Deux approches de la conduite de procédé : automate contre calculateur

par M. Brayard *(Division Automation, Société Générale pour les Techniques Nouvelles (S.G.N.), B.P. n° 30, 78184 Saint-Quentin-en-Yvelines Cedex.)*

Introduction

Depuis plus de 10 ans, nous assistons à un développement assez spectaculaire de l'utilisation des calculateurs pour la conduite des procédés industriels. Cet accroissement correspond au souci des industriels d'améliorer la productivité des installations, mais aussi à la baisse importante du coût des matériels informatiques que le progrès technologique a entraînée. Les fonctions confiées au calculateur sont multiples et souvent complexes, mais leur réalisation demande une équipe d'informaticiens compétents.

Plus récemment, sont apparus sur le marché les automates programmables, dans une gamme très large de possibilités et de coûts. Construits pour la plupart autour d'un microprocesseur, ces appareils doivent leur succès à la facilité avec laquelle ils peuvent être mis en œuvre, puis modifiés, par l'utilisateur non informaticien. Leur langage de programmation, simple et accessible à l'automaticien, est la principale raison de leur grande utilisation dans le domaine des automatismes séquentiels, aussi peu standardisés que les procédés qu'ils contrôlent.
Longtemps considérés comme la solution de tous les problèmes, les microprocesseurs trouvent maintenant leur place comme simples composants d'une boîte noire destinée à répondre à des *besoins spécifiques*. Les systèmes modernes de régulation sont un bon exemple où le microprocesseur réalise la fonction de régulation, la simulation des faces avant de régulateur sur écran de télévision, et permet à un coût marginal faible des tracés de courbes d'historique et de tendance, au sein d'un *système aux possibilités figées*.

L'augmentation sensible des performances des automates programmables (notamment calculs et traitements numériques), les autorise maintenant à traiter aussi bien les variables logiques qu'analogiques, les automatismes séquentiels que les régulations, avec le même langage simple de programmation.

Nous tentons ici de dégager les grandes lignes d'une comparaison entre calculateur et automate, en nous appuyant sur notre expérience dans la conduite des fours de verre textile, que nous avons eu la chance de réaliser dans les 2 approches aux États-Unis et en Espagne, après qu'une première installation à Chambéry ait permis l'identification et la modélisation du procédé. Nous comparerons TOPICS (Table Oriented Process Information and Control System), logiciel fonctionnant sur calculateur General Automation à APILOG, automate programmable développé et construit par SGN (Société Générale pour les Techniques Nouvelles).

I. Constitution des systèmes

I.1. Approche calculateur (TOPICS)

Le système TOPICS a pour fonctions principales l'acquisition et le traitement des mesures, la détection et l'édition des alarmes, le contrôle en boucle fermée et le dialogue opérateur. Il fonctionne bien entendu sur toute la gamme des calculateurs General Automation (SPC. 16 et GA. 16), dont il utilise tous les périphériques « informatiques » (disque, imprimante, console, etc.) et « industriels » (entrées-sorties analogiques et logiques), dont la liste apparaît sous le repère 1, dans la figure 1.
Il se compose, au plan du logiciel, d'un ensemble de modules qui sont soit des programmes *intégrés au système temps réel* RTOS qui gère leur exécution, soit des tables contenant les données des programmes ou assurant les transmissions entre tâches.

La figure 1 explique de façon simplifiée le fonctionnement de TOPICS : pour les *traitements analogiques,* qui sont le cœur du système et pour lesquels des fonctions standards sont fournies, le constructeur a prévu l'enchaînement suivant :

a) Le système temps réel déclenche, à intervalles fixes, le programme AIDOX (repère 4). Celui-ci examine la table des voies et construit la table des voies à traiter (fonction des fréquences indiquées pour chacune), avec en particulier leur adresse et leur gain. Il active ensuite le système d'entrée-sortie (IOS) qui gère les échanges avec tous les périphériques.

b) IOS (repère 1 à gauche de la figure) se charge de l'acquisition des voies et de la mémorisation des valeurs brutes dans la table ARPT (Analog Raw Point Table) 5.

c) Dès qu'une valeur est stockée dans ARPT, le traitement PMS 6 (Process Monitoring System) est lancé : son but est principalement le filtrage, la linéarisation et la détection des alarmes, les valeurs physiques sont stockées dans la table AIPT (Analog Input Point Table) 7.

d) Lorsqu'une action est à entreprendre après le traitement d'une voie (ou à fréquence fixe sur option), les programmes DDC (Direct Digital Control) sont alors activés (repère 8) : ils se chargent du calcul de l'écart entre consigne (dans la table CSPT Control Set Point Table, repère 13) et mesure, des algorithmes PI, PID, PIDI, PII et de la mémorisation dans la table AOPT (Analog Output Point Table, repère 9) de la correction incrémentale ou absolue à appliquer au procédé.

e) Le système d'entrée-sortie IOS se charge enfin de l'envoi des corrections aux actionneurs, par l'intermédiaire des périphériques industriels de sortie (repère 4).

En parallèle, une structure identique peut accueillir tous les programmes que l'utilisateur voudrait écrire pour le traitement de voies tout ou rien, aucun module de traitement n'étant fourni.

Enfin, un *système complexe de communications* (Console Communication System, repère 28) assure la gestion des périphériques de dialogue et la relation entre les questions-réponses des opérateurs et les paramètres de traitement.

La construction du système est facilitée par 3 « compilateurs » qui assurent la conversion d'ordres sources en tables nécessaires au traitement PMS-DDC (compilateur DBASE repère 3), aux formats d'affichage sur les consoles (FORMAT repère 26) et aux relations entre affichage et paramètres (OPCON repère 25).

Pour les 32 voies d'acquisition et les 34 régulations (y compris cascades internes), il a fallu :
- une unité centrale de 32 K mots de 16 bits ;
- un disque 2,5 M octets, non complètement utilisé mais nécessaire pour le dialogue opérateur (et uniquement pour lui) ;
- une console de dialogue ;
- les coupleurs industriels ;
- une imprimante semi-rapide et un lecteur de carte pour le développement et la mise au point (voir § II).

I.2. Approche automate programmable (APILOG)

La structure de l'automate APILOG de SGN, est finalement peu différente de celle d'un calculateur : une unité centrale (processeur de la figure 2) dialogue avec une mémoire (ferrite ou semi-conducteur), des cartes d'entrées-sorties industrielles (jusqu'à 8 bacs de 16 cartes ; 16 voies par carte) et une console de programmation. Divers coupleurs peuvent être adjoints pour la gestion de périphériques divers.

Une première différence essentielle concerne les cartes d'entrées-sorties : la large gamme permet d'accommoder tous les cas particuliers (voir figure 3). Toutes les cartes sont équipées de *visualisation* de l'état de chaque

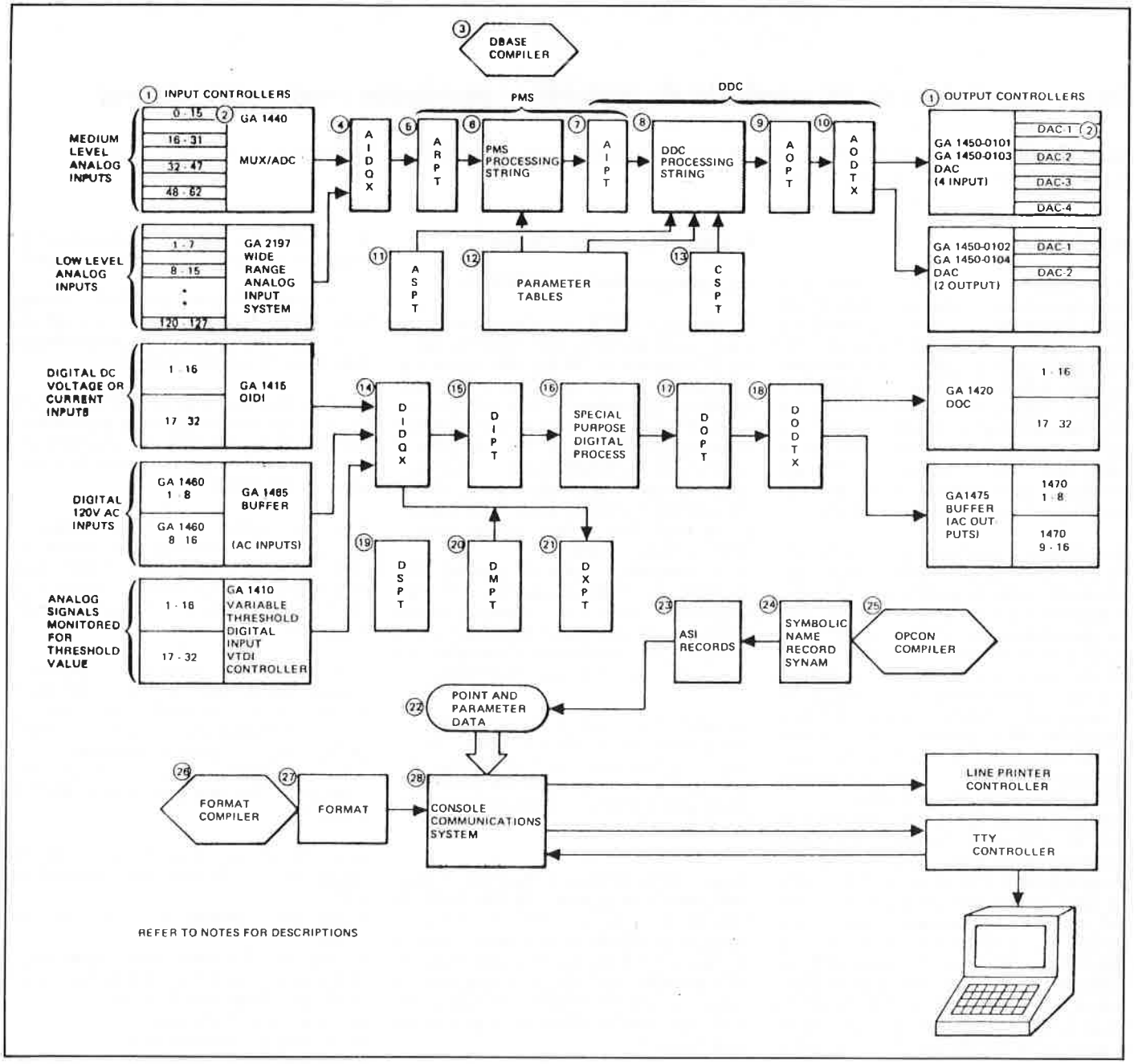


Figure 1. Constitution et fonctionnement de TOPICS.

voie et d'une prise permettant la mise en place d'une « *carte test* » bien utile pour vérifier le câblage, mettre au point une partie de programme sans le procédé, ou pal'ier la défaillance d'un capteur TOR. Le raccordement de chaque voie se fait par 2 fils à câbler sur des bornes à vis, de diamètre de serrage fonction de la puissance. Filtrage symétrique et isolement galvanique participent bien sûr au caractère industriel des entrées-sorties.

Mais c'est le langage qui distingue le plus l'automate programmable : plus de système temps réel à « générer », plus de compilateur, plus de disque. A la place, *un catalogue d'ordres* simples (logique combinatoire et séquentielle) ou puissants (calcul simple ou double précision, linéarisation et PID, messages, etc...) que *l'homme de l'art* (non informaticien) *compose* au clavier de la console de programmation, en une suite d'instructions remplissant les fonctions de logique ou de régulation. Il peut ensuite modifier une ligne ou une partie de son programme de façon interactive, sans avoir besoin ni de « recompiler », ni même d'arrêter *l'exécution du reste du programme.*

La figure 4 illustre la facilité de programmation. Soulignons simplement qu'il n'y a plus à se soucier de l'adressage et de la lecture d'une voie, et que la 8e voie raccordée sur une carte occupant l'emplacement 3 du premier bac a pour adresse 138, qu'elle soit logique (accès par l'ordre SI 138) ou analogique (accès par l'ordre APL 138). La simplicité avec laquelle il est possible de faire appel en nombre illimité à des fonctions complexes telles que files d'attente, temporisations, retards, compteurs, variables numériques, messages, linéarisations ou PID, permet un mélange aisé de logique, de calcul, d'édition et de régulation, donc une programmation évidente à partir de n'importe quel schéma.

II. Mise en œuvre

Trois étapes méritent d'être distinguées dans la mise en œuvre d'une conduite de procédé : le raccordement, la programmation du schéma et le démarrage de l'application.

II.1. Raccordement au procédé

Presque immédiat pour l'automate pourvu que les cartes correspondent bien aux si-

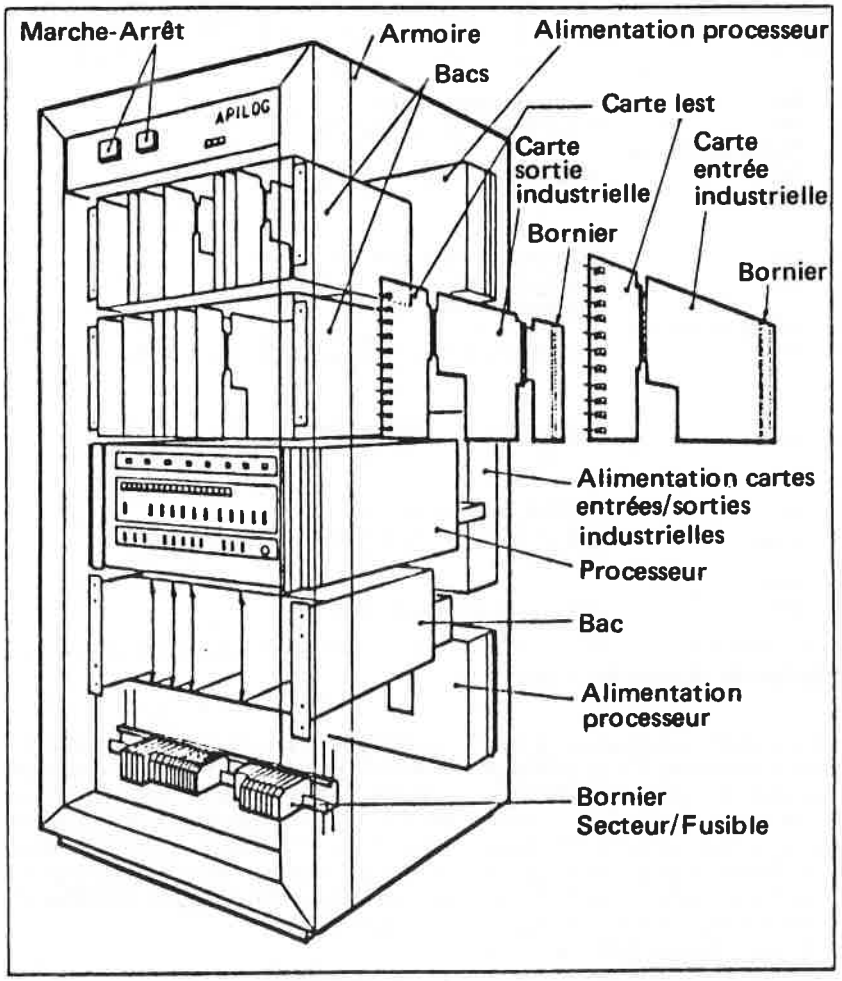


Figure 2. Configuration de l'armoire APILOG utilisée.

gnaux, le raccordement des capteurs ne pose plus de problème majeur à un calculateur bien choisi qui accepte tous les niveaux présents dans une application industrielle. Encore, ne faut-il pas, sous peine de remaniement sérieux du système d'entrée-sortie, acquérir la moitié des voies analogiques en analogique et l'autre moitié à travers un système externe de régulation numérique !

Il n'en est pas de même du raccordement des stations de transfert ou régulateurs de secours dont le mode de dialogue (analogique ou numérique, liaisons individuelles ou banalisées, etc...) n'est jamais supporté de manière standard par le calculateur. Et il est bien plus aisé d'avoir un accès direct au niveau de chaque bit de l'échange par les ordres simples d'un automate, que d'écrire et de mettre au point un driver spécifique compatible avec le système informatique.

II.2. Programmation du schéma

Nous avons dit plus haut que la construction des diverses tables de TOPICS nécessitait la compilation par 3 compilateurs différents d'ordres sources écrits dans un langage spécifique dont il faut apprendre grammaire et syntaxe. Le processus de compilation étant extrêmement complexe, *le temps de traitement est imposant :* on peut estimer à 1/2 journée de travail d'un opérateur averti le temps nécessaire à l'introduction d'une boucle de régulation avec toutes les liaisons la rendant accessible au dialogue ; ceci sans tenir compte de la virgule oubliée dont l'effet ne sera perçu qu'à la fin.
Pour illustrer la facilité de programmation avec l'automate APILOG, considérons la boucle complexe (bien que déjà simplifiée) de régulation du niveau de verre :

Le mélange est introduit de chaque côté du four par 2 enfourneuses de vitesses V10 et V20 (sorties analogiques 040 et 050). Une mesure de niveau No (entrée analogique 010) est comparée à la consigne V0 : l'écart doit jouer sur la somme des vitesses V10 + V20. Les températures de part et d'autre du four doivent se maintenir dans un écart constant : toute variation de la différence de température T10-T20 (entrées analogiques 011 et 012) doit agir sur la différence des vitesses V10-V20.

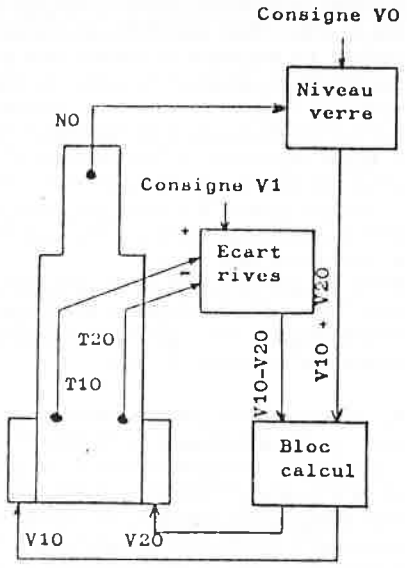
La partie droite de la figure montre la programmation APILOG correspondante utilisant l'ordre PID certains ordres de calcul et de lecture-écriture analogique (la partie filtrage et linéarisation n'est pas représentée ; il conviendrait d'ajouter un ordre par fonction et par voie de mesure).

II.3. Démarrage de l'application

Le procédé étant continu, il faut que sa conduite soit permanente. Chacun sait bien qu'il n'est pas évident de faire fonctionner 24 h sur 24 un logiciel temps réel : il faut régler les uns après les autres tous les conflits entre tâches, qui apparaissent à intervalles de durée croissante avec le degré de mise au point. *L'automate assure au contraire un démarrage immédiat* puisque tout ordre composé de manière interactive n'est accepté que s'il est correct. Cette caractéristique importante permet de ne s'intéresser qu'à la conduite du procédé en oubliant totalement son support d'exécution.

	TENSION	CONTACT	PUISSANCE	ANALOGIQUE	SPECIALE
ENTREES	. 24 V . 5 → 48 V } 5 mA	Idem tension mais Apilog alimente le contact	. 110 V ~ . 220 V ~ } statique . 50 V =	16 voies par carte 11 bits + signe 40 mV → 10V 4 → 20 mA TMC = 100 V	Comptage rapide 20 KHz Positionnement numérique
SORTIES	Idem contact mais 1 seule alimentation externe à câbler par carte de 16	. Relais Reed 24V 100 mA . Relais Siemens 48V → 1A . Statique 50V 40mA . statique 60V 1A	. Relais 220V~ 5A . 250V~4A } statique . 55V=4A	1 voie par carte 0-10V ou -5+5V 0-20 mA	Communication inter-automates Couplage bac à distance

Figure 3. Les cartes d'entrées-sorties APILOG.




```
APL  010          lecture mesure NO
SUB  V0           - consigne de niveau V0
PID  Po,kg,kp,ki,kd
RGT  V100         rangement action (V10 + V20)/2 dans V100

APL  011          lecture mesure T10
SUB  012          - mesure T20
SUB  V1           - consigne V1
PID  P1,kg,kp,ki,kd
RGT  V101         rangement action (V10 - V20)/2 dans V101

APL  V100         (V10 + V20)/2 - (V10 - V20)/2 = V20
SUB  V101
RGT  050          sortie analogique 050
APL  V100         (V10 + V20)/2 + (V10 - V20)/2 = V10
ADD  V101
RGT  040          sortie analogique 040
```

Figure 4. Schéma et programmation de la régulation du niveau de verre.

Dans les 2 approches, les mises au point des cadences de traitement, des divers coefficients de régulation et de linéarisation, se font avec la même facilité, mais l'automate prend encore l'avantage par la souplesse qu'il offre pour les *modifications en ligne* du schéma de régulation ou pour l'introduction de blocs de traitement non prévus dans le logiciel du calculateur (retards, blocages, conditions logiques, etc...).

III. Exploitation

Qu'ils concernent les arrêts, les démarrages, les changements de fabrication ou l'ajustement des paramètres de conduite, les besoins de l'exploitation sont satisfaits aussi bien par l'automate que par le calculateur. Console de mise au point pour l'automate, ou dialogue console pour le calculateur, sont 2 moyens identiques pour la modification d'une ou de plusieurs variables. Quant aux arrêts et démarrages, ils nécessitent un mélange de séquentiel et de régulation mieux appréhendé par l'automate.

Le seule domaine où le calculateur affirme sa supériorité est celui du *format* auquel obéit le dialogue machine-exploitant, lorsque celui-ci a lieu à travers des moyens informatiques (console écran clavier par exemple) ; il autorise le raccordement d'une gamme importante de périphériques, et des langages évolués permettent une gestion facile des échanges, au prix néanmoins d'un effort non négligeable de développement de logiciel.

Il est pourtant presque certain (c'est en tout cas notre avis à la lumière de nos multiples installations) que la console « informatique » avec son clavier de machine à écrire n'est pas un bon support pour les besoins de l'exploitation. Il faut résister à la tentation d'un investissement faible et préférer un pupitre d'exploitation donnant aux opérateurs une vision « en parallèle » et non sérialisée des paramètres importants du procédé qu'ils conduisent, ce pupitre peut alors être raccordé très facilement à l'automate, qui dispose de tous les ordres pour le gérer (impulsion, transcodages, etc.).

Quant aux besoins d'exploitation différée comme le journal de bord ou les historiques qui peuvent être traités par logiciel dans le calculateur, nous sommes convaincus, en tant qu'installateurs devant faire face à des demandes toujours différentes, que la solution passe par le raccordement d'une console « intelligente » à un automate comme APILOG : elle reçoit et transmet des données sur une liaison asynchrone et décharge l'automate de toutes les fonctions annexes, qu'elle traite dans un langage évolué. Plusieurs réalisations récentes, dans d'autres domaines, nous ont montré toute la puissance de cette approche.

IV. Conclusion

En guise de conclusion, nous nous bornerons à établir une comparaison du prix des matériels et des temps passés en programmation de schéma et en démarrage, dans 2 applications strictement identiques de conduite de four de verre textile, l'une réalisée par calculateur + TOPICS, l'autre par APILOG. Chacune comporte 32 voies d'acquisition, 16 sorties sur station de transfert et 34 régulations imbriquées dans un schéma très complexe.

	Matériel FHT	Programmation du schéma (heures)	Démarrage (heures)
Calculateur	170 000	600	300
APILOG	90 000	300	200

Figure 5. Comparaison calculateur-automate APILOG.