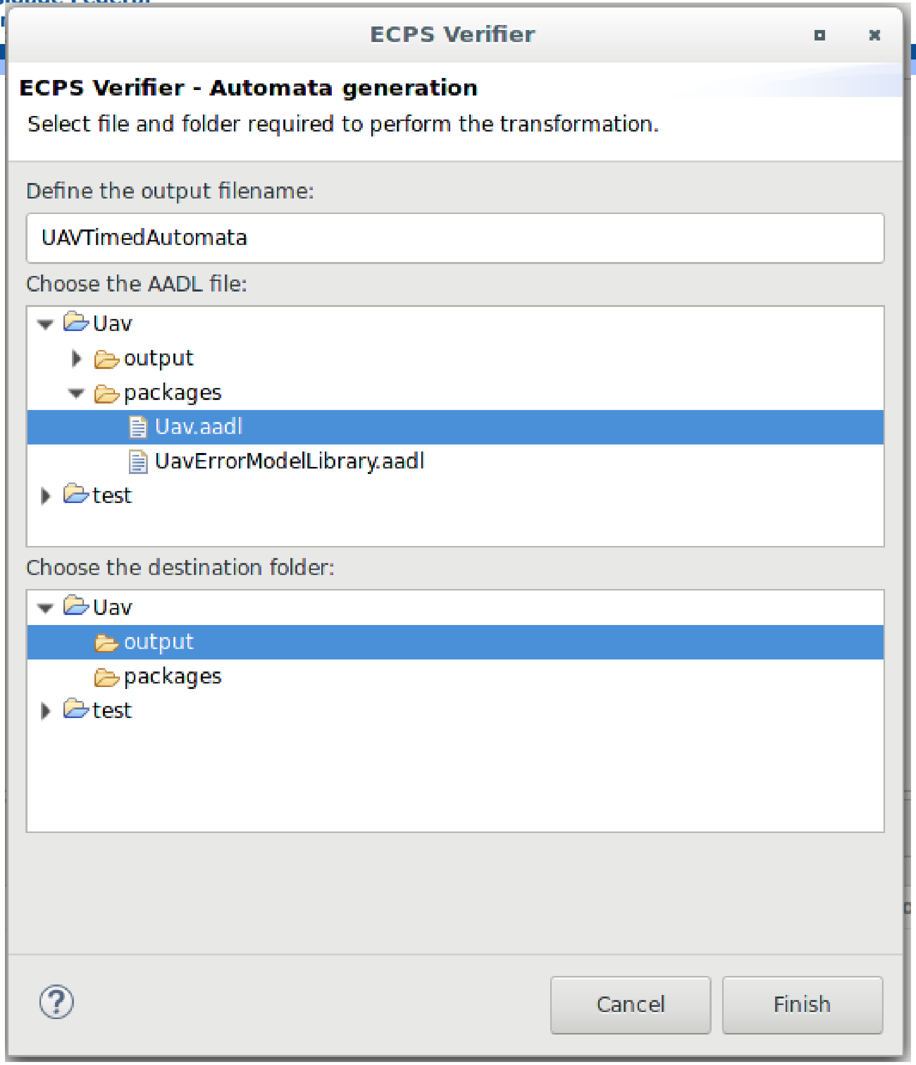
Universidade Federal de Santa Catarina

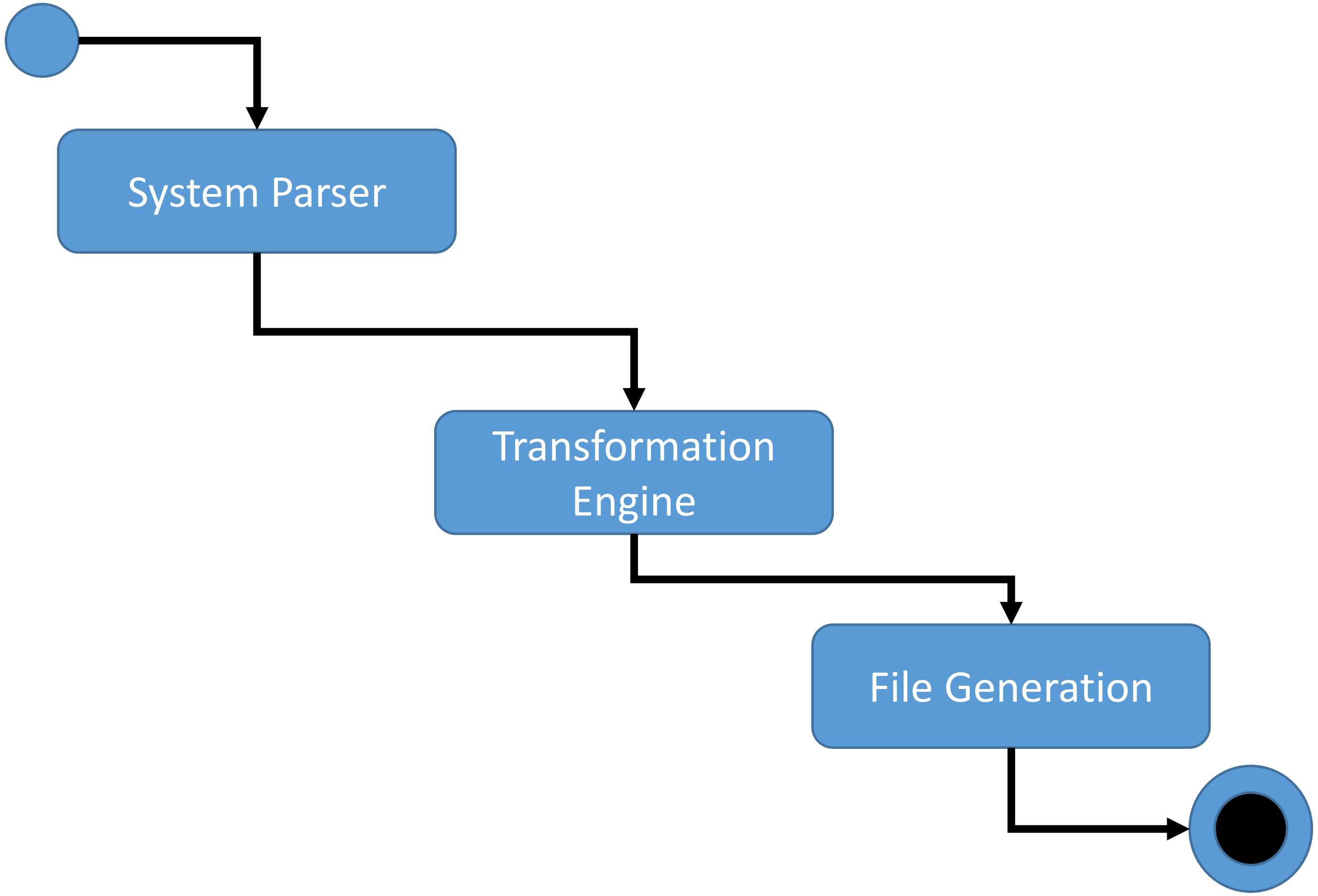
The University of Manchester

# Model Transformation Tool *ECPS Verifier*

- Automate the **UPAAL** model **generation based on AADL** model;
- Based on MDE principles;
- **Transformation rules** support the model construction;
- Auxiliary structures are required to represent UPPAAL properties on the source model;
- Based on metamodels provide the mapping between the languages.

# ECPS Modeling Tool



Universidade Federal de Santa Catarina

# ECPS Modeling – Transformation Engine

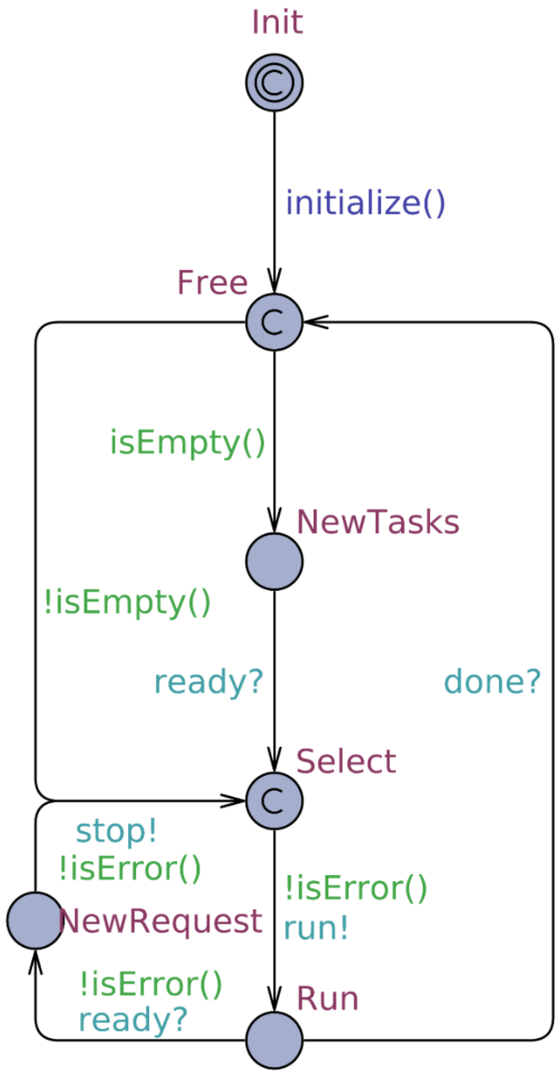
MANCHESTER 1824 The University of Manchester

- Provide the mapping between AADL and UPPAAL structures;
- Based in a set of transformation rules that describe:

1. The **timed automata generation** (UPPAAL template), representing the **thread behavior**, based on the component properties and its behavioral annex;
2. Represent the **devices properties** in a timed automata (UPPAAL template), that details its **operation** and a set of **possible failures**, defined on the error annex;
3. Detail the **threads characteristics** that allow the **system scheduling**.

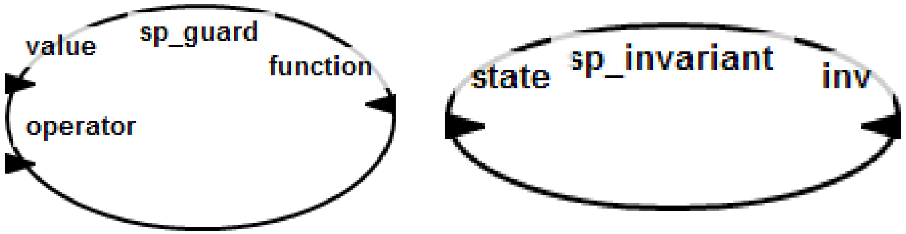
# Scheduler – Auxiliary Structures

- **Automatically included** along the transformation process;
- Uses **Rate-Monotonic** as default scheduling algorithm:
  - Activate the tasks execution according its priorities;



# Auxiliary Functions – Auxiliary Structures

## Auxiliary structures




## Scheduler functions


## Data types

chan

clock

# RMLib



- Runtime monitoring library developed at UFSC
- Suited for non-POSIX RTOS
- Light-weight monitor to collect user-generated events:
  - Task_init_exec
  - Task_end_exec
- Given a task-set specification, it calculates execution times and informs deadline misses
  - Data interpretation can be done online or offline
- Available at https://github.com/EltonBroering/RMLib

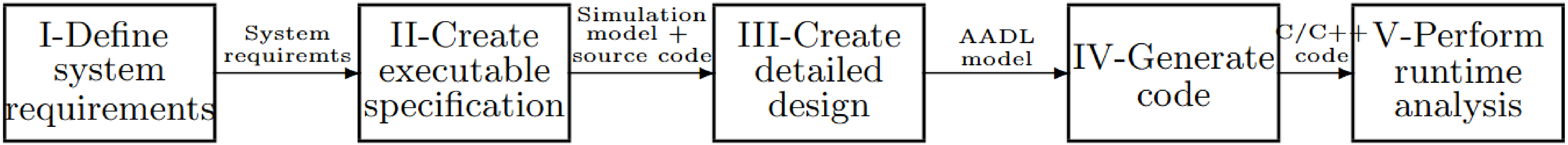
Universidade Federal de Santa Catarina

# Outline

MANCHESTER 1824
The University of Manchester

- Introduction: problem and related tools
- **Design Method Activities & Artifacts**
- Design of UAV’s continuous control architecture
- Conclusions and Future Works

# Design Method Activities & Artifacts



# Method Phases (1/3)

- **Activity 1 – Definition of system requirements:**
  - Set of requirements as spreadsheet information;

- **Activity 2 – Creation of Executable Specification:**
  - Specific view of the project, devoted for simulating the control algorithms;
  - Perform model-checking on the control algorithms with ESBMC

# Method Phases (2/3)



- **Activity 3 – Create detailed design:**
  - AADL modeling of the system;
  - Perform suitable analysis on the developed model with UPPAAL – use our model transformation tool;
- **Activity 4 – Generate code (implementation):**
  - Create C/C++ code from the AADL model;
  - Derived code can be subject of model-checking with ESBMC;

Universidade Federal de Santa Catarina

# Method Phases (3/3)

The University of Manchester

- **Activity 5 – Runtime Analysis**
  - Define execution scenarios and perform runtime monitoring with our RMLib

# Outline

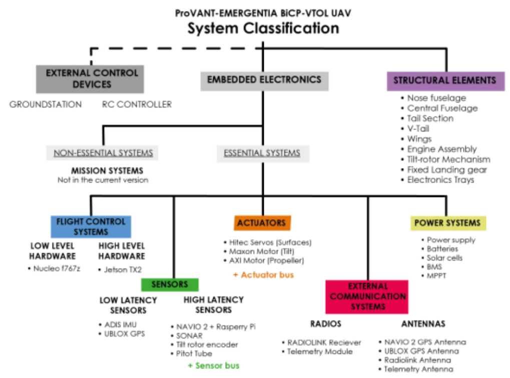
- Introduction: problem and related tools
- Design Method Activities & Artifacts
- **Design of UAV's continuous control architecture**
- Conclusions and Future Works

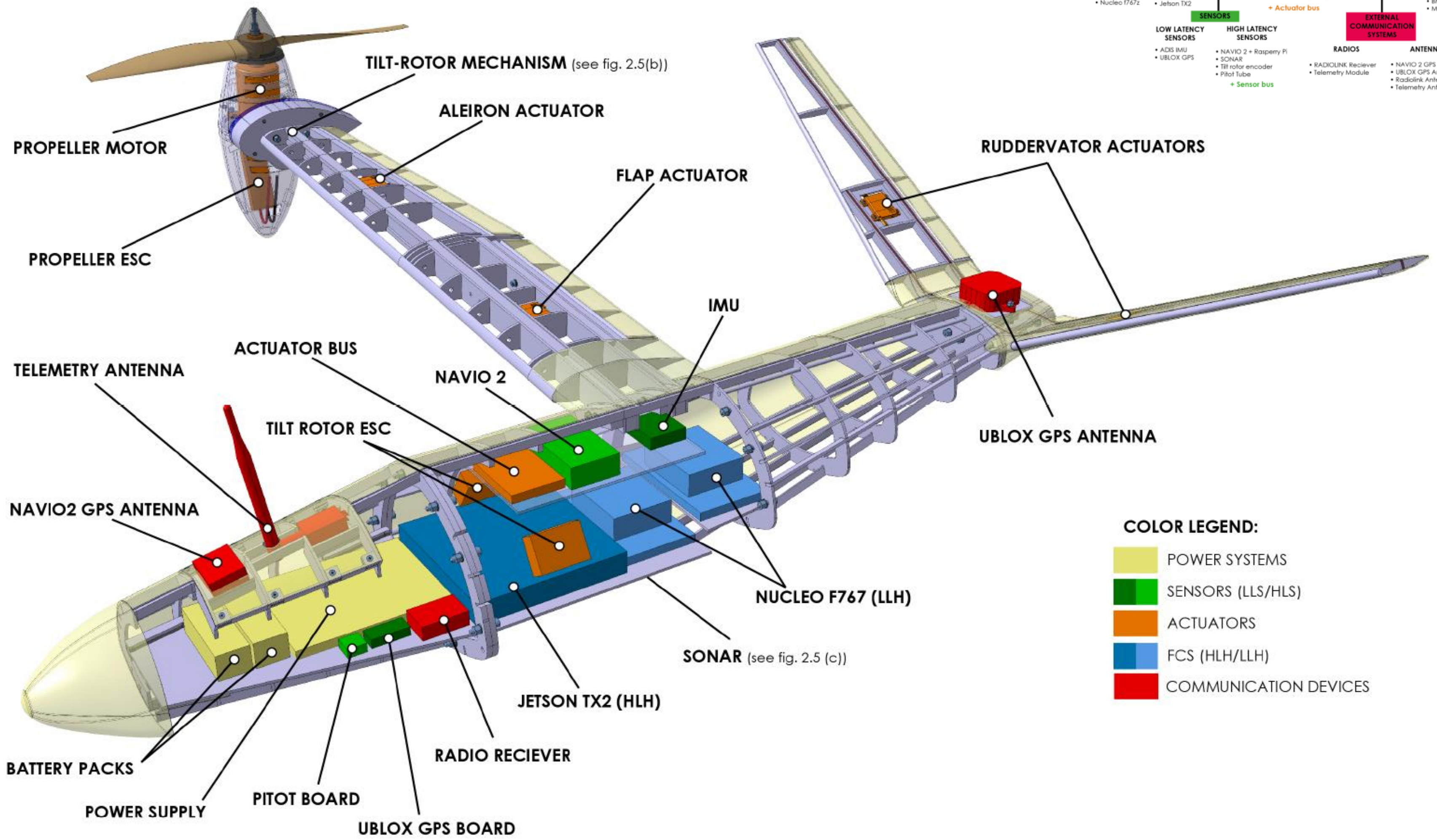
# UAV 4.0 Overview

Universidade Federal de Santa Catarina

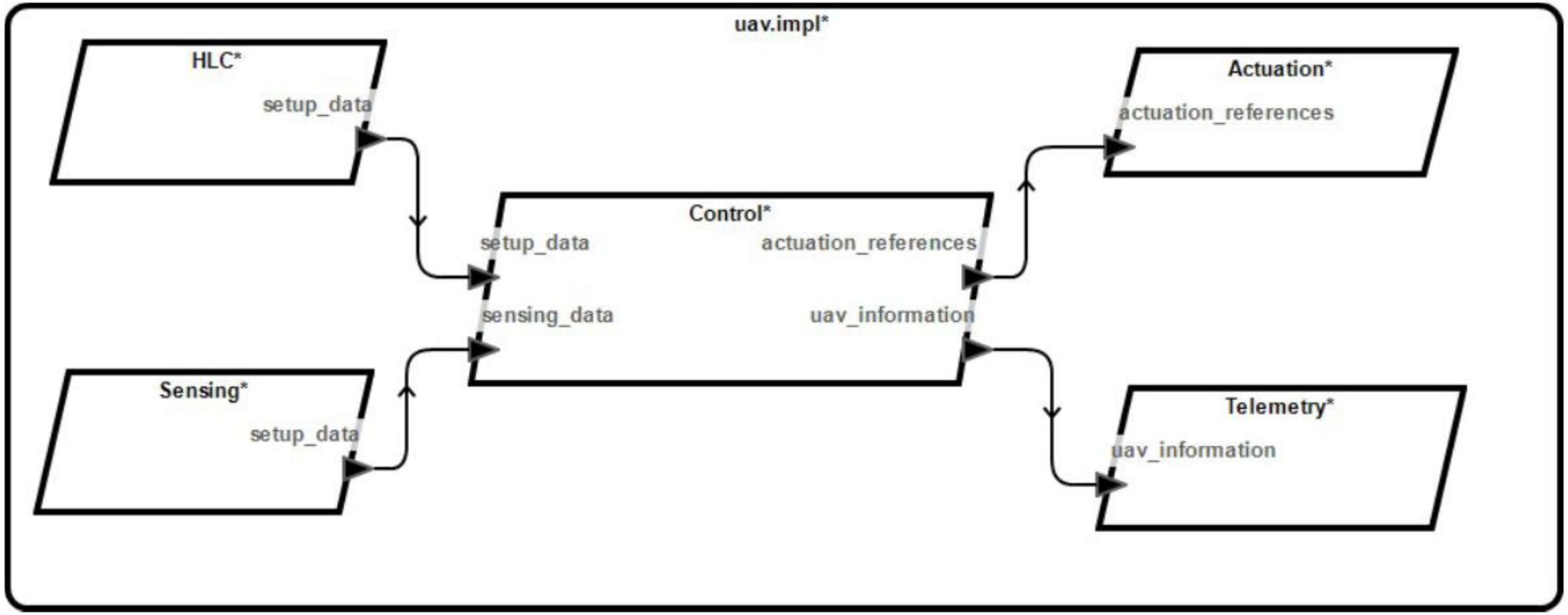
MANCHESTER 1824 The University of Manchester

UFMG



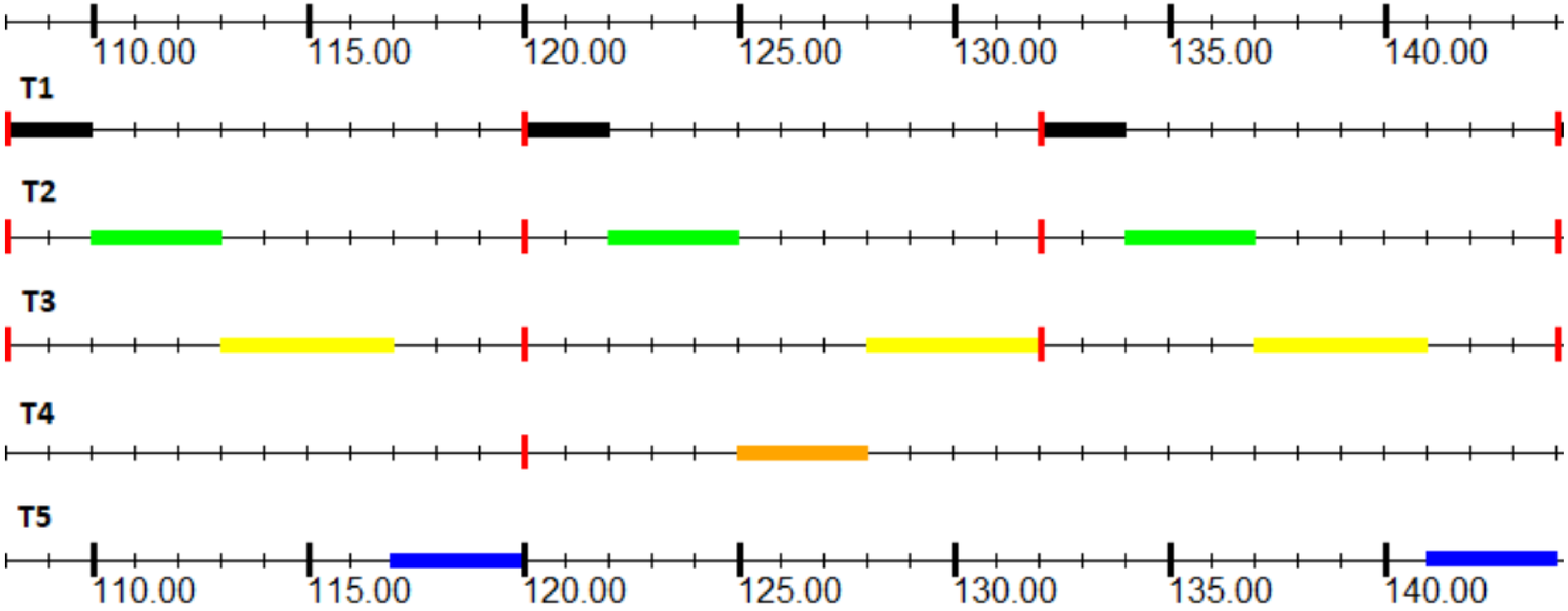
# AADL model for the Basic Flight Support System (BFSS)



# System Threads

| | Period | Deadl. | WCET | Utilizat. | Prior. | Preemptive |
|---|---|---|---|---|---|---|
| **T1-Actuation** | 12 | 12 | 2 | 0.166 | 5 | NO |
| **T2-Sensing** | 12 | 12 | 3 | 0.250 | 4 | NO |
| **T3-Control** | 12 | 12 | 4 | 0.333 | 2 | NO |
| **T4-HLC** | 120 | 120 | 4 | 0.033 | 3 | NO |
| **T5-Telemetry** | 600 | 600 | 130 | 0.216 | 1 | YES |

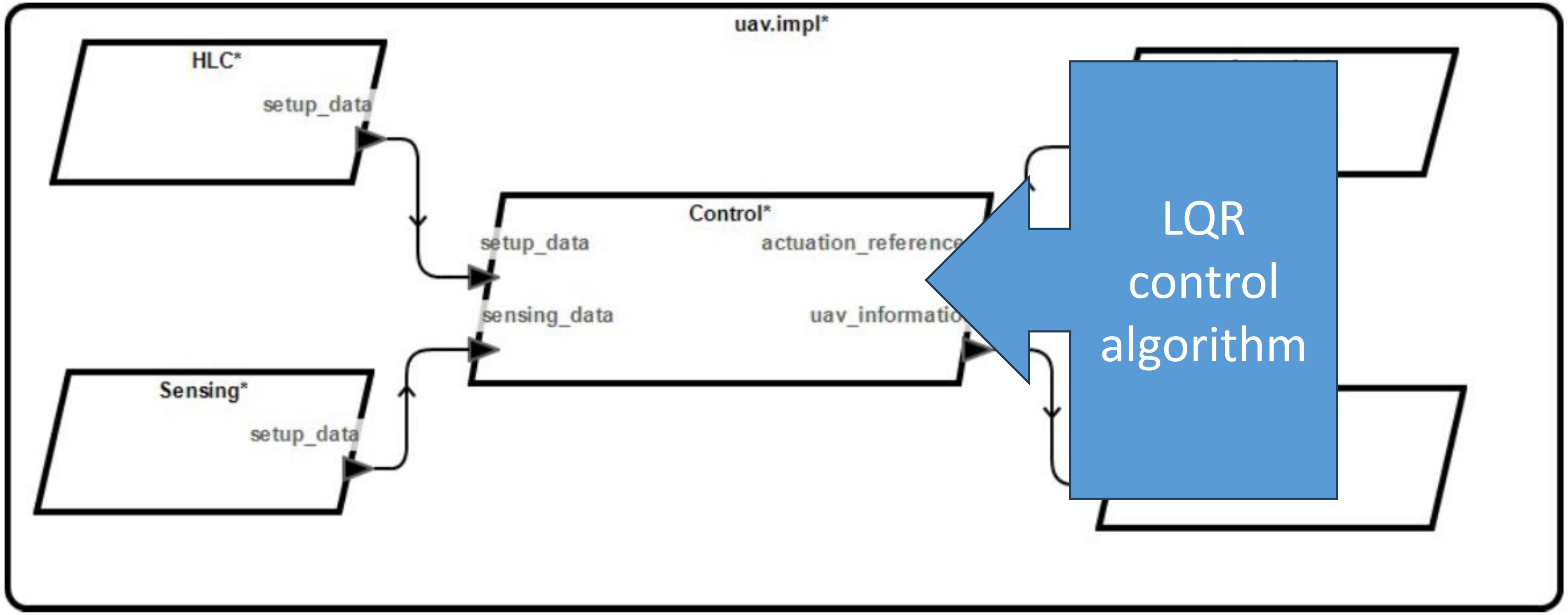
# System Threads Scheduling



# Properties formally verified with UPPAAL

| | |
|---|---|
| Spec.1 | Threads T1–T5 exec. time will never exceed their deadline, i.e., deadlines will not be missed (*Error* state will not be reached). |
| Query | $A \Box\ forall(i : 0 - 4)\ not\ T\_thread(i).Error$ |
| Result | Property is satisfied |

# ESBMC Verification



# ESBMC Verification

Universidade Federal de Santa Catarina

MANCHESTER 1824 The University of Manchester

| | Property | Solver | |
|---|---|---|---|
| | | Incremental | K-induction |
| 1 | floatbv | Passed | Passed |
| 2 | no bounds | Passed | Passed |
| 3 | deadlock | Passed | Passed |
| 4 | no assertions | Passed | Passed |
| 5 | no div by zero | Passed | Passed |
| 6 | no pointer | Passed | Passed |
| 7 | no align | Passed | Passed |
| 8 | no pointer relation | Passed | Passed |
| 9 | memory leak | Passed | Passed |
| 10 | NaN | Failed | Failed |
| 11 | overflow | Failed | Failed |
| 12 | data races | Passed | Passed |
| 13 | lock order | Passed | Passed |

# Runtime monitoring applied to the control thread

```
task_controller(void *pvParameters){
c_control_lqr_init();
while(true){
	vTaskSuspendAll();
	timestamp_runtime(TASK_IDENTIFIER_CONTROLLER,TASK_INIT_EXECUTION);
	c_control_lqr_controller(&controller_input,&controller_ouput);
	timestamp_runtime(TASK_IDENTIFIER_CONTROLLER,TASK_END_EXECUTION);
	vTaskDelay(ValueTaskDelay);
	xTaskResumeAll();
}}
```

# Runtime monitoring results



- No deadline miss observed along executions
- Allowed to validade assumptions for WCET estimations

Additional thread was created to add extra load into the system, leading to deadline misses in the control thread

# Outline

- Introduction: problem and related tools
- Design Method Activities & Artifacts
- Design of UAV’s continuous control architecture
- **Conclusions and Future Works**

# Conclusions

- Proposal to integrate formal verification in the UAV design process:
  - It covers code analysis, scheduling analysis, and runtime monitoring
- Allowed to validate the proposed UAV continuous control architecture under constrained conditions (HIL environment)
- Challenge: formally representing system properties on UPPAAL is not automatic and requires experience

# Future Works

- Investigate solutions to simplify properties specification.
- Adding rigor to requirements specification using Nasa’s FRET tool
- Moving from RT-monitoring to RT-verification
- Repeat RT-M/RT-V with the final UAV system implementation

# Thank you!
# Questions?

**Leandro Buss Becker**

leandro.becker@ufsc.br