CORDIC Background (2A)

-
-

Young Won Lim
05/28/2011

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

CORDIC Background

1.A survey of CORDIC algorithms for FPGAs, Ray Andraka,
www.andraka.com/cordic.htm

Vector Rotation (1)

$$x' = x \cos\phi - y \sin\phi$$

$$y' = y \cos\phi + x \sin\phi$$

$$x' = \boxed{\cos\phi} \cdot [x - y \tan\phi]$$

$$y' = \boxed{\cos\phi} \cdot [y + x \tan\phi]$$

$$x_{i+1} = \boxed{K_i} \cdot [x_i - y_i \cdot d_i \cdot 2^{-i}]$$

$$y_{i+1} = \boxed{K_i} \cdot [y_i + x_i \cdot d_i \cdot 2^{-i}]$$

$$K_i = \cos\phi_i = \cos(\tan^{-1}(2^{-i}))$$

$$= \frac{1}{\sqrt{1 + 2^{-2i}}}$$

$$d_i = \pm 1$$

Restrict rotation angle ➡ $\tan\phi = \pm 2^{-i}$

Multiplication ➡ simple shift

$$y \cdot \tan\phi \qquad\qquad y \cdot 2^{-i}$$
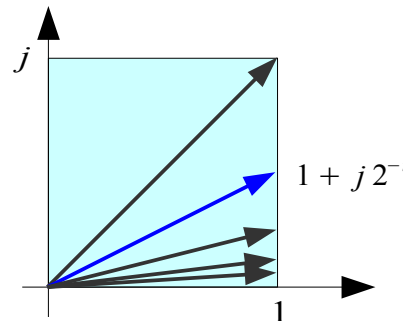$$x \cdot \tan\phi \qquad\qquad x \cdot 2^{-i}$$

regardless of direction ➡ $\cos(\phi) = \cos(-\phi)$

Allowed Rotation Angles

$1 + j\, 2^{-i}$

$\tan\phi$ ➡	2^{-i}
$\cos\phi$ ➡	$\dfrac{1}{\sqrt{1 + 2^{-2i}}}$

$$K_i \leq 1 \qquad\qquad (\; K_i \Longleftarrow \cos\phi \;)$$

Vector Rotation (2)

$$x_{i+1} = \boxed{K_i} \cdot \left[x_i - y_i \cdot d_i \cdot 2^{-i} \right]$$

$$y_{i+1} = \boxed{K_i} \cdot \left[y_i + x_i \cdot d_i \cdot 2^{-i} \right]$$

$$K_i = \cos \phi_i = \cos\left(\tan^{-1}\left(2^{-i} \right) \right)$$

$$= \frac{1}{\sqrt{1 + 2^{-2i}}} \qquad K_i \leq 1$$

$$d_i = \pm 1$$

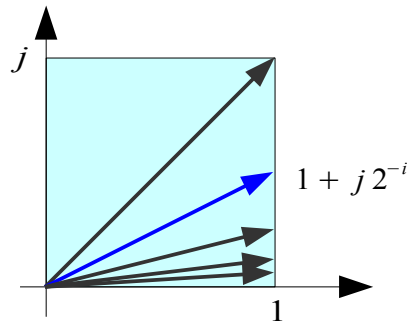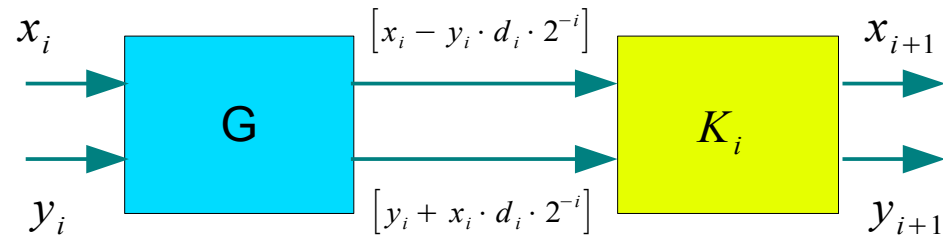$$x_{i+1}{}^2 = K_i{}^2 \cdot \left[x_i{}^2 + y_i{}^2 \cdot 2^{-2i} - 2 x_i y_i d_i \cdot 2^{-i} \right]$$

$$y_{i+1}{}^2 = K_i{}^2 \cdot \left[y_i{}^2 + x_i{}^2 \cdot 2^{-2i} + 2 x_i y_i d_i \cdot 2^{-i} \right]$$

$$x_{i+1}{}^2 + y_{i+1}{}^2 = K_i{}^2 \cdot \left(1 + 2^{-2i} \right) \cdot \left(x_i{}^2 + y_i{}^2 \right)$$

$$G \cdot K_i = 1 \qquad\qquad K_i \leq 1 \qquad G > 1$$

x_i → G $\left[x_i - y_i \cdot d_i \cdot 2^{-i} \right]$ → K_i → x_{i+1}

y_i → $\left[y_i + x_i \cdot d_i \cdot 2^{-i} \right]$ → y_{i+1}

$1 + j\,2^{-i}$

$\tan \phi$ ➡	2^{-i}
$\cos \phi$ ➡	$\dfrac{1}{\sqrt{1 + 2^{-2i}}}$

CORDIC Gain : *growing in magnitude*

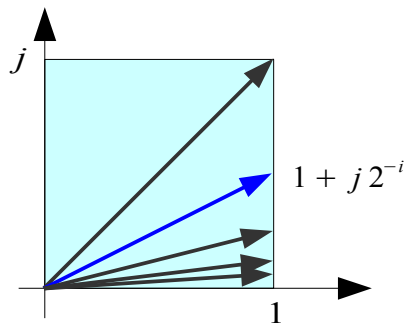$$A_n = \prod_{i=1}^{n} \frac{1}{K_i} = \prod_{i=1}^{n} \sqrt{1 + 2^{-2i}} \rightarrow 1.647$$

Young Won Lim
05/28/2011

Vector Rotation (3)

$$x_{i+1} = \boxed{K_i} \cdot \left[x_i - y_i \cdot d_i \cdot 2^{-i} \right]$$

$$y_{i+1} = \boxed{K_i} \cdot \left[y_i + x_i \cdot d_i \cdot 2^{-i} \right]$$

$$K_i = 1 / \sqrt{1 + 2^{-2i}} \quad \Leftarrow \quad \cos(\phi_i)$$

$$d_i = \pm 1$$

Without Scale Constants K_i

$$x_{i+1} = \left[x_i - y_i \cdot d_i \cdot 2^{-i} \right]$$

$$y_{i+1} = \left[y_i + x_i \cdot d_i \cdot 2^{-i} \right]$$

$$d_i = \pm 1$$

CORDIC Gain : *growing in magnitude*

$$A_n = \prod_{i=1}^{n} \frac{1}{K_i} = \prod_{i=1}^{n} \sqrt{1 + 2^{-2i}} \rightarrow 1.647$$

$$1 / K_i = \sqrt{1 + 2^{-2i}}$$

For correction

Multiplying K_i**'s as a processing gain**

$$\prod_{i=1}^{n} K_i = \prod_{i=1}^{n} \frac{1}{\sqrt{1 + 2^{-2i}}} \rightarrow 0.6073$$

$1 + j\,2^{-i}$

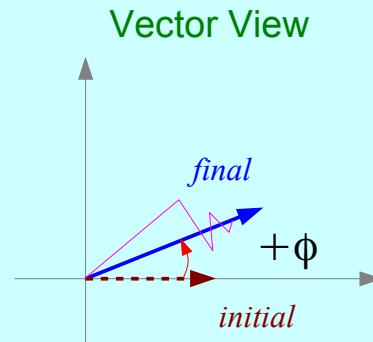| $\tan \phi$	$\Rightarrow$	2^{-i}
$\cos \phi$	$\Rightarrow$	$\dfrac{1}{\sqrt{1 + 2^{-2i}}}$

Angle Accumulator

Rotation Mode

$$z_0 \;\leftarrow\; \phi \quad (desired\ angle)$$

$$z_n \;\rightarrow\; 0$$
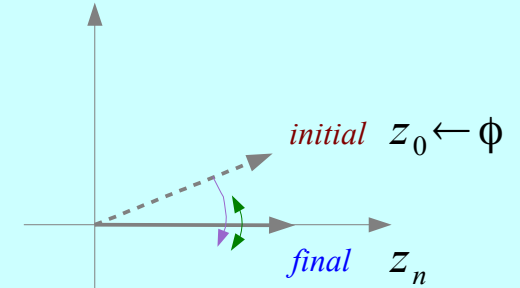
$$z_{i+1} \;=\; z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i \;=\; -1 \quad if \;\; z_i < 0$$

$$d_i \;=\; +1 \quad otherwise$$

Vector View

final

$+\phi$

initial

Minimize the residual angle

Accumulator View

initial $\;z_0 \leftarrow \phi$

final $\;z_n$

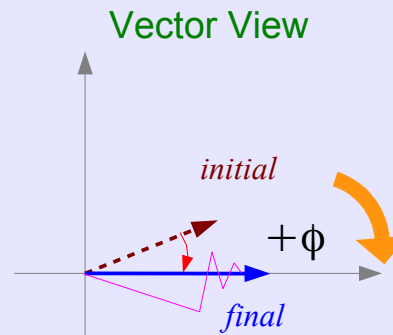**Subtract** *angles at each step*

Vectoring Mode

$$z_0 \;\leftarrow\; 0$$

$$z_n \;\rightarrow\; z_0 + \tan^{-1}(y_0/x_0)$$

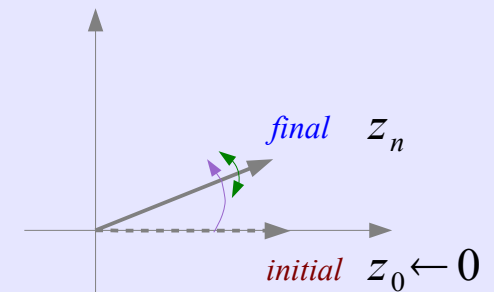$$z_{i+1} \;=\; z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i \;=\; +1 \quad if \;\; y_i < 0$$

$$d_i \;=\; -1 \quad otherwise$$

Vector View

initial

$+\phi$

final

Minimize the residual y component

Accumulator View

final $\;z_n$

initial $\;z_0 \leftarrow 0$
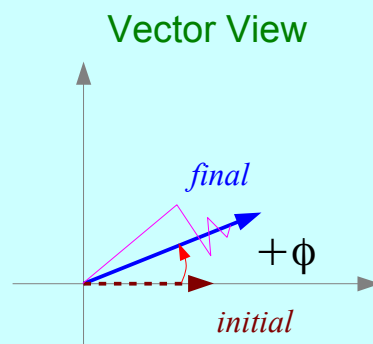
Add *angles at each step*

Rotation Mode

Rotation Mode

$$z_0 \leftarrow \phi \quad (desired\ angle)$$

$$z_n \rightarrow 0$$

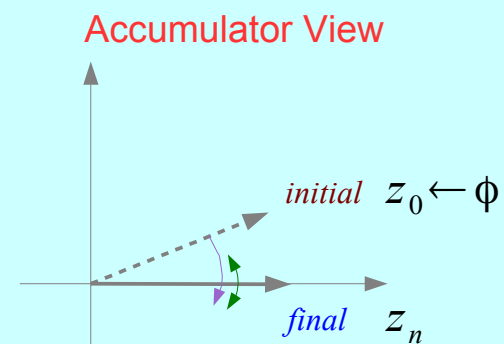$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = -1 \quad if \quad z_i < 0$$

$$d_i = +1 \quad otherwise$$

Vector View

Minimize the residual angle

Accumulator View

initial $z_0 \leftarrow \phi$

final z_n

Subtract *angles at each step*

$$x_{i+1} = x_i - y_i \cdot d_i \cdot 2^{-i}$$

$$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i}$$

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = -1 \quad if \quad z_i < 0$$

$$d_i = +1 \quad otherwise$$

$$x_n = A_n \left[x_0 \cos z_0 - y_0 \sin z_0 \right]$$

$$y_n = A_n \left[y_0 \cos z_0 + x_0 \sin z_0 \right]$$

$$z_n = 0$$

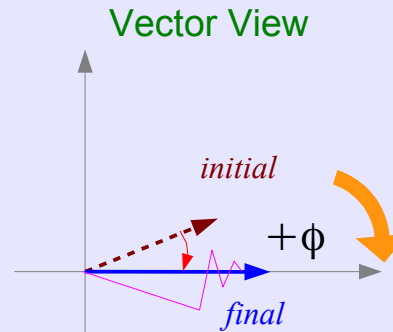$$A_n = \prod_{i=1}^{n} \sqrt{1 + 2^{-2i}}$$

Vectoring Mode

Vectoring Mode

$z_0 \leftarrow 0$
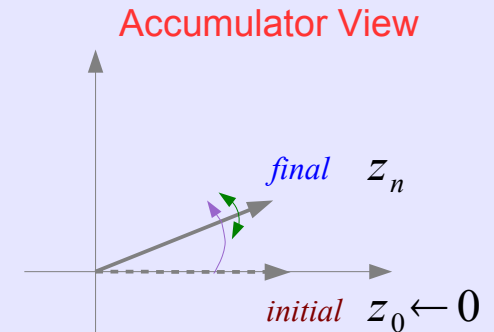
$z_n \rightarrow z_0 + \tan^{-1}(y_0/x_0)$

$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$

$d_i = +1 \quad if \quad y_i < 0$

$d_i = -1 \quad otherwise$

Vector View

initial

$+\phi$

final

Minimize the residual y component

Accumulator View

final z_n

initial $z_0 \leftarrow 0$

Add *angles at each step*

$x_{i+1} = x_i - y_i \cdot d_i \cdot 2^{-i}$

$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i}$

$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$

$d_i = +1 \quad if \quad y_i < 0$

$d_i = -1 \quad otherwise$

$x_n = A_n \sqrt{x_0^2 + y_0^2}$

$y_n = 0$

$z_n = z_0 + \tan^{-1}(y_0/x_0)$

$A_n = \prod_{i=1}^{n} \sqrt{1 + 2^{-2i}}$
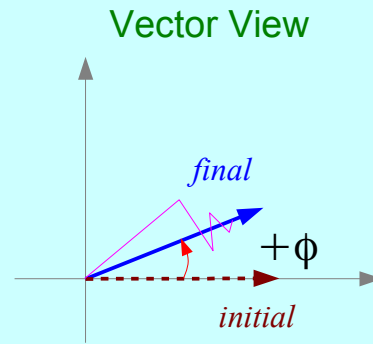
Angle Accumulator – Rotation Mode

Rotation Mode

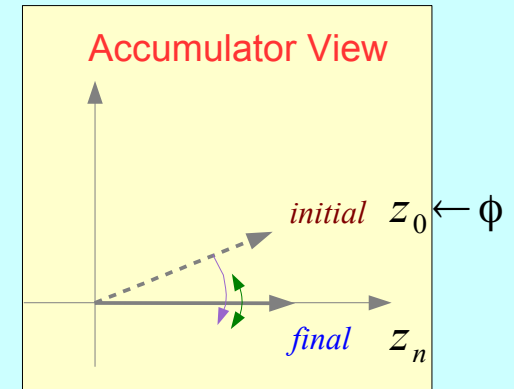$$z_0 \leftarrow \phi \quad (input)$$

$$z_n \rightarrow 0$$

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = -1 \quad if \quad z_i < 0$$

$$d_i = +1 \quad otherwise$$

Vector View

$+\phi$

final

initial

Minimize the residual angle

Accumulator View

initial $z_0 \leftarrow \phi$

final z_n

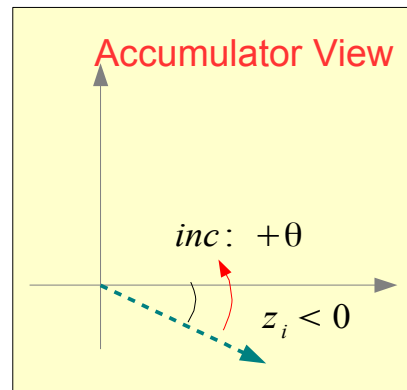***Subtract*** *angles at each step*

$$z_i < 0$$

Increase Angle $\quad d_i = -1$

$$z_{i+1} = z_i + \tan^{-1}(2^{-i})$$

$$z_i \geq 0$$

Decrease Angle $\quad d_i = +1$

$$z_{i+1} = z_i - \tan^{-1}(2^{-i})$$

Accumulator View

$inc: +\theta$

$z_i < 0$

$$z_i < 0$$

Increase Angle $+\theta$

Accumulator View

$z_i \geq 0$

$dec: -\theta$

$$z_i \geq 0$$

Decreases Angle $-\theta$
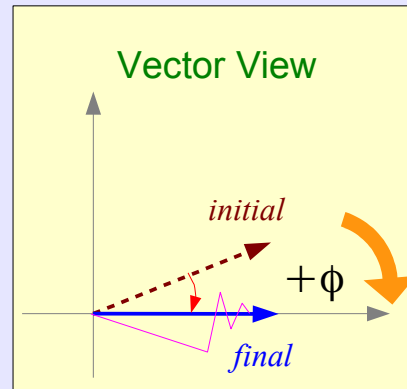
Angle Accumulator – Vectoring Mode

Vectoring Mode

$$z_0 \leftarrow 0$$

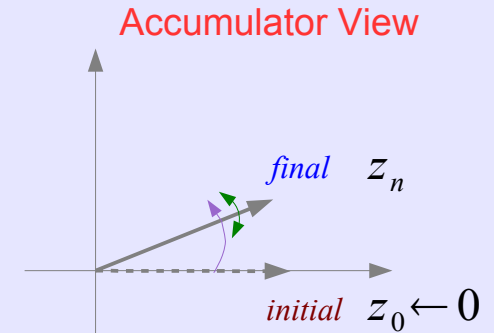$$z_n \rightarrow z_0 + \tan^{-1}(y_0/x_0)$$

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = +1 \quad if \quad y_i < 0$$

$$d_i = -1 \quad otherwise$$

Vector View

initial

$+\phi$

final

Minimize the residual y component

Accumulator View

final z_n

initial $z_0 \leftarrow 0$

Add *angles at each step*

$$y_i < 0$$
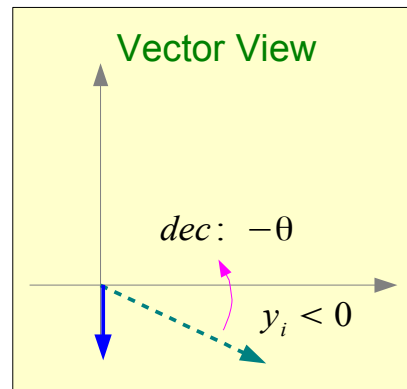
Decrease Angle $d_i = +1$

$$z_{i+1} = z_i - \tan^{-1}(2^{-i})$$

$$y_i > 0$$

Increase Angle $d_i = -1$

$$z_{i+1} = z_i + \tan^{-1}(2^{-i})$$

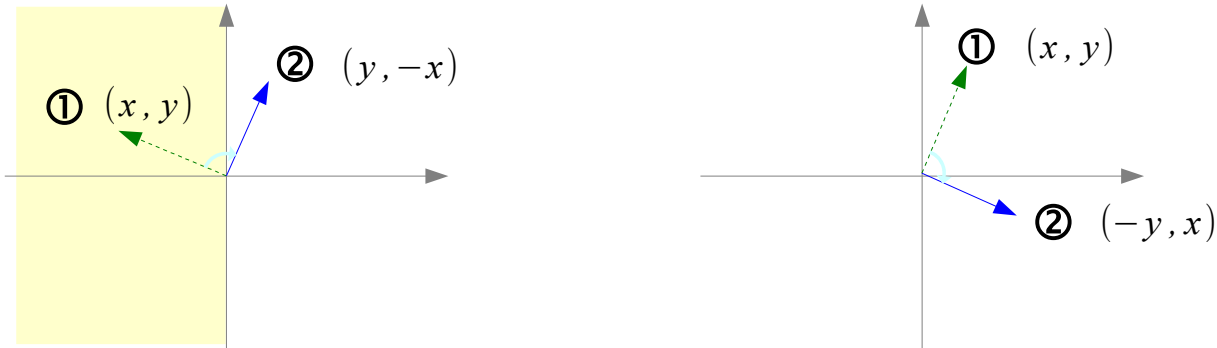### Vector View

$dec: -\theta$
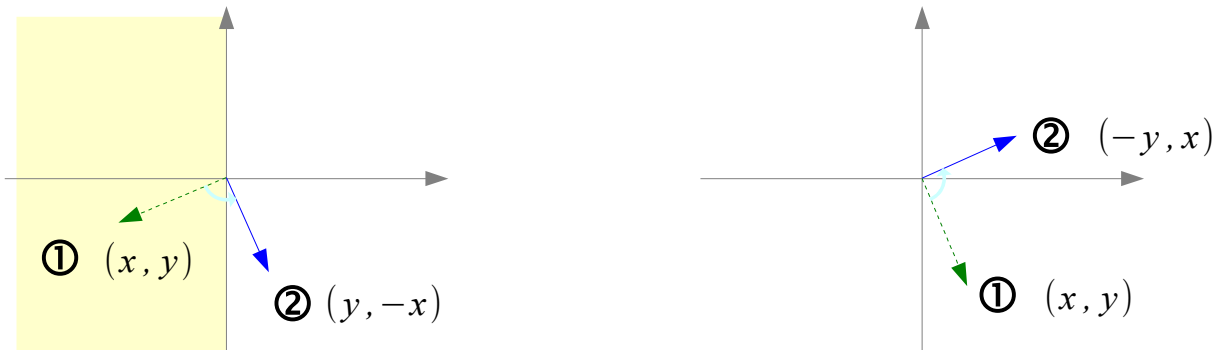
$y_i < 0$

$$y_i < 0$$

Decrease Angle $-\theta$

Vector View

$y_i > 0$

$inc: +\theta$

$$y_i > 0$$

Increases Angle $+\theta$

Initial Rotation $\pm\pi/2$

Positive Phase (y > 0) ➡ **Rotate by −90 degrees**

① (x, y) ② $(y, -x)$

① (x, y) ② $(-y, x)$

Negative Phase (y < 0) ➡ **Rotate by +90 degrees**

① (x, y) ② $(y, -x)$

② $(-y, x)$ ① (x, y)

Resulting Phase ➡ [-90, +90]

$$x' = -d \cdot y$$
$$y' = +d \cdot x$$
$$z' = z + d \cdot \frac{\pi}{2}$$

$$d = +1 \quad if \quad y < 0$$
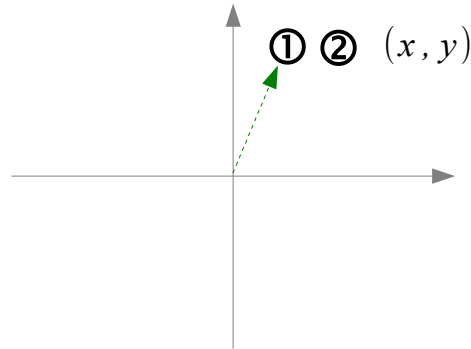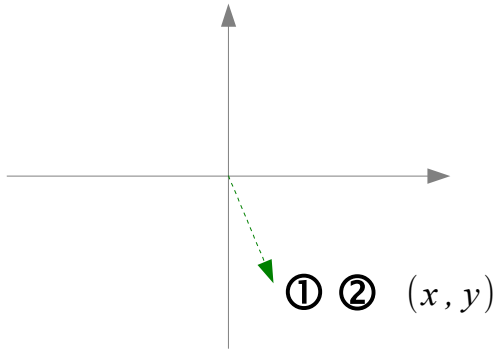$$d = -1 \quad otherwise$$

No magnitude change

$$x' \leftarrow y$$
$$y' \leftarrow x$$

Consistent

Initial Rotation $0, +\pi$

Positive x (x > 0) ➡ Rotate by 0 degrees

① ② (x, y)

① ② (x, y)

Negative x (x < 0) ➡ Rotate by +180 degrees

$z + \pi$?

② $(-x, -y)$

① (x, y)

$z - \pi$?

① (x, y)

② $(-x, -y)$

Resulting Phase ➡ [-90, +90]

$$x' = +d \cdot x$$
$$y' = +d \cdot y$$
$$z' = z \qquad if \quad d = 1$$
$$z' = \pi - z \quad if \quad d = -1$$

$$d = -1 \quad if \quad x < 0$$
$$d = +1 \quad otherwise$$

No magnitude change

$$x' \leftarrow y$$
$$y' \leftarrow x$$

Convenient wiring in FPGA

Application Modes (1)
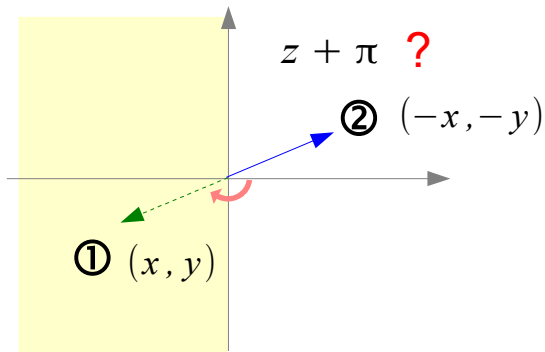
Rotation Mode

Input angle is given

- **sin** and **cos**
- $(r, \theta) \rightarrow (x, y)$
- General vector rotation

$\sin \phi \longleftarrow y$

final

initial

$\longleftarrow +\phi$

x

$\downarrow$

$\cos \phi$

Vectoring Mode

Finding the resulting angle

- **tan^{-1}**
- Vector Magnitude
- $(x, y) \rightarrow (r, \theta)$

initial

(x, y)

$\Longrightarrow +\phi$

final

$\downarrow$

r

Application Modes (2)

A. **sin & cos**

B. **$(r, \theta) \rightarrow (x, y)$**

C. General Vector Rotation

D. **$\tan^{-1}$**

E. Vector Magnitude

F. **$(x, y) \rightarrow (r, \theta)$**

G. **$\sin^{-1}$**

H. **$\cos^{-1}$**

I. Linear Functions

J. Hyperbolic Functions

A. Sine and Cosine

Rotation Mode

$$z_0 \leftarrow \phi \quad (desired\ angle)$$

$$z_n \rightarrow 0$$

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = -1 \quad if \quad z_i < 0$$

$$d_i = +1 \quad otherwise$$

Vector View

final (x_n, y_n)

$+\phi$

initial $(x_0, 0)$

Minimize the residual angle

Accumulator View

initial $z_0 \leftarrow \phi$

final z_n

Subtract angles at each step

Finding Sine and Cosine

$$(x_0,\ 0) \rightarrow (x_n,\ y_n)$$

$$x_n = A_n \cdot x_0 \cos z_0$$

$$y_n = A_n \cdot x_0 \sin z_0$$

CORDIC Gain : *growing in magnitude*

$$A_n = \prod_{i=1}^{n} \frac{1}{K_i} = \prod_{i=1}^{n} \sqrt{1 + 2^{-2i}} \rightarrow 1.647$$

Unscaled Sine and Cosine

$$x_0 \leftarrow \frac{1}{A_n} = 0.6073$$

$$x_n = \cos z_0$$

$$y_n = \sin z_0$$

Modulated Sine and Cosine

$$x_0 \leftarrow \left\{ \prod_{i=1}^{n} K_i \right\} \cdot x_0 = 0.6073 \cdot x_0$$

$$x_n = x_0 \cdot \cos z_0$$

$$y_n = x_0 \cdot \sin z_0$$

LUT → a pair of MULT

CORDIC → rotation operations
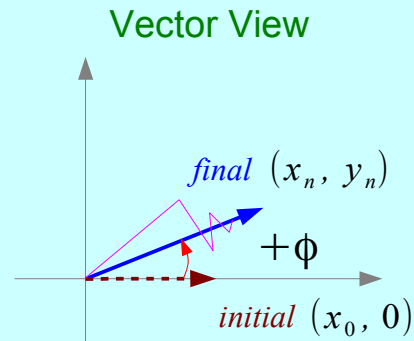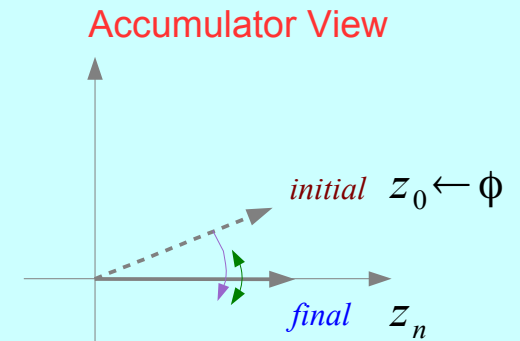
Single MULT

B. Polar to Rectangular

Rotation Mode

$$z_0 \leftarrow \phi \quad (desired\ angle)$$

$$z_n \rightarrow 0$$

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = -1 \quad if \quad z_i < 0$$

$$d_i = +1 \quad otherwise$$

Vector View

final (x_n, y_n)

θ

initial $(r, 0)$

*Minimize the
residual angle*

Accumulator View

initial $z_0 \leftarrow \theta$

final z_n

*Subtract angles
at each step*

Finding Sine and Cosine

$$(x_{0,}\ 0) \rightarrow (x_n,\ y_n)$$

$$x_n = A_n \cdot x_0 \cos z_0$$

$$y_n = A_n \cdot x_0 \sin z_0$$

$$x_0 \leftarrow r, \quad z_0 \leftarrow \theta$$

$$(r,\ 0) \rightarrow (x_n,\ y_n)$$

$$x_n = A_n r \cos \theta$$

$$y_n = A_n r \sin \theta$$

$$x_0 \leftarrow r \cdot \frac{1}{A_n}, \quad z_0 \leftarrow \theta$$

$$\left(\frac{r}{A_n},\ 0\right) \rightarrow (x_n,\ y_n)$$

$$x_n = r \cos \theta$$

$$y_n = r \sin \theta$$

CORDIC Gain : *growing in magnitude*

$$A_n = \prod_{i=1}^{n} \frac{1}{K_i} = \prod_{i=1}^{n} \sqrt{1 + 2^{-2i}} \rightarrow 1.647$$

C. General Vector Rotation

Rotation Mode
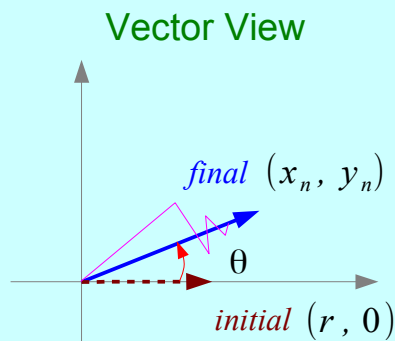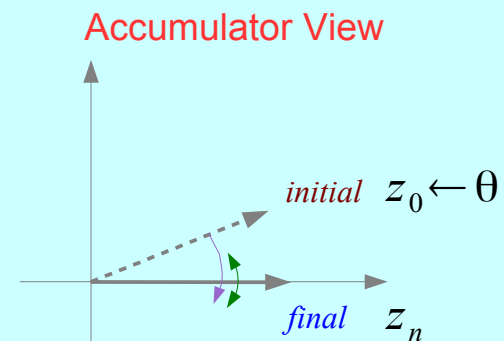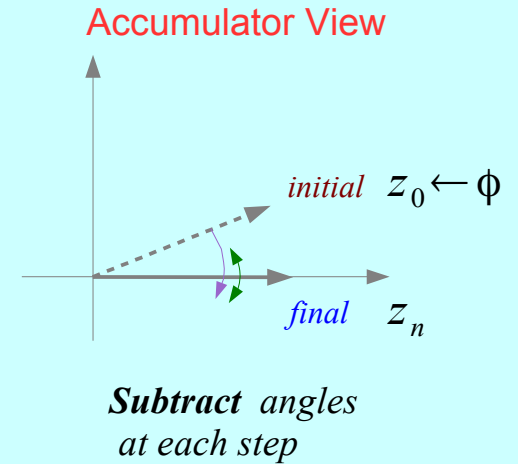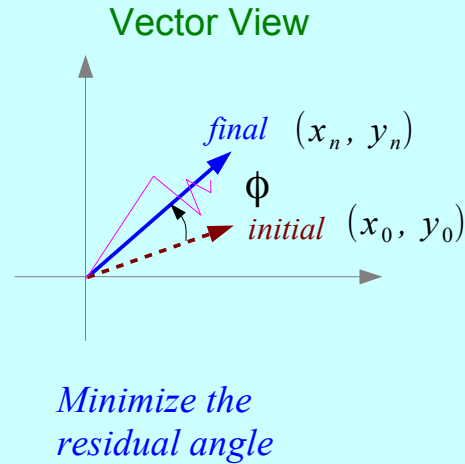
$$z_0 \leftarrow \phi \quad (desired\ angle)$$

$$z_n \rightarrow 0$$

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = -1 \quad if \quad z_i < 0$$

$$d_i = +1 \quad otherwise$$

Vector View

final (x_n, y_n)

ϕ

initial (x_0, y_0)

Minimize the
residual angle

Accumulator View

initial $z_0 \leftarrow \phi$

final z_n

Subtract angles
at each step

Motion Correction and Control System

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = A_n \cdot \begin{bmatrix} \cos z_0 & -\sin z_0 \\ \sin z_0 & \cos z_0 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

* n-dim rotation
$\rightarrow$ tree architecture

Unscaled Rotation

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = A_n \cdot \begin{bmatrix} \cos z_0 & -\sin z_0 \\ \sin z_0 & \cos z_0 \end{bmatrix} \begin{bmatrix} \frac{x_0}{A_n} \\ \frac{y_0}{A_n} \end{bmatrix} \rightarrow$$ A pair of MULT $\Rightarrow$ $$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} \cos z_0 & -\sin z_0 \\ \sin z_0 & \cos z_0 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$
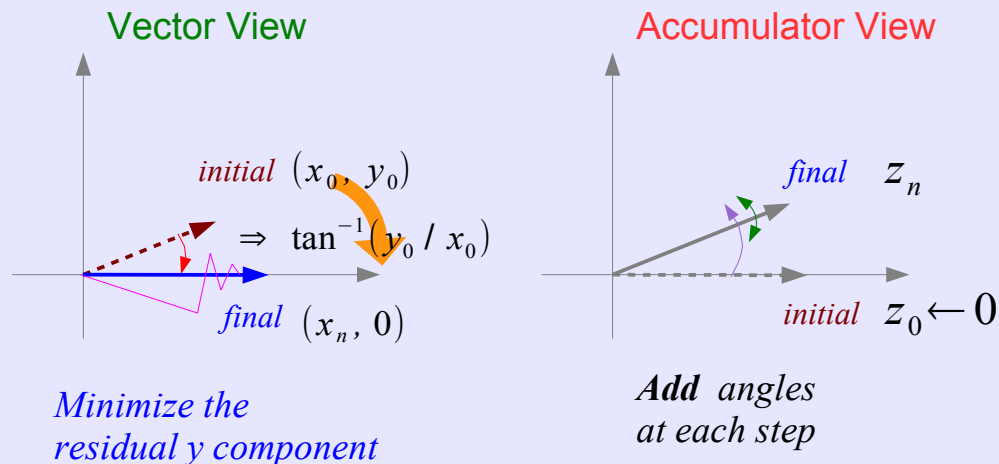
D. Arctangent

Vectoring Mode

$$z_0 \leftarrow 0$$

$$z_n \rightarrow z_0 + \tan^{-1}(y_0/x_0)$$

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = +1 \quad if \quad y_i < 0$$

$$d_i = -1 \quad otherwise$$

Vector View

initial $(x_0,\ y_0)$

$\Rightarrow \quad \tan^{-1}(y_0 / x_0)$

final $(x_n,\ 0)$

Minimize the residual y component

Accumulator View

final $\quad z_n$

initial $\quad z_0 \leftarrow 0$

Add *angles at each step*

Input

$$(x_0,\ y_0) \quad \rightarrow \quad ratio \quad \frac{y_0}{x_0}$$

$$(0,\ y_0) \quad \rightarrow \quad ratio \quad \pm\infty$$

Output

Angle Accumulator Value

$\rightarrow$ *CORDIC gain does not affect*

$$x_n = z_0 + \tan^{-1}(y_0 / x_0)$$
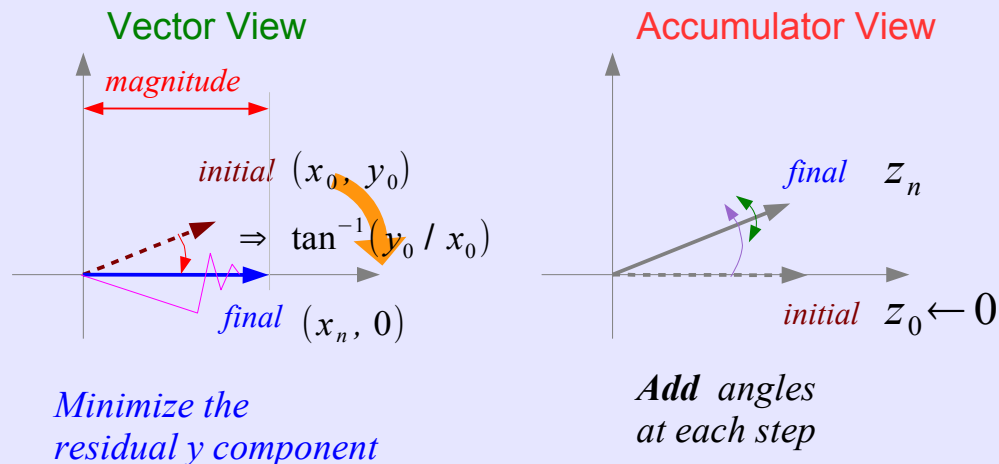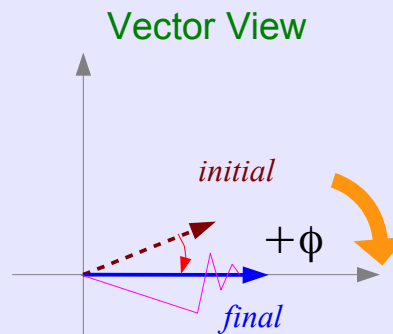
E. Vector Magnitude

Vectoring Mode

$$z_0 \leftarrow 0$$

$$z_n \rightarrow z_0 + \tan^{-1}(y_0/x_0)$$

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = +1 \quad if \quad y_i < 0$$

$$d_i = -1 \quad otherwise$$

Vector View

magnitude

initial (x_0, y_0)

$\Rightarrow \quad \tan^{-1}(y_0 / x_0)$

final $(x_n, 0)$

Minimize the
residual y component

Accumulator View

final z_n

initial $z_0 \leftarrow 0$

Add *angles*
at each step

The magnitude:

- *byproduct of computing arctangent*
- *the result vector is aligned with x-axis*
- *the x component of the result vector*
- *increased by CORDIC gain*
- *can be scaled by the processor gain*
- *one MULT hardware cost*

$$x_n = A_n \sqrt{x_0{}^2 + y_0{}^2}$$

The accuracy of the magnitude result

- *Improves by 2 bits for each iteration performed*

F. Cartesian to Polar Transformation

Vectoring Mode

$$z_0 \leftarrow 0$$
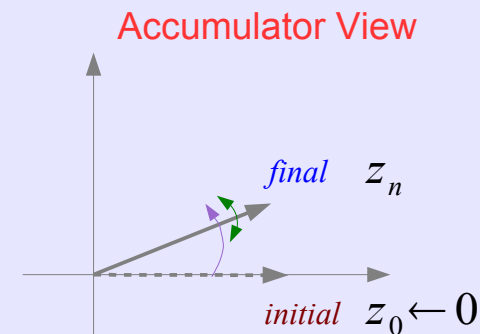
$$\boxed{z_n \rightarrow z_0 + \tan^{-1}(y_0/x_0)}$$

$$\boxed{z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})}$$

$$d_i = +1 \quad if \quad y_i < 0$$

$$d_i = -1 \quad otherwise$$

Vector View

initial

$+\phi$

final

Minimize the
residual y component

Accumulator View

final z_n

initial $z_0 \leftarrow 0$

Add *angles*
at each step

$$input\ vector \quad (x,\ y)$$

$$magnitude \quad r = \sqrt{x^2 + y^2} \qquad \Rightarrow \qquad x_n = A_n \sqrt{x_0^2 + y_0^2}$$

$$phase\ angle \quad \phi = \tan^{-1}(y\ /\ x) \qquad \Rightarrow \qquad z_n = z_0 + \tan^{-1}(y_0\ /\ x_0)$$
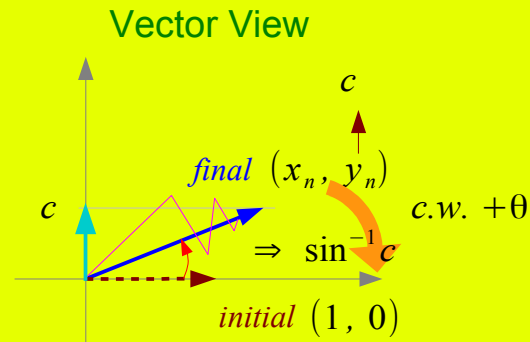
G. ArcSine (1)

Exploit Vector Mode HW

$$z_n \rightarrow z_0 + \sin^{-1}\left(\frac{c}{A_n \cdot x_0}\right)$$
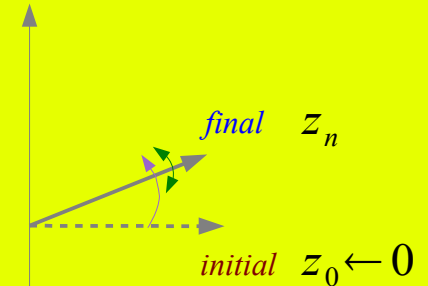
$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = +1 \quad if \quad y_i < c$$

$$d_i = -1 \quad otherwise$$

Vector View

final (x_n, y_n)

c.w. $+\theta$

$\Rightarrow \sin^{-1} c$

initial $(1, 0)$

Minimize the residual y component

Accumulator View

final z_n

initial $z_0 \leftarrow 0$
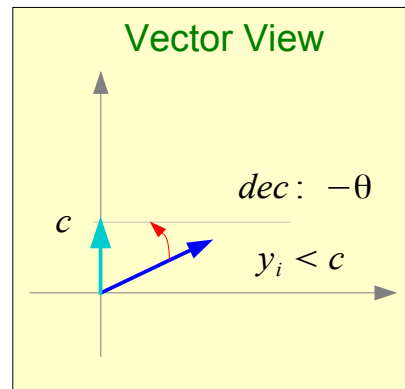
Add angles at each step

$$x_{i+1} = x_i - y_i \cdot d_i \cdot 2^{-i}$$

$$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i}$$

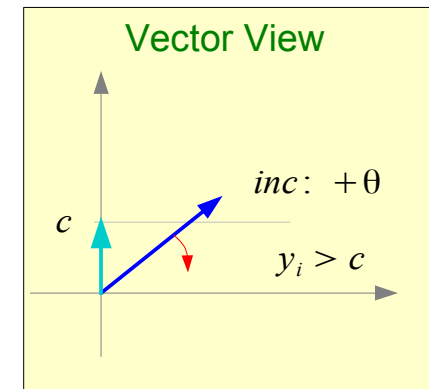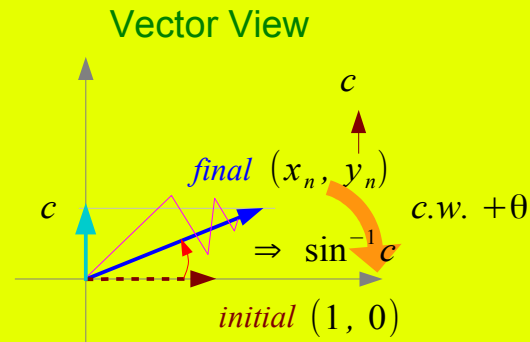$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = +1 \quad if \quad y_i < c$$

$$d_i = -1 \quad otherwise$$

Vector View

dec: $-\theta$

$y_i < c$

$y_i < c$ Dec Angle

Add (−) Angle

Vector View

inc: $+\theta$

$y_i > c$

$y_i > c$ Inc Angle

Add (+) Angle

G. ArcSine (2)

$$x_{i+1} = x_i - y_i \cdot d_i \cdot 2^{-i}$$

$$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i}$$

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = +1 \quad if \quad y_i < c$$

$$d_i = -1 \quad otherwise$$

$$x_n = \sqrt{(A_n \cdot x_0)^2 - c^2}$$

$$y_n = c$$

$$z_n = z_0 + \sin^{-1}\left(\frac{c}{A_n \cdot x_0}\right)$$

$$A_n = \prod_{i=1}^{n} \sqrt{1 + 2^{-2i}}$$

H. Arccosine (1)

$$x_{i+1} = x_i - y_i \cdot d_i \cdot 2^{-i}$$

$$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i}$$

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = +1 \quad if \quad x_i > c$$

$$d_i = -1 \quad otherwise$$

Vector View

$inc: +\theta$

$x_i < c$

$x_i < c$ Inc Angle

Add (+) Angle

Vector View

$dec: -\theta$

$x_i > c$

$x_i > c$ Dec Angle

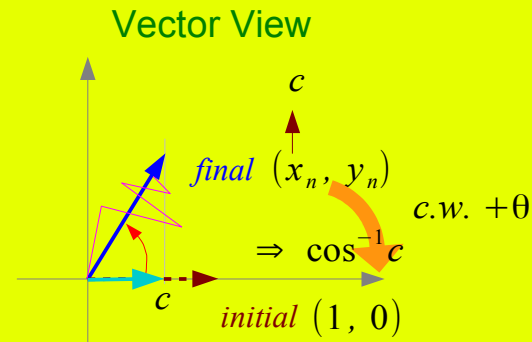Add (−) Angle

H. Arccosine (1)

Exploit Vector Mode HW

$$z_n \rightarrow z_0 + \sin^{-1}\left(\frac{c}{A_n \cdot x_0}\right)$$

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = +1 \quad if \quad y_i < c$$

$$d_i = -1 \quad otherwise$$

Vector View

c

$final\ (x_n, y_n)$

$c.w.\ +\theta$

$\Rightarrow \cos^{-1} c$

c

$initial\ (1, 0)$

Minimize the
residual x component

Accumulator View

$final \quad z_n$

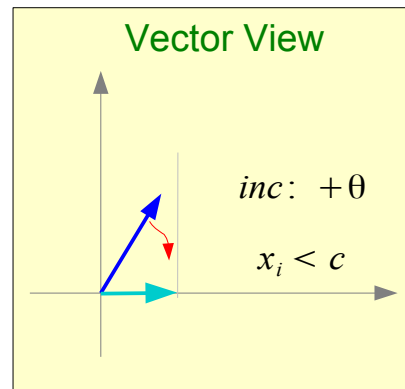$initial \quad z_0 \leftarrow 0$

Add *angles*
at each step

$$x_{i+1} = x_i - y_i \cdot d_i \cdot 2^{-i}$$
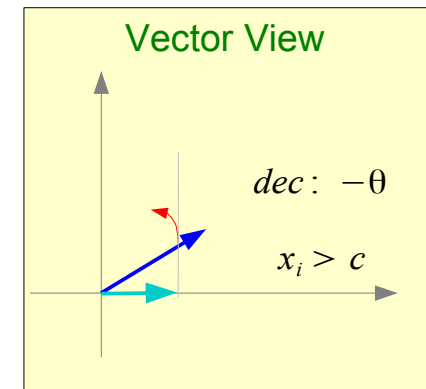
$$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i}$$

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = +1 \quad if \quad x_i > c$$

$$d_i = -1 \quad otherwise$$

$$y_n = \sqrt{(A_n \cdot y_0)^2 - c^2}$$

$$x_n = c$$

$$z_n = z_0 + \cos^{-1}\left(\frac{c}{A_n \cdot y_0}\right)$$

$$A_n = \prod_{i=1}^{n} \sqrt{1 + 2^{-2i}}$$

2A Background

25

I. Linear Functions (1)

Rotation Mode

$z_0 \leftarrow \phi \quad (desired\ angle)$

$z_n \rightarrow 0$

$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$

$d_i = -1 \quad if \quad z_i < 0$

$d_i = +1 \quad otherwise$

Vector View

final

$+\phi$

initial

Minimize the residual angle

Accumulator View

initial $z_0 \leftarrow \phi$
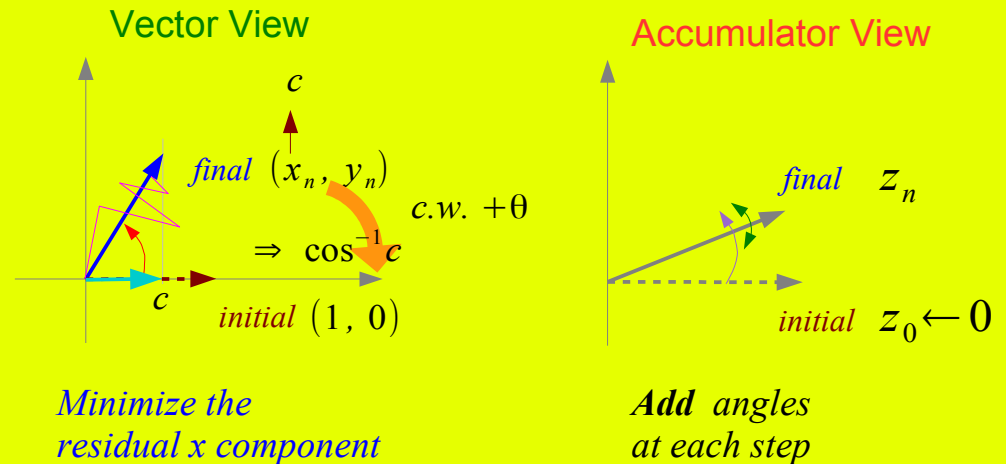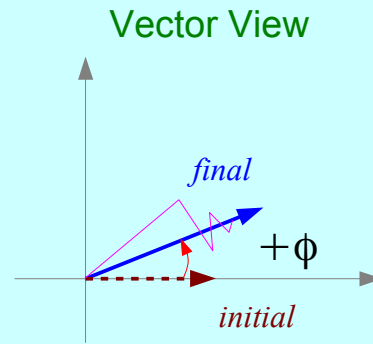
final z_n

Subtract angles at each step

$$x_{i+1} = x_i - y_i \cdot d_i \cdot 2^{-i}$$

$$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i}$$

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$x_{i+1} = x_i - \mathbf{0} \cdot y_i \cdot d_i \cdot 2^{-i} = x_i$$

$$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i}$$

$$z_{i+1} = z_i - d_i \cdot (2^{-i})$$

$$x_n = A_n \left[x_0 \cos z_0 - y_0 \sin z_0 \right]$$

$$y_n = A_n \left[y_0 \cos z_0 + x_0 \sin z_0 \right]$$

$$z_n = 0$$

$$x_n = x_0$$

$$y_n = y_0 + x_0 z_0$$

$$z_n = 0$$

I. Linear Functions (2)

Vectoring Mode

$$z_0 \leftarrow 0$$

$$z_n \rightarrow z_0 + \tan^{-1}(y_0/x_0)$$
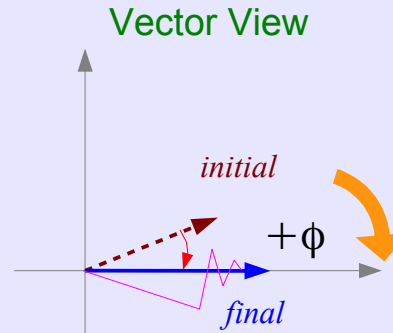
$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = +1 \quad if \quad y_i < 0$$

$$d_i = -1 \quad otherwise$$

Vector View

initial

$+\phi$

final

Minimize the residual y component

Accumulator View

final z_n

initial $z_0 \leftarrow 0$

Add *angles at each step*

$$x_{i+1} = x_i - y_i \cdot d_i \cdot 2^{-i}$$

$$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i}$$

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$x_{i+1} = x_i - \mathbf{0} \cdot y_i \cdot d_i \cdot 2^{-i} = x_i$$

$$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i}$$

$$z_{i+1} = z_i - d_i \cdot (2^{-i})$$

$$x_n = A_n \sqrt{x_0^2 + y_0^2}$$

$$y_n = 0$$

$$z_n = z_0 + \tan^{-1}(y_0/x_0)$$

$$x_n = x_0$$

$$y_n = 0$$

$$z_n = z_0 - (y_0/x_0)$$

J. Hyperbolic Functions (1)

Rotation Mode

$z_0 \leftarrow \phi$ (*desired angle*)

$z_n \rightarrow 0$
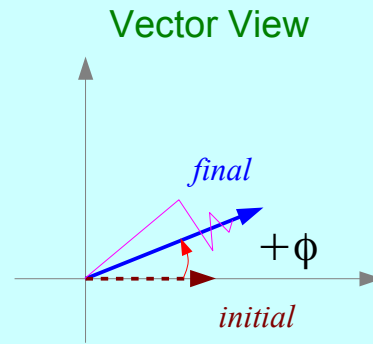
$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$

$d_i = -1 \quad if \quad z_i < 0$

$d_i = +1 \quad otherwise$

Vector View

final

$+\phi$

initial

Minimize the residual angle

Accumulator View

initial $z_0 \leftarrow \phi$

final z_n

Subtract angles at each step

$x_{i+1} = x_i - y_i \cdot d_i \cdot 2^{-i}$

$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i}$

$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$

$x_{i+1} = x_i + y_i \cdot d_i \cdot 2^{-i}$

$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i}$

$z_{i+1} = z_i - d_i \cdot \tanh^{-1}(2^{-i})$

$x_n = A_n \left[x_0 \cos z_0 - y_0 \sin z_0 \right]$

$y_n = A_n \left[y_0 \cos z_0 + x_0 \sin z_0 \right]$

$z_n = 0$

$x_n = A_n \left[x_0 \cosh z_0 - y_0 \sinh z_0 \right]$

$y_n = A_n \left[y_0 \cosh z_0 + x_0 \sinh z_0 \right]$

$z_n = 0$

2A Background

28

J. Hyperbolic Functions (1)

Vectoring Mode

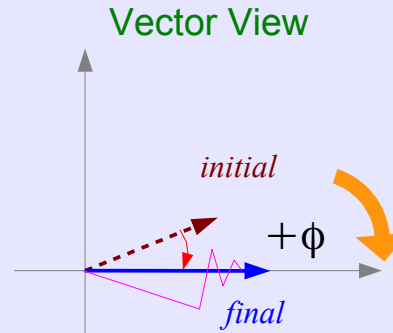$$z_0 \leftarrow 0$$

$$z_n \rightarrow z_0 + \tan^{-1}(y_0/x_0)$$

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = +1 \quad if \quad y_i < 0$$

$$d_i = -1 \quad otherwise$$

Vector View

initial

$+\phi$

final

Minimize the
residual y component

Accumulator View

final z_n

initial $z_0 \leftarrow 0$

Add *angles*
at each step

$$x_{i+1} = x_i - y_i \cdot d_i \cdot 2^{-i}$$

$$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i}$$

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$x_{i+1} = x_i + y_i \cdot d_i \cdot 2^{-i}$$

$$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i}$$

$$z_{i+1} = z_i - d_i \cdot \mathbf{tanh}^{-1}(2^{-i})$$

$$x_n = A_n \sqrt{x_0^2 + y_0^2}$$

$$y_n = 0$$

$$z_n = z_0 + \tan^{-1}(y_0/x_0)$$

$$x_n = A_n \sqrt{x_0^2 - y_0^2}$$

$$y_n = 0$$

$$z_n = z_0 + \mathbf{tanh}^{-1}(y_0/x_0)$$

Unified CORDIC Iteration Eq

$$x_{i+1} = x_i - \boldsymbol{m} \cdot y_i \cdot d_i \cdot 2^{-i}$$

$$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i}$$

$$z_{i+1} = z_i - d_i \cdot \boldsymbol{e}_i$$

$$\boldsymbol{m} = 1 \quad \Rightarrow \quad \boldsymbol{e}_i = \tan^{-1}(2^{-i})$$

$$\boldsymbol{m} = 0 \quad \Rightarrow \quad \boldsymbol{e}_i = (2^{-i})$$

$$\boldsymbol{m} = -1 \quad \Rightarrow \quad \boldsymbol{e}_i = \tanh^{-1}(2^{-i})$$

$$\tan \alpha = \frac{\sin \alpha}{\cos \alpha}$$

$$\tanh \alpha = \frac{\sinh \alpha}{\cosh \alpha}$$

$$\exp \alpha = \sinh \alpha + \cosh \alpha$$

$$\ln \alpha = 2\tanh^{-1}(y/x)$$
$$x = \alpha + 1$$
$$y = \alpha - 1$$

$$(\alpha)^{1/2} = (x^2 - y^2)^{1/2}$$
$$x = \alpha + 1/4$$
$$y = \alpha - 1/4$$

Unified CORDIC Iteration Eq

$$x_{i+1} = x_i - \boldsymbol{m} \cdot y_i \cdot d_i \cdot 2^{-i}$$

$$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i}$$

$$z_{i+1} = z_i - d_i \cdot \boldsymbol{e}_i$$

$$\boldsymbol{m} = 1 \quad \Rightarrow \quad \boldsymbol{e}_i = \tan^{-1}(2^{-i})$$

$$\boldsymbol{m} = 0 \quad \Rightarrow \quad \boldsymbol{e}_i = (2^{-i})$$

$$\boldsymbol{m} = -1 \quad \Rightarrow \quad \boldsymbol{e}_i = \tanh^{-1}(2^{-i})$$

$$\boldsymbol{\cosh} i\, x = \frac{1}{2}(e^{ix} + e^{-ix}) = \cos x \qquad \boldsymbol{\cosh} x = \frac{1}{2}(e^{x} + e^{-x}) = \cos i\, x$$

$$\boldsymbol{\sinh} i\, x = \frac{1}{2}(e^{ix} - e^{-ix}) = i \sin x \qquad \boldsymbol{\sinh} x = \frac{1}{2}(e^{x} - e^{-x}) = -i \sin ix$$

$$\boldsymbol{\tanh} i\, x = \frac{(e^{ix} + e^{-ix})}{(e^{ix} - e^{-ix})} = i \tan x \qquad \boldsymbol{\tanh} x = \frac{(e^{x} + e^{-x})}{(e^{x} - e^{-x})} = -i \tan i\, x$$

Unified CORDIC Iteration Eq

References

[1] http://en.wikipedia.org/
[2] CORDIC FAQ, www.dspguru.com
[3] R. Andraka, A survey of CORDIC algorithms for FPGA based computers
[4] J. S. Walther, A Unified Algorithm for Elementary Functions