SYNTHÈSE DE LA FONCTION ARC TANGENTE

Parmi les problèmes qui nécessitent l'utilisation de fonctions trigonométriques, nombreux sont ceux qui font appel aux fonctions fondamentales (sinus, cos, tan). Dans certains cas, on désire connaître un angle en fonction de projections, par exemple dans le cas de calculs ayant trait à la navigation ou lors de l'élaboration de capteurs angulaires ou de boussoles. Dans ce cas, une fonction primordiale est la fonction arc tangente qui permet de déterminer un angle x à partir des valeurs des deux projections sin(x) et cos(x).

```
COEF_A      EQU         0C914h; 8003H
COEF_B      EQU         41ADh ; 29CFH
; Calcule la valeur de atg(x) avec x = (R6:R7 / 10000h) pour une valeur de x
; positive variant de 0 à 0,25 soit R6:R7 variant de 0000h a 4000h. _atg025
; renvoie dans R6:R7 le résultat avec un facteur d'échelle de (10000h / 2 X PI)
; Ainsi par exemple atg(0.125) = 0.124354    (en radian)
; d'ou R6:R7 = 0.125 X 10000h = 2000h et _atg025(2000h) = 511h = 1297
; et 1297 X 2 X PI / 10000h = 0.124348, résultat approché de atg(0.125)
_atg025:
        MOV     A,R7            ; sauve x
        MOV     R3,A            ; pour les prochaines multiplications
        MOV     ACCU32+1,A      ; et comme numérateur de la division
        MOV     A,R6
        MOV     R2,A
        MOV     ACCU32+0,A
        MOV     R5,#LOW COEF_B
        MOV     R4,#HIGH COEF_B
        CALL    _mul16          ; B X x
        MOV     R6,AR2
        MOV     R7,AR3          ; restaure x
        CALL    _mul16          ; x X (B X x) = (x X x) X B
        MOV     A,R5
        ADD     A,#LOW COEF_A   ; calcule A + ((x X x) X B)
        MOV     R7,A
        MOV     A,R4
        ADDC    A,#HIGH COEF_A
        MOV     R6,A
        CLR     A               ; efface les bits inutilisés
        MOV     R4,A
        MOV     R5,A
        MOV     ACCU32+2,A
        MOV     ACCU32+3,A
        CALL    _div32          ; calcule x / (A + ((x X x) X B))
        MOV     A,ACCU32+3      ; ajoute 4 avant de diviser par 8
        ADD     A,#4            ; pour faire un arrondi juste
        MOV     R6,A
        MOV     A,ACCU32+2
        ADDC    A,#0
        RRC     A
        XCH     A,R6
        RRC     A
        XCH     A,R6
        RRC     A
        XCH     A,R6
        RRC     A
        XCH     A,R6
        RRC     A
        ANL     A,#1FH
        XCH     A,R6
        RRC     A
        MOV     R7,A
        RET
```

■ *Listing 1*

Supposons par exemple que nous désirions réaliser une boussole électronique ; nous utiliserons deux (dans le cas où on ne tient pas compte de la composante verticale) capteurs sensibles au champ magnétique terrestre (par exemple des flux gate ou des capteurs Philips KMZ 10). Ces capteurs seront disposés perpendiculairement l'un par rapport à l'autre, de manière à ce que la valeur restituée par ceux-ci soit maximum sur un des capteurs quand elle est minimum sur l'autre et réciproquement. Si on appelle a et b la tension relevée aux bornes des capteurs on aura :

$a = C1 . \sin(x)$

$b = C2 . \cos(x)$

avec C1 et C2 facteurs liés à l'amplification donnée par le circuit associé au capteurs. Si C1 = C2, le rapport entre les deux tensions issues des capteurs sera égal à :

$a/b = \sin(x)/\cos(x) = \tan(x)$.

La valeur de l'angle x sera donc

x = arc tangente (a/b) - à 180° près -.

Il faut donc calculer arc tangente (x). Comme pour le sinus, il existe un développement en série limitée **(tableau 1)**

1) $\tan(x) = \dfrac{\sin(x)}{\cos(x)}$

2) $\arctan(x) = x - \dfrac{x^3}{3!} + \dfrac{x^5}{5!} - \dfrac{x^7}{7!} + \ldots$

3) $\arctan(x) = \cfrac{x}{1 + \cfrac{x^2}{3\left(1 + \cfrac{4x^2}{3*5\left(1 + \cfrac{9x^2}{5*7\left(1 + \cfrac{16x}{7*9(1+\ldots)}\right)}\right)}\right)}}$

4) $\arctan(x) = \dfrac{\pi}{2} - \arctan\left(\dfrac{1}{x}\right)$

5) $\arctan(x) = \arctan(k) + \arctan\left(\dfrac{x-k}{1+kx}\right)$

■ *Tableau 1*

qui permet d'approcher arc tangente. Malheureusement, ce développement converge très lentement et est donc inutilisable (il faut plusieurs millions de termes pour obtenir une précision de huit chiffres sur le calcul de arc tangente (1)). Il existe un autre développement sous forme fractionnaire beaucoup plus intéressant (3) car il converge très rapidement. En arrêtant le développement à

Ordre 1 : $\arctan(x) = x$

Ordre 3 : $\arctan(x) = \dfrac{3x}{3 + x^2}$

Ordre 5 : $\arctan(x) = \dfrac{x\,(15 + 4x^2)}{15 + 9x^2}$

Ordre 7 : $\arctan(x) = \dfrac{x\,(105 + 55x^2)}{105 + 90x^2 + 9x^4}$

Ordre 9 : $\arctan(x) = \dfrac{x\,(945 + 735x^2 + 64x^4)}{945 + 1050x^2 + 225x^4}$

■ ***Tableau 2 : Approximation d'arc tangente.***

un endroit donné, puis en simplifiant l'équation, on obtient les approximations données dans le **tableau 2**. Plus l'ordre est important, plus la précision est grande et plus les calculs sont compliqués. Pour améliorer la précision et diminuer les calculs, il est nécessaire de réduire la gamme des valeurs pour lesquelles on calcule arc tangente. L'équation (3) permet déjà de réduire ces valeurs à l'intervalle 0..1. À la valeur 0 correspond un angle 0°, à la valeur 1 un angle de 45°, si on désire calculer arc tangente (2), on calculera arc tangente (1/2) et on soustraira la valeur trouvée à PI/2 où 90° si on travaille en degrés. L'équation (4) permet de réduire encore la valeur de calcul de arc tangente à un intervalle aussi petit que l'on désire. Un bon choix si on désire une précision de 16 bits sur les angles et de choisir de calculer arc tangente pour une valeur d'entrée variant de 0 à 0,25 en utilisant l'approximation d'ordre 3. Suivant les intervalles de la valeur d'entrée, on prendra pour k les valeurs suivantes :

k = 1/4 pour les valeurs d'entrée de 0,25 à 0,5,
k = 1/2 pour les valeurs d'entrée de 0,5 à 0,75
k = 3/4 pour les valeurs d'entrée de 0,75 à 1

Ce choix des valeurs de k permet de simplifier les calculs en remplaçant les multiplications par de simples décalages. Le **listing 1** utilise la technique de réduction des valeurs d'entrée pour calculer arc tangente pour des valeurs d'entrée variant de 0 à 1 (1 exclu). Les calculs s'effectuant en entier sur 16 où 32 bits, les valeurs d'entrée ne sont pas représentées sous forme flottante. La valeur 1 est codée par 10000h, 0,5 est codée par 8000h, etc. De même, les angles sont représentés de manière «naturelle», 0 représentant 0° ou 360°, 8000h représentant 180°, 4000h : 90°... Cette formulation tient compte de la nature circulaire de l'appréhensibilité des angles, la valeur E000h représentant également la valeur 270° et -90°.

Le **listing 2** montre le calcul arc tangente pour des valeurs inférieures à 0,25. Pour en revenir à notre boussole, dans la plupart des cas, on désire calculer arc tangente du rapport de deux valeurs. Le listing 3 (sur le serveur ERP uniquement) réalise une telle fonction avec des valeurs positives. Les angles renvoyés par la fonction varient entre 0 et 90°. Les fonctions multiplications 16 X 16 et division 32 / 32 utilisées dans les listings 1, 2 et 3 ont été exposées dans les articles des mois précédents et sont disponibles, ainsi que les listings du présent article, sur le serveur ERP.

J.L VERN.

```
        EXTRN   CODE        (_div32, _atg025)
        EXTRN   DATA        (ACCU32)
; Calcul de arctangente pour une valeur de x < 1. Cette fonction utilise
; _atg025 qui calcule arctangente pour des valeurs < a 0.25. L'identité
; suivante est utilisée pour le calcul :
; atg(x) = atg(k) + atg((x-k)/(1 + kx))
; la valeur 1 est représentée par 10000H. La valeur 0.25 est donc 4000h,
; la valeur 0.5 est 8000h etc.
; x est passé et retourné dans R6:R7. L'angle renvoyé dans R6:R7 est codé
; avec un facteur d'echelle de 2^16 / 2 PI (ou 2^16 / 360 si on est en degrés)
; La valeur 0 degrés est donc codée 0, 45 degrés : 2000h, 90 degres : 4000h etc
; Pour des valeurs supérieures a 1 (angles > 45) on utilisera l'identité
; suivante : atg(x) = 90 - atg(1/x).
; Pour les valeurs négatives atg(-x) = -atg(x)
ATG0    EQU         0
ATG025 EQU          0FB09H          ; atg(0.25) X 2^16 / (2 X PI) (LSB-MSB)
ATG05   EQU         0E412h
ATG075 EQU          0381AH
        RSEG        ?CO?_atg0a1?ATG0A1
TABATG: DW          ATG0,ATG025,ATG05,ATG075
        RSEG        ?PR?_atg0a1?ATG0A1
_atg0a1:
        MOV         A,R6            ; détermine k en fonction de x
        ANL         A,#0C0H
        MOV         R4,A
        PUSH        ACC
        JZ          _atg04          ; R6:R7 < 4000h correspond à x < 0.25
        MOV         A,R7            ; calcule x - k
        ADD         A,R7            ; comme k = 0.25, 0.5, ou 0.75
        MOV         ACCU32+1,A      ; suivant les valeurs de k, le calcul de x - k
        MOV         A,R6            ; revient à calculer la x & B'0011111111111111'
        ADDC        A,R6            ; En multipliant x par 4, on effectue la même
        MOV         ACCU32+0,A      ; opération en changeant en plus le facteur
        MOV         A,ACCU32+1      ; d'échelle pour augmenter la précision
        ADD         A,ACCU32+1
        MOV         ACCU32+1,A
        MOV         A,ACCU32+0
        ADDC        A,ACCU32+0
        MOV         ACCU32+0,A
; calcule 1 + k X x avec k = .25, .5 ou .75. Comme le facteur d'échelle est de
; 4, il faut multiplier x par 1, 2, ou 3.
        MOV         A,R6            ; R2:R3 = R6:R7
        MOV         R2,A
        MOV         A,R7
        MOV         R3,A
        MOV         R5,#0
_atg01: CJNE        R4,#040h,_atg02
        JMP         _atg03
_atg02: MOV         A,R3            ; R5:R6:R7 = R5:R6:R7 + R2:R3
        ADD         A,R7
        MOV         R7,A
        MOV         A,R2
        ADDC        A,R6
        MOV         R6,A
        MOV         A,R5
        ADDC        A,#0
        MOV         R5,A
        MOV         A,R4
        SUBB        A,#040h
        MOV         R4,A
        JMP         _atg01
_atg03: ORL         AR5,#04         ; ajoute 1 (en fait 10000h X 4)
        CLR         A               ; met à 0 les octets inutiles dans le calcul
        MOV         R4,A
        MOV         ACCU32+2,A
        MOV         ACCU32+3,A
        CALL        _div32          ; ACCU32 = ACCU32 / R4:R5:R6:R7
        MOV         A,ACCU32+2
        MOV         R6,A
        MOV         A,ACCU32+3
        MOV         R7,A            ; R6:R7 = (x - k) / (1 + (k X x))
_atg04: CALL        _atg025         ; calcule atg de la nouvelle valeur
        POP         ACC
        RRC         A
        SWAP        A
        ANL         A,#06H
        ADD         A,#LOW TABATG
        MOV         DPL,A
        MOV         A,#HIGH TABATG
        ADDC        A,#0
        MOV         DPH,A
        CLR         A
        MOVC        A,@A+DPTR
        ADD         A,R7            ; et ajoute la constante pré-calculée
        MOV         R7,A
        INC         DPTR
        CLR         A
        MOVC        A,@A+DPTR
        ADDC        A,R6
        MOV         R6,A
        RET
```

■ ***Listing 2***