

Chapter 1

CORDIC

1.1 Algoritmo CORDIC

El algoritmo CORDIC propuesto por Volder realizaba rotaciones en coordenadas circulares. Partiendo de esa base es facil extender su funcionamiento para que realice rotaciones en coordenadas hiperbólicas y lineales. Para lograrlo se agrega una variables que modifica las ecuaciones y además se eligen diferentes angulos para el acumulador de angule (variable z del algoritmo). Las ecuaciones del CORDIC completo son:

$$\begin{aligned}x_{n+1} &= x_n - m \cdot d_n \cdot 2^{-s_{m,n}} \\y_{n+1} &= y_n + d_n \cdot 2^{-s_{m,n}} \\z_{n+1} &= z_n - d_n \cdot \alpha_{m,n}\end{aligned}\tag{1.2}$$

Donde N representa la cantidad de pasos del algoritmo y se cumple que $n = 0, 1, 2, \dots, N-1$. $s_{m,n}$ es una secuencia de números enteros no decreciente llamada *shift sequence* y $\alpha_{m,n}$ representa los angulos rotados para las diferentes coordenadas. d_n es una variable de control que maneja los sumadores/restadores.

$$\alpha_{m,n} = \frac{1}{\sqrt{m}} \cdot \tan^{-1}(\sqrt{m} \cdot 2^{-s_{m,n}})\tag{1.3}$$

Eligiendo correctamente d_n y $s_{m,n}$ el algoritmo converge (ver tabla 1.1). La tabla 1.2 muestra las diferentes salidas del algoritmo. En la table ?? podemos ver las salidas del algoritmo para valores de entrada particulares, estos valores son de especial interés ya que representan operaciones matematicas dificiles de calcular.

Sistema de coordenadas	Shift sequence	Convergencia	Factor de escala
m	$s_{m,n}$	$ A_0 $	$K_m(n \rightarrow \inf)$
1	0,1,2, ... , n	1.74	1.64676
0	1,2,3, ... , n+1	1.0	1.0
-1	1,2,3,4,4, ...	1.13	0.82816

Table 1.1: CORDIC shift sequences

m	Modo	Entradas	Salidas
1	rotation	$x_0 = x$ $y_0 = y$ $z_0 = \theta$	$x_N = K_1 \cdot (x \cos \theta - y \sin \theta)$ $y_N = K_1 \cdot (y \cos \theta + x \sin \theta)$ $z_N = 0$
1	vectoring	$x_0 = x$ $y_0 = y$ $z_0 = \theta$	$x_N = K_1 \cdot \text{sign}(x) \cdot \sqrt{x^2 + y^2}$ $y_N = 0$ $z_N = \theta + \tan^{-1}(y/x)$
0	rotation	$x_0 = x$ $y_0 = y$ $z_0 = \theta$	$x_N = x$ $y_N = y + x \cdot z$ $z_N = 0$
0	vectoring	$x_0 = x$ $y_0 = y$ $z_0 = z$	$x_N = x$ $y_N = 0$ $z_N = z + y/x$
-1	rotation	$x_0 = x$ $y_0 = y$ $z_0 = \theta$	$x_N = K_{-1} \cdot (x \cosh \theta + y \sinh \theta)$ $y_N = K_{-1} \cdot (y \cosh \theta + x \sinh \theta)$ $z_N = 0$
-1	vectoring	$x_0 = x$ $y_0 = y$ $z_0 = \theta$	$x_N = K_{-1} \cdot \text{sign}(x) \cdot \sqrt{x^2 - y^2}$ $y_N = 0$ $z_N = \theta + \tanh^{-1}(y/x)$

Table 1.2: Salidas del algoritmo CORDIC.

Chapter 2

Algoritmo de Briggs para $\ln(x)$

Si se encuentra una secuencia d_k tal que la productoria de x con $(1 + d_k 2^{-k})$ es cercana a 1 entonces vale que:

$$\begin{aligned} x \prod_{k=1}^n (1 + d_k 2^{-k}) &\approx 1 \\ \ln(x) &\approx - \sum_{k=1}^n \ln(1 + d_k 2^{-k}) \end{aligned} \tag{2.2}$$

Chapter 3

BKM

3.1 Origenes

Consideremos el paso básico del algoritmo CORDIC en modo trigonométrico (con $m=1$). Si definimos el número complejo $E_n = x_n + j y_n$ con $j = \sqrt{-1}$, obtenemos $E_{n+1} = E_n(1 + j d_n 2^{-n})$, esta relación es similar al paso básico del algoritmo de Briggs. Esta similitud nos lleva a una generalización de ese algoritmo: podríamos realizar multiplicaciones por términos $(1 + d_n 2^{-n})$, donde los d_n s son números complejos elegidos de tal manera que la multiplicación por d_n pueda reducirse a unas pocas sumas. Entonces se define el algoritmo BKM de la siguiente manera:

$$\begin{cases} E_{n+1} = E_n \cdot (1 + d_n 2^{-n}) \\ L_{n+1} = L_n - \ln(1 + d_n 2^{-n}) \end{cases} \quad (3.1)$$

con $d_n \in \{0, \pm 1, \pm j, \pm 1 \pm j\}$ y $\ln z$ es el número complejo t tal que $\exp t = z$ y cuya parte imaginaria está entre $-\pi$ y π .

3.2 E-mode

Encontrar una secuencia de d_n tal que $L_n \rightarrow 0$ entonces $E_n \rightarrow E_1 e^{L_1}$.

$$\begin{cases} E_n \rightarrow E_1 e^{L_1} \\ L_n \rightarrow 0 \end{cases} \quad (3.2)$$

3.2.1 Canal de datos

$$\begin{cases} E_{n+1}^x = E_n^x + (d_n^x E_n^x - d_n^y E_n^y) 2^{-n} \\ E_{n+1}^y = E_n^y + (d_n^x E_n^y + d_n^y E_n^x) 2^{-n} \end{cases} \quad (3.3)$$

3.2.2 Canal de control

$$\begin{cases} l_n = 2^n L_n \\ l_{n+1} = 2l_n - 2^{n+1} \ln(1 + d_n 2^{-n}) \\ l_{n+1}^x = 2l_n^x - 2^n \ln[1 + d_n^x 2^{-n+1} - (d_n^{x2} + d_n^{y2}) 2^{-2n}] \\ l_{n+1}^y = 2l_n^y - 2^{n+1} d_n^y \arctan\left(\frac{2^{-n}}{1 + d_n^x 2^{-n}}\right) \end{cases} \quad (3.4)$$

$$2^n \ln(1 + d_n 2^{-n}) \rightarrow 1 \quad \text{as } n \rightarrow +\infty \quad (3.5)$$

3.3 L-mode

Encontrar una sequencia de d_n tal que $E_n \rightarrow 1$ entonces $L_n \rightarrow L_1 + \ln(E_1)$.

$$\begin{cases} E_n \rightarrow 1 \\ L_n \rightarrow L_1 + \ln(E_1) \end{cases} \quad (3.6)$$

3.3.1 Canal de datos

$$\begin{cases} L_{n+1}^x = L_n^x - \frac{1}{2} \ln[1 + d_n^x 2^{-n+1} - (d_n^{x2} + d_n^{y2}) 2^{-2n}] \\ L_{n+1}^y = L_n^y - d_n^y \arctan\left(\frac{2^{-n}}{1 + d_n^x 2^{-n}}\right) \end{cases} \quad (3.7)$$

3.3.2 Canal de control

$$\begin{cases} e_n = 2^n (E_n - 1) \\ e_{n+1} = 2(e_n + d_n) + d_n e_n 2^{-n+1} \\ e_{n+1}^x = 2(e_n^x + d_n^x) + (d_n^x e_n^x - d_n^y e_n^y) 2^{-n+1} \\ e_{n+1}^y = 2(e_n^y + d_n^y) + (d_n^y e_n^x + d_n^x e_n^y) 2^{-n+1} \end{cases} \quad (3.8)$$