



## COUPLAGE D'UN SYSTEME DE SIMULATION DE ROBOT

## AVEC UN TERMINAL DE SYNTHESE D'IMAGE

*Coupling of a robot simulation system to an image generation terminal*

C. LAUGIER\* - J. PERTIN-TROCCAZ

LIFIA/IMAG - BP 68 - 38402 ST MARTIN D'HERES CEDEX

**RESUME**

La simulation graphique est potentiellement un excellent outil pour développer et mettre au point des programmes de commande de robots en dehors du site d'opération. Le système que nous présentons dans ce papier permet de réaliser une simulation réaliste de n'importe quelle tâche de manipulation décrite à l'aide du langage LM (LM est un langage de commande de robot défini et implanté au laboratoire LIFIA-IMAG). Ce système inclut des fonctions pour contrôler la cinématique de la tâche ainsi qu'un ensemble de commandes interactives permettant d'"émuler" le processus de simulation (par exemple: entrée de données de capteurs, introduction de perturbations aléatoires dans le modèle de l'univers, modification de l'échelle de temps). Au cours de la présentation, une attention particulière sera portée sur les aspects dynamiques et graphiques. Les résultats obtenus seront illustrés par un film d'animation réalisé à partir d'un programme LM décrivant le montage d'un train miniature. Les images qui composent ce film ont été synthétisées par le terminal HELIOS.

**SUMMARY**

Graphic simulation can be a valuable tool for designing and debugging off-line robot control programs. The system described in this paper allows the user to obtain a realistic simulation for any manipulation task described using the LM language (LM is a robot control language designed and implemented at the LIFIA-IMAG laboratory). This system includes both functions to handle the kinematics of the task, and interactive commands to "emulate" the simulation process according to external data (for example : data provided by sensors, random perturbations introduced into the world data or modification of the time basis). In the paper, we will focus on the dynamic and the graphic features of the system. Results will be illustrated by an animated film produced using an LM program describing the assembly of a miniature train. The shaded pictures of the film have been synthesized by the HELIOS graphical terminal.

**1. INTRODUCTION**

Les systèmes informatiques utilisant des procédés de synthèse d'images sont employés dans des domaines divers incluant les productions audiovisuelles, les jeux, la CAO et la simulation. Les solutions techniques apportées pour répondre aux besoins de ces différents domaines d'application sont en général très différentes tant au point de vue du matériel que de celui du logiciel. Dans le domaine de la simulation, les efforts ont initialement porté sur la recherche de processeurs graphiques rapides, permettant de résoudre le problème crucial du temps réel. Malgré certains résultats spectaculaires (mais coûteux), les matériels développés sont limités à la visualisation rapide (1/25 seconde) de scènes tridimensionnelles observables selon des points de vue variables (exemple : défilement de paysages, mouvement simulé d'un avion...). Les besoins nouveaux créés par l'essor de la productique, ont suscité de nouvelles recherches orientées vers la modélisation de processus physiques, mettant en jeu des mouvements complexes de mécanismes articulés. Le problème posé est alors celui de la simulation graphique de programmes de commandes de robots. Ces programmes nécessitent pour leur développement et pour leur mise au point une excellente maîtrise de l'univers tridimensionnel du robot. Or cette maîtrise est difficile à acquérir sans l'aide d'outils informatiques appropriés.

Plusieurs systèmes de simulation de robot ont été développés dans différents laboratoires de recherche. Certains opèrent à partir de représentations symboliques bidimensionnelles [7,8]. D'autres sont basés sur l'utilisation de systèmes de CAO : CATIA [9,10], GEOMAP [11], SAMMIE [12,13], GDP [14]. Enfin, quelques uns seulement ont été implantés en connexion avec un langage de commande de robot : AML [14], AL [17,18], LM [19]. Une étude comparative des simulateurs existants est présentée dans [1]. Cette étude montre que les possibilités graphiques offertes (degré d'interactivité, qualité des images, rapidité...) varient largement d'une implantation à l'autre. Elle montre également que tous les systèmes précités (ormis EMULA [14] et le simulateur LM) permettent difficilement d'obtenir une simulation "réaliste" des tâches de manipulation. Ceci provient du fait que le déroulement d'un programme de commande de robots varie d'une exécution à l'autre. Ces variations sont dues aux imperfections de la structure mécanique, aux incertitudes relatives aux objets manipulés et aux incidents mécaniques mineurs tels que les glissements. Elles ne sont perçues qu'au travers des informations transmises par les capteurs. Une simulation réaliste passe donc par une gestion élémentaire des capteurs.

Dans ce papier, nous décrivons un système capable de



AVEC UN TERMINAL DE SYNTHESE D'IMAGE  
C. LAUGIER\* - J. PERTIN-TROCCAZ

de réaliser une simulation graphique aussi complète que possible, de n'importe quelle tâche de manipulation décrite à l'aide du langage LM (ce langage est un langage de commande de robot défini et implanté au laboratoire LIFIA-IMAG [2]). Ce système inclut des fonctions pour contrôler la cinématique de la tâche, ainsi que des fonctions pour "émuler" le processus de simulation en fonction de données externes (typiquement données transmises par des capteurs). Il est composé de deux modules indépendants : le simulateur graphique implante en MACLISP sur un ordinateur LSI-11/23. Le simulateur graphique utilise deux systèmes développés par notre équipe : un système graphique conçu pour la manipulation et la visualisation de scènes tridimensionnelles [3], et un système de modélisation capable de manipuler les chaînes cinématiques ouvertes [4]. Les images produites par le simulateur sont synthétisées à l'aide du terminal HELIOS [5]. Un film d'animation montrant l'assemblage d'un train miniature a été réalisé à l'aide du système décrit dans ce papier (cf. annexe).

## 2. DESCRIPTION FONCTIONNELLE DU SIMULATEUR

Trois types de données sont nécessaires pour la simulation graphique de robots :

- un modèle géométrique des objets rigides complété par un modèle cinématique du robot,
- une description de la tâche de manipulation à simuler,
- une représentation des événements à nature physique qui ne peuvent être décrits formellement dans le modèle de l'univers du robot (perturbations par exemple).

Les données du type "modèle géométrique" sont potentiellement présentes dans les bases de données de la CAO. Elles incluent une représentation à base de facettes planes des enveloppes externes de chaque objet rigide présent dans la scène (y compris les éléments constitutifs du robot). Les données de type cinématique sont incorporées à la base de modèles géométriques par

un programme de modélisation de mécanismes. Dans l'implantation actuelle du simulateur, cette base de données est construite à l'aide d'un sous-ensemble du système de modélisation SMGR [4].

Les informations relatives à la tâche de manipulation proviennent d'un programme écrit en langage LM. Elles sont transmises par l'interpréteur LM sous la forme de consignes adressées d'une part aux variables articulaires du manipulateur, et d'autre part aux mécanismes de "mise en service" des capteurs (cf. figure 1).

Les informations concernant le monde physique sont introduites dans le système par l'opérateur. Cette méthode permet d'émuler le processus de simulation en modifiant certains paramètres externes (cf. § 3). Ces paramètres représentent les variations de la base de temps, les données transmises par les capteurs, et les événements physiques (glissements par exemple) introduits dans le modèle de l'univers sous la forme de perturbations aléatoires.

L'ensemble des informations précédentes est utilisé par le simulateur pour construire et maintenir à jour un modèle structuré de l'univers du robot. C'est ce modèle qui évolue au cours de la simulation et qui est périodiquement visualisé par les fonctions graphiques du système. Le simulateur se comporte alors comme un pseudo-robot capable d'une part d'exécuter les instructions du programme de manipulation sur le terminal graphique, et d'autre part d'enregistrer des informations numériques concernant des positions clés et des trajectoires particulières. Dans le premier mode opératoire, le simulateur contrôle la cinématique du robot à partir des informations transmises par l'interpréteur LM. Ces informations correspondent aux différentes valeurs prises par les variables articulaires du manipulateur. Dans l'autre mode opératoire, l'utilisateur contrôle manuellement le robot à l'aide d'un dispositif d'interaction graphique (boule roulante, tablette à digitaliser, clavier de fonctions...). Il travaille alors en mode apprentissage de manière à décrire certaines positions ou trajectoires nécessaires à l'exécution de la tâche programmée. Il peut pour cela manipuler individuellement chaque articulation du robot, ou déplacer directement l'outil terminal en utilisant des fonctions graphiques spécialisées du type de celles introduites dans le système CATIA [9].

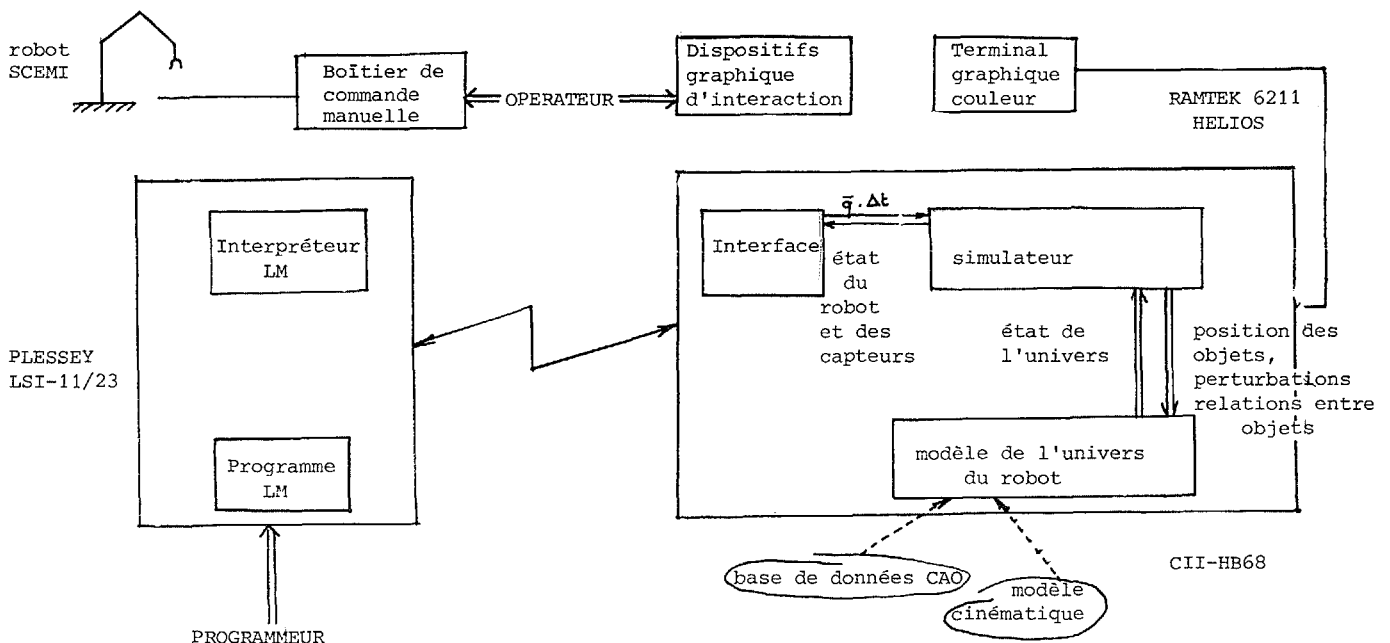


Figure 1 : Structure fonctionnelle du système



### 3. ROLE DES COMMANDES DE SIMULATION

Les commandes de simulation sont codées de manière à être aisément utilisables par un non spécialiste. Elles peuvent être indifféremment insérées dans le programme de manipulation, ou "entrées sur le vif" par l'opérateur à l'aide des mécanismes d'interaction. Dans le premier cas, l'utilisateur obtient après chaque simulation une représentation animée complète de la tâche de manipulation. Il peut ainsi tester la robustesse de son programme vis à vis de différentes situations physiques, en réalisant plusieurs exécutions consécutives. Dans l'autre cas, l'opérateur travaille "en direct" avec le processus de simulation afin de mettre au point ou de compléter le programme concerné. Il peut alors stopper à tout moment l'exécution de la tâche de manière à modifier interactivement le contexte de la simulation (modèle de l'univers du robot, modèle de visualisation et paramètres externes). Il peut également utiliser des fonctions géométriques à base de calculs d'interférence, pour analyser des situations physiques particulières (collisions et contacts) difficiles à déceler sur certaines images produites par le système.

Les commandes utilisables par l'opérateur sont regroupées en trois ensembles permettant respectivement d'interagir avec le modèle de l'univers, de contrôler les images produites par le système et d'entrer les paramètres externes nécessaires à l'émulation du processus de simulation. Les modules auxquels s'adressent ces commandes sont décrits plus en détail dans la suite de l'article. Le lecteur intéressé pourra trouver dans [19] une présentation complète de la sémantique et de la syntaxe associées aux commandes du système.

### 4. LA GESTION DYNAMIQUE DU MODELE

La gestion du modèle de l'univers est réalisée au moyen de trois types d'opérations : (1) le positionnement d'un nouvel objet dans l'espace de travail du robot, (2) le déplacement automatique ou contrôlé de n'importe quel composant de l'univers (objets isolés ou éléments constitutifs du robot), et (3) la modification de la structure relationnelle du modèle.

Les commandes associées aux opérations (1) et (2) sont utilisées pour décrire la dynamique de la scène modélisée. Les autres commandes permettent de définir des contraintes physiques temporaires ou permanentes portant sur des objets particuliers du modèle. Ces objets forment alors des "structures rigides", c'est-à-dire des groupements d'objets solidaires dans leurs déplacements. De telles situations se produisent, par exemple, lorsque des pièces mécaniques ont été assemblées, ou lorsque la pince du robot saisit un objet ou un outil particulier.

Chaque commande de mouvement adressée au simulateur donne lieu à un déplacement instantané d'un ou de plusieurs objets, et/ou à une "déformation" de la structure articulée du robot. Lorsque les ordres de déplacement sont engendrés automatiquement à partir d'instructions du programme de manipulation, la fréquence d'envoi des consignes de position dépend de la base de temps utilisée par le système de commandes du robot (cf. § 6). Dans ce cas, les différentes configurations de la structure mécanique sont calculées en fonction des lois cinématiques incluses dans le modèle. Ce calcul est réalisé en deux étapes complémentaires (cf. figure 2) : (1) la détermination des valeurs articulaires à partir des positions de l'outil terminal (résolution de la cinématique inverse), et (2) l'évaluation des expressions analytiques permettant de définir les positions absolues de chaque composant du robot à l'aide des données précédentes (résolution de la cinématique directe). Le lecteur intéressé pourra

trouver dans [20] une présentation succincte des différentes méthodes utilisables pour résoudre le problème de la cinématique inverse.

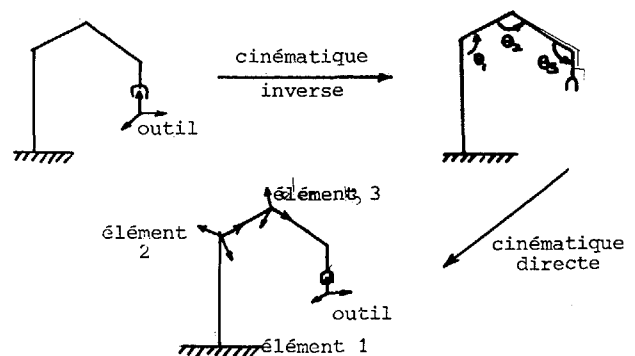


Figure 2 : Exemple de traitement d'une structure mécanique.

### 5. LA GESTION DE L'IMAGE

#### 5.1 Le contrôle de l'image

Les images produites par le système (logiciel + matériel) sont construites à l'aide d'un modèle de visualisation comportant : (1) une description de l'éclairage de la scène, (2) une représentation de l'aspect visuel des objets, et (3) une caractérisation des paramètres d'observation.

Les données liées à l'éclairage et à l'aspect visuel sont pour la plupart traitées directement par le terminal HELIOS lors de l'affichage des images [5]. Ces données sont généralement stockées dans le modèle. Elles peuvent être modifiées à l'aide de certaines commandes du système.

Les paramètres d'observation définissent les éléments nécessaires au calcul des constituants de l'image. Ils jouent de ce fait un rôle important, tant au niveau de la sélection des scènes à visualiser, qu'au niveau de la description des déplacements de "caméra" imposés par les prises de vues. Afin d'avoir un mode de représentation naturel et synthétique de ces paramètres, le simulateur utilise un modèle basé sur le concept de "sphère de vision" (cf. [3]). Le principe consiste à caractériser l'espace observable par une sphère sur laquelle est positionnée une caméra mobile. La latitude, la longitude et la distance focale de cette caméra peuvent être modifiées aisément par l'opérateur à l'aide d'un ensemble de commandes appropriées. La méthode utilisée conduit à une description rapide et claire des modifications du point de vue (cf. annexe). Elle permet également d'engendrer sans efforts des mouvements de caméra destinés à créer des effets spéciaux-tels que des travellings, des panoramiques ou des suivis de corps mobiles. Il suffit, par exemple de fixer le "centre de vision" sur un objet en mouvement, pour contraindre la caméra à suivre automatiquement cet objet au cours de ses évolutions. Dans le même ordre d'idée, l'opérateur peut appliquer un zoom centré sur la pince du robot de manière à maintenir au centre de l'écran une vue détaillée de cette pince, quels que soient les mouvements exécutés par le robot.

#### 5.2 La synthèse de l'image

La synthèse de chaque image est réalisée à partir des informations courantes contenues dans la base de données géométriques et dans le modèle de visualisation. Elle nécessite, pour sa mise en oeuvre, l'utilisation de



deux processus complémentaires permettant de pré-traiter géométriquement les données de l'image, puis d'afficher celle-ci sur le terminal graphique.

La phase de pré-traitement est exécutée par les fonctions de visualisation du simulateur. Elle comporte en séquence les opérations suivantes :

- (1) identification des objets présents dans la sphère de vision et calcul de la position de ces objets par rapport au repère associé à la caméra,
- (2) sélection des objets situés dans le champ visuel,
- (3) suppression des faces cachées (faces orientées "vers l'arrière");
- (4) classement par la méthode des coefficients statiques [21] des faces restantes, de manière à obtenir une liste ordonnée de "l'arrière vers l'avant",
- (5) application de la transformation perspective permettant de passer à une représentation bi-dimensionnelle de la scène,
- (6) coupage de l'image avec les bords de l'écran.

L'affichage de l'image avec élimination des parties cachées est réalisé par le terminal HELIOS qui reçoit pour cela quatre types de données : (1) une description des conditions d'éclairage, (2) une liste ordonnée des contours polygonaux à afficher, (3) une liste des normales extérieures associées aux faces des objets et, (4) une représentation des paramètres d'aspect visuel. Le rôle du terminal consiste alors à remplir la "mémoire de trame" à partir des contours polygonaux, puis à calculer en temps réel la texture et l'ombrage de chaque point de l'image [6].

## 6. LA GESTION DES PARAMÈTRES D'EMULATION

Les commandes d'émulation permettent de modifier la base de temps, d'introduire des perturbations aléatoires dans le modèle de l'univers, et de simuler les capteurs.

Les paramètres temporels sont utilisés pour définir le cycle de base du simulateur, et pour spécifier la fréquence des opérations de visualisation. Le modèle de l'univers est ainsi mis à jour à chaque cycle, alors que les opérations de visualisation n'interviennent qu'à des intervalles de temps définis comme des multiples du cycle de base. Cette méthode permet de conserver un modèle cohérent avec la réalité, quel que soit le rythme de visualisation choisi. Elle permet également d'obtenir des ensembles d'images utilisables pour des films d'animation, en fixant la vitesse de production d'images à 24 images/seconde, et en choisissant un cycle de base égal à celui employé par le système de commande du robot.

Les perturbations peuvent être appliquées aux objets rigides, et aux éléments du robot. Elles sont engendrées par le biais d'une fonction aléatoire fournissant des valeurs choisies dans un intervalle spécifié par l'opérateur.

La simulation des capteurs peut être réalisée soit de manière manuelle, soit de manière automatique. Dans le premier cas, l'utilisateur doit stopper le mouvement lorsqu'il estime qu'un seuil a été atteint, ou qu'une expression booléenne incluant des données de capteurs est devenue vraie. Ce procédé est employé lorsque les informations manipulées sont trop complexes pour être calculées automatiquement par le système (par exemple: valeur de la composante horizontale de la force exercée sur la pince du robot). Dans l'autre mode, la gestion des capteurs est réalisée à l'aide de fonctions géométriques spécialisées basées sur le calcul des

interférences [22]. Ces fonctions permettent de détecter des contacts, ou de déterminer la présence ou l'absence d'objets dans une zone particulière de l'espace (simulation d'un capteur de proximité). La méthode utilisée est inspirée de celle employée dans le système EMULA [14]. Elle consiste à associer dynamiquement les fonctions avec les capteurs concernés. Ainsi, à chaque cycle, toutes les fonctions validées sont invoquées, et les résultats sont affectés automatiquement aux capteurs correspondants.

## 7. MODELISATION DE L'UNIVERS

### 7.1 Le modèle de l'univers

Chaque corps physique (objet isolé ou composant du robot) est représenté par un repère cartésien associé à un ensemble de faces polygonales. Le repère cartésien permet au système de localiser l'objet dans l'espace de travail du robot ; les ensembles de faces sont utilisés pour les opérations de visualisation. La structure de l'univers est caractérisée par un graphe dans lequel chaque arc dénote une relation particulière entre deux entités du modèle, et chaque noeud symbolise un objet de l'univers. Les informations transportées par les arcs représentent des transformations géométriques paramétrées ou constantes. Les données associées aux noeuds du graphe correspondent soit à de simples repères cartésiens (cas des objets fictifs ou des objets composés de plusieurs parties), soit à des listes de faces (cas des objets solides).

Trois types de structures sont utilisées dans le modèle [19] : (1) une structure de type hiérarchique induite par les relations spatiales liant les objets de l'univers, (2) une structure arborescente associée à chaque mécanisme articulé (cf. § 7.2) et (3) une structure de graphe évolutive caractérisant les "structures rigides" temporaires ou permanentes introduites au § 4. Ce mode de représentation multiple permet au système d'optimiser les opérations de mise à jour du modèle, en appliquant plusieurs méthodes de calcul choisies en fonction du contexte. Un parcours de la structure arborescente est, par exemple, tout à fait adapté au mode de calcul induit par les équations cinématiques incluses dans le modèle (cf. figure 3).

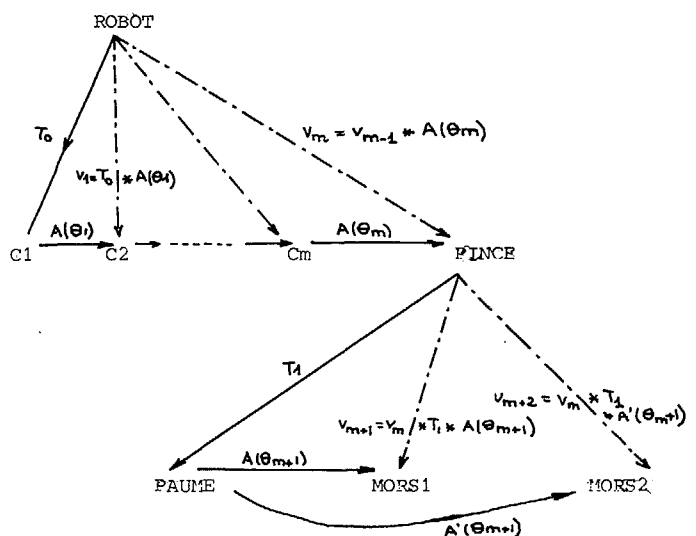


Figure 3 : Evaluation de la structure arborescente associée au modèle cinématique du robot



## 7.2 Le modèle cinématique du robot

Le robot est représenté par une "boucle cinématique ouverte" constituée de  $n$  composants solides liés entre eux par des articulations. Deux types d'articulations de base sont utilisés pour cette description : l'articulation rotoïde qui permet des déplacements en rotation autour d'un axe fixe, et l'articulation prismatique qui autorise des mouvements de translation exécutables dans une direction donnée. D'autres articulations plus complexes peuvent également être décrites à l'aide de ces deux types de liaisons élémentaires. Une liaison de type rotule (trois degrés de liberté en rotation) peut par exemple être représentée par une combinaison de trois liaisons rotoïdes à axes concourants. Dans ce cas, deux éléments fictifs représentés par des repères cartésiens sont introduits automatiquement par le système dans la chaîne cinématique. Certains mécanismes comportent  $n$  corps physiques ( $n > 2$ ) reliés entre eux par une liaison unique. C'est le cas par exemple de la pince du robot qui est constituée d'un corps de base (la "paume") sur lequel s'articulent deux éléments mobiles (les mors). Ce type de mécanisme est décrit dans le modèle à l'aide d'articulations particulières : les articulations poly-rotoïdes et poly-prismatiques. De telles articulations ne constituent que de simples extensions des liaisons élémentaires décrites précédemment. Elles conduisent à répartir sur plusieurs corps une commande de mouvement unique.

Quelques robots possèdent dans leur mécanisme des boucles cinématiques élémentaires de type parallélogramme. Ces boucles, qui permettent d'améliorer les performances mécaniques du bras, sont des "boucles passives", c'est-à-dire des boucles dont les articulations sont commandées directement par la structure mécanique elle-même. Au niveau du modèle, de telles structures sont représentées au moyen d'articulations de type c-rotoïde permettant de définir un couplage entre les articulations "passives" de la boucle et l'articulation "active" commandant le mécanisme (cf. figure 4). Cette méthode, inspirée de celle utilisée dans le système CATIA [9], permet de conserver la structure arborescente adaptée au calcul de la cinématique, sans augmenter artificiellement le nombre de degrés de libertés.

Sur le plan analytique, chaque élément du mécanisme est représenté par un repère cartésien ayant son axe  $z$  colinéaire à l'axe de rotation ou de translation de l'articulation qui commande le mouvement du corps suivant. Les transformations géométriques liées au passage d'un repère à un autre sont représentées par des matrices exprimées en coordonnées homogènes. Chacune de ces matrices est paramétrée par trois valeurs constantes et une valeur variable (la variable articulaire). Ce mode de représentation est décrit plus en détail dans [19].

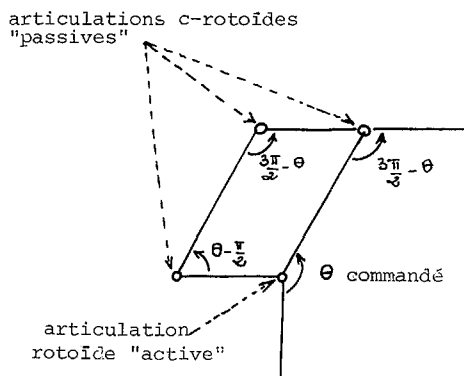


Figure 4 : Structure de type parallèle

## 8. CONCLUSION ET PERSPECTIVES

Le système décrit dans ce papier offre à l'utilisateur de robots industriels un outil puissant pour développer et mettre au point des programmes de commande de robots : il accepte simultanément le langage utilisé pour programmer le robot réel (langage LM en l'occurrence), et le langage employé par le simulateur graphique pour faire le lien avec une base de données CAO. De plus, la répartition des tâches entre deux calculateurs différents permet de pratiquer indépendamment la programmation "en ligne" et la programmation "hors ligne". Dans l'implantation actuelle du système, le simulateur graphique est localisé sur un ordinateur CII-HB68, et l'exécutif LM est situé sur un micro-calculateur LSI-11/23 contrôlant un robot SCEMI six axes.

Les opérations de visualisation du modèle sont réalisées à l'aide des fonctions graphiques du système LISP-3D [3] et des fonctions câblées du terminal HELIOS [5]. Le calcul d'une image composée d'environ six cents faces polygonales nécessite, dans la configuration actuelle du système, entre 4 et 5 secondes de temps CPU. Dans la prochaine version du simulateur, les performances graphiques seront améliorées de manière à obtenir des temps de réponse compatibles avec les impératifs de "temps réel" (plusieurs images complexes par seconde). Ces améliorations seront apportées par le biais d'un processeur graphique spécialisé (à base d'un microprocesseur 68 000), permettant de prendre en charge la majorité des calculs liés au pré-traitement de l'image.

Les concepts développés au cours de cette étude débordent largement du cadre de l'application traitée. En particulier, le système tel qu'il a été conçu possède potentiellement les principales fonctions requises pour la création de séquences animées d'images de synthèse :

- l'image est créée à partir de scènes tridimensionnelles structurées, comportant des corps solides et des objets articulés décrits dans une base de modèles géométriques,
- les mouvements sont engendrés par des commandes de haut niveau basées sur l'utilisation de modèles cinématiques assurant un comportement "autonome" aux objets articulés (il suffit de déplacer "globalement" l'objet pour obtenir automatiquement les déformations correspondantes de sa structure),
- la fréquence de production des images est contrôlée indépendamment de la base de temps associée au processus dynamique, ce qui permet de travailler en mode interactif avec des cadences plus lentes sans nécessairement ralentir les mouvements engendrés par le système,
- l'utilisation de fonctions géométriques de type "capteur" permet de donner aux objets animés un début de comportement "intelligent", en leur associant des "actions réflexes" déclenchées par des événements qui leur sont étrangers (par exemple : arrêt sur un obstacle ou modification de la trajectoire suivie lors de la détection d'un autre objet dans une région particulière de l'espace environnant).

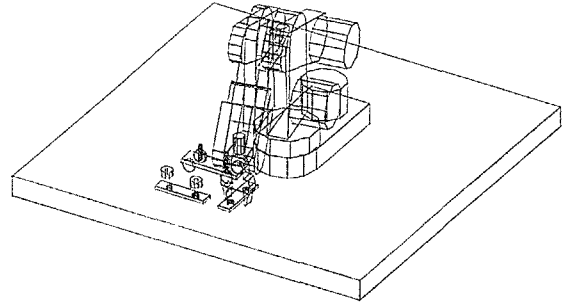
D'autres études réalisées en Intelligence Artificielle dans le domaine de la synthèse automatique de programmes de commande de robots, peuvent également avoir un impact important sur l'évolution future des systèmes de production de films de synthèse. En particulier, certaines méthodes développées en vue d'automatiser la saisie des objets [23] ou de rechercher des trajectoires qui évitent les collisions [24], présentent un intérêt évident pour la génération de scénarios. Il est clair que les techniques d'Intelligence Artificielle combinées avec des fonctions à caractère géométrique trouveront dans un futur proche un écho de plus en plus favorable de la part des concepteurs de systèmes de production d'images de synthèse.



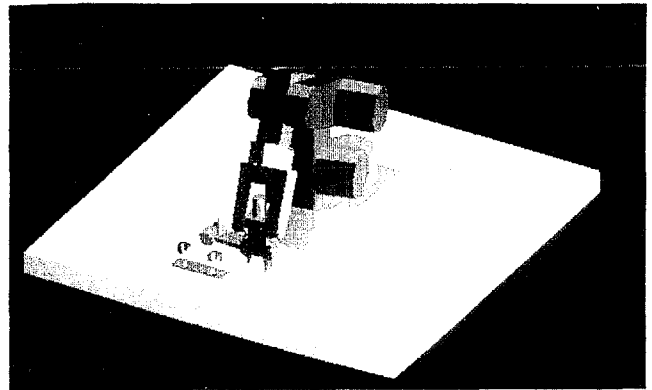
AVEC UN TERMINAL DE SYNTHESE D'IMAGE  
C. LAUGIER\* - J. PERTIN-TROCCAZ

## ANNEXE

```
PROGRAMME FILM ;
TRANSFORMATION T1, T2, T3, T4 ;
REPERE PLANCHE 1, PLANCHE 2, CYLINDRE 1, CYLINDRE 2,
SUPPORT ;
REPERE PRISE_P1, PRISE_P2, PRISE_C1, PRISE_C2 ;
REPERE MONTAGE_P1, MONTAGE_P2, MONTAGE_C1, MONTAGE_C2 ;
REPERE POSAGE_P1, POSAGE_P2, POSAGE_C1, POSAGE_C2 ;
```



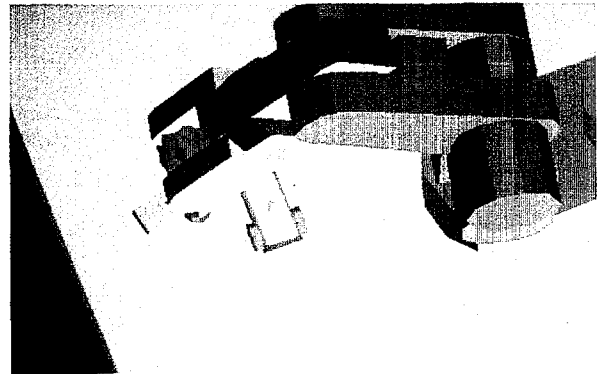
```
FICHER SIMUL ;
FICHER ENTREE SUPPORT ;
FICHER ENTREE PIECE1 ;
FICHER ENTREE PIECE2 ;
DEBUT
Ecrire "PLACER PLANCHE1 T VX -95. T VY 330. T VZ 103."
DANS SIMUL ;
Ecrire "PLACER PLANCHE2 T VX 165. T VY 460. T VZ 103.
R VZ 90." DANS SIMUL ;
Ecrire "PLACER ROND1 T VX 75. T VY 410. T VZ 103."
DANS SIMUL ;
Ecrire "PLACER ROND2 T VX 175. T VY 410. T VZ 103."
DANS SIMUL ;
Ecrire "PLACER WAGON T VX -27. T VY 290.5 T VZ 130.5"
DANS SIMUL ;
Ecrire "CAMERA RH -60." DANS SIMUL ;
Ecrire "IMAGE" DANS SIMUL ;
Ecrire "PAUSE 4." DANS SIMUL ;
Ecrire "TRAVELLING RH 30. 6" DANS SIMUL ;
Ecrire "PAUSE 2." DANS SIMUL ;
FIXER VITESSE ROBOT A 0.2
```



## TRAITEMENT DE LA PREMIERE PLANCHE

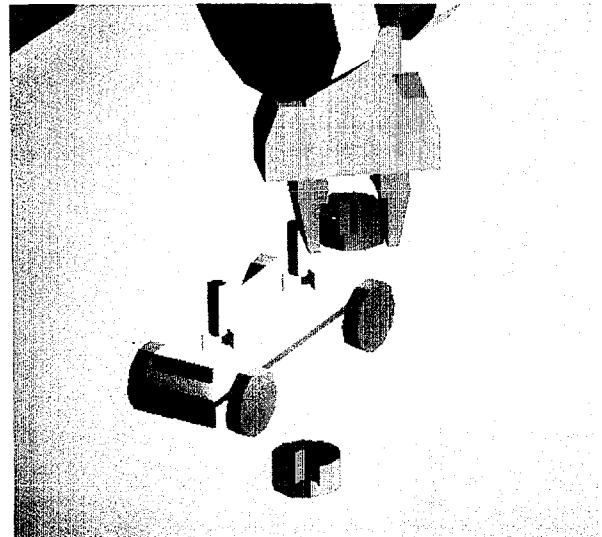
```
LIRE SUPPORT, T1, T2 DANS SUPORT ;
PLANCHE1 := SUPPORT*T2 ;
POSAGE_P1 := SUPPORT*T1 ;
LIRE T1, T2, T3 DANS PIECE1 ;
LIER PRISE_P1 A PLANCHE1 PAR T1 ;
LIER MONTAGE_P1 A PLANCHE1 PAR T2 ;
LIER POSAGE_P2 A PLANCHE1 PAR T3 ;
CO SAISIR ;
DEPLACER ROBOT A PRISE_P1 TRANSLAT(VZ,80.) ;
DEPLACER ROBOT A PRISE_P1 ;
ECARTER PINCE A 40. ;
LIER PRISE_P1 A ROBOT ;
Ecrire "SAISIR PLANCHE1" DANS SIMUL ;

DEPLACER PRISE_P1 DE TRANSLAT(VZ,60.) ;
CO MONTER ;
DEPLACER MONTAGE_P1 A POSAGE_P1*TRANSLAT(VZ,60.) ;
DEPLACER MONTAGE_P1 A POSAGE_P1 ;
ECARTER PINCE A 50.;
DELIER ROBOT DE PRISE_P1 ;
Ecrire "LACHER" DANS SIMUL ;
Ecrire "PAUSE 1." DANS SIMUL ;
Ecrire "CAMERA RH 40." DANS SIMUL ;
Ecrire "CENTREV WAGON" DANS SIMUL ;
Ecrire "IMAGE" DANS SIMUL ;
Ecrire "PAUSE 1." DANS SIMUL ;
Ecrire "TRAVELLING Z 4.0 3" DANS SIMUL ;
DEPLACER ROBOT DE TRANSLAT(VZ,60.)
```



## TRAITEMENT DE LA DEUXIEME PLANCHE

```
FIXER LA VITESSE ROBOT A 0.3 ;
```





#### BIBLIOGRAPHIE

- [1] - C. LAUGIER  
"Combining a high-level language and graphics for robot programming", Colloque Automation et Robotique, Helsinki, Oct. 83
- [2] - J.C. LATOMBE - E. MAZER  
"LM : a high-level programming language for controlling assembly robots", 11th ISIR, Oct. 81, Tokyo
- [3] - C. LAUGIER  
"LISP3D : logiciel pour la manipulation et la visualisation de scènes tridimensionnelles", IMAG, Grenoble, RR n° 328, Sept. 82
- [4] - J. PERTIN-TROCCAZ  
"SMGR : un système de modélisation géométrique et relationnelle pour la robotique", IMAG, Grenoble, RR n° 422, (to appear)
- [5] - F. MARTINEZ - F. FERREIRA  
"HELIOS : terminal interactif pour la synthèse d'images réalistes", congrès AFCET-TTI, Gyf sur Yvette, 1981
- [6] - F. MARTINEZ  
"Vers une approche systématique de la synthèse d'image- aspects logiciel et matériel", Thèse d'Etat, INPG, Grenoble, Nov. 82
- [7] - J. MEYER - R. JAYARAMAN  
"Simulating robotic applications on a personal computer", Computer in Mechanical Engineering, July 83
- [8] - T. ARAI  
"A robot language system with a colour graphic simulator", Proc. of Advanced software in Robotics, Liege, May 83
- [9] - A. LIEGEOIS - E. DOMBRE - P. BORREL  
"Développement d'un système de CAO et de simulation de robots manipulateurs", Premières journées ARA, Poitiers, Sept. 82
- [10] - P. BORREL - F. BERNARD - A. LIEGEOIS -  
D. BOURCIER - E. DOMBRE  
"The robotics facilities in the CAD-CAM CATIA system", edited by B. Rooks, IFS publications, 1983
- [11] - T. SATA - F. KIMURA - A. AMANO  
"Robot simulation system as a task programming tool", 11th ISIR, Tokyo, Oct. 82
- [12] - W.B. HEGINBOTHAM - M. DOONER - K. CASE  
"Rapid assessment of robot performance by interactive computer graphics", Proc. of the 9th ISIR, Washington DC, 1979
- [13] - M. DOONER - N.K. TAYLOR - M.C. BONNEY  
"Planning robot installations by CAD", Computer Aided Design Conference, Brighton, March 82
- [14] - J. MEYER  
"An emulation system for programmable sensory robots", IBM J. Res. Develop, Vol. 25, n° 6, Nov. 81
- [15] - D. ZUHLKE  
"Graphic simulation of robot actions", WZL TH Aachen, Germany, 1983 (unpublished)
- [16] - Y. TSUJIDO - N. KODAIRA - M. OSHINA  
"Realtime motion simulator of robots", Proc. of ICAR 83, Tokyo, 1983
- [17] - B.I. SOROKA  
"Debugging robot programs with a simulator", CAD/CAM-8 Conference, Anaheim, California, Nov. 80
- [18] - B.I. SOROKA  
"A robot simulator", submitted to Robotic Research, January 82

- [19] - C. LAUGIER - C. EVIEUX - J. PERTIN-TROCCAZ  
"Un système de simulation graphique de robots incluant une gestion élémentaire des incidents et des capteurs", IMAG, Grenoble, RR n° 421, March 84
- [20] - J.G. QUEROMES  
"Computer aided design and robotics : a full of promise cooperation", 12th ISIR, Paris, June 82
- [21] - I.E. SUTHERLAND, R.F. SPROULL, R.A. SCHUMACKER  
"A characterization of Ten Hidden-Surface algorithms", ACM comp. surveys, vol. 6, n° 1, 1984
- [22] - J.W. BOYSE  
"Interference detection among solids and surfaces", CACM, vol. 22, nb. 1, January 79
- [23] - C. LAUGIER  
"A program for automatic grasping of objects with a robot-arm", 11th ISIR, Tokyo, October 81
- [24] - T. LOZANO-PEREZ - M.A. WESLEY  
"An algorithm for planning collision-free paths among polyedral obstacles", Communication of the ACM, October 79

#### REMERCIEMENTS

Le travail présenté dans ce papier a été financé en partie par le projet ARA (Automatique et Robotique Avancées) et par l'ADI (Agence de l'Informatique). Les nouveaux développements concernant le processeur graphique spécialisé sont réalisés en collaboration avec la société ITMI (Industrie et Technologie de la Machine Intelligente).