CALCUL DE DÉRIVÉE SUR µC

Le calcul de la dérivée d'une fonction est une des opérations les plus couramment utilisées. Elle est particulièrement utile pour déterminer le sens de variation d'une fonction ou pour évaluer ses maxima et minima. Dans ce dernier cas le maximum ou minimum de la fonction sera atteint quand la dérivée de celle-ci changera de signe. On utilise aussi le calcul de la dérivée dans de nombreux processus de régulation, où l'évaluation de celle-ci permet d'anticiper la correction à effectuer (PID).

La dérivée est une fonction. Sa valeur, en un point donné, est égale à la pente de la droite tangente à ce point de la courbe définie par la fonction. Dans un processus numérique, la fonction dont on cherche la dérivée est une suite de valeurs discrètes. Ces valeurs sont quantifiées, souvent sur des mots de huit à seize bits.

L'algorithme de dérivée le plus simple (algorithme «deux points») consiste à soustraire la valeur la plus récente à la valeur précédente :
Dérivée $_t = (F_t - F_{t-1})/T$. Cette opération est très rapide et nécessite uniquement à chaque instant la mémorisation de la valeur précédente de la fonction. Le résultat donné par cet algorithme n'est souvent pas satisfaisant pour plusieurs raisons :
la variation d'un échantillon à un autre est souvent très faible, de l'ordre de quelques bits ; prenons par exemple le cas d'un signal sinusoïdal échantillonné sur huit bits avec 256 échantillons par période. Les échantillons auront les valeurs : F(i) = Partie entière(127. sin(i . 2 . π / 256)).
A i = 0, la valeur de la dérivée est maximale et est égale à : F(1) - F(0) = 3 (au coefficient 1/T près). Sur toute l'étendue du signal, la valeur de la dérivée sera donc de -3, -2, -1, 0, 1, 2, ou 3. Ce manque de résolution sur le calcul de la dérivée entraînera une instabilité importante dans les systèmes de régulation qui utilisent cet algorithme ; pour les mêmes raisons, la détermination du sommet d'une courbe sera aussi malaisée.
Comme dans le cas d'un dérivateur idéal, le gain augmente avec la fréquence du signal, ce qui tend à augmenter le bruit au détriment des fréquences basses du signal qui sont généralement celles qui nous intéressent.
Un deuxième algorithme («trois points»), aussi simple que le précédent, établit la dérivée en un point comme étant la différence entre la valeur de l'échantillon suivant et de l'échantillon précédent :
Dérivée $_{t-1} = (F_t - F_{t-2})/2T$. Une fois mémorisées les deux valeurs précédentes, on effectue la soustraction de la valeur courante avec l'avant dernière valeur sauvegardée. La réponse en fréquence de cet algorithme s'ap-

$T := 1 \qquad \omega T := 0, \frac{2\cdot\pi}{64} .. \pi$

Réponse en amplitude :

$\mathrm{Idéal}(\omega T) := \omega T$

$\mathrm{DeuxPoints}(\omega T) := \frac{2}{T}\cdot\sin\left(\frac{\omega T}{2}\right)$

$\mathrm{TroisPoints}(\omega T) := \frac{1}{T}\cdot\sin(\omega T)$

$\mathrm{CinqPoints}(\omega T) := \frac{2}{10}\cdot(\sin(\omega T) + 2\cdot\sin(2\cdot\omega T))$

Réponse en phase :

$\mathrm{PhaseDeuxPoints}(\omega T) := \frac{\pi}{2} - \frac{\omega T}{2}$

$\mathrm{PhaseTroisPoints}(\omega T) := \frac{\pi}{2} - \omega T$

$\mathrm{PhaseCinqPoints}(\omega T) := \frac{\pi}{2} - 2\cdot\omega T$

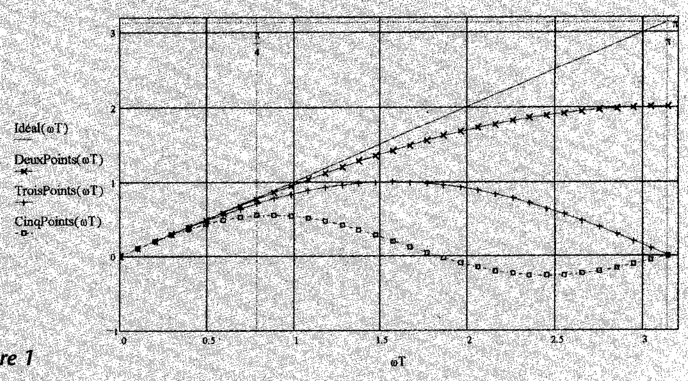

Figure 1

proche de la réponse d'un dérivateur à amplificateur opérationnel auquel on a ajouté une cellule RC intégrateur pour améliorer la stabilité. La sensibilité au bruit est réduite, mais cet algorithme ne différencie bien des signaux que pour des valeurs jusqu'à environ $\pi/6$ radians, soit des fréquences de : $(\pi/6).F_{échantillonnage}/2\pi$ $= F_{échantillonnage}/12$. La résolution (ici le nombre de valeurs discrètes que peut prendre la dérivée) est un peu plus élevée qu'avec l'algorithme précédent ; en prenant le même exemple, la dérivée peut prendre ici toutes les valeurs de -6 à +6.

Le troisième algorithme présenté est le plus utilisé («parabolique cinq points»). Il suppose que les points de part et d'autre du point pour lequel on désire calculer la dérivée sont situés sur une parabole. En utilisant un ajustement polynomial puis en dérivant le résultat on trouve l'algorithme suivant :

Dérivée $_{t-2}$ =

$1/10(-2.F_{t-4} - F_{t-3} + F_{t-1} + 2.F_t)$

où F_t est la valeur courante de la fonction dont on veut la dérivée, F_{t-1} la valeur précédente, ainsi de suite. Le point central pour lequel on a évalué la dérivée est le point F_{t-2}. Ici, on a utilisé 5 points pour l'ajustement polynomial. Bien que l'approximation par rapport au dérivateur idéal ne soit pas aussi bon que dans le cas des algorithmes précédents, cet algorithme a l'avantage de diminuer le bruit haute fréquence normalement associé au processus de différentiation. De plus, le nombre de points intervenant dans le calcul étant plus élevé, la résolution sur la valeur de la dérivée est plus élevée. Enfin, le gain de cet algorithme n'est pas lié à la période T d'échantillonnage. On peut prendre plus de 5 points pour calculer la dérivée. Les coefficients pour 7, 9 et 11 points sont les suivants :

5 points : 1/10 (-2,-1,1,2)
7 points : 1/28 (-3,-2,-1,1,2,3)
9 points : 1/60 (-4,-3,-2,-1,1,2,3,4)
11 points : 1/110 (-5,-4,-3,-2,-1,1,2,3,4,5).

Plus le nombre de points est élevé, plus le lissage est amélioré ; en contrepartie le temps de calcul est plus long, la réponse aux hautes fréquences moins linéaire, etc.

La **figure 1** représente la réponse en amplitude des trois algorithmes décrits. La représentation est faite en coordonnées linéaires et non pas logarithmiques comme il est d'usage dans le domaine analogique. La valeur $\omega T = \pi$ correspond à une fréquence égale à la moitié de la fréquence d'échantillonnage. Le marqueur $\pi/4$ correspond donc à une fréquence huit fois inférieure à la fréquence d'échantillonnage. La **figure 2** représente l'erreur sur la réponse en amplitude des trois méthodes sur l'intervalle $[0..\pi/4]$ exprimée en décibels. Le listing est le codage en assembleur 8051 de l'algorithme «parabolique 5 points».

```
LNGFILE EQU         8*2             ; file de 8 valeurs 16 bits
        DSEG        AT    08
FILE:   DS          LNGFILE         ; FILE doit être à une adresse PAIRE
PTRFILE: DS         1               ; index de la valeur courante
        CSEG        AT    0
        JMP         START           ; vecteur de reset

; PUTFILE met la valeur R6:R7 dans la file circulaire FILE de longueur
; LNGFILE. LNGFILE doit être une puissance de 2. Les valeurs stockés dans
; la file sont sur 16 bits. LES VALEURS SONT STOCKEES LSB PUIS MSB. PUTFILE
; doit être appelée AVANT de calculer la derivée avec DERIVE.
PUTFILE:
        MOV         A,PTRFILE       ; index sur la valeur courante
        ADD         A,#2            ; mot de 16 bits
        ANL         A,#LNGFILE-1    ; LNGFILE est une puissance de 2
        MOV         PTRFILE,A       ; nouvelle valeur de l'index
        ADD         A,#FILE         ; calcule le pointeur sur la valeur
        MOV         R0,A
        MOV         A,R7
        MOV         @R0,A           ; transfere le LSB
        INC         R0              ; attention, #FILE doit etre PAIR
        MOV         A,R6
        MOV         @R0,A           ; puis le MSB
        RET

; Additionne à R6:R7 la A ième valeur de la FILE. A doit être pair.
ADDVAL: ADD         A,PTRFILE
        ANL         A,#LNGFILE-1    ; calcule l'index sur la Aième valeur
        ADD         A,#FILE         ; pointeur sur la valeur dans la FILE
        MOV         R0,A
        MOV         A,@R0           ; prend le LSB
        ADD         A,R7            ; additionne au LSB de R6:R7
        MOV         R7,A
        INC         R0              ; pointe sur le MSB
        MOV         A,@R0
        ADDC        A,R6            ; complète l'addition 16 bits
        MOV         R6,A
        RET

; Soustrait à R6:R7 la A ième valeur de la FILE. A doit être pair.
SUBVAL: ADD         A,PTRFILE
        ANL         A,#LNGFILE-1    ; calcule l'index sur la Aième valeur
        ADD         A,#FILE         ; pointeur sur la valeur dans la FILE
        MOV         R0,A
        CLR         C
        MOV         A,R7            ; prend le LSB
        SUBB        A,@R0           ; soustrait le LSB de la valeur pointée
        MOV         R7,A
        INC         R0              ; pointe sur le MSB
        MOV         A,R6            ; prend le MSB
        SUBB        A,@R0           ; complète la soustraction 16 bits
        MOV         R6,A
        RET

; MUL2 multiplie R6:R7 par deux
MUL2:   MOV         A,R7
        ADD         A,R7
        MOV         R7,A
        MOV         A,R6
        ADDC        A,R6
        MOV         R6,A
        RET

; Calcule la dérivée de la fonction dont les dernières valeurs sont contenues
; dans la file FILE en utilisant une interpolation parabolique 5 points.
; Pour ne pas perdre de résolution, la division par 10 finale n'est pas
; effectuée. Le résultat est dans R6:R7. Le gain de DERIVE est de 10, aussi
; on peut appliquer DERIVE à des données de 12 bits. Si on modifie DERIVE pour
; utiliser l'algorithme 7 points, celui-ci ayant un gain de 28, on ne pourra
; traiter que des données codées au maximum sur 11 bits.
DERIVE: CLR         A
        MOV         R6,A            ; R6:R7 = 0
        MOV         R7,A
        CALL        ADDVAL          ; dans R6:R7 la valeur à [t0]
        MOV         A,#-4*2         ; prépare l'add. de la valeur à [t-4]
        CALL        SUBVAL          ; [t0] - [t-4]
        CALL        MUL2            ; 2[t0] - 2[t-4]
        MOV         A,#-1*2         ; prépare l'add. de la valeur à [t-1]
        CALL        ADDVAL          ; 2[t0] + [t-1] - 2[t-4]
        MOV         A,#-3*2         ; prépare la soust. de la val. à [t-3]
        JMP         SUBVAL          ; 2[t0] + [t-1] + [t-3] - 2[t-4]
```

J.-L. VERN

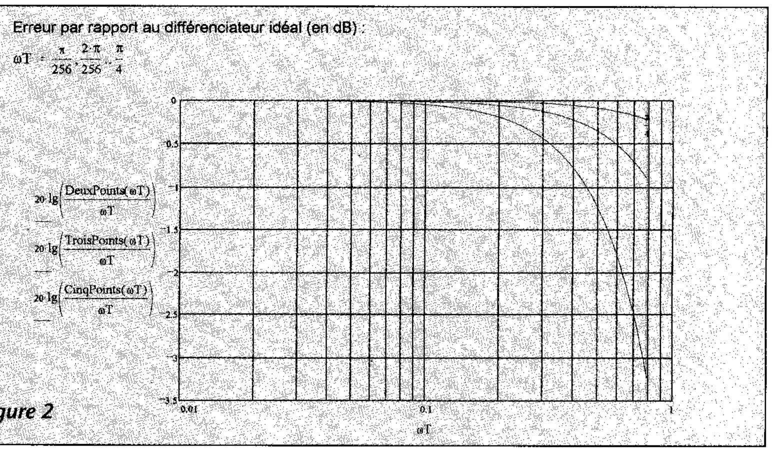

Figure 2