

Analyse des performances temporelles de systèmes de commande distribués sujet à pannes

Synergie Simulation et Model-Checking Paramétré

SIMOP

Durée estimée du projet : 2 ans.

Responsables scientifiques

Olivier De Smet (LURPA), Laurent Fribourg (LSV)

Membres de l'équipe projet

Permanents

LSV : Laurent Fribourg (DR) 20 %, Emmanuelle Encrenaz (MCF) 20 %

LURPA : Olivier De Smet (MCF) 30 %, Bruno Denis (MCF) 20 %

Etudiants

LSV : 1 étudiant en Master M2 Recherche (1^{er} semestre 2007), avec poursuite en thèse sur la thématique (à 50% par an sur 2 ans)

LURPA : 1 doctorant (à 50% par an sur 2 ans)

Problématique scientifique

Contexte

Actuellement, que cela soit dans le domaine des systèmes embarqués pour l'automobile ou des systèmes de production, la commande des systèmes complexes est répartie sur plusieurs contrôleurs reliés entre eux par un réseau (spécialisé ou généraliste, de type Ethernet). Ce réseau permet de découpler l'accès à l'information via des modules d'entrées-sorties de son traitement réalisé dans des contrôleurs distants [Thomasse 99]. Cette nouvelle technologie offre aux industriels de nouvelles possibilités de conception d'architecture de commande, plus flexibles, intégrant notamment la mise en place de stratégies tolérantes aux pannes.

En pratique, les fonctions que la commande doit réaliser sont implantées sur plusieurs contrôleurs distincts ; en mode de fonctionnement nominal, seul un contrôleur particulier doit réaliser une fonction donnée, mais si celui-ci tombe en panne (panne de type crash), un autre contrôleur, sur lequel les fonctionnalités du premier ont préalablement été implantées, peut réaliser, en plus des traitements associés à ses fonctions propres, tout ou partie du traitement des fonctions du contrôleur défaillant. Les performances (temporelles) du contrôleur assurant le traitement de toutes ces fonctions peuvent se trouver dégradées par la surcharge de travail qui lui a été attribué. On parle alors de fonctionnement en mode dégradé.

L'analyse des performances temporelles d'un système distribué (en considérant ses différents modes de fonctionnement) est un sujet de recherche important : il s'agit de déterminer l'impact de la défaillance d'un contrôleur sur les performances globales du système. Cette étude peut mener à l'exploration de différentes possibilités de mise en place de la redondance, en fonction des objectifs de performance à atteindre.

L'évaluation des performances d'un tel système est difficile : la construction d'un modèle fiable (et précis) est indispensable, mais le modèle obtenu est en général trop complexe pour pouvoir être appréhendé dans son intégralité. On a recours à des méthodes de *simulation* du modèle, qui donnent des mesures de performances assez précises *pour le jeu de test (scénario) qui a été appliqué* [Jasperneite 02] [Vitturi 03]. Le problème qui se pose alors est celui de la validité du jeu de test : l'ensemble des simulations réalisées est-il un bon représentant des comportements de l'environnement du système [Meunier 06] ? Une démarche alternative consiste à utiliser des *méthodes exhaustives*, telles le « model-checking », mais qui, en général, ne peuvent produire de résultat que sur des systèmes de petite taille, ou alors après une abstraction du système [Marsal 05]. Se pose alors le problème de la *fiabilité de l'abstraction*, par rapport au système que l'on cherche à analyser.

Le model-checking temporisé est une technique d'analyse des performances temporelles de systèmes concurrents développées depuis 15 ans et qui a permis d'analyser les caractéristiques temporelles de divers protocoles de communication distribués et temporisés (BRP de Philips [Abdulla 99], ABR [Bérard 99]), ou de circuits électroniques asynchrones [Maler 95] [Bozga 02]. Tout récemment, le model-checking temporisé et paramétré a été utilisé pour déterminer les relations entre différents paramètres temporels de systèmes matériels ([Clariso 04] [Chevallier 06]). Cette voie semble intéressante pour analyser les performances d'architectures de commande distribuées en considérant différents modes de fonctionnement.

Objectif

L'objectif du projet est d'évaluer les performances d'une architecture de commande distribuée sujette à panne et d'estimer la plage de valeur des paramètres de fonctionnement de l'architecture qui garantit les performances attendues. Les différents modes de fonctionnement (mode nominal, différentes configurations de mode dégradé) seront pris en compte.

Pour atteindre cet objectif on utilisera le model-checking temporisé paramétré pour sa capacité à fournir des résultats à la fois *quantitatifs* et *garantis* sur les paramètres de fonctionnement recherchés. Cependant sa mise en oeuvre est généralement limitée par la complexité intrinsèque du problème. Afin d'éviter cet écueil on utilisera au préalable des observations issues de simulation pour limiter l'espace de recherche du model-checking. Ainsi la méthode de model-checking temporisé et paramétré sera dirigée par des résultats issus de simulation.

Les résultats obtenus seront confrontés au réel en utilisant la plate-forme d'automatisation en réseau du LURPA, où des pannes pourront être simulées, et les performances temporelles mesurées.

Originalité

Pour obtenir des garanties sur des plages de fonctionnement de notre architecture, le model-checking temporisé paramétré a été retenu. L'utilisation de techniques de model-checking temporisé pour notre problématique mène rapidement à une explosion combinatoire. Pour repousser ces limites, on introduit de l'information issue d'une analyse préalable par simulation.

L'originalité de notre approche consiste à utiliser les résultats d'analyse par simulation, non pas comme palliatif au model-checking défaillant mais à l'utiliser en amont pour identifier de manière experte les zones pertinentes de fonctionnement. Ainsi la simulation fournit une partie des données d'entrée du model-checking et restreint son espace de recherche. Les résultats finaux sont obtenus du model-checker paramétrique et garantissent la validité de la propriété attendue. Ces résultats expriment des plages de valeurs des paramètres de fonctionnement garantissant des temps de réponse de l'architecture distribuée sujette à panne.

La figure 1 illustre le principe de l'approche retenue pour l'analyse, et la place par rapport aux approches dites « classiques » qui sont limitées soit par la portée des résultats d'analyse soit par la petite taille des systèmes analysables.

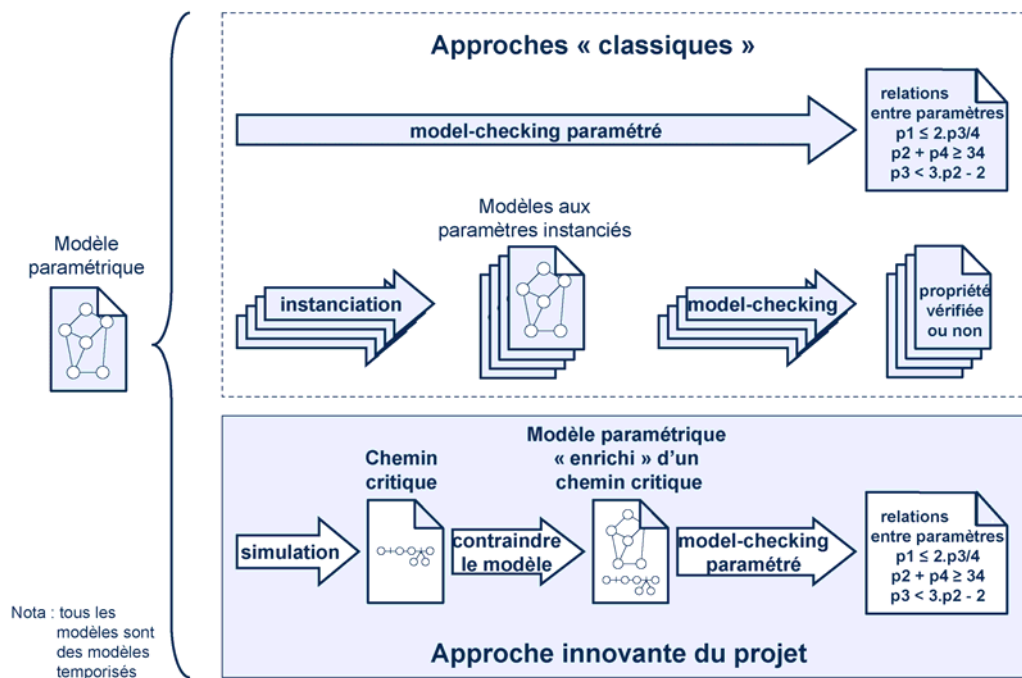


Figure 1 : positionnement de l'approche proposée

Un point fort du projet est la validation des résultats obtenus par confrontation avec le réel. Les plages de valeurs des paramètres issues du model-checker donneront lieu à des réglages sur l'architecture de test (LURPA) puis à des mesures des performances attendues à la fois en mode nominal et avec des pannes simulées.

Organisation du projet

Le projet se décompose en 4 tâches dont les principales entrées et sorties sont illustrées dans la figure 2.

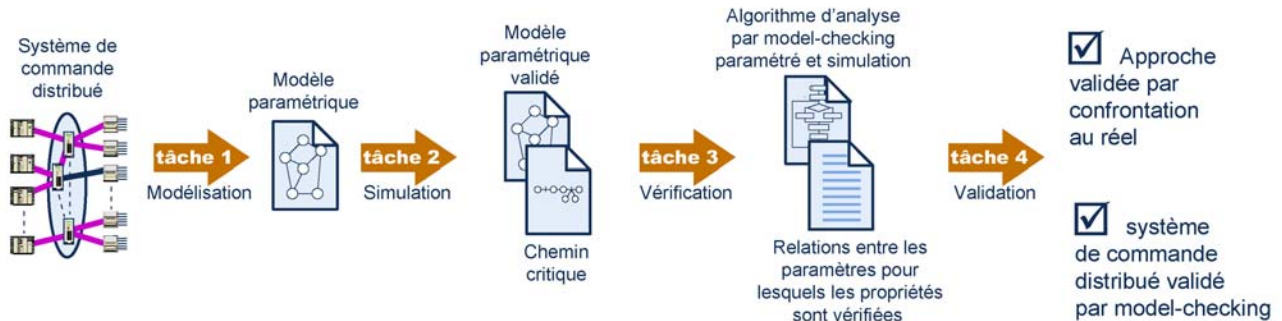


Figure 2 : principales entrées et sorties de chaque tâche

Tâche 1 : Modélisation d'un sous-ensemble d'une architecture de commande distribuée

- 1.1 Définition d'un sous-ensemble d'une architecture de commande disponible au LURPA, sur lequel il est possible d'obtenir des valeurs temporelles concrètes (la référence).
- 1.2 Définition précise de la spécification : il s'agit de l'ensemble des propriétés attendues énoncées comme un ensemble de règles non ambiguës.
- 1.3 Choix de formalismes pour représenter le système dans un objectif de simulation et de model-checking : automates temporisés (où sont représentés les traitements et les délais), réseaux de Petri temporisés.
- 1.4 Choix d'un niveau d'abstraction : il doit être suffisamment précis pour représenter fidèlement les temps de réponse et la fonctionnalité de l'architecture de commande, tout en étant accessible par les outils de model-checking.
- 1.5 Elaboration d'un modèle paramétrique de l'architecture de commande distribuée (sa validation se fera par une première confrontation au réel en tâche 2.3).

Tâche 2 : Analyse par simulation

- 2.1 Définition d'un jeu de test (scénario) pour exciter à la fois le modèle de simulation et le système réel.
- 2.2 Détermination des paramètres temporels intrinsèques (par opposition aux paramètres de fonctionnement) des composants matériels du système de commande par « recalage » entre le comportement du modèle simulé (par exemple avec le simulateur CPNTools) et le comportement observé du réel.
- 2.3 Validation du modèle paramétrique du système par confrontation entre les performances estimées par simulation et celles mesurées sur un système réel (plateforme de test).
- 2.4 Identification d'un ou plusieurs chemins critiques (ordonnancement d'événements conduisant à un temps de réponse maximal).

Tâche 3 : Vérification par analyse avec model-checking paramétré

- 3.1 Analyse du modèle contraint à un chemin critique avec le model-checker paramétrique (par exemple Hytech ou Phaver) à partir des données issues de la tâche 2.4.
- 3.2 Construction d'une stratégie de parcours permettant d'obtenir les relations entre paramètres de délai du modèle.
- 3.3 Fourniture d'une première relation entre paramètres pour déterminer l'influence de la charge du contrôleur (ou de sa surcharge due à la panne d'un autre contrôleur) sur les performances de bout en bout du modèle (temps de réponse).
- 3.4 (En parallèle aux sous-tâches 3.1 à 3.3) Analyse « classique » du modèle par model-checking sur les modèles aux paramètres instanciés, et par model-checking sur modèle paramétré sans apport de la simulation.

Tâche 4 : Validation, comparaison et interprétation des résultats

- 4.1 Validation de l'approche par model-checker paramétrique et simulation en observant sur le système réel si les performances attendues sont atteintes pour le fonctionnement dans les plages de paramètres vérifiées.
- 4.2 Comparaison entre les méthodes d'analyses : par model-checking sur modèles aux paramètres instanciés, par model-checking sur modèle paramétrés, par model-checking sur modèle paramétré mais dirigé par résultats de simulation.
- 4.3 Interprétation détaillée du domaine de variation des paramètres qui garantit le temps de réponse sur la surcharge admissible du contrôleur en mode dégradé (cas de la panne d'un autre contrôleur de l'architecture).

Apport des partenaires

Coté LSV :

- expertise dans la modélisation de systèmes concurrents et temps-réel par automates temporisés,
- expertise en méthodes de vérifications basées sur des modèles formels, notamment les systèmes temporisés paramétrés.

Coté LURPA :

- expertise en modélisation des systèmes de commande en réseaux et des propriétés attendues, et en analyse de modèles formels instanciés,
- plate-forme expérimentale d'automatisation en réseau avec des entrées et des sorties distribuées (partiellement financée par le BQR ENSC 2003) et plate-forme de mesure de performances temporelles PRISME (Brevet ENSC 2001) pour nos résultats au réel.

Organisation pratique

Chacune des 4 tâches sera réalisée conjointement par l'ensemble des participants du projet. Un rythme bimensuel de réunion est prévu afin d'échanger régulièrement les résultats liés à l'avancement du projet.

Description détaillée du projet

Cette section décrit plus précisément les hypothèses de travail ainsi que la conduite des différentes tâches et sous-tâches présentées précédemment.

Tâche 1 : Modélisation d'un sous-ensemble d'une architecture de commande distribuée

La catégorie d'architecture de commande que nous retenons (figure 3) a les caractéristiques suivantes :

- l'architecture est *répartie*. Elle comprend :
 - un ensemble de contrôleurs logiques à moniteur mono tâche cyclique ;
 - un ensemble d'entrées sorties déportées accessibles par tous les contrôleurs ;
 - un réseau de communication périodique assurant pour chaque contrôleur la mise à jour des entrées sorties distribuées.
- les fonctions de commande sont de type *logique* (par opposition à analogique)
- l'affectation des fonctions de commande est *statique* et *redondante*.

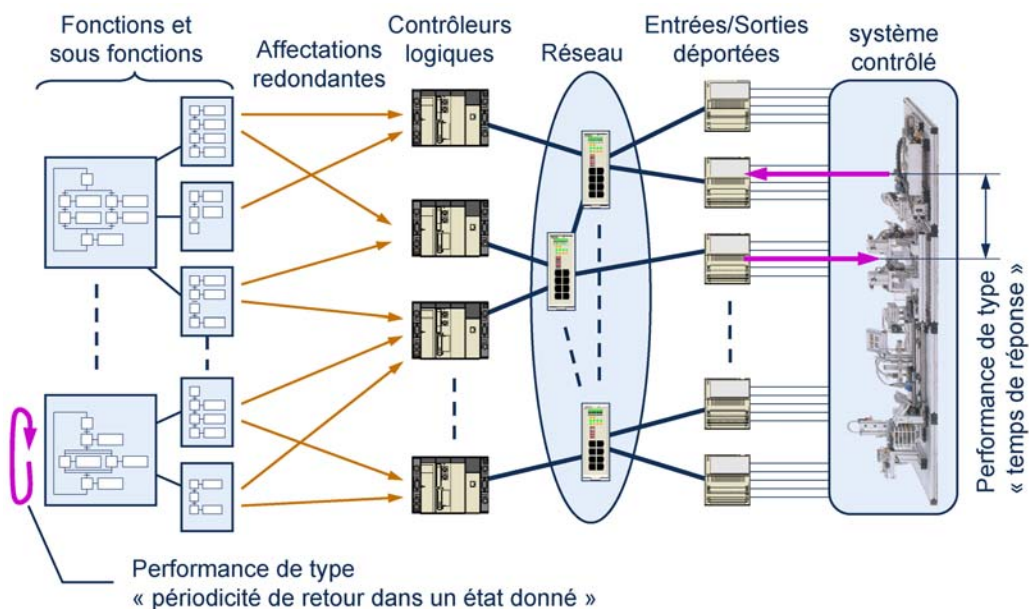


Figure 3 : systèmes de commande distribués retenus

Les types des performances que l'on peut souhaiter sur ces architectures sont :

- *le temps de réponse à une sollicitation du système commandé*, échelle de temps millisecondes, pour un mode de fonctionnement donné ;
- *la périodicité de retour dans un état interne défini* (périodicité de production d'un nouveau produit), échelle de temps en secondes.

Dans cette étude, nous nous limiterons à l'analyse du temps de réponse.

Les tâches 1.1 à 1.4 visent à définir un cas d'étude précis sur lequel l'analyse sera menée, ainsi que les formalismes mis en jeu. Pour l'étude concernée, le système sera composé d'un contrôleur muni de fonctions qui lui sont propres et de fonctions à exécuter en mode dégradé, d'un réseau d'interconnexion et d'un module d'entrées-sorties (figure 4). Le reste de l'architecture sera pris en compte dans les modèles sous la forme de charge, aussi bien pour l'activité du réseau, que pour l'activité du processeur des contrôleurs).

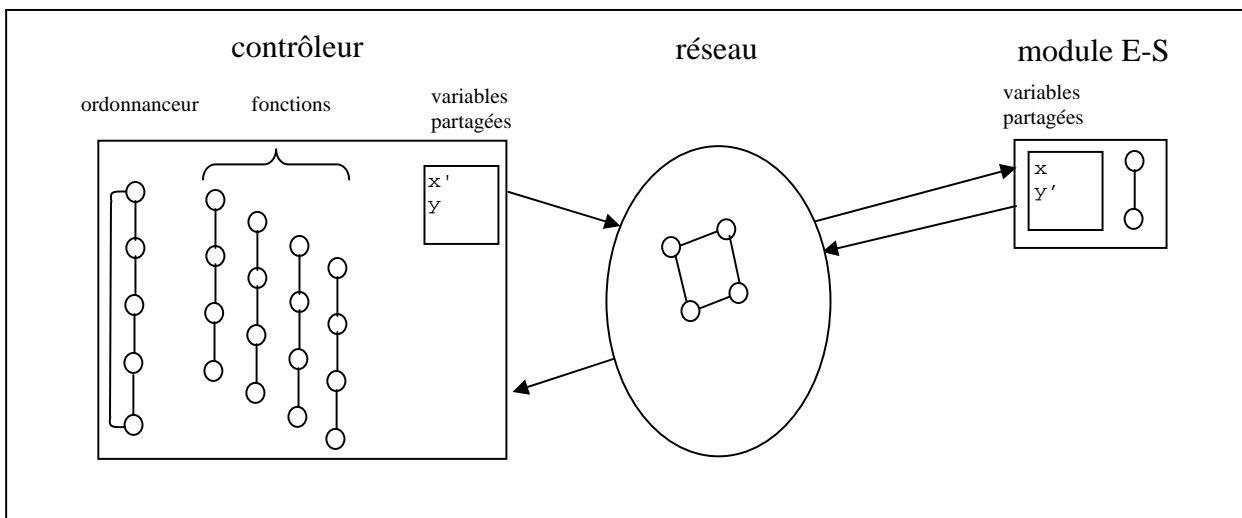


Figure 4 : description schématique du système étudié

Schématiquement, le module E-S produit une information x issue d'un capteur, qui est transmise via le réseau vers le contrôleur et écrite dans la zone des variables partagées de ce dernier (x'). Lorsqu'elle sera élue, cela dépend de la position courante de l'ordonnanceur, la fonction scrutant la zone de variables partagées x effectuera le traitement approprié à la valeur de x' , et produira un résultat y lui-même écrit dans la zone partagée. Ce résultat sera transmis via le réseau vers le module E-S, où il sera écrit dans la zone de variables partagées (y'), il sera utilisé comme commande d'actionneur par le système en aval du module E-S.

On s'intéressera plus particulièrement au temps de réponse de ce système, c'est-à-dire au temps séparant l'événement « écriture de x par le module E-S » dans sa zone de variables partagées et l'événement « écriture de y' par le réseau d'interconnexion » dans la zone de variables partagées du module E-S. Ce temps dépend grandement de la charge du réseau ainsi que de sa politique d'attribution du médium de communication à ces abonnés, de la charge du contrôleur (notamment de son mode de fonctionnement : nominal ou mode dégradé, de la durée d'exécution des autres fonctions implantées sur le même contrôleur, ...). L'analyse devra envisager ces différents facteurs comme des grandeurs paramétrées.

Tâche 2 : Analyse par simulation

Objectif principal de cette tâche est d'identifier les séquences critiques d'évènements qui conduisent le temps de réponse à sa valeur limite admissible (performance attendue) pour les utiliser comme données d'entrée du model-checker paramétrique. Contrairement à des estimateurs comme la médiane ou la moyenne, le maximum est une caractéristique délicate à estimer par simulation pour une performance dont on n'a pas de modèle de la distribution. La construction de scénarii pertinents (représentativité, taille, ...) pour les classes de systèmes et de performances étudiées fera l'objet d'une attention particulière. Pour cela on s'appuiera sur des travaux menés antérieurement au LURPA [Meunier 06].

Un certain nombre de paramètres du modèle sont des caractéristiques temporelles intrinsèques aux composants de l'architecture de commande, comme par exemple le temps nécessaire à une variable d'un module d'entrées-sorties pour traverser la pile du protocole de communication du module. Contrairement aux paramètres de fonctionnement qui dépendent de la façon dont les composants sont configurés et sollicités, les paramètres intrinsèques sont des données constantes du système. La valeur de ces paramètres est nécessaire dans toutes les phases du projet travaillant sur les modèles instanciés, or elles ne sont pas toutes disponibles auprès des constructeurs des dits composants. La simulation sera donc utilisée dans un premier temps pour déterminer la valeur de ces paramètres par un processus d'identification entre le comportement du modèle simulé et le comportement observé sur le composant réel en fonctionnement isolé. La plate-forme de mesure de performances temporelles PRISME du LURPA sera utilisée à cet effet.

Pour valider le modèle paramétrique de l'architecture de commande et en particulier le niveau d'abstraction retenu (tâche 1.4) la simulation va également être utilisée. Ce qui doit être validé, c'est que le modèle représente fidèlement les performances attendues du système. On cherchera donc à comparer les performances temporelles estimées par simulation du modèle aux performances mesurées sur le système réel en fonctionnement. Cette comparaison devra s'effectuer pour des instances représentatives du modèle paramétrique puisqu'un simulateur ne pourra pas manipuler de paramètres symboliques mais seulement des paramètres valués. Les résultats de la comparaison pourront, si nécessaire, donner lieu à des améliorations du modèle.

Dans les sous-tâches qui viennent d'être exposées, la simulation a été employée comme outil d'assistance à la conception d'un modèle paramétrique fidèle. C'est à partir de ce modèle paramétrique validé que la démarche d'analyse qui fait l'originalité de ce projet va prendre place. Il s'agit maintenant d'identifier un ou plusieurs chemins critiques qui conduisent à un temps de réponse maximal du système de commande distribué, ces chemins critiques étant une des données d'entrée de la technique de model-checking paramétré utilisée en tâche 3. La simulation va donc être utilisée pour « animer » le modèle selon un scénario qui comporte un grand nombre de sollicitations au niveau du module déporté d'entrées-sorties. Dans cette simulation on identifiera la ou les sollicitations qui ont produit le temps de réponse maximal et l'on extraira les séquences d'évènements ayant conduit à ces réponses (chemins critiques).

Toutes les simulations seront des simulations à événements discrets, et les outils associés à ces activités devront satisfaire des contraintes parfois antagonistes, permettre des simulations longues et permettre une interactivité suffisante avec l'expert pour identifier des parcours critiques. Le simulateur de Réseaux de Petri Temporisé de haut niveau CPNTools est pressenti.

Tâche 3 : Vérification par analyse avec model-checking paramétré

Le problème de *séparation d'événements* pour des modèles ayant des délais compris dans des intervalles entiers est très difficile : [McMillan 92] ont montré que c'était un problème NP-Complet. De nombreuses approximations polynomiales ont été proposées ([Chakraborty 97], [Belluomini 00]). L'introduction de paramètres ajoute une difficulté supplémentaire.

Plutôt que de considérer le problème de calcul du temps de réponse dans le cas général, on s'intéresse au problème inverse : étant donné le temps de réponse du système dans une configuration donnée, comment extraire de façon automatique un ensemble de contraintes suffisantes sur les paramètres du modèle (les délais correspondant à différentes charges du réseau ou du contrôleur) garantissant que le temps de réponse puisse effectivement être atteint. Ce choix consistant à restreindre l'ensemble des solutions en se focalisant sur un chemin critique particulier permet de traiter des systèmes de grande complexité qui ne sont pas à la portée des outils de vérification lorsque le problème de séparation d'événement est traité dans toute sa généralité. L'exploitation de cet ensemble de contraintes permet de comprendre l'impact des différents facteurs sur l'élaboration du temps de réponse global.

L'analyse par model-checking paramétré consiste à élaborer un modèle sous forme d'automates temporisés dans lequel les temporisations sont des paramètres, puis à développer des algorithmes permettant d'extraire les contraintes algébriques entre les paramètres garantissant des temps de réponse. Ces algorithmes combinent des parcours de l'espace des états du système lorsque les paramètres sont instanciés, et des parcours lorsque le système est paramétré (mais contraint par des relations déduites des parcours instanciés).

Une première étude menée sur un composant mémoire embarquée développée par STMicroelectronics a montré la faisabilité de l'approche, et nous souhaitons appliquer cette démarche sur une architecture plus complexe, et surtout plus flexible.

Tâche 4 : Validation, comparaison et interprétation des résultats

Dans un premier temps la pertinence des résultats obtenus sera quantifiée par rapport au comportement réel du système de test. On observera en particulier si les performances sont garanties pour les plages de paramètres obtenues par model-checker paramétré dirigé par les résultats de simulation. Comme le système réel en fonctionnement correspond à une instance particulière des paramètres, la validation d'une plage de paramètres nécessitera l'observation d'un certain nombre de configurations du système. On choisira ces configurations de préférences sur la frontière du domaine de variation des paramètres, car ce sont celles dont le temps de réponse est sensé être maximal (d'après le modèle). Le respect systématique des propriétés attendues sur le temps de réponse validerait notre approche combinée model-checking paramétré et simulation.

L'apport de l'approche combinée model-checking paramétré et simulation sera abordé en menant une comparaison par rapport aux approches classiques model-checking paramétré et model-checking instancié. Les critères de comparaison seront : la portée des résultats obtenus (limitée à un cas particulier ou valable pour une plage de paramètre) et l'effort de calcul nécessaire.

Finalement, les résultats fournis par le model-checking paramétré seront interprétés vis-à-vis des systèmes de commande répartis. Le premier bénéfice attendu est de pouvoir s'assurer que les temps de réponses d'une architecture sont inférieurs à une limite donnée. Cependant, la sémantique des résultats d'un model-checker paramétré est bien plus riche que la réponse binaire « propriété satisfaite ou non ». On peut s'attendre à des possibilités d'exploitation des résultats en termes de

réglage des paramètres de fonctionnement du mode nominal, ou en termes d'estimation du niveau de surcharge admissible en mode dégradé par le contrôleur qui pallie la défaillance d'un autre.

Bibliographie

- [Thomesse 99] J.P. Thomesse. « Fieldbuses and interoperability ». *Control Engineering Practice* N°7, 1999.
- [Jasperneite 02] J. Jasperneite, P. Neumann, M. Theis, and K. Watson. « Deterministic Real-Time Communication with Switched Ethernet ». *Proceedings of 24th IEEE WFCS*, 2002.
- [Vitturi 03] S. Vitturi. « DP-Ethernet: the Profibus DP protocol implemented on Ethernet ». *Computer Communications* N°26, 2003.
- [Meunier 06] P. Meunier. « Évaluation de performance d'architectures de commande de systèmes automatisés industriels ». Thèse de l'École Normale Supérieure de Cachan, 2006.
- [Marsal 05] G. Marsal, D. Witsch, B. Denis, J.-M. Faure, and G. Frey. « Evaluation of Real-Time Capabilities of Ethernet-based Automation Systems using Formal Verification and Simulation ». *Proceedings of RJCITR'05*, 2005.
- [Abdulla 99] P. Abdulla, A. Annichini, and A. Bouajjani. « Symbolic Verification of Lossy Channel Systems: Application to the Bounded Retransmission Protocol ». *Proceedings of TACAS'99, LNCS 1579*, 1999.
- [Bérard 99] B. Bérard and L. Fribourg. « Automated Verification of a Parametric Real-Time Program: The ABR Conformance Protocol ». *Proceedings of CAV'99, LNCS 1633*, 1999.
- [Maler 95] O. Maler and A. Pnueli. « Timing Analysis of Asynchronous Circuits using Timed Automata ». *Proceedings of CHARME'95, LNCS vol 987*, 1995.
- [Bozga 02] M. Bozga, H. Jianmin, O. Maler, and S. Yovine. « Verification of Asynchronous Circuits using Timed Automata ». *Proceedings of TPTS'02, ENTCS vol 65*, 2002.
- [Clariso 04] R. Clariso and J. Cortadella, « Verification of Timed Circuits with Symbolic Delays ». *Proceedings of ASP-DAC'04*, 2004.
- [Chevallier 06] R. Chevallier, E. Encrenaz, L. Fribourg, and W. Xu. « Verification of the Generic Architecture of a Memory Circuit using Parametric Timed Automata ». *Proceedings of FORMATS'06, LNCS vol 4202*, 2006.
- [McMillan 92] K. McMillan and D. Dill. « Algorithms for Interface Timing Specification ». *Proceedings of ICCD*, 1992.
- [Chakraborty 97] S. Chakraborty and D. Dill. « Approximate Algorithms for Time Separation of Events ». *Proceedings of ICCD*, 1997.
- [Belluomini 00] W. Belluomini and C. Myers. « Timed State Space Exploration using POSETS ». *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, 19(5), 2000.

Production scientifique

Rapports internes : 1 à mi-projet, 1 final.

Plusieurs communications scientifiques dans des congrès et revues d'informatique et d'automatique des systèmes à événements discrets.

Demande financière

- Calculateur dimensionné pour la simulation et le model-checking équipé de : 8,0 k€
 - un processeur de 64 bits (pour repousser les limites en calcul symbolique),
 - 8 Go de RAM (pour repousser les limites de l'explosion combinatoire),
 - 1 disque dur haut débit (pour stocker les fichiers de trace sans ralentir la simulation)
- Licences pour des logiciels de visualisation des résultats 1,0 k€
- Frais de mission (3 manifestations, 2 participants) 9,0 k€

Total de la demande financière : **18,0 k€**