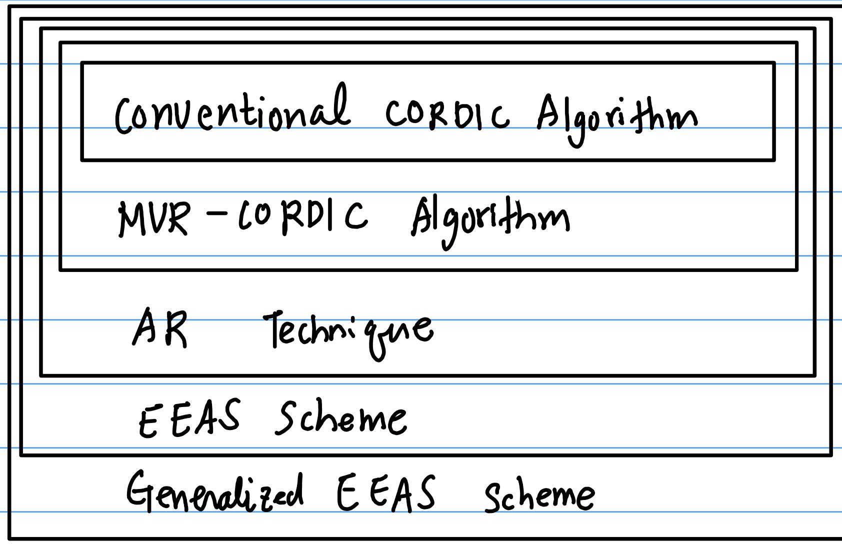
Angle Recoding CORDIC
2. Wu

20180825 Sat

Vector Rotational CORDIC



Vector Rotation Alg	Selection of Rotation Sequence	Elementary Angle Set	Micro Rotation	Angle Quantization θ_i	Angle Quantization N_A
conventional CORDIC	$\mu = \{-1, +1\}$	EAS S	complete	$\mu(i)\, a(i)$	W Fixed
Angle Recoding	$\mu = \{-1, 0, +1\}$	EAS S_1	selective	$\tan^{-1}(\alpha(i) \cdot 2^{-s(i)})$	N' Variable
MVR-CORDIC	$\alpha = \{-1, 0, +1\}$	EAS S_1	selective	$\tan^{-1}(\alpha(i) \cdot 2^{-s(i)})$	R_m Fixed
EEAS	$\alpha_1, \alpha_2 = \{-1, 0, +1\}$	EEAS S_2	selective	$\tan^{-1}(\alpha_0(i) \cdot 2^{-s_0(i)} + \alpha_1(i) \cdot 2^{-s_1(i)})$	R_m Fixed
Generalized EEAS	$\alpha_1, \alpha_2, \cdots, \alpha_{d-1} = \{-1, 0, +1\}$	EEAS S_d $d \geq 3$	selective	$\tan^{-1}(\alpha_0(i) \cdot 2^{-s_0(i)} \ldots + \alpha_{d-1}(i) \cdot 2^{-s_{d-1}(i)})$	R_m Fixed

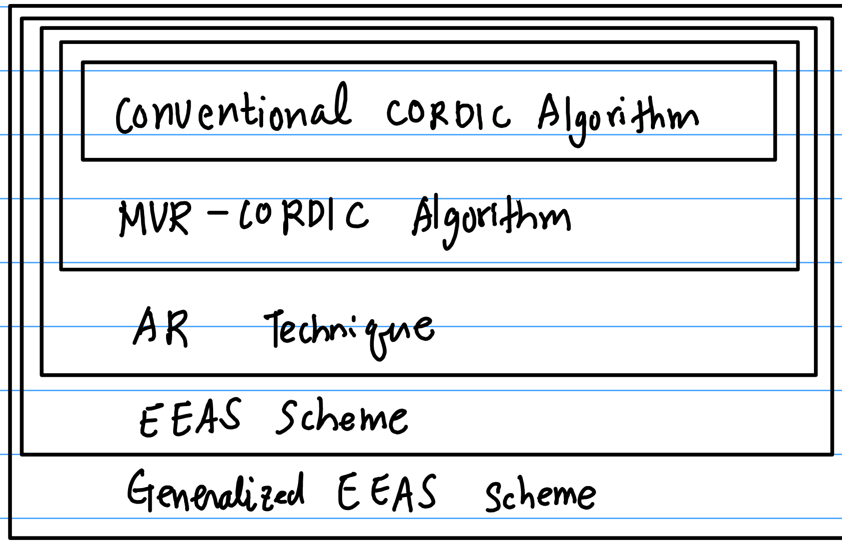
Family of Vector Rotational CORDIC

AQ process (Angle Quantization) —
- CORDIC
- AR (Angle Recoding)
- MVR - CORDIC (Modified Vector Rotation)
- EEAS (Extended Elementary Angle Set)
- Generalized EEAS

AQ process with various EAS and and suitable combinations of subangles

to decompose the target rotational angle into several easy-to-implement subangles

minimizing the angle quantization error ξ_m to obtain the best precision performance



EEAS covers { MVR-CORDIC
AR

a subset of EEAS $\hat{S}_2$ — EAS S_1

MVR-CORDIC — a subset of AR

one constraint on the iteration number

Angle Quantization

Quantization process on the rotational angle θ

decompose the original rotational angle θ
into severales θ_i's

sum up those subangles to approximate
the original angle as close as possible

Minimize the angle quantization error

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

N_A : the number of sub-angles

$$\theta = \theta_0 + \theta_1 + \theta_2 + \cdots + \theta_{N_A-1} + \xi_m$$

design issues in the AQ process

① need to determine the sub-angles θ_i

each θ_i needs to be easy-to-implement

② how to select and combine these sub-angles ξ_m

such that the angle quantization error ξ_m

can be minimized

Angle Quantization

Quantization process on the rotational angle θ

decompose θ into several subangles θ_i's

the angle quantization error

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

N_A the number of subangles
$\theta_0, \theta_1, \cdots, \theta_{N_A-1}$

$$\theta = \theta_0 + \theta_1 + \cdots + \theta_{N_A-1} + \xi_m$$

data : W-bit word length

the iteration number : N $\qquad N \leqslant W$

the restricted iteration number : R_m $\qquad R_m \ll W$

Vector Rotation CORDIC Family

⓪ Conventional CORDIC

① AR

② MVR

③ EEAS

Angle Quantization

Quantization process on the rotational angle θ

decompose θ into several subangles θ_i's

the angle quantization error

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

N_A the number of subangles

$\theta_0, \theta_1, \cdots, \theta_{N_A-1}$

$$\theta = \theta_0 + \theta_1 + \cdots + \theta_{N_A-1} + \xi_m$$

data : W-bit word length		
the iteration number : N	$N \leqslant W$	
the restricted iteration number : R_m	$R_m \ll W$	

CSD (Canonical Signed Digit) Quantization

digital filter designs

coefficients are recoded

in terms of SPT (Signed Power of Two) terms

multiplication can be easily realized

with shift-and-add operations

$$h_2 = (-0.156249)_{10} \Rightarrow (0.0\bar{1}011)_2$$

$w = 8$, 3 non-zero digits

① CSD quantization decomposes

coefficients into several SPT terms

(sub-coefficients)

② the multiplication of a coefficient

can be reformed

through the combination of

the non-zero SPT sub-coefficients

quantize the rotation angle θ

decompose the rotation angle θ
into several sub-angles θ_i's

the rotational operation of each θ_i
should be easily realized

If each θ_i can be realized
using only shift-and-add operations

the rotation of θ can be performed
through successive applications of
sub-angle rotations
in a cost-effective way

Approximation target	Coefficient h_i	Rotation angle θ
Basic Element	Non-zero digit 2^{-i}	Sub-angle $a(i) = \tan^{-1}(2^{-i})$
Basic Operation	shift-and-add operation	2 shift-and-add operations
Approximation Equation	$h_i \approx \sum_{j=0}^{N_D-1} g_j \cdot 2^{-d_j}$	$\theta \approx \sum_{j=0}^{R_m-1} \alpha(j) \cdot a(s(j))$
	$g_j \in \{-1, 0, +1\}$ $d_j \in \{0, 1, \cdots, W-1\}$ $N_D =$ the number of non-zero digits	$N_A =$ the number of sub-angles

try to approach the target rotation angle θ
step by step

decisions are made in each step
by choosing the best combination of $\alpha(i)\, a(s(i))$

So as to minimize $|\xi_m|$

$\alpha(i)$, $a(i)$ are determined such that
the error function is minimized

$$J(i) = |\theta(i) - \alpha(i)\, a(s(i))|$$

$$\theta(i) = \theta - \sum_{m=0}^{i-1} \alpha(m)\, a(s(m))$$

terminated if no further improvement can be found

$$J(i) \geqslant J(i-1)$$

or $\alpha(R_m - 1)$ and $s(R_m - 1)$
are determined at the end

Rotation Angle $\theta = \frac{13\pi}{64}$

Conventional CORDIC

$\bar{\mu} = [1 \ -1 \ 1 \ 1 \ -1 \ 1 \ -1 \ -1 \ 1 \ -1 \ -1 \ 1 \ 1 \ -1 \ 1 \ 1]$

Angle Reioding - Greedy

$\bar{\mu} = [1 \ 0 \ 0 \ -1 \ 0 \ 0 \ -1 \ -1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1]$

MVR - CORDIC - Greedy

$\bar{\alpha} = [1 \ -1 \ -1 \ -1]$

$\bar{s} = [0 \ 3 \ 6 \ 7]$

MVR - CORDIC - Semi Greedy $(P=2)$

$\bar{\alpha} = [1 \ -1 \ -1 \ 1]$

$\bar{s} = [0 \ 3 \ 5 \ 7]$

MVR - CORDIC - TBS

$\bar{\alpha} = [1 \ 1 \ -1 \ -1]$

$\bar{s} = [1 \ 2 \ 4 \ 7]$

EEAS - Greedy

$\bar{\alpha}_0 = [1 \ -1]$ $\bar{\alpha}_1 = [-1 \ -1]$

$\bar{s}_0 = [0 \ 2]$ $\bar{s}_1 = [8 \ 10]$

EEAS - TBS $R_m = 2$

$\bar{\alpha}_0 = [1 \ 1]$ $\bar{\alpha}_1 = [-1 \ -1]$

$\bar{s}_0 = [0 \ 6]$ $\bar{s}_1 = [3 \ 5]$

EEAS - TBS $R_m = 3$

$\bar{\alpha}_0 = [1 \ -1 \ 1]$ $\bar{\alpha}_1 = [-1 \ 1 \ -1]$

$\bar{s}_0 = [0 \ 3 \ 7]$ $\bar{s}_1 = [15 \ 6 \ 2]$

```
>> mu = [1, -1, 1, 1, -1, 1, -1, -1, 1, -1, -1, 1, 1, -1, 1, 1]
>> length(mu)
ans =  16
>> s =[ 0: 15]
>> atan(1)
ans =  0.78540
>> pi/4
ans =  0.78540
>> sum(atan(2.^(-s)) .* mu)
ans =  0.63815
>> 13 * pi / 32
ans =  1.2763
```

```
>> 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
```

```
>> s = [0, 3, 6, 7]
>> alpha = [1, -1, -1, -1]
>> sum(atan(2.^(-s) .* alpha))
ans =  0.63761
>>
```

```
>> s = [0, 3, 5, 7]
>> alpha = [1, -1, -1, 1]
>> sum(atan(2.^(-s) .* alpha))
ans =  0.63762
```

```
>> alpha = [1, 1, -1, -1]
>> s = [1, 2, 4, 7]
>> sum(atan(2.^(-s) .* alpha))
ans =  0.63840
```