

Angle Recoding 2. Wu

2. AR (Angle Recode)

20180904 Tue

Copyright (c) 2015 - 2018 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

① AR [Hu]

skip certain micro rotations

the rotation sequence $\mu(i) = \{-1, 0, +1\}$

$\mu(i) = 0 \rightarrow \text{skip}$

desire to minimize

$$\sum_{i=0}^N |\mu(i)|$$

so that the total number of CORPIC iterations can be minimized

Angle Recoding \leftarrow Multiplier Recoding

angle recoding method for efficient implementation of the CORDIC algorithm
Hu & Naganathan, ISCAS 89

Greedy algorithm

the angle quantization error

$$\xi_{m, AR} \equiv \theta - \sum_{i=0}^M \mu(i) a(i)$$

Skipping rotations

$$\mu(i) = 0$$

null operations

$$\begin{cases} x(i+1) = x(i) \\ y(i+1) = y(i) \end{cases}$$

keep the iteration number N the same

in the AQ perspective, remove these steps
and use different index variable

"index variable conversion"

the iteration number N becomes N'

the number of subangles N_A of EAS

$N_A \leftarrow N'$ not fixed but varying

Optimization Problem

given a rotation angle θ
find rotation sequence $\mu(i) \in \{-1, 0, +1\}$
for $0 \leq i \leq N-1$

Such that angle quantization error

$$|\xi_{m,AR}| < \alpha(N-1)$$

and the effective iteration number

$$N' = \sum_{i=0}^N |\mu(i)| \quad \text{is minimized}$$

$$\theta(0) = \theta, \{ \mu(i) = 0, 0 \leq i \leq N-1 \}, k=0$$

repeat until $|\theta(k)| < a(N-1) \rho_0$

Choose $i_k, 0 \leq i_k \leq N-1$

$$| |\theta(k)| - a(i_k) | = \underset{0 \leq i \leq N-1}{\text{Min}} | |\theta(k)| - a(i) |$$

$$\theta(k+1) = \theta(k) - \mu(i_k) a(i_k)$$

$$\mu(i_k) = \text{Sign}(\theta(k))$$

AQ & AR

$$\begin{aligned}\xi_{m, AR} &\equiv \theta - \sum_{i=0}^{N-1} \mu(i) a(i) && N && AR \\ &= \theta - \left[\sum_{j=0}^{N-1} \tan^{-1}(\alpha(j) \cdot 2^{-s(j)}) \right] && N' && AQ \\ &= \theta - \left[\sum_{j=0}^{N-1} \tilde{\theta}(j) \right] && N' && \end{aligned}$$

$$N' \triangleq \sum_{i=0}^{N-1} |\mu(i)|$$

the effective transition number
count non-zero rotation

$s(j) \in \{0, 1, \dots, N-1\}$ the rotational sequence
determines the micro-rotation angle
in the j -th iteration

$\alpha(j) \in \{-1, 0, +1\}$ the directional sequence
 $\{-1, +1\}$ controls the direction of
(after sampling) the j -th micro-rotation of $a(s(j))$

$\tilde{\theta}(j)$ the j th micro-rotation angle
$$\tilde{\theta}(j) = \tan^{-1}(\alpha(j) \cdot 2^{-s(j)})$$

AR essentially tries to approximate θ
 with the combination of **selected** angle elements
 from a **pre-defined elementary angle set (EAS)**.

the EAS consists of **all** possible values of $\tilde{\theta}(j)$'s

the EAS S_i in AR

$$S_i = \left\{ \tan^{-1}(\alpha^* \cdot 2^{-s^*}) : \alpha^* \in \{-1, 0, +1\}, s^* \in \{0, 1, \dots, N-1\} \right\}$$

$$\begin{aligned} \xi_{m, AR} &\equiv \theta - \sum_{i=0}^{N-1} \mu(i) a(i) \\ &= \theta - \left[\sum_{j=0}^{N-1} \tan^{-1}(\alpha(j) \cdot 2^{-s(j)}) \right] \\ &= \theta - \left[\sum_{j=0}^{N-1} \tilde{\theta}(j) \right] \end{aligned}$$

$$\xi_m \triangleq \theta - \sum_{i=0}^{N_A-1} \theta_i$$

AR performs AQ of the target angle θ

the sub-angle θ_i becomes $\tilde{\theta}(i) = \tan^{-1}(\alpha(i) \cdot 2^{-s(i)})$

$$N_A \longrightarrow N' \triangleq \sum_{i=0}^{N-1} |\mu(i)|$$

the effective transition number

given a rotation angle θ

AR

find rotation sequence $\mu(i) \in \{-1, 0, +1\}$
for $0 \leq i \leq N-1$

AQ

find the combination of elementary angles
from \mathcal{S}_1

Such that angle quantization error

$$|\xi_{m,AR}| < \alpha(N-1)$$

and the effective iteration number

$$N' = \sum_{i=0}^N |\mu(i)| \quad \text{is minimized}$$

EAS S_1 AR

elementary angle value

- $r(1) = \text{atan}(-2^{\{-0\}})$
- $r(2) = \text{atan}(-2^{\{-1\}})$
- $r(3) = \text{atan}(-2^{\{-2\}})$
- $r(4) = \text{atan}(-2^{\{-3\}})$
- $r(5) = \text{atan}(-2^{\{-4\}})$
- $r(6) = \text{atan}(-2^{\{-5\}})$
- $r(7) = \text{atan}(-2^{\{-6\}})$
- $r(8) = \text{atan}(-2^{\{-7\}})$
- $r(9) = \text{atan}(0)$
- $r(10) = \text{atan}(2^{\{-7\}})$
- $r(11) = \text{atan}(2^{\{-6\}})$
- $r(12) = \text{atan}(2^{\{-5\}})$
- $r(13) = \text{atan}(2^{\{-4\}})$
- $r(14) = \text{atan}(2^{\{-3\}})$
- $r(15) = \text{atan}(2^{\{-2\}})$
- $r(16) = \text{atan}(2^{\{-1\}})$
- $r(17) = \text{atan}(2^{\{-0\}})$

N_A

N conventional CORDIC

```
>> mu = [1, 0, 0, -1, 0, 0, -1, -1, 0, 0, 0, 1, 0, 0, 0, 1]
>> length(mu)
ans = 16
>> s = [ 0: 15]
>> atan(1)
ans = 0.78540
>> pi/4
ans = 0.78540
>> sum(atan(2.^(-s)) .* mu)
ans = 0.63813
>> 13 * pi / 32
ans = 1.2763
```

i	Conventional CORDIC	only $\{-1, +1\}$	Angle Recode	skip allowed $\{-1, 0, +1\}$
0	$S(0) = 0$	$\alpha(0) = +1$	$S(0) = 0$	$\alpha(0) = +1$
1	$S(1) = 1$	$\alpha(1) = -1$	$S(1) = 1$	$\alpha(1) = 0$
2	$S(2) = 2$	$\alpha(2) = +1$	$S(2) = 2$	$\alpha(2) = 0$
3	$S(3) = 3$	$\alpha(3) = +1$	$S(3) = 3$	$\alpha(3) = -1$
4	$S(4) = 4$	$\alpha(4) = -1$	$S(4) = 4$	$\alpha(4) = 0$
5	$S(5) = 5$	$\alpha(5) = +1$	$S(5) = 5$	$\alpha(5) = 0$
6	$S(6) = 6$	$\alpha(6) = -1$	$S(6) = 6$	$\alpha(6) = -1$
7	$S(7) = 7$	$\alpha(7) = -1$	$S(7) = 7$	$\alpha(7) = -1$
8	$S(8) = 8$	$\alpha(8) = +1$	$S(8) = 8$	$\alpha(8) = 0$
9	$S(9) = 9$	$\alpha(9) = -1$	$S(9) = 9$	$\alpha(9) = 0$
10	$S(10) = 10$	$\alpha(10) = -1$	$S(10) = 10$	$\alpha(10) = 0$
11	$S(11) = 11$	$\alpha(11) = +1$	$S(11) = 11$	$\alpha(11) = +1$
12	$S(12) = 12$	$\alpha(12) = +1$	$S(12) = 12$	$\alpha(12) = 0$
13	$S(13) = 13$	$\alpha(13) = -1$	$S(13) = 13$	$\alpha(13) = 0$
14	$S(14) = 14$	$\alpha(14) = +1$	$S(14) = 14$	$\alpha(14) = 0$
15	$S(15) = 15$	$\alpha(15) = +1$	$S(15) = 15$	$\alpha(15) = +1$

	Angle		Removing skipped angle	
i	Recode			
0	$S(0) = 0$	$\alpha(0) = +1$	0	$S(0) = 0$ $\alpha(0) = +1$
1	$S(1) = 1$	$\alpha(1) = 0$	1	
2	$S(2) = 2$	$\alpha(2) = 0$	2	
3	$S(3) = 3$	$\alpha(3) = -1$	3	$S(1) = 3$ $\alpha(1) = -1$
4	$S(4) = 4$	$\alpha(4) = 0$	4	
5	$S(5) = 5$	$\alpha(5) = 0$	5	
6	$S(6) = 6$	$\alpha(6) = -1$	6	$S(2) = 6$ $\alpha(2) = -1$
7	$S(7) = 7$	$\alpha(7) = -1$	7	$S(3) = 7$ $\alpha(3) = -1$
8	$S(8) = 8$	$\alpha(8) = 0$	8	
9	$S(9) = 9$	$\alpha(9) = 0$	9	
10	$S(10) = 10$	$\alpha(10) = 0$	10	
11	$S(11) = 11$	$\alpha(11) = +1$	11	$S(4) = 11$ $\alpha(4) = +1$
12	$S(12) = 12$	$\alpha(12) = 0$	12	
13	$S(13) = 13$	$\alpha(13) = 0$	13	
14	$S(14) = 14$	$\alpha(14) = 0$	14	
$w-1 = 15$	$S(15) = 15$	$\alpha(15) = +1$	15	$S(5) = 15$ $\alpha(5) = +1$
				$N-1$

i	Conventional CORDIC	only $\{-1, +1\}$	Angle Quant	Skip allowed $\{-1, 0, +1\}$
0	$S(0) = 0$	$\alpha(0) = +1$	$S(0) = 0$	$\alpha(0) = +1$
1	$S(1) = 1$	$\alpha(1) = -1$	$S(1) = 3$	$\alpha(1) = -1$
2	$S(2) = 2$	$\alpha(2) = +1$	$S(2) = 6$	$\alpha(2) = -1$
3	$S(3) = 3$	$\alpha(3) = +1$	$S(3) = 7$	$\alpha(3) = -1$
4	$S(4) = 4$	$\alpha(4) = -1$	$S(4) = 11$	$\alpha(4) = +1$
5	$S(5) = 5$	$\alpha(5) = +1$	$S(5) = 15$	$\alpha(5) = +1$
6	$S(6) = 6$	$\alpha(6) = -1$		
7	$S(7) = 7$	$\alpha(7) = -1$		
8	$S(8) = 8$	$\alpha(8) = +1$		
9	$S(9) = 9$	$\alpha(9) = -1$		
10	$S(10) = 10$	$\alpha(10) = -1$		
11	$S(11) = 11$	$\alpha(11) = +1$		
12	$S(12) = 12$	$\alpha(12) = +1$		
13	$S(13) = 13$	$\alpha(13) = -1$		
14	$S(14) = 14$	$\alpha(14) = +1$		
15	$S(15) = 15$	$\alpha(15) = +1$		

W

N'

```

#include <stdio.h>
#include <math.h>

#define N 16

int find_index(double a[], double atheta) {
    int i, ik = N;
    double minval = 1.0e+10;

    for (i=0; i<N; ++i) {
        if (minval > fabs(atheta - a[i])) {
            ik = i;
            minval = fabs(atheta - a[i]);
        }
    }

    return ik;
}

int main(void) {
    double a[N];
    double theta = 0.63761;
    int ik, uik;
    int k, i;
    int u[N];
    double angle;

    for (i=0; i<N; ++i)
        a[i] = atan(1./pow(2, i));

    for (i=0; i<N; ++i)
        u[i] = 0;

```

```

k = 0;
while ((fabs(theta) >= a[N-1]) && (k < N)) {
    ik = find_index(a, fabs(theta));

    uik = (theta >= 0) ? +1 : -1;

    printf("k=%2d theta=%10.7f ", k, theta);
    printf("ik=%2d uik=%+d ", ik, uik);

    theta = theta - uik * a[ik];

    printf("a[%2d]=%10.7f, new theta=%10.7f \n",
        ik, a[ik], theta);

    u[ik] = uik;

    k++;
}

angle = 0.0;
for (i=0; i<N; ++i) {
    angle += u[i] * a[i];

    printf("i=%2d u[%2d]=%+d a[%2d]=%f angle=%f \n",
        i, i, u[i], i, a[i], angle);
}
}

```

k= 0 theta= 0.6376100 ik= 0 uik=+1 a[0]= 0.7853982, new theta=-0.1477882
k= 1 theta=-0.1477882 ik= 3 uik=-1 a[3]= 0.1243550, new theta=-0.0234332
k= 2 theta=-0.0234332 ik= 5 uik=-1 a[5]= 0.0312398, new theta= 0.0078067
k= 3 theta= 0.0078067 ik= 7 uik=+1 a[7]= 0.0078123, new theta=-0.0000057
i= 0 u[0]=+1 a[0]=0.785398 angle=0.785398
i= 1 u[1]=+0 a[1]=0.463648 angle=0.785398
i= 2 u[2]=+0 a[2]=0.244979 angle=0.785398
i= 3 u[3]=-1 a[3]=0.124355 angle=0.661043
i= 4 u[4]=+0 a[4]=0.062419 angle=0.661043
i= 5 u[5]=-1 a[5]=0.031240 angle=0.629803
i= 6 u[6]=+0 a[6]=0.015624 angle=0.629803
i= 7 u[7]=+1 a[7]=0.007812 angle=0.637616
i= 8 u[8]=+0 a[8]=0.003906 angle=0.637616
i= 9 u[9]=+0 a[9]=0.001953 angle=0.637616
i=10 u[10]=+0 a[10]=0.000977 angle=0.637616
i=11 u[11]=+0 a[11]=0.000488 angle=0.637616
i=12 u[12]=+0 a[12]=0.000244 angle=0.637616
i=13 u[13]=+0 a[13]=0.000122 angle=0.637616
i=14 u[14]=+0 a[14]=0.000061 angle=0.637616
i=15 u[15]=+0 a[15]=0.000031 angle=0.637616