

## *Projet BIP*

*Contrôle/commande de robots marcheurs et applications*

*Rhône-Alpes*

THÈME 4A

*R* *apport*  
*d'Activité*

2000



## Table des matières

<b>1</b>	<b>Composition de l'équipe</b>	<b>3</b>
<b>2</b>	<b>Présentation et objectifs généraux</b>	<b>4</b>
<b>3</b>	<b>Fondements scientifiques</b>	<b>5</b>
3.1	Les robots bipèdes . . . . .	5
3.1.1	Motivations . . . . .	5
3.1.2	Problématique scientifique . . . . .	5
3.2	Contrôle/commande de systèmes robotiques complexes . . . . .	7
<b>4</b>	<b>Domaines d'applications</b>	<b>9</b>
<b>5</b>	<b>Logiciels</b>	<b>9</b>
<b>6</b>	<b>Résultats nouveaux</b>	<b>11</b>
6.1	Modélisation et commande . . . . .	11
6.1.1	Poursuite des études sur le double pendule sous-actionné . . . . .	11
6.1.2	Modélisation et commande d'un robot marcheur anthropomorphe . . . . .	12
6.1.3	Marche statique dans le plan sagittal . . . . .	12
6.1.4	Mouvements posturaux statiquement stables . . . . .	14
6.1.5	Planification de trajectoires pour la locomotion bipède en environnement 3D partiellement structuré . . . . .	17
6.1.6	Activités autour des prototypes de bipèdes . . . . .	18
6.2	Contrôle/commande de systèmes robotiques complexes . . . . .	20
6.2.1	Simulateur temps-réel de robots . . . . .	20
6.2.2	Reconstruction 3D de l'environnement par vision stéréo . . . . .	20
6.3	Aide à la conception de systèmes de contrôle/commande . . . . .	22
6.3.1	Spécification de missions robotiques . . . . .	22
6.3.2	Tolérance aux pannes . . . . .	23
6.3.3	Vérification de propriétés de programmes ORCCAD . . . . .	23
6.3.4	Génération automatisée de contrôleurs . . . . .	24
6.4	Modèles du temps-réel et langages réactifs . . . . .	25
6.4.1	Modèles du temps-réel . . . . .	25
6.4.2	Langages de programmation d'automates industriels . . . . .	26
<b>7</b>	<b>Contrats industriels (nationaux, européens et internationaux)</b>	<b>26</b>
7.1	VIGOR . . . . .	26
7.2	TELEDIMOS . . . . .	27
7.3	Transfert industriel . . . . .	27

<b>8</b>	<b>Actions régionales, nationales et internationales</b>	<b>27</b>
8.1	Actions nationales . . . . .	27
8.1.1	GdR-PRC "Commande de robots à pattes" . . . . .	27
8.1.2	PSLC : Pôle Systèmes et Logiciels Critiques . . . . .	27
8.1.3	Groupe COSED . . . . .	27
8.1.4	Collaborations internes à l'Inria . . . . .	28
8.1.5	Collaborations avec d'autres laboratoires . . . . .	28
8.2	Actions européennes . . . . .	29
<b>9</b>	<b>Diffusion de résultats</b>	<b>29</b>
9.1	Animation de la Communauté scientifique . . . . .	29
9.2	Exposition Universelle . . . . .	30
9.3	Enseignement . . . . .	30
<b>10</b>	<b>Bibliographie</b>	<b>31</b>

# 1 Composition de l'équipe

## Responsable scientifique

Bernard Espiau [DR Inria]

## Assistante de projet

Véronique Roux [AJT, à temps partiel dans le projet]

## Personnel Inria

Alain Girault [CR]

Eric Rutten [CR]

Philippe Sardain [en délégation de l'IUT de Poitiers]

Daniel Simon [CR]

## Ingénieur expert

Mathieu Personnaz [Projet TELEDIMOS, depuis le 1er mars 2000]

## Chercheurs doctorants

Christine Azevedo [Boursier MENRT]

Nicolas Andreff [ATER à l'ENSIMAG jusqu'au 1er septembre 2000]

Antonio Medina-Rodriguez [Boursier CONACYT (Mexique), depuis le 16 octobre 2000]

Pierre-Brice Wieber [Boursier DGA, jusqu'au 1er septembre 2000]

## Chercheurs post-doctorants

Nathalie Cislo [Post Doctorant, en coopération avec les Moyens Robotiques, jusqu'au 31 août 2000]

Cătălin Dima [Post Doctorant TOLERE, jusqu'au 31 octobre 2000]

## Stagiaires

Fanny Benattar [juin-juillet 2000, Maîtrise]

Claudio Pinello [juin-juillet 2000, PhD, University of Berkeley]

Mathieu Potier [mars-septembre 2000, DEA]

David Thomas [mars-juillet 2000, DEA]

## Collaborateurs extérieurs

Hassane Alla [LAG]

Carlos Canudas de Wit [LAG]

Nathalie Ciso [UFRAPS, depuis le 1er septembre 2000]

## 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.

Le projet BIP est actif dans les projets Européens VIGOR, TELEDIMOS, 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<sup>[VDO92,Win90]</sup>, 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 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.

**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<sup>[GTE98]</sup>.

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,

- 
- [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.  
 [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.



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 Contrôle/commande de systèmes robotiques complexes

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

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.

L'architecture 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.

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 ;
- 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 ;

---

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

- 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 <sup>[RW87]</sup>, 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, 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.

Enfin, 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 suggère le transfert industriel lié à ORCCAD.

## 5 Logiciels

**Participants :** Daniel Simon, Alain Girault, Bernard Espiau, Roger Pissard-Gibollet [Moyens robotiques], Soraya Arias [Moyens robotiques, correspondant].

**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.

---

[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.

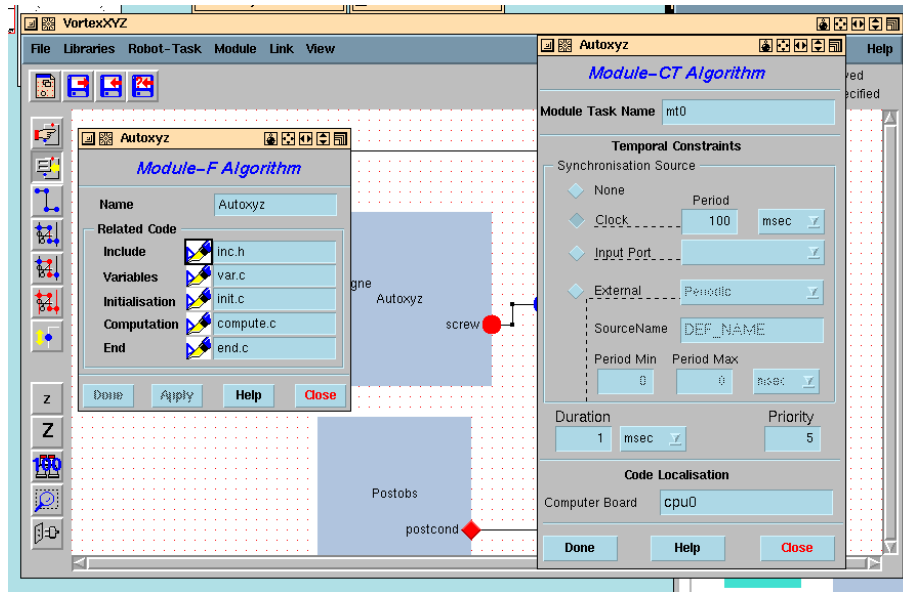


FIG. 1 – Éditeur de Tâches-Robot

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 (figure 1)
- 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 ou SOLARIS

ORCCAD V3.0 $\beta$  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/>

Les fonctionnalités améliorées dans cette nouvelle version sont : un générateur de code pour le simulateur SIMPARC (Jean-Jacques Borrelly, INRIA-SOPHIA-ANTIPOLIS), un exécutif pour Posix/SOLARIS 2.6 et des modifications mineures de l'IHM.

ORCCAD présente désormais une partie graphique développée sous Ilog Views 3.11. Il est disponible à la fois sous SOLARIS 2.5 (et sup.) 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. De plus, ORCCAD permet de générer maintenant des programmes exécutables :

- pour le noyau temps réel VXWORKS,
- pour SOLARIS 2.5 (et sup),
- pour LINUX (RedHat 6.1), sachant que la cadence de l'application est contrainte à une granularité de 10ms.

## 6 Résultats nouveaux

### 6.1 Modélisation et commande

**Participants** : N. Andreff, C. Azevedo, C. Canudas de Wit [LAG], W. Chatila [LAG], N. Cislo, B. Espiau, R. Pissard-Gibollet [M.R.], P. Sardain [LMS], P.-B. Wieber.

#### 6.1.1 Poursuite des études sur le double pendule sous-actionné

Un robot marcheur est par nature un système non-linéaire sous-actionné. On sait que la commande de tels systèmes nécessite des méthodes très particulières, comme celles mises en œuvre sur l'Acrobot ou le Pendubot. Usuellement, on cherche à stabiliser un point, plus rarement une trajectoire temporelle. Dans le cas qui nous concerne, la marche stationnaire étant un mouvement périodique, on va s'intéresser aux trajectoires dans le plan de phase qui sont des orbites fermées. Nos études passées avaient porté sur deux points (cf rapports d'activité des années précédentes) :

- l'analyse des orbites naturelles passives pour le double pendule libre sur un plan (compas);
- la réalisation d'une commande permettant la synthèse d'une marche balistique pour un double pendule fixé et actionné à la hanche, représentant une jambe.

Nous avons cette année proposé une approche différente de ce problème, selon la voie suivante : on définit un changement de coordonnées paramétré, de même dimension que celle de l'actionnement, représentant une contrainte à réaliser. On applique ensuite une commande permettant d'amener à zéro ces nouvelles coordonnées (de préférence en temps fini, mais éventuellement asymptotiquement par découplage et linéarisation par feedback partiels). Le comportement résultant après stabilisation est libre et représente une sorte de dynamique zéro qui dépend du choix des paramètres. Dans certains cas, et s'il n'y a pas d'amortissement dans le système, ce comportement est périodique. On peut donc spécifier le comportement désiré du

système par une telle orbite (caractérisée par un niveau de pseudo-énergie et par les paramètres associés). Le problème consiste alors à, partant d'un certain état, atteindre cette orbite-cible et à s'y maintenir. Les seules actions disponibles sont le changement des paramètres. Nous avons seulement partiellement résolu ce problème, en exhibant des conditions locales d'existence d'un centre et en proposant une méthode de "sauts" successifs de paramètres permettant d'atteindre la cible. La difficulté majeure provient du fait que le système projeté sur la contrainte n'est plus Lagrangien et que, donc, l'énergie n'est pas une fonction de Lyapunov naturelle. La caractérisation analytique de l'équation de l'orbite, et donc de son niveau de pseudo-énergie, apparaît impossible en général. Par contre, l'idée de sauts successifs d'une orbite à l'autre en tenant compte des contraintes sur l'état et les paramètres est séduisante et se prête bien à une analyse d'accessibilité en termes de systèmes hybrides. L'exemple du double pendule a ainsi été traité avec succès ([15]) dans le cadre d'une collaboration avec E. Asarin, T. Dang et O. Maler (VERIMAG).

### 6.1.2 Modélisation et commande d'un robot marcheur anthropomorphe

Cette année a vu la soutenance de la thèse de Pierre-Brice Wieber [8] sur la modélisation et commande d'un robot marcheur anthropomorphe.

Cette thèse propose de donner un nouvel éclairage sur la dynamique des robots marcheurs, développant tout particulièrement la distinction entre les changements de posture d'un robot, ses déplacements, et ses rotations. La nécessité de prendre appui sur le sol pour pouvoir contrôler ses déplacements est alors démontrée en s'appuyant sur une reformulation de la dynamique des systèmes mécaniques à l'aide du principe de Gauss. La modélisation qui est proposée s'appuie par ailleurs sur un modèle de contact rigide avec frottements.

Cette modélisation est ensuite validée lors de diverses expérimentations menées avec le robot BIP, ce qui a constitué une part importante de ce travail de thèse. Ce choix de modélisation permet également de mettre en évidence les conditions qui président à l'équilibre ou au déséquilibre d'un robot marcheur, amenant à analyser selon une grille de lecture unique les différents moyens qui peuvent être mis en oeuvre pour contrôler cet équilibre. Cette analyse est alors mise à profit pour proposer une loi de commande qui devrait permettre d'assurer l'équilibre du robot même en présence de fortes perturbations.

### 6.1.3 Marche statique dans le plan sagittal

Un premier prototype mécanique du robot a été réalisé par le Laboratoire de Mécanique des Solides de Poitiers (LMS). Ce prototype a été instrumenté par le service des Moyens Robotiques de l'INRIA-RHÔNE-ALPES. Cette version était composée des deux jambes du robot (8 degrés de liberté (ddl)), et était dotée d'une armoire de commande reliée par un cordon ombilical au robot (Fig. 3). Elle est associée à un chariot par l'intermédiaire d'une barre de torsion (Fig. 2), qui permet de maintenir le robot dans le plan sagittal. On a ainsi un système à six degrés de liberté.

Le principe du contrôleur que nous avons utilisé est la poursuite de trajectoires articulaires générées hors-ligne. Ces trajectoires ont été calculées de manière à garantir l'équilibre du robot à tout moment. Des postures statiquement stables ont été générées en utilisant une fonction

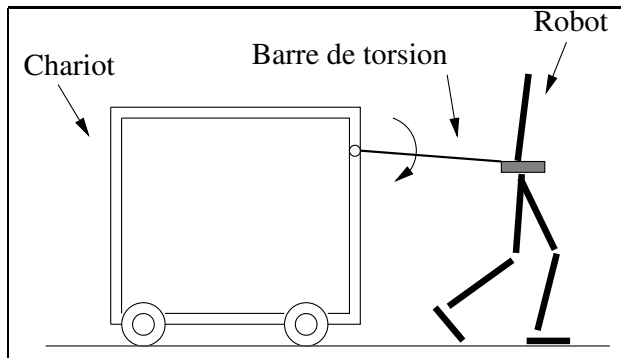


FIG. 2 – Version simplifiée du robot

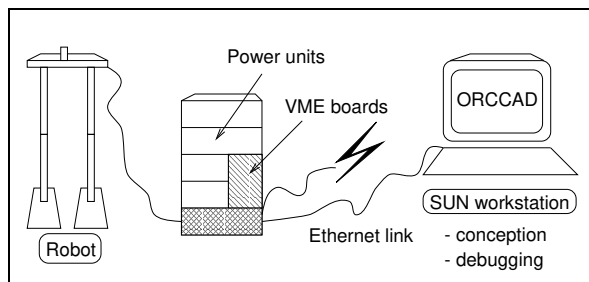


FIG. 3 – Architecture de la première version du robot (8 ddl)

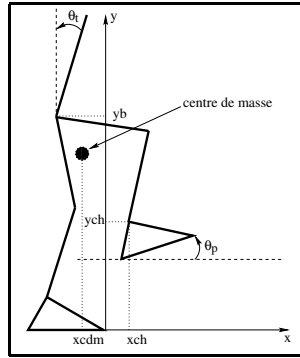


FIG. 4 – Définition de la fonction de sortie

de sortie (Fig. 4) spécifiant l'allure générale désirée pour le robot à des instants donnés de la démarche [27]. Cette fonction de sortie vérifie des contraintes garantissant que l'équilibre statique est conservé et que le pied de vol ne heurte pas intempestivement le sol. En interpolant entre les postures ainsi obtenues on obtient un mouvement dans l'espace pour le robot. Ce mouvement est en suite traduit en termes de trajectoires des articulations du robot.

Nous avons développé un contrôleur permettant de poursuivre les trajectoires articulaires générées. Les différents programmes ont été implémentés sous l'environnement ORCAD [1].

L'interfaçage logiciel entre les entrées/sorties robot (actionneurs, capteurs) et le modèle mécanique théorique a été réalisé en deux couches logicielles (drivers et modèles de conversion). L'utilisation de capteurs de force permet de détecter les changements de support. La commande utilisée fait partie des commandes dites passives, elle comprend un correcteur de type Proportionnel Dérivé associé à une compensation du vecteur de gravité et une compensation des frottements secs (estimés expérimentalement). Nous avons ainsi développé une procédure robot et les tâches robots associées (Fig. 5) pour la définition de la mission de marche du robot.

Nous sommes parvenus à réaliser expérimentalement différentes démarches [16]. Ces résultats ont été présentés à l'Exposition Universelle de Hanovre (2000). Nous avons, à cette occasion, mené une étude sur la consommation d'énergie comparée entre différentes démarches de marche statique [27].

#### 6.1.4 Mouvements posturaux statiquement stables

Le second prototype mécanique dans sa version complète (15 ddl) a été réalisé par le LMS et instrumenté par le service des Moyens Robotiques de l'INRIA-RHÔNE-ALPES. Les algorithmes de la version 8 ddl ont été transposés à la version 15 ddl. Ce travail a fait l'objet d'un rapport technique [28]. L'armoire de commande est intégrée à la structure mécanique et tient lieu de tronc (Fig. 7).

Le principe de génération de trajectoires pour les articulations est basé sur une commande par fonction de tâche s'adaptant aux déséquilibres du robot.

Nous sommes parvenus à réaliser expérimentalement les mouvements posturaux en appui sur un pied [16]. Ces résultats ont été présentés à l'Exposition Universelle de Hanovre.



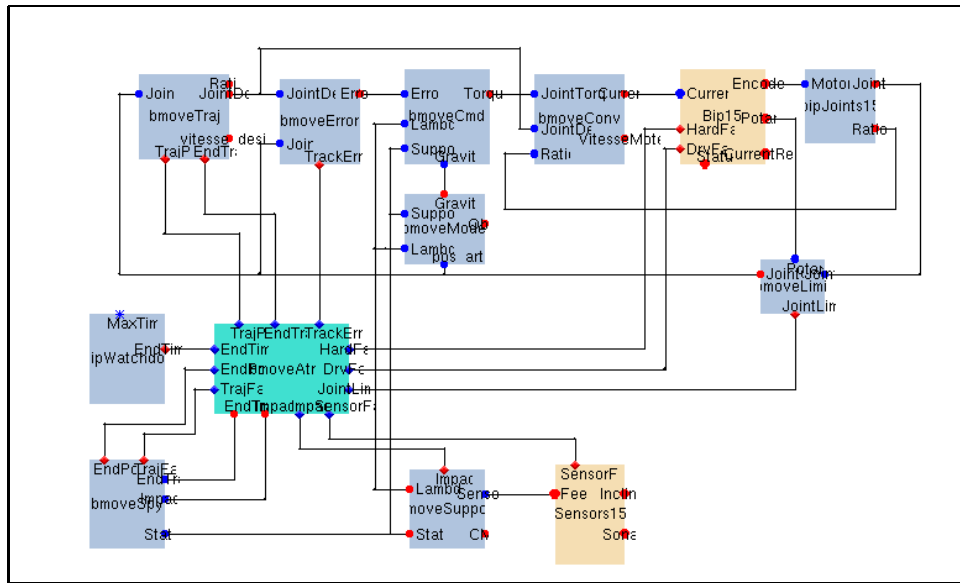


FIG. 5 – Tâche robot pour la poursuite de trajectoires articulaires

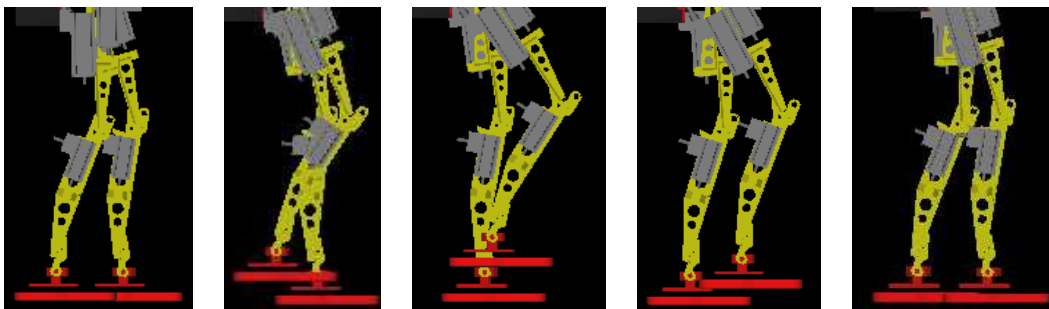


FIG. 6 – Pas d'une démarche

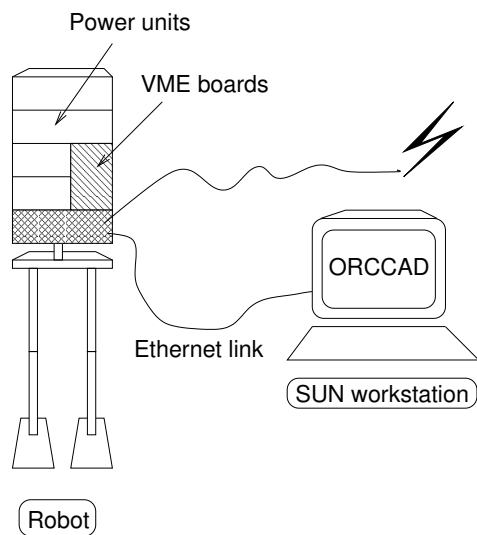


FIG. 7 – Architecture du robot Bip2000

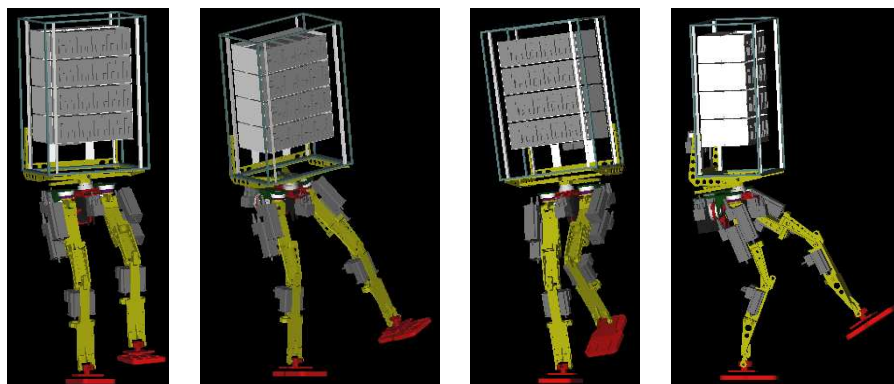


FIG. 8 – Mouvements posturaux

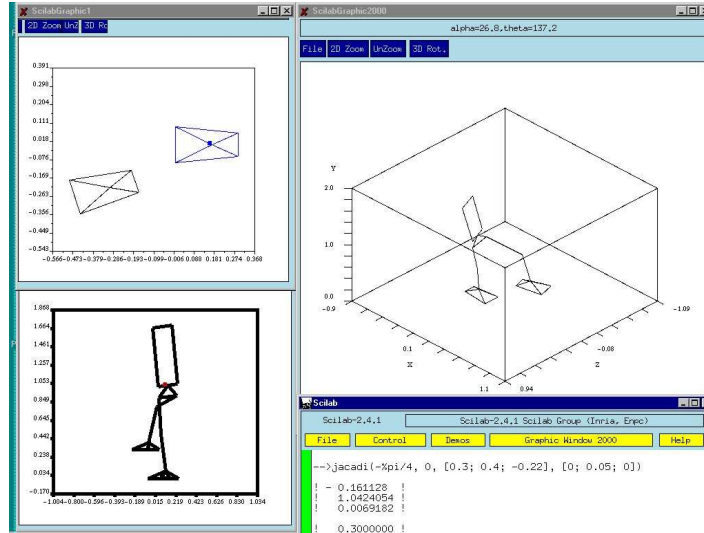


FIG. 9 – Jacadi, sous Scilab, pour la génération de postures statiquement stables

### 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.

La génération d'un chemin pour un robot bipède sur un terrain partiellement structuré peut être considérée comme la recherche d'une séquence de postures réalisables, telles que le robot conserve sa stabilité et telles que le mouvement préserve sa continuité. Un algorithme, basé sur A\*, a été utilisé pour déterminer une séquence des pas réalisables entre des points donnés d'un environnement 3D. Chaque pas est une trajectoire, une séquence de postures statiquement stables, qui respectent les contraintes liées au robot et à l'environnement.

Un pas est réalisable quand le robot est capable de mouvoir le pied en vol sur un point d'appui possible pour son pied, et de transférer le centre de pression du précédent point d'appui sur le suivant. Cela entend qu'il existe un mouvement du robot entre un point d'appui et un autre qui satisfasse les contraintes suivantes : la stabilité statique est préservée à chaque instant ; le mouvement respecte les contraintes liées aux amplitudes des articulations et des couples moteur, et à l'évitement d'obstacles.

Les mouvements locaux sont calculés en utilisant une méthode d'optimisation non linéaire qui prend en compte ces contraintes, en général des inégalités, mais aussi des contraintes d'égalité (la position désirée du pied et l'orientation de certains segments). Le logiciel baptisé « Jacadi » (voir Fig. 9), développé pour la génération de postures statiquement stables, sous Scilab, a été étendu dans ce contexte.

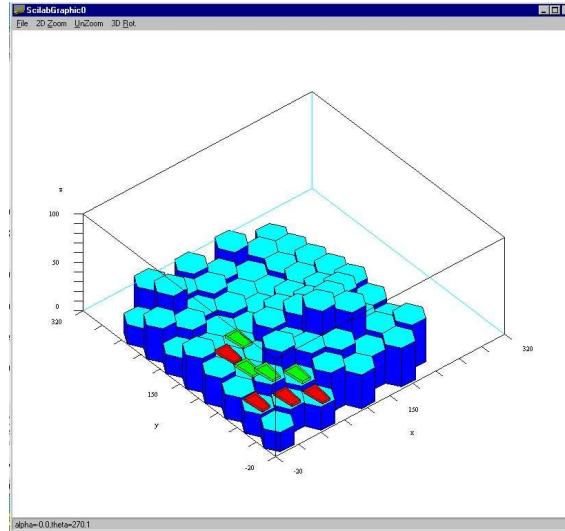


FIG. 10 – Détermination d'un chemin pour BIP2000 dans un environnement 3D partiellement structuré

Les résultats concernent un environnement modélisé à l'aide de colonnes hexagonales de différentes tailles. Le terrain est alors divisé selon une grille hexagonale (le principal avantage de cette grille inspirée des jeux de guerre est de permettre des mouvements similaires dans toutes les directions, ce qui n'est pas le cas avec les grilles carrées) (voir Fig. 10).

Le choix de fonctions de coût adéquates, pour l'optimisation sous contraintes des postures, a permis de maximiser l'espace accessible du pied en vol. Différentes heuristiques permettent de déterminer de bons chemins.

### 6.1.6 Activités autour des prototypes de bipèdes

Deux prototypes du robot bipède (Fig. 11) ont été réalisés, avec le laboratoire de mécanique des solides de Poitiers, et donnent lieu à un travail de mise en place, lié aux expérimentations des résultats du projet et à l'Exposition Universelle de Hanovre en 2000 (voir Section 9.2).

La phase de développement technologique s'est terminée vers la fin du premier semestre. Les recherches entreprises ensuite visent à exploiter le potentiel expérimental de notre démonstrateur :

- Établissement d'un processus d'initialisation du robot, permettant de définir les positions articulaires de référence par identification,
- Étude de la marche avec des pieds quasi-rigides, en collaboration avec une équipe de bio-mécaniciens,
- Développement d'une commande "en couple" au niveau des chevilles, tirant parti de l'instrumentation des pieds en capteurs d'efforts.

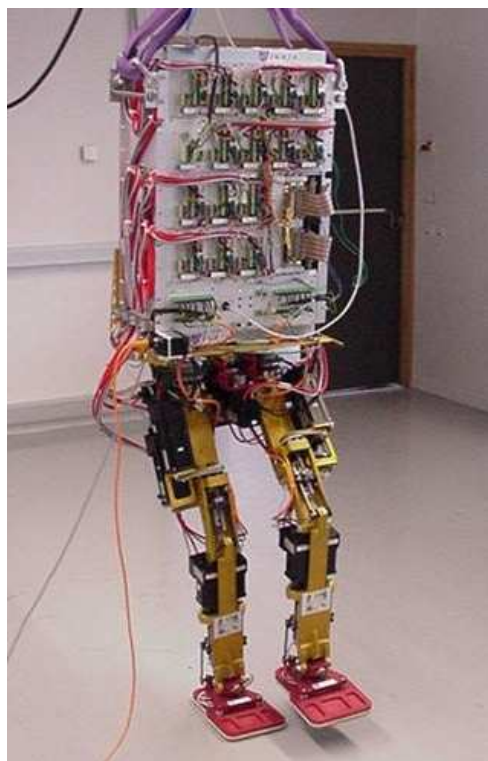


FIG. 11 – *Le robot bipède bip2000*

- Recherche d'allures de marche 3D, combinant marche sagittale optimale d'une part, et mouvement dans le plan frontal, défini pour l'instant sur des bases simples, d'autre part.
- Préparation de la mise en œuvre de la marche dynamique; sur ce sujet, une collaboration avec le laboratoire Sport et Performance Motrice de l'UFRAPS, à l'Université Joseph Fourier (Grenoble 1) est en initialisation. Il s'agit d'étudier le comportement humain en cas de perturbation du contrôle postural, afin de dégager des invariants transposables au cas d'un robot bipède.

Ces actions pourront démarrer dès le retour de Hanovre du robot marcheur.

## 6.2 Contrôle/commande de systèmes robotiques complexes

**Participants :** M. Personnaz, D. Simon.

### 6.2.1 Simulateur temps-réel de robots

L'une des tâches à réaliser dans le cadre du projet TELEDIMOS (voir Section 7.2) est la réalisation d'un simulateur temps-réel du système de téléopération, en particulier prenant en compte la dynamique du robot. Un tel logiciel est indispensable à l'entraînement des opérateurs et à la validation *a priori* de la tâche à exécuter.

Le logiciel a été réalisé à l'aide d'ORCCAD et comprend les modules suivants [32] :

- un générateur de trajectoires dans l'espace opérationnel utilisant le calcul de la pseudo-inverse de la matrice Jacobienne du bras,
- une commande simple (PID avec compensation de gravité),
- un intégrateur numérique à pas variable appelé par la fonction pilote du module RESOURCE\_PHYSIQUE à la cadence de la commande,
- une interface graphique de paramétrage et de pilotage de la simulation.

Les modèles géométriques, cinématiques et dynamiques du robot ont été automatiquement générés par ROBOTDYN, précédemment développé dans le projet. Le logiciel utilise le générateur de code original d'ORCCAD et fonctionne sous SOLARIS et LINUX (avec une résolution d'horloge limitée à 10 msec). L'application particulière de l'excavatrice TELEDIMOS fonctionne très largement en temps-réel sur un PC obsolète. Cette structure logicielle ouverte peut par ailleurs être facilement adaptée à d'autres types de robots pour effectuer des simulations de type "hardware-in-the-loop" bien qu'il ne soit pas formellement possible de prouver que la simulation sera bien temps-réel [26].

### 6.2.2 Reconstruction 3D de l'environnement par vision stéréo

La participation de l'INRIA au projet TELEDIMOS comprend notamment l'intégration d'un système de vision élaboré à partir des compétences algorithmiques et de prototypes logiciels

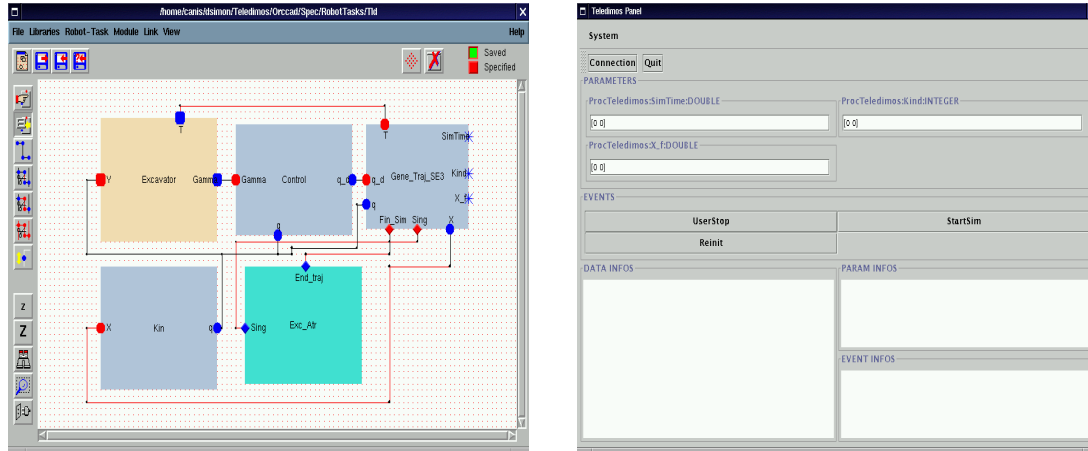


FIG. 12 – *Spécification et panneau de contrôle du simulateur*

issus du projet MOVI. Le système permet l'acquisition des coordonnées 3D de points de l'espace observés par une paire de caméras stéréoscopiques. Le logiciel TELE2 constitue la partie logicielle du système de vision. Il intègre les fonctionnalités suivantes:

- Acquisition en temps-réel d'images captées par une caméra numérique,
- Étalonnage individuel des caméras à l'aide d'une mire,
- Étalonnage de la paire stéréoscopique à l'aide d'une mire (géométrie épi-polaire),
- Sélection de points d'intérêt par un filtre de Harris,
- Construction interactive d'un ensemble de points 3D observés par la paire de caméras stéréoscopiques,
- Construction automatique et périodique d'un ensemble de points 3D,
- Accès aux résultats via une liaison TCP/IP,
- Mise en œuvre rapide et aisée des différentes fonctionnalités grâce à une interface graphique ergonomique et conviviale (programmée en Java).

Ce logiciel fonctionne sous LINUX et SOLARIS (voir Fig. 13). Les algorithmes implantés ont été testés et leurs résultats analysés. Quelques centaines de points 3D sont appariés en quelques secondes sur un PC classique, avec un taux d'erreur en appariement automatique, dépendant du type de scène observée, pouvant atteindre 30%. Le développement se poursuit par l'analyse de la précision de la reconstruction et la diminution du taux d'appariements erronés. Le logiciel est accompagné d'un document de spécifications fonctionnelles et d'un guide utilisateur [31], [30].

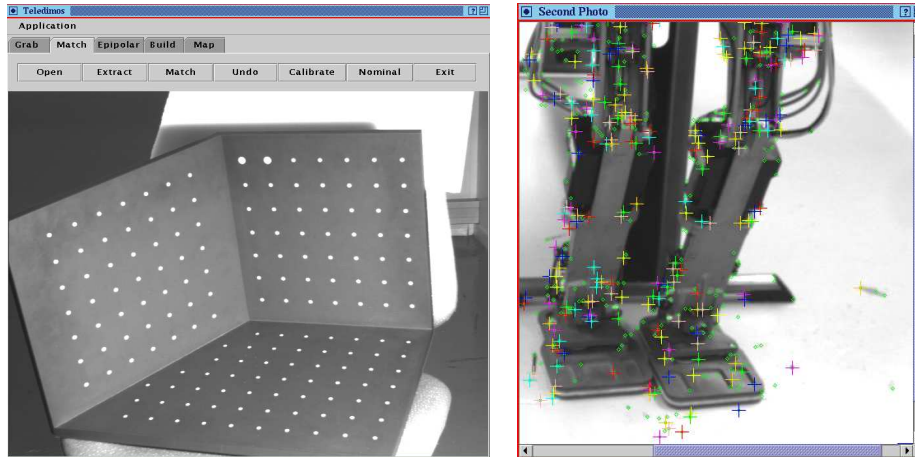


FIG. 13 – Calibration d'une camera et Reconstruction automatique de points 3D

### 6.3 Aide à la conception de systèmes de contrôle/commande

**Participants :** H. Alla [LAG], S. Arias [M.R.], F. Benattar, C. Dima, A. Girault, T. Grandpierre [SOSSO, INRIA-ROCQUENCOURT], H. Marchand [EP-ATR, INRIA-RENNES], R. Mateescu [VASY], A. Medina, C. Pinello, R. Pissard-Gibollet [M.R.], E. Rutten, D. Simon, Y. Sorel [SOSSO, INRIA-ROCQUENCOURT].

#### 6.3.1 Spécification de missions robotiques

ORCCAD est principalement destiné aux applications temps-réel critiques en robotique, dans lesquelles les aspects relevant de l'automatique (asservissements) sont amenés à interagir étroitement avec ceux manipulant des événements discrets (spécifications de missions).

Le comportement réactif des actions de base est spécifié grâce à une interface graphique générant automatiquement le code ESTEREL correspondant. Cette spécification utilise des classes de signaux pré-définis : pré-conditions, post-conditions et exceptions. Deux nouvelles fonctionnalités ont été définies :

- La mise en œuvre d'applications utilisant du parallélisme d'exécution (robots coopérants avec recouvrement d'actions par exemple) a fait apparaître le besoin d'introduire un nouveau type d'événement (signaux de synchronisation) permettant une synchronisation plus fine entre actions et une spécification plus naturelle d'applications complexes. Ces signaux, appelés **SynchroSig**, permettent le démarrage d'une action sur un sous-système, sur un point intermédiaire de l'exécution d'une action d'un autre sous-système, sans que cette dernière soit arrêtée.
- Une structure souvent utilisée dans l'élaboration de procédures de contrôle de robots est la structure alternative : il s'agit, pour commander un robot en vue d'un objectif particulier, de choisir en fonction du contexte et de son évolution une action à exécuter parmi un sous-ensemble d'actions possibles. Ce mécanisme de garde doit être exempt d'interblocage et



doit préserver certaines propriétés fondamentales du système, par exemple qu'un robot est commandé par une et une seule action à un instant donné. Il n'est par conséquent pas évident de le coder à la main, et c'est pourquoi nous avons défini la construction **Altern**.

Ces deux nouvelles entités (SynchroSig et Altern) ont été intégrées dans l'interface de spécification, l'interface de vérification et le générateur de code d'ORCCAD.

### 6.3.2 Tolérance aux pannes

L'année 2000 marque la fin de l'ARC TOLERE, dont Alain Girault a été le coordinateur. Le but était d'étudier et de proposer des algorithmes pour produire automatiquement du code réparti tolérant aux pannes pour des systèmes réactifs embarqués.

Nous avons restreint notre problème de génération de code tolérant aux pannes à un problème d'ordonnancement réparti et tolérant aux pannes. Les algorithmes ont été considérés de la façon la plus générale comme des graphes acycliques : les nœuds sont des opérations de calcul, et les arcs sont des dépendances de données. De même les architectures ont été considérées comme des graphes non-orientés bipartites : les nœuds sont ou bien des processeurs ou bien des canaux de communication, et les arcs sont des liaisons entre un processeur et un canal. Le principe de base pour obtenir la tolérance aux pannes est la réplication des opérations de calcul ainsi que des communications des dépendances de données. La détection des pannes se fait à l'aide de chiens de garde.

Le travail a consisté dans l'énoncé des principes d'exécution des ordonnancements tolérants aux pannes et de détection des pannes à l'aide des chiens de garde, ainsi que dans le calcul théorique des valeurs des délais qui arment les chiens de garde. L'ensemble de ce travail est basé sur une généralisation du principe d'ordonnancement topologique qui permet d'ordonner certains graphes ayant des cycles, comme le sont les graphes associés aux ordonnancements avec répliques.

Enfin une partie de ces résultats a été acceptée pour publication dans une conférence [19].

### 6.3.3 Vérification de propriétés de programmes ORCCAD

Un programme ORCCAD s'articule autour d'un contrôleur à événements discrets écrit en ESTEREL [BG92], modélisé par un automate d'états fini, aux transitions étiquetées. Les propriétés que l'on désire vérifier font intervenir le séquençage, à la fois des transitions dans l'automate, et des actions à l'intérieur des étiquettes. Pour cela, on peut utiliser une logique temporelle spécifique, appelée «  $\mu$ -calcul régulier sans alternance » [25], extension du «  $\mu$ -calcul modal sans alternance » [EL86]. Nous faisons des expériences avec l'outil **evaluator** version 3.0<sup>1</sup>,

---

1. <http://www.inrialpes.fr/vasy/cadp/man/evaluator.html>

---

[BG92] G. BERRY, G. GONTHIER, « The ESTEREL Synchronous Programming Language: Design, Semantics, Implementation », *Science of Computer Programming* 19, 2, 1992, p. 87–152.

[EL86] E. A. EMERSON, C.-L. LEI, « Efficient Model Checking in Fragments of the Propositional Mu-Calculus », in : *Proceedings of the 1st LICS*, p. 267–278, 1986.

qui met en œuvre un algorithme d'évaluation efficace, et est distribué avec CADP (Cæsar Al-débaran Development Package<sup>2</sup>). Nous l'utilisons en connection avec ORCCAD, diversifiant ainsi les possibilités de vérification qui lui sont associées.

Nous continuons de mener une réflexion sur l'utilisabilité de ces techniques par des non-spécialistes. Les méthodes actuellement associées à ORCCAD, reposant sur XEVE, font intervenir la visualisation d'automates, dont la taille rend la lecture malaisée, et le diagnostic difficile et peu sûr. L'utilisation du *model checking* nécessite d'un autre côté de formuler les propriétés dans un formalisme particulier, intrinsèque au modèle sous-jacent (logique temporelle ou  $\mu$ -calcul), et sans rapport *a priori* avec les structures manipulées dans les applications. Nous nous proposons donc de définir un cadre d'expression de propriétés spécifique au domaine, en deux niveaux :

- des propriétés génériques décrites en langage naturel, pour lesquelles l'utilisateur ne doit préciser que les éventuels paramètres. Par exemple, il s'agit de la sûreté et de la vivacité du contrôleur, du fait que l'occurrence d'un événement donné provoque le démarrage d'une certaine loi de commande, Ces propriétés génériques sont ré-définies dans ORCCAD, sous la forme de macro-expressions.
- des propriétés particulières à chaque mission pour lesquelles nous étudions un langage de propriétés spécifique au domaine des systèmes de contrôle/commande, qui serait compilé vers un formalisme logique.

#### 6.3.4 Génération automatisée de contrôleurs

La problématique générale est celle des méthodes d'utilisation concrète de cette technique encore neuve qu'est la synthèse de contrôleurs à événements discrets, dans le cadre de la programmation de systèmes robotiques ou de contrôle-commande. La validation formelle de logiciels critiques consiste à spécifier des propriétés logiques d'un programme, et à les vérifier ensuite, par exemple à l'aide de techniques de model-checking, sur un automate qui représente l'ensemble des comportements possibles du programme. Cela produit un résultat de type oui/non, et parfois un diagnostic sous forme de contre-exemples. Le principe de la synthèse prend un point de vue en quelque sorte inverse. A partir de propriétés que doit avoir un contrôleur, la synthèse de contrôleurs discrets produit directement l'automate contraint, c'est-à-dire celui qui ne présente que les comportements qui satisfont les propriétés données. Les avantages attendus pour le concepteur de contrôleurs sont d'une part l'obtention de contrôleurs automatiquement corrects vis-à-vis des objectifs, et d'autre part une spécification qui soit abstraite des réglages fins de synchronisation.

Des travaux exploratoires ont commencé, qui font intervenir des expérimentations utilisant, en coopération avec H. Marchand (EP-ATR, INRIA-RENNES), l'environnement synchrone SIGNAL/SIGALI, qui a l'avantage d'offrir une approche complète : spécification, calcul, et exécution et simulation du contrôleur résultant. Les pistes explorées sont les suivantes :

- Les objectifs de synthèse peuvent correspondre aux propriétés attachées aux tâches de commande et aux applications telles que définies dans ORCCAD. Elles concernent des

---

<sup>2</sup>. CADP: <http://www.inrialpes.fr/vasy/cadp>

éléments liés à l'environnement, ou à la synchronisation des tâches. Par ce biais on réutilise une partie de l'expérience de l'activité de vérification d'applications ORCCAD, notamment dans le cadre de l'ARC TOLERE.

- Dans une autre approche, on s'intéresse à des objectifs plus « internes », où on étudie des structures de tâches comportant plusieurs modes d'exécution. On veut les utiliser pour définir des missions de contrôle-commande, et utiliser la synthèse de contrôleur pour gérer le passage entre ces modes, de façon à respecter les propriétés données en objectif. Les modes peuvent correspondre à des modes dégradés, ou à des modes de calcul présentant diverses formes d'approximations, enclenchés sur la satisfaction de leur conditions d'applicabilité, et ayant des coûts de calculs différents, ou une consommation d'énergie, ou d'autres critères à optimiser. Les automates de modes définis par F. Maranchi et Y. Rémond (VERIMAG) sont un outil de spécification adapté pour cette structuration. Une perspective est de considérer la définition d'un langage de spécification d'applications qui soit fondée sur les opérations de synthèse intervenant dans sa compilation.
- Un aspect important dans notre contexte est celui de l'utilisation, particulièrement de l'exécution, du contrôleur obtenu. On peut l'envisager :
  - automatique, de même que l'automate usuellement obtenu par compilation d'un programme réactif : il faut alors obtenir un contrôleur complètement déterminé, éventuellement en utilisant la synthèse optimale;
  - interactive, où un opérateur aurait toute liberté de décision à l'intérieur du sous-espace d'états de contrôle sûr vis-à-vis des objectifs; on a alors un équivalent de la téléopération dans le domaine du contrôle discret.

D'autres aspects de l'utilisation des contrôleurs concernent leur relation aux fonctionnalités décisionnelles auxquelles contribuent la planification de tâches (au sens de contrôle des tâches pour mener vers un but), ou l'apprentissage (au sens de restriction d'un ensemble de comportements possibles en fonction de la perception).

## 6.4 Modèles du temps-réel et langages réactifs

**Participants :** C. Dima, F. Jiménez-Fraustro [EP-ATR, INRIA-RENNES], P. Le Guernic [EP-ATR, INRIA-RENNES], E. Rutten.

### 6.4.1 Modèles du temps-réel

**Étude des automates temps-réel.** Nous avons étudié du point de vue des propriétés de déterminisation et complémentation une classe des automates avec ajout de temps-réel. La déterminisation a nécessité l'étude des propriétés d'une algèbre de Kleene des ensembles des réels positifs. Dans cette algèbre, la sous-algèbre générée à partir des intervalles à l'aide de l'union, la complémentation, la sommation et l'étoile de Kleene possède une propriété de forme normale qui permet d'éliminer les compléments [21].

**Modélisation algébrique des cellules de mémoire dans des circuits flot de données synchrones.** Nous avons étudié une sous-classe des circuits flots de données synchrones ayant des composants de retardement – i.e. des cellules de mémoire qui mémorisent une valeur à chaque cycle et la produisent au cycle suivant. Nous avons modélisé ces circuits comme des morphismes dans des théories algébriques (à la Lawvere) et les cellules de mémoire par une opération de point fixe (itération). On a étudié les propriétés équationnelles de l'itération et nous avons prouvé que le problème d'égalité des comportements des circuits flots de données synchrones dans des théories algébriques librement engendrées est décidable [20].

#### 6.4.2 Langages de programmation d'automates industriels

La programmation des systèmes de contrôle/commande dans les automatismes industriels relève des possibilités d'élargissement des domaines d'applications d'ORCAD. La problématique y fait intervenir la spécification de systèmes hybrides (commande continue et contrôle discret), la gestion de la complexité des contrôleurs, la sécurité critique nécessitant des méthodes et outils de validation et vérification, et les plateformes de mise en œuvre diverses et éventuellement distribuées. Dans ce domaine, la programmation se fait souvent sur la base d'automates programmables industriels (API). Leur conception repose sur des standards, tels que la norme de l'International Electrotechnical Commission IEC 61131, et en particulier sa partie IEC 61131-3 concernant les langages de programmation [CEI93].

Nous menons avec F. Jiménez-Fraustro et P. Le Guernic (projet EP-ATR, INRIA-RENNES) une activité visant l'intégration de ce standard de spécification et de l'environnement de programmation synchrone SIGNAL : il s'agit d'établir un modèle de ces langages, visant à offrir à de telles spécifications l'accès aux outils d'analyse et de mise en œuvre des techniques synchrones. Nous avons construit un modèle en SIGNAL des langages ST (texte structuré) [12] et FBD (blocs diagrammes fonctionnels). Enfin, les unités d'organisation de programmes (UOP), niveau structurel encapsulant tout programme écrit dans un des langages de la norme, ont aussi été modélisés. Par ailleurs, une expérimentation de la connection depuis ces langages vers les fonctionnalités de l'environnement SIGNAL a été menée, concernant une liaison avec SIMULINK [23]. La spécification d'un contrôleur pour un système de pompage a été fait en une UOP programmée en ST, et le modèle SIGNAL connecté avec un modèle de la dynamique de ce système.

## 7 Contrats industriels (nationaux, européens et internationaux)

### 7.1 VIGOR

**Participants :** B. Espiau, N. Andreff.

BIP participe avec MOVI au projet LTR VIGOR (*Visually Guided Robots Using Uncalibrated Cameras*), dont les partenaires sont : l'INRIA, l'université Hébraïque de Jérusalem, l'université

---

[CEI93] CEI, «Norme Internationale – Automates programmables : Langages de programmation», *rapport de recherche n° IEC 1131 partie 3*, CEI/IEC (Commission Électrotechnique Internationale/International Electrotechnical Commission), 1993.

de Karlsruhe, l'université de Cambridge, la société SINTERS et les chantiers navals d'Odense.

## 7.2 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).

## 7.3 Transfert industriel

Le travail de conception et réalisation des robots bipèdes avec le LMS a donné lieu dépôt par ce dernier d'un brevet industriel protégeant la technologie des transmissions mécaniques.

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

# 8 Actions régionales, nationales et internationales

## 8.1 Actions nationales

### 8.1.1 GdR-PRC "Commande de robots à pattes"

B. Espiau est co-responsable avec Carlos Canudas de Wit du projet GdR-PRC du CNRS "Commande de robots à pattes".

### 8.1.2 PSLC : Pôle Systèmes et Logiciels Critiques

Le PSLC est un projet qui a été déposé auprès du Pôle Numérique de la Région Rhône-Alpes, sous l'impulsion du laboratoire CRNS VERIMAG, en regroupant plusieurs équipes de recherche ainsi que plusieurs entreprises de la Région Rhône-Alpes. Il comporte des projets pilotes ayant pour finalité la création d'activité économique. A. Girault participe à ce pôle.

### 8.1.3 Groupe COSED

E. Rutten participe à un groupe de travail de l'EEA consacré à la commande opérationnelle des systèmes à événements discrets<sup>3</sup>. Il succède, en élargissant et renouvelant la thématique, au groupe consacré au Grafcet et aux formalismes pour les automatismes industriels.

---

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

#### 8.1.4 Collaborations internes à l'Inria

- Avec le service des Moyens Robotiques de l'UR, sur ORCCAD et sur la réalisation du bipède.
- Avec le projet MOVI, sur l'asservissement visuel non calibré (doctorant : N. Andreff). Le projet LTR VIGOR, à l'initiative de MOVI, est également une opération commune sur ce sujet.
- Avec le projet VASY, sur la vérification des propriétés temporelles des tâches robotiques en ORCCAD.
- Avec le projet SOSSO dans le cadre de l'action incitative TOLERE.
- Avec le 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).

#### 8.1.5 Collaborations avec d'autres laboratoires

- L'équipe BIP est à l'origine du projet multi-laboratoires BIP ayant pour objectif la réalisation d'un robot bipède mécanique et de sa commande [4]. Les partenaires en sont le LAG (Carlos Canudas de Wit), le laboratoire de mécanique des solides de Poitiers (Guy Bessonnet et Philippe Sardain) et le laboratoire des matériaux à usage sportif de Poitiers (Alain Junqua). Deux prototypes de robots à 15 degrés de liberté ont été réalisés cette année, aboutissement de cinq années de collaboration (Figure 11). Avec le LAG, la collaboration s'élargit à des domaines plus amont et est très active : doctorants en commun, groupes de travail, séminaires.
- Le projet participe à l'Action de Recherche Coopérative (ARC) AVEC (Asservissement visuel en environnement complexe) qui voit coopérer les projets BIP et MOVI (INRIA-RHÔNE-ALPES), ICARE (INRIA-SOPHIA-ANTIPOLIS) et VISTA (INRIA-RENNES).
- Le projet participe, Alain Girault en étant le coordinateur, à l'Action de Recherche Coopérative (ARC) TOLERE<sup>4</sup> (Code réparti tolérant aux pannes pour systèmes embarqués), dont les partenaires sont les projets BIP et SOSSO (INRIA-ROCQUENCOURT).

L'année 2000 marque la fin de l'ARC TOLERE, et nous avons organisé une journée de présentation des résultats, le 24 octobre, qui a réuni les participants de BIP et SOSSO ainsi que Gilles Kahn et quelques spécialistes de la tolérance aux pannes : David Powell (CNRS, LAAS), Pascale Minet (INRIA, projet HYPERCOM) et Isabelle Puaut (INRIA, projet SOLIDOR).

---

4. <http://www.inrialpes.fr/bip/people/girault/Projets/Incitative/index.html>

- Le projet participe, Eric Rutten en étant le coordinateur, à une action locale de recherche coopérative dont les partenaires sont le projet BIP et le LAG (équipe Conception de Systèmes Sûrs, H. Alla), sur le sujet du *Contrôle de Tâches Robotiques : Langages spécialisés et Synthèse de contrôleurs*.

## 8.2 Actions européennes

- Le projet LTR VIGOR a débuté en février 1998.
- Le projet IST TELEDIMOS a débuté en janvier 2000.
- Le projet BIP est membre depuis l’automne 1998 du réseau européen CLAWAR (Climbing and Walking Robots).

## 9 Diffusion de résultats

### 9.1 Animation de la Communauté scientifique

B. Espiau est membre des comités de programme pour IEEE ICRA 2000, CLAWAR 2000, ISRA 2000 ; membre du jury d’habilitation à diriger des recherches de Ph. Even ; membre du jury de thèse de doctorat de R. Kocik (SOSSO) ; direction de thèse de P.B. Wieber, soutenue le 6 décembre.

E. Rutten est membre des comités de programme pour *Euromicro Conference on Real-Time Systems, ERTS’00*, *Salon Real-Time Systems RTS 2001*, *congrès MSR 2001 (Modélisation des Systèmes Réactifs)* et *Euromicro Conference on Real-Time Systems, ERTS’01* ; membre du *BoD* de la société Euromicro ; rapporteur de la thèse de doctorat de Christine Largouët.

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

- Bernard Espiau a donné un séminaire invité au Jet Propulsion Laboratory de la NASA (Pasadena, Californie), mai 2000.
- Eric Rutten a donné des séminaires invités au LSV (Laboratoire Spécification et Vérification), ENS, Cachan, et à l’Université de Bretagne Occidentale, Brest, janvier 2000.
- Alain Girault et Eric Rutten ont participé au *Workshop on Synchronous Languages*, à Hyères en novembre-décembre 1999, et à Saint Nazaire, en novembre-décembre 2000.
- Daniel Simon a participé et donné une conférence à la journée SEE *Robotique et sûreté de fonctionnement*, Paris, Novembre 2000.
- Christine Azevedo a participé à l’*International Symposium on Rehabilitation and Motricity in Engineering and Medicine*, Valladolid, Espagne, 2000.
- Cătălin Dima a participé et donné une conférence à l’école d’été MoVeP’2k à l’Ecole Centrale de Nantes, 19-23 juin 2000.



FIG. 14 – BIP2000 au sein du Pavillon Français de l'Exposition Universelle 2000, à Hanovre

- Bernard Espiau a donné une conférence sur les robots marcheurs aux Journées des Jeunes Chercheurs en Robotique (JJCR), à Rennes, en octobre 2000.

Le projet a invité à donner un séminaire : Fernando Jiménez (EP-ATR, INRIA-RENNES).

## 9.2 Exposition Universelle

Le projet BIP, avec le Laboratoire de Mécanique des Solides (LMS, Poitiers), a participé au Pavillon Français à l'Exposition Universelle 2000, en présentant le robot bipède conçu et développé conjointement. L'exposition a eu lieu à Hanovre, du 1er juin au 31 octobre 2000, dans le cadre de la thématique : Homme - Nature - Technologie. L'exposition du robot bipède dans le Pavillon Français de l'exposition, pendant cinq mois, a nécessité un travail préparatoire conséquent. Ce travail a été réalisé grâce à l'aide de la Délégation à la Communication de l'INRIA, en liaison avec le Commissariat Général du Pavillon Français, maître d'ouvrage, et la société CAP PRODUCTIONS, responsable de la scénographie du Pavillon. La Fig. 14 représente le robot BIP2000 en situation, au sein du Pavillon. BIP2000 a reçu la visite de 27000 personnes par jour en moyenne.

## 9.3 Enseignement

- N. Andreff : monitorat en Informatique à l'INPG.
- C. Azevedo : monitorat en Électronique à l'Université de Savoie jusqu'en septembre 2000, puis en Automatique à l'INPG (ENSIEG).
- B. Espiau : cours de commande de robots, École des Mines de Paris.



- D. Simon et E. Rutten : co-encadrement du stage de TER et d'été de Maîtrise de Fanny Benattar.
- A. Girault et E. Rutten : compilation, en 2ème année de l'ENSIMAG.
- A. Girault, D. Simon et E. Rutten proposent le cours « Programmation temps-réel et réactive » du 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, J.-C. FERNANDEZ, A. GIRAULT, «An Algorithm for Reducing Binary Branchings», *in: Fifteenth Conference on the Foundations of Software Technology and Theoretical Computer Science, FST&TCS'95*, P. Thiagarajan (éditeur), LNCS, 1026, Springer-Verlag, Bangalore, India, décembre 1995.
- [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, à paraître dans *Int. Journal of Robotics Research*.
- [6] C. SAMSON, B. ESPIAU, M. L. 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] P. WIEBER, *Dynamique et commande d'un robot marcheur anthropomorphe*, Thèse de Doctorat en automatique, École Nationale Supérieure des Mines de Paris, 2000, à paraître.

### 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*, 2001, à paraître.
- [10] B. ESPIAU, «La robotique: état des lieux et perspectives», *Encyclopaedia Universalis: la Science au présent*, 2000.

- [11] B. ESPIAU, «Stability Issues in the Control of Anthropomorphic Biped Robots», *ERCIM News*, 42, 2000, p. 37–38.
- [12] F. JIMÉNEZ-FRAUSTRO, E. RUTTEN, «Modélisation synchrone de standards de programmation de systèmes de contrôle: le langage ST de la norme CEI 1131-3», *Revue de l'électricité et de l'électronique (SEE)*, 3, mars 2000, p. 60–68.
- [13] H. MARCHAND, ÉRIC RUTTEN, M. LE BORGNE, M. SAMAAN, «Formal Verification of Programs specified with SIGNAL: Application to a Power Transformer Station Controller», *Science of Computer Programming*, 2000, à paraître.

### Communications à des congrès, colloques, etc.

- [14] N. ANDREFF, B. ESPIAU, R. HORAUD, «Visual Servoing from Lines», *in: IEEE International Conference on Robotics and Automation*, San Francisco, 2000.
- [15] 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. submitted.
- [16] C. AZEVEDO, THE BIP TEAM, «Control Architecture and Algorithms of the Anthropomorphic Biped Bip2000», *in: Proceedings of the International Symposium on Mobile, Climbing and Walking Robots, CLAWAR'00, Madrid, Spain*, 2000.
- [17] C. AZEVEDO, «Architecture et algorithmes de commande du robot anthropomorphe Bip2000», *in: Actes des Journées des Jeunes Chercheurs en Robotique 13, Rennes, France*, 2000.
- [18] 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. submitted (réponse le 15 décembre).
- [19] 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)*, Mantova, Italy, février 2001. à paraître.
- [20] C. DIMA, «A fixpoint semantics for "memory cells" in synchronous dataflows», *in: Proceedings of the 2nd workshop on Fixed Points in Computer Science, Paris, 22-23 juillet*, 2000. à paraître.
- [21] C. DIMA, «Real-Time Automata and the Kleene Algebra of Sets of Real Numbers», *in: Proceedings of STACS 2000, Lille, France*, p. 279–289, 2000. LNCS no. 1770, Springer Verlag.
- [22] B. ESPIAU, P. SARDAIN, «The Anthropomorphic Robot BIP2000», *in: IEEE International Conference on Robotics and Automation*, San Francisco, 2000.
- [23] F. JIMÉNEZ-FRAUSTRO, E. RUTTEN, «Hybrid simulation of IEC-61131 PLC programs using Signal and Simulink», *in: Proceedings of the 4th International Conference on Automation of Mixed Processes, ADPM'00*, 18-19 September, Dortmund, Germany, p. 171–176, 2000.
- [24] B. LAMIROY, N. ANDREFF, B. ESPIAU, R. HORAUD, «Controlling Robots with Two Cameras: How to Do it Properly», *in: IEEE International Conference on Robotics and Automation*, San Francisco, 2000.

- [25] R. MATEESCU, M. SIGHIREANU, «Efficient On-the-Fly Model-Checking for Regular Alternation-Free  $\mu$ -Calculus», *in: 5th International Workshop on Formal Methods for Industrial Critical Systems, FMICS'2000*, Berlin, Germany, avril 2000, <http://www.inrialpes.fr/vasy/Publications/Mateescu-Sighireanu-00.html>.
- [26] D. SIMON, M. PERSONNAZ, R. HORAUD, «TELEDIMOS Telepresence simulation platform for civil work machines: real-time simulation and 3D vision reconstruction», *in: IARP Workshop on Advances in Robotics for Mining and Underground Applications*, Brisbane, Australia, 2-4 Octobre 2000.

### Rapports de recherche et publications internes

- [27] C. AZEVEDO, N. ANDREFF, «Étude expérimentale des premières démarches du robot BIP2000», *Rapport de Recherche n° 4017*, INRIA, 2000.
- [28] C. AZEVEDO, R. PISSARD-GIBOLLET, «Contrôleur du robot Bip2000», *Rapport technique*, INRIA, 2000, à paraître.
- [29] A. GIRAULT, C. LAVARENNE, M. SIGHIREANU, Y. SOREL, «Fault-Tolerant Static Scheduling for Real-Time Distributed Embedded Systems», *Rapport de Recherche n° 4006*, INRIA, septembre 2000, (Submitted to IEEE Trans. on Parallel and Distributed Systems).

### Divers

- [30] M. PERSONNAZ, «How to estimate the internal parameters with Tele2 version2», Novembre 2000, Rapport Teledimos.
- [31] M. PERSONNAZ, «Perception S/S: Fonctionnal Specifications», Juillet 2000, Rapport Teledimos.
- [32] D. SIMON, «Real-time Simulator: Updated Fonctionnal Specification Document and Prototype User's Manual», Juillet 2000, Rapport Teledimos.