CORDIC ALGORITHM AND IMPLEMENTATIONS

- CORDIC METHOD
- ROTATION AND VECTORING MODE
- CONVERGENCE, PRECISION AND RANGE
- SCALING FACTOR AND COMPENSATION
- IMPLEMENTATIONS: word-serial and pipelined
- EXTENSION TO HYPERBOLIC AND LINEAR COORDINATES
- UNIFIED DESCRIPTION
- REDUNDANT ADDITION AND HIGH RADIX

MAIN USES

- REALIZATION OF ROTATIONS
- CALCULATION OF TRIGONOMETRIC FUNCTIONS
- CALCULATION OF INVERSE TRIGONOMETRIC FUNCTION $\tan^{-1}(a/b)$
- CALCULATION OF $\sqrt{a^2 + b^2}$, etc.
- EXTENDED TO HYPERBOLIC FUNCTIONS
- DIVISION AND MULTIPLICATION
- CALCULATION OF SQRT, LOG, AND EXP
- FOR LINEAR TRANSFORMS, DIGITAL FILTERS, AND SOLVING LIN. SYSTEMS
- MAIN APPLICATIONS: DSP, IMAGE PROCESSING, 3D GRAPHICS, ROBOTICS.

CORDIC ALGORITHM

- CIRCULAR COORDINATE SYSTEM

- PERFECT ROTATION:

$$\begin{aligned} x_R &= M_{in}\cos(\beta+\theta) = x_{in}\cos\theta - y_{in}\sin\theta \\ y_R &= M_{in}\sin(\beta+\theta) = x_{in}\sin\theta + y_{in}\cos\theta \end{aligned}$$

- M_{in} – THE MODULUS OF THE VECTOR
- β – THE INITIAL ANGLE

- IN MATRIX FORM:

$$\begin{bmatrix} x_R \\ y_R \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix} = ROT(\theta) \begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix}$$

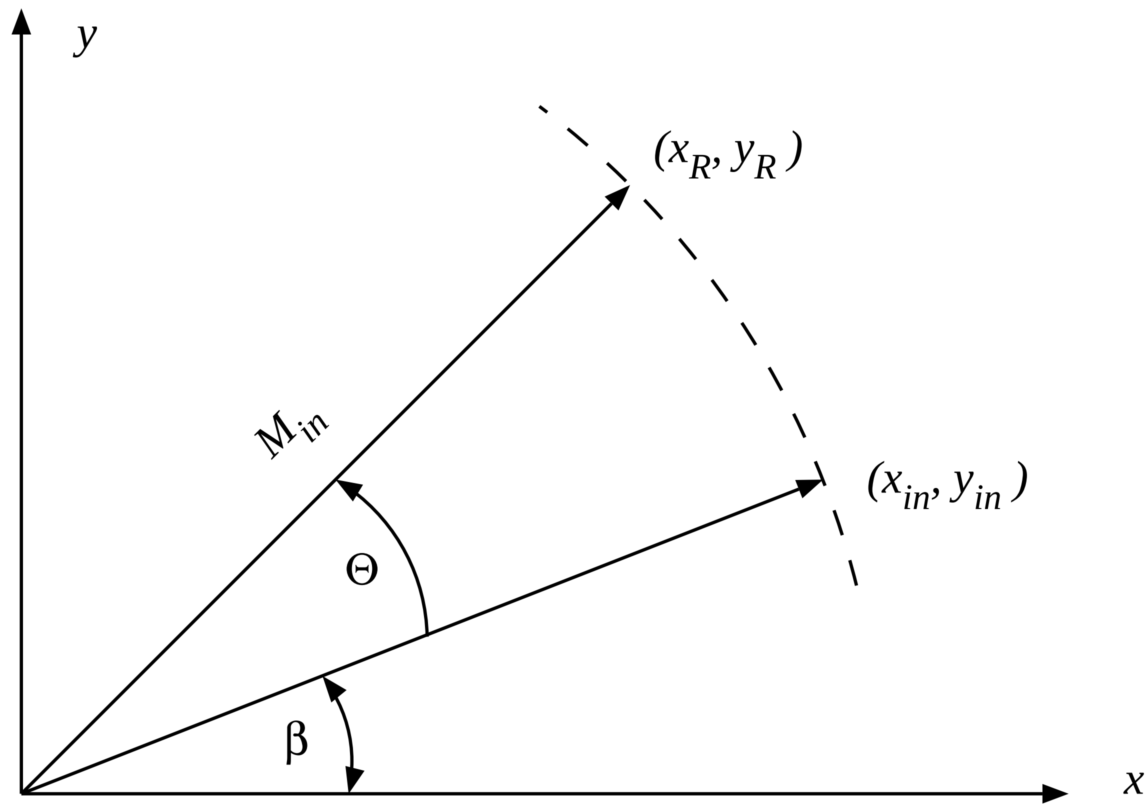

Figure 11.1: VECTOR ROTATION

MICRO-ROTATIONS

- USE ELEMENTARY ROTATION ANGLES α_j

- DECOMPOSE THE ANGLE θ:

$$\theta = \sum_{j=0}^{\infty} \alpha_j$$

SO THAT

$$ROT(\theta) = \prod_{j=0}^{\infty} ROT(\alpha_j)$$

- THEN $ROT(\alpha_j)$:

$$\begin{aligned} x_R[j+1] &= x_R[j]\cos(\alpha_j) - y_R[j]\sin(\alpha_j) \\ y_R[j+1] &= x_R[j]\sin(\alpha_j) + y_R[j]\cos(\alpha_j) \end{aligned}$$

SIMPLIFYING MICRO-ROTATIONS

- HOW TO AVOID MULTIPLICATIONS?

1. DECOMPOSE ROTATION INTO:

   SCALING OPERATION AND ROTATION-EXTENSION

$$\begin{aligned} x_R[j+1] &= \cos(\alpha_j)(x_R[j] - y_R[j]\tan(\alpha_j)) \\ y_R[j+1] &= \cos(\alpha_j)(y_R[j] + x_R[j]\tan(\alpha_j)) \end{aligned}$$

2. CHOOSE ELEMENTARY ANGLES

$$\alpha_j = \tan^{-1}(\sigma_j(2^{-j})) = \sigma_j \tan^{-1}(2^{-j})$$

WITH $\sigma_j \in \{-1, 1\}$

RESULTS IN ROTATION-EXTENSION RECURRENCE WITHOUT MPYs

$$\begin{aligned} x[j+1] &= x[j] - \sigma_j 2^{-j} y[j] \\ y[j+1] &= y[j] + \sigma_j 2^{-j} x[j] \end{aligned}$$

$\Longrightarrow$ONLY ADDITIONS AND SHIFTS

ROTATION-EXTENSION (cont.)

- ROTATION-EXTENSION SCALES MODULUS $M[j]$

$$M[j+1] = K[j]M[j] = \frac{1}{\cos \alpha_j} M[j] = (1+\sigma_j^2 2^{-2j})^{1/2} M[j] = (1+2^{-2j})^{1/2} M[j]$$

- TOTAL SCALING FACTOR

$$K = \prod_{j=0}^{\infty} (1 + 2^{-2j})^{1/2} \approx 1.6468$$

CONSTANT, INDEPENDENT OF THE ANGLE

- RECURRENCE FOR DECOMPOSITION/ACCUMULATION OF ANGLE:

$$z[j+1] = z[j] - \alpha_j = z[j] - \sigma_j \tan^{-1}(2^{-j})$$

IMPLEMENTATION OF CORDIC ITERATION

CORDIC MICROROTATION

$$
\begin{aligned}
x[j+1] &= x[j] - \sigma_j 2^{-j} y[j] \\
y[j+1] &= y[j] + \sigma_j 2^{-j} x[j] \\
z[j+1] &= z[j] - \sigma_j \tan^{-1}(2^{-j})
\end{aligned}
$$

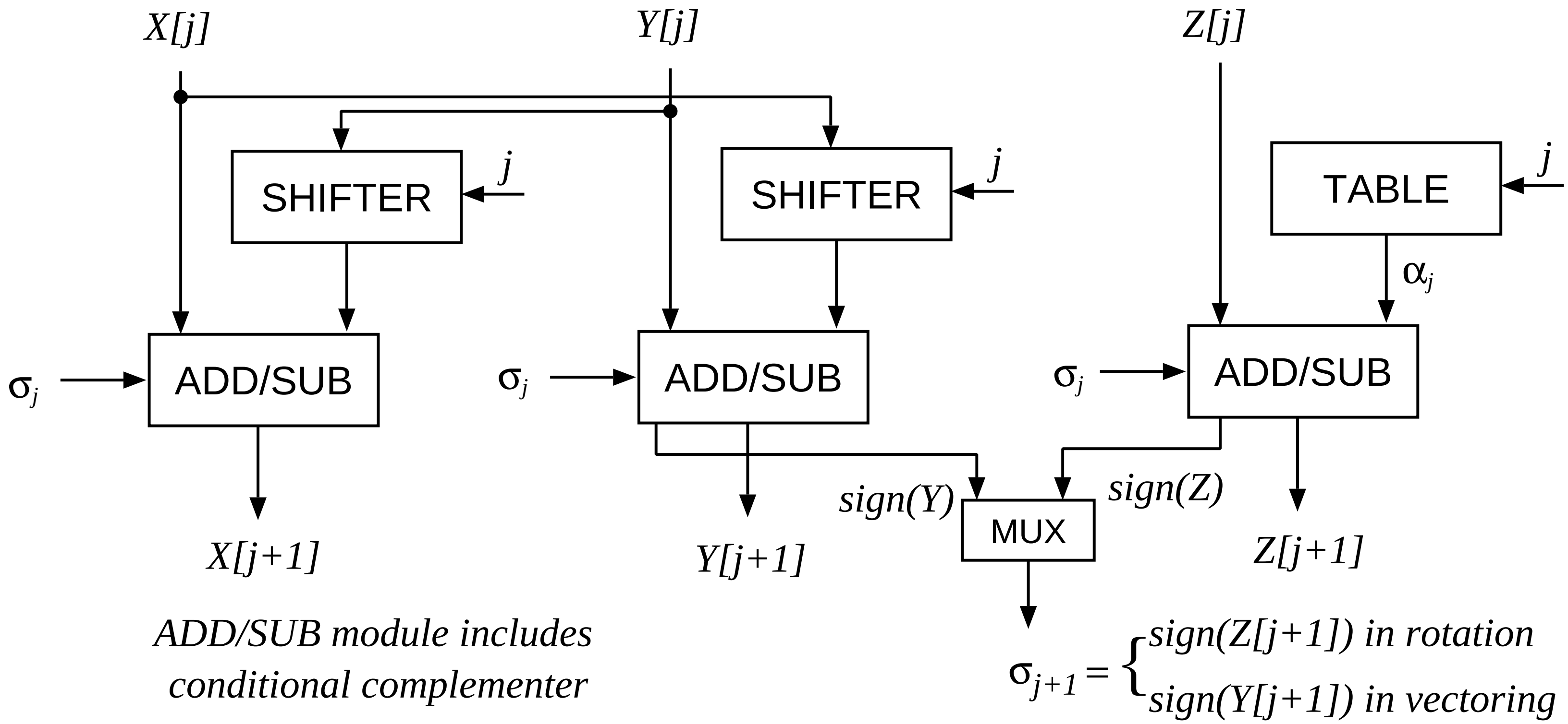

Figure 11.2: IMPLEMENTATION OF ONE ITERATION.

ROTATION MODE

- ROTATE AN INITIAL VECTOR (x_{in}, y_{in}) BY θ
- DECOMPOSE THE ANGLE

$$z[j+1] = z[j] - \sigma_j \tan^{-1}(2^{-j})$$

$$z[0] = \theta \quad x[0] = x_{in} \quad y[0] = y_{in}$$

$$\sigma_j = \begin{cases} 1 & if\ z[j] \geq 0 \\ -1 & if\ z[j] < 0 \end{cases}$$

- PERFORM MICRO-ROTATIONS
- FINAL VALUES

$$\begin{aligned} x_f &= K(x_{in} \cos\theta - y_{in} \sin\theta) \\ y_f &= K(x_{in} \sin\theta + y_{in} \cos\theta) \\ z_f &= 0 \end{aligned}$$

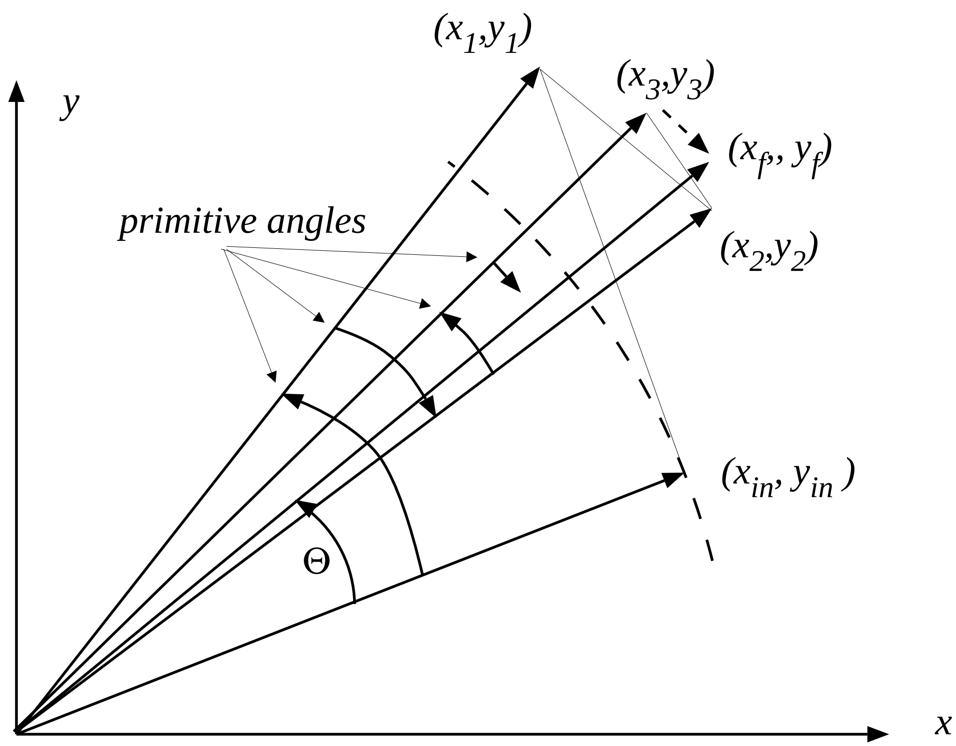

Figure 11.3: Rotating a vector using microrotations.

EXAMPLE OF ROTATION

ROTATE (x_{in}, y_{in}) BY 67° USING $n = 12$ MICRO-ROTATIONS

INITIAL COORDINATES: $x_{in} = 1$, $y_{in} = 0.125$

FINAL COORDINATES: $x_R = 0.2756$, $y_R = 0.9693$

j	$z[j]$	σ_j	$x[j]$	$y[j]$
0	1.1693	1	1.0	0.125
1	0.3839	1	0.875	1.125
2	-0.0796	-1	0.3125	1.1562
3	0.1653	1	0.7031	1.4843
4	0.0409	1	0.5175	1.5722
5	-0.0214	-1	0.4193	1.6046
6	0.0097	1	0.4694	1.5915
7	-0.0058	-1	0.4445	1.5988
8	0.0019	1	0.4570	1.5953
9	-0.0019	-1	0.4508	1.5971
10	0.0000	1	0.4539	1.5962
11	-0.0009	-1	0.4524	1.5967
12	-0.0004	-1	0.4531	1.5965
13			0.4535	1.5963

EXAMPLE 11.1 (cont.)

- AFTER COMPENSATION OF SCALING FACTOR $K = 1.64676$ COORDINATES ARE $x[13]/K = 0.2753$ and $y[13]/K = 0.9693$

- ERRORS $< 2^{-12}$

SPECIAL CASES

- TO COMPUTE $\cos\theta$ AND $\sin\theta$

  MAKE INITIAL CONDITION $x[0] = 1/K$ AND $y[0] = 0$

- IN GENERAL, FOR a AND b CONSTANTS

  $a\cos\theta - b\sin\theta$

  $a\sin\theta + b\cos\theta$

  COMPUTED BY SETTING $x[0] = a/K$ AND $y[0] = b/K$

VECTORING MODE

- ROTATE INITIAL VECTOR (x_{in}, y_{in}) UNTIL $y = 0$
- FOR INITIAL VECTOR IN THE FIRST QUADRANT:

$$\sigma_j = \begin{cases} 1 & \text{if } y[j] < 0 \\ -1 & \text{if } y[j] \geq 0 \end{cases}$$

- ACCUMULATE ROTATION ANGLE IN z
- FOR $x[0] = x_{in}$, $y[0] = y_{in}$ and $z[0] = z_{in}$, THE FINAL VALUES ARE

$$\begin{aligned} x_f &= K(x_{in}^2 + y_{in}^2)^{1/2} \\ y_f &= 0 \\ z_f &= z_{in} + \tan^{-1}(\frac{y_{in}}{x_{in}}) \end{aligned}$$

EXAMPLE OF VECTORING

- INITIAL VECTOR $(x_{in} = 0.75,\ y_{in} = 0.43)$
- y FORCED TO ZERO IN $n = 12$ MICRO-ROTATIONS
- ROTATED VECTOR: $x_R = \sqrt{x_{in}^2 + y_{in}^2} = 0.8645$, $y_R = 0.0$
- ROTATED ANGLE $z_f = \tan^{-1}(\frac{0.43}{0.75}) = 0.5205$

j	$y[j]$	σ_j	$x[j]$	$z[j]$
0	0.43	-1	0.75	0.0
1	-0.32	1	1.18	0.7853
2	0.27	-1	1.34	0.3217
3	-0.065	1	1.4075	0.5667
4	0.1109	-1	1.4156	0.4423
5	0.0224	-1	1.4225	0.5047
6	-0.0219	1	1.4232	0.5360
7	0.0002	-1	1.4236	0.5204
8	-0.0108	1	1.4236	0.5282
9	-0.0053	1	1.4236	0.5243
10	-0.0025	1	1.4236	0.5223
11	-0.0011	1	1.4236	0.5213
12	-0.0004	1	1.4236	0.5208
13			1.4236	0.5206

EXAMPLE 11.2 (cont.)

- ACCUMULATED ANGLE $z[13] = 0.5206$
- AFTER PERFORMING COMPENSATION OF $K = 1.64676$, $x[13]/K = 0.864$
- ERRORS $< 2^{-12}$

CONVERGENCE, PRECISION, AND RANGE

- ROTATION MODE
- CONVERGENCE

$$|z[i]| \leq \sum_{j=i}^{\infty} \tan^{-1}(2^{-j})$$

$$\theta_{max} = z[0]_{max} = \sum_{j=0}^{\infty} \tan^{-1}(2^{-j}) \approx 1.7429 \ (99.88^o)$$

FOR THIS ANGLE ALL $\sigma_j = 1$ and $z[j] > 0$.

- CONSIDER $\theta < \theta_{max}$

$$|z[i]| \leq \tan^{-1}(2^{-(i-1)})$$

- CONSEQUENTLY

$$\tan^{-1}(2^{-i-1}) \leq \sum_{j=i}^{\infty} \tan^{-1}(2^{-j})$$

OR

$$\tan^{-1}(2^{-i}) \leq \sum_{j=i+1}^{\infty} \tan^{-1}(2^{-j})$$

SATISFIED FOR ALL i

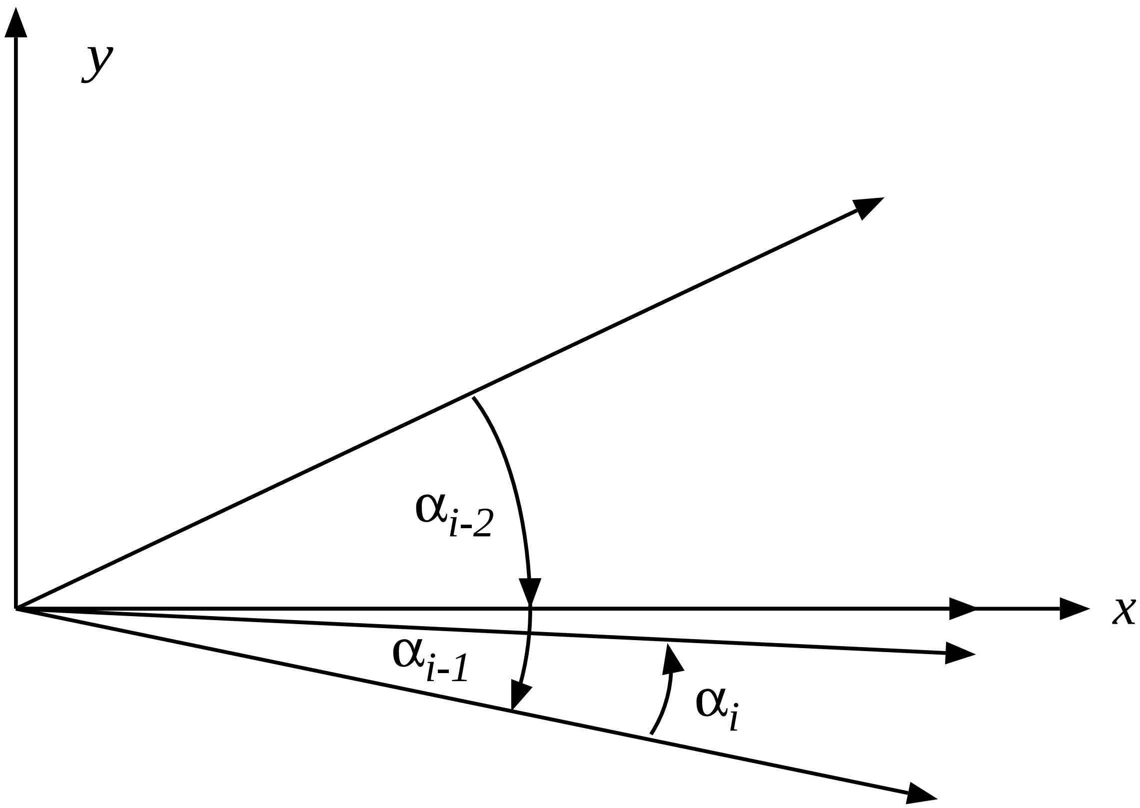

Figure 11.4: CONVERGENCE CONDITION: THE MAXIMUM NEGATIVE CASE.

PRECISION AND RANGE FOR n ITERATIONS

- n ITERATIONS (FINITE SEQUENCE)
- RESIDUAL ANGLE AFTER n ITERATIONS $z[n]$

$$|z[n]| \leq \tan^{-1}(2^{-(n-1)})$$

$$2^{-n} < \tan^{-1}(2^{-(n-1)}) < 2^{-(n-1)}$$

- THE MAXIMUM ANGLE FOR CONVERGENCE

$$\theta_{max} = \sum_{i=0}^{n-1} \tan^{-1}(2^{-j}) + 2^{-n+1}$$

- 2^{-n+1} THE MAXIMUM RESIDUAL ANGLE

COMPENSATION OF SCALING FACTOR

- MOST DIRECT METHOD: MULTIPLY BY $1/K$
- USE SCALING ITERATIONS OF THE FORM (1 ± 2^{-i})

$$x_s = x \pm x(2^{-i})$$

- USE REPETITIONS OF CORDIC ITERATIONS

$$|z[i+1]| \leq \tan^{-1}(2^{-i})$$

- OPTIMIZATION: FIND THE MINIMUM NUMBER OF SCALING ITERATIONS PLUS REPETITIONS SO THAT THE SCALE FACTOR IS COMPENSATED.

Table 11.4: Scale factor compensation for $n = 24$

Scaling iterations	(-1)(+2)(-5)(+10)(+16)(+19)(+22)
Scalings + repetitions	(-2)(+16)(+17) 1,3,5,6

IMPLEMENTATIONS

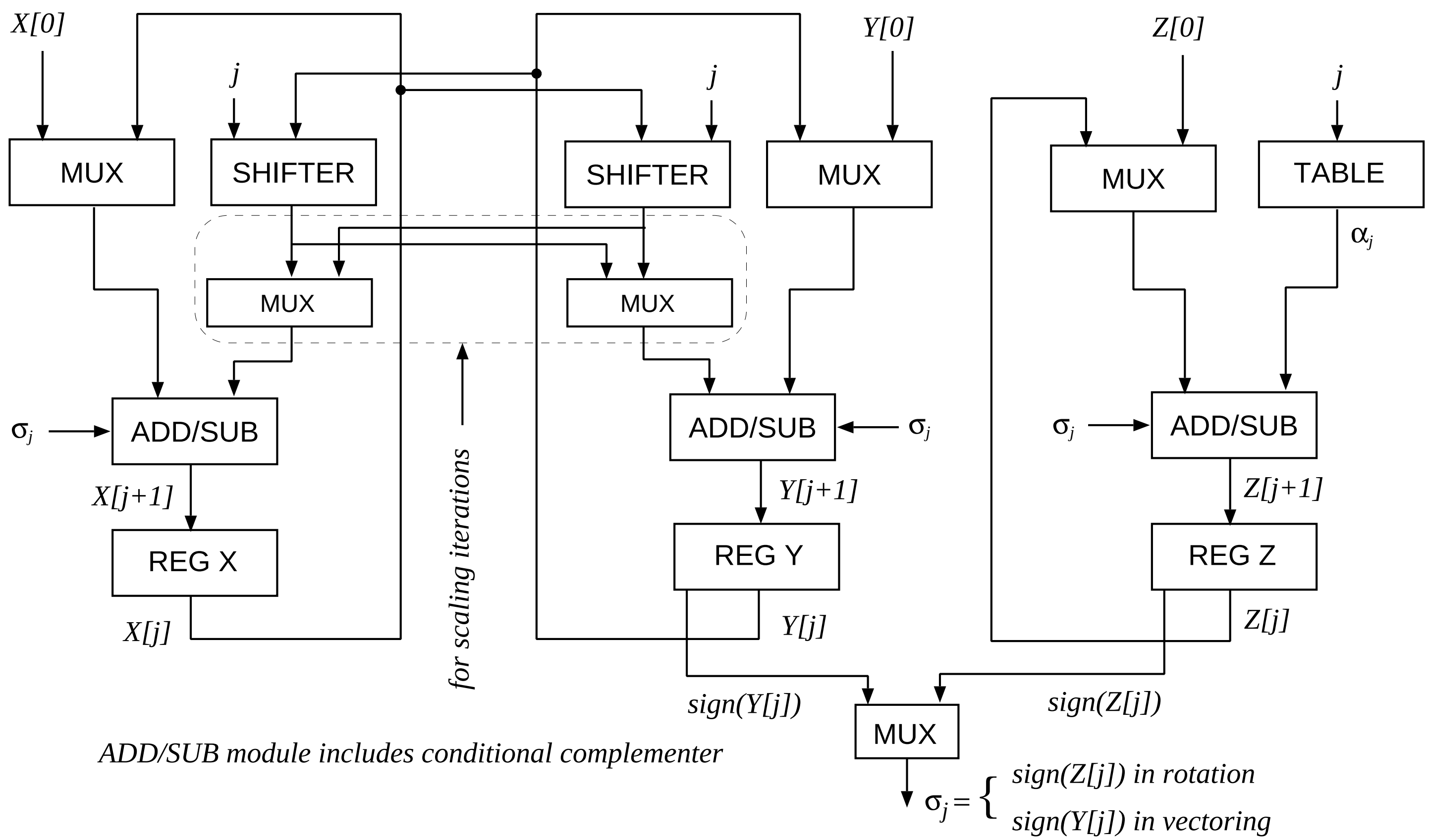

Figure 11.5: WORD-SERIAL IMPLEMENTATION.

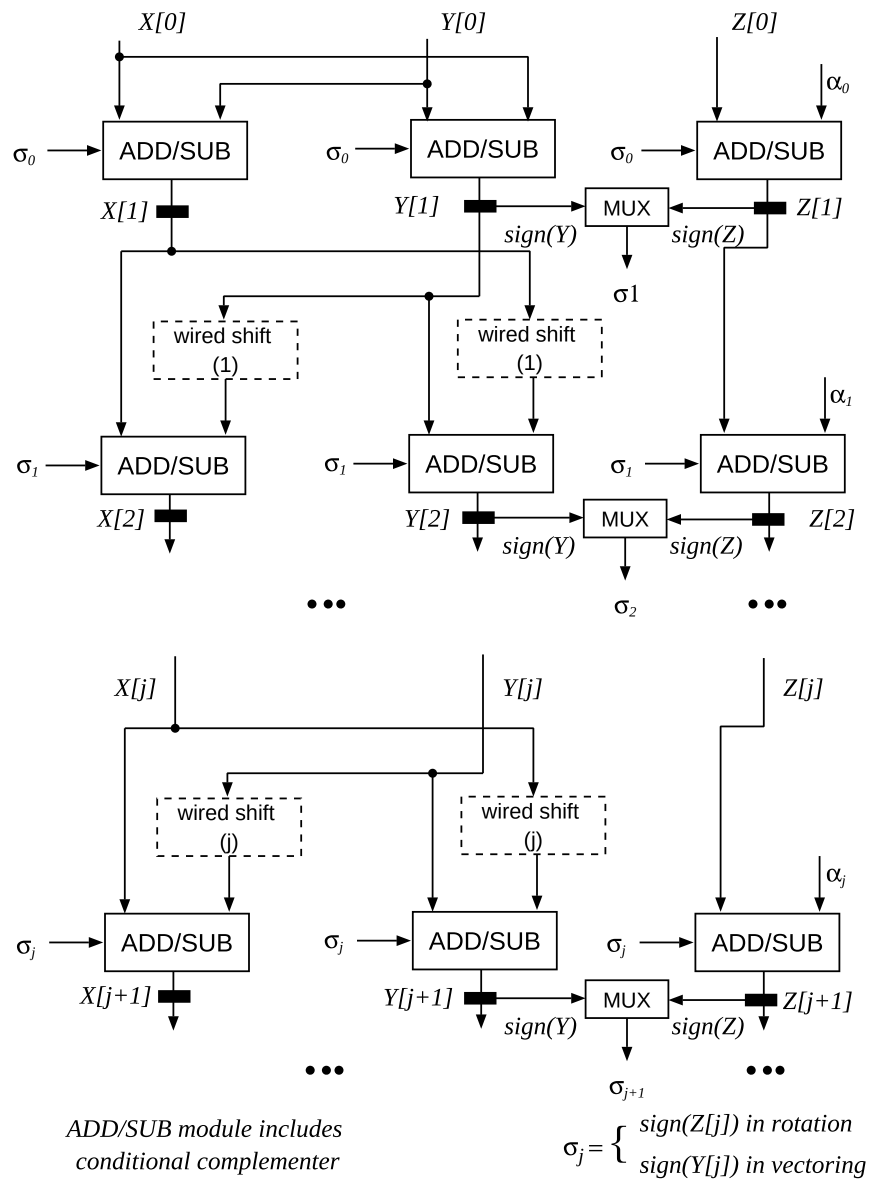

Figure 11.6: PIPELINED IMPLEMENTATION.

EXTENSION TO HYPERBOLIC AND LINEAR COORDINATES

- HYPERBOLIC COORDINATES

$$\begin{bmatrix} x_R \\ y_R \end{bmatrix} = \begin{bmatrix} \cosh\theta & \sinh\theta \\ \sinh\theta & \cosh\theta \end{bmatrix} \begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix}$$

- CORDIC HYPERBOLIC MICROROTATION:

$$\begin{aligned} x[j+1] &= x[j] + \sigma_j 2^{-j} y[j] \\ y[j+1] &= y[j] + \sigma_j 2^{-j} x[j] \\ z[j+1] &= z[j] - \sigma_j \tanh^{-1}(2^{-j}) \end{aligned}$$

- SCALING FACTOR IN ITERATION j

$$K_h[j] = (1 - 2^{-2j})^{1/2}$$

- $\tanh^{-1} 2^0 = \infty$ (and $K_h[0] = 0$) $\Longrightarrow$NECESSARY TO BEGIN FROM ITERATION $j = 1$

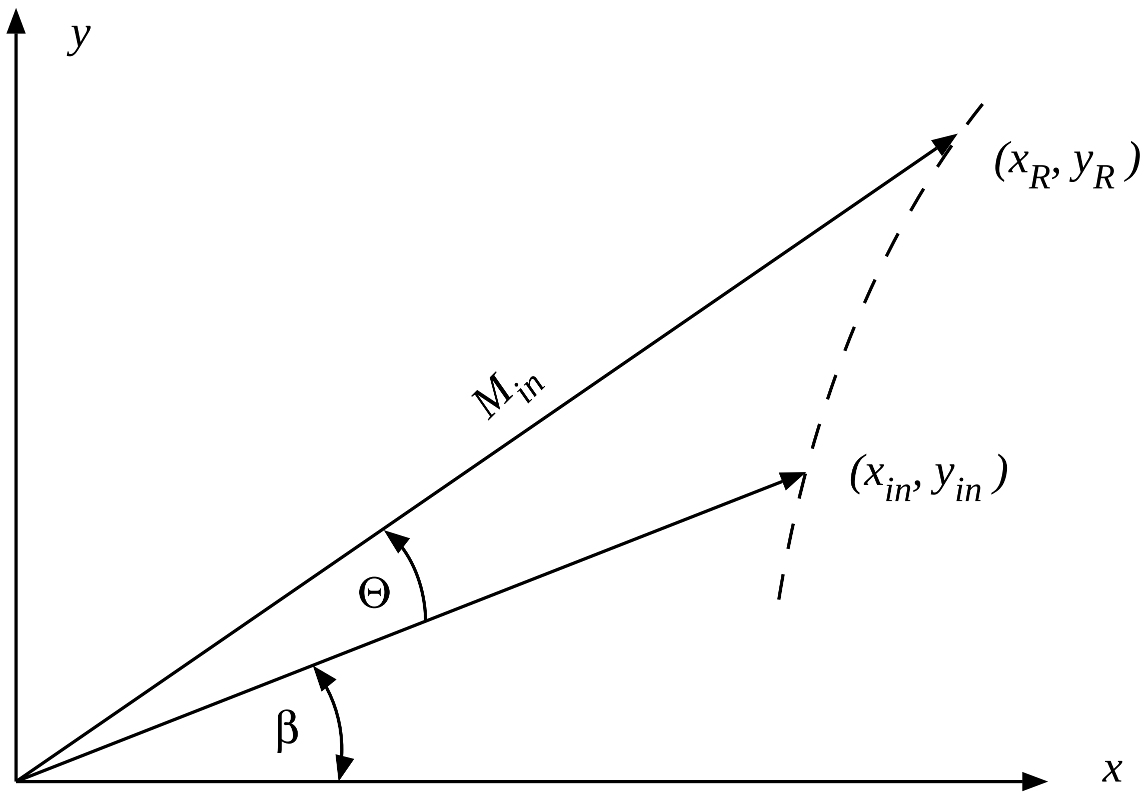

Figure 11.7: ROTATION IN HYPERBOLIC COORDINATE SYSTEM.

CONVERGENCE PROBLEM

- DOES NOT CONVERGE WITH SEQUENCE OF ANGLES $\tanh^{-1}(2^{-j})$ SINCE

$$\sum_{j=i+1}^{\infty} \tanh^{-1}(2^{-j}) < \tanh^{-1}(2^{-i})$$

- A SOLUTION: REPEAT SOME ITERATIONS

$$\sum_{i=j+1}^{\infty} \tanh^{-1}(2^{-i}) < \tanh^{-1}(2^{-j}) < \sum_{i=j+1}^{\infty} \tanh^{-1}(2^{-i}) + \tanh^{-1}(2^{-(3j+1)})$$

$\Longrightarrow$REPEATING ITERATIONS $4, 13, 40, ..., k, 3k + 1, ...$ RESULTS IN A CONVERGENT ALGORITHM.

- WITH THESE REPETITIONS

$$K_h \approx 0.82816$$

$$\theta_{max} = 1.11817$$

HYPERBOLIC ROTATION AND VECTORING

FINAL VALUES:

- FOR ROTATION MODE

$$\begin{aligned} x_f &= K_h(x_{in}\cosh\theta + y_{in}\sinh\theta) \\ y_f &= K_h(x_{in}\sinh\theta + y_{in}\cosh\theta) \\ z_f &= 0 \end{aligned}$$

- FOR VECTORING MODE

$$\begin{aligned} x_f &= K_h(x_{in}^2 - y_{in}^2)^{1/2} \\ y_f &= 0 \\ z_f &= z_{in} + \tanh^{-1}(\frac{y_{in}}{x_{in}}) \end{aligned}$$

LINEAR COORDINATES

$$x_R = x_{in}$$
$$y_R = y_{in} + x_{in} z_{in}$$

$$x[j+1] = x[j]$$
$$y[j+1] = y[j] + \sigma_j 2^{-j} x[j]$$
$$z[j+1] = z[j] - \sigma_j (2^{-j})$$

THE SCALING FACTOR IS 1.
FOR THE VECTORING MODE THE FINAL VALUES

$$x_f = x_{in}$$
$$z_f = z_{in} + \frac{y_{in}}{x_{in}}$$

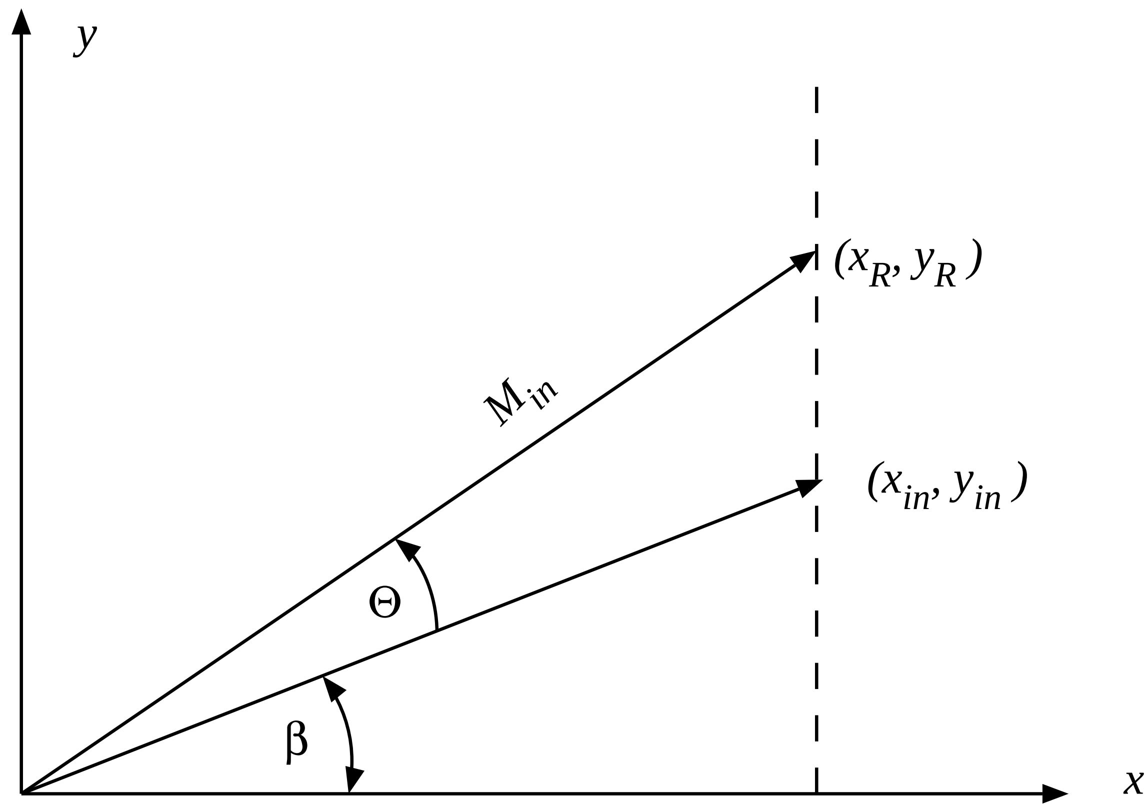

Figure 11.8: ROTATION IN LINEAR COORDINATE SYSTEM.

UNIFIED DESCRIPTION

- $m = 1$ FOR CIRCULAR COORDINATES
- $m = -1$ FOR HYPERBOLIC COORDINATES
- $m = 0$ FOR LINEAR COORDINATES
- UNIFIED MICROROTATION IS

$$
\begin{aligned}
x[j+1] &= x[j] - m\sigma_j 2^{-j} y[j] \\
y[j+1] &= y[j] + \sigma_j 2^{-j} x[j] \\
z[j+1] &= \begin{cases} z[j] - \sigma_j \tan^{-1}(2^{-j}) & \text{if } m = 1 \\ z[j] - \sigma_j \tanh^{-1}(2^{-j}) & \text{if } m = -1 \\ z[j] - \sigma_j (2^{-j}) & \text{if } m = 0 \end{cases}
\end{aligned}
$$

ALSO $z[j+1] = z[j] - \sigma_j m^{-1/2} \tan^{-1}(m^{1/2} 2^{-j})$

- THE SCALING FACTOR IS

$$K_m[j] = (1 + m 2^{-2j})^{1/2}$$

Table 11.5: UNIFIED CORDIC

Coordinates	Rotation mode $\sigma_j = sign(z[j])^+$	Vectoring mode $\sigma_j = -sign(y[j])^+$
Circular $(m = 1)$ $\alpha_j = \tan^{-1}(2^{-j})$ initial $j = 0$ $j = 0, 1, 2, .., n$ $K_1 \approx 1.64676$ $\theta_{max} \approx 1.74329$	$x_f = K_1(x_{in}\cos(z_{in}) - y_{in}\sin(z_{in}))$ $y_f = K_1(x_{in}\sin(z_{in}) + y_{in}\cos(z_{in}))$ $z_f = 0$	$x_f = K_1(x_{in}^2 + y_{in}^2)^{1/2}$ $y_f = 0$ $z_f = z_{in} + \tan^{-1}(\frac{y_{in}}{x_{in}})$
Linear $(m = 0)$ $\alpha_j = 2^{-j}$ initial $j = 0$ $j = 0, 1, , 2, ..., n$ $K_0 = 1$ $\theta_{max} = 2 - 2^{-n}$	$x_f = x_{in}$ $y_f = y_{in} + x_{in}z_{in}$ $z_f = 0$	$x_f = x_{in}$ $y_f = 0$ $z_f = z_{in} + \frac{y_{in}}{x_{in}}$
Hyperbolic $(m = -1)$ $\alpha_j = \tanh^{-1}(2^{-j})$ initial $j = 1$ $j = 1, 2, 3, 4, 4, 5...13, 13, ...$ $K_{-1} \approx 0.82816$ $\theta_{max} \approx 1.11817$	$x_f = K_{-1}(x_{in}\cosh(z_{in}) + y_{in}\sinh(z_{in}))$ $y_f = K_{-1}(x_{in}\sinh(z_{in}) + y_{in}\cosh(z_{in}))$ $z_f = 0$	$x_f = K_{-1}(x_{in}^2 - y_{in}^2)^{1/2}$ $y_f = 0$ $z_f = z_{in} + \tanh^{-1}(\frac{y_{in}}{x_{in}})$

$^+ \ sign(a) = 1 \ if \ a \geq 0, \quad sign(a) = -1 \ if \ a < 0.$

OTHER FUNCTIONS

Table 11.6: SOME ADDITIONAL FUNCTIONS

m	Mode	Initial values			Functions	
		x_{in}	y_{in}	z_{in}	x_R	y_R or z_R
1	rotation	1	0	θ	$\cos\theta$	$y_R = \sin\theta$
-1	rotation	1	0	θ	$\cosh\theta$	$y_R = \sinh\theta$
-1	rotation	a	a	θ	ae^{θ}	$y_R = ae^{\theta}$
1	vectoring	1	a	$\pi/2$	$\sqrt{a^2+1}$	$z_R = \cot^{-1}(a)$
-1	vectoring	a	1	0	$\sqrt{a^2-1}$	$z_R = \coth^{-1}(a)$
-1	vectoring	$a+1$	$a-1$	0	$2\sqrt{a}$	$z_R = 0.5\ln(a)$
-1	vectoring	$a+\frac{1}{4}$	$a-\frac{1}{4}$	0	$\sqrt{a}$	$z_R = \ln(\frac{1}{4}a)$
-1	vectoring	$a+b$	$a-b$	0	$2\sqrt{ab}$	$z_R = 0.5\ln(\frac{a}{b})$

Note: the final values x_R and y_R are obtained after compensation of the scale factor.

REDUNDANT REPRESENTATION

- CRITICAL PATH of CORDIC ITERATION: ADDER (CPA)
- TO REDUCE IT: USE OF REDUNDANT ADDER
- PROBLEM WITH SIGN DETECTION:
  - If $\sigma \in \{-1, 1\}$, must convert to conventional - NO GOOD
  - If $\sigma \in \{-1, 0, 1\}$, can use estimate in selection

    $\Rightarrow$ SCALING FACTOR NO LONGER CONSTANT
- TWO APPROACHES FOR $\sigma \in \{-1, 0, 1\}$
  1. CALCULATE VARIABLE SCALING FACTOR AND PERFORM COMPENSATION
  2. DOUBLE-ROTATION APPROACH
- TWO APPROACHES FOR $\sigma \in \{-1, 1\}$
  1. USE ADDITIONAL ITERATIONS (Correcting iterations)
  2. USE 2 CORDIC MODULES (Plus/Minus)

DOUBLE ROTATION APPROACH

- σ_j is $\{-1, 0, 1\}$
- To maintain the constant scale factor, perform a double rotation
  - $\sigma_j = 1$. Both rotations are by angle $\tan^{-1}(2^{-(j+1)})$
  - $\sigma = 0$. The two rotations are by the angles $\tan^{-1}(2^{-(j+1)})$ and $-\tan^{-1}(2^{-(j+1)})$.
  - $\sigma_j = -1$. Both rotations are by the angle $-\tan^{-1}(2^{-(j+1)})$.
- Consequently, the scaling factor is constant and has value

$$K = \prod_{j=1}^{n}(1 + 2^{-2j})$$

- The elementary are $\alpha_j = 2\tan^{-1}(2^{-(j+1)})$

RECURRENCES FOR DOUBLE ROTATION

$$
\begin{aligned}
x[j+1] &= x[j] - q_j 2^{-j} y[j] - p_j 2^{-2j-2} x[j] \\
y[j+1] &= y[j] + q_j 2^{-j} x[j] - p_j 2^{-2j-2} y[j] \\
z[j+1] &= z[j] - q_j (2 \tan^{-1}(2^{-(j+1)}))
\end{aligned}
$$

- Two control variables $(q_j,\ p_j)$: (1,1) for $\sigma_j = 1$; (0,-1) for $\sigma_j = 0$; and (-1,1) for $\sigma_j = -1$
- The value of σ_j determined from an estimate of variable ($z[j]$ for rotation and $y[j]$ for vectoring)
- since the variable converges to 0, the estimate of the sign uses the bits $j-1$, j, and $j+1$.
- Advantage: uses a redundant representation and produces a constant scaling factor
- Disadvantage: the recurrence requires three terms instead of two