Système de pilotage (CNC-2)

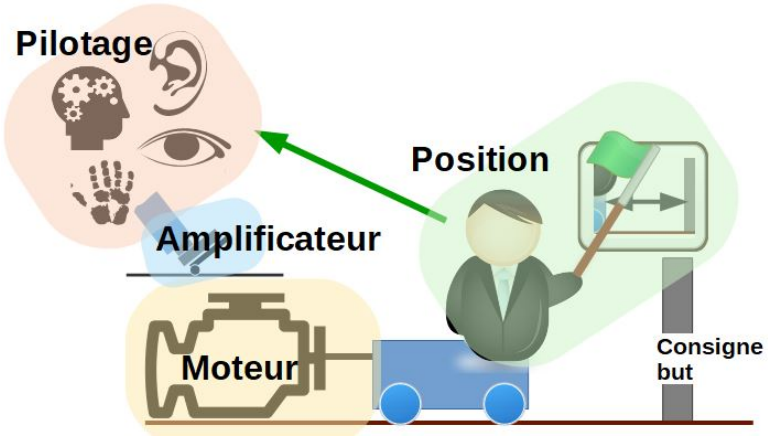
On parle système de pilotage dans ce deuxième article qui fait partie d'un ensemble d'article décrivant les commandes de machines-outil, les CNC.

1. *Régulation de mouvement en machine-outil.*
2. *Présentation de trois systèmes de pilotage (un ancien, un astucieux et un open source).*
3. *Le parcours d'outil, le générateur de trajectoire*
4. *Les correcteurs et les ajustements.*

Chaque système me permettra d'introduire des notions intéressantes en pilotage de position et par extension en commande numérique.
L'ancien système : la base de temps, la traînée
Le système astucieux : le temps réel, la simultanéité
Le système open source : l'automate, le directeur de commande

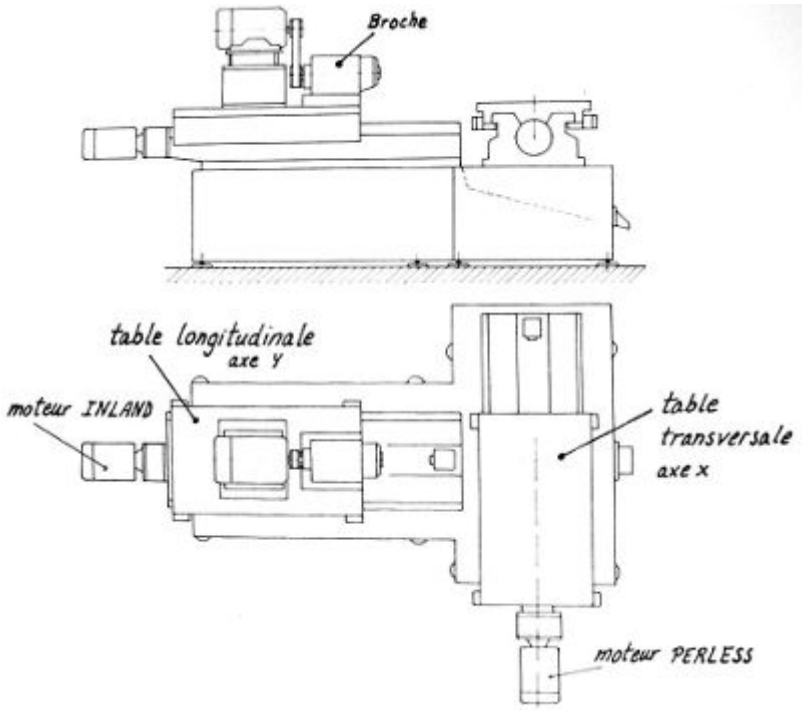
Rappel

Rapide retour sur l'article précédent, surtout pour ceux qui arrivent ici sans l'avoir parcouru. On parle de régulation de position, le sujet est de contrôler le déplacement d'un chariot sans que la position finale (le but, la consigne) ne soit dépassé. Pour ce faire, on dispose d'un système qui a, au minimum, un retour d'information sur la position réelle du chariot. Nous avons séparé, un peu arbitrairement, les divers éléments de ce dispositif de contrôle comme suit :

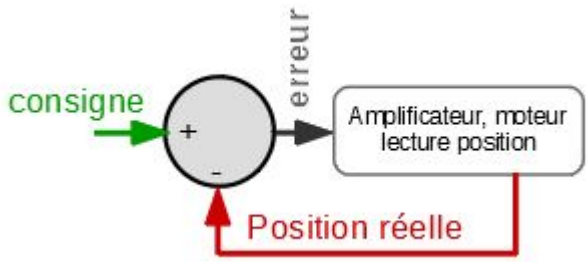


Un ancien pilotage

Ce système de pilotage datant des années fin 1970, ça ne nous rajeunis pas, a été réalisé pour une machine rudimentaire composée de deux axes orthogonaux (x,y) et d'une broche. Je vous parlerais uniquement de la partie pilotage, car je n'ai plus aucun document concernant les autres parties. Ce pilotage était réalisé par un automate programmable (un TSX-80 de télémécanique pour les connaisseurs), vous avez quelques explications sur les automates dans la partie "pilote open source". Des moteurs puissants à courant continu pour les axes et une broche de 12KW présente un ensemble mécanique assez imposant malgré l'absence d'axe Z. Lors des déplacements à pleine vitesse des coulisses (6m/min) ou lors d'arrêts sécurité, c'était relativement spectaculaire.

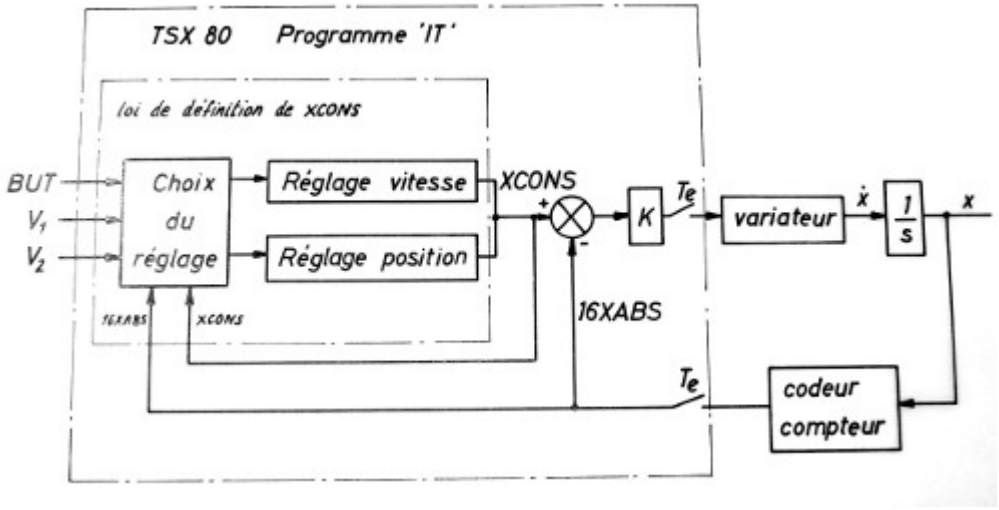


Mais ce qui nous intéresse ici, c'est le principe du système de pilotage et les fonctions de programmation des mouvements plus que la machine elle-même.



Pour rappel, une base de temps est nécessaire pour pouvoir calculer la position à atteindre et ce surtout pour vraiment piloter (contrôler) les axes. Il est possible d'envoyer directement l'erreur de position à l'amplificateur (différence entre la position actuelle et celle à atteindre), mais dans ce cas, le système moteur/amplificateur devra se débrouiller tout seul pour arriver à déplacer les coulisses le mieux possible. On comprend que si pendant la plus grande partie du déplacement l'amplificateur est saturé (ce qui correspond a une très grande erreur), la régulation correspondra : "à fond les manettes", ce qui n'est pas à proprement parlé un "pilotage". Si l'on veut effectivement contrôler le mouvement des axes (accélération, vitesse) le tout dans un mouvement fluide et sans problème, cette méthode n'est pas la plus efficace.

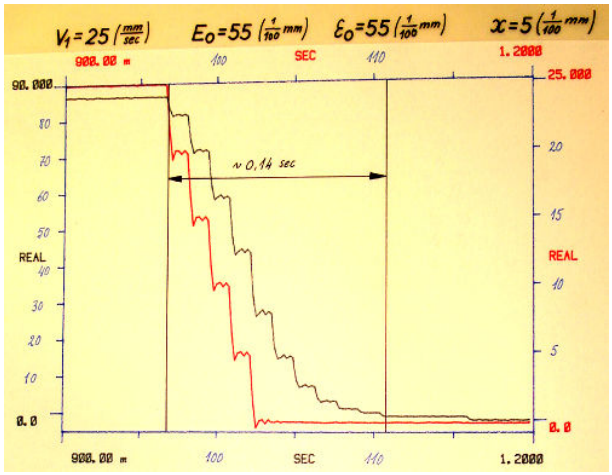
Donc une autre voie est généralement utilisée, on essaye de déterminer une position intermédiaire qui peut être atteinte en gardant un contrôle total des éléments de la chaîne de régulation. C'est en calculant cette (ces) position(s) intermédiaire(s) que le pilotage (le calcul) est assez utile en fin de compte ! D'abord le schéma de principe général de cette régulation, à gauche les consignes et à droite la sortie, la position :

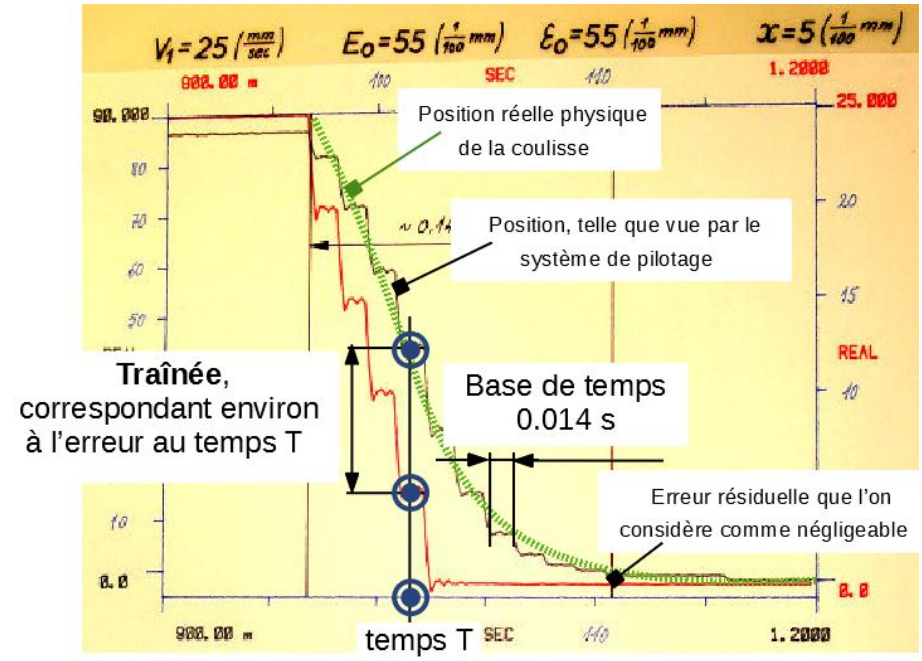


Vous trouvez les éléments principaux en entrée (à gauche) la consigne (la position à atteindre, le BUT) et deux vitesses. Celle du mouvement (V1) ainsi que la vitesse à la consigne (V2). À la sortie du pilote, la boîte, le programme du TSX80, vous trouvez le signal pilotant le moteur via son variateur (amplificateur). L'élément qui peut surprendre dans cet ancien pilote est la double case "réglage vitesse" et "réglage position". Ce schéma est un peu trompeur car en fin compte c'est toujours une régulation de position dont il s'agit.

C'est un problème uniquement d'adéquation technique entre les performances demandées (vitesse déplacement maximum) et les capacités de l'automate utilisé le TSX80. Bien sûr, actuellement ce type de schéma n'aurait pas lieu d'être, mais à l'époque … Quelques chiffres pour étayer ces explications : Vitesse maximum des axes : 100mm/s (environ 6m/min) avec un temps de base de 0.014s (carte temps réelle adjointe à l'automate, pulse en interruption sur l'automate) soit un calcul de position possible chaque 1,4mm ce qui un peu juste pour un contrôle en position lors du déplacement rapide. Les temps d'accélération ou temps d'arrêt tournaient autour 0.2s, ce qui n'était pas mal. Nous avons donc séparé le mouvement en 3 parties, au départ régulation en position, puis régulation en vitesse et pour finir une régulation de position. Le changement de type de régulation dépendait de la traînée (voir ci-dessous).

Ce choix de la méthode de régulation était uniquement de l'apanage du pilote, donc automatique est en fait transparent pour l'utilisateur. Voici la mesure d'un freinage (0.14s) sur une position de consigne depuis la vitesse de 25mm/s, c'est un relevé qui visualisait les valeurs dans l'automate, le TSX80. Vous remarquerez les escaliers de la courbe rouge (consigne) et la courbe noire (la position "réelle").





Un des éléments à relever c'est le décalage entre la position réelle (ligne noire) et la position calculée (ligne rouge), on appelle cette valeur la traînée et je vous rappelle que cette valeur est nécessaire sinon il n'y a pas d'erreur, donc pas de correction à envoyer à l'amplificateur, donc pas de mouvement. C'est parce qu'il y a une différence entre la position réelle et celle calculée qu'un signal est envoyé au moteur.

Pour compléter ce système de pilotage, une partie du programme de l'automate permettait

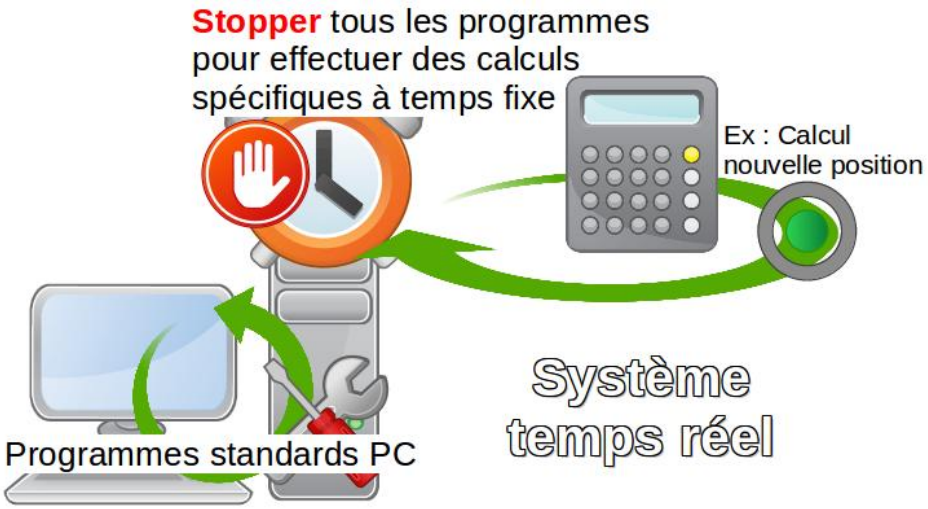
de piloter la machine en mode manuel, semi-manuel ou automatique. Avec ce dernier mode, via un langage propre (comme un codeG pour les spécialistes), on pouvait programmer, tester et exécuter tout de sorte d'enchaînement de mouvements, avec des déclenchements d'opérations particulières. C'était une programmation en style "boîtes fonctionnelles" métier "mécanicien-usineur", Grafcet pour les connaisseurs, le TSX80 utilisant déjà ce principe.

Bien sûr l'interface homme-machine, n'avait rien à voir avec ce que l'on trouve actuellement, on se contentait d'une ligne numérique (peut-être alphanumérique) si mes souvenirs sont bons.

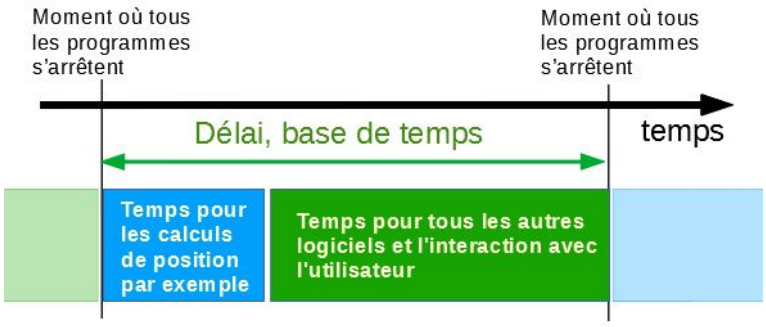
Un pilotage astucieux

Dans les années 1985-1995, un fabricant de machine de soufflage de PET (la fabrication des bouteilles d'eau, par exemple) rencontra des difficultés avec les automates du marché, car ils étaient trop lents pour couvrir le pilotage de ces machines multi-axes. Une équipe a eu l'idée d'utiliser des PC avec Windows95/98 comme système de pilotage d'axes de machines. Le problème est que le système de Windows n'est pas prévu pour cela, bien que présentant d'innombrables avantages au niveau de l'interface, du prix, de la disponibilité et l'interconnexion, cela particulièrement par rapport aux automates de l'époque.

Les quelques spécialistes qui lisent ceci ont déjà les sourcils froncés car Windows et temps réel ce n'est pas trop compatible. Alors avant de poursuivre, posons-nous la question : qu'est-ce "temps réel" dans un système informatique ?

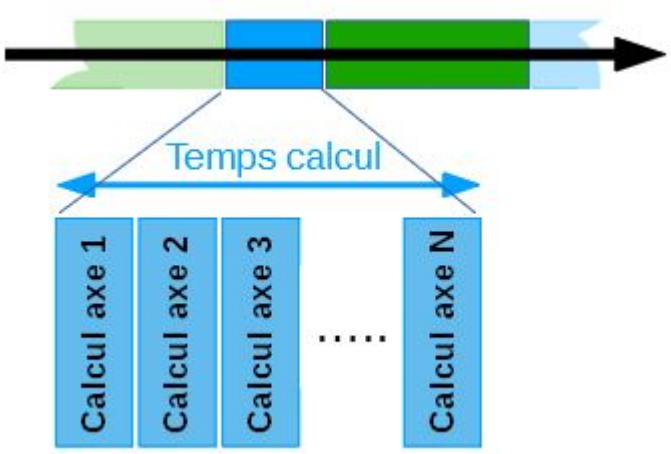


Pour qu'un système de pilotage soit temps réel, il est nécessaire que le programme respecte une horloge et que non seulement les opérations soit effectuées mais qu'elles le soient en un moment précis. Je vous laisse revoir l'article sur la régulation de position et aussi la partie sur l'ancien pilotage sur l'utilité d'un délai fixe. Lorsque l'on parle de temps réel, on sous-entend également la base de temps (laps de temps entre deux opérations devant répondre à un délai impératif, un délai donc), généralement il y a plusieurs bases de temps, dans le sens que l'on doit calculer plusieurs choses à intervalles réguliers, mais différents (on y reviendra avec le système open source).



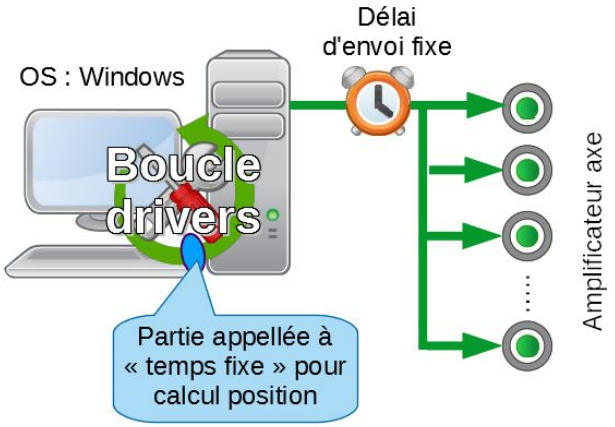
Par contre, dans son utilisation normale le système d'exploitation Windows n'a pas besoin de respecter des délais impératifs, il doit permettre d'effectuer toutes les opérations mais dans un temps relatif, par exemple si l'affichage d'une information à un peu de retard, ce n'est pas primordial pour l'utilisateur. Notez que cela est relatif, une seconde c'est beaucoup de temps pour certain et c'est rapide pour d'autre. Cela doit aller vite, mais il n'y a pas d'impératif temporel. Schématisation d'un temps réel, comment se découpent les

tâches à effectuer.



Donc il fallut rendre Windows "temps réel", comment faire cela ? C'est là toute l'astuce de ce système de pilotage, la simultanéité. Si l'on prend le temps de la zone bleue, qui correspond aux calculs des positions de consigne, que nous pouvons comprendre ?

Nous l'avons vu après un calcul de position d'axe, on envoie l'erreur à l'amplificateur du moteur de l'axe. Vous sentez immédiatement le problème si votre machine à beaucoup d'axes, entre le calcul et l'envoi de la nouvelle consigne entre l'axe 1 et le dernier axe calculé, un grand délai est possible. Ce qui est ennuyeux pour des machines ou le synchronisme des axes peut avoir une forte incidence sur les résultats espérés. Si on procède à un envoi groupé, ou plus exactement la prise en compte, de toutes les nouvelles positions, nous aurons une simultanéité de la régulation des axes. C'est la voie utilisée. Naturellement le calcul des positions doit être lui aussi relativement à délai fixe et presque simultané. Ce qui compte c'est que la lecture des positions réelles ainsi que la prise en compte des nouvelles consignes soient, pour elle-même, le plus possible simultané et à horaire fixe.



On “déplace” le temps réel à la prise en compte des nouvelles positions ce qui permet d’avoir une “certaine souplesse” sur le moment du calcul. Je ne vous explique pas les précautions à prendre à chaque ajout de drivers, le temps de latence (temps de réponse, attente entre la demande d’une action et son déclenchement) doivent être impérativement contrôlés. Évitez que les utilisateurs installent subrepticement le dernier joystick !

Ce type de commande numérique est encore en vente, sous toutes réserves, voir le site FastWare.

Un pilotage open source

Le type de système de pilotage que propose LinuxCNC est “standard” à beaucoup de système du marché, comme Fanuc, Fagor, GE, Num, Siemens, etc. et bien sûr les systèmes basés sur PC (Personnal Computer) plus ou moins industriel.

Dans le cas présent on utilise une base matérielle (PC standard, renforcé ou industriel) sur lequel on installe un noyau Linux (système d’exploitation) ainsi que des logiciels qui permettent d’obtenir un PC doté de caractéristiques temps réel. Il est utilisable jusqu’à 8 axes simultanés peut-être 9.

Un système de pilotage est généralement constitué de divers éléments et pour contrôler une machine-outil on utilise généralement une partie avec automate et une partie temps réel,

mais comme on va le voir, ce type de partage des tâches n'est pas aussi clair.

"HISTORIQUE" DES CNC

Les premiers outils permettant à l'homme de produire les objets furent leurs mains, des outils manuels, des outils motorisés, des machines-outils, des CN (numerical control) , des CNC (computer numerical control), des centres d'usinages, des cellules de production, etc. Une chose intéressante à remarquer, c'est éloignement progressif de l'humain du lieu de production, du vrai travail. Pour ce qui nous concerne, l'évolution des dénominations de machines-outil à CNC est assez intéressantes et met en avant le passage d'une commande logique simple au calcul sophistiqué. Les premiers pilotes/contrôleurs des machines-outils, ce fut l'humain, puis les relais (armoire à relais), les automates programmables (API, automate programmable industriel) et puis les ordinateurs temps réels.

Qu'est-ce qu'un automate programmable

C'est généralement un système séquentiel, une boucle infinie exécutée après une partie d'initialisation. Elle lit les entrées, et exécute séquentiellement une logique qui mets à jour les sorties (certains automates, ces mises à niveau des sorties sont réalisées en différé). En fait son programme s'exécute un peu comme un système à relais. Pour réaliser cela, il propose divers modules comme un watchdog (sécurité), des cartes d'entrées/sorties spécialisées, des modules de communications (bus de terrain), des temporisateurs et d'autres éléments spécialisés "métier".

Pourquoi l'utilise-t-on dans une machine moderne ?

Un automate est un système relativement sûr (matériel et logiciel renforcés pour une ambiance industrielle souvent perturbée). Pour un industriel outre sa robustesse, la pérennité des solutions est un facteur important, pensez que pour un système la durée approvisionnement doit être encore de 10 ans. Une maintenance simplifiée est aussi un point important ainsi que l'accès à des modules, bibliothèques "métier" éprouvées.

Ordinateurs temps réels.

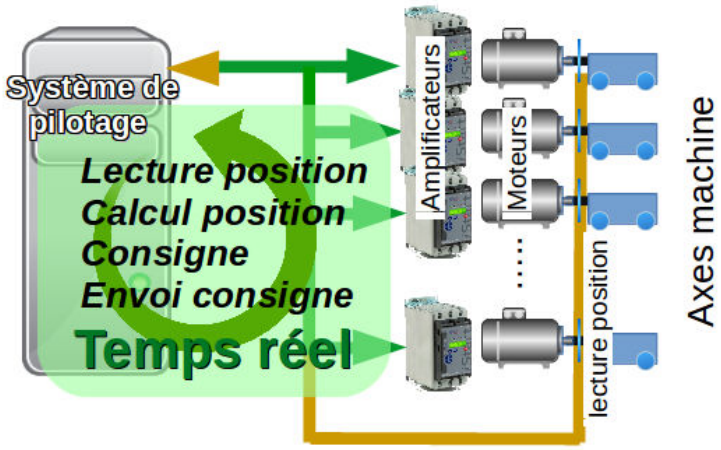
Depuis leur apparition, les ordinateurs ont tout de suite intéressé les industriels bien qu'au départ leur utilisation en milieu industriel n'était pas simple (perturbation, ambiance et rapide évolution). Notons cependant que certains produits comme le processeur Z80 a une durée de vie remarquable (>40ans), car il est encore et toujours fabriqué et utilisé. Mais au-delà de quelques exceptions pour reste c'est plus dur (matériel de stockage, carte spécifique, norme, etc.). Les avantages de ces computers sont leurs disponibilités, leurs possibilités de communication externe, l'interface homme/machine, leur puissance de calcul et leur prix relativement bas.

COMMANDE NUMÉRIQUE

On a vu donc une convergence s'opérer entre ces deux mondes. Les systèmes basés sur les automates se sont adjoint des PC extérieurs au début, puis de plus en plus incorporé, et de l'autre côté des ordinateurs avec des automates collés, puis inclus sont apparus. Actuellement il est difficile de qualifier une CNC d'un automate avec un PC ou un PC avec un automate. Il y a des solutions entièrement automates ou l'on ne parle pas de PC mais avec des interfaces hommes/machine spécifiques ainsi que des PC avec des automates logiciels "noyés" dans leur système.

LINUXCNC

C'est la solution du PC avec un automate logiciel. La partie temps réel est donc composée d'un automate, d'une interface homme/machine et d'un directeur de commande. Une remarque avant de poursuivre, l'interface homme/machine n'a pas fondamentalement besoin d'un temps réel, mais pour simplifier certains cas particuliers ainsi qu'alléger la programmation, la maintenance, l'affichage a une base de temps propre différente de celle du générateur de trajectoire. Pour lier un système PC (la carte mère) aux amplificateurs des moteurs d'axes ainsi qu'à la lecture des positions, il est nécessaire que le système de dialogue soit aussi temps réel. Ceci explique pourquoi LinuxCNC ne peut pas directement piloter des axes machines via un bus tel que USB, car ce bus n'a aucune garantie de temps réel. Généralement linuxCNC utilise le bus parallèle ou un système dédié (>12 drivers) comme les cartes électroniques de Mesa Electronics.



Le directeur de commande

Le but de la CNC est de pouvoir depuis un parcours virtuel réaliser une forme réelle selon la technique de fabrication utilisée. Ce qui signifie que des phénomènes dynamiques et cinématiques apparaissent aux niveaux des axes machines et de l'interaction de la fabrication. La forme même du trajet demandé peut amener la machine à ralentir ou accélérer et présenter un effet perturbateur des mouvements.

Pour fixer les déplacements de la machine, généralement un fichier définissant le trajet (souvent appelé parcours d'outils) est élaboré hors machine par un système FAO. Sans entrer dans le détail ce fichier indique la série de mouvements que la machine doit exécuter.

Au niveau de la CNC, on retrouve bien la problématique de la régulation, un axe avec une demande de déplacement et une vitesse à respecter. Le directeur de commande va définir chaque point intermédiaire selon les possibilités de chaque axe de la machine et prenant en compte le synchronisme des axes et les caractéristiques de la machine. Si une coulisse travaille à vide ou est surchargée avec un grand poids (outil ou pièce), les informations et

les consignes ne pourront pas être identiques pour un même parcours. Le rôle de ce directeur de commande est d'interpréter au mieux les demandes fixées par le fichier provenant de la FAO. Donc vous sentez bien que ce parcours ne sera pas exactement ce qui est demandé en fin compte. Il y a beaucoup de travaux en cours dans ce domaine, car ce principe n'est pas optimum bien que presque toutes les machines actuelles fonctionnent comme cela, indépendamment du type de CNC.

REMARQUE

J'ai utilisé LinuxCNC (anciennement EMC2) pour quelques machines numériques, essentiellement des machines-outil de type fraiseuses. Vous trouverez un article sur une de mes applications ici.

Conclusion

Le système de pilotage en régulation de position peut présenter divers aspects et surtout son principe risque de varier. Mais en fin compte, c'est presque toujours l'erreur entre la consigne et la position "réelle" qui est le premier paramètre, ce qui normal, car une régulation de position c'est assurer une position !

Suite : Le parcours d'outil

Précédent : La régulation de position