

Programmation des robots

Nous allons analyser dans ce dossier les divers moyens de fournir aux moteurs les informations nécessaires pour les mouvoir ; ces informations doivent permettre de réaliser des trajectoires plus ou moins complexes, selon le travail à effectuer.

A travers différentes étapes, les informations de l'armoire de commande du robot vont être transformées en mouvements, ceux-ci seront asservis en position et (ou) en vitesse permettant ainsi de commander plusieurs axes dans l'espace.

Chaque axe d'un robot est équipé de capteurs qui détermineront position et vitesse au cours du déplacement.

Dans une programmation par apprentissage, par exemple, les positions prises par ces capteurs en mouvement sont mesurées et mises en mémoire dans l'armoire du robot. Au cours de la phase travail, l'ordinateur va chercher dans sa mémoire les informations correspondant aux coordonnées des points enregistrées.

Les trajectoires

L'opérateur, pour faire exécuter une tâche à un robot, peut travailler en trajectoire en point à point, ou en trajectoire en continu.

Le choix de la méthode dépendra du travail auquel est destiné le robot mais aussi de ses capacités, d'où la nécessité de choisir judicieusement son robot, pour parer à toute éventualité. Les premiers robots utilisés industriellement étaient programmés en point à point et destinés principalement à des opérations de soudage par points ou de manutention,

cas pour lesquels il n'est pas nécessaire de contrôler continuellement la trajectoire de l'organe terminal, mais où, par contre, il faut aller le plus vite possible d'un point à un autre, la trajectoire suivie entre deux points pouvant être différente d'un cycle sur l'autre.

Cette imprécision est d'autant plus faible que la distance qui sépare deux points est courte ; l'unité de pilotage donne à chaque axe sa loi de déplacement, comprenant le temps du mouvement, sa vitesse et son accélération.

Plusieurs cas existent.

- Tous les axes peuvent être mûs pendant un temps identique. Dans ce cas, le système analyse quel

est l'axe le plus lent en vitesse maximum à parvenir au point final. Ensuite, il optimise les vitesses des autres axes pour que tous de rencontrent au même endroit en même temps.

- Tous les axes partent en même temps avec une accélération et une vitesse maximum, pour s'arrêter chacun leur tour lors de l'arrivée sur le point final.

Lors de l'achat du robot, l'utilisateur devra :

- regarder le nombre de points susceptibles d'être mémorisés dans l'armoire de commande et prévoir une marge relativement importante de pas supplémentaires ;
- s'informer des possibilités d'extension mémoire pour profiter pleinement d'un des avantages du robot : la flexibilité.

Pourcentage des robots par type d'applications et de programmations dans les PME/PMI

	Apprentissage	Langage	CAO
Soudage à l'arc	91 %	9 %	
Projection	99 %		1 %
Assemblage	49 %	51 %	
Manutention	81 %	19 %	
Parachèvement	92 %	8 %	
Autre	91 %	9 %	
Total	82,3 %	17,6 %	0,1 %

Lorsque l'application concerne des tâches comme la projection, le soudage à l'arc ou le parachèvement, les trajectoires point à point restent impraticables et l'utilisateur devra passer à une trajectoire continue.

Dans ce cas, la mémoire du robot enregistre la trajectoire effectuée par le robot à intervalle de temps régulier et, par la suite, retranscrit ces données.

Il est possible de la part de l'utilisateur, s'il ne fait pas un apprentissage ou une programmation en direct, de fournir des instructions par rapport à des points rentrés en mémoire, comme l'interpolation linéaire entre deux points ou circulaire entre trois points ; il peut aussi utiliser des fonctions comme le TCL (Tool Center Point) qui permet de définir la position de l'outil dans un espace cartésien ; l'armoire du robot possède alors un transformateur de coordonnées qui détermine les consignes à envoyer aux différents axes pour que l'outil se trouve toujours dans la configuration désirée.

Pour programmer ces trajectoires, plusieurs méthodes sont employées : la programmation par apprentissage, la programmation par langage et la programmation par CAO, que nous analyserons successivement.

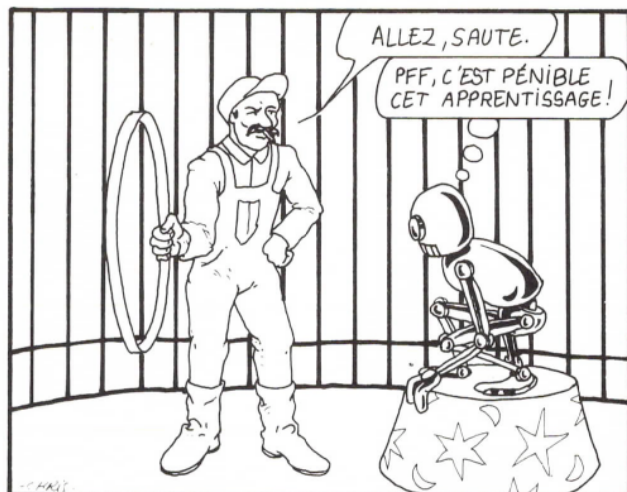
La programmation par apprentissage

Celle-ci correspond à une phase durant laquelle le robot est indisponible au travail industriel sur le site. Le temps passé doit donc être le plus court possible pour ne pas gêner la production de l'atelier.

Pour ce faire, deux méthodes d'apprentissage sont proposées : l'apprentissage direct et l'apprentissage indirect.

1) L'apprentissage direct

Dans ce cas, le robot est directement pris «en main» par le programmeur, ce dernier déplaçant le robot soit en point à point, soit en continu ; le choix dépend de l'application finale.

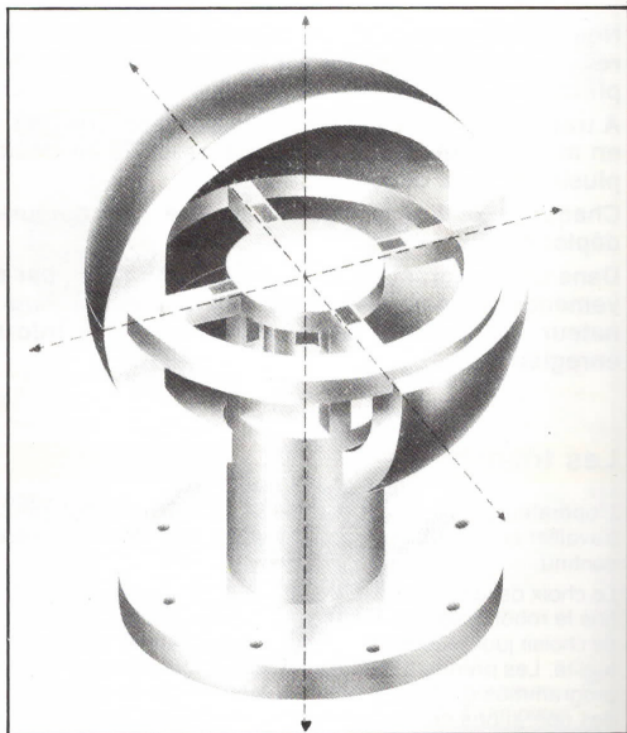


Pour de la projection, le mouvement effectué par l'opérateur est enregistré en continu puis retranscrit.

Dans le cas d'un petit robot de manipulation, l'opérateur amène le robot dans la position choisie et en actionne un interrupteur pour valider le point. Ce type de programmation doit bien sûr être compatible avec la mécanique du robot mais aussi avec les capacités physiques de l'opérateur. La trajectoire ainsi décrite reste peu précise et une amélioration par l'intermédiaire du pupitre est souhaitable, si l'armoire le permet (ce qui n'est pas toujours le cas).

2) L'apprentissage indirect

Ce procédé est plus fréquemment utilisé, d'une part parce que 98 % des robots peuvent être programmés de cette manière, d'autre part parce que les moyens pour parvenir au résultat final sont très diversifiés (télécommande, pantin, joystick...).



Langages commercialisés en France

Nom du Langage	Développé par	Date début d'étude	Disponible sur les robots	Robots commercialisés par	Langage niveau				Type du langage	Possibilité de vision	Possibilité de capteur d'effort
					Action-neurs	Effec-teur	Objet	Objectif			
AML	IBM	1977	IBM (Cartésien)	IBM		x			Pascal-PLA	non	oui
AML/E	IBM	1980	Sankyo (Scara)	IBM		x			Pascal-PLA	non	oui
IRL	Microbot	1978	Microbot	Electro-outil					CNC	non	non
LM	IMAG spécification	1979	Citroën (RM) Commercy Scemi Sormel Hitachi gdA AID	Citroën-Itmi Commercy-Itmi Scemi-Itmi Sormel-Itmi Itmi gdA-Itmi Robotronics-Itmi		x			Pascal	oui	oui
	ITMI suivi et évolution	1982									
LPR	RNUR DTAA	1978	Acma	Acma		x			Assembleur	oui	oui
RAIL	Automatix	1979	Hitachi Kuka Jungheinrich Automatix	Automatix		x			Pascal	oui	oui
ROL	Gixi	1979	Pas encore commercialisé sur robot			x			Basic	oui	non
SIGLA	Olivetti	1974	Sigma	Olivetti	x				CNC	non	oui
VAL	Vicarm Unimation	1974	Puma Unimate	Staübli		x			Basic	oui	non
VAL II	Unimation	1979	Puma Unimate	Staübli		x			Pascal	oui	oui
HELP	DEA	1980	Pragma	DEA		x			Pascal	oui	oui

Ces dernières années, la programmation par apprentissage a beaucoup évolué ; en apprentissage indirect sont apparus successivement les pupitres de commande solidaires de l'armoire, les boîtiers portatifs de plus en plus légers et fonctionnels avec ou sans manche à balai de manipulation ; actuellement certains boîtiers travaillent en dialogue conversationnel homme-machine, préfigurant la programmation par langage proprement dite.

Les possibilités de ces moyens de commande et leurs facilités d'emploi restent un critère de choix très important, étant donné le temps nécessaire pour programmer un robot.

Les plus perfectionnés permettent de passer, par l'actionnement d'un interrupteur, de l'apprentissage axe par axe à la commande directe ou à l'utilisation du transformateur de coordonnées pour commander directement les mouvements de l'extrémité de l'outil dans un espace cartésien ; mais aussi le pilotage des axes d'un positionneur de soudage, la programmation d'une fonction de palettisation à partir de la première position enregistrée, tout ceci à partir de la même boîte à boutons.

Pour des trajectoires plus complexes, en projection notamment, il existe deux sortes de procédés évitant la manipulation directe du robot. Toutes deux utilisant un bras d'apprentissage et en rendent, ou non, le robot esclave. Ce bras est

appelé, suivant le constructeur, pantin ou syntaxeur. C'est une structure identique au robot mais plus légère. Les trajectoires sont enregistrées à partir de ce bras, puis reproduites par le robot.

Le passage du bras d'apprentissage au robot n'est pas évident, car ce dernier n'est pas toujours en mesure d'effectuer des accélérations aussi rapides que celles qui ont été enregistrées.

Autre possibilité : le robot devient esclave le temps de la programmation. Le système est alors comparable à un fonctionnement de téléopération. L'opérateur positionne le robot à l'endroit voulu à l'aide du bras d'apprentissage. Le robot suit chacun des mouvements du bras en temps réel pour réaliser le même geste directement sur la pièce.

En fin de cycle, le robot devient alors maître et reproduit le cycle programmé. Si des améliorations doivent être apportées au programme, le bras pourra être à nouveau utilisé ; il faudra vérifier si l'armoire de commande du robot permet tous ces améliorations (insertion de sous-programmes dans le programme de base, répétition de sous-programmes, etc...).

Programmation par apprentissage

Pour illustrer ce mode de programmation, nous avons décidé de montrer les possibilités d'un des systèmes présents sur le marché. Chaque autre constructeur propose des variantes avec ou sans dialogue de plus ce type de programmation par apprentissage est en perpétuelle amélioration.

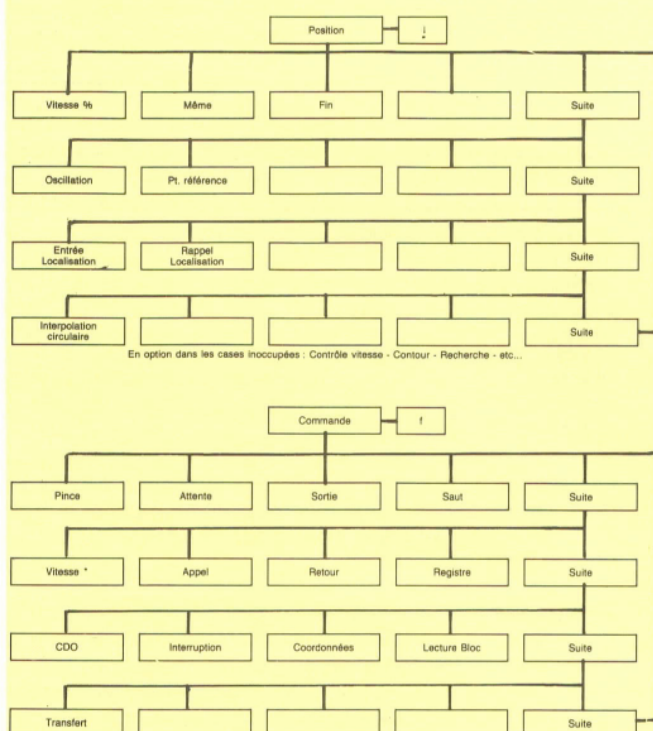
Le système que nous allons détailler est celui d'Aséa qui proposait il y a trois ans un nouveau type de programmation pour ses robots.

Les améliorations résidant principalement : premièrement en un combinateur à trois degrés de libertés permettant de positionner le corps et l'effecteur du robot en position et en orientation ; deuxièmement en un dialogue homme-machine avec lequel le programmeur trouve la tâche simplifiée.

Le dialogue

La procédure de programmation réside sur le principe du dialogue interactif avec l'opérateur. A cet effet, le pupitre de programmation est équipé d'un afficheur comprenant deux lignes de quarante caractères chacune.

Au-dessous et à gauche de la fenêtre d'affichage se trouvent cinq touches multifonctions qui permettent de simplifier le dialogue en sélectionnant les «chapitres» suivants :



Positions, Instructions de commande, Fonctions automatiques, Fonctions d'édition, Commande manuelle du système.

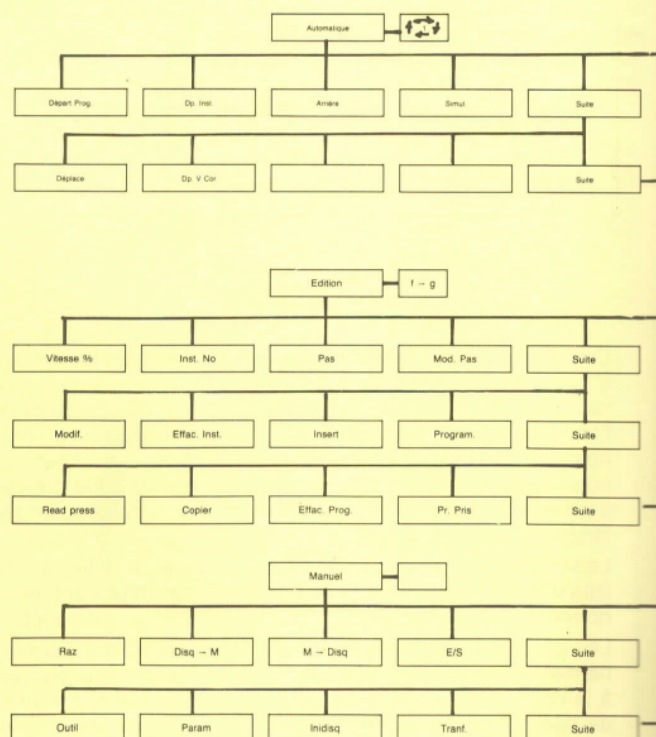
L'opérateur va donc choisir une de ces cinq fonctions et le système va lui demander des informations complémentaires jusqu'à ce que l'instruction complète désirée ait été pleinement définie.

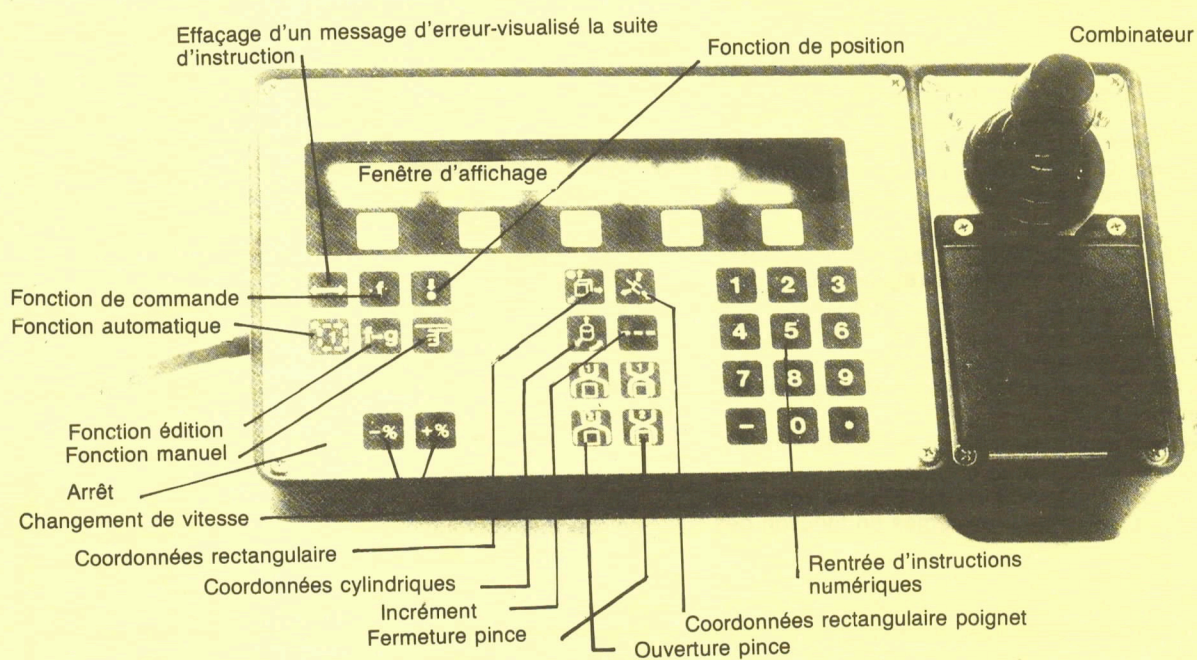
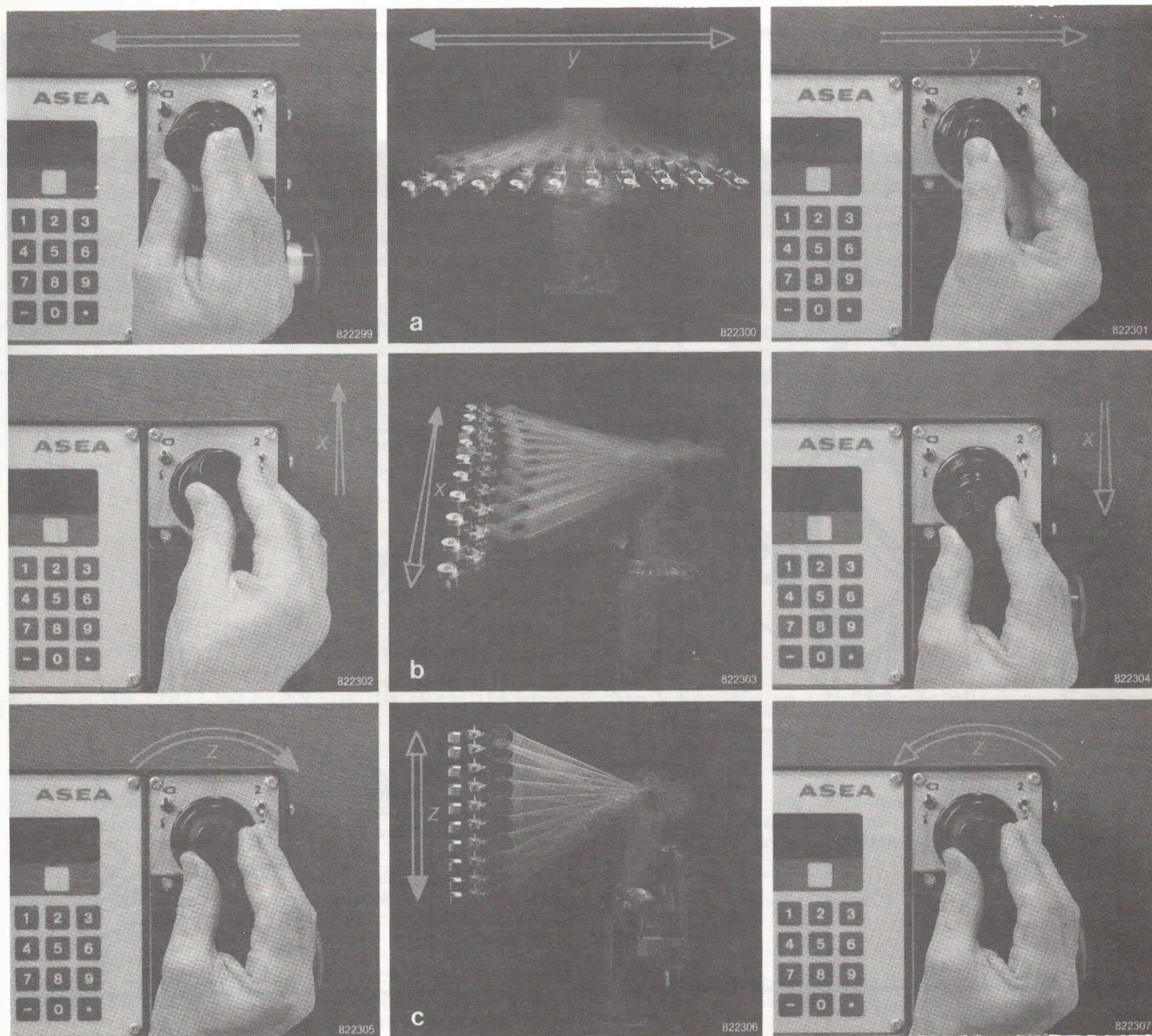
La première opération du programmeur va être d'amener le robot ou l'axe extérieur dans la position voulue, dans le système de coordonnées choisi, à l'aide du combinateur à trois degrés de liberté ; un commutateur régissant les mouvements du joystick en déplacement du CDO (Centre d'outil) d'une position à une autre, en rotation autour du CDO ou en mouvement d'axes extérieurs.

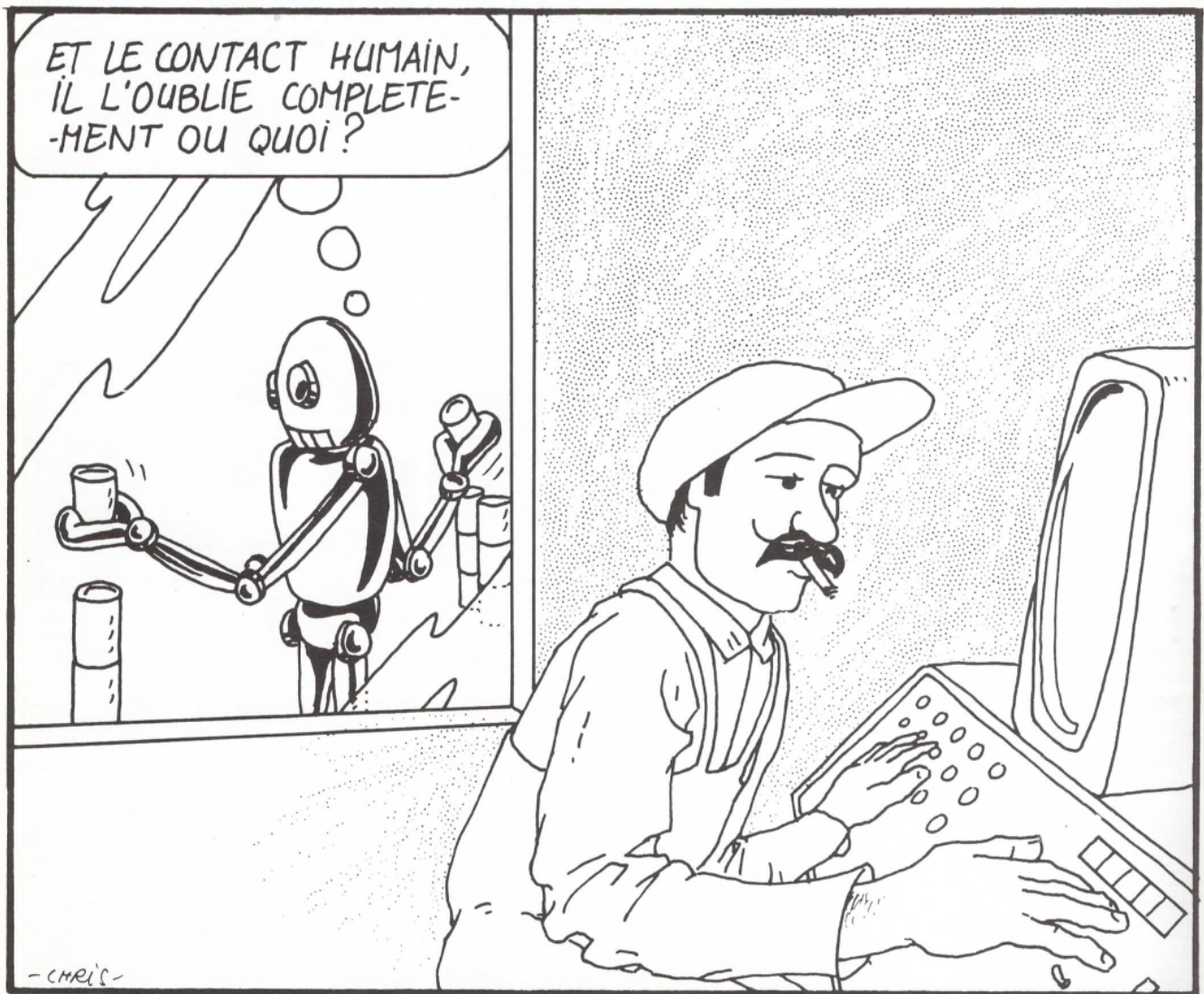
Chacune des positions est introduite dans le programme au moyen du boîtier de programmation, les instructions indépendantes de la position du robot sont introduites à tout moment dans le programme.

Un bloc de programme est composé d'un programme principal et d'un certain nombre de sous-programmes. Chaque pas comprend une instruction et un numéro d'instruction avec ou sans informations complémentaires.

L'exécution du programme est possible soit en continu soit par commande manuelle, instruction par instruction.







La programmation par langage

Une rapide parenthèse avant d'aborder ce deuxième type de programmation : il est flagrant de constater que les quelques robots fonctionnant actuellement en site industriel et programmés par langage le sont, pour la plupart, dans des industries de pointe (CIT, HP, IBM...) et que les industriels mécaniques ont du mal à accepter cette façon de programmer leurs robots.

Ces derniers préfèrent une méthode plus conventionnelle et plus en rapport avec leur compétence en mécanique ; par contre, la barrière d'un langage informatique n'existe pas pour les industries électroniques par exemple.

Le vieil adage de «la mécanique aux mécaniciens» et «l'informatique aux informaticiens» paraît encore bien ancré dans nos entreprises nationales.

Ce type de programmation par langage a été développé, il y a une vingtaine d'années, pour permettre de prendre en compte les notions d'intelligence artificielle et de permettre ainsi aux robots de «décider» eux-mêmes en fonction des objectifs à atteindre.

Cette ambition s'est concrétisée par un certain nombre de langages se situant à des niveaux de commande différents. Ceux-ci sont couramment subdivisés en quatre grandes classes. Les niveaux actionneurs, effecteurs, objets et objectifs.

a) le niveau actionneur

Celui-ci est inspiré des méthodes de programmation de machines à commande numériques ; l'opérateur pilote directement le robot par l'intermédiaire de ses actionneurs.

Ce moyen de programmation reste limité. Il est difficile de programmer un robot à coordonnées sphériques par ce moyen ; d'ailleurs, les constructeurs proposant ce niveau de programmation par langage sont des spécialistes de robots à coordonnées cartésiennes et parfois Scara.

b) le niveau effecteur.

Les premiers langages de ce deuxième niveau sont apparus au début des années 70. Ceux-ci permettent de décrire la tâche à exécuter non en termes de déplacement des actionneurs mais en termes de déplacement de l'outil du robot.

Comme précédemment, ils permettent la programmation de robots à coordonnées cartésiennes mais aussi sphériques en spécifiant uniquement les points de passage du robot et les orientations de ses différents axes.

Par ailleurs, il est possible d'intégrer, dans le cycle de programmation, des capteurs extérieurs ainsi que des outillages spécifiques qui vont travailler avec le robot.

c) le niveau objet

L'un des gros inconvénients des niveaux précédents est leur temps de programmation. Pour insérer, par exemple, deux pièces dans le cas du niveau effecteur, les différents points stratégiques de passage de l'outil doivent être décrits (soit une dizaine d'instructions en moyenne).

Au «niveau objet», l'opérateur doit définir les opérations à exécuter sur les objets, indépendamment des actionneurs et de l'effecteur. Ceci est d'un avantage évident, mais il reste un inconvénient : l'obligation de modéliser les pièces à manipuler ainsi que l'environnement. En effet, plus nous montons vers les niveaux supérieurs et plus l'unité de commande devra effectuer le chemin inverse en transformant les informations du niveau objet au niveau effecteur pour retraduire ces données en niveau actionneur.

C'est pour cette raison que, si les deux premiers stades de programmation par langage sont implantés sur site industriel, il n'y a pas, à l'heure actuelle, d'exemple de réalisation du niveau objet fonctionnant à 100 %.

La plus grande facilité du programmeur à travailler se transforme en difficulté pour le système de commande.

d) le niveau objectif

...Niveau à faire rêver bien des industriels mais qui malheureusement ne verra pas le jour rapidement ! En effet, le programmation d'une tâche en niveau objectif consiste essentiellement en la définition de l'objectif final à obtenir.

La commande du robot doit générer le cycle adéquat pour obtenir le résultat, donc réaliser d'elle-même le niveau objet pour redescendre ensuite jusqu'au niveau actionneur.

Comme nous venons de la voir, la programmation par langage est encore en phase de développement ; elle sera bientôt le passage obligé pour programmer des applications complexes avec un taux de flexibilité important, mais aussi pour prendre en compte plusieurs capteurs analogiques ou encore pour permettre la synchronisation de plusieurs robots avec interaction entre eux.

Système de CAO possédant un module robotique spécifique (commercialisé en France)

Nom du module	Nom du constructeur du module	Nom du revendeur	Année de mise en commercialisation	Robots modélisés	Etude d'implantation	Génération de programmes	Nombre de pro-modules vendus	Prix moyen des modules	
Place	Mc Donnel Douglas	Mc Donnel Douglas	1982	Asea, General Electric, GMF, IBM, Cybotech, Prab, Unimation, Yaskawa, GCA, Microbot, OTC, Adept, Manteg...	oui	non	/	350 KF	
Command	Mc Donnel Douglas	Mc Donnel Douglas	1982	Cincinnati-Milacron, Unimation, GCA, Cybotech. Possibilité pour l'utilisateur de développer son translateur	non	oui	/	100 KF	
Build	Mc Donnel Douglas	Mc Donnel Douglas	1982	Module pour les développeurs de robots qui permet d'associer la cinématique à la géométrie du robot.	/	/	/	150 KF	
Adjust	Mc Donnel Douglas	Mc Donnel Douglas	1982	Module pour calibrer la maquette virtuelle à la maquette réelle, en réinjectant les données nouvelles dans Place	/	/	/	100 KF	
Module robotique	Dassault Systèmes	IBM et Dassault Systèmes	1983	12 robots en bibliothèque, étudié pour que l'utilisateur modèle ses robots.	oui	oui	5 en F. 20 à l'étranger	loué à 5 000F/ mois	
Robographix Version 1	Computer-vision	Computer vision	1983	Unimation, Automatix, Cincinnati-Milacron, Asea.	oui	oui	8 en France	350 KF	
Robographix version 2	Computer-vision	Computer-vision	1986	Equipé d'un générateur de robots	oui	oui	/	/	
Grasp.	Byg Systems	Byg systems sur Apollo	1985	/	oui	oui	/	320 KF	

Il est à noter que beaucoup d'autres constructeurs proposent des modules de cinématique ou d'autres logiciels permettant de modéliser des robots, mais cette fonction reste pour eux souvent marginale ; aussi nous n'avons mentionné que les fournisseurs possédant un module spécifique.

La programmation par CAO

Étudiée en vue d'applications diverses, la CAO commence à apporter une solution supplémentaire en matière de programmation, mais la percée de ce nouveau moyen dépendra fortement de la pénétration des systèmes de CAO dans les entreprises.

De plus, outre la possession d'un poste de CAO, l'entreprise devra obligatoirement obtenir une définition géométrique du robot, mais aussi des pièces et de l'environnement atteignable par le robot.

Côté modélisation du robot, certains constructeurs de systèmes de CAO, en collaboration avec leurs homologues robots, proposent un éventail assez large de robots modélisés, facilitant considérablement la tâche de l'utilisateur, quoique certains robots posent de part leur architecture certains problèmes.

Pour les objets, cette modélisation indispensable est parfois déjà existante chez l'utilisateur qui a employé son système de CAO pour étudier ses pièces. Il restera quand même à rajouter certaines indications pour permettre le travail de programmation (position des points de soudure par exemple).

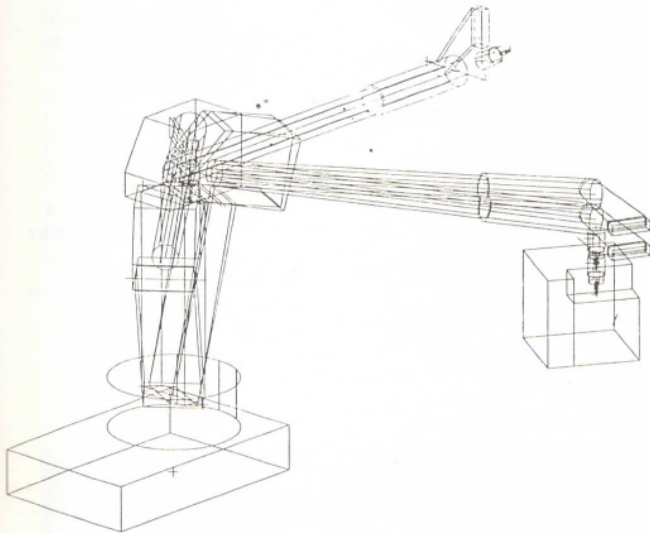


Fig.2 - Robot simulation

L'environnement restera à définir, celui-ci n'étant pas présent lors de la première installation.

Lorsque toutes les opérations de modélisations ont été mises au point, il reste à réaliser le programme par un dialogue homme-machine et faire les opérations de simulation, pour enfin aboutir à l'implantation industrielle.

Ce type de programmation est tout à fait bien approprié pour mettre en œuvre des programmes hors-ligne, évitant ainsi l'arrêt intempestif et souvent prolongé de robots. On peut choisir de l'utiliser pour des séries très courtes (robot de lavage de Meudon) ou des implantations importantes où la programmation peut être mise au point bien avant l'arrivée effective des robots sur le site ; le système de CAO aide aussi dans ce cas à l'implantation la plus judicieuse possible des robots (cas de lignes de soudage par points).

Conclusions

Il se dégage de ce dossier trois grandes classes de moyens pour programmer un robot : l'apprentissage, le langage et la CAO.

A chacune de ces possibilités, se rattache plus particulièrement des types d'applications. Nous avons essayé dans le tableau ci-après de répartir, suivant les applications, les moyens de programmation.

Restent plusieurs facteurs dont nous n'avons pas parlé et qui sont pourtant prépondérants, comme la compatibilité entre robots. Dans ce domaine, la robotique en est encore à ses tout débuts et il est encore impossible de programmer un robot d'un constructeur X à partir de l'armoire de chez Y, en programmation par apprentissage.

Par contre, les groupes qui ont mis au point les premiers langages de programmation spécifiques ont été plus soucieux de cet aspect du problème et ont la plupart du temps commercialisé un même langage à différents constructeurs

Certains de ces constructeurs proposent ainsi que le marché une unité de commande adaptable sur plusieurs mécaniques de robots, ce qui permet d'utiliser le même type de programmation pour un robot qui va effectuer de la manutention que pour des travaux de soudage à l'arc ou d'assemblage. Certaines de ces armoires vont même jusqu'à proposer un système de vision intégré.

Ce type de démarche est à encourager et serait particulièrement appréciable pour les PMI ayant l'utilité de plusieurs robots dans des secteurs divers et pas spécialement utilisatrices de systèmes de CAO. Dans ce sens, il serait très souhaitable que les constructeurs de robots acceptent de commercialiser des mécaniques nues, comme le font déjà, notamment, plusieurs entreprises japonaises.

G. Fages

Pour en savoir plus :

- A comparative study of robot languages - D.G. Shin - S. Bonner
- Le langage LM - E. Mazer - J.F. Miribel (Cepadues)
- Les robots : programmation, apprentissage et intelligence artificielle - G. Liégeois.
- Langage et méthode de programmation - M. Parent - C. Laureau (Hermès)
- Support de cours ITMI
- Conception et implantation d'un système de programmation de robots - Thèse de J.F. Miribel.
- Le guide de l'utilisateur 86 - G. Fages (E.T.A.)

Cinq face à la rédaction

«Les robots et leurs programmations»

Nos invités :

Pierre Montcuquet, responsable au sein de la société ITMI (Industrie et technologie de la machine intelligente) des produits robots-vision. L'objectif de cette société est d'associer à la robotique, la vision et l'intelligence artificielle. Ces trois domaines convergent au niveau de la recherche et de certaines applications industrielles. Créée en 1982, ITMI est une société de 50 personnes et commercialise divers produits dont notamment, le langage LM.

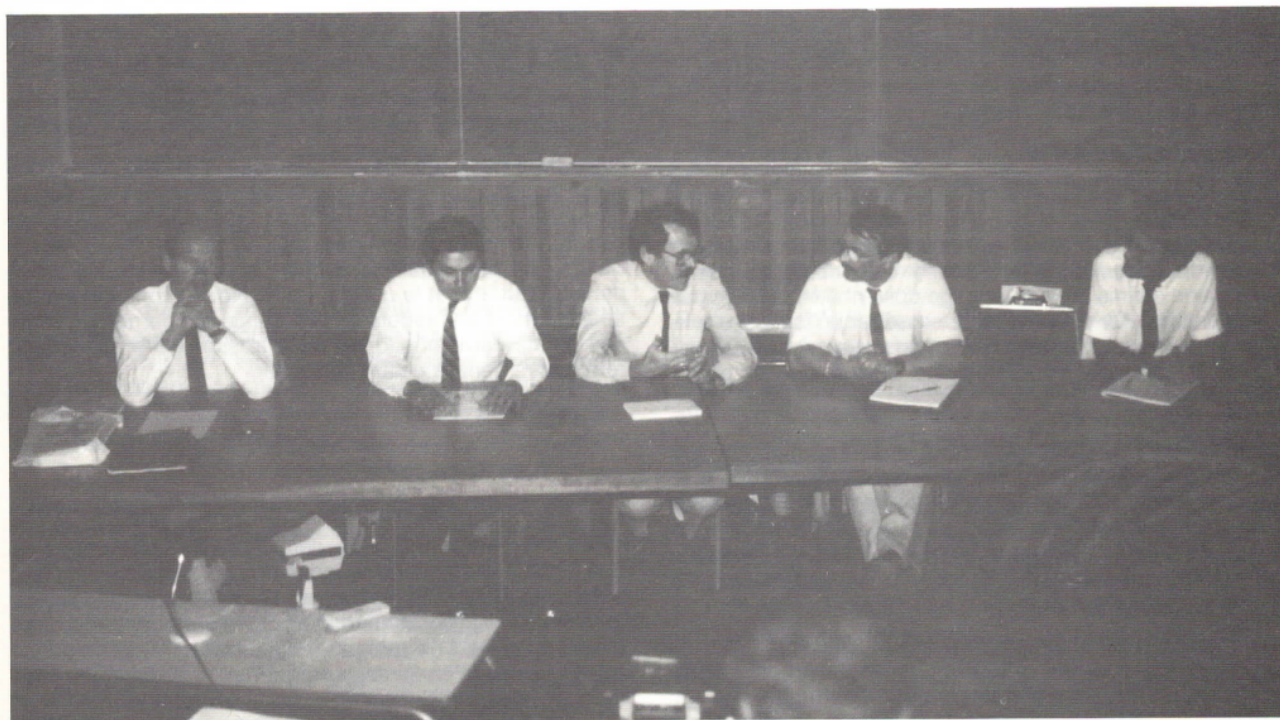
Guy Boulanger, responsable système et formation de la division robotique d'ASEA France. Asea AB est une société suédoise dont l'activité principale touche le domaine de l'électricité et du transport d'énergie. L'activité robotique au sein du groupe a démarré en 1970 par l'étude du premier robot tout électrique. La France commercialise la gamme de robots en les proposant dans divers domaines d'applications. Ces robots, du type point à point sont programmés par apprentissage en mode conversationnel à l'aide d'un pupitre, spécialisé portable.

Alain Riaud, responsable programmation à la société STAU-BLI, importatrice et distributrice des robots de la gamme Unimation. Activité principale de STAU-BLI : les programmeurs

de machines textiles et les raccords rapides. Sur un effectif total de 1000 personnes, l'activité robotique emploie 50 personnes. STAUBLI se présente comme distributeur et intégrateur de robots au sein d'applications et développe actuellement de nouveaux robots : les robots Unimate sont programmés par langage VAL I ou II.

Aldo Arban, responsable division système robotisé d'Elsag-DEA France (filiale du groupe italien Eltag). Celui-ci a pour objectif de réunir dans sa section automatisation industrielle toutes les composantes de l'usine du futur : CAO, automatisation des moyens d'usinage, contrôle automatique et robots d'assemblage. L'activité française pour sa division assemblage concerne la production et l'ingénierie de systèmes robotisés réalisés à partir des bras DEA-Pragma. Les robots sont programmés en semi-rigide et semi-apprentissage.

Bernard Cordobes, ingénieur commercial chez Mac Donnel Douglas, auprès de la division CFAO installée depuis peu en France, pour commercialiser d'une part le système de CFAO «Unigraphix» et d'autre part le système de robotique «Place» et aider ainsi les sociétés dans la simulation de cellules de fabrication et la programmation des robots.



Axes : Il existe trois grandes familles de programmation des robots : par apprentissage, par langage et par CAO. Nous aimerions savoir ce qui, pour chacun d'entre vous a déterminé votre choix.

Boulanger : L'installation d'un robot peut se faire dans trois domaines d'applications différents : le process, l'assemblage ou la manutention-service machine. A la conception des robots en 1970, ASEA a opté pour le premier créneau : le process (ébavurage, soudage...). La notion de suivi de trajectoire est dans ce cas très importante ; c'est donc à partir de cet objectif que le principe de programmation du robot a été défini. A l'aide d'un combinateur, l'opérateur déplacerait les axes le long de la trajectoire en programmant point par point. De plus, il fallait pouvoir programmer le robot sur son site, de manière à pouvoir contrôler à tout moment ses possibilités de trajectoire ; ceci correspond bien à une programmation par apprentissage.

Arban : Et vous servez-vous de ce même moyen pour l'assemblage ? Ce doit être très lourd à manipuler !

Boulanger : Dans les cas d'assemblage traités jusqu'alors, nous avons travaillé en programmation par apprentissage, ce type de programmation peut s'appliquer à toutes les trajectoires où le positionnement des pièces est parfaitement maîtrisé.

Arban : Je fais personnellement une autre approche qui consiste à dire : il existe deux extrêmes de programmation : - l'une très rigide comme celle utilisée pour les machines-outils : elle convient très bien pour les robots dont le nombre d'axes n'est pas trop élevé - inférieur à dix - avec des trajectoires pas trop complexes.

- à l'autre bout, se trouve des trajectoires très complexes (projection, soudage à l'arc...) dont le nombre d'axes en mouvement est assez variable. Si, dans ce cas, un robot est auto-suffisant pour une application, l'apprentissage convient très bien.

Entre ces deux extrêmes, se situe l'assemblage. Un système robotisé d'assemblage atteint un degré de complexité d'autant plus élevé que l'utilisateur lui demande une intelligence propre, conférant au robot la capacité de réagir à des situations «critiques».

Un système d'assemblage de plus de dix axes en mouvement doit posséder un logiciel susceptible d'assurer un pilotage asynchrone permettant à chaque système de la chaîne de réagir indépendamment les uns des autres sans interférence sur le reste de la machine.

La programmation par apprentissage et la programmation rigide ne permettent pas de répondre à ce cas de l'assemblage.

Nous proposons donc dans ce cas une programmation rigide pour la fonctionnalité de la machine. Cependant les points de l'espace définis dans le langage doivent être réalisés par apprentissage. Ce qui permet de posséder un logiciel d'une puissance énorme, tel un iceberg, la partie cachée restant la plus importante, l'opérateur ne voyant que la partie émergente.

Montcuquet : On constate que la programmation par apprentissage limite les possibilités d'expressions de l'opérateur, il existe une solution plus générale, le langage de programmation, car il offre la plus grande souplesse au niveau des systèmes de commande. C'est un véritable langage informatique évolué intégrant de plus des instructions spécifiques de robotique.

Notre langage par exemple, proche du Pascal, de part sa structure possède des primitives de déplacements et de modélisation de l'environnement géométrique.

- «En informatique l'Opérateur qui fait la saisie n'a pas écrit le programme - c'est la même chose pour les Robots»

Je pense qu'à partir de cette définition les langages doivent être la base de tout système de programmation de robots et permettre aussi bien l'apprentissage que la liaison à un système de CAO. En informatique, l'opérateur qui fait la saisie n'a généralement pas écrit le programme qu'il utilise. C'est la même chose pour les robots : à partir d'un système de programmation écrit en langage pour un type d'application, l'opérateur qui programme le robot joue le rôle de l'opérateur de saisie. A mon avis, ne posséder que l'apprentissage comme moyen de programmation limite très vite le champ d'action possible.

Boulanger : Pour notre part nous développons actuellement un système de programmation off-line de robots à partir d'un micro, et ceci sur le même principe : conception du programme à l'extérieur et apprentissage des points sur le lieu de travail, cet apprentissage peut-être remplacé par le calcul des points à l'aide d'un outil mathématique adapté.

Cordobes : Il existe une polémique entre la notion d'apprentissage et celle de programmation par langage. Nous avons, nous, abordé le problème différemment. Mac Donnel Douglas, constructeur d'avions, s'était confronté dès 1970 aux problèmes de programmation de cellules dans le projet Icam de l'Us Air Force. Il était nécessaire de développer un «outil» commun permettant de programmer toutes les commandes d'une cellule de fabrication. Ceci a conduit dans une première étape au développement d'un langage évolué (basé sur le langage APT).

Mais deux points délicats subsistaient : la restitution exacte de l'emplacement des différents composants, et le «debugging» qui consiste à essayer le programme sans risque pour la production.

«Place» que nous proposons actuellement sur le marché consiste en une simulation interactive sur écran, avec appel des différents composants de la cellule (robots, chariots...), positionnement de ces éléments en 3D et description des mouvements de chaque composant. Pour les robots, nous possédons une bibliothèque de près de 50 modèles standards du marché.

Riaud : Connaît-on avec exactitude cet environnement ?

Boulanger : En plus, il faut ramener aussi la précision de l'ordinateur à la précision de la mécanique des robots ?

Cordobes : La précision n'est plus à mettre en cause aujourd'hui, nous n'utilisons que des ordinateurs 32 bits qui garantissent les différents calculs. Quant à l'exactitude de l'environnement nous avons aussi une solution, car c'est vrai qu'il existe des différences entre la cellule virtuelle créée dans le système et la cellule réelle. Nous avons développé spécialement un module de calibration qui permet de saisir des points caractéristiques dans le site réel et de recalculer rétroactivement la cellule virtuelle.

- «Il existe deux types de situation, la CAO et la programmation assistée par opérateur»

Riaud : Il existe, à mon sens, deux types de situations : d'un côté la CAO intéressant les industries qui réalisent déjà leur fabrication et possèdent des modèles CAO de pièces ; de l'autre côté, la programmation assistée par opérateur (dans ce cas, les pièces ne sont pas toujours identifiées). Je pense indispensable le choix de programmation spécifiques pour chacun de ces utilisateurs.

Cordobes : Compte tenu du budget nécessaire pour l'acquisition d'un système CAO, il n'est pas possible d'en envisager une diffusion tous azimuts. Le marché est constitué pour nous d'une part des grandes sociétés qui ont investi dans la CAO, et d'autre part par les sociétés de services spécialisées en robotiques - d'ailleurs peu nombreuses actuellement -, qui font différentes études et analyses pour les PMI. Les constructeurs de robots peuvent également utiliser notre système pour certaines études.

Axes : Peut-on dire que certains types d'applications se rattachent plus particulièrement à un procédé de programmation ?

Boulanger : Absolument. Ainsi lorsque la trajectoire a son importance, on utilise préférentiellement une programmation par apprentissage.

Montcuquet : Il faut noter que les premières applications ont concerné des process choisis pour leur simplicité et pour lesquelles les systèmes de programmation par apprentissage étaient suffisants. Ces applications représentent encore le gros du marché, il y a une évolution vers des applications plus complexes pour lesquelles les limites de la programmation par apprentissage apparaissent (domaine de l'assemblage) (malgré le rajout de fonctions telles que les transferts de coordonnées, la vision, etc...). Pour ce type d'applications le langage de programmation offre davantage de possibilités. Nous proposons pour notre part, un contrôleur avec différentes mécaniques de robots et qui permet d'aborder ces nouvelles applications.

- «La fonction primordiale du robot, est-ce de faire de la production ou de l'informatique ?»

Boulanger : Quelle est, à votre avis, la fonction primordiale du robot : est-ce de faire de la production ou de l'informatique ?

Montcuquet : Le langage de programmation ne doit pas être accessible à l'opérateur dans l'atelier, mais au niveau du service méthode qui sélectionne ou crée le programme en fonction du type de travail à réaliser (assemblage, soudage...), ici s'arrête la partie «informatique». L'opérateur n'a plus qu'à réaliser une phase d'apprentissage réduite sur le site, d'où un gain important pour la production.

- «En Robotique, vous créez actuellement le métier de préparateur de programmes Robotiques»

Cordobes : En CN par exemple, nous en sommes actuellement à la suppression du travail de programmeur : à partir d'une simulation CAO, on fait télécharger le programme dans la machine. En robotique, vous créez actuellement ce métier : celui de préparateur de programmes robotiques, alors que j'arrive, moi, avec une solution qui vous permet dès aujourd'hui d'éviter cette étape.

Axes : Il semble que vous conceviez des robots pour les bureaux d'études. Or, c'est dans les ateliers qu'on les attend !...

Riaud : Nous possédons deux niveaux de langage, le VAL I (première génération) et le VAL II. Mais Unimation a aussi créé un niveau intermédiaire : le VAL +. Ceci semble indiquer qu'il faut orienter les langages vers l'application, plutôt que vers la sophistication de langages généraux.

Arban : Le robot ne doit pas être source de blocage. Même dans le cas de machine complexes, le langage doit être développé de façon simple et facilement abordable. Plus le logiciel est sophistiqué, moins l'utilisateur devrait avoir à pénétrer dans la fameuse partie immergée de l'iceberg.

Boulanger : Dans bien des cas, il faut simuler l'expérience de l'opérateur. Or, dans ce type d'applications, c'est l'apprentissage qui répond le mieux. Le problème pour une PMI se pose en termes de rentabilité financière de l'opération, et celle-ci passe par une compression maximum du temps de mise en place du robot lui-même.

Montcuquet : Actuellement, l'apprentissage couvre encore 80 % des applications, mais, dans un futur proche, l'aspect informatique et les langages vont devenir le dénominateur commun de tous les modes de programmation.

Axes : Si un utilisateur achète un système robotisé programmable par langage et veut, plus tard, modifier son application, doit-il forcément recontacter le constructeur ?

Montcuquet : Oui, à moins qu'il soit en mesure de le faire lui-même.

Arban : Si vous modifiez une partie de l'application, vous aurez non seulement la programmation à modifier mais aussi la périphérie.

Axes : Certaines applications montrent que le VAL semble plus facilement assimilable par l'utilisateur que le LM par exemple. Comment l'expliquez-vous ?

Montcuquet : Je pense que le LM est difficile à programmer au niveau utilisateur il n'a pas été conçu pour que ce dernier programme son application en LM. C'est un outil de base et non pas le langage direct avec le programmeur.

L'erreur vient de certains constructeurs qui achètent le langage LM nu, sans utilitaires particuliers. C'est le constructeur qui doit fournir ou nous demander ces utilitaires. Le système sera d'autant plus simple qu'il proposera une bibliothèque de logiciels spécifiques.

- «Plus une machine devient intelligente et performante, plus le dialogue homme-machine devient logiquement simple».

Arban : C'est la machine elle-même qui doit devenir un robot grâce au logiciel. Pendant le cycle, le système analyse pourquoi une pièce ne s'assemble pas et pendant ce temps les autres robots continuent à travailler et la productivité n'en est pas altérée.

Plus une machine devient intelligente et performante, plus le dialogue homme-machine devient logiquement simple.

Boulanger : Ce que nous disons chez ASEA, c'est que nous avons un langage tout à fait transparent pour l'utilisateur qui programme le robot à l'aide d'un langage conversationnel, sans savoir ce qu'il y a «derrière».

Cordobes : La CAO répond bien à ces problèmes, car le module post-processeur transforme le langage neutre de notre machine en langage spécifique du robot (LM-VAL...). Avec la CAO, le dilemme sur le langage le plus simple n'aura pas d'intérêt. De plus, si vous avez dix robots différents avec dix langages différents, seule la CAO pourra répondre.

Riaud : Tenez-vous compte des problèmes locaux de vision par exemple ?

Cordobes : Nous avons encore des difficultés pour résoudre ces cas précis, mais nos laboratoires étudient sérieusement cet aspect.

Axes : Mais ces «utilitaires» dont vous parlez, existent-ils ?

Montcuquet : Nous avons développé un certain nombre d'apprentissages généraux qui se trouvent sur notre contrôleur, et d'autres utilitaires seront présentés dès 86.

Riaud : Les nôtres sont en cours d'étude et seront bientôt proposés.

Axes : Y aura-t-il, à l'avenir, compatibilité des langages ?

Montcuquet : La plupart des langages actuels sont liés à des constructeurs de robots. Notre attitude est de proposer notre langage à un maximum de sociétés : c'est le seul moyen pour qu'il soit largement adopté et qu'ainsi il évolue.

Arban : Il faut noter que le langage est souvent associé à l'architecture du système commercialisé par le constructeur.

Axes : Quels progrès peut-on attendre au niveau des langages ?

Montcuquet : Les besoins en flexibilité des robots de demain nécessiteront l'intégration progressive d'intelligence artificielle par exemple pour la programmation automatique... Cette évolution prévisible rejoint le concept de la Machine Intelligente qui est à l'origine de la création de notre société.

Boulanger : De toute façon, la progression sera vraisemblablement la même qu'en informatique : la programmation par mots machine, puis par assembleur, puis par langage évolué. Reste à savoir si en robotique toutes les tâches pourront être programmées au niveau «objectif».

Axes : Quel niveau de formation est exigé pour programmer un robot ?

Boulanger : Nous proposons une formation à la programmation d'une durée d'une semaine, le travail comprend 1/3 de théorie sur le principe de programmation et 2/3 d'utilisation pratique du robot à l'aide d'exemples concrets. Après quoi l'utilisateur, sans formation initiale, repart en sachant programmer le robot.

Arban : Nous avons, pour notre part, deux niveaux de formation, le premier en une demi-journée sur site pour apprendre à l'opérateur à manœuvrer la machine. Le second niveau permet à l'utilisateur de pénétrer dans la partie immergée de l'iceberg.

Stage de 15 jours. Niveau de base : BTS.

Riaud : Pour le VAL I, l'utilisateur apprend en une semaine de formation à intervenir au niveau application ; pour le VAL II, trois semaines sont prévues.

Montcuquet : Nos cours durent une semaine et requièrent un niveau de base de BTS ou ingénieur. ☐

Ce «Cinq face» a été réalisé en public pendant les journées de présentation de matériels scientifiques du Cast à Lyon. Le prochain débat public aura lieu le mercredi 3 décembre dans le cadre du salon «Educatec-Formation» sur le thème de la formation en robotique.

Le début de ce débat organisé par Axes Robotique aura lieu à 16 heures précises. En préambule, l'Afri organise des conférences sur le thème de «La robotique-des métiers pour demain» qui auront lieu le matin et en début d'après-midi. Si vous êtes intéressé par les conférences contactez l'Afri (1) 547.69.33 ; par le débat contactez Axes Robotique 88 23 28 81.

Venez nombreux poser vos questions.