(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) N° de publication : **2 569 877**
(à n'utiliser que pour les commandes de reproduction)

(21) N° d'enregistrement national : **85 10286**

(51) Int Cl⁴ : G 05 B 19/18.

# (12) DEMANDE DE BREVET D'INVENTION A1

(22) Date de dépôt : 5 juillet 1985.

(30) Priorité : US, 31 août 1984, n° 646.355.

(43) Date de la mise à disposition du public de la demande : BOPI « Brevets » n° 10 du 7 mars 1986.

(60) Références à d'autres documents nationaux apparentés :

(71) Demandeur(s) : *GENERAL ELECTRIC COMPANY.* — US.

(72) Inventeur(s) : Joseph Edward DeKlotz, Thomas Edward Manuel et David Arthur Palmer.

(73) Titulaire(s) :

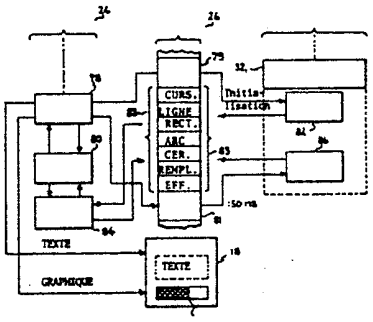
(74) Mandataire(s) : M. Dugas, GETSCO.

## (54) Procédé de visualisation graphique pour un système de commande numérique de machine-outil.

(57) L'invention concerne les techniques de commande numérique des machines-outils.

Un système de commande numérique pour une machine-outil comprend notamment des sous-programmes 82, 86 qui sont exécutés lorsque l'opérateur demande la visualisation d'une page d'information d'état de machine. L'un des sous-programmes 82 initialise une visualisation graphique prédéterminée 88 à partir de fichiers de commande de visualisation 80. Le second sous-programme 86 est ensuite exécuté périodiquement pour effectuer une mise à jour dynamique du graphique visualisé.

Application à la visualisation des conditions de fonctionnement des machines-outils à commande numérique.




FR 2 569 877 - A1



Vente des fascicules à l'IMPRIMERIE NATIONALE, 27, rue de la Convention — 75732 PARIS CEDEX 15

La présente invention concerne de façon générale des systèmes de commande numérique par ordinateur pour des machines-outils et elle porte plus particulièrement sur la visualisation graphique d'information liée au fonctionnement de tels outils et au processus d'usinage.

Les systèmes de commande numérique destinés à commander des machines-outils conformément à l'exécution de programmes enregistrés dans un ordinateur sont bien connus. De tels systèmes commandent l'opération d'usinage sous l'action d'un ou de plusieurs programmes de pièces qui sont enregistrés dans une mémoire après avoir été chargés dans la mémoire au moyen, par exemple, d'une bande perforée. A la demande, le programme de pièce fait accomplir à la machine une série d'étapes de fabrication. Lorsque, par exemple, la machine comprend un outil de coupe, le mouvement relatif entre la pièce usinée et l'outil de coupe s'effectue normalement selon trois axes mutuellement perpendiculaires, conformément au programme de pièce qui impose le mouvement de l'outil de coupe. De plus, un système de commande numérique par ordinateur peut être conçu de façon à commander d'autres fonctions telles que des changements d'outil, la vitesse de la broche, la circulation du fluide de refroidissement, la sélection de palettes et la génération de messages et de graphiques. Le nombre et le type de fonctions varient considérablement en fonction du type de la machine-outil qui est commandée. Dans le passé, on a effectué la commande au moyen de circuits câblés spécialisés fonctionnant

sous la dépendance de conditions détectées sur la machine-outil et d'ordres contenus dans le programme de pièce.

Cependant, plus récemment, on a incorporé dans un système informatique la logique associée à des systèmes de commande de machine-outil, au moyen d'un ensemble de programmes enregistrés (c'est-à-dire d'un logiciel), pour réaliser la commande d'ensemble. Le logiciel est spécialement adapté à une application particulière et on peut le modifier aisément à la demande pour satisfaire des besoins spécifiques. L'adjonction d'une commande programmable à un système de commande numérique procure un certain nombre d'avantages dont le premier et le plus important consiste en ce qu'il suffit de reprogrammer la commande pour lui permettre de fonctionner avec une machine-outil particulière. De plus, la programmation par l'utilisateur est relativement plus aisée, à cause du type de jeu d'instructions qui est normalement employé et des caractéristiques d'édition qui sont normalement disponibles.

Une commande numérique par ordinateur, ou simplement une CNO, destinée à commander le fonctionnement d'une machine-outil, peut comprendre deux logiciels séparés, à savoir la "logique de commande numérique", qu'on appelle le logiciel de commande numérique, dont le contenu est plus ou moins fixe, et la "logique de commande de machine", qu'on appelle le programme LCM, qui est modifiable et peut être adapté à une machine-outil particulière. Dans l'industrie, cette adaptation est habituellement effectuée par le constructeur d'équipements spécifiques (OEM). Le logiciel de commande numérique commande la fabrication des pièces, le mouvement selon les axes, la visualisation sur un tube cathodique et d'autres fonctions auxiliaires, tandis que le programme LCM commande essentiellement, sinon exclusivement, le fonctionnement de la machine-outil. La demande de brevet des E.U.A. n° 415 041 déposée le 7 septembre 1982 décrit en outre un

logiciel d'interface, qu'on appelle une Fenêtre d'Adaptation, qui est constitué par un groupe de programmes, de réseaux d'entrée/sortie et d'indicateurs d'état qui assurent la liaison entre le logiciel de commande numérique, le programme LCM et la machine-outil. L'utilisation d'une fenêtre d'adaptation procure au constructeur d'équipements spécifiques, ou au client, une grande souplesse dans l'adaptation d'une commande numérique par ordinateur à une machine-outil particulière. Dans l'un de ses aspects, l'invention utilise une fenêtre d'adaptation.

Une fois que les commandes numériques par ordinateur ont été programmées et fonctionnent, elles remplissent leur fonction automatiquement et pratiquement sans intervention du personnel d'exploitation. Il est néanmoins nécessaire de temps en temps que l'opérateur (ou d'autres) puisse discerner rapidement des résultats de fonctionnement, des données d'usinage, des conditions de la machine, etc. En bref, il est nécessaire qu'il existe une communication aisée entre la commande numérique par ordinateur et le personnel qui s'occupe du processus commandé. Dans une commande numérique par ordinateur de type caractéristique, une partie considérable de cette communication s'effectue par une présentation visuelle sur un tube cathodique. La majeure partie de l'information présentée est cependant sous la forme d'un texte qui doit être lu intégralement pour être pleinement compris. Il demeure donc un besoin portant sur une aide graphique, de façon que le personnel d'exploitation puisse connaître très rapidement et très aisément des conditions de fonctionnement telles que la vitesse de la broche, ou le temps d'utilisation cumulé d'un outil, en jetant seulement un regard sur l'écran du tube cathodique.

L'invention a donc pour but d'apporter un per-

fectionnement à des systèmes de commande numérique pour des machines-outils.

L'invention a plus particulièrement pour but d'apporter un perfectionnement à des moyens destinés à présenter une information sur un dispositif de visualisation graphique pour un système de commande numérique par ordinateur, de façon que des opérateurs de machine-outil, du personnel de maintenance ou autre puissent discerner aisément et rapidement les conditions de fonctionnement de la machine-outil.

Brièvement, on atteint les buts précédents ainsi que d'autres, dans un système de commande numérique par ordinateur, par l'utilisation d'un nouvel ensemble de sous-programmes incorporés dans la partie du programme de logique de commande de machine (LCM) qui est spécialement accessible pour l'adaptation de la commande numérique par ordinateur à une machine-outil particulière. Les sous-programmes sont exécutés lorsque l'opérateur de la machine-outil demande la visualisation d'une page de visualisation d'information d'état de la machine. Les sous-programmes comprennent une routine destinée à initialiser une visualisation graphique prédéterminée à partir de fichiers de commande de visualisation du logiciel de commande numérique. Ceci est suivi par une seconde routine qui est exécutée périodiquement lorsque la visualisation graphique est active, pour mettre à jour de façon dynamique l'information graphique qui est visualisée. Pendant chacune des deux routines, on appelle une ou plusieurs de six procédures de "tracé" pour produire la visualisation graphique désirée. Ces procédures comprennent le déplacement d'un curseur graphique jusqu'à une position donnée sur l'écran du tube cathodique, le tracé d'une ligne, le tracé d'un rectangle, le tracé d'un arc, le tracé d'un cercle, le remplissage d'une zone prédéterminée et l'effacement d'une zone, qui peut comprendre

la totalité de l'écran de visualisation.

En combinant ces procédures, on peut créer une représentation graphique de grandeurs physiques sur l'écran du tube cathodique. Ces graphiques sont d'une forme que l'observateur peut comprendre rapidement. On peut dans certains cas leur donner une forme semblable à celle d'appareils de mesure analogiques. A titre d'exemple, en combinant deux arcs contigus, on peut créer une visualisation analogique de la vitesse d'une broche, en pourcentage de la vitesse maximale. Dans un tel cas, on amène le curseur à une position spécifique, on trace ensuite un premier arc dont la longueur représente la vitesse réelle de la broche, et on trace ensuite un second arc, contigu au premier. Les deux arcs représentent ensemble la vitesse maximale de la broche. Lorsque la vitesse de la broche change, les angles des arcs varient mutuellement. L'impression obtenue est assez semblable à celle correspondant à un tachymètre à affichage analogique. On notera qu'on peut créer des graphiques à segments, des diagrammes à secteurs et diverses autres formes de graphiques.

La suite de la description se réfère aux dessins annexés qui représentent respectivement :

Figure 1 : un schéma simplifié représentatif d'un système de commande numérique par ordinateur connu, de l'art antérieur, destiné à la commande d'une machine-outil ;

Figure 2 : un schéma représentant de façon générale le logiciel contenu dans le système de commande numérique par ordinateur représenté sur la figure 1 ;

Figure 3 : un schéma synoptique représentant de façon plus détaillée le logiciel de commande numérique par ordinateur représenté sur la figure 2 ;

Figure 4 : un schéma représentant la manière selon laquelle on établit le programme de commande de

machine (LCM) ;

Figure 5 : un schéma synoptique représentant de façon générale la séquence d'opérations du logiciel de commande numérique par ordinateur ;

Figure 6 : un schéma représentatif des diverses pages de visualisation qui peuvent être présentées sur l'écran d'un tube cathodique qui fait partie du poste de commande numérique représenté sur la figure 1 ;

Figure 7 : un schéma synoptique représentant des sections de sous-programmes qu'on appelle des "modules", dans le programme LCM de constructeur d'équipements spécifiques (OEM) représenté sur la figure 3 ;

Figure 8 : un schéma synoptique représentant de façon plus détaillée le mode de réalisation préféré de l'invention, dans lequel on ajoute des graphiques à une page de visualisation de format défini, pour un constructeur d'équipements spécifiques (OEM), présentée sur le tube cathodique du poste de commande numérique représenté sur la figure 1 ; et

Figure 9 : un organigramme du procédé de mise en oeuvre de la visualisation graphique sur une page de visualisation pour OEM qui est présentée au poste de commande numérique.

En considérant maintenant les dessins et plus particulièrement la figure 1, on voit la représentation, dans un but d'illustration, de l'environnement fonctionnel dans lequel l'invention est mise en oeuvre. La figure 1 montre, à titre d'exemple, un système de commande numérique par ordinateur (CNO) destiné à commander une machine-outil 10 et qui comprend trois éléments de matériel principaux. Ces éléments sont les suivants : le poste de commande numérique 12, le poste de commande de machine 14 et le coffret du système de commande numérique 16, dont le but est de loger les principales cartes de circuit imprimé, non représentées. Une telle configuration constitue un

exemple caractéristique d'un système de commande numérique de la marque Mark Century 2000 de la firme General Electric Company. Ce système est capable d'enregistrer des programmes d'ordinateur complets et de les appeler selon une séquence désirée, d'éditer des programmes par l'adjonction ou la suppression de sous-programmes dans des sections ou blocs appelés modules, et d'exécuter ensuite un ensemble complet d'instructions qui accomplit une tâche spécifique sur une machine-outil telle qu'un tour ou une fraiseuse.

Le poste de commande numérique 12 comprend un moniteur vidéo sous la forme d'un dispositif de visualisation à écran cathodique 18, et un clavier 20 ayant des touches de fonctions spéciales pour introduire en mémoire des ensembles d'information, qu'on appelle des fichiers, et pour commander ensuite la lecture de l'information de fichier d'une manière bien connue, pour l'exécution de programmes.

Le poste de commande de machine 14 comprend un ensemble de commandes manuelles et de boutons-poussoirs 22 qu'on utilise pour des fonctions accomplies par l'opérateur et concernant la machine, comme la sélection de mode, l'accès aux décalages, l'avance pas à pas, la vitesse de la broche, la vitesse d'avance, etc.

Les fonctions accomplies par le système de commande numérique par ordinateur représenté sur la figure 1 sont déterminées par des programmes enregistrés de façon interne qu'on appelle le logiciel de commande numérique par ordinateur. Le logiciel de commande numérique par ordinateur est constitué par trois ensembles de programmes différents, comme le montre la figure 2, et il comprend la logique de commande numérique 24 (qu'on appellera ci-après le logiciel de commande numérique), la fenêtre LCM 26 et la logique de commande de machine 28 (qu'on appellera ci-après le programme LCM). Le logiciel de commande numérique 24 commande la fabrication de pièces, le

mouvement selon les axes de la machine et la présentation de messages pour un tube cathodique. Le logiciel de commande numérique constitue une partie fixe ou permanente du système de commande numérique par ordinateur que l'utilisateur ne peut normalement pas changer. Le programme LCM commande de façon spécifique le fonctionnement de la machine-outil et il est constitué par un grand nombre de sous-programmes, contenus dans des sections modulaires appelées modules, qui permettent au constructeur d'équipements spécifiques (OEM) d'adapter le système à l'utilisateur. La fenêtre LCM 26 consiste en un logiciel, tel que celui décrit dans la demande de brevet des E.U.A. n° 415 041 précitée, qui permet au programme LCM 28 de communiquer avec le logiciel de commande numérique 24 et qui donne accès aux entrées et aux sorties de la machine-outil 10.

Comme la figure 3 le montre de façon plus détaillée, le programme LCM 28 est constitué par deux parties : la LCM fixe 30 et la LCM d'adaptation 32. La LCM d'adaptation contient les programmes destinés à commander le fonctionnement de la machine-outil 10, en utilisant le logiciel de fenêtre LCM 26 pour assurer l'interface nécessaire avec la machine. C'est la LCM d'adaptation 32 qui est spécialement conçue pour adapter la commande numérique par ordinateur à la commande d'une machine-outil particulière. La LCM fixe 30 est au contraire un logiciel dont la fonction fondamentale est de contrôler et de commander les dispositifs destinés à l'opérateur dans le poste de commande de machine 14 (figure 1), et on l'utilise sans aucune modification dans la plupart des applications.

La fenêtre LCM 26 est constituée par une série de procédures et de fonctions de fenêtre, 34, par des réseaux d'indicateurs d'état 36 et par des réseaux d'entrée et de sortie 38 qui, en majeure partie, ne peuvent pas être

changés, de façon que l'intégrité du logiciel de commande numérique ne puisse pas être compromise par des fonctions et des procédures quelconques programmables par l'utilisateur qui sont programmées dans le programme LCM 28.

Le logiciel de commande numérique 24 est constitué par un ensemble de routines destinées à commander la fabrication de pièces, 40, le mouvement selon les axes, 42, et les visualisations pour l'opérateur, 44, et qui procurent également des programmes commandant le mouvement de la broche, des temporisateurs, des fonctions mathématiques et des données de configuration de machine. Les routines de visualisation 44 présentent un intérêt particulier dans l'invention dans la mesure où elles comprennent des routines qui permettent à la LCM d'adaptation 32 de sélectionner ou "d'appeler" au choix certaines routines de page de visualisation du logiciel de commande numérique, afin de visualiser pour l'opérateur, à la demande, l'information d'état de machine désirée.

Avant de décrire l'invention en détail, on considérera brièvement la structure du programme LCM 28. Le programme LCM est constitué par une séquence exécutable de sous-programmes exprimés dans un langage de programmation de haut niveau qui ressemble à une langue naturelle comme l'anglais. Ce langage, appelé Programmable Controller Language (PCL), consiste en un groupe de données et d'instructions associées, similaires au PASCAL, qui sont destinées à l'accomplissement d'une opération particulière. On crée séparément des sous-programmes et on peut les utiliser aussi souvent qu'il est nécessaire pour exécuter soit une procédure qui accomplit une opération, soit une fonction qui accomplit une opération et retourne une valeur. Chaque programme comporte un identificateur (qu'on appelle un "appel") qui, lorsqu'il est rencontré, provoque l'exécution du programme considéré. Chaque sous-programme est introduit par une déclaration suivie par les

constantes, variables, réseaux ou types de données énumérés qu'il contient. Pour un système de commande numérique par ordinateur Mark Century 2000, on pourra se référer à une publication de General Electric intitulée "Introduction To PCL/MCL",NEC 1214, novembre 1983, pour avoir une explication détaillée du langage PCL. L'invention présente une utilité particulière lorsqu'elle est utilisée en association avec un système de commande numérique par ordinateur Mark Century 2000, mais il apparaîtra clairement que l'invention n'est pas limitée à ce système et peut tout aussi bien être utilisée avec d'autres systèmes de commande numérique.

En considérant maintenant la figure 4, on note que l'adaptation d'un programme MCL 28 pour la machine-outil 10 commence par l'introduction par un concepteur du système d'une série de sous-programmes PCL, désignés par la référence 46, dans le système de commande numérique par ordinateur, au moyen du clavier 20 du poste de commande numérique 12, dans lequel ces sous-programmes sont appliqués à un éditeur de fichiers 48. Ce dernier les transforme en fichiers PCL qui sont enregistrés dans une mémoire, non représentée. Les fichiers PCL sont également appliqués à un compilateur PCL 52 dans lequel ils sont ensuite traduits en code machine et sont enregistrés sous la forme d'un fichier de code machine 54. Pour achever le traitement du programme, les fichiers de code machine sont ensuite appliqués à un éditeur de liens 56 qui produit un seul fichier de code machine 58. Le fichier de code machine enchaîné est ensuite appliqué à un module d'implantation 60 qui affecte des adresses dans une mémoire vive, non représentée, à l'information contenue dans le fichier de sortie de l'éditeur de liens, et qui crée un nouveau fichier 62 contenant les affectations en mémoire vive. L'information provenant du fichier de sortie de l'éditeur de liens est la version finale en code

machine du programme LCM et elle est ensuite exécutée pendant le fonctionnement de la machine-outil 10.

Lorsque le logiciel de commande numérique par ordinateur, comprenant le programme LCM 28, est en place, l'exécution du logiciel de commande numérique 24 donne lieu à des appels répétés du programme LCM 28, comme le montre la figure 5. Chaque passe résultante par le programme LCM constitue ce qu'on appelle un balayage. Au cours de chaque balayage du programme LCM, les routines LCM fixes sont exécutées en premier pour contrôler et commander les dispositifs du poste de commande de machine 14. Ceci est suivi par l'exécution des routines LCM d'adaptation qui définissent le déroulement séquentiel et l'appel des sous-programmes de la machine-outil.

Le programme LCM d'adaptation 32 peut comprendre un certain nombre de modules de sous-programmes, chacun d'eux étant disponibles pour une adaptation individuelle. On peut augmenter ou diminuer le nombre de modules dans une adaptation particulière quelconque, en fonction de l'application considérée. La figure 7 représente certains modules de sous-programmes dont on pourrait disposer de façon caractéristique. L'invention concerne l'un des modules de sous-programmes, à savoir les pages de visualisation de LCM $32_1$. Cependant, à titre d'exemple, les modules de sous-programmes peuvent comprendre : un Module de Mise Sous Tension $32_2$ qui comprend les routines de démarrage et les verrouillages de sécurité utilisés sur la machine au cours de son démarrage ; un Module de Convoyeur $32_3$ qui contient toute la commande manuelle et de programme de pièce pour les transferts ; un Module de Contre-Poupée $32_4$ qui commande la contre-poupée sur le tour, un Module de Lubrification $32_5$ qui met en oeuvre la lubrification automatique de la machine qui doit être accomplie ou contrôlée de façon régulière ; un Module de Fourreau $32_6$ qui commande le fourreau de la contre-poupée

sous la dépendance du programme de pièce ; un Module de Mandrin $32_7$ qui réalise la commande de toutes les opérations de serrage et de desserrage du mandrin pendant l'exécution d'un programme de pièce ; un Module de Tourelle Porte-outils $32_8$ qui commande la tourelle porte-outils sous la dépendance du programme de pièce ; un Module de Broche $32_9$ qui commande la vitesse de la broche d'un tour, et un Module de Fluide de Refroidissement $32_{10}$ qui commande le fluide de refroidissement de la machine.

Après avoir ainsi présenté l'environnement fonctionnel, on peut maintenant considérer les détails de l'invention. L'invention concerne la visualisation graphique d'information sur le tube cathodique du moniteur vidéo 18 du poste de commande numérique 12 qui est représenté sur la figure 1. Le clavier 20 du poste de commande numérique 12 comprend une rangée de dix touches 66 par lesquelles un opérateur peut sélectionner une page quelconque parmi dix pages de visualisation principales qui sont fréquemment utilisées. Chaque page de visualisation consiste en une visualisation vidéo du type à balayage par trame et elle est générée à partir de fichiers de commande de visualisation enregistrés en mémoire, non représentés, qui font partie du logiciel de commande numérique 24. La rangée de touches de pages de visualisation 66 est représentée de façon plus détaillée sur la figure 6. On peut également sélectionner des pages de visualisation supplémentaires, appelées sous-pages, au moyen d'un jeu de touches ou de boutons-poussoirs désignés par la référence 68 sur la figure 1. Quatre touches de curseur sont prévues pour permettre la commande des mouvements vers le haut et le bas et vers la droite et la gauche d'un curseur, pour l'édition des données et le défilement vertical. Les sélections de sous-pages de droite sont représentées juste au-dessus de la rangée de

touches de pages de visualisation 66, tandis que les sélections de sous-pages de gauche sont représentées au-dessous de la rangée de touches de pages de visualisation, et sont respectivement désignées par les références 67 et 69. Les sélections de sous-pages de droite 67 comprennent, par exemple : "ETAT DE L'OUTIL" ; "POSITION DE MACHINE " ; "DONNEES D'OUTILLAGE" ; etc, tandis que les sélections de sous-pages de gauche 69 comprennent "ETAT TECH", "POSITION DE PROGRAMME", etc.

Au moyen de la LCM d'Adaptation 32, le logiciel du système de commande numérique par ordinateur offre au constructeur de la machine-outil ou au concepteur OEM la possibilité de définir le format de trois pages de visualisation de "machine". On peut ainsi utiliser l'adaptation pour visualiser n'importe quelle information d'état de machine désirée, pour l'assistance de l'opérateur ou du personnel d'entretien. On peut sélectionner ces pages de visualisation par la touche "MACHINE" 70 et par les options de pages de visualisation 72 et 74, correspondant respectivement aux sous-pages de droite et de gauche, sur la figure 6.

En considérant maintenant la figure 8, on note que lorsque l'opérateur sélectionne l'une des trois pages de visualisation de machine, le logiciel de commande numérique 24 réagit à une procédure de page de visualisation 78 conformément au type de la page sélectionnée, c'est-à-dire la page de gauche, la page de droite ou la page principale. L'information de texte, par opposition à l'information graphique, est alors appelée à partir des fichiers de commande de visualisation 80 dans lesquels elle a été enregistrée précédemment. L'information de texte, sous forme appropriée, est ensuite transmise au dispositif de visualisation à tube cathodique 18.

Lorsque des graphiques doivent être incorporés dans la visualisation complète faite sur le tube cathodi-

que, un "appel" incorporé dans les fichiers de commande de visualisation 80 déclenche les procédures nécessaires pour mettre en oeuvre la partie graphique de la visualisation. La routine de page de visualisation 78 réagit à la demande de visualisation graphique incorporée en déclenchant un appel vers la routine de page de visualisation LCM 32, par l'intermédiaire d'un point d'entrée 79 de la fenêtre 26. Une routine de changement de visualisation graphique 82 est exécutée en réponse. Cette routine 82 est accessible, par exemple à un constructeur d'équipements spécifiques adaptant une commande à une machine-outil particulière, pour réaliser l'adaptation de la partie statique de la visualisation graphique. L'adaptation est écrite en PCL, comme décrit ci-dessus.

La visualisation graphique que procure la routine de changement de visualisation graphique 82 est statique dans le sens où une fois que le format et le contenu de la visualisation ont été sélectionnés, ils demeurent inchangés jusqu'à ce qu'on effectue une nouvelle adaptation.

La routine de changement de visualisation graphique 82 est capable d'appeler certaines fonctions graphiques de base, par l'intermédiaire de points d'entrée 83 dans la fenêtre 26. A titre d'exemple, la routine de changement de visualisation graphique 82 dispose de fonctions permettant de : (1) déplacer le curseur du tube cathodique vers une position particulière de l'écran ; (2) tracer une ligne entre des points spécifiés ; (3) tracer un rectangle, connaissant les coordonnées de points correspondant à des angles opposés ; (4) tracer un arc de rayon et d'angle donnés ; (5) tracer un cercle ; (60) emplir une zone particulière de l'écran ; et (7) effacer une zone de l'écran.

Trois fonctions graphiques sont elles-mêmes des routines d'un type classique qui sont disponibles sur

appel dans la mémoire du système de commande numérique par ordinateur, non représentée. Ces fonctions constituent conjointement la routine de visualisation graphique 84. Des points d'entrée de fenêtre 83 séparés sont prévus pour chaque fonction graphique, de façon que chaque fonction soit disponible de la manière nécessaire pour composer une visualisation personnalisée.

Les routines de visualisation graphique 84 sont fusionnées avec le texte par l'intermédiaire des fichiers de commande de visualisation 80, puis l'information correspondante est visualisée sur le tube cathodique 18.

La partie statique de la visualisation graphique procure un fond ou un cadre dans lequel on peut visualiser sous forme graphique des données de fonctionnement en temps réel. La routine de mise à jour de visualisation graphique 86 fournit virtuellement en temps réel ces données de fonctionnement dynamiques. Dans le mode de page de visualisation, la procédure de page de visualisation 78 effectue un appel périodique, par exemple toutes les 150 millisecondes, par le point d'entrée de fenêtre 81, dirigé vers la routine de mise à jour de visualisation graphique 86. La routine 86 est spécialement adaptée de façon à échantillonner l'information exigée qui est disponible par nature dans le système de commande numérique. A titre d'exemple, pour un tour, on dispose aisément d'une information de vitesse de la broche, en temps réel. Cette information est simplement cadrée par la routine de mise à jour de visualisation graphique 86 qui déclenche alors une ou plusieurs des fonctions graphiques (par les points d'entrée 83), de façon à visualiser l'information proportionnellement à sa vraie valeur en temps réel.

La routine de mise à jour de visualisation graphique 86 fonctionne ainsi de façon pratiquement identique à la routine de changement de visualisation graphique 82, à l'exception du fait que la routine 86 est appelée de

façon répétée et est conçue de façon à inclure des données de fonctionnement instantanées. Les informations graphiques statiques et dynamiques qui sont ainsi produites sont fusionnées avec le texte par l'intermédiaire des fichiers de commande de visualisation 80 et de la routine de page de visualisation 78 pour la présentation sur le tube cathodique.

Pour aider à la compréhension de l'invention, on envisagera une visualisation graphique de type caractéristique qu'on pourrait désirer. A titre d'exemple, pour créer un graphique à segments de la vitesse de la broche pour un tour, exprimée en pourcentage de la vitesse maximale de la broche, on donne simplement à la LCM d'adaptation 32 une configuration capable de créer deux rectangles contigus, comme l'indique la référence 88 sur le tube cathodique 18 de la figure 8. Bien qu'il puisse y avoir une partie statique dans la visualisation d'ensemble, le graphique à segments qui doit être créé est composé essentiellement par des données qui sont toutes des données en temps réel mises à jour de façon répétitive. Du fait que les parties statiques et dynamiques sont obtenues par des procédures très similaires, il suffit simplement d'envisager la partie dynamique de l'information graphique à visualiser.

Pour créer le graphique à segments de la figure 8, la routine de page de visualisation 78 appelle toutes les 150 millisecondes la routine de mise à jour de visualisation graphique 86, par l'intermédiaire du point d'entrée 81 de la fenêtre LCM. La routine de mise à jour de visualisation graphique 86 dispose de données en temps réel concernant la vitesse de la broche du tour, qui proviennent du système de commande. Cette information est cadrée et des ordres de tracé sont ensuite émis conformément à l'information cadrée. A titre d'exemple, quatre ordres sont nécessaires : (1) déplacer le curseur jusqu'à

un point de départ sélectionné d'un premier rectangle, dans le coin inférieur gauche de celui-ci ; (2) tracer le premier rectangle jusqu'à un point final situé dans le coin supérieur droit, ce point étant fonction de la vitesse réelle de la broche ; (3) déplacer à nouveau le curseur vers le point de départ d'un second rectangle, dans le coin inférieur gauche de celui-ci, de façon que le second rectangle commence à l'endroit où le premier rectangle se termine ; et (4) tracer ensuite le second rectangle jusqu'au point final dans le coin supérieur droit, ce point représentant la vitesse maximale de la broche. Ainsi, lorsque la vitesse de la broche augmente ou diminue, les deux rectangles augmentent et/ou diminuent de longueur de façon correspondante. L'aspect visuel est celui d'un graphique à segments représentant la vitesse réelle de la broche.

L'Annexe I est un programme PCL pour créer un graphique à segments de la manière décrite ci-dessus. L'Annexe II contient des routines destinées à produire les appels de fenêtre pour générer les graphiques.

En considérant maintenant la figure 9, on voit un organigramme qui illustre les étapes fonctionnelles qui sont mises en oeuvre dans le logiciel pour générer des graphiques sur les pages de visualisation de la LCM d'adaptation pour OEM, qui sont présentées sur le moniteur vidéo 18. La première étape est désignée par la référence 90 et constitue un appel de l'une des trois pages de visualisation de "machine", par l'intermédiaire du module de page de visualisation LCM 32, lorsqu'un opérateur demande une page de visualisation LCM particulière. On rappelle que ceci se produit lorsqu'un opérateur actionne l'une des touches faisant partie des claviers 66 et 68 du poste de commande numérique 12 (figure 1). Un test est ensuite effectué pour déterminer l'existence et la disponibilité d'une page particulière, ce qui

est indiqué par l'étape 92. L'étape 93 correspond à l'obtention de valeurs suivantes des paramètres désirés qui doivent faire l'objet d'une visualisation graphique.

Après la détermination de la page de visualisation et des valeurs de paramètres, il y a émission d'un appel pour présenter la page de visualisation, et la routine de page de visualisation 78 est alors exécutée, comme l'indique l'étape 96. La partie d'initialisation de la génération graphique est ensuite exécutée (étape 98) et est suivie par une mise à jour périodique de la visualisation graphique, ce qui est indiqué par l'étape 100. Pendant chacune des étapes de génération graphique 98 et 100, il y a des appels de programme 82 et 86 émis par le programme LCM d'adaptation 32, pour exécuter sept procédures de tracé 102, 104, ... 114, avec les paramètres appropriés, pour produire les visualisations graphiques désirées en mettant en oeuvre les sept fonctions exposées précédemment telles que : le déplacement du curseur graphique jusqu'à une position donnée, le tracé d'un rectangle, etc.

On a ainsi décrit et représenté un procédé et un système pour produire des représentations graphiques sur un moniteur de visualisation de machine-outil à commande numérique, afin de visualiser à la demande de l'opérateur une page de visualisation définie par un constructeur d'équipements spécifiques (OEM).

ANNEXE I

```
PROCEDURE GRFDSP_MAIN IS  -- start of srfdsp_main

BEGIN

  CASE GRFDSP_MASTER IS

    WHEN INIT_STATE =>
      GRFDSP_MASTER := RUN_STATE;

    WHEN RUN_STATE =>

      IF GRFDSP_OK
        THEN
          CASE GRFDSP_STATE IS

            WHEN GRFDSP_STANDBY =>
              if active_disp_page=90
              then
                 TEMP := draw_clear;
                 srfdsp_state := write_state;
                 OLDXCURSOR := 0;
              end if;

            when write_state =>
              if active_disp_page=90
              then
                xcursor := trunc((pros_s * nc_sso_value)/10.0);
                if xcursor > 400
                then
                  xcursor := 400;
                elsif xcursor <= 0
                then
                  xcursor := 1;
                end if;
                xcursor := xcursor + 60;
                IF OLDXCURSOR /= XCURSOR
                THEN
                  OLDxcursor := XCURSOR;
                  temp := draw_move(60,120);
                  temp := draw_rectangle(OLDxcursor,75,4,8);
                  temp := draw_move(OLDxcursor,120);
                  temp := draw_rectangle(461,95,3,8);
                END IF;
              ELSE
                GRFDSP_STATE := GRFDSP_STANDBY;
                TEMP := DRAW_CLEAR;
              end if;
          END CASE;

      ELSE  -- not srfdsp_ok
          GRFDSP_MASTER := ERROR_STATE;
      END IF;
  END CASE;
```

```
PROCEDURE START_SEQ( OP_CODE : ENTIER ) ;                    (début   3629)

*****************************************************************************
* DATE DE PROGRAMMATION          : 03/01/84                                 *
* NOM DU PROGRAMMEUR             : BILL GODDARD                             *
* DESCRIPTION                    : INITIALISE "GRAPHICS_SEQ" AVEC LE CARACTERE <EM>  *
*                                  SUIVI PAR LE PARAMETRE "OP_CODE".        *
*                                                                           *
* PARAMETRES ASSOCIES                                                       *
* REFERENCES DES DONNEES GLOBALES                                           *
*                                                                           *
*****************************************************************************

CONST
    EM = CHR(16+19) ;
BEGIN
    GRAPHICS_SEQ( 1 ) := EM;
    GRAPHICS_SEQ( 2)  := CHR( OP_CODE );
    GRAPHICS_SEQ.LEN  := 2;
END ;

                                                    ( fin     3629)
```

```
PROCEDURE ADD_INT_TO_SEQ( I : ENTIER );                    (début  3629)
*****************************************************************************
* DATE DE PROGRAMMATION              : 03/01/84                             *
* NOM DU PROGRAMMEUR                 : BILL GODDARD                         *
* DESCRIPTION                        : AJOUTE LE PARAMETRE ENTIER A LA CHAINE *
*                                      "GRAPHICS_SEQ". ON UTILISE DEUX OCTETS DE *
*                                      LA CHAINE. LES SEPT BITS DE PLUS FORT POIDS *
*                                      DU PARAMETRE VONT DANS LE PREMIER OCTET.  **
*                                      LES SEPT BITS DE MOINDRE POIDS VONT DANS  *
*                                      LE SECOND OCTET. LE BIT DE PLUS FORT POIDS *
*                                      DE CHAQUE OCTET EST FIXE A UN POUR INDIQUER *
*                                      UN OCTET DE DONNEES.                   *
*                                                                             *
*                                                                             *
* PARAMETRES ASSOCIES                                                        *
* REFERENCES DES DONNEES GLOBALES    : GRAPHICS_SEQ, LOCAL, KGTASK.          *
*                                                                             *
*****************************************************************************

BEGIN
GRAPHICS_SEQ( GRAPHICS_SEQ.LEN + 1 ) := CHR(((I DIV 128) AND 127) + 128);
GRAPHICS_SEQ( GRAPHICS_SEQ.LEN + 2) := CHR(((I MOD 128)AND 127) + 128);
GRAPHICS_SEQ.LEN := GRAPHICS_SEQ.LEN + 2;
END;
                                                           (fin  3629)
```



2569877

```
PROCEDURE END_AND_SEND_SEQ;                                    (début   3629)
*****************************************************************************
*                                                                           *
* DATE DE PROGRAMMATION                : 03/01/84                           *
* NOM DU PROGRAMMEUR                   : BILL GODDARD                       *
* DESCRIPTION                          : TERMINE LA CHAINE "GRAPHICS_SEQ" EN*
*                                        AJOUTANT LE CARACTERE <EM> ET EMET LA*
*                                        CHAINE VERS LE SGCN.*              *
*                                                                           *
* PARAMETRES ASSOCIES                                                       *
* REFERENCES DES DONNEES GLOBALES      : GRAPHICS_SEQ, LOCAL, KGTASK.       *
*                                        KGD_CRT_FILE_CODE                  *
*                                                                           *
*****************************************************************************
CONST
    EM = CHR( 16#19 );
BEGIN
    GRAPHICS_SEQ.LEN := GRAPHICS_SEQ.LEN + 1;
    GRAPHICS_SEQ( GRAPHICS_SEQ.LEN ) := EM;
    Ksr_PUTSTR ( GRAPHICS_SEQ ) ;
END ;
                                                               (fin    3629)
```

* SGCN = Système graphique pour la commande numérique

**ANNEXE II (suite)**

```
                                                    (début  3629)
PROCEDURE KGR_MOVE ;
          (  X : ENTIER;     )
          (  Y : ENTIER      )
*************************************************************
* DATE DE PROGRAMMATION      : 03/01/84                     *
* NOM DU PROGRAMMEUR         : BILL GODDARD                 *
* DESCRIPTION                : EMET UNE SEQUENCE DE COMMANDE*
*                              GRAPHIQUE VERS SGCN POUR AMENER LE
*                              CURSEUR GRAPHIQUE A LA POSITION
*                              X, Y                         *
*                                                           *
*************************************************************
BEGIN
    START_SEQ( 2#01001010 );
      ADD_INT_TO_SEQ( X );
      ADD_INT_TO_SEQ( Y );
    END_AND_SEND_SEQ;
END;
                                                    ( fin   3629)
```

```
PROCEDURE KGR_LINE;                                          (début  3629)
        (         ( X       : ENTIER;     )
        (           Y       : ENTIER;     )
        (           COLOR   : ENTIER;     )
        (           PATTERN : ENTIER )    )
*****************************************************************************
*                                                                           *
* DATE DE PROGRAMMATION                    : 03/01/84                       *
* NOM DU PROGRAMMEUR                       : BILL GODDARD                   *
* DESCRIPTION                              : EMET UNE SEQUENCE DE COMMANDE  *
*                                            GRAPHIQUE VERS LE SGCN QUI TRACE *
*                                            UNE LIGNE ALLANT DE LA POSITION  *
*                                            COURANTE DU CURSEUR GRAPHIQUE    *
*                                            JUSQU'A X, Y.                    *
*                                                                           *
*****************************************************************************
BEGIN
    START_SEQ( 2#01000000 );
    ADD_INT_TO_SEQ( X );
    ADD_INT_TO_SEQ( Y );
    ADD_COLOR_PATTERN( COLOR. PATTERN );
    END_AND_SEND_SEQ
END;
                                                             (fin     3629)
```



2569877

**ANNEXE II (suite)**

```
PROCEDURE KGR_RECTANGLE;                              (début  3629)
      ( X         : ENTIER      )
      ( Y         : ENTIER      )
      ( COLOR     : ENTIER      )
      ( PATTERN   : ENTIER  )     )
*****************************************************************
*                                                               *
* DATE DE PROGRAMMATION       : 03/01/84                        *
* NOM DU PROGRAMMEUR          : BILL GODDARD                    *
* DESCRIPTION                 : EMET UNE SEQUENCE DE COMMANDE   *
*                               GRAPHIQUE VERS LE SGCN QUI TRACE*
*                               UN RECTANGLE                    *
*                                                               *
*****************************************************************

BEGIN
    START_SEQ( 2#01000010 );
    ADD_INT_TO_SEQ( X );
    ADD_INT_TO_SEQ( Y );
    ADD_COLOR_PATTERN( COLOR, PATTERN );
    GRAPHICS_SEQ.LEN := GRAPHICS_SEQ.LEN + 1;    ( ajoute une rotation de
    GRAPHICS_SEQ( GRAPHICS_SEQ.LEN ) := CHR( 2#10000000 );    valeur zéro )
    END_AND_SEND_SEQ
END;                                                  ( fin   3629)
```

```
PROCEDURE KGR_ARC;                                          (début  3629)
        (       ( X       : ENTIER;     )
        (         Y       : ENTIER;     )
        (         CX      : ENTIER;     )
        (         CY      : ENTIER;     )
        (         DIR     : ENTIER;     )
        (         COLOR   : ENTIER;     )
        (         PATTERN : ENTIER )    )
*****************************************************************************
* DATE DE PROGRAMMATION                    : 03/01/84                       *
* NOM DU PROGRAMMEUR                       : BILL GODDARD                   *
* DESCRIPTION                              : EMET UNE SEQUENCE DE COMMANDE  *
*                                            GRAPHIQUE VERS LE SGCN QUI TRACE*
*                                            UN ARC.                        *
*                                                                           *
*****************************************************************************
BEGIN
    START_SEQ( 2#01000100 );
    ADD_INT_TO_SEQ( X );
    ADD_INT_TO_SEQ( Y );
    ADD_INT_TO_SEQ( CX );
    ADD_INT_TO_SEQ( CY );
    GRAPHICS_SEQ.LEN := GRAPHICS_SEQ.LEN + 1 ;
    GRAPHICS_SEQ( GRAPHICS_SEQ.LEN ) := CHR( 128 + DIR );
    ADD_COLOR_PATTERN( COLOR, PATTERN );
    END_AND_SEND_SEQ
END;
                                                            (fin    3629)
```

ANNEXE II (suite)

```
                                        (début  3629)

PROCEDURE KGR_CIRCLE;
          (                 )
          (RADIUS   : ENTIER; )
          (COLOR    : ENTIER; )
          (PATTERN  : ENTIER  )
 ********************************************************************
 * DATE DE PROGRAMMATION      : 03/01/84                            *
 * NOM DU PROGRAMMEUR         : BILL GODDARD                        *
 * DESCRIPTION                : EMET UNE SEQUENCE DE COMMANDE       *
 *                              GRAPHIQUE VERS LE SGCN QUI TRACE    *
 *                              UN CERCLE.                          *
 *                                                                  *
 ********************************************************************

BEGIN
    START_SEQ( 2#01000110 );
    ADD_INT_TO_SEQ( RADIUS );
    ADD_COLOR_PATTERN( COLOR, PATTERN );
    END_AND_SEND_SEQ
END;

                                        (fin  3629)
```

```
PROCEDURE KGR_FILL;                                          (début  3629)
                (          (   X       : ENTIER;       )
                (              Y       : ENTIER;       )
                (              COLOR   : ENTIER;       )
                (              PATTERN : ENTIER    )   )
*****************************************************************************
* DATE DE PROGRAMMATION                    : 03/01/84                       *
* NOM DU PROGRAMMEUR                       : BILL GODDARD                   *
* DESCRIPTION                              : EMET UNE SEQUENCE DE COMMANDE  *
*                                            GRAPHIQUE VERS LE SGCN QUI REMPLIT *
*                                            LA ZONE IDENTIFIEE PAR X ET Y.  *
*                                                                           *
*****************************************************************************

BEGIN
    START_SEQ( 2#01001001 );
    ADD_INT_TO_SEQ( X );
    ADD_INT_TO_SEQ( Y );
    ADD_COLOR_PATTERN( COLOR, PATTERN );
    END_AND_SEND_SEQ
END;
                                                             (fin   3629 )
```



2569877

```
PROCEDURE KGR_CLEAR;                                          (début  3629)
*****************************************************************************
*                                                                           *
* DATE DE PROGRAMMATION                    : 03/01/84                       *
* NOM DU PROGRAMMEUR                       : BILL GODDARD                   *
* DESCRIPTION                              : EMET UNE SEQUENCE DE COMMANDE  *
*                                            GRAPHIQUE VERS LE SGCN QUI EFFACE *
*                                            LA ZONE GRAPHIQUE              *
*                                                                           *
*****************************************************************************

BEGIN
    START_SEQ( 2#01001000 );
    ADD_INT_TO_SEQ( 0 );          ( g1 = g2 = 0, efface l'écran complet )
    ADD_INT_TO_SEQ( 0 );
    END_AND_SEND_SEQ
END;
```

## REVENDICATIONS

1. Procédé de commande du fonctionnement d'un système de commande numérique par ordinateur pour visualiser une information prédéterminée concernant le fonctionnement d'un système d'usinage, sous la forme de graphiques, caractérisé en ce qu'il comprend les opérations suivantes : (a) on demande une visualisation du type à balayage par trame sur une dispositif de visualisation (18) observable par un opérateur ; (b) on détermine des valeurs de l'information prédéterminée qu'on désire visualiser ; (c) on génère initialement une représentation graphique de l'information prédéterminée ; et (d) on met périodiquement à jour cette représentation graphique pour visualiser des valeurs variables de l'information prédéterminée.

2. Procédé selon la revendication 1, caractérisé en ce que l'information prédéterminée consiste en une information d'état de machine.

3. Procédé selon la revendication 1, caractérisé en ce que la visualisation du type à balayage par trame comprend une page parmi un ensemble de pages de visualisation présentées sur un moniteur vidéo (18) pour visualiser une information d'état de machine spécifiée par le constructeur d'équipements spécifiques (OEM).

4. Procédé selon la revendication 1, caractérisé en ce que la page de visualisation prédéterminée comprend également une information de message sous forme alphanumérique.

5. Procédé selon la revendication 1, caractérisé en ce que les opérations de génération et de mise à jour de la représentation graphique comprennent sélectivement les opérations consistant à amener un curseur graphique à une position donnée, à tracer une ligne, à tracer un rectangle, à tracer un arc, à tracer un cer-

cle, à emplir une zone prédéterminée de l'emplacement de la représentation graphique, et à effacer une zone prédéterminée de l'emplacement de la représentation graphique.

6. Procédé selon la revendication 5, caractérisé en ce que le système de commande numérique par ordinateur comprend une section logique de commande numérique (24) et une section logique de commande de machine (LCM) (28), réalisées par logiciel, et en ce que l'opération de demande de la page de visualisation comprend la demande d'une page de visualisation de LCM correspondant à une information d'état de machine prédéterminée.

7. Procédé selon la revendication 6, caractérisé en ce que l'opération de demande d'une page de visualisation de LCM comprend en outre l'inclusion d'un appel dans des fichiers de commande de visualisation (80) de la section de commande numérique (24), pour la génération de graphiques (88), cette section de commande numérique réagissant à l'appel et déclenchant un appel vers la section LCM (28), et la section LCM exécutant une ou plusieurs routines qui comprennent des appels renvoyés vers la section de commande numérique (24) pour produire les graphiques (88).

8. Procédé selon la revendication 7, caractérisé en ce que le système de commande numérique par ordinateur comprend en outre une interface (26) réalisée par logiciel qui établit une liaison de communication entre la section de commande numérique (24), la section LCM (28) et le dispositif d'entrée/sortie du système d'usinage, et en ce que l'opération consistant à effectuer des appels d'exécution comprend la transmission de ces appels par ladite interface (26).

9. Procédé selon la revendication 8, caractérisé en ce que la section LCM (28) comprend une section d'adaptation (32) réalisée par logiciel et programmable par le constructeur d'équipements spécifiques (OEM), et

les appels d'exécution de logiciel pour une visualisation graphique proviennent d'un module de sous-programmes de la section LCM d'adaptation (32).

10. Procédé selon la revendication 9, caractérisé en ce que le dispositif de visualisation comprend un moniteur de type vidéo (18) comportant une section de présentation, et en ce que les opérations de génération et de mise à jour de la représentation graphique comprennent sélectivement les opérations consistant à amener un curseur graphique à une position donnée sur l'écran de présentation, à tracer une ligne de longueur prédéterminée, à tracer un rectangle de dimensions prédéterminées, à tracer un arc de longueur prédéterminée, à tracer un cercle de diamètre prédéterminé, à emplir une zone prédéterminée de l'écran et à effacer une zone prédéterminée de l'écran.

11. Système de commande numérique destiné à commander une machine-outil (10) comprenant des moyens (18) destinés à visualiser une information pour un opérateur, caractérisé en ce qu'il comprend : un logiciel de logique de commande numérique (24) et un logiciel de logique de commande de machine (LCM) (28), comprenant respectivement des premier et second ensembles de programmes d'ordinateur enregistrés, pouvant être exécutés à la demande pour commander le fonctionnement du système ; le premier ensemble de programmes peut, lorsqu'il est appelé, accéder à un fichier de commande de visualisation (80) et lire ce fichier, transmettre des signaux vers les moyens de visualisation (18) pour présenter à l'opérateur une information sous forme de texte, et déclencher un appel vers le second ensemble de programmes pour générer une visualisation graphique ; et le second ensemble de programmes comprend un ensemble de sous-programmes qui sont exécutés sous la dépendance de l'appel pour générer des appels en retour vers le premier ensemble de programmes, afin de visualiser une information prédéterminée sous la

forme de ladite visualisation graphique.

12. Système selon la revendication 11, caractérisé en ce que le second ensemble de programmes est au moins partiellement constitué par un ensemble de modules de sous-programmes ($32_1$ - $32_{10}$) qui peuvent être adaptés à une application particulière par un constructeur d'équipements spécifiques (OEM).

13. Système selon la revendication 12, caractérisé en ce que les modules de sous-programmes comprennent un module ($32_1$) destiné à appeler au moins une visualisation graphique sous la forme d'une page de visualisation d'information d'état de machine.

14. Système selon la revendication 13, caractérisé en ce que le module appelant la visualisation graphique comprend un premier sous-programme (82) destiné à appeler initialement la génération d'une visualisation graphique et un second sous-programme (86) destiné à mettre périodiquement à jour la visualisation graphique générée initialement, cette mise à jour comprenant l'effacement des visualisations graphiques présentées sur un écran.

15. Système selon la revendication 14, caractérisé en ce que les deux sous-programmes comprennent des routines destinées à appeler les programmes de la commande numérique, et en ce que les programmes de la commande numérique comprennent des routines (84) destinées à réaliser le déplacement d'un curseur graphique, le tracé d'une ligne, le tracé d'un rectangle, le tracé d'un arc, le tracé d'un cercle, le remplissage d'une zone prédéterminée et l'effacement d'une zone prédéterminée ainsi que de la visualisation graphique complète.

16. Système selon la revendication 15, caractérisé en ce qu'il comprend en outre une interface (26) constituée par un troisième ensemble de programmes enregistrés (34), un ensemble d'indicateurs d'état (36) et un

ensemble de réseaux d'entrée/sortie (38), et cette interface relie les premier et second ensembles de programmes (24, 28) et la machine-outil (10), et les appels de programme entre les premier et second ensembles de programmes sont transmis par cette interface (26).

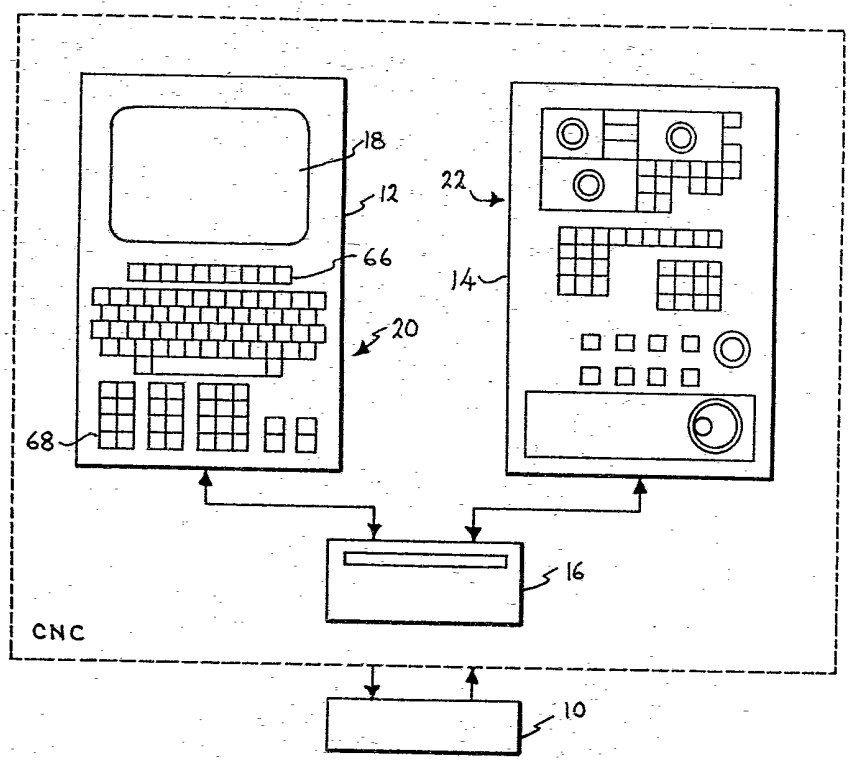



FIG. 1

LOGICIEL DE COMMANDE NUMERIQUE PAR ORDINATEUR

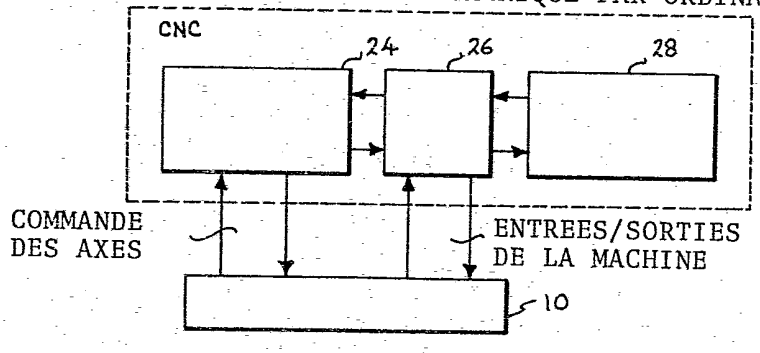



FIG. 2

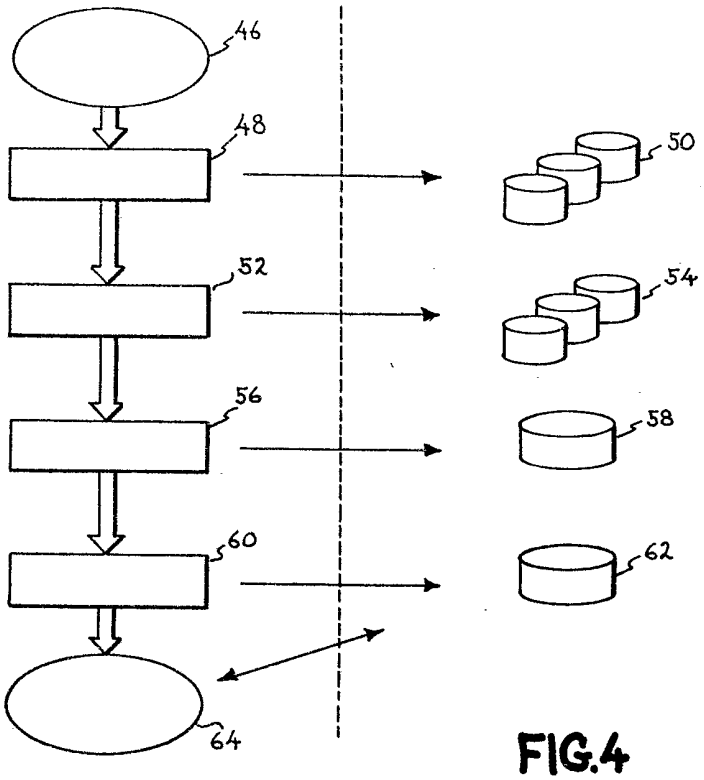



FIG.4

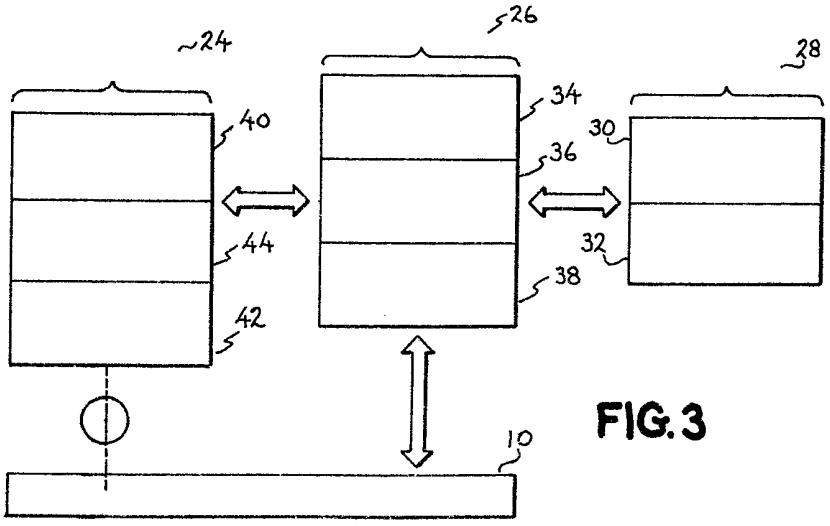



FIG.3

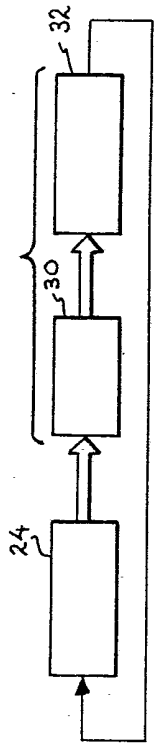

28
32
30
24

FIG.5

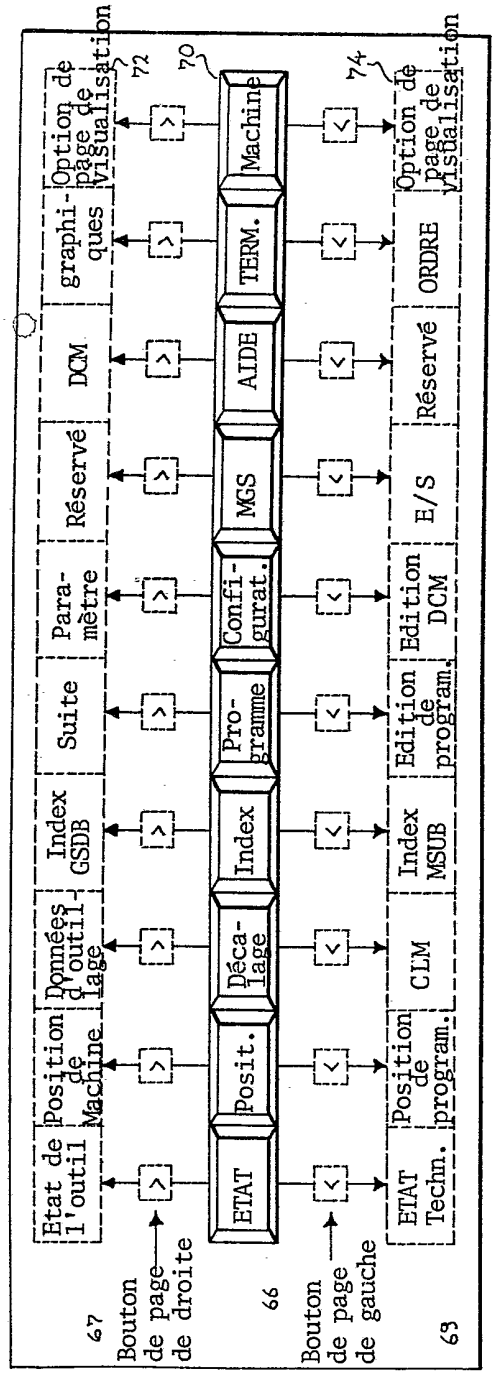

Etat de l'outil
Position de Machine
Données d'outil-lage
Index GSDB
Suite
Para-mètre
Réservé
DCM
graphi-ques
Option de page de visualisation
ETAT
Posit.
Déca-lage
Index
Pro-gramme
Confi-gurat.
MGS
AIDE
TERM.
Machine
ETAT Techn.
Position de program.
CLM
Index MSUB
Edition de program.
Edition DCM
E/S
Réservé
ORDRE
Option de page de visualisation
Bouton de page de droite
Bouton de page de gauche
67
66
63
70
72
74

FIG.6

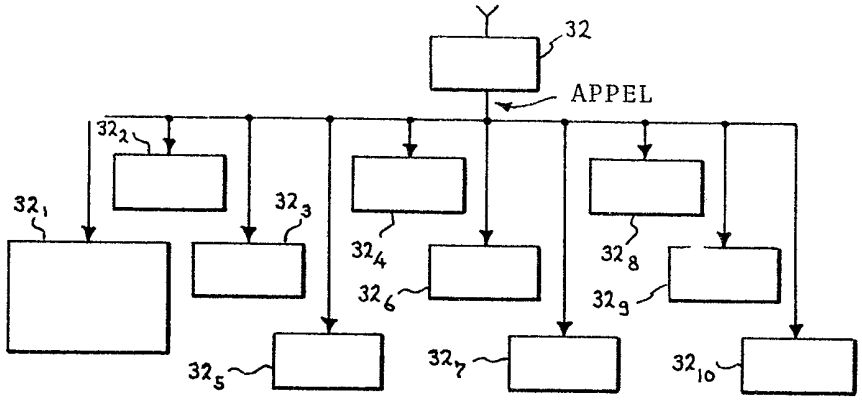

32
APPEL
32₂
32₁
32₃
32₄
32₈
32₆
32₉
32₅
32₇
32₁₀

FIG.7

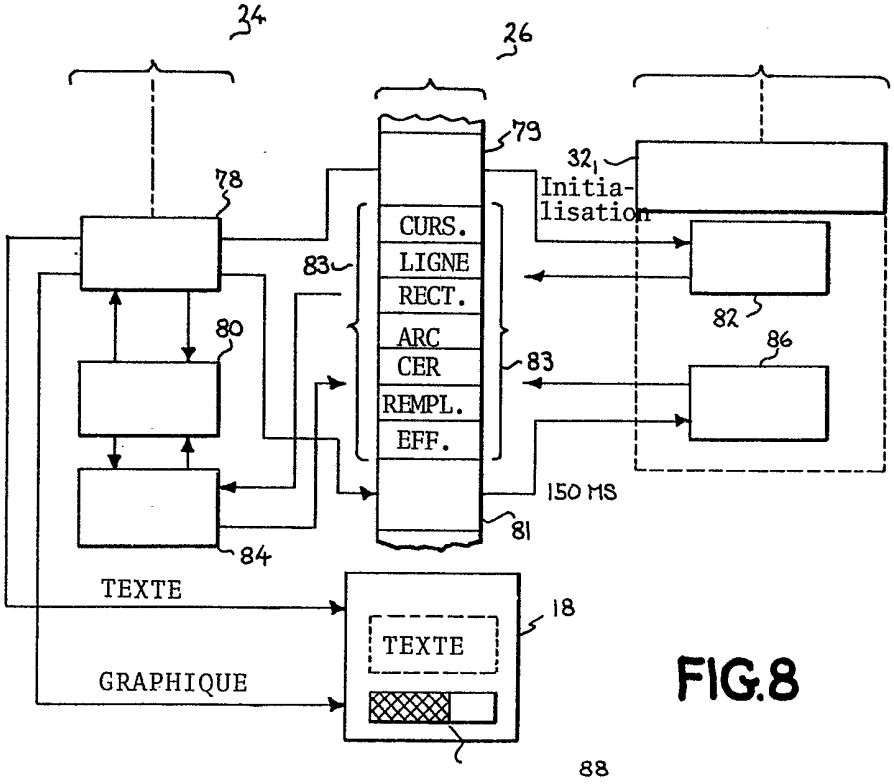

24
26
79
78
32₁
Initia-
lisation
CURS.
LIGNE
RECT.
ARC
CER
REMPL.
EFF.
83
80
82
86
150 MS
81
84
TEXTE
18
TEXTE
GRAPHIQUE
88

FIG.8

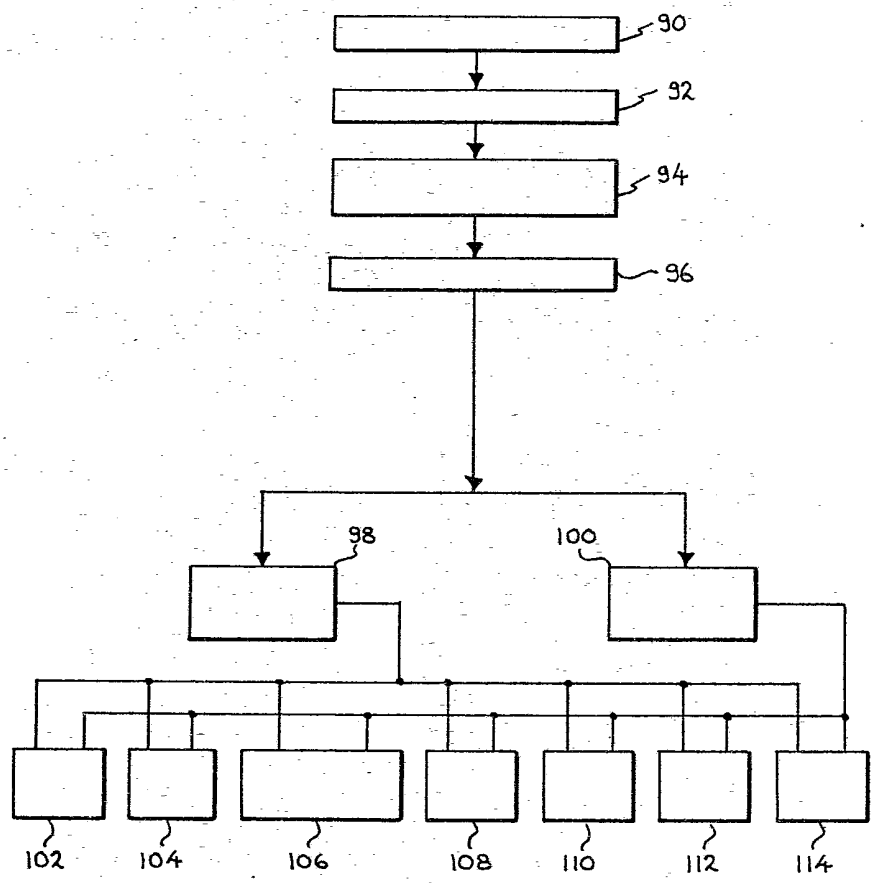



FIG.9