

## Systèmes numériques de contrôle commande de procédé

8.1 De la régulation analogique à la régulation numérique.....	215
8.2 Introduction aux systèmes numériques.....	219
8.3 Aspect matériel d'un système numérique.....	224
8.4 Aspect logiciel d'un système numérique.....	232
8.5 Différents types de systèmes numériques de contrôle de procédé.....	235
8.6 Influence du traitement numérique sur la régulation.....	240

## NOTATIONS UTILISÉES DANS LE CHAPITRE 8

Bd	: Baud (unité de débit numérique).
C	: Consigne locale.
CAN	: Convertisseur analogique numérique.
CNA	: Convertisseur numérique analogique.
Im	: Intensité modulée.
k	: Kilo (multiple d'unité informatique : $2^{10}$ )
LSB	: Bit de poids le plus faible.
MSB	: Bit de poids le plus fort.
M	: Signal de mesure.
Ma <sub>(t)</sub>	: Signal de mesure analogique.
Md <sub>(t)</sub>	: Signal de mesure discret
Mn <sub>(t)</sub>	: Information de mesure discrète.
p	: Opérateur de Laplace.
q	: Quantum.
r	: Résolution.
RS	: Liaisons séries normalisées. (232C, 422, ...).
S, S <sub>(t)</sub>	: Signal de sortie.
Sa <sub>(t)</sub>	: Signal de sortie analogique.
Sd <sub>(t)</sub>	: Signal de mesure discret.
Sn <sub>(t)</sub>	: Information de sortie discrète.
T	: Transmetteur.
T <sub>c</sub>	: Période de calcul ou temps de cycle de calcul.
T <sub>e</sub>	: Période d'échantillonnage.
T <sub>l</sub>	: Temps libre.
T <sub>t</sub>	: Temps de traitement ou temps d'exécution d'un programme.
τ	: Temps mort d'un procédé.
θ	: Constante de temps d'un procédé.

La micro-électronique a permis la réalisation de composants intégrant des fonctions multiples et différentes.

Parmi ces composants, le microprocesseur ( $\mu P$ ) joue un rôle fondamental dans la gestion d'informations numériques. Il a besoin, pour fonctionner, d'un environnement utilisant d'autres composants tels que les mémoires intégrées et les circuits d'interface. Ces composants entrent dans la fabrication d'éléments (unité de calcul, clavier alphanumérique, moniteur vidéo, lecteur de disques, ...) qui constituent l'architecture des calculateurs numériques.

Ces derniers, souvent utilisés pour des tâches de gestion et de supervision, trouvent aujourd'hui leur place dans le domaine du contrôle de procédés ; on les appelle dans ce cas des systèmes numériques de contrôle commande. Ils supplantent le matériel analogique et imposent au technicien de régulation de posséder des connaissances en numérique. Ce chapitre a donc pour objectifs :

- De rendre plus abordable le vocabulaire employé en informatique industrielle.
- De développer les concepts matériel et logiciel (langages, programmation, ...) des systèmes numériques.
- De faire le point sur le matériel numérique.
- D'aborder les problèmes de traitement numérique, liés au contrôle de procédés.

### 8.1 DE LA RÉGULATION ANALOGIQUE À LA RÉGULATION NUMÉRIQUE

Ce paragraphe donne une présentation générale du traitement numérique dans un calculateur. Une étude plus détaillée est développée au paragraphe 8.6.

En régulation analogique les signaux évoluent d'une façon continue dans le temps.

Actuellement les systèmes numériques utilisent encore des signaux d'entrées analogiques. Ceci nécessite d'échantillonner ces signaux à intervalles de temps régulier  $T_e$ , on obtient ainsi une suite de valeurs dites discrètes (fig. 8.1).

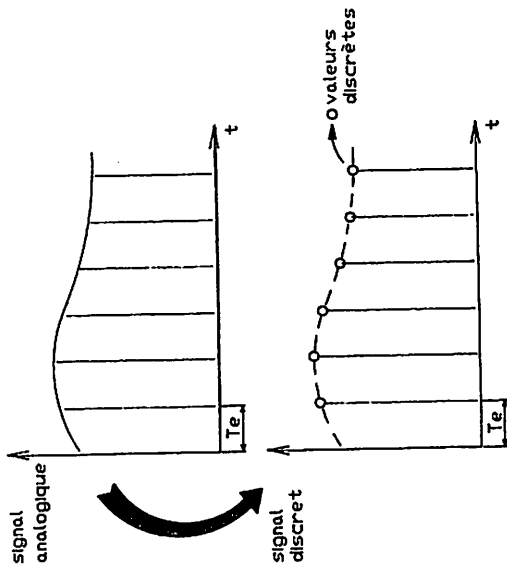


Fig. 8.1 : Échantillonnage d'un signal.

Chaque valeur discrète est convertie en une valeur numérique suivant un code prédéterminé. Cette valeur numérique ou digitale est ensuite stockée en mémoire en vue d'un traitement ultérieur ; elle se présente sous l'aspect d'une succession d'états électriques qui traduisent des états binaires (présence de tension = état à 1, absence de tension = état à 0). Cet état binaire représente l'élément de base de l'information numérique appelé bit.

La figure 8.2 montre l'élaboration de l'information numérique (codée sur 8 bits) à partir d'un signal de mesure analogique d'un transmetteur de pression.

La figure 8.2 (a) représente l'évolution de la grandeur physique pression.

La figure 8.2 (b) représente le signal analogique 4 à 20 mA délivré par le transmetteur. Ce signal converti en volts est envoyé sur une des voies, (V1 à V8) d'un multiplexeur qui assure une sélection périodique de la voie concernée (V3). Cette sélection correspond à la période d'échantillonnage  $T_e$ .

La figure 8.2 (c) représente les valeurs discrètes du signal analogique, prélevées à chaque période, qui seront converties en informations numériques.

La figure 8.2 (d) représente deux valeurs numériques (mots binaires, ex. : 8 bits) correspondant à la conversion de deux valeurs discrètes. Cette fonction est assurée par un convertisseur analogique numérique CAN. Les informations numériques successives sont véhiculées par des lignes parallèles (bus) pour être stockées dans les mémoires.

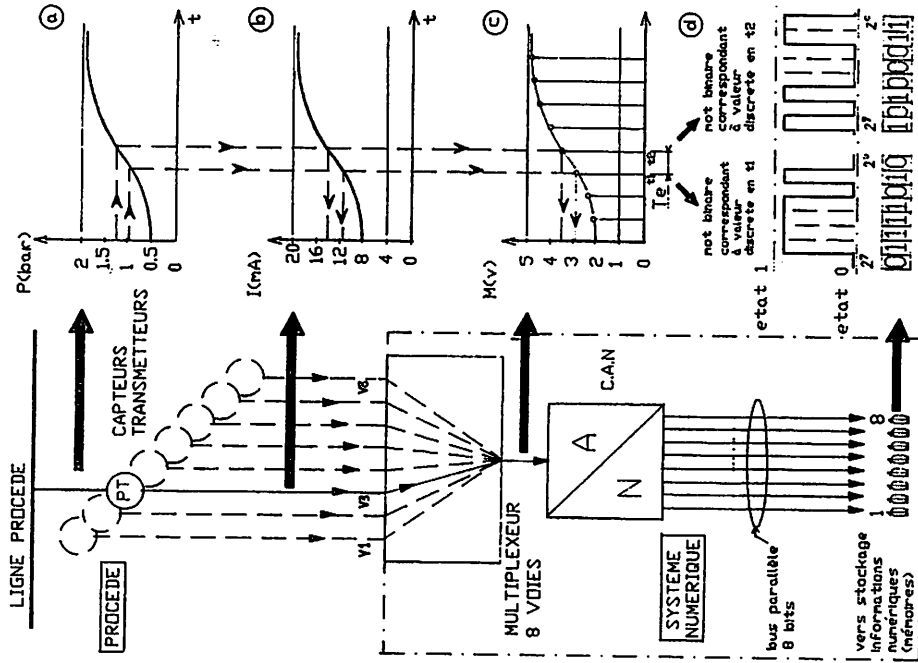


Fig. 8.2 : Conversion d'un signal analogique.

Que deviennent les informations numériques correspondantes au signal d'entrée ? La figure 8.3 (a) montre l'évolution de ces informations en prenant l'exemple d'un régulateur numérique. De par sa technologie, ce régulateur ainsi que tout système numérique ne peut traiter qu'une opération à la fois, ceci implique donc une sortie calculée à étapes périodiques. Cette sortie se concrétise par une suite de valeurs discrètes.

La figure 8.3 (b) permet de comparer les signaux d'un régulateur analogique et d'un régulateur numérique.

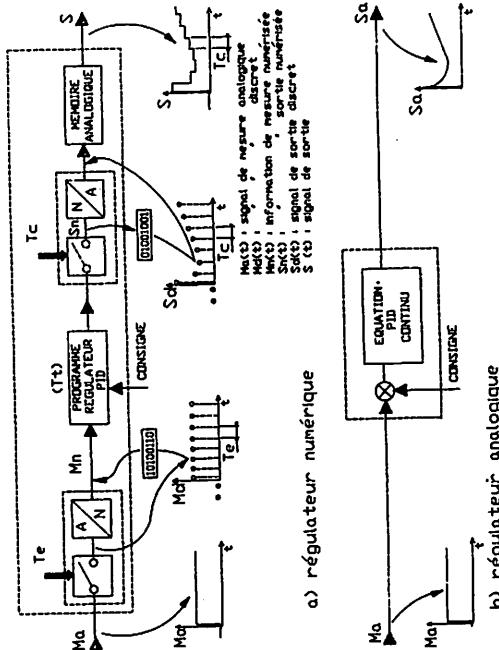


Fig. 8.3.

La procédure du traitement numérique est la suivante :

Après conversion analogique numérique, les valeurs numériques  $M_n$  servent au calcul répétitif du signal de sortie. Ce calcul est réalisé par un algorithme PID programmé. Les valeurs numériques de sortie  $S_n$ , résultats du calcul, sont traduites en valeurs discrètes  $S_d$  à l'aide d'un convertisseur numérique analogique CNA. Le régulateur doit commander un organe de réglage qui exige un signal continu, les valeurs discrètes  $S_d$  sont donc traitées par une mémoire analogique. Cette mémoire permet le maintien de la dernière valeur discrète pendant la période  $T_e$  afin d'éviter le retour à 0 de l'organe de réglage ; il résulte alors un signal de sortie  $S$  en escalier pratiquement assimilable au signal analogique  $S_a$ .

L'échantillonnage du signal d'entrée et la réactualisation du signal de sortie ne s'effectuent pas de façon synchrone ; ils sont décalés au minimum d'un temps qui résulte du temps de conversion du CAN, ainsi que du temps d'exécution  $T_1$  du programme du régulateur numérique (alarmes, calcul de l'écart, calcul PID, ...).

Sur certains régulateurs ou systèmes numériques, la période de réactualisation  $T_c$  du signal de sortie, peut-être différente de  $T_e$ . Le temps  $T_c$  correspond à la période de renouvellement de l'exécution du programme (fig. 8.4). Chaque valeur de sortie réactualisée correspond au résultat du dernier calcul de l'algorithme.

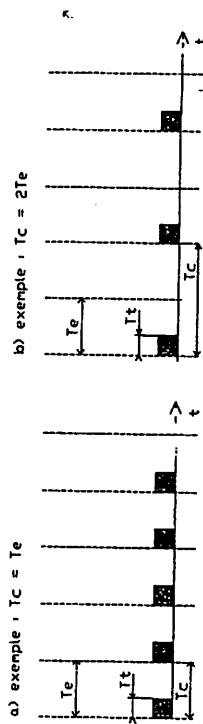


Fig. 8.4 : Traitement numérique : décomposition des temps.

Citons quelques avantages et inconvénients de la technologie numérique. Parmi les avantages :

- Les signaux numériques se prêtent bien aux calculs, sont peu sensibles aux bruits et faciles à mémoriser.
- L'intégration de fonctions analogiques en tant que logiciels a permis de réduire les coûts d'équipement et de câblage.
- L'intégration de nouvelles fonctions est possible par programmation.
- De nouvelles fonctions inexistantes en analogique sont aujourd'hui réalisables (ex. : PID auto-adaptatif).
- Les calculs sont plus précis qu'en analogique, la précision ne se dégrade pas.
- Les instruments numériques offrent une souplesse et une rapidité de réglage importantes.
- Les lignes de communications numériques permettent le transport d'un grand nombre de données à distance, ce qui rend possible la supervision centralisée.

L'utilisation de la technologie numérique présente toutefois certains inconvénients :

- La mise en œuvre nécessite de nouvelles compétences pour l'utilisateur.
- Dans la mise au point des boucles de régulation numériques, il faut tenir compte de nouveaux paramètres tels que les temps  $T_e$  et  $T_c$ .

## 8.2 INTRODUCTION AUX SYSTÈMES NUMÉRIQUES

Avant d'étudier la structure fonctionnelle d'un ordinateur, analysons les diverses formes d'informations numériques qui sont exploitées à l'intérieur.

### 8.2.1 INFORMATIONS NUMÉRIQUES

La communication avec le milieu extérieur d'un système numérique nécessite l'utilisation de caractères alphanumériques : chiffres, lettres et caractères symboliques (\*, F3, ESC, Ctrl, #, \$, à, ...).

Le milieu extérieur émet ou reçoit également des signaux analogiques et des états logiques.

Les signaux, états et caractères alphanumériques sont traduits en mots binaires grâce à des codes pour être traités par le système (fig. 8.5).

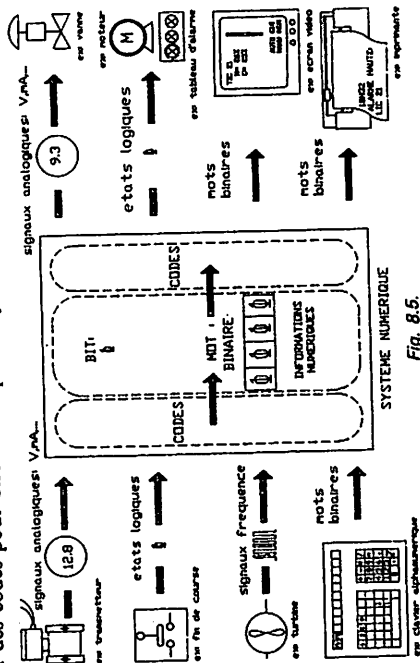


Fig. 8.5.

La longueur d'un mot binaire constitué de 8 bits est appelé octet, ce terme est pris comme unité informatique. Le préfixe kilo ( $1 \text{ kilo} = 2^{10} = 1024$ ) est souvent associé à l'octet pour traduire des capacités de stockage pour les mémoires du calculateur (fig. 8.6). La taille d'un mot (8, 16, 32 bits) donne une image de la puissance d'un ordinateur ; ainsi un ordinateur de 16 bits est un système numérique qui traite des données numériques codées sur 16 bits. Ceci signifie que les mémoires de cet ordinateur stockent ou dérivent des mots de 16 bits.

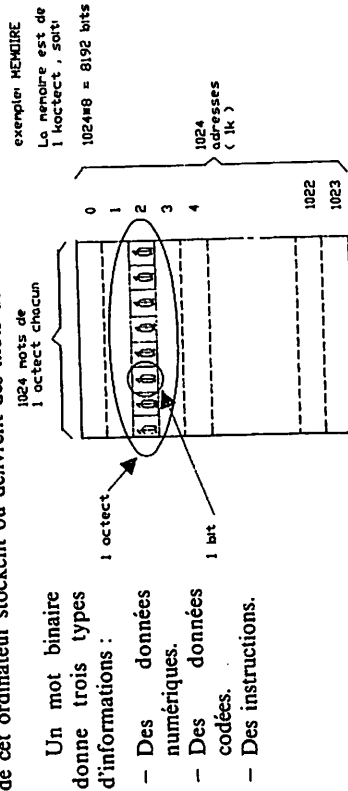


Fig. 8.6 : Définition : mot, octet, bit.

- Un mot binaire donne trois types d'informations :
- Des données numériques.
  - Des données codées.
  - Des instructions.

### 8.211 Données numériques en binaire pur

Ce sont des informations utilisées pour représenter des valeurs analogiques ou des états logiques. Le code binaire pur est souvent utilisé pour traduire une valeur analogique en binaire, le mot binaire résultant est ensuite utilisé dans le calculateur. Le tableau 8.1 donne la correspondance décimal / binaire pur sur mots de 8 bits.

Tableau 8.1 : Conversion décimal-binaire.

Décimal	Équivalent binaire (poids binaires)							
	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	0
3	0	0	0	0	0	1	0	1
...	-	-	-	-	-	-	-	-
127	0	1	1	1	1	1	1	1
...	-	-	-	-	-	-	-	-
255	1	1	1	1	1	1	1	1

La figure 8.7 représente la conversion d'un signal analogique 4 à 20 mA en un mot de 8 bits codé en binaire pur. Une conversion sur n bits, donne  $2^n$  mots. Le bit de poids le plus faible ( $2^0$ ) s'appelle le LSB, celui de poids le plus fort ( $2^7$ ) le MSB.

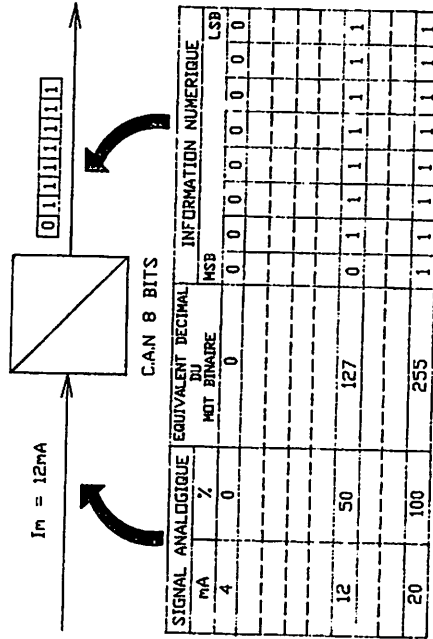


Fig. 8.7 : Conversion d'un signal 4 à 20 mA en binaire.

### Exemple :

Si  $n = 8$ , il existe  $2^8 = 256$  mots de 8 bits. Pour la conversion on utilise le tableau 8.1.  $3$  en décimal  $\rightarrow 00000011$  en binaire.  $3 = 2^1 + 2^0$ .

### 8.212 Données codées

Un système numérique doit être capable de gérer des informations alpha-numériques, ce qui nécessite un code ; le plus répandu est le code ASCII (Tableau 8.2). Ce code est utilisé dans la transmission d'informations entre le système numérique et les unités entrées / sorties telles qu'imprimante, clavier et écran de visualisation.

La figure 8.8 montre la traduction d'un début de texte sur un écran de visualisation.

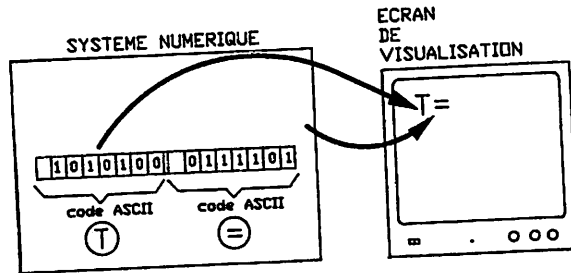


Fig. 8.8.

Tableau 8.2 : Liste partielle du code ASCII.

Caractères	Code ASCII	Caractères	Code ASCII
A	100 0001	0	011 0000
B	100 0010	1	011 0001
C	100 0011	2	011 0010
D	100 0100	3	011 0011
E	100 0101	4	011 0100
F	100 0110	5	011 0101
G	100 0111	6	011 0110
H	100 1000	7	011 0111
I	100 1001	8	011 1000
J	100 1010	9	011 1001
K	100 1011	Espace	010 0000
L	100 1100	.	010 1110
M	100 1101	(	010 1000
N	100 1110	+	010 1011
O	100 1111	\$	010 0100
P	101 0000	*	010 1010
Q	101 0001	)	010 1001
R	101 0010	-	010 1101
S	101 0011	/	010 1111
T	101 0100	=	011 1101
...	.....	...	.....

### 8.213 Instructions

Les instructions se décomposent en deux parties, qui sont :

- Le code opération, qui définit le type d'opération à réaliser sur une donnée.
- L'adresse dans la mémoire où réside la donnée sur laquelle se réalise l'opération.

Les formats et codes de ces instructions, sont variables d'un système à l'autre.

## 8.22 STRUCTURE FONCTIONNELLE D'UN SYSTÈME NUMÉRIQUE

Les calculateurs, quelle que soit leur taille ou leur puissance, utilisent des processeurs câblés (circuits logiques câblés) ou des microprocesseurs (circuits intégrés). Ces circuits opèrent de façon séquentielle sur des mots binaires (données numériques, données codées, instructions). Chaque séquence s'exécute au rythme d'une horloge. Pour prélever les informations, les traiter, les stocker, tous les calculateurs numériques réalisent trois fonctions qui sont :

- Une fonction mémoire.
- Une fonction exécution.
- Une fonction de tâches spécifiques.

Les mémoires peuvent stocker plusieurs milliers d'informations numériques telles que des données numériques, des données codées et des instructions d'exécution de programmes.

L'exécution automatique de programme permet de prélever une à une dans la mémoire les instructions de ce programme et de les exécuter. Le programme de calcul de l'algorithme d'un régulateur est un exemple type d'une exécution automatique.

Les tâches spécifiques donnent au calculateur la possibilité de se dérouter en cours d'exécution de programme, afin d'exécuter des tâches prioritaires (alarmes, ...).

Pour réaliser les fonctions précédentes, le système numérique a besoin d'une architecture matérielle qui se compose de trois parties principales représentées par la figure 8.9.

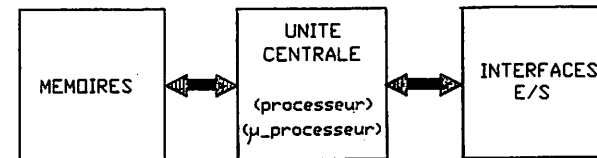


Fig. 8.9 : Parties principales d'un calculateur numérique.

### 8.3 ASPECT MATÉRIEL D'UN SYSTÈME NUMÉRIQUE

Tous les micro-ordinateurs présentent des architectures matérielles (Hardware) proches les unes des autres. Les systèmes numériques qui sont utilisés en mesure et régulation, ont des interfaces de procédé. La figure 8.10 donne l'architecture type d'un micro-ordinateur. Le rôle des différents éléments, repérés de 1 à 10, est décrit par la suite.

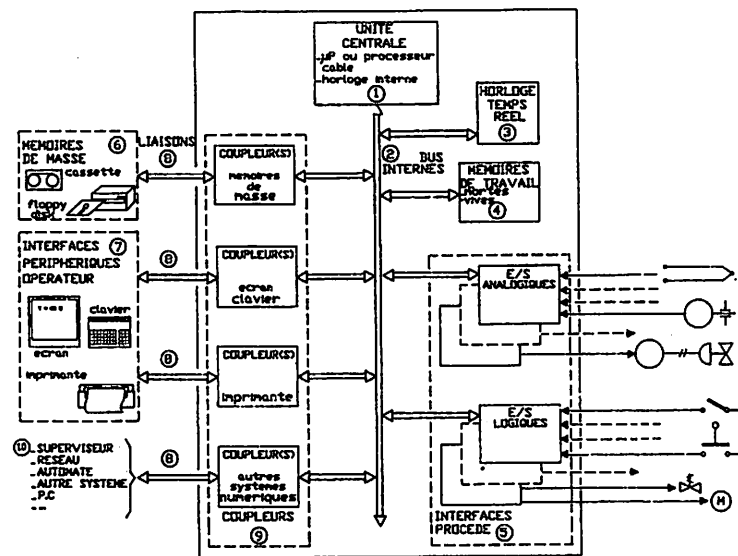


Fig. 8.10 : Architecture type d'un micro-ordinateur orienté procédé.

#### 8.31 UNITÉ CENTRALE

L'unité centrale (élément 1, fig. 8.10) est le cerveau du micro-ordinateur, elle doit :

- Commander et synchroniser toutes les tâches des éléments du micro-ordinateur.
- Rechercher en mémoire les mots d'instructions et de données.
- Transmettre et recevoir des données par les interfaces E et S.
- Décoder les instructions.
- Effectuer les opérations arithmétiques et logiques.
- Gérer les commandes pour les périphériques E et S.
- .....

Toutes ces fonctions s'exécutent au rythme de l'horloge interne qui peut battre à plus de 1 million de cycles par seconde (suivant le type de micro-processeur). Une instruction ou séquence peut nécessiter de 2 à 20 cycles d'horloge.

Pour échanger des informations avec les différents éléments du micro-ordinateur, l'unité centrale utilise des liaisons électriques appelées bus internes (2). On distingue :

- Le bus de contrôle (control bus).
- Le bus d'adresse (adress bus).
- Le bus des données (data bus).

Pour gérer les interfaces d'entrée et de sortie, on utilise une horloge dite temps réel (élément 3, fig. 8.10).

#### 8.32 MÉMOIRES

On distingue deux types de mémoires :

- Les mémoires de travail.
- Les mémoires de masse.

##### 8.321 Mémoires de travail

Parmi les mémoires de travail (élément 4, figure 8.10), on rencontre les mémoires dites mortes et les mémoires dites vives. Ce sont principalement des mémoires à circuits intégrés qui regroupent des informations destinées à l'exécution des programmes (données, instructions).

Les mémoires mortes (fig. 8.11) conservent leurs informations lors d'une coupure d'alimentation électrique, elles sont dites non volatiles. Elles contiennent des informations telles que :

- Instructions de programme (ex. : algorithme régulateur PID).
- Constantes (ex. : nombre  $e = 2,718$ ).
- Code de conversion (ex. : table ASCII).

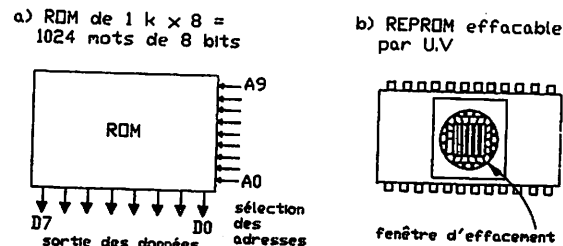


Fig. 8.11 : Mémoires mortes.

Les plus utilisées sont :

- Les mémoires ROM à lecture uniquement et ineffaçables.
- Les mémoires EPROM reprogrammables et effaçables. Elles s'effacent électriquement (EAROM) ou par rayonnement ultraviolet (REPRUM). On les utilise par exemple pour le stockage d'un programme spécifique.
- Les mémoires vives (fig. 8.12), peuvent être lues et écrites. Elles sont destinées à recevoir et à délivrer des informations telles que :
  - Valeurs de mesure.
  - Valeurs de consigne.
  - Valeurs d'actions PID
  - Résultats de calculs.

Les mémoires vives les plus utilisées sont les RAM à circuits intégrés. Elles sont volatiles ce qui nécessite une sauvegarde par une alimentation de secours (batterie).

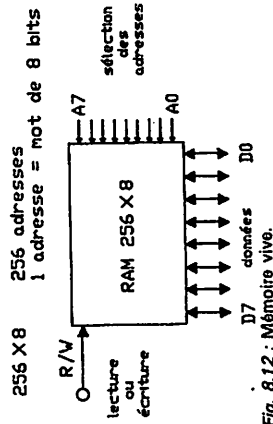


Fig. 8.12 : Mémoire vive.

La mémoire de travail d'un système numérique utilise plusieurs circuits intégrés. La capacité d'une mémoire s'exprime généralement en Kilo-octets (Koctet) ou Mega-octets (Mocet).

Par exemple :

- Un circuit intégré RAM de 4 Koctets a une capacité de 4 096 bits.
- La capacité d'une mémoire vive d'un système numérique est de 1 Mocet ce qui nécessite plusieurs circuits intégrés.

En comparaison, l'espace mémoire nécessaire pour traiter l'algorithme d'un régulateur est de l'ordre de 400 octets, celui d'une vue graphique est de l'ordre de 2 Koctets, alors qu'une valeur analogique nécessite de 3 à 6 octets.

### 8.322 Mémoires de masse

Ces mémoires (élément 6, fig. 8.10) qui permettent de stocker une grande quantité de données, sont utilisées occasionnellement, par exemple lors de la sauvegarde d'un programme. Parmi ces mémoires citons :

- Les cassettes magnétiques.
- Les disquettes ou disques souples (FLOPPY DISK).
- Les mémoires à bulles à circuits intégrés.
- Les disques durs.

Les deux premiers cités sont fréquemment utilisés pour les systèmes de contrôle de procédés.

La cassette magnétique (fig. 8.13 (a)) peut enregistrer des informations déposées en série (blocs sériels). Elle a une capacité mémoire de l'ordre de 5 Mocets. L'accès aux informations est séquentiel, ce qui nécessite un temps d'accès important. Une des méthodes d'enregistrement utilisée est basée sur la modulation de fréquence (fig. 8.13 (b)).

Les disques souples ont un diamètre de 130 mm (5 pouces 1/4) ou de 200 mm (8 pouces). Celui de 130 mm par exemple est organisé en 77 pistes et 26 secteurs par piste. Chaque secteur stocke 128 ou 256 octets d'information (simple ou double densité), ce qui conduit à des capacités de 256 Koctets à 1 Mocets (fig. 8.13 (c et d)).

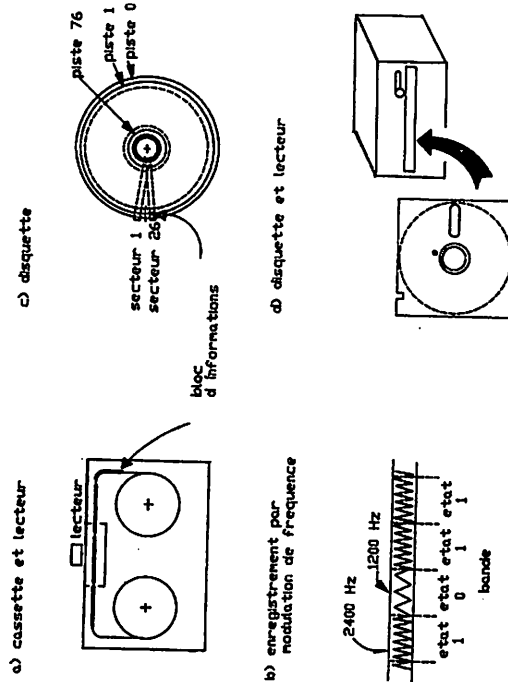


Fig. 8.13 : Mémoires de masse.

### 8.33 INTERFACES DE PROCÉDÉS

Les interfaces de procédés (élément 5, fig. 8.10) sont des circuits qui génèrent ou reçoivent des signaux analogiques et logiques d'un procédé.

La figure 8.14 montre différents types d'interfaces procédés.

On distingue :

- Les cartes ou modules d'entrée et sortie analogiques.
- Les cartes ou modules d'entrée et sortie logiques.



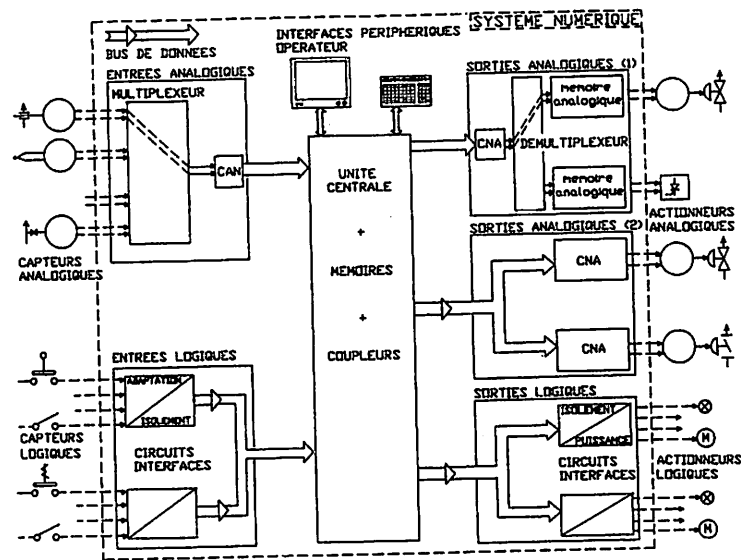


Fig. 8.14 : Interfaces de procédés.

Il existe également d'autres types d'interfaces :

- Carte d'entrée fréquence (turbine, vortex, ...).
- Carte de sortie impulsionnelle (commande triac, ...).

### 8.331 Interfaces d'entrée analogique, résolution, quantum

Le principe de la conversion analogique numérique a été étudié au paragraphe 8.1 ; nous complétons cette étude par des notions sur la résolution, le quantum et le temps de conversion.

#### Résolution et quantum

La figure 8.15 représente un exemple de conversion en binaire pur sous 8 bits. Le quantum  $q$  représente la plus petite variation du signal d'entrée qui entraîne le changement d'état du bit de poids le plus faible. Le quantum s'exprime en unités physiques. Il est donné dans notre exemple par la relation :

$$q = V_m \cdot r$$

La résolution  $r$ , exprimée en %, est donnée par :

$$r = \frac{1}{2^n} \cdot 100$$

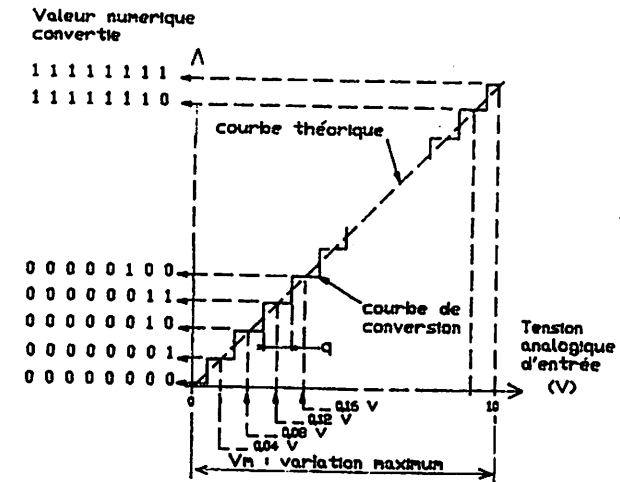


Fig. 8.15 : Courbe de conversion A/N sur 8 bits.

Exemple :

Dans la figure 8.15, la valeur de  $r$  est :

$$r = 1/2^8 \cdot 100 = 0,0039 \cdot 100$$

$$r = 0,4 \%$$

Avec  $V_m = 10 \text{ V}$ , le quantum  $q$  est :

$$q = 10,4/100 = 0,04 \text{ V}$$

$$q = 0,04 \text{ V}$$

Remarque : Les conversions s'effectuent généralement sur 10 bits ( $r = 0,1 \%$ ). De ce fait l'erreur de résolution est faible comparée à l'erreur de précision des capteurs industriels.

#### Temps de conversion

Notons que la commande et le contrôle des circuits de l'interface d'entrée (multiplexeur, CAN, ...) sont effectués par l'unité centrale. Le temps de conversion du convertisseur analogique numérique dépend de sa technologie. Ce temps de conversion (ex.  $100 \mu\text{s}$ ) est faible par rapport à la période d'échantillonnage  $T_e$  (ex. :  $T_e = 500 \text{ ms}$ ).

### 8.332 Interfaces de sortie analogique, résolution

La figure 8.14 montre deux modes de conversion numérique analogique, les systèmes numériques sont équipés de l'un ou de l'autre.

Le premier mode (élément 1, fig. 8.14) est réalisé à l'aide d'un CNA unique et d'un démultiplexeur. Une mémoire analogique par voie est nécessaire pour maintenir constant le signal de sortie lors de la conversion sur les autres voies.

Le deuxième mode (élément 2, fig. 8.14) utilise un CNA pour chaque voie de sortie.

La figure 8.16 représente une conversion numérique analogique sur 8 bits.

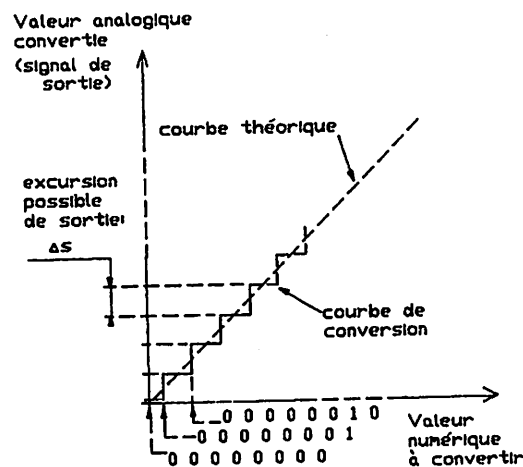


Fig. 8.16 : Courbe de conversion N/A sur 8 bits.

La variation minimale  $\Delta S$  du signal de sortie dépend de la résolution du CNA, celle-ci s'exprime en % par la relation :

$$r = \frac{1}{2^n - 1} \cdot 100$$

### 8.333 Interfaces d'entrée et sortie logiques

Les signaux d'entrée et de sortie sont des états logiques (0 et 1) identiques à ceux des états logiques internes du système numérique. Des interfaces sont néanmoins nécessaires pour adapter les niveaux de tension et assurer les fonctions d'isolement et de puissance (sorties).

### 8.34 INTERFACES PÉRIPHÉRIQUES OPÉRATEUR

Les interfaces périphériques opérateur (éléments 7, fig. 8.10) permettent le dialogue entre l'opérateur et le système numérique (dialogue homme-machine). Parmi ces interfaces citons :

- Les écrans de visualisation.
- Les claviers destinés à la conduite du procédé.
- Les claviers de programmation ou de configuration.

- Lecteurs de cassettes magnétiques ou de disques souples.
- Imprimantes d'édition (messages, alarmes, ...).

Ces interfaces sont en relation avec l'unité centrale, grâce à des liaisons numériques (élément 8, fig. 8.10) et des coupleurs (élément 9, fig. 8.10). Les coupleurs utilisent des circuits intégrés tels que ACIA, PIA, UART, ...

### 8.35 LIAISONS NUMÉRIQUES

Les liaisons numériques (élément 8, fig. 8.10) sont de deux types :

- Parallèle.
- Série.

Elles permettent la communication entre l'unité centrale et les divers périphériques (interfaces opérateurs, mémoire de masse, ...). Ces liaisons sont nécessaires également pour communiquer avec d'autres systèmes (élément 10, fig. 8.10).

Les liaisons s'appuient sur un support matériel (bus) et sur un protocole.

Le bus de communication peut être parallèle ou série comme l'indique la figure 8.17. Les échanges sont unidirectionnels ou bidirectionnels.

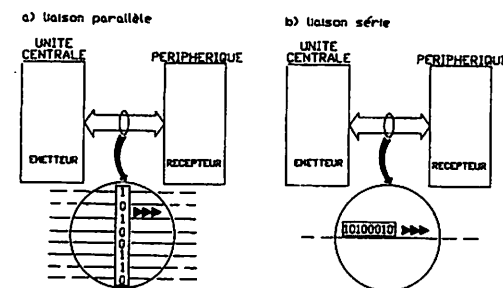


Fig. 8.17 : Liaisons numériques.

Le protocole de transmission est un ensemble de règles qui définissent les échanges entre l'émetteur et le récepteur. Les éléments d'un protocole sont :

- La vitesse d'échange.
- Le mode de transmission.
- Le code de transmission.

Le protocole doit être identique pour l'émetteur et le récepteur.

**Vitesse d'échange :** Elle s'exprime en Baud et représente le nombre de bits par seconde (ex. : 300, 9 600, 19 200, ...).

**Mode de transmission :** Il définit la structure d'une information numérique, celle-ci est constituée de bits de début et de fin d'information, d'un bit de parité qui assure une détection d'erreur de transmission, de bits de données qui représentent l'information à transmettre.

**Code de transmission :** Suivant le code employé, les bits de données représentent une valeur, un caractère alphanumérique, un symbole, ...

La figure 8.18 illustre une liaison série dans laquelle, les 7 bits de données codés en ASCII (voir tableau 8.2) représentent la transmission de la lettre W.

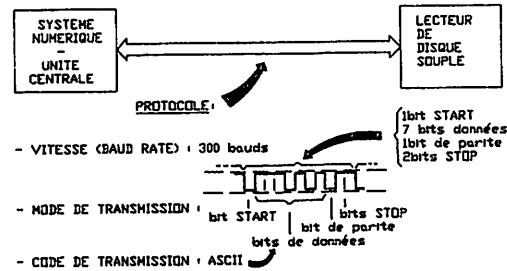


Fig. 8.18 : Exemple de liaison série.

Les bus parallèles, permettent des vitesses d'échange élevées, ils peuvent être utilisés pour relier un clavier alphanumérique à l'unité centrale ou comme bus interne.

Les bus série sont utilisés pour relier l'unité centrale à des périphériques tels que :

- Lecteur de disques souples (floppy disk).
- Automate programmable, ou autre système numérique (régulateur, superviseur, ordinateur personnel, ...).

Les liaisons série tendent à être normalisées, les normes en cours sont par exemple :

- RS 232C, RS 422.
- V 24, V 10.

Ces normes définissent :

- Vitesse maximum en baud.
- Nombre de conducteurs.
- Longueur maximum des câbles.
- Niveaux de tensions.
- Connecteurs et repères.
- Etc.

## 8.4 ASPECT LOGICIEL D'UN SYSTÈME NUMÉRIQUE

### 8.41 LANGAGES DE PROGRAMMATION

Aux éléments matériels d'un micro-ordinateur (**HARDWARE**) est associé un logiciel (**SOFTWARE**) qui représente l'ensemble des programmes propres à être exécutés.

La programmation d'un système nécessite l'utilisation de langages. Ces langages doivent être traduits en binaire (langage machine), seul langage interprété par le micro-ordinateur. La figure 8.19 représente les différentes familles de langages de programmation, depuis la programmation en langage machine, jusqu'aux langages évolués très proches du langage parlé.

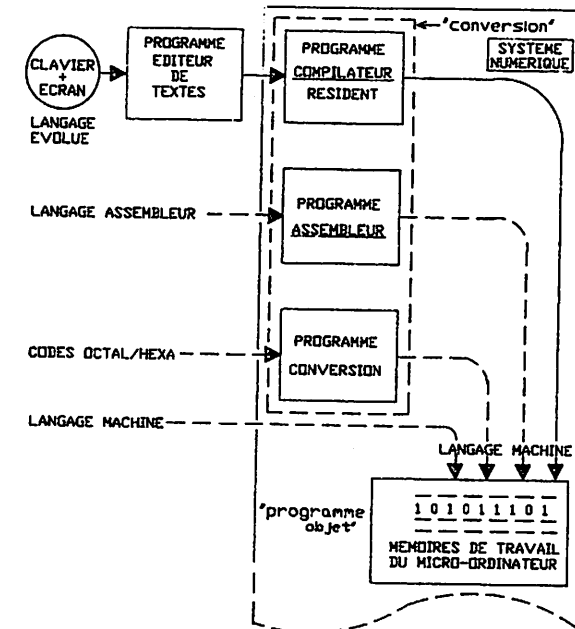


Fig. 8.19 : Langages et conversion en langage machine.

Dans les systèmes numériques, la programmation est le plus souvent réalisée à partir de langages évolués qui sont traduits en langage machine par une opération de compilation (programme compilateur résident). Les langages évolués les plus connus sont le FORTRAN, le BASIC et le PASCAL.

Les constructeurs de systèmes numériques de contrôle de procédé développent leurs propres langages (FORTRAN, FORTH, TML, ...). L'exemple suivant montre l'écriture d'une addition sous trois langages différents.

$$\begin{aligned} C &= A + B ; \\ X_1 + X_2 &= : .Y ; \\ C100 &= C101 C102 + ; \end{aligned}$$

Les programmes sont écrits en langage évolué à l'aide d'un éditeur de textes et sont visualisés sur un écran. L'éditeur de textes est un programme d'aide qui

permet d'écrire ou de corriger les lignes du programme. Le programme définitif (programme source) peut être ensuite stocké en mémoire RAM ou EPROM ou en mémoire de masse (ex. floppy disk).

## 8.42 CONFIGURATION ET PROGRAMMATION

Le logiciel permet suivant le cas :

- De configurer un système.
- De programmer un système.
- De configurer et programmer un système.

Le logiciel peut être également figé (programme non modifiable), ce qui est le cas de nombreux régulateurs numériques monoblocs.

### 8.421 Configuration

Un système est dit configurable lorsque les différentes fonctions (régulation, calcul, ...) sont présentes sous forme de blocs logiciels stockés en mémoire morte. L'ensemble des fonctions préprogrammées constitue une bibliothèque de blocs (fig. 8.20). Notons que chaque fonction est présente unitairement en mémoire morte.

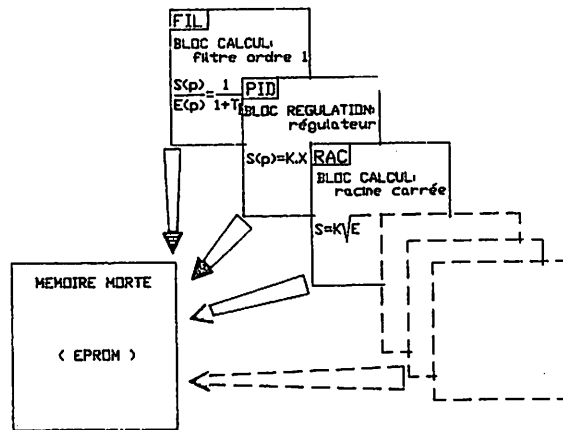


Fig. 8.20 : Exemple de bibliothèque de blocs.

A l'aide d'instructions spécifiques, on peut réaliser les opérations suivantes :

- Dupliquer  $n$  fois un bloc en mémoire vive (fig. 8.21 (a)).
- Fixer les paramètres de chaque bloc (fig. 8.21 (b)).
- Relier la sortie d'un bloc à une ou plusieurs entrées d'autres blocs (SOFTWIRING) (fig. 8.21 (c)).

La configuration permet de créer ou de modifier les boucles de régulation. La mise en œuvre d'une configuration est plus simple que l'utilisation d'un langage de programmation, mais est limitée aux fonctions existantes en bibliothèque. La sauvegarde de tous les éléments d'une configuration est possible en mémoire de masse.

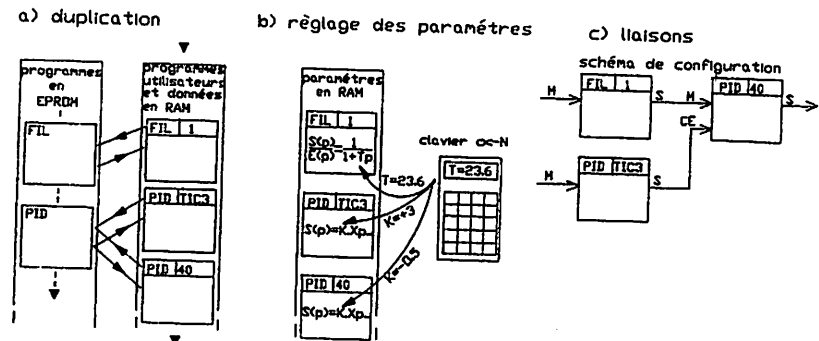


Fig. 8.21 : Opérations de configuration.

### 8.422 Programmation

Un système est dit programmable lorsque l'on crée des fonctions grâce à un langage évolué, ce qui nécessite de la part de l'utilisateur la connaissance du langage, de la structure logique du système et dans certains cas de la mise en œuvre d'expressions mathématiques telles que des équations récurrentes.

La programmation permet de créer des algorithmes adaptés à une application particulière quelle que soit sa complexité. Comme pour la configuration, la sauvegarde de la programmation est possible en mémoire de masse.

## 8.5 DIFFÉRENTS TYPES DE SYSTÈMES NUMÉRIQUES DE CONTRÔLE COMMANDE DE PROCÉDÉS

Nous venons de voir les éléments qui constituent tout système numérique. Compte tenu de cette constitution, on peut classer les systèmes numériques industriels utilisés en régulation de la façon suivante :

- Régulateurs numériques.
- Systèmes modulaires.
- Systèmes répartis.
- Systèmes centralisés.

## 8.51 RÉGULATEURS NUMÉRIQUES

Leurs faces avant et leur conduite sont très proches de celles des régulateurs analogiques. Les fonctions internes réalisées en numérique offrent plus de possibilités :

- Traitement de signaux analogiques (ex. : filtrage) et signaux logiques (ex. passage auto/manu sur état logique externe).
- Utilisation de plusieurs entrées analogiques (ex. : 3 signaux pour un débit de gaz corrigé).
- Choix des entrées (ex. : 4 à 20mA, 0 à 10 V).
- Choix du signal de sortie en fonction de l'organe de réglage commandé (continu, discontinu, ...).
- Sélection d'alarmes (absolue, sur écart, sur vitesse, ...).

Les fonctions principales de régulation sont figées, configurables ou programmables. Dans le premier cas, il est possible d'obtenir d'autres fonctions par changement de mémoire morte. Dans les deux autres cas, il est possible de configurer ou de programmer des fonctions spécifiques par :

- Un clavier incorporé au régulateur.
- Une liaison numérique et un clavier.

Le changement des paramètres de régulation s'effectue à l'aide d'une microconsole, d'un clavier interne ou par l'intermédiaire d'une liaison numérique et d'un clavier.

Les régulateurs numériques se rencontrent sous différentes formes ; de la plus simple, voisine de celle d'un régulateur analogique jusqu'à celle d'un régulateur multiboucle programmable.

Si une liaison numérique est prévue sur ces régulateurs, ils peuvent s'intégrer dans un système numérique modulaire.

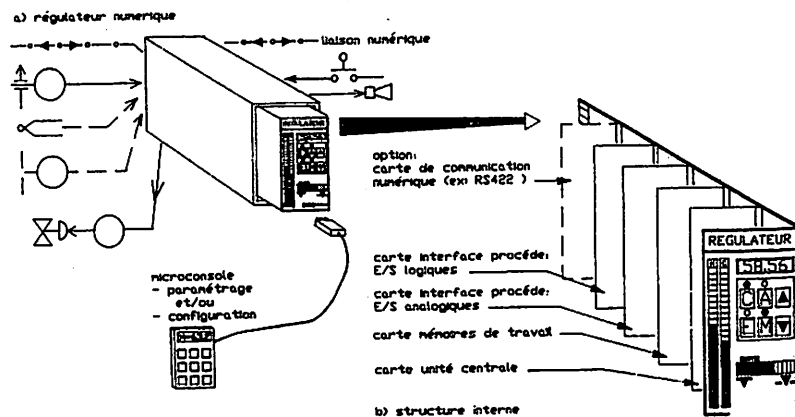


Fig. 8.22 : Exemple de régulateur numérique monobloc.

## 8.52 SYSTÈMES NUMÉRIQUES MODULAIRES

La figure 8.23 montre l'architecture d'un système modulaire. Les différents modules sont connectés sur une liaison numérique commune. Ils ont des fonctions préprogrammées figées, ou ils sont configurables ou programmables. Parmi ces modules, on distingue :

- Des régulateurs numériques du type de ceux décrits en 8.51.
- Des modules orientés calculs.
- Des modules orientés traitement logique.

Chaque module peut être utilisé ou conduit individuellement.

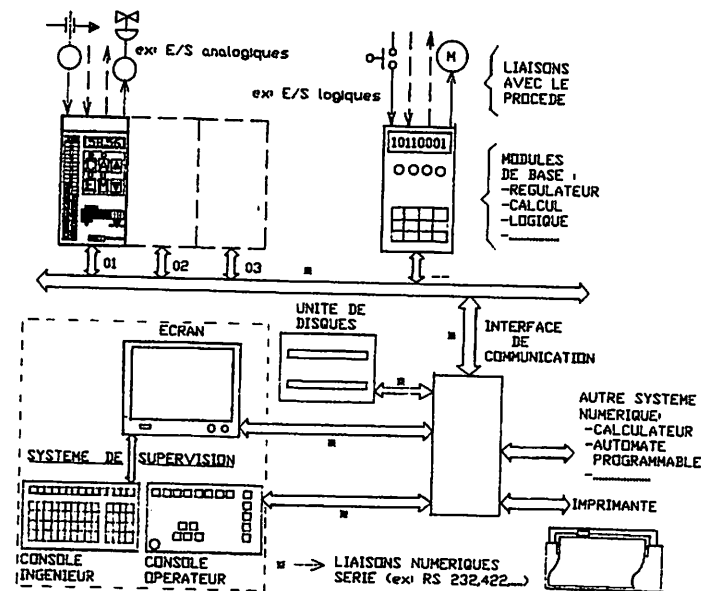


Fig. 8.23 : Exemple de système numérique modulaire.

Un interface de communication (superviseur) associé à des éléments périphériques (mémoire de masse, claviers, écran, ...) permet de gérer l'ensemble des modules ; les fonctions suivantes sont alors possibles :

- Conduite centralisée de l'ensemble des modules.
- Échanges d'informations entre modules.
- Configuration, programmation, réglage des paramètres d'un module à l'aide d'une console ingénieur.
- Sauvegarde ou chargement du programme d'un module par la mémoire de masse (unité de disque).

D'autres systèmes numériques tels qu'automate programmable ou ordinateur peuvent être reliés à l'interface de communication, ce qui permet d'augmenter la capacité de traitement du système.

Le système de supervision peut être un micro-ordinateur personnel (PC), dans lequel sont implantés des logiciels spécifiques (conduite, programmation, aide, ...).

### 8.53 SYSTÈMES NUMÉRIQUES RÉPARTIS

La figure 8.24 montre un système à cartes réparties aveugles. Ces systèmes peuvent être considérés comme des systèmes modulaires dont les modules seraient sans face avant.

Les cartes à micro-processeurs (régulation, acquisitions, traitements logiques, ...) gèrent un nombre limité de boucles de régulation (8 ou 16). Chaque carte est en fait un micro-ordinateur et possède tous les éléments qui lui confèrent son autonomie, justifiant ainsi le terme de répartis par opposition aux systèmes centralisés.

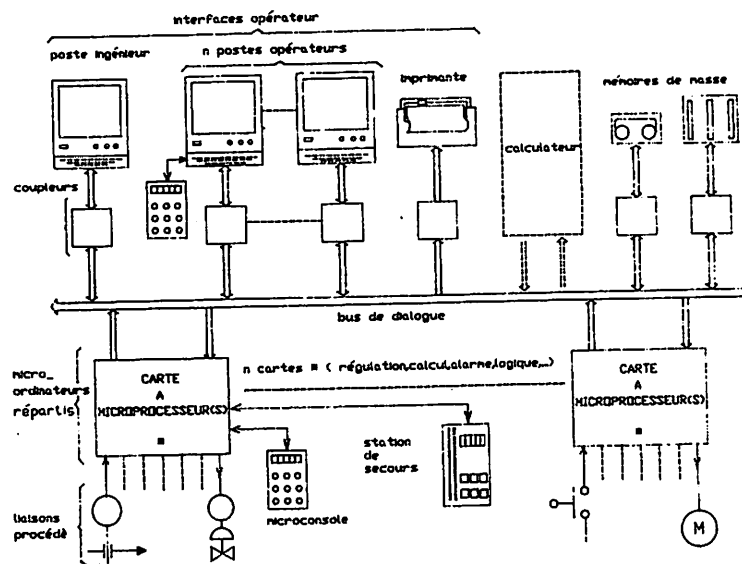


Fig. 8.24 : Exemple de système numérique réparti.

Des régulateurs de secours peuvent assurer une redondance pour certaines boucles de régulation.

Sur chaque carte, les mémoires mortes permettent d'utiliser des fonctions préprogrammées ; ces systèmes sont donc configurables. Sur certains systèmes, il est en plus possible de programmer de nouvelles fonctions.

Les cartes réparties peuvent être reliées à des interfaces opérateurs par l'intermédiaire de bus de dialogue, ce qui permet d'assurer la supervision de l'ensemble.

Ces systèmes sont bien adaptés pour gérer un grand nombre de boucles. Ils permettent une souplesse de conduite en utilisant des moyens tels que :

- Gestion d'alarmes.
- Conduite par synoptiques pilotables.
- Vues d'historiques.

### 8.54 SYSTÈMES NUMÉRIQUES CENTRALISÉS

Dans ces systèmes, une seule unité centrale gère l'ensemble des entrées/sorties.

La figure 8.25 représente un système de ce type ; le calculateur assure l'ensemble des tâches. Ces systèmes sont souvent orientés vers la supervision et la gestion. Les fonctions sont :

- Mémorisation et traitement des informations.
- Calculs associés à la conduite de procédés (supervision de points de consigne).
- Bilans.

Les fonctions de régulation et de calcul qui viennent s'ajouter aux précédentes, sont configurables et programmables.

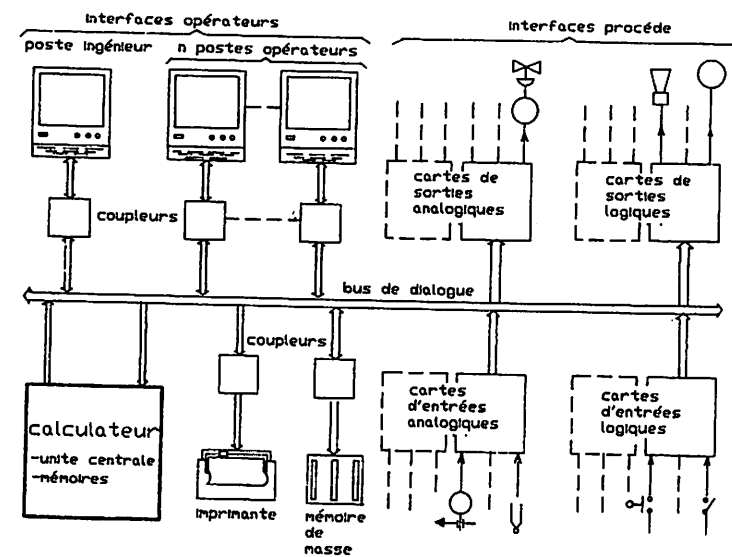


Fig. 8.25 : Exemple de système numérique centralisé.

Une défaillance du calculateur entraîne l'indisponibilité de toutes les boucles de régulation. Parmi les solutions à ce problème, notons :

- Deuxième calculateur de secours (STAND-BY).
- Régulateurs de secours sur certaines boucles de régulations.
- Commandes manuelles de secours.

## 8.6 INFLUENCE DU TRAITEMENT NUMÉRIQUE SUR LA RÉGULATION

Les trois principaux paramètres qui interviennent sur la régulation par systèmes numériques sont :

- $T_e$  : période d'échantillonnage (généralement fixe).
- $T_c$  : temps de cycle de calcul (fixe ou modifiable).
- $T_i$  : temps d'exécution d'un programme (exemple : algorithme régulateur).

### 8.61 TEMPS DE CYCLE DE CALCUL

Autres appellations : Cycle ou période de scrutation.  
Période de traitement.  
Période de l'algorithme.

#### 8.611 Choix de $T_c$

Le temps de cycle de calcul  $T_c$  peut être différent de la période d'échantillonnage  $T_e$ . Lorsque  $T_c$  est réglable, il est souhaitable de respecter la condition suivante :

$$T_c \geq T_e$$

La majorité des systèmes numériques remplissent cette condition. Le choix  $T_c$  est dicté par le type de mesure à régler (évolution lente ou rapide). Il faut également vérifier que la valeur de  $T_e$  est adaptée à la charge de l'unité centrale.

La période  $T_c$  doit être telle que le signal, exploité par le programme du calculateur, se rapproche le plus possible du signal analogique. La figure 8.26 montre des valeurs de  $T_c$  correctes ( $T_c = T_e$ ) et incorrectes ( $T_c = 4T_e$ ).

Si la mesure varie rapidement, il faut un temps  $T_c$  faible, cette remarque montre que  $T_e$  et en conséquence  $T_c$  doivent être adaptés au signal de mesure. A titre d'indication, vérifier que la valeur de  $T_e$  est :

- inférieure à 500 ms pour une boucle rapide (ex. : débit),
- d'environ 5 s pour une boucle lente (ex. : température).

En comparaison, les valeurs typiques pour  $T_e$  sont :

- de l'ordre de 50 ms pour un régulateur numérique monoboucle,
- de 100 ms à 1 s suivant les systèmes numériques centralisés ou répartis.

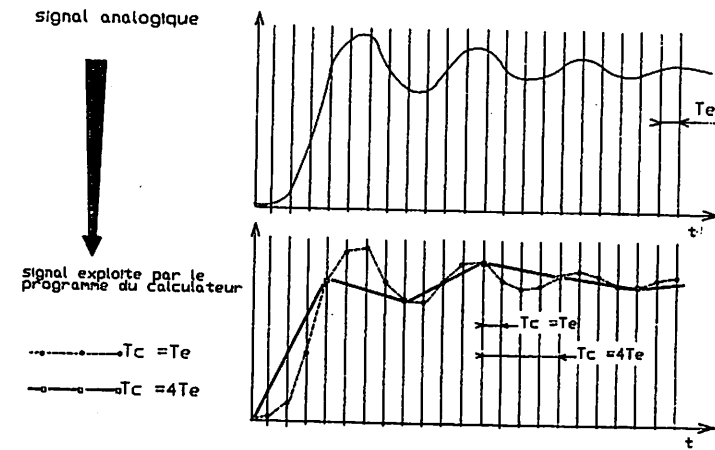


Fig. 8.26 : Échantillonnage d'un signal.

### 8.612 Charge de l'unité centrale

La figure 8.27 montre le traitement séquentiel des blocs logiciels de régulation pour deux cas de figure : (1)  $T_c = T_e$  et (2)  $T_c = 2T_e$ .

Les temps de conversion analogique numérique et numérique analogique étant considérés comme négligeables par rapport au temps de traitement des blocs logiciels  $T_i$  (ex. : 20 ms) ne sont pas représentés.

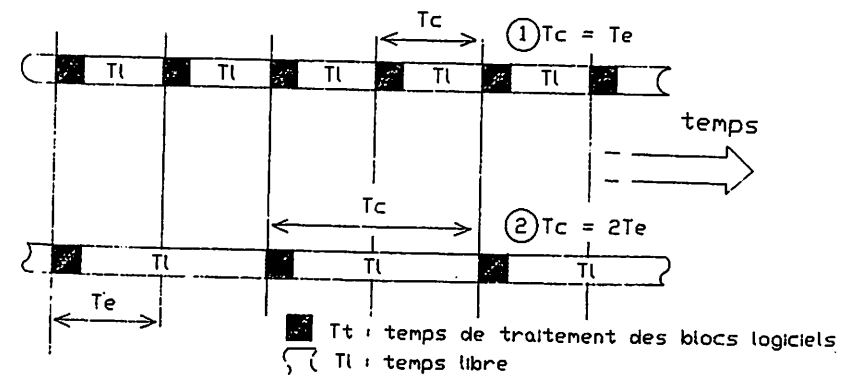


Fig. 8.27 : Charge unité centrale.

En observant ce schéma, on constate que  $T_c$  doit être choisi le plus grand possible, ceci augmente le temps libre  $T_1$  et permet de diminuer la charge de l'unité centrale.

### 8.613 Conclusions sur $T_c$

L'étude précédente montre que le choix de  $T_c$  est un compromis entre un  $T_c$  faible suivant la grandeur physique à régler et un  $T_c$  fort pour réduire la charge de l'unité centrale.

L'identification d'un procédé permet de déterminer une valeur pratique de  $T_c$ , suivant le type de procédé à dominante temps mort  $\tau$  ou à dominante constante de temps  $\theta$  :

- $1/8$  de  $\tau < T_c < 1/4$  de  $\tau$
- $1/10$  de  $\theta < T_c < 1/6$  de  $\theta$ .

**Remarque :** Pour les procédés très rapides (ex. : débit, vitesse), il faut s'assurer en plus que la période d'échantillonnage soit suffisamment faible.

**Exemple :** Pour un procédé à dominante temps mort  $\tau$

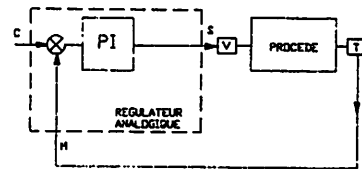
$$\begin{aligned} T_c &= 250 \text{ ms} \\ \tau &= 10 \text{ s} \\ \tau/8 < T_c < \tau/4 \\ 1,25 < T_c < 2,5 \text{ s} \\ \text{On choisit } T_c &= 2 \text{ s.} \end{aligned}$$

La condition  $T_c \geq T_c$  est respectée.

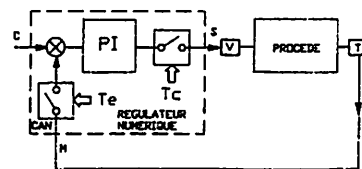
### 8.62 APPLICATION SUR LA STABILITÉ D'UNE RÉGULATION NUMÉRIQUE

La figure 8.28 représente un procédé régulé soit par un régulateur analogique, soit pas un régulateur numérique. Les régulateurs sont de mode PI et de structures identiques.

a) boucle analogique



b) boucle numérique



$T_e$  : période d'échantillonnage fixe ( $T_e < T_c$ )  
 $T_c$  : temps de cycle de calcul

Fig. 8.28 : Boucles analogique et numérique.

Les paramètres du procédé identifié sont :

$$\begin{aligned} \tau &= 8 \text{ s} \\ \theta &= 14 \text{ s} \\ G_s &= 1. \end{aligned}$$

La figure 8.29 montre les résultats du réglage en asservissement avec un régulateur analogique.

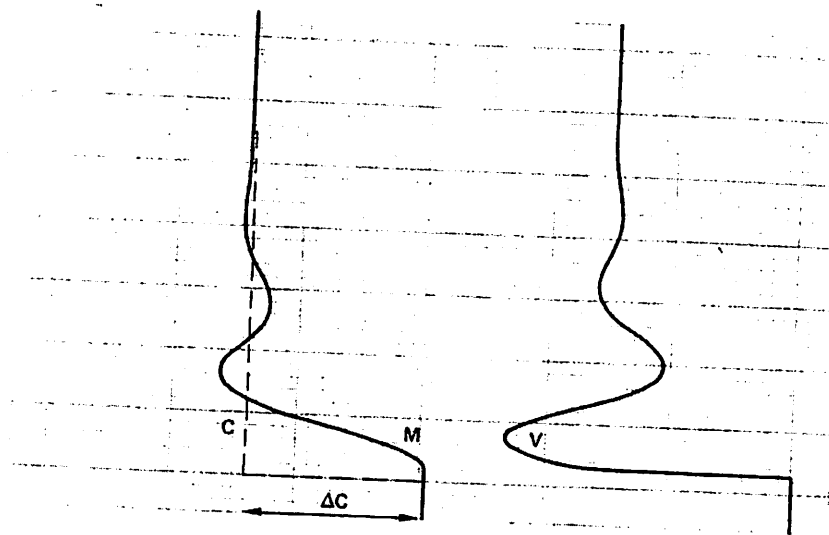


Fig. 8.29 : Réponse avec régulateur analogique.

La figure 8.30 montre les résultats du réglage en asservissement avec les mêmes valeurs d'actions que précédemment pour deux valeurs de  $T_c$  différentes.

En utilisant les relations données en 8.613, on obtient :  $1 \text{ s} < T_c < 2 \text{ s}$ .

Avec  $T_c = 1 \text{ s}$ , la stabilité est conservée.

Avec  $T_c = 4 \text{ s}$ , on observe une dégradation de la stabilité.

D'un point de vue pratique, retenons qu'avec une valeur de  $T_c$  faible devant  $\tau$  ou  $\theta$ , les méthodes de réglages de boucles qui s'appliquent à l'analogique restent valables en numérique.

Signalons que sur la plupart des systèmes, les actions de réglages doivent changer lorsqu'on modifie  $T_c$ . Il faut en tenir compte lors des réglages des boucles.