

Angle Recoding 2. Wu

3. MVR

20180828 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".

② MVR (Modified Vector Rotational)

two modifications

① **repetition** of elementary angles

each micro-rotation of elementary angle
can be performed repeatedly

- more possible combinations
- smaller ξ_m

② **confinement** of total micro-rotation number

confine the iteration number

in the micro-rotation phase
to R_m ($R_m \ll W$)

The role of R_m is quite similar
to the **number of non-zero digit**
 N_D in CSD recoding scheme

the angle quantization error

$$\xi_{m, \text{MVR}} \triangleq \theta - \sum_{i=0}^{R_m-1} \alpha(i) a(s(i))$$

the rotational sequence

$$s(i) \in \{0, 1, \dots, W-1\}$$

the micro-rotation angle
in the i -th iteration

the directional sequence

$$\alpha(i) \in \{-1, 0, +1\}$$

the direction of the i -th
micro-rotation of $a(s(i))$

$$\alpha(i) a(s(i)) = \tilde{\theta}(j)$$

the rotational sequence

$$s(i) \in \{0, 1, \dots, W-1\} \quad [0, 3, 6, 7]$$

the directional sequence

$$\alpha(i) \in \{-1, 0, +1\} \quad [1, -1, -1, 1]$$

$$\text{atan}(2^0) - \text{atan}(2^{-3}) - \text{atan}(2^{-6}) + \text{atan}(2^{-7})$$

$$\alpha(i) \alpha(s(i)) = \tilde{\theta}(j)$$

MVR-CORDIC Algorithm with $R_n = 4$	Greedy Algorithm	3	$\bar{\alpha} = [1 \ -1 \ -1 \ -1]$ $\bar{s} = [0 \ 3 \ 6 \ 7]$	$5.2891 \cdot 10^{-4}$
	Semi-greedy Algorithm ($D = 2$)	4	$\bar{\alpha} = [1 \ -1 \ -1 \ 1]$ $\bar{s} = [0 \ 3 \ 5 \ 7]$	$5.2033 \cdot 10^{-4}$
	TBS Algorithm	5	$\bar{\alpha} = [1 \ 1 \ -1 \ -1]$ $\bar{s} = [1 \ 2 \ 4 \ 7]$	$2.5911 \cdot 10^{-4}$

```
>> 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
```

```
0 7.85398163397448e-01
1 4.63647609000806e-01
2 2.44978663126864e-01
3 1.24354994546761e-01
4 6.24188099959574e-02
5 3.12398334302683e-02
6 1.56237286204768e-02
7 7.81234106010111e-03
8 3.90623013196697e-03
9 1.95312251647882e-03
10 9.76562189559319e-04
11 4.88281211194898e-04
12 2.44140620149362e-04
13 1.22070311893670e-04
14 6.10351561742088e-05
15 3.05175781155261e-05
```

$w=16$

0	7.85398163397448e-01	s(0)
1	4.63647609000806e-01	.
2	2.44978663126864e-01	.
3	1.24354994546761e-01	s(1)
4	6.24188099959574e-02	.
5	3.12398334302683e-02	.
6	1.56237286204768e-02	s(2)
7	7.81234106010111e-03	s(3)
8	3.90623013196697e-03	.
9	1.95312251647882e-03	.
10	9.76562189559319e-04	.
11	4.88281211194898e-04	.
12	2.44140620149362e-04	.
13	1.22070311893670e-04	.
14	6.10351561742088e-05	.
15	3.05175781155261e-05	.

0	7.85398163397448e-01	s(0)
3	1.24354994546761e-01	s(1)
6	1.56237286204768e-02	s(2)
7	7.81234106010111e-03	s(3)

$R_m = 4$

0	7.85398163397448e-01
1	4.63647609000806e-01
2	2.44978663126864e-01
3	1.24354994546761e-01
4	6.24188099959574e-02
5	3.12398334302683e-02
6	1.56237286204768e-02
7	7.81234106010111e-03
8	3.90623013196697e-03
9	1.95312251647882e-03
10	9.76562189559319e-04
11	4.88281211194898e-04
12	2.44140620149362e-04
13	1.22070311893670e-04
14	6.10351561742088e-05
15	3.05175781155261e-05

AQ & MVR CORDIC

$$\xi_{m, MVR} \triangleq \theta - \left[\sum_{j=0}^{R_m-1} \alpha(j) a(s(j)) \right]$$

the rotational sequence $s(j)$

$$j = 0, 1, 2, \dots, R_m-1$$



$$s(j) \in \{0, 1, \dots, W-1\} \quad \text{rotational sequence}$$

determines the micro-rotation angle $a(s(j))$
in the j -th iteration

the directional sequence $\alpha(j)$

$$\alpha(j) \in \{-1, 0, +1\}$$

controls the direction of the j -th
micro-rotation of $a(s(j))$

$$\alpha(j) a(s(j)) = \tilde{\theta}(j)$$

$$\begin{array}{l} i = 0, 1, 2, 3, \dots, W-1 \\ s(j) = 0, 1, 2, 3, \dots, W-1 \quad \text{rotational sequence} \\ \alpha(j) = -1, 0, 0, +1, \dots, -1 \quad \text{directional sequence} \\ j = 0, -, -, 1, \dots, R_m-1 \quad \text{effective iteration number} \\ R_m \ll W \end{array}$$

sub-angle $(\alpha(j) a(s(j))) \sim \tilde{\theta}(j)$

$$\xi_{m,AR} = \theta - \left[\sum_{j=0}^{N'-1} \tan^{-1}(\alpha(j) \cdot 2^{-s(j)}) \right]$$
$$= \theta - \left[\sum_{j=0}^{N'-1} \tilde{\theta}(j) \right], \quad \tilde{\theta}(j) = \tan^{-1}(\alpha(j) \cdot 2^{-s(j)})$$

$$N' \triangleq \sum_{j=0}^{N-1} |\mu(j)| \quad \text{the effective iteration number}$$

EAS formed by MVR-CORDIC
is the same as AR
also performs AQ

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

the EAS S_1 in AR

$$S_1 = \{ \tan^{-1}(\alpha^* \cdot 2^{-s^*}) : \alpha^* \in \{-1, 0, +1\}, s^* \in \{0, 1, \dots, N-1\} \}$$

The major difference

1) the total number of sub-angles N_A

the total iteration number

in the micro-rotation phase

is kept fixed to a pre-defined value of R_m

$$N_A = R_m$$

2) the sub-angle θ_i corresponds to $\alpha^{(j)} a(s^{(j)})$

$$\theta_j = \alpha^{(j)} a(s^{(j)}) = \tilde{\theta}_j$$

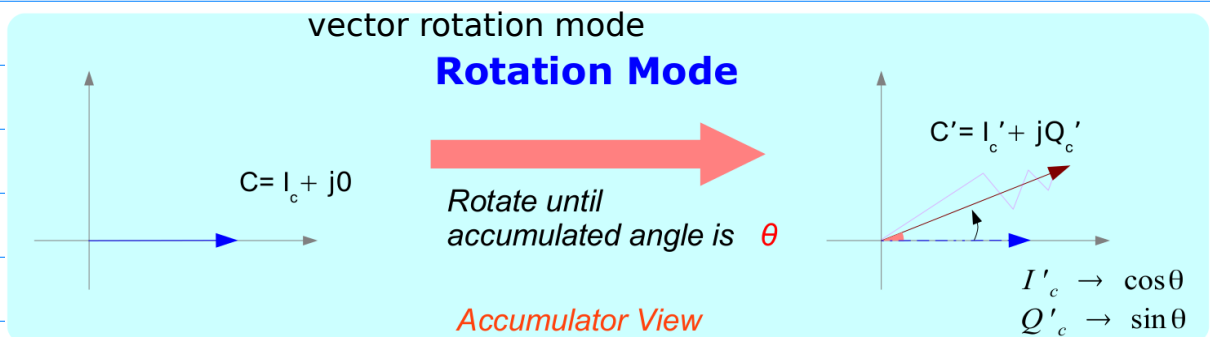
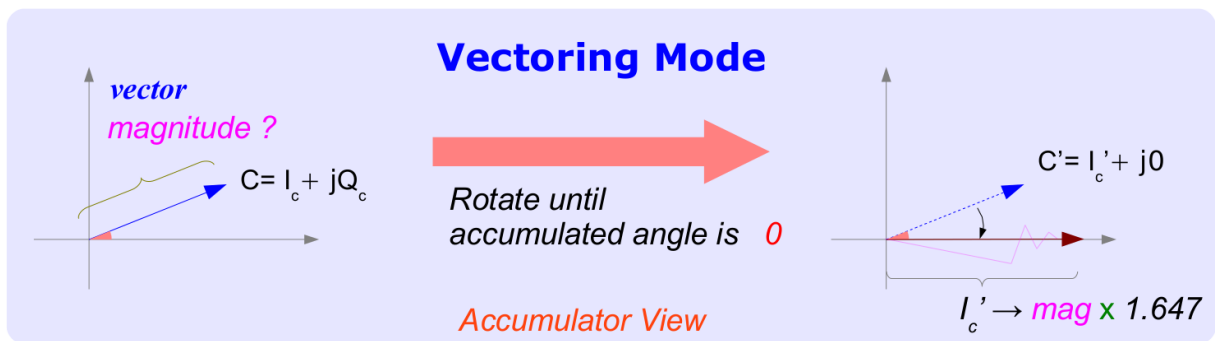
MVR (Modified Vector Rotation)

1) Repeat of Elementary Angles θ_i, θ_i

2) fixed total micro-rotation Number R_m

* Vector Rotation Mode

* and the rotation angles are known in advance



Modified Vector Rotational MVR CORDIC

- reduce the iteration number
- maintaining the SQNR performance
- modifying the basic microrotation procedure

Three Searching Algorithm

- ① the selective prerotation
- ② the selective scaling
- ③ iteration-tradeoff scheme

Optimization Problem

EAS point of view

Given θ , find the combination of R_m elementary angles from EAS S_i , such that the angle quantization error $|\xi_{m, \text{MUR}}|$ is minimized.

Semi-greedy algorithm

trade offs between computational complexities and performance

Key issue in the MVR-CORDIC
is to find the best sequences of
 $s(i)$ and $\alpha(i)$ to minimize $|\xi_m|$
subject to the constraint that
the total iteration number is confined to R_m

- 1) Greedy Algorithm
- 2) Exhaustive Algorithm
- 3) Semigreedy Algorithm

1) Greedy Algorithm

try to approach the target rotation angle, θ , step by step
in each step, decisions are made on $\alpha(i)$ and $s(i)$
by choosing the best combination of $\alpha(i)$ and $s(i)$
so as to minimize $|\xi_m|$

$\alpha(i)$ and $s(i)$ are determined such that
the error function $J(i) = |\theta(i) - \alpha(i) a(s(i))|$ is minimized

$\theta(i)$: the residue angle in the i -th step

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

the searching is terminated

if no further improvements can be found

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

$\alpha(R_m-1)$ and $s(R_m-1)$ are determined

at the end of the searching

the greedy algorithm terminates

Only when the residue angle error

cannot be further reduced.

Initialization:

given θ , w , R_m

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

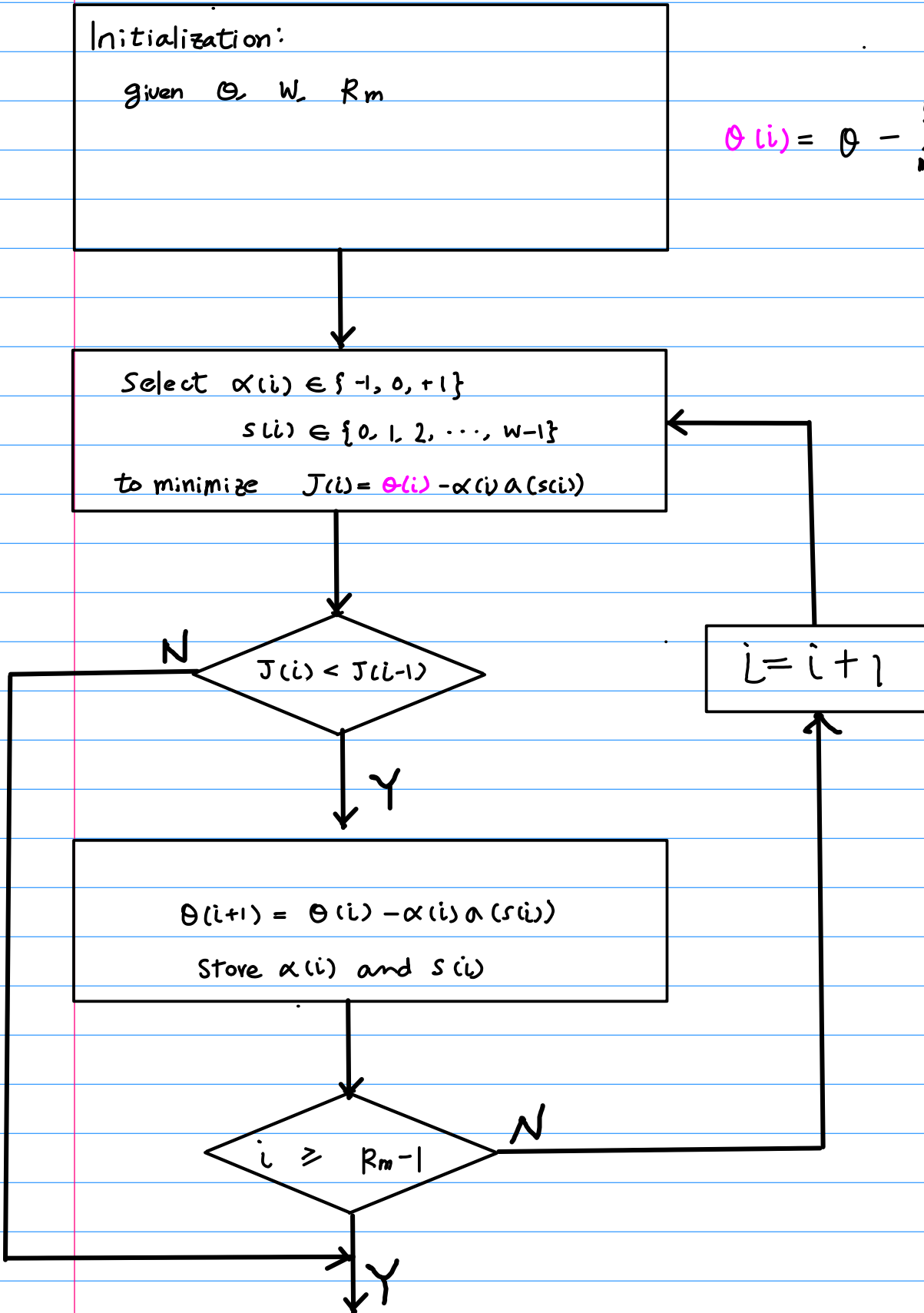
Select $\alpha(i) \in \{-1, 0, +1\}$
 $s(i) \in \{0, 1, 2, \dots, w-1\}$
to minimize $J(i) = \theta(i) - \alpha(i) a(s(i))$

N
Y
 $J(i) < J(i-1)$

$\theta(i+1) = \theta(i) - \alpha(i) a(s(i))$
Store $\alpha(i)$ and $s(i)$

N
Y
 $i \geq R_m - 1$

$i = i + 1$



2) Exhaustive Algorithm

search for the entire solution space

all possible combinations of

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

in a single step

decisions for $\alpha(i)$ and $s(i)$, $0 \leq i \leq R_m-1$

by minimizing the error function

$$J = \left| 0 - \sum_{i=0}^{R_m-1} \alpha(i) a(s(i)) \right|$$

global optimal solution

Initialization:

given Θ, W, R_m

let $\theta(0) = \Theta,$

$i = 0$

$J(-1) = \infty$

Select $\alpha(i) \in \{-1, 0, +1\}$

$s(i) \in \{0, 1, 2, \dots, W-1\}$

for $0 \leq i \leq R_m - 1$

to minimize $J(i) = \theta - \sum_{l=0}^{R_m-1} \alpha(l) a(s(i))$

$$(3 \cdot W) \cdot (3 \cdot W) \dots (3 \cdot W) \\ = 3^{R_m} \cdot W^{R_m}$$

Store $\alpha(i)$ and $s(i)$

for $0 \leq i \leq R_m - 1$

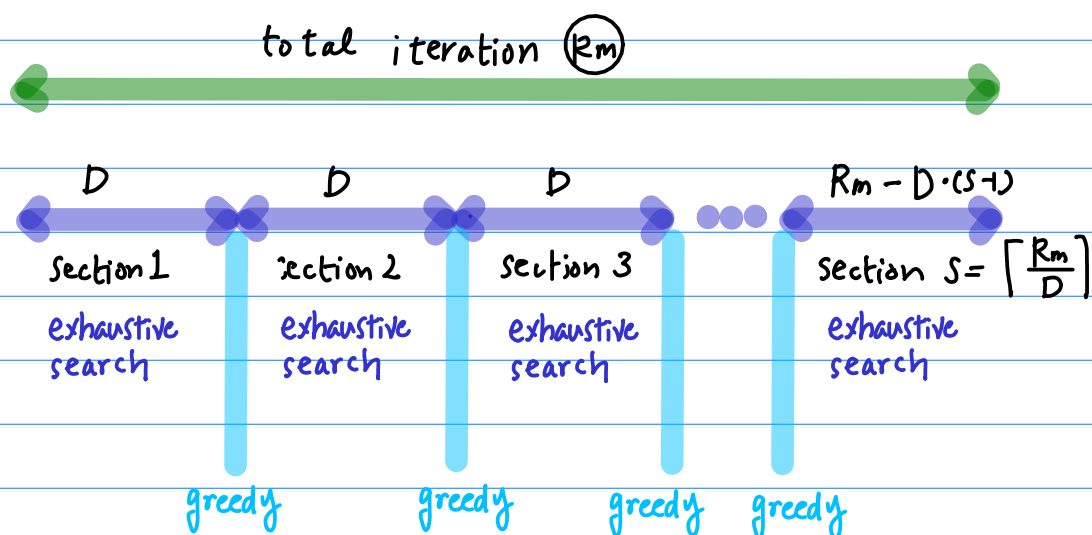
3) Semi-greedy Algorithm

a combination of greedy and exhaustive algorithm

the search space of $\alpha(i)$ and $s(i)$ for $0 \leq i \leq R_m - 1$
are divided into several sections

with D iterations as a segment
↓
block length ↓
 block

the segmentation scheme



in the i -th block

decision of $\alpha(k)$ and $s(k)$ for $iD \leq k \leq (i+1)D-1$

$$\text{minimizes } J = \left| \theta(i) - \sum_{k=iD}^{(i+1)D-1} \alpha(k) a(s(k)) \right|$$

$$\text{where } \theta(i) = \theta - \sum_{m=0}^{i-1} \left[\sum_{k=mD}^{(m+1)D-1} \alpha(k) a(s(k)) \right]$$

the residue angle in the i -th step

$$s = \left\lceil \frac{R_m}{D} \right\rceil$$

$$\theta(i) = \theta - \left[\sum_{k=0D}^{1D-1} \alpha(k) a(s(k)) + \sum_{k=1D}^{2D-1} \alpha(k) a(s(k)) + \dots + \sum_{k=(i-1)D}^{iD-1} \alpha(k) a(s(k)) \right]$$

Initialization:

given θ, W, R_m

let $\theta(0) = \theta,$

$i = 0$

$J(-1) = \infty$

Select $\alpha(k) \in \{-1, 0, +1\}$

$s(k) \in \{0, 1, 2, \dots, W-1\}$

for $iD \leq k \leq (i+1)D - 1$

to minimize $J(i) = \theta(i) - \sum_{k=iD}^{(i+1)D-1} \alpha(k) a(s(k))$

N
Decision: $J(i) < J(i-1)$

$\theta(i+1) = \theta(i) - \sum_{k=iD}^{(i+1)D-1} \alpha(k) a(s(k))$

Store $\alpha(k), s(k)$

Decision: $i \geq \lceil \frac{R_m}{D} \rceil - 1$
N

$i = i + 1$

Y

