

Chapitre 4

Programmation des équations récurrentes

- 4.1. Résolution des équations récurrentes**
 - 4.1.1. Transformée en « z » de l'équation récurrente
 - 4.1.2. Application
- 4.2. Calculateur pour la commande numérique**
 - 4.2.1. Différentes fonctions du calculateur numérique
 - 4.2.2. Calculateur « temps réel »
 - 4.2.3. Aspects pratiques
- 4.3. Programmation des algorithmes de commande**
 - 4.3.1. Principes de la programmation “ temps réel ”
 - 4.3.2. Exemple de programmation

4.1. RESOLUTION DES EQUATIONS RECURRENTES

4.1.1. TRANSFORMEE EN « z » DE L'EQUATION RECURRENTTE

La transformée en z permet de résoudre les équations récurrentes (ou équations aux différences) telles que :

$$b_0 s_k + b_1 s_{k-1} + b_2 s_{k-2} + \dots + b_d s_{k-d} = a_0 e_k + a_1 e_{k-1} + a_2 e_{k-2} + \dots + a_n e_{k-n}$$

$$\sum_{j=0}^d b_j s_{k-j} = \sum_{i=0}^n a_i e_{k-i}$$

Posons $\mathbf{Z}[s_k] = S(z)$ et $\mathbf{Z}[e_k] = E(z)$. Il vient alors :

$$\sum_{j=0}^d b_j \mathbf{Z}[s_{k-j}] = \sum_{i=0}^n a_i \mathbf{Z}[e_{k-i}]$$

En appliquant la propriété $\mathbf{Z}\{x_{k-j}\} = z^{-j} X(z) + \sum_{m=-j}^{-1} x_m z^{-(m+j)}$ on peut écrire :

$$\sum_{j=0}^d b_j z^{-j} S(z) + b_j \sum_{m=-j}^{-1} s_m z^{-(m+j)} = \sum_{i=0}^n a_i z^{-i} E(z) + a_i \sum_{m=-i}^{-1} e_m z^{-(m+i)}$$

$$S(z) \sum_{j=0}^d b_j z^{-j} = E(z) \sum_{i=0}^n a_i z^{-i} + I(z)$$

$I(z)$ est un polynôme qui rend compte des conditions initiales.

$$I(z) = \sum_{i=0}^n a_i z^{-i} \sum_{m=-i}^{-1} e_m z^{-m} - \sum_{j=0}^d b_j z^{-j} \sum_{m=-j}^{-1} s_m z^{-m}$$

En définitive il vient :

$$S(z) = E(z) \frac{\sum_{i=0}^n a_i z^{-i}}{\sum_{j=0}^d b_j z^{-j}} + \frac{I(z)}{\sum_{j=0}^d b_j z^{-j}}$$

4.1.2. APPLICATION

a. *Enoncé du problème posé :*

Considérons le filtre numérique d'équation récurrente $s_k + s_{k-1} = e_{k-1}$ et supposons qu'il admette les conditions initiales $s_{-1} = 1$ et $e_{-1} = 0$.

Sachant qu'il est excité par la séquence $e_k = k\alpha^k$, on désire calculer sa séquence de sortie s_k .

b. *Résolution du problème :*

$$S(z) + z^{-1}S(z) + s_{-1} = z^{-1}E(z) + e_{-1} \quad \Rightarrow \quad S(z) + z^{-1}S(z) + 1 = z^{-1}E(z)$$

$$S(z) = \frac{z^{-1}E(z)}{1+z^{-1}} - \frac{1}{1+z^{-1}} = \frac{E(z)}{z+1} - \frac{z}{z+1}$$

Calculons $E(z)$.

$$E(z) = \mathbf{Z}[k\alpha^k] = \sum_{k=0}^{\infty} k\alpha^k z^{-k} = -z \frac{d}{dz} \left[\sum_{k=0}^{\infty} \alpha^k z^{-k} \right] = -z \frac{d}{dz} \left[\frac{1}{1-\alpha z^{-1}} \right] = \frac{\alpha z}{(z-\alpha)^2}$$

On peut aussi trouver ce résultat en consultant le dictionnaire des transformées en z . En effet on remarque que les termes de la suite $\{e_k\}$ correspondent à l'échantillonnage du signal continu $e(t) = t.e^{-at}$ avec $T = 1s$ et $\alpha = e^{-a}$ alors :

$$e(kT) = kT.e^{-akT} = k.\alpha^k$$

$$\text{Or } \mathbf{Z}[te^{-at}] = \frac{Tze^{-aT}}{(z-e^{-aT})^2} \quad \text{ainsi} \quad \mathbf{Z}[k\alpha^k] = \frac{\alpha z}{(z-\alpha)^2}$$

Revenons au calcul de la séquence de sortie du filtre numérique.

$$S(z) = \frac{\alpha z}{(z-\alpha)^2(z+1)} - \frac{z}{z+1}$$

$$s_k = \sum_{\substack{\text{pôles de } S(z) \\ z=-1 \text{ (simple)} \\ z=\alpha \text{ (double)}}} \text{Résidus de } z^{k-1}S(z) = \sum_{\substack{\text{pôles de } S(z) \\ z=-1 \text{ (simple)} \\ z=\alpha \text{ (double)}}} \text{Résidus de } \left\{ \frac{\alpha z^k}{(z-\alpha)^2(z+1)} - \frac{z^k}{z+1} \right\}$$

$$s_k = -(-1)^k + \sum_{\substack{\text{pôles de } S(z) \\ z=-1 \text{ (simple)} \\ z=\alpha \text{ (double)}}} \text{Résidus de } \left\{ \frac{\alpha z^k}{(z-\alpha)^2(z+1)} \right\} = -(-1)^k + R_{-1} + R_\alpha$$

Ainsi :

$$R_{-1} = \frac{\alpha(-1)^k}{(1+\alpha)^2} \text{ et } R_{-\alpha} = \left[\frac{d}{dz} \frac{\alpha z^k}{(z+1)} \right]_{z=\alpha} = \left[\frac{(z+1)k\alpha z^{k-1} - \alpha z^k}{(z+1)^2} \right]_{z=\alpha} = \frac{(\alpha+1)k\alpha^k - \alpha^{k+1}}{(\alpha+1)^2}$$

$$s_k = \frac{(\alpha+1)k\alpha^k - \alpha^{k+1} + \alpha(-1)^k}{(\alpha+1)^2} - (-1)^k$$

c. Etude du cas particulier $\alpha=1$

$$S(z) = \frac{z}{(z-1)^2(z+1)} - \frac{z}{z+1} = \frac{-z^3 + 2z^2}{z^3 - z^2 - z + 1} = \frac{-1 + 2x}{1 - x - x^2 + x^3} \text{ avec } x = z^{-1}$$

Déterminons les échantillons par division selon les puissances croissantes de x .

-1	+2x					+1	-x	-x ²	+x ³		
+1	-x	-x ²	+x ³			-1	+x	+0x ²	+2x ³	+x ⁴	+....
0	+x	-x ²	+x ³								
	-x	+x ²	+x ³	-x ⁴							
	0	0	+2x ³	-x ⁴							
			-2x ³	+2x ⁴	+2x ⁵	-2x ⁶					
			0	+x ⁴	+2x ⁵	-2x ⁶					

Ainsi :

$$S(z) = -1 + z^{-1} + 0z^{-2} + 2z^{-3} + z^{-4} + \dots$$

Chaque échantillon vérifie bien l'expression $s_k = \frac{2k-1-3(-1)^k}{4}$

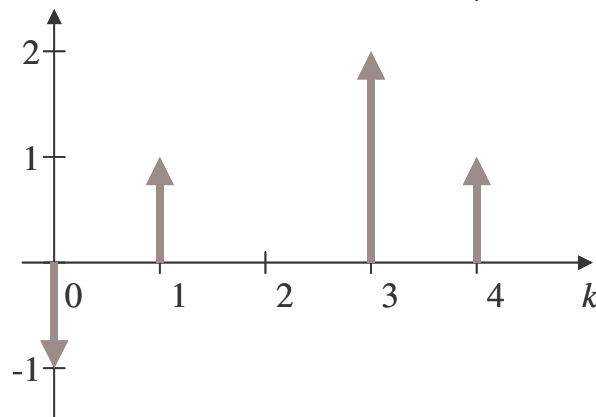


Figure 1 : Réponse du filtre numérique

4.2. CALCULATEUR POUR LA COMMANDE NUMERIQUE

4.2.1. DIFFERENTES FONCTIONS DU CALCULATEUR NUMERIQUE

Le calculateur est un *processeur numérique « temps réel »* qui doit :

- *collecter les informations* délivrées par des capteurs, les traiter éventuellement (e.g. linéarisation) et les traduire en unité convenable ;
- *élaborer le signal de commande* conformément à l'algorithme correspondant à l'équation récurrente du filtre numérique dont on désire réaliser la synthèse ;
- *effectuer les traitements “annexes”* concernant le pilotage de l'installation comme la tenue du “cahier de marche”. Ces informations sont précieuses car elles permettent de procéder à des bilans et d'optimiser le fonctionnement de la chaîne de commande ;
- *surveiller le processus et délivrer les alarmes* associées aux dépassements de seuils critiques.

4.2.2. CALCULATEUR « TEMPS REEL »

La notion de temps réel doit être précisée car un calculateur numérique exécute les opérations de manière séquentielle et non simultanée. Le temps d'exécution d'un cycle comportant de multiples tâches d'acquisition, de traitement et de sorties de données, est en général très court devant les constantes de temps caractéristiques du processus piloté. On choisira le processeur numérique et les composants qui l'environnent en conséquence. Toutefois si le traitement des données introduit un retard « pur » jugé important on doit en tenir compte dans la synthèse du correcteur numérique (Cf. § 3.5.). Par ailleurs, pour des applications nécessitant un volume de calcul élevé à des vitesses importantes, il peut être indispensable d'utiliser des unités de calcul câblées ou microprogrammées indépendantes de l'unité centrale.

Au chapitre 0 nous avons décrit l'architecture d'un dispositif de commande numérique :

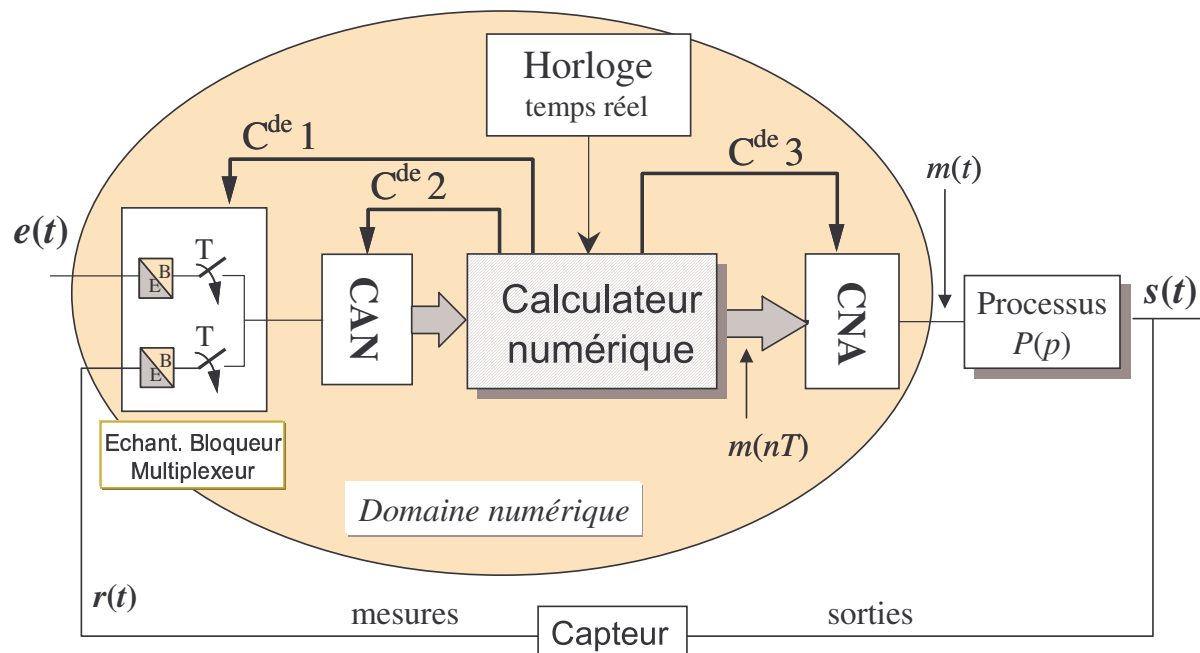


Figure 2 : Architecture d'une boucle de commande temps réel

Il s'agit de concevoir le logiciel qui sollicite en temps opportun les divers périphériques

concernés et exécute les opérations mathématiques correspondant à la résolution de l'algorithme de commande. Le rythme de travail du dispositif est donné par une **horloge temps réel** (RTC = real time clock). Le signal délivré est appliqué à une entrée du **système d'interruptions**, en général la plus prioritaire¹.

Le fonctionnement d'un système temps réel repose sur l'utilisation du système d'interruptions. Lorsqu'un signal extérieur (e.g. l'horloge RTC) apparaît, il « attaque » le dispositif d'interruptions et prévient ainsi le calculateur qu'il a une opération urgente à effectuer. Le calculateur termine l'instruction en cours d'exécution, sauvegarde le contexte de travail du programme interrompu (état interne du processeur, registres, compteurs, adresses, etc.), se déroute vers l'adresse du programme à traiter. Lorsque le traitement est terminé l'état du calculateur au moment de l'interruption est restauré et le calculateur reprend normalement l'exécution du programme précédent.

4.2.3. ASPECTS PRATIQUES

a. Acquisition des données

En général il est nécessaire d'acquérir plusieurs données à la fois. Ainsi pour un asservissement on mesurera l'entrée² (ou la consigne) et le signal de retour. On rappelle que les signaux doivent être filtrés avant échantillonnage (Cf. § 1.3.3.) afin d'éliminer les bruits. Pour économiser le nombre de CAN on pourra utiliser un **multiplexeur temporel** comportant une batterie d'interrupteurs électroniques identifiés par leur adresse. La sélection de chaque interrupteur est réalisée par le processeur qui indique sur le « bus de commande » l'adresse du « switch » concerné. L'adressage des voies étant séquentiel on introduit un retard avec les acquisitions de données qui ne sont pas synchrones. Afin de remédier à cet inconvénient on peut placer un **échantillonneur-bloqueur** sur chaque voie (mesure des signaux au même instant et mémorisation de leur valeur à l'instant de mesure). Les signaux numérisés par le CAN sont transmis au processeur numérique par le bus de données. Ces signaux sont traités en temps différé, entre les instants d'échantillonnage.

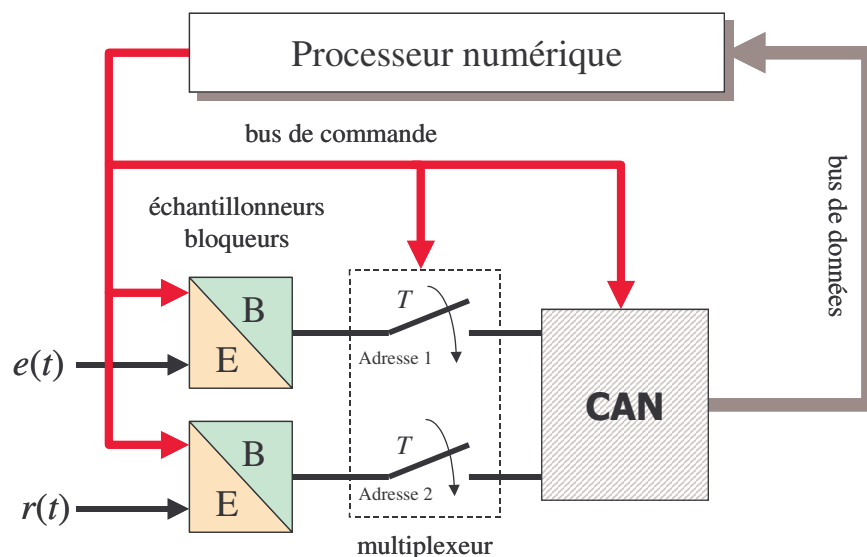


Figure 3 : Détails de réalisation de l'acquisition des signaux d'entrée

On notera que le capteur peut être numérique, auquel cas l'architecture du dispositif

¹ La plus prioritaire peut être attribuée au système de sécurité.

² Souvent la consigne est fournie par l'opérateur à partir du clavier. Cette donnée est mémorisée.

d'acquisition est différente.

b. Période d'échantillonnage

Le choix de la période d'échantillonnage T résulte toujours d'un compromis. En effet, la période d'échantillonnage T doit permettre au calculateur de réaliser un cycle complet de calcul :

- une trop grande période d'échantillonnage appauvrit les performances de la boucle de commande ;
- inversement une période d'échantillonnage courte augmente la quantité d'information recueillie et peut laisser penser qu'en se rapprochant de la commande continue on améliore significativement les performances.

Or ce choix conduit généralement à des problèmes tels que :

- précision des coefficients et programmation de ces données sur des processeurs à nombre de bits limité ;
- robustesse du correcteur, c'est à dire aptitude de la boucle de commande à limiter la dégradation des performances malgré une évolution des paramètres caractéristiques du processus ;
- saturation des actionneurs (signaux de commande trop importants pour une commande en temps minimum) ;
- apport de bruits, la dérivée d'un signal bruité est donnée par :

$$\left[\frac{d(x \pm b)}{dt} \right]_{t=kT} \cong \left(\frac{x_k - x_{k-1}}{T} \right) \pm \frac{2b}{T}$$

si la valeur de T est trop petite alors $x_k \approx x_{k+1}$ et la dérivée est noyée dans le bruit.

Le choix de T doit aussi respecter la condition de SHANNON : $T \leq \pi/\omega_{\max}$. Aussi filtre-t-on les signaux avant échantillonnage afin d'éliminer les bruits. Cette précaution constitue une garantie contre le repliement de spectre. En pratique on adopte :

$$\frac{\pi}{12,5\omega_{\max}} \leq T \leq \frac{\pi}{2,5\omega_{\max}} \Rightarrow 5F_{\max} \leq F_{\text{éch.}} \leq 25F_{\max}$$

<i>Premier ordre</i>	<i>Deuxième ordre</i>
$\frac{1}{1 + \tau p}$	$\frac{\omega_n^2}{p^2 + 2\zeta\omega_n p + \omega_n^2}$
$F_{\max} \cong \frac{1}{2\pi\tau}$	$F_{\max} \cong \frac{\omega_n}{2\pi}$
$\frac{\tau}{4} \leq T_{\text{éch.}} \leq 1,25\tau$	$\frac{1}{4\omega_n} \leq T_{\text{éch.}} \leq \frac{1,25}{\omega_n}$

τ et ω_n sont les paramètres du système bouclé et non ceux du processus. Signalons qu'en

général la détermination de F_{\max} est toujours délicate. Enfin rappelons que si l'on modifie T il convient de recalculer les transformées en z des transmittances concernées.

4.3. PROGRAMMATION DES ALGORITHMES DE COMMANDE

4.3.1. PRINCIPES DE LA PROGRAMMATION “ TEMPS REEL ”

A la mise sous tension du calculateur le module d'initialisation ajuste le contenu des compteurs et celui des autres positions mémoires intervenant dans les calculs. Le programme principal qui gère les différents travaux confiés au calculateur se déroule normalement lorsque à l'instant $t = kT$ l'horloge temps réel (RTC) délivre un signal au système d'interruption qui force l'exécution d'un sous-programme d'interruption implanté à l'adresse S/P INT1. Après avoir sauvegardé le contexte de travail du processeur (contenu du compteur ordinal et des divers registres), S/P INT1 force le saut vers le sous-programme « commande » (S/P CDE) qui exécute les opérations décrites par le schéma suivant.

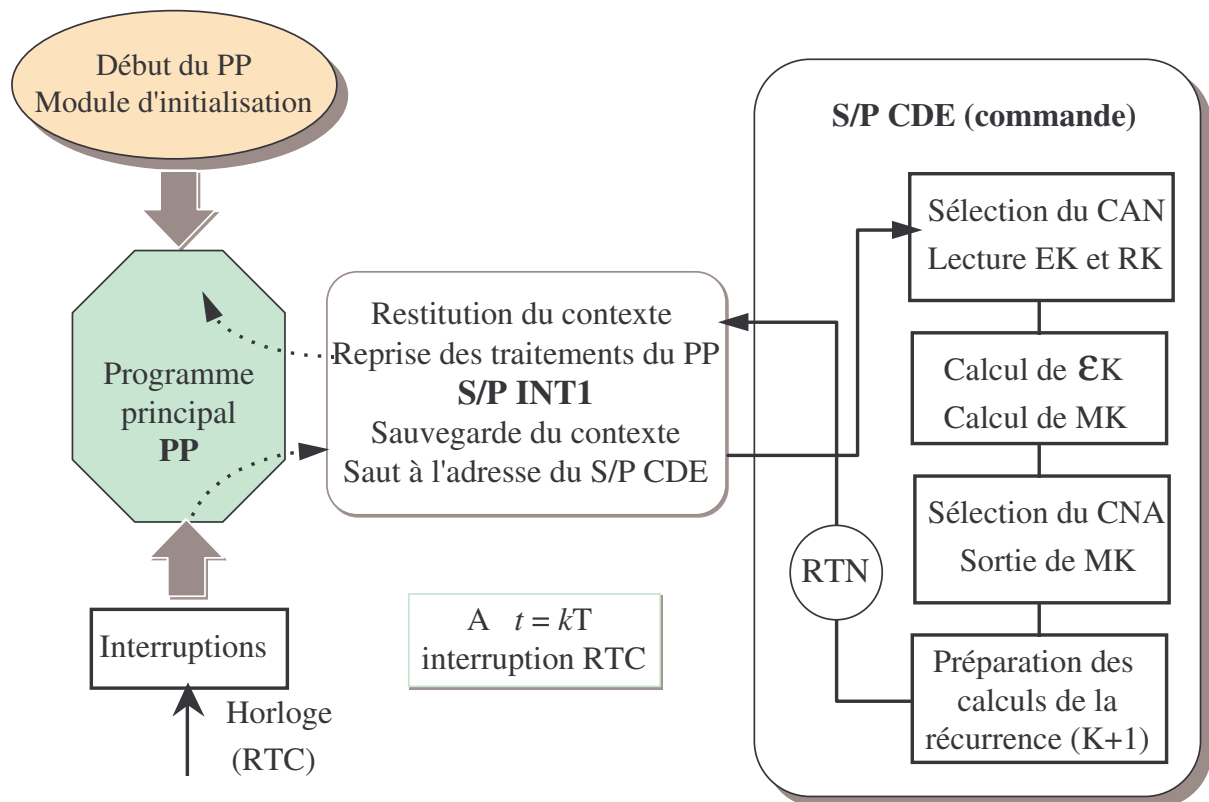


Figure 4 : Organisation générale du logiciel temps réel

Le calcul de l'échantillon de commande MK est programmé selon les données de l'équation récurrente.

La méthode de programmation adoptée vise à réduire le temps qui s'écoule entre l'acquisition des signaux (EK et RK) et la sortie de la commande MK. Ainsi lorsque le signal MK a été transmis au CNA on peut préparer le calcul de la prochaine récurrence en actualisant la valeur du terme représentatif du passé.

La chronologie des travaux est la suivante :

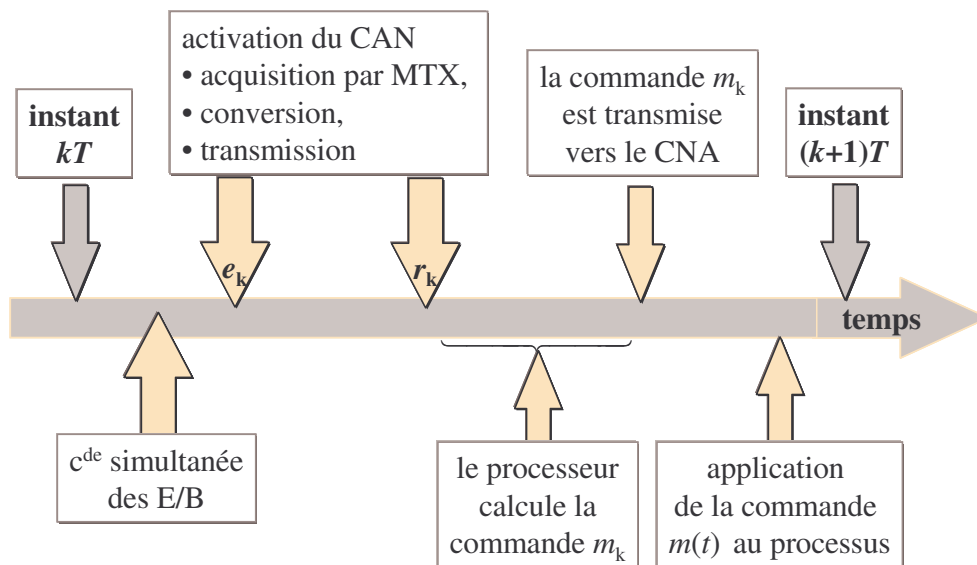


Figure 5 : Chronologie des travaux

4.3.2. EXEMPLE DE PROGRAMMATION

Soit à programmer le filtre numérique $H(z) = \frac{M(z)}{\mathcal{E}(z)} = \frac{2z - 1,2}{z + 0,8} = \frac{2 - 1,2z^{-1}}{1 + 0,8z^{-1}}$

Il correspond à l'équation récurrente $m_k = 2\epsilon_k - 1,2\epsilon_{k-1} - 0,8m_{k-1}$

L'horloge RTC est à la période de 1 ms et la période d'échantillonnage est égale à 1s.

L'algorithme de programmation est le suivant :

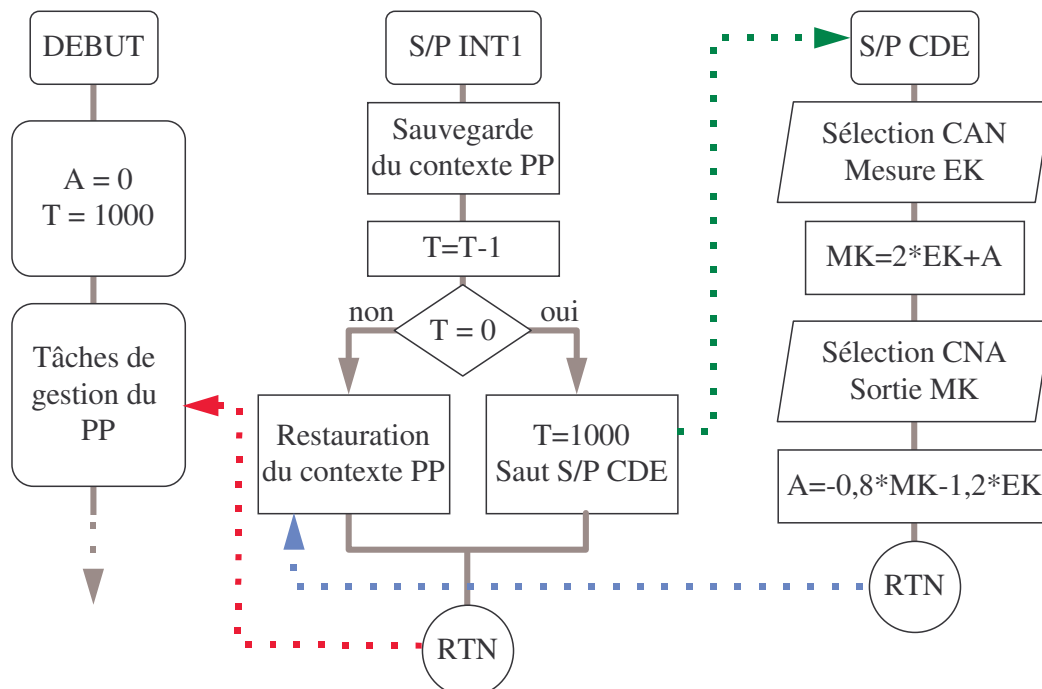


Figure 6 : Logiciel de programmation d'une équation récurrente