Parametric schedulability analysis of a launcher flight control system under reactivity constraints

Étienne André[1,2,3] Emmanuel Coquard[4] Laurent Fribourg[5] **Jawher Jerray**[1]
David Lesens[4]

[1] Université Paris 13, LIPN, CNRS, UMR 7030, F-93430, Villetaneuse, France

[2] JFLI, CNRS, Tokyo, Japan

[3] National Institute of Informatics, Japan

[4] ArianeGroup SAS

[5] LSV, ENS de Cachan & CNRS, France

27 juin 2019

Context : Verifying real-time systems

- Real-time systems :
  - Strong constraints on time. (e. g., a response passed a deadline is invalid even if its content appears to be correct.)
  - Real-time systems are everywhere
- Critical real-time systems :
  - Failures (in correctness or timing) may result in dramatic consequences

Context : Verifying real-time systems

- Real-time systems :
  - Strong constraints on time. (e. g., a response passed a deadline is invalid even if its content appears to be correct.)
  - Real-time systems are everywhere
- Critical real-time systems :
  - Failures (in correctness or timing) may result in dramatic consequences

Deepwater Horizon

Amagasaki Railway Accident

Flight 214 crash of Asiana Airlines

Real-time system

A real-time system is made of a set of tasks to execute on a processor

Real-time system

A real-time system is made of a set of tasks to execute on a processor

A task is characterized by :

- B : its best-case execution time
- W : its worst-case execution time
- D : its relative deadline

Real-time system

A real-time system is made of a set of tasks to execute on a processor

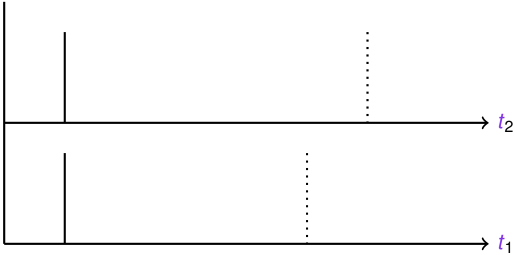
A task is characterized by :

- B : its best-case execution time
- W : its worst-case execution time
- D : its relative deadline

Tasks have instances that are activated (usually periodically)

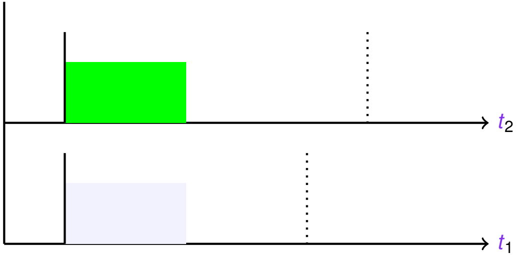
Example : shortest job first (SJF)

Task	B	W	D
t_1	3	3	4
t_2	2	2	5



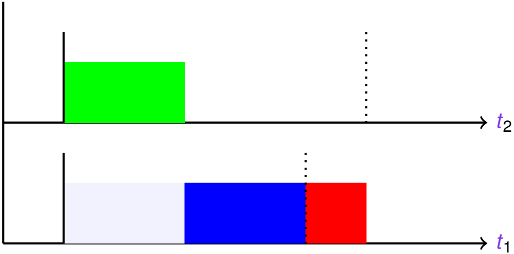
Example : shortest job first (SJF)

Task	B	W	D
t_1	3	3	4
t_2	2	2	5



Example : shortest job first (SJF)

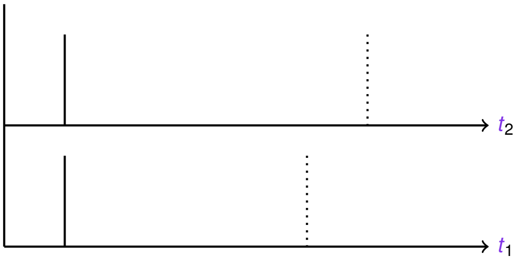
Task	B	W	D
t_1	3	3	4
t_2	2	2	5



Task t_1 misses its deadline

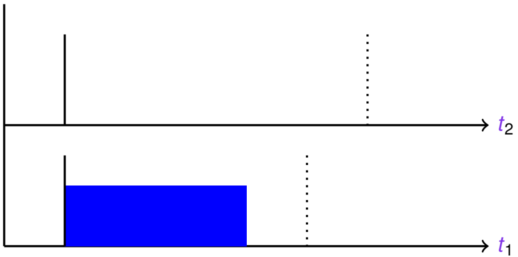
Example : preemptive fixed priority scheduler (FPS)

Task	B	W	D	priority
t_1	3	3	4	high
t_2	2	2	5	low



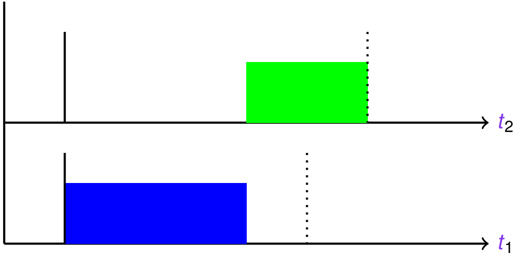
Example : preemptive fixed priority scheduler (FPS)

Task	B	W	D	priority
t_1	3	3	4	high
t_2	2	2	5	low



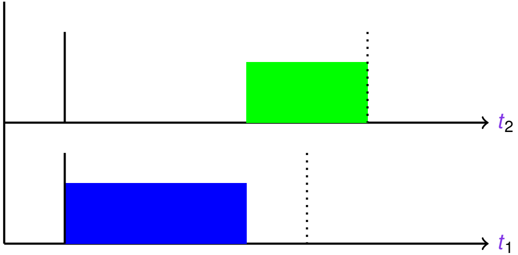
Example : preemptive fixed priority scheduler (FPS)

Task	B	W	D	priority
t_1	3	3	4	high
t_2	2	2	5	low



Example : preemptive fixed priority scheduler (FPS)

Task	B	W	D	priority
t_1	3	3	4	high
t_2	2	2	5	low



The system is schedulable

Scheduling

Scheduling

- Decide which task the processor runs at each moment.

- **Timing constraint :** priority, deadline, reactivity, preemption, . . .
- **Two main contexts :**
  - Centralized system [LL73]
  - Distributed system [TS06]

. [LL73] C. L. Liu et J. W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment", **Journal of the ACM**, t. 20, n^{o} 1, p. 46-61, 1973, ISSN : 0004-5411. DOI : 10.1145/321738.321743.

. [TS06] A. S. Tanenbaum et M. v. Steen, **Distributed Systems : Principles and Paradigms (2Nd Edition)**. Upper Saddle River, NJ, USA : Prentice-Hall, Inc., 2006, ISBN : 0132392275.

Schedulability analysis

Definition

- A system is schedulable if all tasks meet their deadline for all possible behaviors (according to the periods, interarrival rates, dependencies between tasks. . .).

Related work

Scheduling

- In [BB04], an efficient approach for testing schedulability for RMS (rate monotonic) analysis is proposed, through a “parameter” to balance complexity versus acceptance ratio.

Scheduling with model checking

- In [AAM06], TAs are used to solve job-shop and scheduling problems. This allows to find an optimal schedule and to model naturally more complex systems. However, comparing to our work this approach does not take into account parameters.
- The problem solved in [FBG+10] gathers with our approach. The major difference between those two approaches is summed up in the fact that, here, there are two distinct notions of “thread” and “processing”, while in [FBG+10] there was only one notion called “task”.

. [BB04] E. Bini et G. C. Buttazzo, “Schedulability Analysis of Periodic Fixed Priority Systems”, **IEEE Transactions on Computers**, t. 53, n^{o} 11, p. 1462-1473, 2004. DOI : 10.1109/TC.2004.103.

. [AAM06] Y. Abdeddaïm, E. Asarin et O. Maler, “Scheduling with timed automata”, **Theoretical Computer Science**, t. 354, n^{o} 2, p. 272-300, 2006.

. [FBG+10] J. Forget et al., “Scheduling Dependent Periodic Tasks without Synchronization Mechanisms”, in **RTAS**, Stockholm, Sweden : IEEE Computer Society, 2010, p. 301-310. DOI : 10.1109/RTAS.2010.26.

Related work

Scheduling with parameters

- In [CPR08], PTAs are used to encode real-time systems. A subclass (with bounded offsets, parametric WCETs) is exhibited that gives exact results. However, reactivities are not considered in [CPR08] unlike our approach.
- In [FLMS12], we performed robust schedulability analysis on an industrial case study, using the inverse method for PTAs implemented in IMITATOR. The main different between [FLMS12] and our case study comes from the system differs : [FLMS12] considers multiprocessor, and preemption can only be done at fixed instants, which therefore resembles more Round Robin than real FPS.

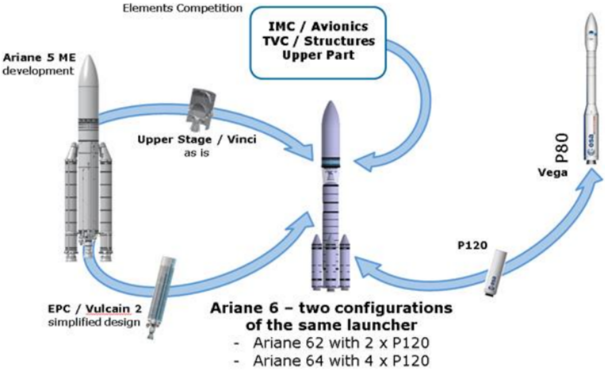
. [CPR08] A. Cimatti, L. Palopoli et Y. Ramadian, "Symbolic Computation of Schedulability Regions Using Parametric Timed Automata", in **RTSS**, Barcelona, Spain : IEEE Computer Society, 2008, p. 80-89, ISBN : 978-0-7695-3477-0. DOI : `10.1109/RTSS.2008.36`.

. [FLMS12] L. Fribourg et al., "Robustness Analysis for Scheduling Problems using the Inverse Method", in **TIME**, Leicester, UK : IEEE Computer Society Press, 2012, p. 73-80. DOI : `10.1109/TIME.2012.10`.

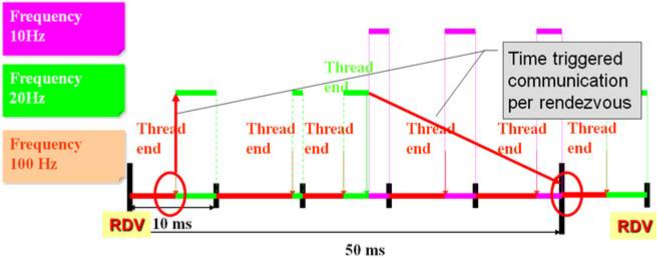
Ariane 6 industrial scenario

Objective

- Find flight control scheduling for the launcher, i.e. find the values of the task parameters (e. g., WCET) which meet the scheduling requirements (e. g., deadline).



Ariane 6



Input :

- Values of tasks priorities, task periods, set of reactivities (a reactivity is the maximum time from a data input and its output).
- Uncertainties on WCET, . . .
- Requirements (deadlines, . . .)

Output :

- Set of values for the uncertain parameters in order to meet the requirements of the scheduling.

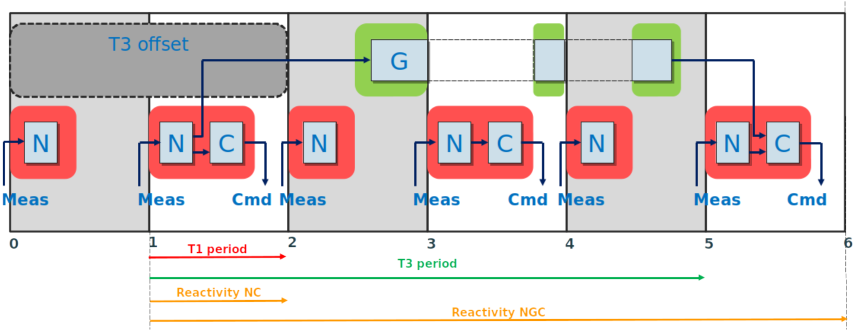
Data

Threads
Thread
ThreadT1
ThreadT2
ThreadT3

Processings		
Processing	**Period**	**WCET**
Control	10ms	3ms
Navigation	5ms	1ms
Guidance	60ms	15ms
Monitoring	20ms	5ms

Reactivities	
Reactivity	**Value**
Meas → Navigation → Guidance → Control → Cmd	150ms
Meas → Navigation → Control → Cmd	15ms
Meas → Navigation → Monitoring → Safeguard	55ms

Reactivity



Reactivities

Objectives

The objectives of our work

Determine the values of offsets and deadlines for threads such that :

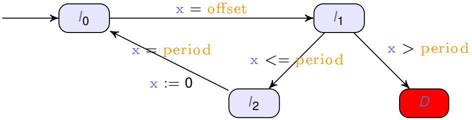
- the system is schedulable
- all reactivities are satisfied.

Our solution

- Method : Parametric timed model checking
- Formalism : parametric timed automata
- Toolkit : IMITATOR

- Translate each element of the system (threads, tasks, scheduling policy, reactivities) to a network of PTA. This elements are synchronized with each other.
- Unknown constants of the PTA correspond to the unknown constants of the problem (offset, deadline).
- The synthesis of constants in the PTA corresponds to the values for which the system is schedulable.

Parametric timed model checking

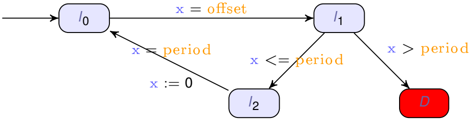


A model of the system

D is unreachable

A property to be satisfied

Parametric timed model checking



A model of the system

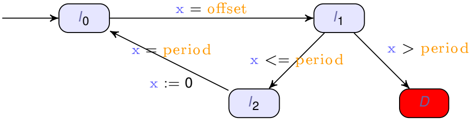
$\models$?

D is unreachable

A property to be satisfied

- Question : for what values of the parameters does the model of the system satisfy the property ?

Parametric timed model checking



A model of the system

? $\models$

D is unreachable

A property to be satisfied

- Question : for what values of the parameters does the model of the system satisfy the property ?

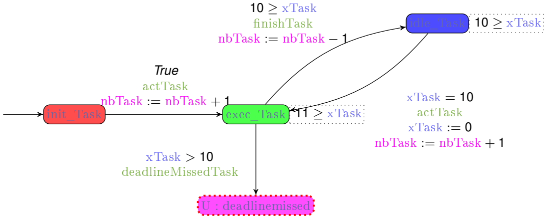
Yes if...

$$\text{offset} < \text{period}$$
$$\wedge \text{period} < 17.54$$

Why parametric timed automata ?

Parametric timed automata [AHV93]

- Formal semantics : automated formal analyzes possible.
- Allow very high **expressivity** : encoding inter-task dependencies, different scheduling policies [FLMS12], sporadic or periodic tasks, etc.
- Can be extended with stopwatches, to model preemption.
- Existence of the model checker IMITATOR.



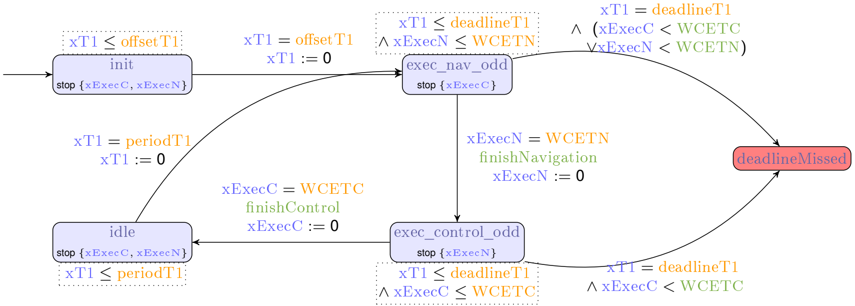
Example of PTA

. [AHV93] R. Alur, T. A. Henzinger et M. Y. Vardi, "Parametric real-time reasoning", in **STOC**, San Diego, California, United States : ACM, 1993, p. 592-601, ISBN : 0-89791-591-7.

. [FLMS12] L. Fribourg et al., "Robustness Analysis for Scheduling Problems using the Inverse Method", in

Modeling offsets and deadlines

Example of Offset and Deadline Modeling for Thread T1 :

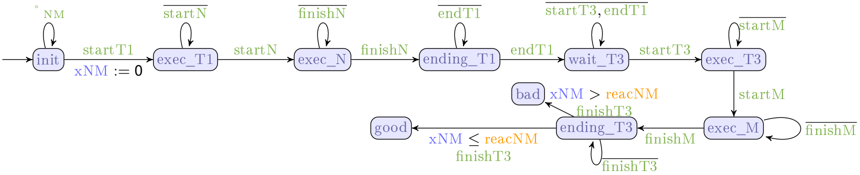


Fragment of automaton threadT1

Modeling reactivities

A reactivity is required between a data read from the avionics bus (a measurement) and a data written to the avionics bus (a command)

Example of reactivity modeling Navigation → Monitoring :



Reactivity Navigation → Monitoring

Experimental environment

- Translate the network of PTA to the IMITATOR input language [AFKS12].
- IMITATOR is a tool for modeling and verifying real-time systems with unknown constants modeled with parametric timed automata[AHV93]. This parametric model checker takes as input networks of PTA extended with useful features such as synchronization actions and discrete variables.

. [AFKS12] É. André et al., "IMITATOR 2.5 : A Tool for Analyzing Robustness in Scheduling Problems",, t. 7436, Paris, France, 2012. DOI : 10.1007/978-3-642-32759-9_6.

. [AHV93] R. Alur, T. A. Henzinger et M. Y. Vardi, "Parametric real-time reasoning", in STOC, San Diego, California, United States : ACM, 1993, p. 592-601, ISBN : 0-89791-591-7.

The results

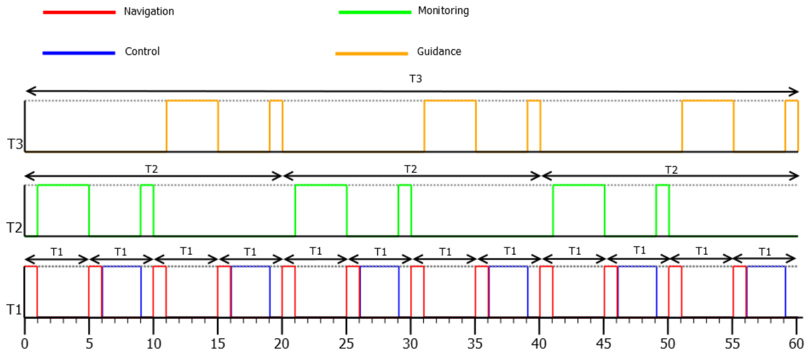
The type of scheduling used in these results is : Fixed-priority scheduling (FPS) with preemption

- First result : we checked the instantiated model by setting the offsets to 0 and the deadlines to the period of each Thread. In that case, all three reactivity automata are included in the model.

The results of IMITATOR and their execution times

	Result	E.T(seconds)
Result 1	True	109

The results



Chronogram of the scheduling of the instantiated model

Results

- Second result : we have parameterized the offsets of the threads and we have instantiated the deadlines to the value of the periods.

The results of IMITATOR and their execution times

	Result	E.T(seconds)
Result 2	$11 >= \mathrm{offsetT2}$ $\wedge\ \mathrm{offsetT3} >= 0$ $\wedge\ \mathrm{offsetT2} >= \mathrm{offsetT3}$ $\wedge\ 1 >= \mathrm{offsetT3}$ $\wedge\ \mathrm{offsetT1} = 0$ **OR** $\wedge\ \mathrm{offsetT3} > \mathrm{offsetT2}$ $\wedge\ 1 >= \mathrm{offsetT2}$ $\wedge\ \mathrm{offsetT2} >= 0$ $\wedge\ 11 >= \mathrm{offsetT3}$ $\wedge\ \mathrm{offsetT1} = 0$ **OR** $\wedge\ \mathrm{offsetT2} >= 0$ $\wedge\ \mathrm{offsetT1} > 0$ $\wedge\ 11 >= \mathrm{offsetT2}$ $\wedge\ 4 >= \mathrm{offsetT1}$ $\wedge\ \mathrm{offsetT3} = 0$	2303

The results

- Third result : we have initialized the offsets of the threads to 0 and we have parameterized the deadlines.

The results of IMITATOR and their execution times		
	Result	E.T(seconds)
Result 3	$deadlineT2 >= 11$ $\wedge$ &$deadlineT1 >= 4$ $\wedge$ &$5 >= deadlineT1$ $\wedge$ &$20 >= deadlineT2$ $\wedge$ &$deadlineT3 = 60$	637

conclusion and perspectives

Conclusion

- Formally check that the FPS type scheduling can be a solution for our problem.
- Check that reactivities are met for which we proposed a compositional solution.
- Determine the offsets and deadlines of threads.

Perspectives

- Automate the assignment of processings in threads.
- Take into account the switch between two threads due to the copy of data between the contexts of each thread which is in our example 500 μs.
- Minimize the execution time of the algorithm.

Y. ADBEDDAÏM, E. ASARIN et O. MALER, "Scheduling with timed automata", **Theoretical Computer Science**, t. 354, n° 2, p. 272-300, 2006.

É. ANDRÉ, L. FRIBOURG, U. KÜHNE et R. SOULAT, "IMITATOR 2.5 : A Tool for Analyzing Robustness in Scheduling Problems",, t. 7436, Paris, France, 2012. DOI : 10.1007/978-3-642-32759-9_6.

R. ALUR, T. A. HENZINGER et M. Y. VARDI, "Parametric real-time reasoning", in **STOC**, San Diego, California, United States : ACM, 1993, p. 592-601, ISBN : 0-89791-591-7.

E. BINI et G. C. BUTTAZZO, "Schedulability Analysis of Periodic Fixed Priority Systems", **IEEE Transactions on Computers**, t. 53, n° 11, p. 1462-1473, 2004. DOI : 10.1109/TC.2004.103.

A. CIMATTI, L. PALOPOLI et Y. RAMADIAN, "Symbolic Computation of Schedulability Regions Using Parametric Timed Automata", in **RTSS**, Barcelona, Spain : IEEE Computer Society, 2008, p. 80-89, ISBN : 978-0-7695-3477-0. DOI : 10.1109/RTSS.2008.36.

J. FORGET, F. BONIOL, E. GROLLEAU, D. LESENS et C. PAGETTI, "Scheduling Dependent Periodic Tasks without Synchronization Mechanisms", in **RTAS**, Stockholm, Sweden : IEEE Computer Society, 2010, p. 301-310. DOI : 10.1109/RTAS.2010.26.

L. Fribourg, D. Lesens, P. Moro et R. Soulat, “Robustness Analysis for Scheduling Problems using the Inverse Method”, in **TIME**, Leicester, UK : IEEE Computer Society Press, 2012, p. 73-80. doi : 10.1109/TIME.2012.10.

C. L. Liu et J. W. Layland, “Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment”, **Journal of the ACM**, t. 20, nº 1, p. 46-61, 1973, issn : 0004-5411. doi : 10.1145/321738.321743.

A. S. Tanenbaum et M. v. Steen, **Distributed Systems : Principles and Paradigms (2Nd Edition)**. Upper Saddle River, NJ, USA : Prentice-Hall, Inc., 2006, isbn : 0132392275.

Source of the graphics used I

Title : Deepwater Horizon offshore drilling unit on fire
Author : Unknown
Source : https://commons.wikimedia.org/wiki/File:Deepwater_Horizon_offshore_drilling_unit_on_fire_2010.jpg
License : Public domain

Title : The Amagasaki rail crash
Author : To Sawa
Source : https://commons.wikimedia.org/wiki/File:Fukuchiyama_joko20051.jpg
License : CC BY-SA 3.0

Title : Asiana Airlines Plane Crash
Author : Alexander Novarro
Source : https://commons.wikimedia.org/wiki/File:Asiana_Airlines_Plane_Crash.png
License : CC BY-SA 3.0