

Sujet de stage recherche de niveau M2

## Vérification distribuée de systèmes temps-réel

**Encadrants :** Étienne ANDRÉ et Laure PETRUCCI  
**Adresse électronique :** {prenom.nom}@lipn.univ-paris13.fr  
**Laboratoire d'accueil :** LIPN, CNRS UMR 7030, Université Paris 13, Sorbonne Paris Cité

### Contexte

Les systèmes temps réel sont devenus omniprésents ces dernières années. Certains d'entre eux (pilotage automatique des avions et drones, voitures sans chauffeur, systèmes bancaires) sont critiques en ce sens qu'aucune erreur ne doit survenir. Tester de tels systèmes peut éventuellement détecter la présence de bugs, mais pas en garantir l'absence. Il est alors nécessaire d'utiliser des méthodes telles que le model checking [BK08] afin de prouver formellement la correction d'un système.

Les systèmes temps réel sont caractérisés par un ensemble de constantes temporelles, telles que la période de lecture d'un capteur d'altitude sur un drone, le temps de traversée d'un circuit par le courant électrique, ou le délai avant la retransmission des données dans un téléphone portable. Bien que des techniques permettant de vérifier le système pour *un* ensemble de constantes existent, vérifier formellement le système pour de nombreuses valeurs de ces constantes peut demander un temps extrêmement long, voire infini si l'on cherche à vérifier des ensembles denses de valeurs.

Il est alors intéressant de raisonner paramétriquement, en considérant que ces constantes temporelles sont inconnues, c'est-à-dire des *paramètres*, et synthétiser une contrainte sur ces paramètres garantissant la correction du système. Une méthode, la *méthode inverse* [AS13, APP13], a été proposée dans le cadre des automates temporisés [AHV93], et des réseaux de Petri temporels [JK09, TLR09], deux formalismes largement utilisés. À partir d'une valeur de référence des constantes temporelles correspondant à un comportement correct, cette méthode synthétise une contrainte sur les paramètres garantissant le même comportement correct. Par conséquent, ceci garantit que le système sera également correct pour toute valeur des paramètres satisfaisant cette contrainte.

### Sujet du stage

Afin de tirer parti de la puissance des processeurs multi-cœurs et des clusters, l'algorithme de vérification devra être redéfini pour s'adapter au cas distribué. Le but est que, pour un processeur à  $n$  cœurs, l'algorithme adapté puisse être (presque)  $n$  fois plus rapide que sur un mono-cœur – et similairement pour les clusters. Il pourra être intéressant de se baser en partie sur une approche modulaire proposée pour les réseaux de Petri temporisés [LP07].

Une implémentation sera également effectuée par l'étudiant afin de valider l'approche proposée. Une option possible est de réutiliser l'outil IMITATOR [AFKS12].

## Mots-clés

Méthodes formelles, model-checking, algorithmique distribuée, systèmes temps-réel, synthèse de paramètres

## Compétences

Une connaissance d'un ou plusieurs des concepts suivants serait un plus, sans être indispensable pour autant : calcul distribué, automates temporisés, OCaml.

## Conditions

Le stage se déroule au LIPN (Laboratoire d'Informatique de Paris Nord) à l'Université de Paris 13, Sorbonne Paris Cité (campus de Villetaneuse).

Rémunération standard.

En fonction de la motivation et du projet professionnel du candidat, ce stage est susceptible de se prolonger par une thèse.

## Références

- [AFKS12] Étienne André, Laurent Fribourg, Ulrich Kühne, and Romain Soulat. IMITATOR 2.5 : A tool for analyzing robustness in scheduling problems. In *Proceedings of the 18th International Symposium on Formal Methods (FM'12)*, volume 7436 of *Lecture Notes in Computer Science*, pages 33–36. Springer, 2012.
- [AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. Parametric real-time reasoning. In *STOC'93*, pages 592–601. ACM, 1993.
- [APP13] Étienne André, Laure Petrucci, and Giuseppe Pellegrino. Precise robustness analysis of time Petri nets with inhibitor arcs. In *FORMATS'13*, volume 8053 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2013.
- [AS13] Étienne André and Romain Soulat. *The Inverse Method*. FOCUS Series in Computer Engineering and Information Technology. ISTE Ltd and John Wiley & Sons Inc., 2013. 176 pages.
- [BK08] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, 2008.
- [JK09] Kurt Jensen and Lars Michael Kristensen. *Coloured Petri Nets – Modelling and Validation of Concurrent Systems*. Springer, 2009.
- [LP07] Charles Lakos and Laure Petrucci. Modular state space exploration for timed Petri nets. *Journal of Software Tools for Technology Transfer*, 9(3-4) :393–411, jun 2007.
- [TLR09] Louis-Marie Traounez, Didier Lime, and Olivier H. Roux. Parametric model-checking of stopwatch Petri nets. *Journal of Universal Computer Science*, 15(17) :3273–3304, 2009.