

De A à Z:

Un robot d'assemblage et son langage

Un robot français né l'année dernière a acquis, en très peu de temps, une réputation enviable: c'est le robot d'assemblage de la Scemi.

Les concepteurs s'étaient fixé des objectifs ambitieux: le robot devait être capable de déplacer rapidement une masse d'un kilo, avec une précision permettant l'assemblage de pièces mécaniques; le volume balayé devait avoir un rayon d'action de 60 cm.

L'examen détaillé du robot – mais aussi la description du langage qu'il utilise – montre de quelle manière la Scemi a atteint ses objectifs.

La morphologie

Le premier choix concernait la morphologie du robot: orthogonale, cylindrique ou polaire? Les deux premières solutions présentant des caractéristiques limitées rendant les assemblages difficiles – voire impossibles – dans certains plans, il restait la morphologie polaire, qui seule permet d'atteindre tous les points dans l'espace, quelle que soit l'orientation.

Les six degrés de liberté utilisés sont répartis en deux groupes: les trois premiers axes constituent le bras, les trois autres le poignet. Pour augmenter les performances dynamiques, les deux bras principaux sont dotés d'un système d'équilibrage du couple statique dû à la pesanteur; ainsi le robot peut-il exécuter des déplacements à une vitesse pouvant aller, en bout de pince, jusqu'à 2.7 m/seconde.

Lorsque les points de programme sont définis, le calculateur détermine la position de chacun des six axes et leur coefficient de vitesse du mouvement correspondant, transmet ces informations à un microprocesseur qui donne les consignes de vitesse. Ainsi est définie la boucle de vitesse, réalisée par un variateur et une dynamo tachymétrique.

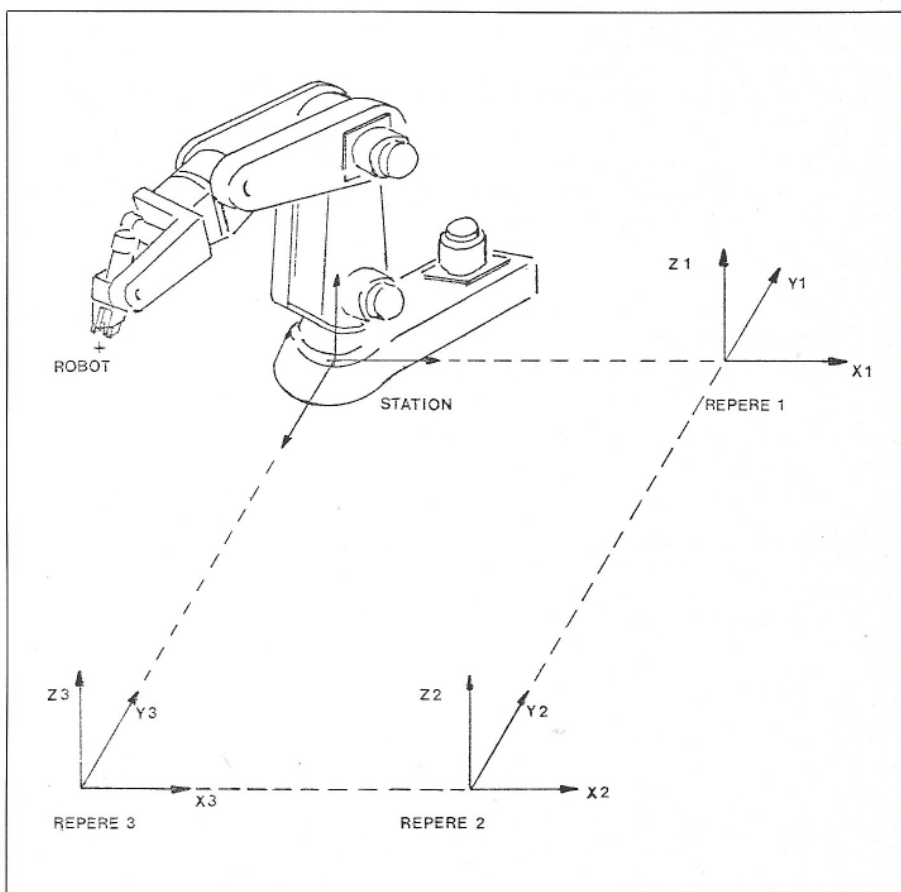
Calcul et contrôle

Le rôle du microprocesseur ne se limite pas au calcul de la consigne de vitesse. Il assure plusieurs autres fonctions:

- surveillance des situations anormales, par exemple arrivée en butée de l'un des axes;

- contrôle des compteurs en position (saturation, dépassement)
- détection d'erreurs de «traînée» pouvant provenir, par exemple, de la rencontre d'un obstacle au cours d'un mouvement du bras.

Le microprocesseur peut, en outre, interrompre le programme de l'unité centrale, soit pour demander de nouvelles consignes, soit pour signaler un mauvais fonctionnement.



Un Robot de A à Z (suite)

Implantation

Le système de commande est contenu dans une armoire qui renferme le calculateur, l'ensemble mémoire à deux disquettes, les variateurs de vitesse et la partie alimentation secteur, relayage et protections.

Pour le dialogue utilisateur-machine, on retrouve un terminal de visualisation avec écran et clavier et une imprimante. Un boîtier de commande relié à l'armoire permet de déplacer manuellement le robot, en actionnant un manche à balai, et de lancer des programmes de travail.

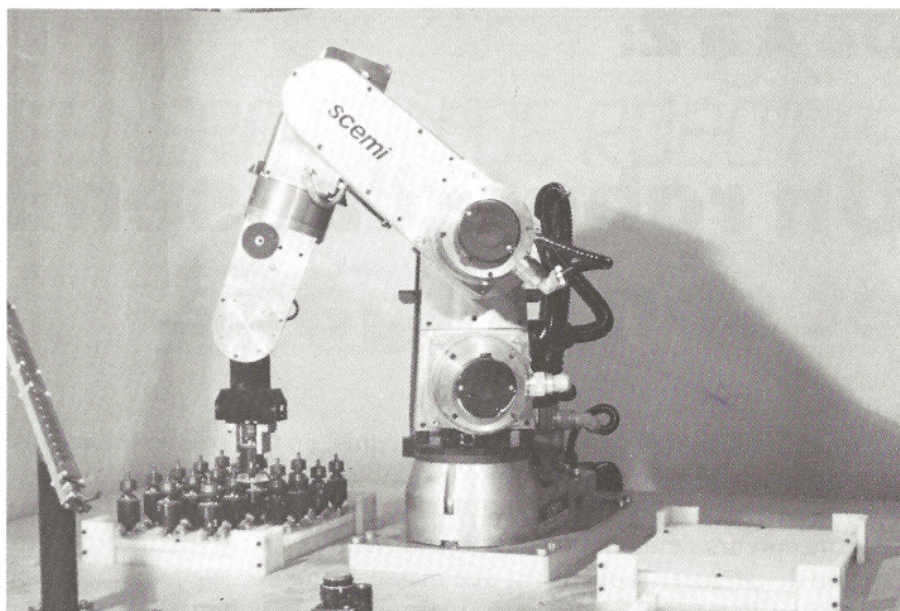
Le calculateur travaille en 16 bits avec 64 K octets de mémoire extensible. Deux disquettes assurent la mémoire auxiliaire, une pour le logiciel d'exploitation, l'autre pour l'étude des programmes d'application. Lorsque l'étude est terminée elle ne supporte plus que le programme d'application. Il suffit, pour changer le programme, de remplacer la disquette «utilisation»: l'opération est quasi-immédiate.

Des interfaces analogiques et digitales prévoient des liaisons avec les capteurs extérieurs afin de commander des organes périphériques du robot.

La programmation: apprentissage ou langage

Deux possibilités pour la programmation: par apprentissage, ou programmation symbolique par langage.

La programmation par apprentissage consiste à exécuter manuellement les différents mouvements du robot avec le manche à balai, en enregistrant les points de destination successifs, les vitesses, les temporisations et toutes les conditions des mouvements. Cela présente des inconvénients, comme la difficulté d'utiliser des capteurs évolués, ou la nécessité de procéder à la reprogrammation intégrale si une partie seulement du programme ne convient pas. Ce temps d'enregistrement par apprentissage est souvent long et, de plus, il immobilise le robot pendant la durée de la programmation.



Le robot en phase de démonstration

La programmation symbolique par langage permet d'exploiter plus efficacement l'ensemble robot, capteurs et ordinateurs. Un langage de programmation a été mis au point, étudié spécialement pour ce robot à partir du «Pascal» par la Scemi, en collaboration avec l'Institut de Mathématiques Appliquées de Grenoble.

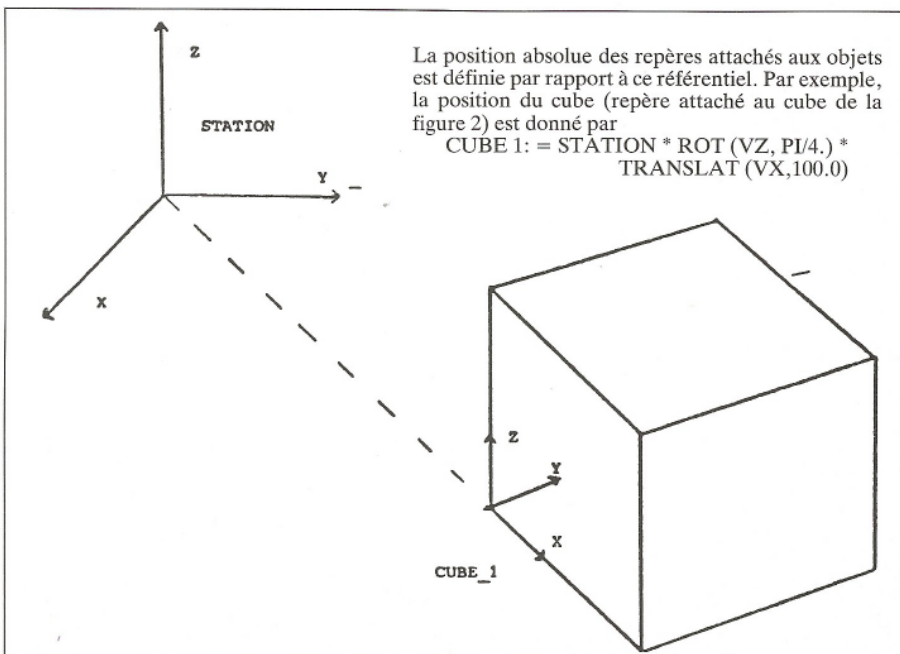
Un langage haute performance

Dans ce langage de haut niveau, une bibliothèque de sous programmes ou de fichier autorise un gain de temps important dans la construction des programmes. Le cycle à exécuter est défini comme une suite d'opérations élémentaires, présentées sous forme d'algorithmes tels que Prendre-Poser-Insérer.

Chacune de ces opérations peut faire l'objet d'une procédure. Il suffit ensuite de combiner les différentes procédures pour réaliser l'opération d'assemblage. Exemple: Déplacer goujon à position 3 jusqu'à FZ 0.5.

Autre avantage du langage LM: il incite le programmeur à structurer ses programmes et à définir explicitement les types de données qu'il veut manipuler. Les risques d'erreurs sont ainsi réduits et, par conséquent, la sûreté d'utilisation est plus grande.

L'espace physique dans lequel évolue le robot est «modélisé» à l'aide de repères cartésiens. On peut ainsi spécifier en X, Y, Z un point à atteindre ou une trajectoire passant par des points intermédiaires. Les repères sont également utilisés pour définir la position et l'orientation des objets et de l'outil terminal du robot.



Langage «effecteur»

Bien entendu, le langage prend en charge les calculs nécessaires pour effectuer les déplacements demandés et pour maintenir à jour position et orientation des objets.

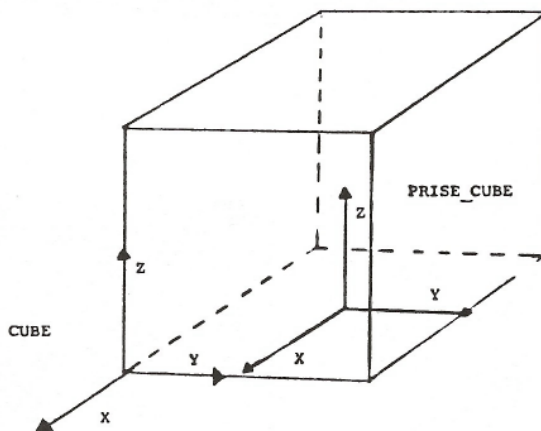
Le langage LM est donc un langage de niveau «effecteur», ce qui signifie que le programmeur peut raisonner directement en termes de déplacement de l'outil terminal du robot ou des objets qu'il transporte.

Quatre types de déplacements sont généralement nécessaires dans une opération d'assemblage:

- les déplacements à grande vitesse pour amener le robot sur sa zone de travail;
- les déplacements d'objets dans un univers encombré;
- les déplacements soumis à des conditions d'arrêt;
- les déplacements asservis à des données de capteurs.

Tous ces déplacements sont exprimables directement, à l'aide d'instructions spécifiques prévues dans le langage LM.

L'utilisateur dispose ainsi, grâce à ce langage, des outils qui lui permettent de programmer lui-même les réactions de son matériel aux réactions des capteurs. En utilisant des capteurs évolués, il pourra réaliser des opérations d'une précision supérieure à celle du robot lui-même et pallier les dispersions dues aux matériels et aux pièces à assembler.



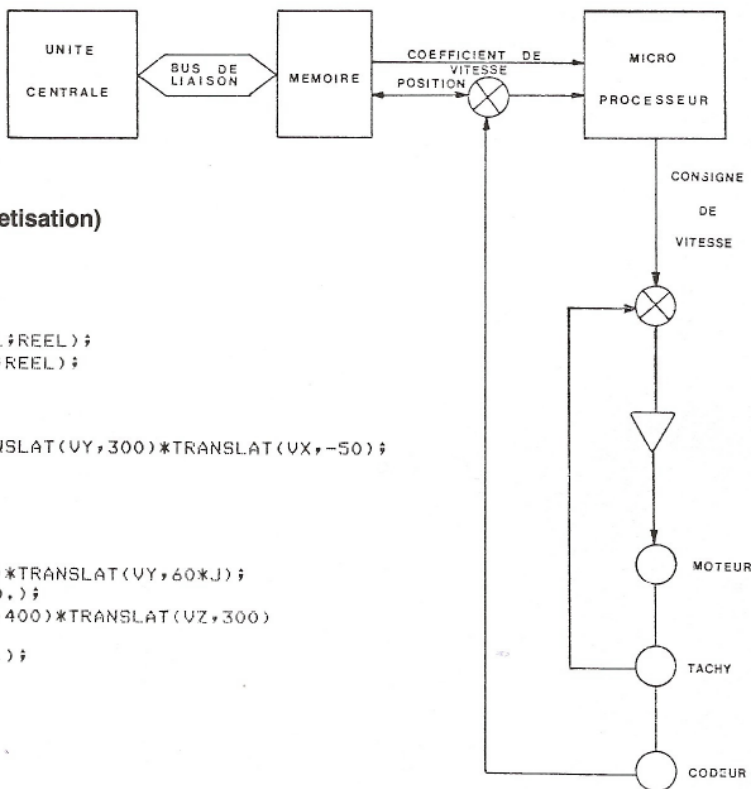
Plusieurs repères peuvent être attachés à un même objet, ce qui permet de décrire cet objet avec toutes les positions caractéristiques qui lui sont associées. On peut par exemple attacher un repère de prise PRISE_CUBE au repère CUBE.

Des perspectives

Le langage LM autorise enfin l'exécution de plusieurs tâches en parallèle. Ainsi peut-on, à la fois, commander un déplacement, surveiller une condition d'arrêt tout en continuant l'exécution du programme. Il devient possible de synchroniser le robot avec des équipements périphériques et d'adapter son comportement à des données fournies en temps réel.

Une autre possibilité offerte par le langage LM est de distinguer le déplacement robot proprement dit, et les actions de ses outils. Le robot peut donc travailler à partir d'un équipement périphérique très simple – bol vibrant, visseuse... et gagner ainsi en flexibilité. On conçoit que des recherches soient déjà entreprises pour l'équipement des ateliers flexibles.

SCHEMA SIMPLIFIE DE LA COMMANDE D'AXE



Exemple de programme de déplacements: (Palletisation)

```

PROGRAMME P5;
  REPERE RA, POS;
  ENTIER I, J;
  EXT PROCEDURE TEMPO(REEL);
  EXT PROCEDURE PRENDRE(REPERE; REEL; REEL);
  EXT PROCEDURE DEPOSE(REPERE; REEL; REEL);
  REEL GLOBAL V, TEM;

DEBUT
  V:=1; FIXER VITESSE ROBOT A V;
  RA:=STATION*TRANSLAT(VZ, 100)*TRANSLAT(VY, 300)*TRANSLAT(VX, -50);

CYCLE
  I:=0;
  TANTQUE I<=3 FAIRE
    J:=0;
    TANTQUE J<=2 FAIRE
      POS:=RA*TRANSLAT(VX, 50*I)*TRANSLAT(VY, 60*J);
      TEM:=2; PRENDRE(POS, 30., 30.);
      POS:=STATION*TRANSLAT(VX, 400)*TRANSLAT(VZ, 300);
      *ROT(VZ, PI);
      TEM:=2; DEPOSE(POS, 30., 30.);
      J:=J+1;
    FINFAIRE;
    I:=I+1;
  FINFAIRE;
  INCYCLE;
FIN;
  
```

Nota: Distances en mm
Angles en radians