

Projet BIP

Contrôle/commande de robots marcheurs et applications

Rhône-Alpes

THÈME 4A

R *apport*
d'Activité

2001

Table des matières

1	Composition de l'équipe	3
2	Présentation et objectifs généraux	3
3	Fondements scientifiques	4
3.1	Les robots bipèdes	4
3.1.1	Motivations	4
3.1.2	Problématique scientifique	5
3.2	Analyse, modélisation et commande de systèmes dynamiques non-réguliers . . .	6
3.2.1	Généralités	6
3.2.2	Applications	7
3.2.3	Problèmes ouverts	8
3.3	Contrôle/commande temps-réel de systèmes robotiques complexes	8
3.3.1	Programmation des contrôleurs hybrides	8
3.3.2	L'environnement de conception ORCCAD	8
3.3.3	Développements futurs et problèmes ouverts	9
4	Domaines d'applications	10
5	Logiciels	11
6	Résultats nouveaux	13
6.1	Modélisation et commande	13
6.1.1	Génération de trajectoires de marche 3D	13
6.1.2	Commande optimale contrainte avec horizon de commande	13
6.1.3	Activités sur le prototype BIP	14
6.1.4	Systèmes mécaniques non-réguliers	14
6.1.5	Planification de trajectoires pour la locomotion bipède en environnement 3D partiellement structuré	14
6.1.6	Autoroutes automatisées	15
6.1.7	Véhicules autonomes	16
6.2	Aide à la conception de systèmes temps-réel de contrôle/commande	16
6.2.1	Tolérance aux pannes	16
6.2.2	Recherche de chemins disjoints dans un réseau de processeurs	17
6.2.3	Répartition automatique de programmes synchrones	17
6.2.4	Génération automatisée de contrôleurs	18
6.2.5	Langages de programmation d'automates industriels	18
6.2.6	Vérification de logiciel périodique multi-cadence	19
6.2.7	Ordonnancement régulé	19

7 Contrats industriels (nationaux, européens et internationaux)	20
7.1 TELEDIMOS	20
7.2 Schneider Electric	20
7.3 ST Microelectronics	21
7.4 Transfert industriel	21
8 Actions régionales, nationales et internationales	21
8.1 Actions nationales	21
8.1.1 GdR-PRC "Commande de robots à pattes"	21
8.1.2 Groupe COSED	21
8.1.3 Projet Robea Electrostimulation des paraplégiques	21
8.1.4 Pre-projet Robea Commande des Hélicoptères	21
8.1.5 Collaborations internes à l'Inria	22
8.1.6 Collaborations avec d'autres laboratoires	22
8.2 Actions européennes	22
9 Diffusion de résultats	23
9.1 Animation de la Communauté scientifique	23
9.2 Enseignement	24
10 Bibliographie	24

1 Composition de l'équipe

Responsable scientifique

Eric Rutten [CR Inria]

Assistante de projet

Françoise de Coninck [à temps partiel dans le projet]

Personnel Inria

Bernard Brogliato [DR, depuis le 1er sept. 2001]

Bernard Espiau [DR, à temps partiel]

Alain Girault [CR]

Philippe Sardain [en délégation de l'IUT de Poitiers, jusqu'au 31 août 2001]

Daniel Simon [CR]

Ingénieurs experts

Claire Gignoux [Projet TELEDIMOS, depuis le 1er mars 2001]

Olivier Testa [Ingénieur associé, depuis le 1er sept. 2001]

Chercheurs doctorants

Christine Azevedo [Boursier MENRT]

Jean-Matthieu Bourgeot [Boursier MENRT, depuis le 1er oct. 2001]

Antonio Medina-Rodriguez [Boursier CONACYT (Mexique)]

Chercheurs post-doctorants

Vincent Acary [Post-doc. CNRS-Schneider Electric, depuis le 1er sept. 2001]

Karine Altisen [Post-doc. UR, à partir du 1er déc. 2001]

Remco Leine [Post-doc. TU Eindhoven, depuis le 1er sept. 2001]

Stagiaires

Fanny Benattar [février-juin 2001, DEA IVR]

Jean-Matthieu Bourgeot [février-juin 2001, DEA RVMSC, Versailles]

Fabien Lydoire [février-juin 2001, DEA IVR]

James Laine [février-juin 2001, Ingénieur/IUP La Rochelle]

Clément Ménier [stagiaire d'été ENS Lyon]

Prajna Prakash Parida [mai-juillet 2001, BsC I.I.T. Kharagpur, Inde]

Jean-Philippe Roussel [février-juin 2001, Ingénieur ESSAIM, Mulhouse]

Collaborateur extérieur

Nathalie Cislo [SPM/UFRAPS, UJF]

2 Présentation et objectifs généraux

Le projet a pour objectif l'étude générique des divers aspects intervenant dans le contrôle/commande des systèmes robotiques complexes. Les activités du projet se déclinent selon deux grands thèmes scientifiques : la modélisation et la commande de systèmes marcheurs d'une part ; l'étude de méthodes et d'outils pour le contrôle/commande temps-réel d'autre part.

L'intérêt des systèmes marcheurs réside dans leur capacité d'adaptation à des terrains ou des sols variés, leur permettant de se déplacer dans des milieux très contraints en dimensions, voire mal structurés. Les robots anthropomorphes sont donc particulièrement aptes à évoluer

dans nos environnements courants, privés ou industriels, essentiellement conçus pour la bipédie. Ainsi, les domaines d'application visés sont-ils en priorité les robotiques personnelle, de service et d'intervention. Parallèlement, le projet s'attache à développer des activités de modélisation dans certains domaines de la biomécanique. Enfin, le projet ne souhaite pas que les applications de ses recherches se limitent au domaine des seuls robots marcheurs, dont le marché industriel est actuellement marginal. C'est pourquoi les techniques étudiées sont voulues suffisamment génériques pour faire l'objet de mises en œuvre dans d'autres domaines (voir la section 4).

Le contrôle/commande temps-réel de systèmes robotiques complexes relève de la problématique plus générale de la modélisation et programmation de systèmes mêlant aspects de commande (continue échantillonnée) et de contrôle (discret). On les trouve en robotique, mais aussi en avionique, dans l'automobile et dans les automatismes de production. Ils sont réactifs, ont des modes multiples, des cadences multiples, et doivent être tolérants aux pannes. Nous abordons ces problèmes sous l'angle des méthodes formelles, notamment l'approche synchrone. Dans un souci d'utilisabilité par des spécialistes du domaine cible, et non de celui des techniques formelles, nous proposons des langages spécialisés pour la spécification de missions, de tâches, et de propriétés. Ce dernier point est lié à l'utilisation de méthodes de vérification formelle. La tolérance aux pannes est abordée comme un problème d'ordonnancement réparti avec réplication. Enfin, la génération automatisée de contrôleurs est fondée sur l'utilisation de la synthèse de contrôleurs discrets. L'environnement ORCCAD intègre certaines des solutions proposées.

Parmi ses relations internationales, le projet BIP est actif dans le projet Européen TELE-DIMOS, et le réseau CLAWAR.

3 Fondements scientifiques

3.1 Les robots bipèdes

Mots clés : commande, commande référencée capteur, mécanique, mécanique des solides, modélisation, robotique, robotique mobile, simulation de système mécanique, temps-réel, asservissement visuel.

3.1.1 Motivations

Dans la classe des systèmes mobiles, les robots marcheurs, par exemple hexapodes, présentent des avantages certains sur leurs homologues à roues dès que le sol n'est plus plan ou libre : le franchissement des obstacles est plus aisé, l'emprise au sol plus faible, l'adaptabilité meilleure. Ceci concerne les grands domaines de la robotique non-manufacturière : exploration, maintenance, intervention, service. Cependant, dès lors que l'environnement de travail du système est conçu pour l'homme, la technologie multipode doit en général laisser place à la bipédie si l'on désire conserver de bonnes possibilités de déplacement et d'action sans modifier l'environnement. D'où l'intérêt assez récent que porte la communauté mondiale de Recherche et Développement en robotique aux systèmes dits humanoïdes, destinés à accompagner l'homme dans certaines de ses activités personnelles ou professionnelles. Par exemple, une certaine forme d'assistance à domicile de personnes à mobilité réduite, pour des tâches très

routinières, pourrait être assurée par des robots bipèdes, car ceux-ci ont la faculté de pouvoir se déplacer sans adaptation particulière de l'environnement.

Si la faisabilité de tels systèmes reste largement hors d'atteinte pour ce qui est de robots aux capacités d'autonomie décisionnelle élevées, le niveau actuel de la technologie permet par contre d'envisager la réalisation de machines capables de se déplacer en marchant dans des conditions bien déterminées et d'exécuter quelques actions très simples.

3.1.2 Problématique scientifique

Conception générale. La difficulté de conception d'un robot marcheur bipède résulte, d'une part de la nouveauté du concept, qui fait qu'il n'existe pas encore de solutions éprouvées, et d'autre part de contraintes spécifiques au principe même de la marche bipédique : tout d'abord, le système doit comporter un nombre élevé d'articulations dans un espace de volume réduit. De plus, la localisation de celles-ci est elle-même fort contrainte par des raisons de position des articulations, d'encombrement et de répartition des masses, ce qui nécessite par exemple l'étude de groupes moto-transmetteurs spécifiques. Par ailleurs, le système doit être en permanence en équilibre, soit statique, soit dynamique, ce qui peut nécessiter des couples instantanés élevés. Ceux-ci se répartissent d'ailleurs au cours du temps sur l'ensemble des articulations, posant ainsi le problème d'une gestion globale et dynamique de la puissance nécessaire. Ce dernier point est également lié aux choix technologiques concernant l'autonomie : la place disponible est faible et la masse embarquable limitée (elle se situe en général en hauteur (tronc) et a de ce fait une grande influence sur la dynamique du système).

Ce problème reste donc largement ouvert, l'état de l'art en la matière étant constitué par les très belles réalisations de Honda, les robots humanoïdes P2 et P3 présentés en 1997.

Modélisation. Un robot bipède peut être modélisé comme une chaîne articulée arborescente de corps rigides dans l'espace tridimensionnel. La marche est caractérisée par différentes phases [VDO92,Win90], principalement (pour une jambe donnée) le balancement (35 % du cycle) et le support (65 % du cycle). Il existe de plus une phase de double support qui dure environ 12 % du cycle et qui est absente dans la course. Ces phases se décomposent elles-mêmes plus finement en fonction d'événements liés au mouvement du centre de gravité et, surtout, du pied. Ces différentes phases se caractérisent par des contacts divers entre le système et le sol.

Ainsi, le modèle mécanique d'un tel système comporte-t-il trois aspects :

- une dynamique de système rigide articulé libre dans l'espace représentable par des équations lagrangiennes ;
- un ensemble, variable selon les phases, de contraintes d'égalités et d'inégalités traduisant l'existence de contacts sans pénétration et sans glissement que *doit* respecter le système ; chaque ensemble définit un mode de fonctionnement ;
- un choix de loi d'impact modélisant les transitions (supposées instantanées) entre modes.

Nous sommes donc en présence d'un système dynamique hybride complexe, dont l'étude reste encore peu explorée.

[VDO92] C. L. VAUGHAN, B. L. DAVIS, J. C. O'CONNOR, *Dynamics of Human Gait*, Human Kinetic, 1992.
[Win90] D. A. WINTER, *Biomechanics and Motor Control of Human Movement*, Wiley-Interscience, 1990.

Commande. Un marcheur humain adopte naturellement une allure régulière et symétrique dite *de confort* dans laquelle sa consommation d'énergie est minimale pour une vitesse de progression raisonnable. Par ailleurs, certaines classes de systèmes mécaniques hybrides, par exemple un robot sauteur ou une transmission avec jeu, présentent souvent des comportements cycliques limites correspondant à des équilibres dynamiques, parfois mêmes naturellement stables dans un certain domaine. Dans le cas du robot marcheur le plus simple, le compas sur une pente, ces cycles correspondent à des trajectoires périodiques passives (sans actionnement extérieur), dans lesquelles la transition entre énergie cinétique et énergie potentielle est entièrement équilibrée par l'absorption d'énergie à l'impact [GTE98].

Tout ceci nous incite donc à rechercher une approche de la commande dans laquelle les trajectoires sont cycliques et énergétiquement optimales, en un sens qui reste à préciser. Un autre point est que ce qui importe vraiment est la progression globale du robot tout en préservant une stabilité mécanique particulière qui est de nature dynamique. Les approches classiques, comme le suivi précis de trajectoires articulaires figées sont donc inadaptées, sauf si l'on s'intéresse principalement au contrôle de la posture. Le domaine n'étant pas lui-même stabilisé (et loin s'en faut), le chercheur explorera donc des techniques de commande variées : optimisation non-différentiable, commande prédictive, commande par apprentissage itératif, commande par fonction de tâches [6]... On n'oubliera pas enfin la commande référencée capteurs [3], qui permettra de prendre en compte des mesures locales de distance, de proximité, de réaction au sol...

3.2 Analyse, modélisation et commande de systèmes dynamiques non-réguliers

Mots clés : systèmes mécaniques, impacts, contraintes unilatérales, complémentarité, modélisation, analyse, simulation, commande, analyse convexe..

3.2.1 Généralités

Les systèmes dynamiques (de dimension finie ou infinie, mais nous nous contenterons de systèmes de dimension finie) sont dits non-réguliers à partir du moment où leur modèle comporte des ingrédients impliquant une non-différentiabilité de l'état. Cette non-différentiabilité peut provenir de différentes sources (par exemple une excitation extérieure de type impulsive, qui conduit à ce que l'on appelle des équations différentielles à mesure). Une classe importante de tels systèmes est composée des systèmes dynamiques de complémentarité, qui sont d'une façon générique décrits comme suit :

[GTE98] A. GOSWANI, B. THUILOT, B. ESPIAU, « A Study of the Passive Gait of a Compass-like Biped Robot: Symmetry and Chaos », *Int. Journal of Robotics Research* 17, 12, décembre 1998.

$$\left\{ \begin{array}{l} \dot{x} = f(x, u, \lambda) \\ \mathcal{C}^* \ni y \perp \Lambda(\lambda) \in \mathcal{C} \\ g(y, \lambda, u, x, t) = 0 \\ \text{Loi de ré-initialisation de l'état } x(\cdot) \end{array} \right. \quad (1)$$

où \mathcal{C}^* et \mathcal{C} sont deux cones pôlaires, \perp est l'orthogonalité, λ est un multiplicateur, u une entrée de commande, $g(\cdot)$ et Λ des fonctions. Ces systèmes dynamiques peuvent être considérés de deux points de vue différents : le point de vue *systèmes dynamiques hybrides* et le point de vue *inclusions différentielles*. L'approche hybride provient du fait qu'il est naturel de définir différents *modes* de fonctionnement à partir de $y_i = 0$ ou $y_i > 0$, où y_i est une composante du vecteur y . L'approche par inclusions différentielles provient basiquement de l'équivalence entre les conditions de complémentarité $0 \leq y \perp \lambda \geq 0$ et l'inclusion $-\lambda \in \partial\psi_{\mathbb{R}^+}(y)$, où $\psi_{\mathbb{R}^+}(\cdot)$ est la fonction indicatrice de \mathbb{R}^+ (qu'on peut en mécanique assimiler à un potentiel "unilatéral"), et ∂ est sa sous-différentielle au sens de l'analyse convexe. On voit donc aisément à partir de (1) que cette dynamique peut se ré-écrire comme une inclusion. Lorsque le passage d'un mode à l'autre ne peut se faire qu'avec un saut dans l'état $x(\cdot)$, alors la dynamique continue de (1) comporte une mesure de Dirac, et Moreau définit alors le terme d'inclusion différentielle à mesure. Ceci dit, tous les systèmes de complémentarité ne sont pas à mesure, certains ayant une solution $x(\cdot)$ absolument continue.

Dans (1), les deuxième et troisième lignes définissent donc les modes du système hybride ainsi que les conditions pour que les transitions entre ces modes se produisent, alors que la quatrième ligne définit la façon dont les transitions sont effectuées par l'état continu $x(\cdot)$.

3.2.2 Applications

Les systèmes (physiques ou abstraits) qui peuvent être représentés par des modèles tels que (1) sont nombreux : Systèmes mécaniques avec contraintes unilatérales et frottement sec (le robot bipède est un exemple typique, chaînes cinématiques avec jeu, toutes tâches robotiques comportant du contact unilatéral, phénomènes de *liquid slosh* etc), circuits électriques avec diodes et/ou transistors MOS idéaux, commande optimale avec contraintes sur l'état, boucle fermée d'un système commandé par un algorithme MPC, etc.

Pour ces systèmes une approche unifiée peut se révéler très utile afin d'élucider certaines de leurs propriétés liées à la commande (stabilisation, commandabilité, observabilité, identification), ou bien les problèmes plus mathématiques (existence, unicité, dépendance continue par rapport aux conditions initiales, bifurcations), voire pour la simulation numérique qui est évidemment un aspect très important.

L'intérêt de cette classe de modèles réside principalement dans le fait qu'elle n'est pas trop large, donc permet des études poussées, mais est en même temps assez riche pour inclure de nombreuses applications. Il ne semble pas raisonnable d'attaquer le problème des systèmes dynamiques hybrides dans sa généralité, par contre se concentrer sur des sous-classes de tels systèmes se révèle très fructueux. Par exemple il est important de noter que (1) est un système

hybride “continu”, dans le sens que ce sont les variables continues x et u qui sont prépondérantes dans l'évolution (par exemple il n'y a pas de commande discrète qui permettrait de commuter d'un mode à l'autre : seul le signal u peut être utilisé).

3.2.3 Problèmes ouverts

Les problèmes ouverts sont nombreux. Nous nous concentrons essentiellement sur les aspects numériques et de commande, avec quelques excursions vers des problèmes liés à la nature des solutions (bifurcations, problèmes de Painlevé dans les systèmes mécaniques avec frottement sec).

3.3 Contrôle/commande temps-réel de systèmes robotiques complexes

Mots clés : architecture de contrôle, contrôle/commande, système embarqué, programmation synchrone, vérification formelle, environnement de programmation, temps-réel, système hybride.

3.3.1 Programmation des contrôleurs hybrides

Un des problèmes principaux de la programmation des systèmes robotiques est d'aborder dans un cadre cohérent la description de l'évolution dynamique des deux composantes – continue et discrète – de ces systèmes hybrides. Ceci nécessite d'intégrer différents formalismes, trop souvent spécialisés dans un domaine précis et déconnectés les uns des autres. Ainsi, l'étude des *architectures de contrôle* est primordiale pour développer concepts et méthodes permettant de gérer la complexité de la spécification et de l'analyse de la globalité du système.

Un contrôleur de robots doit être capable de traiter une grande diversité d'actions, allant par exemple de la commande d'une cellule flexible d'assemblage en milieu industriel au contrôle d'un véhicule autonome opérant en milieu hostile. La diversité des applications potentielles en robotique et le désir de mettre en œuvre des algorithmes de commande sophistiqués impliquent une ouverture du système de programmation. D'un point de vue industriel, les contrôleurs actuels restent le plus souvent des machines fermées, dotées d'un langage limité ne permettant pas d'exécuter des actions complexes telles que des commandes référencées capteurs. Outre leurs limitations fonctionnelles, ils ne permettent pas de traiter correctement les aspects temps-réel sous-jacents à toute application robotique, même simple.

3.3.2 L'environnement de conception ORCCAD

L'environnement de contrôle ORCCAD [7, 1] propose actuellement deux niveaux : le niveau *fonctionnel*, où les aspects commande en temps discrétisé sont prépondérants, et le niveau *contrôle*, qui gère principalement les aspects réactifs. Au niveau fonctionnel, les actions élémentaires sont définies sous forme de Tâches-Robot : ce concept clé spécifie dans la même entité une partie algorithmique (loi de commande) et une partie logique réactive. Il s'agit d'un objet informatique hybride, à l'interface des aspects temps continu et discret. En l'absence de modèles unifiés et opérationnels des systèmes hybrides, cette approche permet de cantonner l'interface temps continu/discret dans la coquille réactive de la Tâche-Robot. Les actions

élémentaires exécutées par le robot sont réalisées par l'exécution périodique d'une loi de commande en boucle fermée dont la conception repose sur les techniques de l'automatique classique comme la spécification par fonction de tâches [6], la commande référencée capteurs ou encore la commande instationnaire pour systèmes non holonomes. Cette loi de commande est *encapsulée* dans un comportement logique réactif rythmant son exécution à l'aide de signaux typés (pré-conditions, exceptions et post-conditions) constituant sa *vue externe*. Ce comportement réactif est codé à l'aide du langage synchrone ESTEREL, ce qui permet de bénéficier des outils de vérification formelle associés FC2TOOLS/ATG.

Au niveau contrôle, les actions complexes sont spécifiées sous la forme de Procédures-Robot par composition structurée et hiérarchique de Tâches-Robot, manipulées par l'intermédiaire de leur vue externe, et de traitements d'exceptions. À la différence de ces dernières, les Procédures-Robot n'ont pas de partie commande, autre que celle des Tâches-Robot qui les composent. Nous pouvons ainsi définir de façon cohérente et dans un même formalisme des actions de complexité croissante, gérant les modes nominaux d'une mission robotique complexe ainsi que les modes dégradés dépendant des exceptions spécifiées et du contexte d'exécution. Tous ces aspects logiques sont également spécifiés en ESTEREL du niveau le plus haut (mission) jusqu'aux appels système gérant l'exécution des lois de commande, un soin tout particulier étant pris quant à la gestion des transitions entre commandes successives. Ceci nous permet d'effectuer *a posteriori* une vérification formelle des programmes produits à un grand niveau de détail.

La formalisation des structures manipulées présente de nombreux avantages permettant la définition d'un environnement de programmation cohérent et convivial [SKE98] :

- ceci permet de concevoir des interfaces de programmation adaptées à chaque type d'utilisateur du système. Ainsi, l'automaticien chargé du développement d'une Tâche-Robot peut en coder le comportement réactif sans rien connaître à ESTEREL, la vue externe de la Tâche-Robot étant générée automatiquement à partir d'une fenêtre graphique. D'autre part, le concepteur de mission ne code que la partie spécifique de son application, l'essentiel du code source, et en particulier celui permettant la connexion avec le système d'exploitation sous-jacent, étant généré automatiquement de façon totalement transparente ;
- l'identification de certaines propriétés génériques d'un système ainsi structuré (vivacité, sûreté) permet d'automatiser partiellement la génération de critères abstraits facilitant ainsi l'utilisation des outils de vérification ;
- le code temps-réel téléchargeable est automatiquement généré à partir des spécifications de haut niveau.

3.3.3 Développements futurs et problèmes ouverts

- De nombreux travaux restent cependant à réaliser et certains problèmes restent ouverts :
- en tant qu'elle vise la conception de systèmes embarqués temps-réel à sûreté critique, la génération de code doit être améliorée afin de produire du code réparti optimisé et tolérant aux pannes ;

[SKE98] D. SIMON, K. KAPellos, B. ESPIAU, « Formalization of hybrid structures in robot controllers: the Orccad approach », *in*: 9th Symposium on information control in manufacturing (INCOM'98), IFAC, Nancy, France, juin 1998.

- l'utilisation des méthodes de vérification formelle actuelles reste difficile pour un non spécialiste du domaine, notamment en ce qui concerne le diagnostic et l'interprétation des résultats. Nous nous heurtons également rapidement à des problèmes de taille d'automates à vérifier, même si la méthodologie d'ORCCAD permet d'effectuer une spécification et des vérifications incrémentales. La prise en compte de contraintes temporelles n'est pas actuellement réalisée ;
- enfin, il manque au niveau supérieur une couche *décisionnelle* de type planification réactive, permettant une éventuelle reprogrammation partielle de la mission en cas de défaillance non spécifiée ou encore de spécifier des missions complexes de façon conviviale en évitant des fautes ou oublis de spécification. Nous fondons de sérieux espoirs sur l'utilisation de techniques de synthèse de contrôleurs pour systèmes à événements discrets, initialisés par Ramadge et Wonham ^[RW87], pour construire ce niveau sur un formalisme mathématique solide.

Nous proposons ainsi, dans un environnement cohérent, un ensemble d'outils intégrés pour le support de ces méthodes et concepts bien formalisés ; ils permettent de programmer, de valider et de générer du code, de manière transparente pour l'utilisateur.

4 Domaines d'applications

Mots clés : robotique, système embarqué, mécanique, robot marcheur.

Les domaines d'applications des robots marcheurs sont principalement ceux de la robotique non-manufacturière. Plus précisément, le projet viserait idéalement la robotique d'assistance (personnes âgées, handicapés) et d'intervention en milieu hostile terrestre, comme le nucléaire, en tant que secteurs applicatifs. Le premier domaine est également concerné par nos activités en biomécanique ; celles-ci ont aussi des applications dans le domaine du matériel à usage sportif. Les techniques de commande référencée vision sans étalonnage étudiées dans le projet trouvent quant à elles des applications dans d'autres domaines, dont la robotique manufacturière (manipulation, soudage, inspection) et les robots mobiles.

Les travaux concernant la programmation sûre sont d'une portée très large puisque s'appliquant à une large classe de systèmes critiques. Sont visés en particulier les systèmes présentant une forte interaction avec l'environnement, par le biais de nombreux capteurs et actionneurs, comme les véhicules automatiques ou les engins sous-marins autonomes. Le domaine d'application peut s'élargir aux systèmes de contrôle/commande pour la production, et aux automates programmables, comme le suggèrent les contacts industriels liés à ORCCAD.

Les applications concernant les systèmes dynamiques non-réguliers sont nombreuses, en mécanique mais aussi en électronique. Un domaine très important est le prototypage virtuel, qui nécessite des algorithmes de simulation adéquats. De tels systèmes ne peuvent être intégrés par des méthodes classiques pour l'intégration des ODE ou des DAE. Par exemple les accumulations finies d'événements, qui sont fréquentes en mécanique (infinité de rebonds en temps fini), requièrent des techniques très spécifiques.

[RW87] P. RAMADGE, W. WONHAM, « Supervisory control of a class of discrete event processes », *SIAM Journal of Control and Optimization* 25, janvier 1987, p. 206–223.

5 Logiciels

Participants : Soraya Arias [Moyens robotiques, correspondant], Claire Gignoux, Daniel Simon, Olivier Testa.

Mots clés : robotique, contrôle/commande, temps-réel, spécification, vérification.

Résumé : ORCCAD (*Open Robot Controller Computer Aided Design*) est un environnement logiciel permettant de concevoir et de mettre en œuvre le contrôle et la commande d'un système robotique complexe. Il permet également la spécification et la validation des missions à réaliser par ce système.

Fonctionnalités d'ORCCAD. ORCCAD est principalement destiné aux applications temps-réel critiques en robotique, dans lesquelles les aspects relevant de l'automatique (les asservissements, les commandes) sont amenés à interagir étroitement avec ceux manipulant des événements discrets. De tels systèmes sont souvent qualifiés d'hybrides. ORCCAD est conçu et développé en commun entre les projets BIP et ICARE (Sophia Antipolis) et le service des Moyens Robotiques de l'UR Rhône-Alpes. Ses principaux constituants sont :

- un éditeur de composants logiciels élémentaires (Modules)
- un éditeur graphique pour la spécification de Tâches-Robot
- un éditeur de Procédures-Robot, générant automatiquement la plus grande partie du code de contrôle ESTEREL
- un langage de spécification de missions robotiques (MAESTRO)
- une interface de vérification générant des critères abstraits pour les outils FC2TOOLS/ATG
- une connexion avec des simulateurs logiques (XES) ou hybrides (SIMPARC)
- un générateur de code temps-réel pour VXWORKS, SOLARIS ou LINUX

Version diffusée. ORCCAD V3.1 est distribué sur une demi-douzaine de sites académiques à des fins d'évaluation. Une présentation du logiciel est disponible à l'URL : <http://www.inrialpes.fr/iramr/pub/Orccad/>

Cette version dispose des fonctionnalités suivantes : un générateur de code pour le simulateur SIMPARC (Jean-Jacques Borrelly, INRIA-SOPHIA-ANTIPOLIS), un exécuteur pour VXWORKS 5.2 et 5.3, pour SUNOS 5.6 et 5.8 ainsi que pour LINUX RED HAT 6.1.

ORCCAD présente une partie graphique développée sous Ilog Views 3.11. Il est disponible à la fois sous SUNOS 2.6 et sous LINUX (RedHat 6.1). Il supporte la version 5.21 (et sup.) d'ESTEREL ainsi que des outils associés (ATG, XES) et compatibles avec cette version.

Nouveaux développements. Cette année ont été mis en chantier les projets suivants :

- développement sous JAVA de l'interface graphique du logiciel.
- Simulateur temps-réel générique.

Un patron (template) de simulateur générique a été développé pour faciliter la génération de simulateurs d'autres mécanismes : il intègre certaines classes de Dynamechs (<http://dynamechs.sourceforge.net/>), un logiciel de génération de modèles dynamiques de

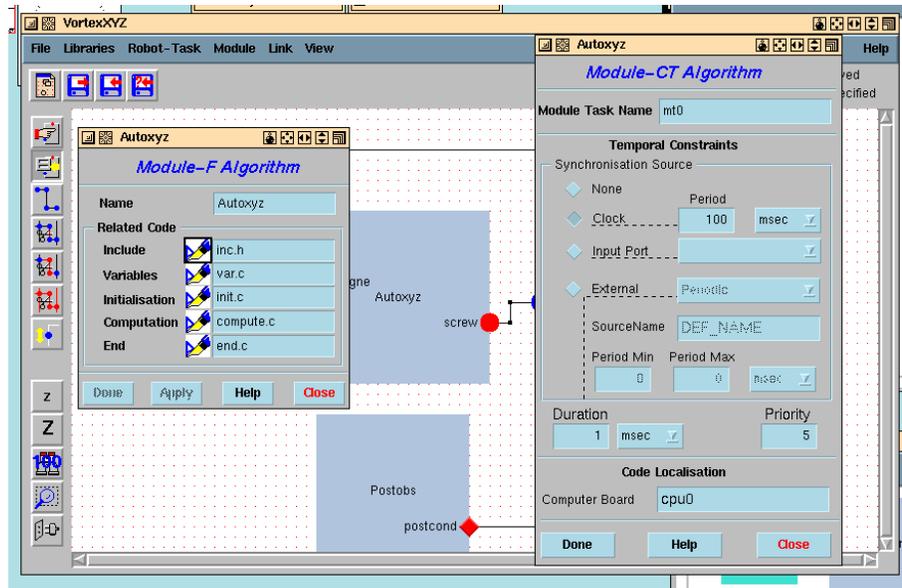


FIG. 1 – Éditeur de Tâches-Robot

robots à base fixe ou mobile, éventuellement soumis à des efforts hydrodynamiques, et pouvant comporter des arborescences et des boucles cinématiques fermées. La description du mécanisme est simplement donnée dans un fichier ASCII, l'utilisateur n'ayant plus qu'à intégrer sa loi de commande dans une tâche-robot pour obtenir un simulateur.

- Tâches-Robot multi-cadences. Le générateur de code a été étendu afin de pouvoir spécifier des cadences d'exécution différentes pour les modules algorithmiques d'une même loi de commande. Un module peut être synchronisé soit sur une horloge système, soit par la production d'une sortie par le module le précédant dans le chemin d'exécution. Les différents modules peuvent être synchronisés plus ou moins lâchement, de façon à minimiser les durées de latence en évitant les inter-blocages. Un mécanisme à tampons multiples est inséré automatiquement pour protéger les variables partagées de façon asynchrone [38].
- Vérification de tâches temps-réel multi-cadences. Un module logiciel permet de générer à partir de la description d'une tâche-robot un modèle de vérification temporelle sous forme de graphe d'événements temporisés. Ce modèle représente une implantation mono-processus. Il peut être analysé par un nouveau module de ERS implantant le modèle de tâches périodiques et de préemption utilisant l'algèbre (max,plus). L'analyse produit une analyse d'interblocages, un test de respect d'échéances et un diagramme d'activités des tâches [36].
- Portage sur RTAI. Le générateur de code exécutable est en cours de portage sur RTAI (<http://www.aero.polimi.it/projects/rtai/>), une variante temps-réel et open source de Linux. Des systèmes d'analyse de traces d'exécution (lxt) et de débogage dans l'espace utilisateur (lxrt) peuvent être associés au code téléchargeable.

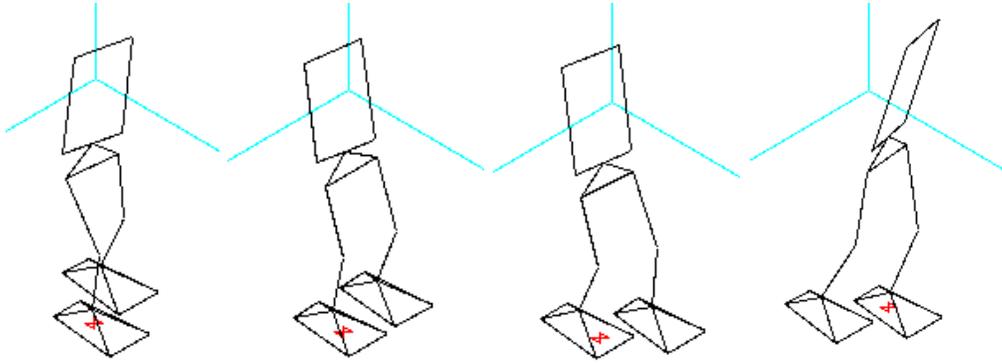


FIG. 2 – Exemple de démarche obtenue par la génération de trajectoires optimales

6 Résultats nouveaux

6.1 Modélisation et commande

Participants : S. Arias [M.R.], C. Azevedo, J.M. Bourgeot, B. Brogliato, C. Canudas de Wit [LAG], N. Cislo, B. Espiau, A. Girault, F. Lydoire, H. Mathieu [M.R.], P. Poignet [LIRMM], P. Sardain [LMS], J.-P. Roussel, P.-B. Wieber.

6.1.1 Génération de trajectoires de marche 3D

Ce travail concerne la génération de démarches paramétrées pour le robot BIP. Ces trajectoires sont les solutions d'un problème d'optimisation minimisant un critère énergétique soumis à des contraintes. La paramétrisation des démarches se fait sur des paramètres tels que : longueur de pas, hauteur d'obstacle à franchir... La méthode a permis de calculer des mouvements articulaires décrivant un enchaînement de pas séparés par des phases de double-support (voir Fig. 2) [39]. Une documentation sur l'utilisation du simulateur est en cours de rédaction.

6.1.2 Commande optimale contrainte avec horizon de commande

La méthode développée consiste à calculer en-ligne le vecteur de commande à envoyer à chaque instant aux actionneurs du robot de manière à ce qu'il puisse se déplacer. Ce vecteur permet de minimiser un critère énergétique et satisfait un ensemble de contraintes de type in-égalité. L'optimisation est réalisée à chaque instant sur un horizon de prédiction. L'originalité réside dans le fait qu'aucune trajectoire n'est fournie à l'algorithme. De plus un jeu unique de contraintes est fourni et permet de marcher normalement ou d'enjamber des obstacles en fonction des événements. Les tests sur une perturbation de type poussée dans le dos sont encourageants. Les premiers résultats de simulation ont permis de démontrer la faisabilité d'une telle méthode (voir Fig. 3) [19].

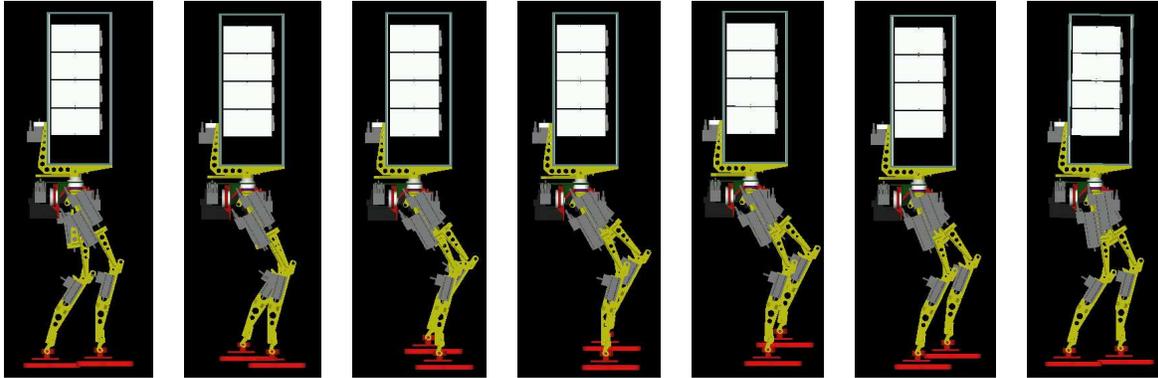


FIG. 3 – *Démarche obtenue avec la méthode d'optimisation contrainte avec horizon de commande*

6.1.3 Activités sur le prototype BIP

Le prototype du bipède exposé à Hanovre a été remis comme convenu au LMS de Poitiers. La formation de P. Sardain et J.P. Gazeau du LMS à l'utilisation de la plate-forme expérimentale BIP et aux outils développés a été réalisée en collaboration avec le service Moyens Robotiques.

Des documentations techniques sont réalisées concernant le contrôleur du robot BIP2000 [31] ou les outils de simulation [35].

6.1.4 Systèmes mécaniques non-réguliers

Les résultats et travaux récents portent sur quatre points principaux :

- L'extension du théorème de la passivité lorsque l'on considère des non-linéarités multivoques [21]. Ce thème fait l'objet d'un travail en commun avec le projet Numopt.
- Un survol et la présentation d'un cadre général d'étude pour les robots bipèdes [33]. Ceci s'insère dans un cadre plus général d'étude des systèmes dynamiques de complémentarité [12, 13]. Notons que les travaux sur les hélicoptères [18] vont aussi faire intervenir de tels modèles pour les phases d'atterrissage et de décollage.
- L'étude des bifurcations introduites par les paradoxes de Painlevé dans des systèmes mécaniques simples avec une contrainte unilatérale et frottement de Coulomb [34].
- L'application du contact de Hertz au calcul de coefficients de restitution analytiques en élastique, visco-élastique, visco-elasto-plastique. L'extension au cas des chocs multiples. Ceci en vue d'améliorer le logiciel de simulation de Schneider Electric.

6.1.5 Planification de trajectoires pour la locomotion bipède en environnement 3D partiellement structuré

Les résultats obtenus en génération de trajectoire pour la marche statiquement stable et pour le contrôle postural ont servi de point de départ pour considérer la planification de trajectoires en environnement 3D. La planification de trajectoires, généralement envisagée

dans des environnements très structurés ou dans le cadre de mouvements unidirectionnels, a été considérée dans un environnement 3D partiellement structuré, et en multidirectionnel.

Deux approches complémentaires ont été explorées afin de tenir compte des capacités de franchissement d'un robot bipède en fonction du terrain sur lequel il évolue.

Une première méthode [22] repose sur une discrétisation du terrain en portions planes d'altitudes variables.

Une deuxième approche [37], associée à une modélisation du terrain sous forme de plans inclinés, aborde la planification en deux étapes :

- premièrement, le calcul d'une trajectoire de référence en utilisant des techniques classiques empruntées à la planification des robots mobiles ; la trajectoire de référence est calculée de façon à minimiser les inclinaisons du terrain et le roulis du robot le long du parcours ;
- deuxièmement, une méthode de planification locale autour d'une position de référence.

6.1.6 Autoroutes automatisées

Nous poursuivons des travaux commencés pendant qu'Alain Girault était post-doc dans l'équipe PATH de UC Berkeley en 1997. Nous nous intéressons au problème du contrôle hybride de véhicules autonomes roulant sur des autoroutes automatisées. Les véhicules sont autonomes, c'est-à-dire qu'ils ne communiquent ni les uns avec les autres ni avec l'infrastructure. Deux problèmes doivent être résolus : un véhicule roulant sur une voie simple d'autoroute ne doit jamais entrer en collision avec son véhicule leader ; de plus un véhicule arrivant sur l'autoroute à une des entrées doit être capable de s'insérer depuis sa voie d'accélération vers la voie principale, là encore sans qu'il y ait de collision. Afin de résoudre ces deux problèmes, nous proposons d'équiper chaque véhicule d'un contrôleur hybride, constitué de plusieurs lois de commande continues enfouies dans un automate d'états fini. L'automate indique à quel moment un véhicule donné doit entrer sur l'autoroute, s'insérer dans la voie principale, quitter l'autoroute... Les lois de commande continues indiquent quelle accélération le véhicule doit avoir pour éviter les collisions avec ses véhicules voisins. En spécifiant soigneusement ces lois de commande continues ainsi que les conditions gardant les transitions de l'automate, nous avons démontré trois résultats importants [32]. Notre premier résultat établit les conditions initiales sous lesquelles un véhicule suiveur n'entre jamais en collision avec son véhicule leader. Notre second résultat étend le premier à une file de véhicules autonomes. Notre troisième résultat établit que si tous les véhicules sont équipés de notre contrôleur hybride, alors aucune collision ne peut se produire sur l'autoroute, et que tous les véhicules ou bien réussissent à s'insérer avec succès, ou bien sont obligés de quitter la voie d'accélération quand ils en atteignent la fin. Enfin, nous avons réalisé la microsimulation d'une autoroute, programmée avec le simulateur hybride SHIFT développé par l'équipe PATH de l'Université de Berkeley.

6.1.7 Véhicules autonomes

Suite à l'étude théorique précédemment publiée sur une loi de commande longitudinale pour véhicules autonomes [GY99], nous avons mis en œuvre cette loi de commande sur les véhicules CYCAB de l'INRIA Rhône-Alpes [42]. La programmation a été faite avec l'environnement ORCCAD. Cette mise en œuvre a posé plusieurs difficultés :

- Au bas niveau, le CYCAB est commandé en vitesse alors que notre loi de commande calcule une accélération. Il a donc fallu insérer un intégrateur en sortie.
- Le capteur frontal du CYCAB est un télémètre laser qui fournit la distance le séparant du véhicule leader. Or notre loi de commande a besoin non seulement de la distance séparant le véhicule du véhicule leader, mais aussi de la vitesse relative. Il a donc fallu insérer un dérivateur en entrée.
- La distance fournie par le télémètre laser est bruitée, ce qui rend son utilisation brute, et encore plus l'utilisation de sa dérivée, impossible. Il a donc fallu insérer un filtrage afin de lisser les données.

Une fois ces difficultés surmontées, nous avons testé notre loi de commande avec deux CYCABS (un leader et un suiveur) sur le parking de l'INRIA Rhône-Alpes. Ces tests ont été concluant et ont donné lieu à plusieurs démonstrations.

6.2 Aide à la conception de systèmes temps-réel de contrôle/commande

Participants : H. Alla [LAG], K. Altisen, S. Arias [M.R.], F. Benattar, C. Dima [VERIMAG], A. Girault, T. Grandpierre [SOSSO, INRIA-ROCQUENCOURT], F. Jiménez-Fraustro [EP-ATR, INRIA-RENNES], H. Marchand [EP-ATR, INRIA-RENNES], A. Medina, C. Ménier, P. P. Parida, C. Pinello [UC Berkeley, USA], E. Rutten, M. Sighireanu [LIAFA, Paris], D. Simon, Y. Sorel [SOSSO, INRIA-ROCQUENCOURT].

6.2.1 Tolérance aux pannes

Nous poursuivons nos travaux commencés dans le cadre de l'ARC TOLERE [23, 24, 25]. Le but global est la génération automatique de code réparti tolérant aux pannes. Nous nous sommes plus particulièrement intéressés aux opérations d'entrée/sortie. Pour ces opérations, le principe de base de notre méthode, consistant à répliquer les opérations, ne marche pas. Nous avons proposé le mécanisme suivant pour les opérations d'entrée (symétrique pour les opérations de sortie) : chaque opération d'entrée est connectée à n capteurs, et doit recevoir au moins $m \leq n$ valeurs pour s'exécuter correctement. Ce mécanisme de type m -sur- n permet de spécifier beaucoup plus finement l'interface entre l'application et son environnement. Les problèmes sur lesquels nous travaillons à présent concernent le déterminisme des valeurs produites par plusieurs réplicas d'opérations d'entrée, ainsi que sur la séparation de l'architecture cible en deux ou plus sous-architectures disjointes suite à la panne d'un processeur et/ou d'un moyen de communication.

[GY99] A. GIRAULT, S. YOVINE, « Stability Analysis of a Longitudinal Control Law for Autonomous Vehicles », in : *IEEE Conference on Decision and Control, CDC'99*, IEEE, Phoenix, USA, décembre 1999. RESEARCH REPORT INRIA 3498.

6.2.2 Recherche de chemins disjoints dans un réseau de processeurs

Nous considérons un réseau de processeurs reliés par des liaisons point-à-point. Notre but est de trouver tous les chemins de communication disjoints partant d'un processeur source donné et aboutissant à un processeur destination donné. Par disjoint nous entendons que deux chemins ne doivent avoir en commun aucun processeur autre que le processeur source et le processeur destination. Notre motivation est la *tolérance aux pannes*, avec l'idée qu'un des chemins ainsi trouvés sera le chemin principal, alors que les autres chemins seront les chemins de secours.

Il existe deux classes d'algorithmes pour trouver les chemins disjoints dans un graphe : les algorithmes *globaux* qui nécessitent la connaissance complète de la topologie du réseau, et les algorithmes *locaux* qui ont besoin uniquement que chaque processeur connaisse tous ses voisins. Un autre point de vue est qu'un algorithme global doit être exécuté *hors-ligne*, alors qu'un algorithme local peut très bien être exécuté *en-ligne*, et donc dans un contexte où le réseau est *dynamique* (panne dynamique des processeurs et des liens de communication, possibilité de rajouter dynamiquement des nouveaux processeurs...). L'algorithme que nous avons mis au point est un algorithme local. Nous envisageons dans l'avenir d'en évaluer la complexité pratique. Pour cela, il faudra en programmer une simulation et générer aléatoirement plusieurs réseaux de processeurs pour produire des « benchmarks ».

6.2.3 Répartition automatique de programmes synchrones

D'une façon générale, nous travaillons à obtenir des mises en œuvre réparties de programmes synchrones, et sur des problèmes connexes tels que l'optimisation de programmes répartis [26, 14] ou la mise en œuvre logicielle [16].

Dernièrement nous avons plus particulièrement travaillé sur l'extension de résultats antérieurs sur la répartition de programmes synchrones [2] aux nouvelles formes de compilation du langage ESTEREL. La motivation est que les travaux précédents sur la répartition se basaient sur le format intermédiaire en automate d'états fini, format qui ne permet pas de compiler les gros programmes à cause de l'explosion combinatoire liée au parallélisme. Ce format est communément appelé OC. Donc concrètement, nous avons travaillé sur deux autres formats intermédiaires : les circuits séquentiels d'une part (appelé SC), et le format propre au compilateur de France Telecom R&D d'autre part (appelé CP). Ces trois formats intermédiaires diffèrent de par la nature de la structure de contrôle des programmes. Ceci est résumé dans le tableau suivant :

format	structure du contrôle		
OC	séquentielle	explicite	statique
CP	parallèle	explicite	statique
SC	parallèle	implicite	dynamique

Nous avons donc proposé un algorithme de répartition de programmes SC, qui fonctionne sur le même principe que celui pour les programmes OC [40]. A cause de la nature parallèle et dynamique de la structure de contrôle des programmes SC, cet algorithme est plus basique, c'est-à-dire qu'il réalise beaucoup moins d'optimisations sur le nombre et le placement des

émission et réceptions de messages entre les sites de calcul. Cet algorithme a été mis en œuvre dans l'outil SCREP. Nous avons de plus proposé un algorithme de répartition de programmes CP qui généralise celui pour les programmes OC afin de traiter les branchements parallèles. En effet les programmes OC ne contiennent que des branchements de type `if then else`.

6.2.4 Génération automatisée de contrôleurs

Nous poursuivons notre exploration de méthodes d'application de la synthèse de contrôleurs discrets, dans le cadre de la programmation de systèmes robotiques, ou plus généralement de contrôle-commande. L'approche est de définir des éléments de base de construction de modèles, facilement composables, et présentant les états et événements contrôlables qui permettent d'assurer des objectifs de base. Dans la perspective d'une utilisation automatisée de la synthèse de contrôleur discret, un utilisateur définit son système en composant de tels modules de base, choisit ses objectifs à partir de propriétés qui lui sont proposées, et peut alors obtenir un contrôleur correct, sans que tout ceci ne requière une compétence forte en modèles et algorithmes de synthèse.

Plus précisément, nous cherchons à mettre au point des schémas génériques de tâches inspirés de ceux d'ORCCAD, dans le sens où nous nous plaçons au même niveau d'abstraction des tâches : un automate local de contrôle de phases de commande [29]. Ces tâches présentent des contraintes quant à leur activation, que ce soit pour chaque actionneur, entre les tâches qui peuvent le commander (une et une seule doit être active à la fois), ou entre différents sous-systèmes (modes de fonctionnement exclusifs pour des raisons de partage de ressource ou d'incompatibilité due à leur effet dans l'environnement). Ces propriétés peuvent être particulières à une architecture de systèmes, indépendamment d'une mission donnée, ou au contraire liées à un enchaînement de tâches particulier qu'on veut obtenir.

Ce travail se fait en coopération avec le projet EP-ATR, IRISA/INRIA-Rennes (Hervé Marchand). Concrètement, notre approche repose sur l'utilisation de l'outil de calcul formel SIGALI. C'est l'outil de vérification par *model-checking* associé au langage synchrone et environnement de programmation SIGNAL [15]. Il est étendu à des fonctionnalités de synthèse de contrôleurs discrets, exploitant des notions de hiérarchie et d'optimalité. Nous avons traité un exemple inspiré des systèmes d'excavation étudiés dans le projet TELEDIMOS [41].

Nous nous intéressons à une étude de la définition de tâches multi-modes, qui modélisent la commutation entre des implémentations différentes d'une même fonction, caractérisées par des critères comme le coût en temps de calcul, ou l'énergie consommée, et la qualité du résultat fourni [28]. Sur ces bases nous utiliserons la synthèse optimale pour générer des contrôleurs qui assurent par exemple que la somme des temps de calcul parallèles n'excède pas la période globale, tout en maximisant la qualité. Nous nous intéressons aussi à la prise en compte d'aspects de mise en œuvre et de tolérance aux pannes. Dans le cadre d'une coopération avec le LAG (H. Alla), nous étudions ces aspects ainsi que des questions plus fondamentales sur les modèles.

6.2.5 Langages de programmation d'automates industriels

Un modèle a été finalisé de langages de la norme IEC 61131 en termes du langage synchrone SIGNAL ; ce travail, en coopération avec le projet EP-ATR, IRISA/INRIA-Rennes, a donné lieu

à la soutenance de la thèse de F. Jiménez [8], co-encadrée par E. Rutten et P. Le Guernic. Ce modèle repose sur une traduction structurelle, qui fait usage des particularités de SIGNAL, notamment le sur-échantillonnage : ce dernier permet de construire un modèle dont la granularité est assez fine pour distinguer les calculs internes des interactions avec l'environnement (lecture des entrées, écritures des sorties), et qui permet de prendre en compte des programmes comportant des itérations non-bornées [27]. Ce faisant nous réutilisons des concepts mis au point en relation avec la modélisation synchrone de STATECHARTS [11].

6.2.6 Vérification de logiciel périodique multi-cadence

Les modules communiquants et synchronisés de façon plus ou moins lâches utilisés pour construire les commandes de programmes ORCCAD s'implantent sous forme de tâches temps-réel généralement périodiques, gérées par un système d'exploitation à priorités fixes. Cet ensemble de tâches périodiques peut être modélisé sous forme d'un Graphes d'Événements Temporisés (GET), ayant un modèle linéaire dans l'algèbre $(\max, +)$. L'ordonnanceur temps-réel est représenté par une fonction de contraction du temps appliquée aux tâches préemptées ce qui permet de garder une structure de GET et donc d'appliquer les résultats classiques de $(\max, +)$.

Tous les GET ne représentent pas le fonctionnement d'un système temps-réel : en particulier, pour représenter l'implantation du système de tâches sur une machine d'exécution, il faut ajouter au modèle des contraintes assurant qu'au plus une seule transition temporisée par processeur ne soit active à un instant donné. Les conflits entre tâches d'égale priorité sont également traités par de nouvelles contraintes de synchronisation. Un modèle de vérification correspondant au cas mono-processeur peut être alors automatiquement généré à partir de la description d'une tâche-robot spécifiée sous ORCCAD [36].

Un certain nombre de problèmes relatifs au comportement temporel de l'ensemble de modules peuvent alors être résolus de façon analytique : nous pouvons prouver que le système atteint bien un régime périodique après une période transitoire, dont nous pouvons calculer la durée. La durée du cycle en régime permanent est également calculable. Enfin le respect des échéances temporelles de chacune des tâches peut aussi être prouvée [10]

6.2.7 Ordonnement régulé

Les applications de contrôle/commande, que l'on trouve par exemple en robotique, nécessitent d'exécuter des activités informatiques ayant des caractéristiques temporelles différentes. On trouve par exemple :

- des lois de commande périodiques, où la valeur de contraintes de cadence et de latence conditionne les performances de la commande ;
- des lois de commande périodiques multi-cadence, où l'on peut affecter des caractéristiques temporelles (période, latence...) différentes à certaines parties de l'algorithme pour en augmenter l'efficacité ;
- des activités, répétitives ou non, dont la durée d'exécution dépend de l'environnement et n'est pas connue a priori (planification, traitement d'images...);

- des activités sporadiques, représentant par exemple les changements d'état et de configuration du système, les traitements d'exceptions ou les réactions aux pannes.

Ces divers processus partagent un support d'exécution de capacité bornée en devant respecter des contraintes temps-réel, spécifiées sous la forme d'échéances temporelles à respecter. Le respect de ces échéances peut être strict (tout dépassement est interdit) ou lâche (les commandes périodiques peuvent tolérer sans dommages des variations autour de valeurs nominales de périodes et de latences, sans doute grâce à leur marge de phase).

Les méthodes classiques d'ordonnancement à priorités fixes ne sont pas capables de traiter efficacement les aléas temporels, et leur stricte application conduit généralement à une sous-utilisation importante de la puissance de calcul, sans robustesse vis à vis des erreurs de modélisation conduisant à des surcharges. L'affectation de priorités aux processus, fonction de leur urgence ou de leur importance relative, reste un problème délicat. D'autre part, une caractéristique intéressante de ces systèmes est d'être, au moins en partie, des systèmes fonctionnant en boucle fermée. Une approche récente et prometteuse consiste à réguler en boucle fermée les paramètres d'ordonnancement en fonction de mesures effectuées sur le processus commandé (e.g. l'erreur de poursuite) et/ou sur son contrôleur (e.g. la charge de calcul).

La validité de ce concept est en cours d'évaluation dans le projet, sur un exemple que l'on peut considérer comme canonique en robotique : la commande d'un bras manipulateur, où l'on constate expérimentalement que le calcul d'un modèle dynamique explicite du mécanisme peut être exécuté à une fréquence inférieure à celle de la chaîne de commande directe. Il n'y a pas de solution théorique connue pour calculer une valeur optimale ou même raisonnablement efficace à la valeur de ces fréquences, que l'on pense pouvoir être ajustées par le régulateur. Les expériences en cours, utilisant une variante de contrôleur généré par ORCCAD, doivent également permettre d'évaluer les contraintes liées à ce concept, notamment concernant le sur-coût induit par la méthode et sa complexité de mise en œuvre.

7 Contrats industriels (nationaux, européens et internationaux)

7.1 TELEDIMOS

BIP et MOVI font partie du consortium IST TELEDIMOS (*Telepresence simulation platform supporting civil works machines in remote dismantling waste disposal and large scale demining operations*). Ce projet de 2 ans a démarré en janvier 2000. Il s'agit d'un projet de R&D destiné à intégrer une plateforme de téléopération autour d'une excavatrice, les applications visées étant le démantèlement d'usines nucléaires et l'intervention en travaux publics. Nous y sommes chargés de la réalisation d'un simulateur temps-réel du robot et de l'intégration d'un système de reconstruction 3D par vision stéréo. Le consortium réunit des industriels (Hellenic Technology of Robotics, Hellenic Technodomiki, Techniki Demosion, British Nuclear Fuels, JC Bamford) et des laboratoires de recherche (CRIIF, EARLR et INRIA).

7.2 Schneider Electric

V. Acary dans le cadre de son post-doc co-encadré par B. Brogliato et M. Abadie (Schneider Electric) sont impliqués dans un contrat post-doc CNRS-Schneider Electric, sur la simulation

des disjoncteurs moyenne et basse tension.

7.3 ST Microelectronics

Nous sommes en contact avec des équipes de R&D de ST Microelectronics à Crolles. On étudie les potentiels de l'approche synchrone sur le flot de conception chez STMicroelectronics, particulièrement du point de vue de l'apport de la synthèse de contrôleurs discrets dans le cas de systèmes multi-tâches.

7.4 Transfert industriel

Le projet participe au transfert du savoir-faire de l'INRIA en contrôle/commande temps-réel, en particulier autour d'ORCCAD, vers la société ATHYS, *start-up* créée en juillet 2000.

Il est aussi en contact avec des équipes de R&D de France Telecom.

8 Actions régionales, nationales et internationales

8.1 Actions nationales

8.1.1 GdR-PRC "Commande de robots à pattes"

J.M. Bourgeot participe au projet GdR-PRC du CNRS "Commande de robots à pattes".

8.1.2 Groupe COSED

A. Medina et E. Rutten participent à un groupe de travail COSED de l'EEA consacré à la commande opérationnelle des systèmes à événements discrets¹. Il succède, en élargissant et renouvelant la thématique, au groupe consacré au Grafcet et aux formalismes pour les automatismes industriels.

8.1.3 Projet Robea Electrostimulation des paraplégiques

Le projet BIP (C. Azevedo, B. Brogliato) participe au projet Robea "Contrôle du mouvement du membre inférieur humain paralysé sous stimulation électrique" dirigé par D. Guiraud, avec le LIRRM et le Centre Propara (Montpellier, PUPH Rabishong). BIP apporte son expertise sur la commande des systèmes mécaniques non-réguliers. La collaboration comporte également une étude sur la modélisation des muscles pour la commande, en vue de simulations de la marche sur le robot bipède de l'INRIA. P.B. Wieber (projet Numopt) participe activement à cette activité.

8.1.4 Pre-projet Robea Commande des Hélicoptères

B. Brogliato est impliqué dans le pré-projet Robea "Modélisation et Commande d'Hélicoptères" qui regroupe Heudiasyc de l'UT Compiègne (R. Lozano), l'Université d'Evry (T. Hamel), l'Université d'Amiens (C. Pegard). Des contacts avec l'Onera de Salon de Provence

¹<http://japura.lurpa.ens-cachan.fr/cosed>

(A. Desopper) sont également établis sur ce thème, et un sujet commun de DEA concernant la commande des phases d'atterrissage et de décollage a été proposé pour 2001/2002.

8.1.5 Collaborations internes à l'Inria

- Service des Moyens Robotiques de l'UR, sur ORCCAD et sur la réalisation du bipède.
- Projet MOVI, dans le projet TELEDIMOS.
- Projet SOSSO dans la suite de l'ARC TOLERE, et en lien avec les programmes français AEE (Architecture Electronique Embarquée) et européen EAST-EEA.
- Projet EP-ATR, INRIA-RENNES, sur :
 - l'utilisation de méthodes et outils de synthèse de contrôleurs discrets (H. Marchand),
 - la modélisation synchrone des langages de programmation d'automates pour le contrôle/commande (F. Jiménez et P. Le Guernic).
- Avec le projet TRIO, INRIA-LORRAINE, ainsi qu'avec le LIRMM, sur l'utilisation des algèbres $(\max,+)$, dans le cadre de l'action GARP de l'INRIA-LORRAINE.
- Action de recherche coopérative de l'INRIA FISC : formalisation et instrumentation des scénarios ; l'ARC FISC est une action de recherche coopérative de l'INRIA lancée en janvier 2001 et d'une durée de deux ans. Elle regroupe un ensemble d'équipes autour d'un thème commun : les langages de scénarios pour les télécommunications et l'automatique. Quinze chercheurs répartis dans quatre laboratoires participent à l'action. Les laboratoires impliqués sont : l'IRISA (Rennes), le LIAFA (université Paris 7), l'INRIA Rocquencourt et l'INRIA Rhône-Alpes (Grenoble).

8.1.6 Collaborations avec d'autres laboratoires

- La thèse de J.M. Bourgeot, encadrée par B. Brogliato, est co-encadrée par C. Canudas de Wit (LAG).
- Nathalie CISLO (SPM), collaboratrice extérieure de BIP, assure le lien entre les recherches sur la locomotion bipède, humaine ou robotique.
- Eric Rutten coopère avec le LAG (équipe Conception de Systèmes Sûrs, H. Alla), sur le sujet du *Contrôle de Tâches Robotiques : Langages spécialisés et Synthèse de contrôleurs*.
- Eric Rutten coopère avec VERIMAG (F. Maraninchi) sur les automates de modes et la synthèse de contrôleurs.

8.2 Actions européennes

- BIP participe avec MOVI au projet IST TELEDIMOS (Telepresence simulation platform supporting civil works machines in remote dismantling waste disposal and large scale demining operations), qui débuté en janvier 2000. En 2001 ce projet a permis de financer le contrat d'ingénieur-expert de Claire Gignoux pour effectuer le portage d'ORCCAD sur RTAI.
- Le projet BIP est membre depuis l'automne 1998 du réseau européen CLAWAR (Climbing and Walking Robots).
- BIP participe au réseau Européen Artist (*European Network of Excellence on "Advanced Real-Time Systems"*), en particulier à son action intitulée *Flexible Real-Time Systems*

for QoS Management.

9 Diffusion de résultats

9.1 Animation de la Communauté scientifique

B. Espiau a participé au séminaire de Mécanique et Robotique en hommage à B. Roth, à Versailles en juillet 2001, en faisant une présentation intitulée “Modelling and control of anthropomorphic biped robots” ; a été invité à la journée prospective du Sénat sur la robotique, le 27 juin 2001 ; est membre du comité de programme de *ICRA'02*, de la commission d'évaluation de l'Ifremer, et participe au Comité Scientifique “ACI jeunes chercheurs”.

E. Rutten est membre des comités de programme pour *Euromicro Conference on Real-Time Systems*, *ERTS'01* et *ERTS'02*, *Salon Real-Time Systems RTS 2001* et *RTS 2002*, et congrès *MSR 2001 (Modélisation des Systèmes Réactifs)*, et membre des comités d'organisation et de programme de *Slap'02 : Synchronous Languages, Applications, and Programming* ; membre du *BoD* de la société Euromicro ; rapporteur de la thèse de doctorat de Géraud Canet (ENS Cachan), membre des jurys de thèse de doctorat Karine Altisen (UJF/Verimag), Pascal Ogor (Université de Bretagne Occidentale), Camille Schnakenbourg (ENS Cachan) ; membre titulaire de la CS 27 de l'Université de Bretagne Occidentale (Brest).

B. Brogliato est Editeur Associé à Automatica ; membre du comité international de programme de International Conference on Nonsmooth/Nonconvex Mechanics, Greece, 2002 ; réviseur pour Mathematical Reviews (Mécanique des Solides 70EXX, commande des systèmes 93DXX) ; membre des jurys de thèse de O. Janin (ENTPE, Vaulx-en-Velin, juillet 2001) et N. Chriette (CEMIF, Evry, décembre 2001) ; membre titulaire de la CSES 61 de l'Université Claude Bernard Lyon 1 ; membre élu de EUCA (European Union of Control Associations) depuis septembre 2001.

A. Girault est membre des comités d'organisation et de programme de *Slap'02 : Synchronous Languages, Applications, and Programming* et membre du comité d'organisation de *ETAPS'02 : European Joint Conferences on Theory and Practice of Software*. Il a organisé avec A. Benveniste (INRIA-RENNES) une session invitée à la *IEEE Conference on Decision and Control, CDC'01*, Orlando, USA, décembre 2001, intitulée : *Synchronous Programming : A Formal Approach for Specifying and Implementing Automatic Control Software*.

En dehors des conférences citées en bibliographie, les chercheurs du projet ont participé ou présenté leurs travaux dans les manifestations suivantes :

- B. Brogliato a donné une conférence sur les systèmes mécaniques non-réguliers aux Journées Nationales de la Recherche en Robotique, Presqu'île de Giens, octobre 2001, et a organisé un workshop sur ce thème à la conférence European Control Conference, Porto, Septembre 2001.
- C. Azevedo et A. Medina ont participé aux Journées Nationales de la Recherche en Robotique, Presqu'île de Giens, octobre 2001
- A. Medina a participé au (*Symposium on the Supervisory Control of Discrete Event Systems*, SCODES 2001, workshop satellite de la conférence CAV'01.

Le projet a invité à donner un séminaire : Prof. Kazuhiko Terashima et Prof. Hideo Kitagawa (Toyohashi University of Technology, Japon), Karine Altisen et Gregor Gößler (VERIMAG)

9.2 Enseignement

- C. Azevedo : monitorat en Automatique à l'INPG (ENSIEG).
- B. Brogliato : systèmes dissipatifs et commande, 3ème année de l'ENSIEG.
- A. Girault et E. Rutten : compilation, en 2ème année de l'ENSIMAG.
- A. Girault, D. Simon et E. Rutten donnent le cours « Programmation temps-réel et réactive » dans le DEA Image-Vision-Robotique commun à l'INPG et à l'Université Joseph Fourier.

10 Bibliographie

Ouvrages et articles de référence de l'équipe

- [1] J. BORRELLY, E. C. MANIÈRE, B. ESPIAU, K. KAPellos, R. PISSARD-GIBOLLET, D. SIMON, N. TURRO, « The ORCCAD Architecture », *International Journal of Robotics Research* 17, 4, 1998, p. 338–359.
- [2] P. CASPI, A. GIRAULT, D. PILAUD, « Automatic Distribution of Reactive Systems for Asynchronous Networks of Processors », *IEEE Trans. on Software Engineering* 25, 3, mai/juin 1999, p. 416–427.
- [3] B. ESPIAU, F. CHAUMETTE, P. RIVES, « A New Approach to Visual Servoing in Robotics », *IEEE Trans. on Robotics and Automation* 8, 3, 1992.
- [4] B. ESPIAU, THE BIP-TEAM, « Bip : a joint project for the Development of an Anthropomorphic Biped Robot », in : *Int. Conf. on Advanced Robotics*, Monterey, CA, USA, juillet 1997.
- [5] A. GOSWAMI, B. THUILOT, B. ESPIAU, « Compass-Like Biped Robot Part I : Stability and Bifurcation of Passive Gaits », *Rapport de Recherche n°2996*, INRIA, octobre 1996, <http://www.inria.fr/rrrt/rr-2996.html>.
- [6] C. SAMSON, B. ESPIAU, M. LE BORGNE, *Robot Control : The Task Function Approach*, Oxford Science Publications, 1991.
- [7] D. SIMON, B. ESPIAU, E. CASTILLO, K. KAPellos, « Computer-Aided Design of a Generic Robot Controller Handling Reactivity and Real-Time Control Issues », *IEEE Transactions on Control Systems Technology* 1, 4, décembre 1993, <http://www.inrialpes.fr/iramr/pub/Orccad>.

Thèses et habilitations à diriger des recherches

- [8] F. F. JIMÉNEZ FRAUSTRO, *Conception sûre des automatismes industriels : modélisation synchrone de langages d'automates programmables de la norme CEI-61131-3*, Thèse de Doctorat en Informatique, IFSIC, Université de Rennes 1, 28 mars 2001.

Articles et chapitres de livre

- [9] N. ANDREFF, R. HORAUD, B. ESPIAU, « Robot hand-eye calibration using structure-from-motion », *International Journal of Robotics Research* 20, 3, 2001, p. 228–248.
- [10] F. BACCELLI, B. GAUJAL, D. SIMON, « Analysis of Preemptive Periodic Real Time Systems using the (max,plus) Algebra With applications in robotics », *IEEE Trans. on Control Systems Technology*, 2002, à paraître.

- [11] J.-R. BEAUVAIS, E. RUTTEN, T. GAUTIER, R. HOUEBINE, P. LE GUERNIC, Y. TANG, « Modelling Statecharts and Activitycharts as Signal equations », *ACM Transactions on Software Engineering and Methodology* 10, 4, octobre 2001, p. 397–451.
- [12] B. BROGLIATO, A. TEN DAM, L. PAOLI, F. GÉNOT, M. ABADIE, « Numerical simulation of finite-dimensional multibody nonsmooth mechanical systems », *ASME Applied Mechanics Reviews*, 2001, à paraître.
- [13] B. BROGLIATO, « On the control of non-smooth complementarity dynamical systems », *Phil. Trans. R. Soc. Lond. A*, vol.359, 2001.
- [14] A. GIRAULT, « Elimination of Redundant Messages with a Two-Pass Static Analysis Algorithm », *Parallel Computing*, 2002, à paraître.
- [15] H. MARCHAND, E. RUTTEN, M. LE BORGNE, M. SAMAAAN, « Formal Verification of Programs specified with SIGNAL : Application to a Power Transformer Station Controller », *Science of Computer Programming* 41, 1, 2001, p. 85–104.

Communications à des congrès, colloques, etc.

- [16] C. ANDRÉ, F. BOULANGER, A. GIRAULT, « Software Implementation of Synchronous Programs », in : *International Conference on Application of Concurrency to System Design, ICACSD'01*, IEEE, p. 133–142, Newcastle, UK, June 2001.
- [17] E. ASARIN, S. BANSAL, B. ESPIAU, T. DANG, O. MALER, « On Hybrid control of Underactuated Systems », in : *Fourth International Workshop on Hybrid Systems, Rome*, mars 2001.
- [18] J. AVILA-VILCHIS, B. BROGLIATO, R. LOZANO, « Nonlinear control of helicopters », in : *European Control Conference ECC01, Porto*, septembre 2001.
- [19] C. AZEVEDO, « On the Interaction between Human and Biped Robot Walking », in : *Proceedings of the International Symposium on Mobile, Climbing and Walking Robots, CLAWAR'01, Karlsruhe, Germany*, 2001.
- [20] B. BROGLIATO, M. MABROUK, A. ZAVALA-RIO, « A controllability criterion for juggling mechanical systems », in : *3rd Contact Mechanics International Symposium*, June 17-21 2001.
- [21] B. BROGLIATO, « Absolute stability and the Lagrange-Dirichlet theorem with monotone multi-valued mappings », in : *IEEE Conference on Decision and Control*, 2001.
- [22] N. CISLO, B. ESPIAU, « Path-Planning for Biped Locomotion in a 3D Partially Structured Environment », in : *32nd International Symposium on Robotics, ISR'2001, Seoul, Corée*, avril 2001.
- [23] C. DIMA, A. GIRAULT, C. LAVARENNE, Y. SOREL, « Off-Line Real-Time Fault-Tolerant Scheduling », in : *9th Euromicro Workshop on Parallel and Distributed Processing, PDP'01*, p. 410–417, Mantova, Italy, February 2001.
- [24] A. GIRAULT, C. LAVARENNE, M. SIGHIREANU, Y. SOREL, « Fault-Tolerant Static Scheduling for Real-Time Distributed Embedded Systems », in : *21st International Conference on Distributed Computing Systems, ICDCS'01*, IEEE, p. 695–698, Phoenix, USA, April 2001. Extended abstract.
- [25] A. GIRAULT, C. LAVARENNE, M. SIGHIREANU, Y. SOREL, « Generation of Fault-Tolerant Static Scheduling for Real-Time Distributed Embedded Systems with Multi-Point Links », in : *IEEE Workshop on Fault-Tolerant Parallel and Distributed Systems, FTPDS'01*, IEEE, San Francisco, USA, April 2001.
- [26] A. GIRAULT, « Elimination of Redundant Messages with a Two-Pass Static Analysis Algorithm », in : *9th Euromicro Workshop on Parallel and Distributed Processing, PDP'01*, p. 178–185, Mantova, Italy, February 2001.

- [27] F. JIMÉNEZ-FRAUSTRO, E. RUTTEN, « A synchronous model of IEC 61131 PLC languages in Signal », in : *Proceedings of the 13th Euromicro Conference on Real-Time Systems, ECRTS'01*, June 13th-15th, 2001, Delft, The Netherlands, p. 135–142, 2001.
- [28] F. MARANINCHI, Y. RÉMOND, E. RUTTEN, « Effective programming language support for discrete-continuous mode-switching control systems », in : *Proceedings of the 40th IEEE Conference on Decision and Control, CDC'01*, december 4-7, 2001, Orlando, Florida, p. à paraître, 2001.
- [29] E. RUTTEN, « A framework for using discrete control synthesis in safe robotic programming and teleoperation », in : *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'2001*, may 21–26, 2001, Seoul, Korea, p. 4104–4109, 2001.
- [30] D. SIMON, A. GIRAULT, « Synchronous Programming of Automatic Control Applications Using Orccad and Esterel », in : *IEEE Conference on Decision and Control, CDC'01*, Orlando, USA, December 2001.

Rapports de recherche et publications internes

- [31] C. AZEVEDO, R. PISSARD-GIBOLLET, « Contrôleur du robot Bip2000 », *Rapport Technique n°0249*, INRIA, 2001, <http://www.inria.fr/rrrt/rt-0249.html>.
- [32] A. GIRAULT, « Design of an Hybrid Controller for Autonomous Vehicles Driving on Automated Highways », *Research Report n°4286*, Inria, November 2001, <http://www.inria.fr/rrrt/rr-4286.html>.
- [33] Y. HURMUZLU, F. GÈNOT, B. BROGLIATO, « Modelling , stability and control of bipedal robots— A general framework », *Research Report n°4290*, Inria, October 2001, <http://www.inria.fr/rrrt/rr-4290.html>.
- [34] R. LEINE, B. BROGLIATO, « Bifurcations and periodic motion induced by the Painlevé paradox », *Research Report n°4322*, Inria, December 2001, <http://www.inria.fr/rrrt/rr-4322.html>.
- [35] F. LYDOIRE, « Simulateur du robot Bip2000 », *Rapport technique*, INRIA, 2001, à paraître.

Divers

- [36] F. BENATTAR, *Programmation et vérification de tâches robotiques multicadences*, Mémoire, DEA IVR, INPG/UJF, 2001.
- [37] J.-M. BOURGEOT, *Planification et génération de trajectoires d'un robot bipède en environnement non structuré*, Mémoire, DEA Réalité Virtuelle et Maîtrise des Systèmes Complexes, RVMSC, Versailles Saint-Quentin-en-Yvelines, 2001.
- [38] J. LAINE, *Génération automatique de code temps-réel pour le contrôle et la commande de systèmes robotiques complexes*, Mémoire, IUP Génie Informatique, La Rochelle, 2001.
- [39] F. LYDOIRE, *Simulation d'un robot marcheur et génération de démarches paramétrées*, Mémoire, DEA IVR, INPG/UJF, 2001.
- [40] C. MÉNIER, *Répartition de circuits séquentiels*, Mémoire, stage d'été, ENS Lyon, 2001.
- [41] P. P. PARIDA, *A case study in discrete controller synthesis : from specification to graphical simulation*, Mémoire, Training Report, Indian Institute of Technology Kharagpur, 2001.
- [42] J.-P. ROUSSEL, *Réalisation d'une loi de commande pour le contrôle longitudinal d'un véhicule automatique*, Mémoire, stage d'Ingénieur, Ecole Supérieure des Sciences Appliquées pour l'Ingénieur de Mulhouse, ESSAIM, 2001.