The ROSACE Case Study

Thomas Loquen[1], Eric Noulard[1], Claire Pagetti[1,2], David Saussié[3] and Pierre Siron[4]

[1]ONERA - Toulouse, France

[2]INPT/ENSEEIHT - Toulouse, France

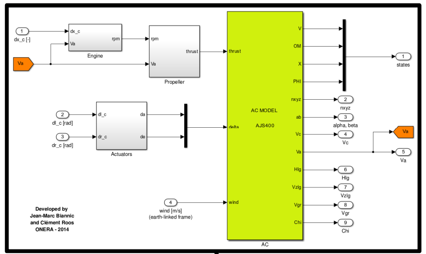
[3]Polytechnique - Montréal, Canada

[4]ISAE - Toulouse, France

Outline

- **General overview**
- Simulink specifications
- Checker
- Example of implementation
- Conclusion and perspectives

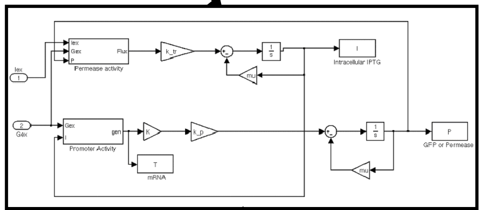
Trends in avionic domain



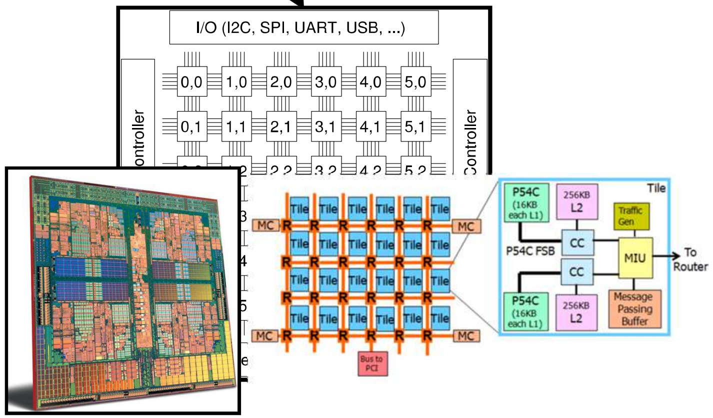
Control design level

- Steps: non linear ➔ linearization around a flight condition ➔ controller synthesis ➔ digitalization (sampling periods)
- Tools: Matlab / Simulink

Implementation level



- Coding of elementary blocks: Scade, Lustre …
- Coding of multi-periodic assembly: home made language, manual coding, …



COTS hardware integration

- Automatic code generator
- Manual code
- Low level services

Open source case study

ROSACE (Research Open-Source Avionics and Control Engineering)

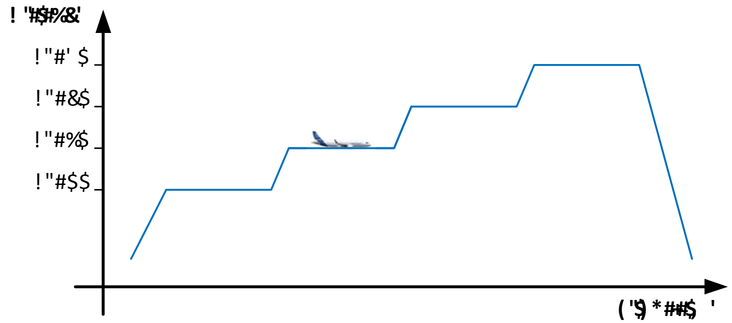
Originally presented in

Claire Pagetti, David Saussié, Romain Gratia, Eric Noulard et Pierre Siron. “The ROSACE Case Study : From Simulink Specification to Multi/Many-Core Execution”. In : 20th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS’14).

svn repository

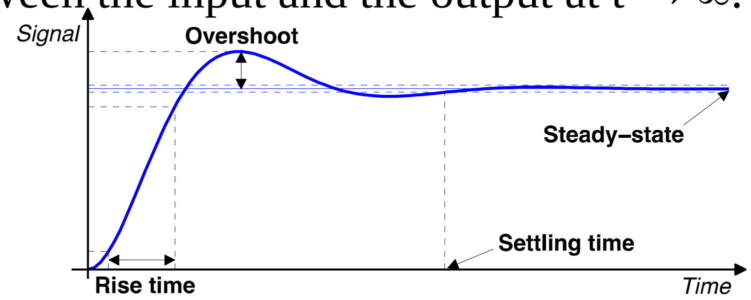
- https: //svn.onera.fr/schedmcore/branches/ROSACE_CaseStudy
- Content
  1. the SIMULINK specification (folder *simulink*)
  2. a checker to verify that an implementation fulfills the high level properties (folder *checker*)
  3. two examples of implementations (folder *prelude_implementations*)

Avionic use case: Longitudinal Flight Controller



- Longitudinal motion of a medium-range civil aircraft in *en-route* phase
  - *Cruise:* maintains a constant altitude h and a constant airspeed Va
  - *Change of cruise level* subphases:

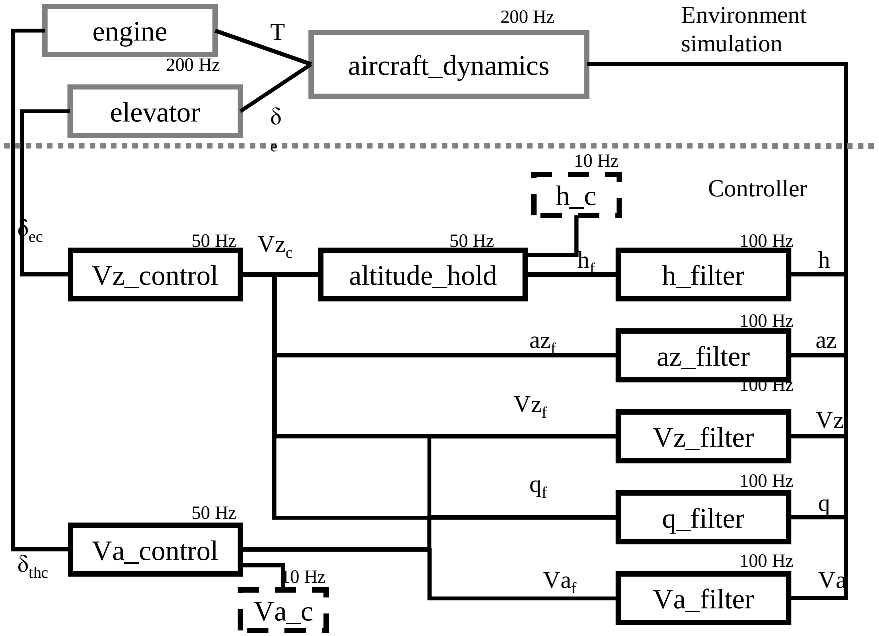
    commands a constant vertical speed Vz (rate of climb)
    - Ex: FL300 → FL320 → FL340 → FL360
    - FL300 = pressure altitude of 30000 ft
- **Performance requirements for change of cruise levels**
  - **P1 settling time**: time required to settle within 5% of the steady-state value
  - **P2 overshoot:** maximum value attained minus the steady-state value
  - **P3 rise time:** time to rise from 10% to 90% of the steady-state value
  - **P4 steady-state error:** difference between the input and the output at t → ∞.



Outline

- General overview
- **Simulink specifications**
- Checker
- Example of implementation
- Conclusion and perspectives

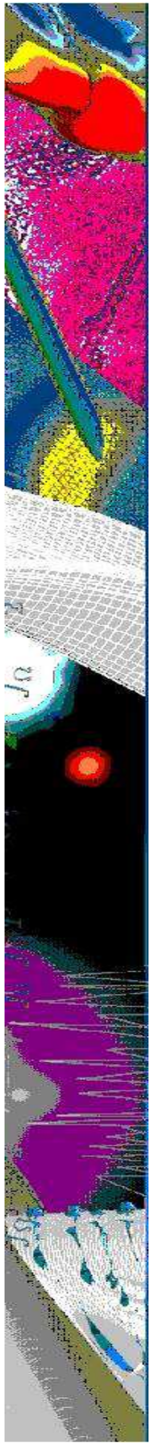
Longitudinal flight controller architecture



Flight condition:
h = 10000 m, Va = 230 m/s

Outputs	V_z V_a h a_z q	vertical speed true airspeed altitude vertical acceleration pitch rate
Filtered outputs	V_{z_f} V_{a_f} h_f a_{z_f} q_f	vertical speed true airspeed altitude vertical acceleration pitch rate
Reference inputs	h_c V_{a_c}	altitude command airspeed command
Commanded inputs	V_{z_c} δ_{e_c} δ_{th_c}	vertical speed command elevator deflection command throttle command
Aircraft inputs	δ_{e_c} T	elevator deflection engine thrust

- 5 filters consolidate the measured outputs provided by the sensors
- 3 controllers track accurately: altitude (h_c), vertical speed (V_{zc}) and airspeed commands (V_{ac})
- rate choices
  1. for controllers:
     - closed-loop system with the continuous-time controller can tolerate a pure time delay of 1 s before destabilizing ➔ sampling period ≤ 1 Hz
     - performances ➔ sampling period ≤ 10 Hz
  2. for environment: 200 Hz to model a continuous-time phenomenon

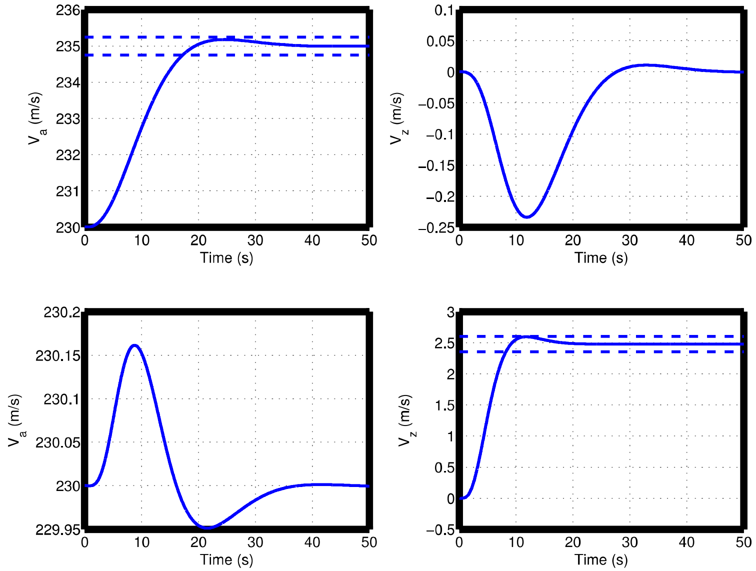


Validation objective (and analysis at Simulink level)

1. Analysis of V_a and V_z loops with separate step demands

airspeed variation of 5 m/s

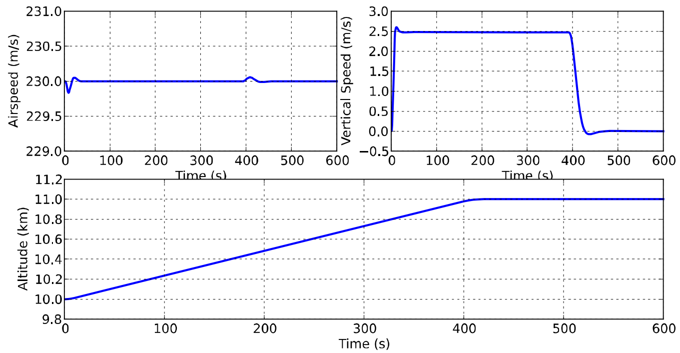
vertical speed demand of V_{zc}= 2.5m/s



2. Analysis of P4: input is a step climb

altitude change of 1000 m

- first phase: constant vertical speed demar
- second phase: altitude reaching



Validation objectives

Results for the decoupled approach

Property	Objective		Results in SIMULINK
P1 5% settling time	V_z	$\leq 10\,\mathrm{s}$	$8.22\,\mathrm{s}$
	V_a	$\leq 20\,\mathrm{s}$	$17.22\,\mathrm{s}$
P2 Overshoot	V_z	$\leq 10\%$	4.72%
	V_a	$\leq 10\%$	3.65%
P3 Rise time	V_z	$\leq 6\,\mathrm{s}$	$5.09\,\mathrm{s}$
	V_a	$\leq 12\,\mathrm{s}$	$11.6\,\mathrm{s}$
P4 Steady-state error	V_z	$\leq 5\%$	0.83%
	V_a	$\leq 5\%$	0.11%

Outline

- General overview
- Simulink specifications
- **Checker**
- Example of implementation
- Conclusion and perspectives

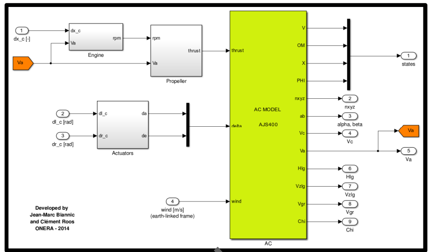
Property checker

- Python script: *check_result.py*
- Goal: verify that a tracing of a given simulation is compliant with the time-domain performance requirements of the previous table
  - Input format: CSV (comma separated value) file
  - 3 property checking: decoupled scenarios in Va and Vz and step climb
  - Possibility to draw the performances
- For a new implementation, the user must
  - Trace exactly the same variables. Which variables must be traced is detailed at the beginning of *simulink-run- scenarioX-results.csv*.
  - apply the same input step to the controller.
  - store the simulation (or execution) tracings in a .csv file.
  - call the property checker

Outline

- General overview
- Simulink specifications
- Checker
- **Example of implementation**
- Conclusion and perspectives

Forces 3



Control design level

ONERA Sim2LustrePrelude toolbox (Matlab) – open source – on demand

SchedMcore

```
node voter (s1, s2, s3: real)
returns (sensor: real);
let
   : : :
tel
```

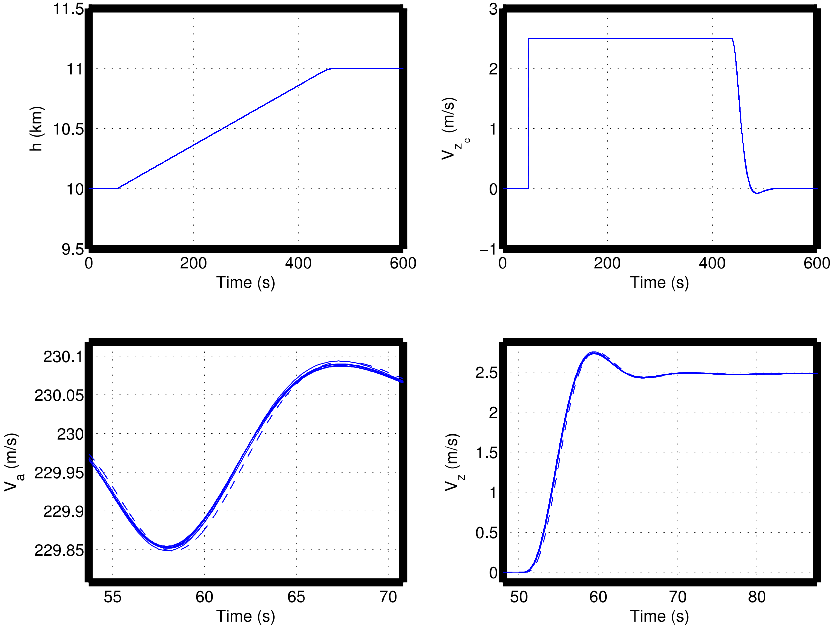
```
Imported node voter ((s1, s2, s3: real)
returns (sensor: real) wcet 5;
node ex (i1 : rate(8,0); i2 : rate(20,0))
returns (01, 02);
let
   : : :
tel
```

Implementation level

- Steps:
  - coding of elementary blocks: Lustre
  - coding of multi-periodic assembly: Prelude
  - simulation with
    - SchedMcore
      http://sites.onera.fr/schedmcore/
    - lustrec compiler
      cavale.enseeiht.fr/redmine/projects/lustrec
    - prelude compiler
      http://www.lifl.fr/~forget/prelude.html

Results

- **Objective:**
  - Validate the implementation wrt performance requirements
- **Input:**
  - Simulink specification
- **Results**
  - Lustre and Prelude associated files
  - Almost the same as those obtained with Simulink



Conclusion & perspectives

- Open case study for the community

- Future work:
  - Extension of the case study to consider lateral motion

Thank you for your attention