INTÉGRATION SUR MICROCONTRÔLEUR

Après avoir abordé le mois dernier les problèmes liés à la dérivation numérique, il était naturel de nous intéresser ce mois-ci à l'intégration numérique. Le calcul de l'intégrale d'une série de valeurs est utilisé dans de nombreuses applications. Ainsi que nous l'avions évoqué le mois dernier, le calcul de l'intégrale intervient par exemple dans les processus de régulation utilisant des boucles PID (PID pour Proportionnel, Intégral, Dérivé) et plus généralement, on utilise l'intégration chaque fois que l'on désire calculer la surface délimitée par la courbe définie par une fonction. Dans le monde analogique, un intégrateur est simplement réalisé avec un amplificateur opérationnel dans la contre-réaction duquel on a mis une capacité. La valeur à intégrer est introduite via une résistance dans l'entrée négative de l'amplificateur, le résultat de l'intégration est donné par la sortie de l'amplificateur opérationnel. Les intégrateurs analogiques sont soumis à la dérive des composants, aux charges résiduelles de la capacité d'intégration, et à la limite de la tension de sortie maximum avant saturation de l'amplificateur. Les intégrateurs digitaux ne sont pas soumis à ces limitations.

L'algorithme le plus simple d'intégration numérique est l'intégration rectangulaire. La valeur de l'intégrale est approchée par la somme de surfaces rectangulaires. La «hauteur» de chaque rectangle est donnée par la valeur numérique entrée dans l'intégrateur, la valeur correspondant à la surface du rectangle est additionnée à la valeur de l'intégrale calculée précédemment.

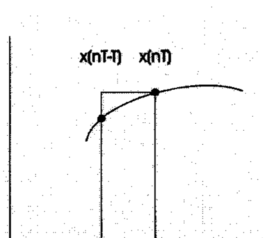
$Y_t = Y_{t-1} + (T \times X_t)$

Y_t est la valeur de l'intégrale actuelle, Y_{t-1} la valeur précédente, X_t le nouvel échantillon introduit dans l'intégrateur, T la période d'échantillonnage. Pour simplifier le calcul, on prend souvent T = 1, le calcul de l'intégrale se résume alors à effectuer la somme des valeurs successives qui sont entrées dans l'intégrateur. La **figure 1a** explicite l'opération réalisée lors de l'intégration rectangulaire.

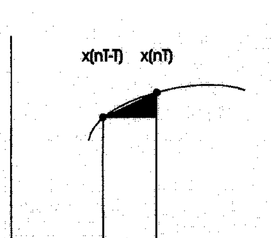
Les réponses en amplitude et en phase de l'intégrateur sont les suivantes :

Amplitude $(\omega T) = T/(2 \times \sin(\omega T/2))$

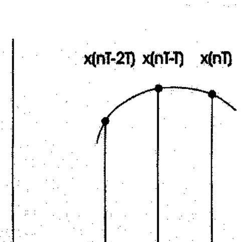
Phase $(\omega T) = -(\Pi + \omega T)/2$.



■ *Figure 1a*



■ *Figure 1b*



■ *Figure 1c*

Le deuxième algorithme d'intégration est l'intégration trapézoïdale. Cet algorithme améliore l'algorithme d'intégration rectangulaire en ajoutant un élément triangulaire à l'approximation rectangulaire, ainsi qu'on peut le voir à la **figure 1b**. L'algorithme d'intégration devient alors :

$Y_t = Y_{t-1} + T \times X_{t-1} + (T/2) \times [X_t - X_{t-1}]$

Amplitude $(\omega T) = (T/2) \times \cos(\omega T/2)$

Phase $(\omega T) = -\Pi/2$.

L'élément triangulaire ajouté correspond au terme $(T/2) \times [X_t - X_{t-1}]$.

Pour simplifier les calculs, on pourra prendre T = 2, on obtient :

$Y_t = Y_{t-1} + X_{t-1} + X_t$.

La valeur Y_t obtenue sera évidemment le double de son homologue rectangulaire. Cette technique d'intégration est très simple et améliore grandement l'algorithme d'intégration rectangulaire.

Le troisième algorithme présenté est

```
        DSEG    AT      08
WORD    EQU     2                               ; l'entrée est sur 2 octets
DWORD   EQU     4                               ; la sortie (l'intégrale) sur 4 octets
MSB     EQU     0                               ; les données 16 bits sont classées
LSB     EQU     1                               ; MSB:LSB
; Réservation de la mémoire nécessaire à la sauvegarde des états antérieurs
XT:     DS      WORD                            ; les valeurs d'entrées sont sur 16bits
YT:     DS      DWORD                           ; les valeurs integrées sont sur 32bits
XT_1:   DS      WORD                            ; valeurs à t-1
YT_1:   DS      DWORD
XT_2:   DS      WORD                            ; valeurs à t-2
YT_2:   DS      DWORD
        CSEG    AT      0
; Met à zéro les valeurs de la zone tampon XT et YT.
; Cette routine doit être appelée avant de procéder à l'intégration de données
ZERO:   MOV     R7,#3*(DWORD+WORD)              ; 3 éléments (2 si trapèze)
        MOV     R0,#XT                          ; début de la zone à effacer
        CLR     A
ZERO0:  MOV     @R0,A
        INC     R0
        DJNZ    R7,ZERO0
        RET
; multiplie R6:R7 par 2
MUL2:   MOV     A,R7
        ADD     A,R7
        MOV     R7,A
        MOV     A,R6
        ADDC    A,R6
        MOV     R6,A
        RET
; déplace les R7 data pointés par R0 vers R1
; R0 et R1 pointent les DERNIERES données des deux buffer.
; Si les zones de départ et d'arrivée ne sont pas disjointes, R0 doit être
; supérieur a R1.
DEPLACEDATA:
        MOV     A,@R0
        MOV     @R1,A
        DEC     R0
        DEC     R1
        DJNZ    R5,DEPLACEDATA
        RET
; Intègre R6:R7 résultat dans YT. La valeur dans R6:R7 ne doit pas dépasser
; une précision de 14 bits. La méthode utilisée est l'intégration
; trapézoïdale. Les valeurs traitées sont signées.
; Y(nT) = Y(nT-2T) + T*X(nT-T) + (T/2)*[X(nT) - X(nT-T)]
; Pour ne pas perdre de la précision, on prend T = 2. L'équation se résume
; alors à : Y(nT) = Y(nT-2T) + X(nT-T) + X(nT)
; Si on veut avoir un résultat compatible avec celui que l'on aurait obtenu
; en faisant une intégration rectangulaire simple (avec T = 1), il faudra
; bien sur diviser le résultat final (YT) par 2.
TRAPEZE:
        MOV     R5,#6                           ; déplace YT -> YT_1
        MOV     R0,#XT+DWORD+WORD-1             ; et XT -> XT_1
        MOV     R1,#XT_1+DWORD+WORD-1
        ACALL   DEPLACEDATA
        MOV     A,R6
        MOV     XT+MSB,A                        ; sauve le MSB de la valeur courante XT
        MOV     A,R7
        MOV     XT+LSB,A                        ; sauve le LSB de XT
        ADD     A,XT_1+LSB                      ; XT + XT_1
        MOV     R7,A
        MOV     A,XT_1+MSB
        ADDC    A,R6
        MOV     R6,A
        MOV     A,YT_1+3                        ; YT_2 + [XT + XT_1]
        ADD     A,R7
        MOV     YT+3,A
        MOV     A,YT_1+2
        ADDC    A,R6
        MOV     YT+2,A
MOV     R5,#0                                   ; teste si négatif
        MOV     A,R6
        JNB     ACC.7,TRAPE0
        MOV     R5,#0FFH
TRAPE0: MOV     A,YT_1+1                        ; additionne le MSB
        ADDC    A,R5                            ; 0 ou 0xFF suivant le signe
        MOV     YT+1,A
        MOV     A,YT_1+0
        ADDC    A,R5
        MOV     YT+0,A
        RET
; Intègre R6:R7 résultat dans YT. La valeur dans R6:R7 ne doit pas dépasser
; une précision de 13 bits. La méthode utilisée est l'intégration de Simpson.
; Il faut éviter d'utiliser cette méthode si le spectre du signal d'entrée
; s'étend au dela de la fréquence d'échantillonnage /4. Dans ce cas, il est
; préférable d'utiliser l'intégration trapèzoïdale ou rectangulaire (ou de
; filtrer le signal à intégrer).
; Y(nT) = Y(nT-2T) + (T/3)*[X(nT) + 4*X(nT-T) + X(nT-2T)]
; Pour ne pas perdre de la precision, T est pris arbitrairement égal à 3
; Si on veut avoir un résultat compatible avec celui que l'on aurait obtenu
; en faisant une integration rectangulaire simple (avec T = 1), il faudra
; bien sur diviser le résultat (YT) par 3.
SIMPSON:
        MOV     R5,#6*2                         ; déplace YT -> YT_1
        MOV     R0,#XT_1+DWORD+WORD-1           ; et YT_1 -> YT-2
        MOV     R1,#XT_2+DWORD+WORD-1           ; ainsi que XT -> XT_1
        ACALL   DEPLACEDATA                     ; et XT_1 -> XT_2
        MOV     A,R6
        MOV     XT+MSB,A                        ; sauve le MSB de la valeur courante XT
        MOV     A,R7
        MOV     XT+LSB,A                        ; sauve le LSB de XT
        ADD     A,XT_2+LSB                      ; XT + XT_2
        MOV     YT+LSB,A                        ; sauve temporairement dans YT
        MOV     A,R6
        ADDC    A,XT_2+MSB
        MOV     YT+MSB,A
        MOV     A,XT_1+LSB                      ; prend XT_1
        MOV     R7,A
        MOV     A,XT_1+MSB
        MOV     R6,A
        CALL    MUL2                            ; multiplie XT_1 par 4
        CALL    MUL2
        MOV     A,YT+LSB                        ; prend XT + XT_2
        ADD     A,R7
        MOV     R7,A                            ; XT + 4*XT_1 + XT_2
        MOV     A,YT+MSB
        ADDC    A,R6
        MOV     R6,A
        MOV     A,YT_2+3                        ; 3*YT_2 + [XT + 4*XT_1 + XT_2]
        ADD     A,R7
        MOV     YT+3,A                          ; sauve dans YT
        MOV     A,YT_2+2
        ADDC    A,R6
        MOV     YT+2,A
        MOV     R5,#0                           ; teste si [XT + 4XT_1 + XT_2] négatif
        MOV     A,R6
        JNB     ACC.7,SIMPS0
        MOV     R5,#0FFH
SIMPS0: MOV     A,YT_2+1                        ; additionne le MSB
        ADDC    A,R5                            ; 0 ou 0xFF suivant le signe
        MOV     YT+1,A
        MOV     A,YT_2+0
        ADDC    A,R5
        MOV     YT+0,A
        RET
```

■ *Listing 1*

l'intégration par la méthode de Simpson. On considère ici que trois points successifs sont sur une parabole. L'équation qui régit cet algorithme est la suivante :

$Y_t = Y_{t-2} + (T/3) \times [X_t - 4 \times X_{t-1} + X_{t-2}]$

Amplitude (ωT)

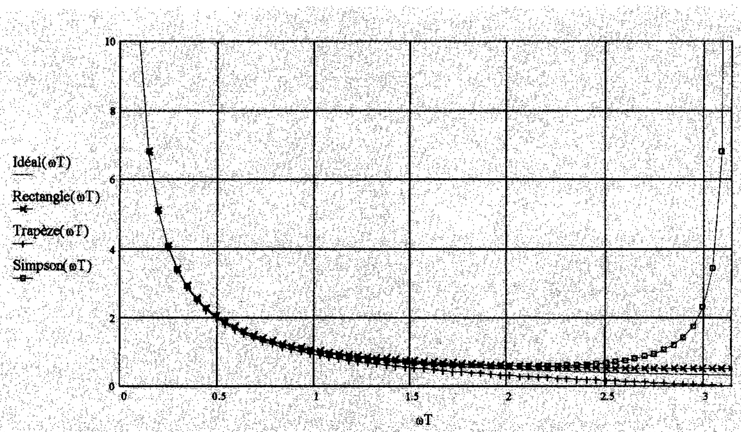
$= (T/3) \times [(2 + \cos(\omega T))/\cos(\omega T)]$

Phase $(\omega t) = - \Pi/2$.

Cette technique approche l'intégrateur idéal d'une manière bien meilleure que les autres méthodes déjà exposées (**figure 1c**). Cependant, l'intégration par la méthode de Simpson n'est utilisable que pour des fréquences inférieures au quart de la fréquence d'échantillonnage. Il faut donc absolument proscrire cette méthode dans le cas d'intégration de signal présentant un bruit important dans ce cas, on préférera l'intégration trapézoïdale.

La **figure 2** montre les réponses en amplitude des différents intégrateurs. On voit clairement qu'au-dessus de $\Pi/2$ (quart de la fréquence d'échantillonnage), la réponse de l'intégrateur de Simpson s'écarte considérablement de la réponse de l'intégrateur idéal.

Le **listing 1** est la réalisation logicielle de l'algorithme d'intégration trapézoïdale et de l'intégration par la méthode de Simpson en utilisant un microprocesseur de type 80C5X.



■ *Figure 2*

J.L VERN