

9

CORDIC algoritam

CORDIC (COordinate Rotation DIgital Computer) algoritam [54, 55, 56] je razvijen za potrebe izračunavanja trigonometrijskih funkcija korišćenjem male tabele konstanti i operacija sabiranja, oduzimanja i pomeranja. Algoritmom se mogu izračunavati trigonometrijske funkcije $\sin(x)$, $\cos(x)$, inverzne trigonometrijske funkcije $\arcsin(x)$, $\arccos(x)$, $\arctan(y/x)$, i modul vektora $\sqrt{x^2 + y^2}$. Pored trigonometrijskih funkcija, generalizovanim CORDIC algoritmom se mogu izračunavati hiperboličke funkcije $\sinh(x)$ i $\cosh(x)$ i elementarne funkcije xy , x/y , \sqrt{x} , e^x , $\log(x)$.

Algoritam se može efikasno implementirati u softveru i na mikrokontrolerima bez hardverskih množača, pošto se ne koriste operacije množenja, i za N bita tačnosti složenost algoritma je $O(N)$. Hardverske implementacije CORDIC algoritma mogu biti iterativne uz malo zauzeće resursa, ili protočne kada je potrebna velika brzina izračunavanja.

Softverske i hardverske implementacije CORDIC algoritma se koriste za:

- Izračunavanje elementarnih funkcija
 - Kalkulatori, matematički koprocesori,
 - Softverske biblioteke
- Transformacije koordinatnog sistema
 - $(x, y) \rightarrow (\rho, \phi)$, $\rho = \sqrt{x^2 + y^2}$, $\phi = \arctan(y/x)$
 - $(\rho, \phi) \rightarrow (x, y)$, $x = \rho \cos(\phi)$, $y = \rho \sin(\phi)$
- Hardverska implementacija dekompozicije matrice
 - QR, SVD
- Numerički kontrolisani oscilator
 - DDS - Direct Digital Synthesis
- Kompleksni mešač
 - Pomeranje spektra bez generisanja bočnih opsega
- Digitalna modulacija/demodulacija
 - AM, PM, FM, ASK, PSK, FSK, QAM

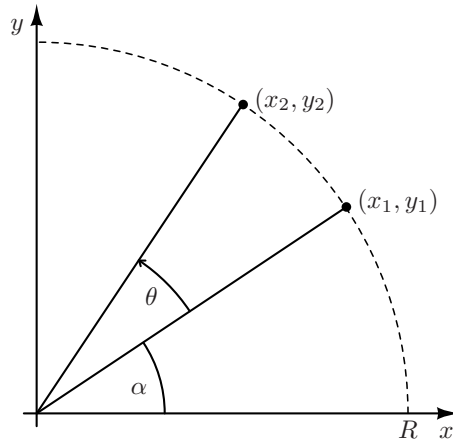
9.1 Rotacija vektora u ravni

Vrednosti trigonometrijskih funkcija se mogu izračunati rotacijom jediničnog vektora $[1, 0]^T$ u ravni za ugao θ , pošto je rezultat rotacije vektor $[\cos(\theta), \sin(\theta)]^T$. Rotacija vektora u ravni za ugao θ se može izvršiti u jednom koraku ili iterativno, izvršavanjem

mikro rotacija po uglovima θ_i , gde je $\theta = \sum \theta_i$. Osnovna ideja CORDIC algoritma je u izboru uglova θ_i za koje se mikro rotacije mogu efikasno izračunati.

Rotacijom vektora (x_1, y_1) za ugao θ dobija se vektor (x_2, y_2)

$$\begin{aligned}
 x_1 &= R \cos(\alpha) \\
 y_1 &= R \sin(\alpha) \\
 x_2 &= R \cos(\alpha + \theta) \\
 &= \underbrace{R \cos \alpha}_{x_1} \cos \theta - \underbrace{R \sin \alpha}_{y_1} \sin \theta \\
 &= x_1 \cos \theta - y_1 \sin \theta \\
 y_2 &= R \sin(\alpha + \theta) \\
 &= \underbrace{R \cos \alpha}_{x_1} \sin \theta + \underbrace{R \sin \alpha}_{y_1} \cos \theta \\
 &= x_1 \sin \theta + y_1 \cos \theta
 \end{aligned} \tag{9.1}$$



Slika 9.1: Rotacija vektora u ravni.

Rotacija vektora $[x_1, y_1]^T$ za ugao θ se može predstaviti u matičnom obliku

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \mathbf{ROT}(\theta) \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \tag{9.2}$$

gde je rotaciona matrica $\mathbf{ROT}(\theta)$ data sa:

$$\mathbf{ROT}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}. \tag{9.3}$$

9.2 Svojstva rotacione matrice

Pregled svojstava rotacione matrice $\mathbf{ROT}(\theta)$ je sažet u narednim odeljcima.

9.2.1 Determinanta rotacione matrice

Determinanta rotacione matrice se može direktno izračunati iz definicije:

$$\det [\mathbf{ROT}(\alpha)] = \cos^2 \alpha + \sin^2 \alpha = 1. \quad (9.4)$$

Pošto je $\det [\mathbf{ROT}(\alpha)] = 1$, rotaciona matrica ne menja modul vektora već samo njegov pravac.

9.2.2 Ortogonalnost rotacione matrice

Ortogonalnost rotacione matrice se može iskazati kao

$$\mathbf{ROT}(\alpha)^T = \mathbf{ROT}(\alpha)^{-1}, \quad (9.5)$$

odakle sledi $\mathbf{ROT}(\alpha)\mathbf{ROT}(\alpha)^T = \mathbf{I}$, što se može direktno pokazati:

$$\begin{aligned} \mathbf{ROT}(\alpha)\mathbf{ROT}(\alpha)^T &= \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \\ &= \begin{bmatrix} \cos^2 \alpha + \sin^2 \alpha & 0 \\ 0 & \cos^2 \alpha + \sin^2 \alpha \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{I} \end{aligned} \quad (9.6)$$

Takođe važi i

$$\mathbf{ROT}(-\alpha) = \mathbf{ROT}(\alpha)^{-1}. \quad (9.7)$$

Iz (9.7) sledi $\mathbf{ROT}(\alpha)\mathbf{ROT}(-\alpha) = \mathbf{I}$, što je i očekivano pošto se rotacije za isti ugao u suprotnim smerovima poništavaju.

9.2.3 Komutativnost rotacione matrice

Komutativnost rotacione matrice

$$\mathbf{ROT}(\alpha)\mathbf{ROT}(\beta) = \mathbf{ROT}(\beta)\mathbf{ROT}(\alpha), \quad (9.8)$$

se može pokazati direktno:

$$\begin{aligned}
 \mathbf{ROT}(\alpha)\mathbf{ROT}(\beta) &= \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{bmatrix} \\
 &= \begin{bmatrix} \cos \alpha \cos \beta - \sin \alpha \sin \beta & -\cos \alpha \sin \beta - \sin \alpha \cos \beta \\ \sin \alpha \cos \beta + \cos \alpha \sin \beta & -\sin \alpha \sin \beta + \cos \alpha \cos \beta \end{bmatrix}, \quad (9.9) \\
 &= \begin{bmatrix} \cos(\alpha + \beta) & -\sin(\alpha + \beta) \\ \sin(\alpha + \beta) & \cos(\alpha + \beta) \end{bmatrix} = \mathbf{ROT}(\alpha + \beta)
 \end{aligned}$$

$$\begin{aligned}
 \mathbf{ROT}(\beta)\mathbf{ROT}(\alpha) &= \begin{bmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{bmatrix} \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \\
 &= \begin{bmatrix} \cos \beta \cos \alpha - \sin \beta \sin \alpha & -\cos \beta \sin \alpha - \sin \beta \cos \alpha \\ \sin \beta \cos \alpha + \cos \beta \sin \alpha & -\sin \beta \sin \alpha + \cos \beta \cos \alpha \end{bmatrix}. \quad (9.10) \\
 &= \begin{bmatrix} \cos(\alpha + \beta) & -\sin(\alpha + \beta) \\ \sin(\alpha + \beta) & \cos(\alpha + \beta) \end{bmatrix} = \mathbf{ROT}(\alpha + \beta)
 \end{aligned}$$

Iz (9.9) i (9.10) dobijamo

$$\mathbf{ROT}(\alpha)\mathbf{ROT}(\beta) = \mathbf{ROT}(\beta)\mathbf{ROT}(\alpha) = \mathbf{ROT}(\alpha + \beta), \quad (9.11)$$

odakle sledi da komutativnost rotacione matrice važi.

9.2.4 Asocijativnost

Asocijativnost rotacione matrice

$$[\mathbf{ROT}(\alpha)\mathbf{ROT}(\beta)]\mathbf{ROT}(\gamma) = \mathbf{ROT}(\alpha)[\mathbf{ROT}(\beta)\mathbf{ROT}(\gamma)], \quad (9.12)$$

se može pokazati korišćenjem (9.11). Leva strana (9.12) se može preurediti u

$$[\mathbf{ROT}(\alpha)\mathbf{ROT}(\beta)]\mathbf{ROT}(\gamma) = \mathbf{ROT}(\alpha + \beta)\mathbf{ROT}(\gamma) = \mathbf{ROT}(\alpha + \beta + \gamma). \quad (9.13)$$

Desna strana (9.12) je

$$\mathbf{ROT}(\alpha)[\mathbf{ROT}(\beta)\mathbf{ROT}(\gamma)] = \mathbf{ROT}(\alpha)\mathbf{ROT}(\beta + \gamma) = \mathbf{ROT}(\alpha + \beta + \gamma), \quad (9.14)$$

odakle sledi da su leva i desna strana (9.12) jednake.

9.2.5 Dekompozicija ugla

Dekompozicija ugla pokazuje da se rotacija za ugao $\theta = \sum \theta_i$ može izvršiti kompozicijom rotacija po uglovima θ_i . Prilikom dokazivanja komutativnosti pokazano je da važi:

$$\mathbf{ROT}(\alpha + \beta) = \mathbf{ROT}(\alpha)\mathbf{ROT}(\beta),$$

Rotacija za ugao $\theta = \sum_{i=0}^N \theta_i$ se može predstaviti kao

$$\mathbf{ROT}\left(\sum_{i=0}^N \theta_i\right) = \mathbf{ROT}\left(\theta_0 + \sum_{i=1}^N \theta_i\right) = \mathbf{ROT}(\theta_0)\mathbf{ROT}\left(\sum_{i=1}^N \theta_i\right). \quad (9.15)$$

Iterativnom primenom dolazimo do zaključka da se rotacija za ugao $\sum_{i=0}^N \theta_i$ može izvršiti u $N + 1$ rotacija za uglove θ_i

$$\mathbf{ROT}\left(\sum_{i=0}^N \theta_i\right) = \prod_{i=0}^N \mathbf{ROT}(\theta_i). \quad (9.16)$$

9.3 Osnovna ideja CORDIC algoritma

Pretpostavimo da se ugao θ se može predstaviti kao:

$$\theta = \sum_i \sigma_i \theta_i, \quad (9.17)$$

gde su θ_i elementarni uglovi i $\sigma_i = \{-1, 1\}$. CORDIC algoritam se može formulisati i za slučaj $\sigma_i = \{-1, 0, 1\}$, ali u tom slučaju multiplikativni faktor $K(n)$ zavisi od ugla. Rotacija vektora za ugao θ se može izvršiti mikrorotacijama:

$$\mathbf{ROT}(\theta) = \mathbf{ROT}\left(\sum \sigma_i \theta_i\right) = \prod_i \mathbf{ROT}(\sigma_i \theta_i). \quad (9.18)$$

Osnovna ideja CORDIC algoritma je u izboru uglova θ_i tako da se mikrorotacija može izračunati samo korišćenjem sabiranja i pomeranja. Mikrorotacija za elementarni ugao $\sigma_i \theta_i$ se može preurediti u oblik:

$$\begin{aligned} \mathbf{ROT}(\sigma_i \theta_i) &= \begin{bmatrix} \cos \sigma_i \theta_i & -\sin \sigma_i \theta_i \\ \sin \sigma_i \theta_i & \cos \sigma_i \theta_i \end{bmatrix} \stackrel{*}{=} \begin{bmatrix} \cos \theta_i & -\sigma_i \sin \theta_i \\ \sigma_i \sin \theta_i & \cos \theta_i \end{bmatrix} \\ &= \cos \theta_i \begin{bmatrix} 1 & -\sigma_i \tan \theta_i \\ \sigma_i \tan \theta_i & 1 \end{bmatrix} \\ &= \frac{1}{\sqrt{1 + \tan^2 \theta_i}} \begin{bmatrix} 1 & -\sigma_i \tan \theta_i \\ \sigma_i \tan \theta_i & 1 \end{bmatrix} \end{aligned} \quad (9.19)$$

U koraku \star su $-\sin \sigma_i \theta_i = -\sigma_i \sin \theta_i$ i $\cos \sigma_i \theta_i = \cos \theta_i$ dobijeni pod pretpostavkom $\sigma_i = \pm 1$. Izborom elementarnog ugla

$$\theta_i = \arctan 2^{-i}, \quad (9.20)$$

dobijamo:

$$\mathbf{ROT}(\sigma_i \theta_i) = \frac{1}{\sqrt{1 + 2^{-2i}}} \begin{bmatrix} 1 & -\sigma_i 2^{-i} \\ \sigma_i 2^{-i} & 1 \end{bmatrix}. \quad (9.21)$$

Zanemarujući multiplikativni faktor $1/\sqrt{1 + 2^{-2i}}$, matrica mikrorotacije sadrži samo operacije promene znaka, i množenje faktorom 2^{-i} , odnosno pomeranje u desno za i bita. Obe operacije se mogu efikasno izvršiti u hardveru ili softveru.

Zamenom (9.21) u (9.18) dobijamo

$$\mathbf{ROT} \left(\sum_i \sigma_i \theta_i \right) = \prod_i \mathbf{ROT}(\sigma_i \theta_i) = \prod_i K_i \prod_i \mathbf{R}_i, \quad (9.22)$$

gde su multiplikativni faktori K_i

$$K_i = \frac{1}{\sqrt{1 + 2^{-2i}}}, \quad (9.23)$$

i matrice mikrorotacija \mathbf{R}_i

$$\mathbf{R}_i = \begin{bmatrix} 1 & -\sigma_i 2^{-i} \\ \sigma_i 2^{-i} & 1 \end{bmatrix}. \quad (9.24)$$

Vrednost multiplikativnog faktora K_i ne zavisi od σ_i , već samo od broja iteracije i , tako da se za zadati broj iteracija ukupan multiplikativni faktor može izračunati:

$$K(n) = \prod_{i=0}^{n-1} K_i = \prod_{i=0}^{n-1} \frac{1}{\sqrt{1 + 2^{-2i}}}. \quad (9.25)$$

Granična vrednost za $n \rightarrow \infty$ iznosi

$$K_\infty = \lim_{n \rightarrow \infty} K(n) \approx 0.60726. \quad (9.26)$$

Korišćenjem (9.25) rotaciona matrica se može izraziti kao:

$$\mathbf{ROT} \left(\sum_i \sigma_i \theta_i \right) = K(n) \prod_{i=0}^{n-1} \mathbf{R}_i. \quad (9.27)$$

Iz (9.27) se vidi da se rotacija vektora za ugao $\theta = \sum \sigma_i \theta_i$ može izvršiti kompozicijom mikrorotacija \mathbf{R}_i , koje sadrže samo promenu znaka, sabiranje i pomeranje u desno, i množenjem rezultata mikrorotacija konstantom $K(n)$. Multiplikativni faktor se može implementirati množačem konstantom, ili zanemariti, što rezultuje pojačanjem $K(n)^{-1}$.

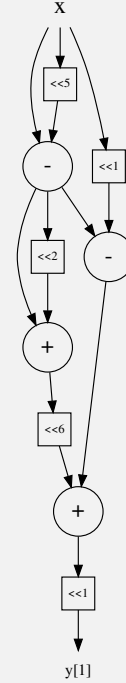
Primer 9.1: Množac konstantom

Množac konstantom se može napraviti pomoću sabirača i pomerača, npr. korišćenjem Butovog algoritma [57] ili algoritima [58, 59, 60] koji su implementirani u [61].

Reprezentacija konstante $K(15) = 0.60725294$ sa 15 bita je $2^{15}K(15) = 19898$. Množac konstantom 19898, prikazan na slici 9.2, se može napraviti korišćenjem 4 sabirača i 5 pomerača. Fiksni pomerači u hardverskim realizacijama ne zahtevaju dodatne resurse, pošto se realizuju povezivanjem.

Primer softverske implementacija množača konstantom je dat u nastavku.

```
t4 = x<<5;    /* 32*/
t3 = t4-x;    /* 31*/
t6 = x<<1;    /* 2*/
t5 = t3-t6;    /* 29*/
t8 = t3<<2;    /* 124*/
t7 = t3+t8;    /* 155*/
t9 = t7<<6;    /* 9920*/
t2 = t5+t9;    /* 9949*/
y = t2<<1;    /* 19898*/
```



Slika 9.2:
 $Y = 19898X$

Zanemarujući multiplikativni faktor $K(n)$, rotacija vektora $[x_p, y_p]^T$ za ugao $\theta = \sum_{i=0}^{n-1} \sigma_i \theta_i$ se može izvršiti u n mikrorotacija:

$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = \prod_{i=0}^{n-1} \mathbf{R}_i \begin{bmatrix} x_p \\ y_p \end{bmatrix} = \mathbf{R}_{n-1} \mathbf{R}_{n-2} \cdots \mathbf{R}_0 \begin{bmatrix} x_p \\ y_p \end{bmatrix}. \quad (9.28)$$

Izraz (9.28) se može izračunati iterativno formiranjem niza

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \mathbf{R}_i \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} 1 & -\sigma_i 2^{-i} \\ \sigma_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}, \quad (9.29)$$

gde su $[x_0 \ y_0]^T = [x_p \ y_p]^T$ i $[x_r \ y_r]^T = [x_n \ y_n]^T$. Množenjem matrice \mathbf{R}_i i vektora $[x_i \ y_i]^T$ dobijamo izraz za iterativno izračunavanje mikrorotacija:

$$\begin{aligned} x_{i+1} &= x_i - \sigma_i y_i 2^{-i} = x_i - \sigma_i y_i \gg i \\ y_{i+1} &= y_i + \sigma_i x_i 2^{-i} = y_i + \sigma_i x_i \gg i' \end{aligned} \quad (9.30)$$

gde $x \gg i$ predstavlja operaciju binarnog pomeranja za i mesta u desno.

Ukupan ugao rotacije vektora $\sigma = \sum \sigma_i \theta_i$ je određen elementarnim uglovima $\sigma_i = \arctan 2^{-i}$ i nizom $\sigma_i = \pm 1$. Pošto se rotacija vektora izvršava iterativno, potrebno je uvesti akumulator ugla z koji će čuvati trenutnu vrednost ugla. Izbor $\sigma_i = \pm 1$ odgovara rotaciji vektora za $\pm \theta_i$, odakle je:

$$z_{i+1} = z_i - \sigma_i \theta_i = z_i - \sigma_i \arctan 2^{-i}.$$

Uglovi mikrorotacija $\theta_i = \arctan 2^{-i}$ se mogu unapred izračunati i tabelirati, kao što je pokazano u primeru 9.2.

Primer 9.2: Vrednosti elementarnih uglova mikrorotacija θ_i

Mikrorotacija \mathbf{R}_i rotira vektor za θ_i :

$$\theta_i = \arctan 2^{-i}.$$

Vrednosti ugla θ_i su date u tabeli za $i = 0..9$. Veličina tabele se može optimizovati pošto je za $i \gg 1$

$$\arctan 2^{-i} \approx 2^{-i}.$$

i	θ_i	
	[rad]	[°]
0	$7.8540 \cdot 10^{-1}$	45.00000
1	$4.6365 \cdot 10^{-1}$	26.56505
2	$2.4498 \cdot 10^{-1}$	14.03624
3	$1.2435 \cdot 10^{-1}$	7.125016
4	$6.2419 \cdot 10^{-2}$	3.576334
5	$3.1240 \cdot 10^{-2}$	1.789911
6	$1.5624 \cdot 10^{-2}$	0.895174
7	$7.8123 \cdot 10^{-3}$	0.447614
8	$3.9062 \cdot 10^{-3}$	0.223811
9	$1.9531 \cdot 10^{-3}$	0.111906

Izbor σ_i je moguće izvršiti na dva načina:

Rotacioni mod

$$\sigma_i = \text{sgn}(z_i)$$

σ_i se bira tako da $z \rightarrow 0$
Vektor se rotira za ugao z_0

Vektorski mod

$$\sigma_i = -\text{sgn}(y_i)$$

σ_i se bira tako da $y \rightarrow 0$
Vektor se rotira na x osu

U oba moda rada izvršava se algoritam sukcesivnih aproksimacija tako što se u rotacionom modu σ_i bira da anulira vrednost rezidualnog ugla z_n , dok se u vektorskom modu σ_i bira tako da se anulira y_n .

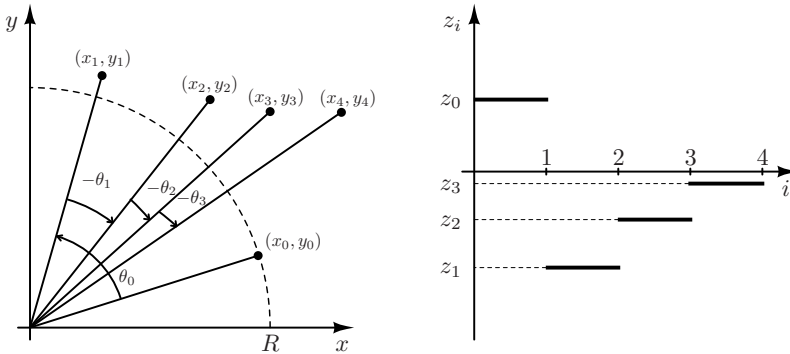
CORDIC iteracija u oba moda rada se može formulisati kao:

$$\begin{aligned}
 x_{i+1} &= x_i - \sigma_i y_i \gg i \\
 y_{i+1} &= y_i + \sigma_i x_i \gg i \\
 z_{i+1} &= z_i - \sigma_i \arctan 2^{-i} = z_i - \sigma_i \theta_i. \\
 \sigma_i &= \begin{cases} \operatorname{sgn}(z_i) & , \text{ rotacioni mod} \\ -\operatorname{sgn}(y_i) & , \text{ vektorski mod} \end{cases}
 \end{aligned} \tag{9.31}$$

9.4 CORDIC algoritam u rotacionom modu

Iteracije za izračunavanje CORDIC algoritma u rotacionom modu su date u (9.32). Izborom $\sigma_i = \operatorname{sgn}(z_i)$ elementarni uglovi θ_i se oduzimaju od trenutne vrednosti ugla z_i ako je $z_i > 0$, odnosno sabiraju ako je $z_i < 0$, usled čega rezidualni ugao z_n teži nuli, kao što je ilustrovano na slici 9.3. U primeru sa slike 9.3 se u prvom koraku vrši rotacija za ugao $\theta_0 > z_0$, i da se sve ostale rotacije vrše u suprotnom smeru, odnosno da je prva rotacija u suštini nepotrebna. Nepotrebne rotacije bi se mogle izbeći ukoliko bi se dozvolilo da se σ_i bira iz skupa $\{-1, 0, 1\}$, i u tom slučaju bi z_i moglo monotono težiti nuli, ali bi ukupan multiplikativni faktor zavisio od ugla rotacije.

$$\begin{aligned}
 x_{i+1} &= x_i - \sigma_i y_i \gg i \\
 y_{i+1} &= y_i + \sigma_i x_i \gg i \\
 z_{i+1} &= z_i - \sigma_i \arctan 2^{-i}. \\
 \sigma_i &= \operatorname{sgn}(z_i)
 \end{aligned} \tag{9.32}$$



Slika 9.3: CORDIC algoritam u rotacionom modu.

U i -toj iteraciji algoritma izvršava se mikrorotacija R_i koja rotira vektor za ugao $\sigma_i \theta_i$ uz povećanje modula vektora za K_i^{-1} . Posle n iteracija vektor $[x_0 \ y_0]^T$ je rotiran *tačno* za ugao $\theta = z_0 - z_n$ i skaliran konstantom $K(n)^{-1}$

$$\begin{aligned}
 x_n &= K(n)^{-1} [x_0 \cos \theta - y_0 \sin \theta] \\
 y_n &= K(n)^{-1} [x_0 \sin \theta + y_0 \cos \theta]
 \end{aligned} \tag{9.33}$$

Rezidualni ugao $z_n = \theta - z_0 = \sum \sigma_i \theta_i - z_0$, koji predstavlja grešku rotacije, se može proizvoljno smanjiti povećavanjem broja iteracija, što je pokazano u odeljku 9.6 o tačnosti i konvergentnosti algoritma.

9.4.1 Izračunavanje trigonometrijskih funkcija

Trigonometrijske funkcije $\cos(\theta)$ i $\sin(\theta)$ se mogu izračunati izvršavanjem CORDIC algoritma u rotacionom modu sa početnim vrednostima $[x_0 \ y_0] = [1 \ 0]^T$ za ugao $z_0 = \theta$, i množenjem sa multiplikativnim faktorom $K(n)$. Množenje multiplikativnim faktorom se može izbeći izborom $[x_0 \ y_0]^T = [K(n) \ 0]^T$. Posle n iteracija rezultat je:

$$\begin{aligned} x_n &= \cos(\theta - z_n) \stackrel{n \gg 1}{\approx} \cos \theta \\ y_n &= \sin(\theta - z_n) \stackrel{n \gg 1}{\approx} \sin \theta \end{aligned} \quad (9.34)$$

Primer 9.3: Izračunavanje cos i sin

CODRIC u rotacionom modu

$n = 15$ iteracija

Početne vrednosti

$$x_0 = K(15) = 0.60725294$$

$$y_0 = 0.0$$

$$z_0 = 1.27409035 \text{ rad } (73^\circ)$$

Krajnje vrednosti

$$x_{15} = 0.29234355$$

$$\cos(z_0 - z_{15}) = 0.29234355$$

$$\cos z_0 = 0.2923\textbf{7170}$$

$$y_{15} = 0.95631336$$

$$\sin(z_0 - z_{15}) = 0.95631336$$

$$\sin z_0 = 0.9563\textbf{0476}$$

$$z_{15} = -0.00002944 \text{ rad}$$

i	x	y	z
0	0.60725294	0.00000000	1.27409035
1	0.60725294	0.60725294	0.48869219
2	0.30362647	0.91087940	0.02504458
3	0.07590662	0.98678602	-0.21993408
4	0.19925487	0.97729769	-0.09557909
5	0.26033598	0.96484426	-0.03316028
6	0.29048736	0.95670876	-0.00192044
7	0.30543593	0.95216990	0.01370329
8	0.29799711	0.95455612	0.00589094
9	0.29426837	0.95572017	0.00198471
10	0.29240173	0.95629491	0.00003159
11	0.29146785	0.95658046	-0.00094497
12	0.29193493	0.95643814	-0.00045669
13	0.29216843	0.95636687	-0.00021255
14	0.29228518	0.95633120	-0.00009048
15	0.29234355	0.95631336	-0.00002944

9.4.2 Rotacija vektora u ravni

Rotacija vektora u ravni se koristi u algoritmu **QR** dekompozicije matrice, koji je predstavljen u odeljku 9.12. CORDIC algoritmom u rotacionom modu se vektor $[x_0 \ y_0]^T$ može rotirati u ravni za ugao θ postavljanjem početne vrednosti akumulatora ugla $z_0 = \theta$.

Izvršavanjem n iteracija CORDIC algoritma u rotacionom modu dobija se:

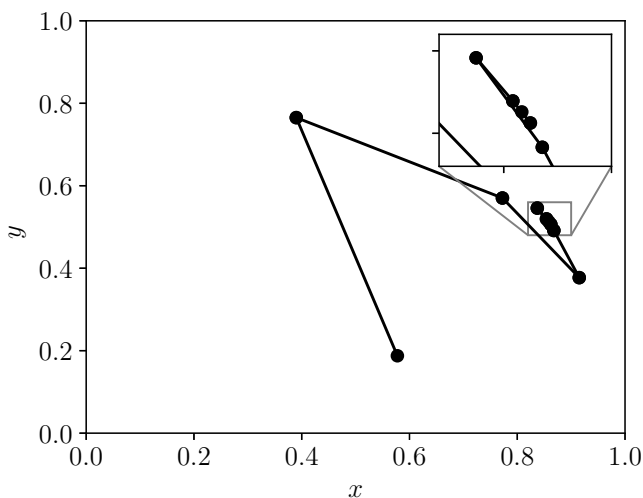
$$\begin{aligned} x_n &= K(n)^{-1} [x_0 \cos(\theta - z_n) - y_0 \sin(\theta - z_n)] \\ y_n &= K(n)^{-1} [x_0 \sin(\theta - z_n) + y_0 \cos(\theta - z_n)] \end{aligned} \quad (9.35)$$

Za mali rezidualni ugao $z_n \approx 0$ ($n \gg 1$) (9.35) se može aproksimirati kao:

$$\begin{aligned} x_n &\stackrel{n \gg 1}{\approx} K(n)^{-1} [x_0 \cos \theta - y_0 \sin \theta] \\ y_n &\stackrel{n \gg 1}{\approx} K(n)^{-1} [x_0 \sin \theta + y_0 \cos \theta] \end{aligned} \quad (9.36)$$

Rotirani vektor je skaliran za $K(n)^{-1}$ i potrebno je koristiti množač konstantom, prikazan u primeru 9.1, da bi se dobila prava vrednost.

Na slici 9.4 je prikazana rotacija vektora $[x_0 \ y_0]^T = K(8)[\cos \alpha \ \sin \alpha]$, gde je $\alpha = 18^\circ$, za ugao $z_0 = 13^\circ$. Početni vektor je pomnožen multiplikativnom konstantom $K(8) \approx 0.607$, tako da rotirani vektor ima jedinični modul. U i -oj iteraciji mikrorotacija R_i rotira vektor za ugao $\sigma_i \theta_i$ i skalira modul vektora za $K_i^{-1} = \sqrt{1 + 2^{-2i}}$. Iz trajektorije vektora se može videti da se rotacije vrše u oba smera, što je posledica restrikcije da se σ_i bira is skupa $\{-1, 1\}$.



Slika 9.4: Rotacija vektora u ravni.

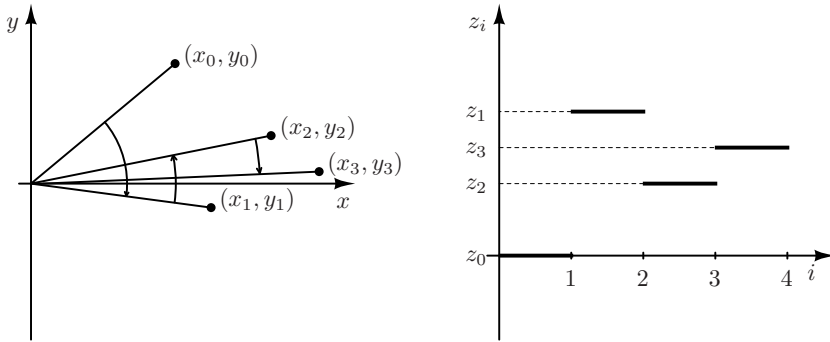
9.5 CORDIC algoritam u vektorskom modu

Iteracija CORDIC algoritma u vektorskom modu je data sa:

$$\begin{aligned} x_{i+1} &= x_i - \sigma_i y_i \gg i \\ y_{i+1} &= y_i + \sigma_i x_i \gg i \\ z_{i+1} &= z_i - \sigma_i \arctan 2^{-i} \\ \sigma_i &= -\text{sgn}(y_i) \end{aligned} \quad (9.37)$$

Izbor $\sigma_i = -\text{sgn}(y_i)$ rotira vektor $[x_i \ y_i]^T$ prema x osi. Trenutna vrednost akumulatora ugla $z_i = z_0 + \sum_{j=0}^{i-1} \sigma_j \theta_j$ sadrži ukupnu vrednost ugla izvršenih rotacija. Posle n iteracija rotirani vektor sa x osom zaklapa ugao manji od θ_n i skaliran je konstantom $K(n)^{-1}$:

$$\begin{aligned} x_n &= K(n)^{-1} \cos(\theta - z_n) \sqrt{x_0^2 + y_0^2} \\ y_n &= K(n)^{-1} \sin(\theta - z_n) \sqrt{x_0^2 + y_0^2} \\ z_n &= z_0 + \sum_{i=0}^{n-1} \sigma_i \theta_i \end{aligned} \quad (9.38)$$



Slika 9.5: CORDIC algoritam u vektorskom modu.

Kako je greška aproksimacije ugla

$$|\theta - z_n| < 2^{-(n-1)}, \quad (9.39)$$

za $n \gg 1$ imamo

$$\begin{aligned} x_n &\approx K(n)^{-1} \sqrt{x_0^2 + y_0^2} \\ y_n &\approx 0 \\ z_n &\approx z_0 + \theta = z_0 + \arctan \frac{y_0}{x_0} \end{aligned} \quad (9.40)$$

Primer 9.4: Izračunavanje $\sqrt{x^2 + y^2}$ i $\arctan(y/x)$

CORDIC u vektorskom modu

 $n = 15$ iteracija

Početne vrednosti

$$x_0 = K(15) \cos 73^\circ$$

$$y_0 = K(15) \sin 73^\circ$$

$$z_0 = 0.00$$

Krajnje vrednosti

$$x_{15} = 1.00000000$$

$$\sqrt{x_0^2 + y_0^2} = 1.00000000$$

$$y_{15} = -0.00002944$$

$$z_{15} = 1.27411980 \text{ rad}$$

$$\arctan \frac{y_0}{x_0} = 1.27409035$$

$$y_{15} \approx z_{15} - \arctan \frac{y_0}{x_0}$$

i	x	y	z
0	0.17754358	0.58071887	0.00000000
1	0.75826245	0.40317529	0.78539816
2	0.95985009	0.02404407	1.24904577
3	0.96586111	-0.21591845	1.49402444
4	0.99285092	-0.09518581	1.36966944
5	0.99880003	-0.03313263	1.30725063
6	0.99983543	-0.00192013	1.27601080
7	0.99986543	0.01370230	1.26038707
8	0.99997248	0.00589085	1.26819941
9	0.99999549	0.00198471	1.27210564
10	0.99999936	0.00003159	1.27405876
11	0.99999940	-0.00094497	1.27503532
12	0.99999986	-0.00045669	1.27454704
13	0.99999997	-0.00021255	1.27430290
14	0.99999999	-0.00009048	1.27418083
15	1.00000000	-0.00002944	1.27411980

9.6 Tačnost i konvergentnost CORDIC algoritma

Maksimalni ugao rotacije θ_{\max} koji se može predstaviti sumom elementarnih uglova $\theta_i = \arctan 2^{-i}$ je:

$$\theta_{\max} = \sum_{i=0}^{\infty} |\sigma_i \theta_i| = \sum_{i=0}^{\infty} \arctan 2^{-i} \approx 1.74329 \text{ (99.88°)}. \quad (9.41)$$

Uglovi $|\theta| > \theta_{\max}$ se ne mogu predstaviti elementarnim uglovima θ_i i CORDIC algoritam ne konvergira.

U rotacionom modu je potrebno obezbediti $|\theta| < \theta_{\max}$ kako bi algoritam konvergirao, što se može postići prerotacijom vektora pre izvršavanja CORDIC algoritma. Pod pretpostavkom da je $2\pi \geq \theta \geq 0$, prerotacija početnog vektora $[x_p \ y_p]^T$ je data sa:

$$z[0] = \begin{cases} \theta - \pi & , 3\pi/2 \geq \theta > \pi/2 \\ \theta - 3\pi/2 & , \theta > 3\pi/2 \\ \theta & , \text{inače} \end{cases}. \quad (9.42)$$

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{cases} [-x_p & -y_p]^T & , 3\pi/2 \geq \theta > \pi/2 \\ [y_p & -x_p]^T & , 2\pi \geq \theta > 3\pi/2 \\ [x_p & y_p]^T & , \text{inače} \end{cases} \quad (9.43)$$

U vektorskom modu je potrebno obezbediti da se $[x_0, y_0]^T$ nalazi u prvom ili četvrtom kvadrantu rotacijom početnog vektora $[x_p \ y_p]^T$ za π :

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{cases} [x_p & y_p]^T & , x_p \geq 0 \\ [-x_p & -y_p]^T & , x_p < 0 \end{cases} \quad (9.44)$$

$$z_0 = \begin{cases} 0 & , x_p \geq 0 \\ \pi & , x_p < 0 \end{cases} \quad (9.45)$$

CORIDC algoritam je konvergentan za $z_0 < z_{\max}$ ako se u svakoj iteraciji ugao može aproksimirati sa greškom manjom od θ_{n-1}

$$|z_i| < \theta_{n-1} + \sum_{j=i}^{n-1} \theta_j. \quad (9.46)$$

Prvo ćemo pokazati da se i -ti ugao θ_i može predstaviti preostalim uglovima sa greškom manjom od θ_{n-1}

$$\theta_i - \sum_{j=i+1}^{n-1} \theta_j < \theta_{n-1}. \quad (9.47)$$

Kako bismo pokazali da (9.47) važi, potrebno je odrediti u kojoj relaciji su θ_i i θ_{i+1} . Pošto je $\theta_i = \arctan 2^{-i}$, granična vrednost odnosa elementarnih uglova je:

$$\lim_{n \rightarrow \infty} \frac{\theta_n}{\theta_{n+1}} = \frac{1}{2} \lim_{n \rightarrow \infty} \frac{4^{n+1} + 1}{4^n + 1} = 2. \quad (9.48)$$

Izvod θ_n/θ_{n+1} je uvek pozitivan,

$$\frac{d}{dn} \frac{\theta_n}{\theta_{n+1}} > 0 \quad \text{za } n \geq 0, \quad (9.49)$$

tako da je (9.48) ujedno i maksimalni odnos θ_n/θ_{n+1} , iz čega sledi:

$$\frac{\theta_i}{\theta_{i+1}} < 2, \quad (9.50)$$

$$\theta_{i+k} > 2^{-k} \theta_i. \quad (9.51)$$

Iz sume elementarnih uglova:

$$\sum_{j=i+1}^{n-1} \theta_j = \theta_{i+1} + \theta_{i+2} + \dots + \theta_{n-1}, \quad (9.52)$$

i (9.51) možemo dobiti nejednakost:

$$\theta_{i+1} + \theta_{i+2} + \dots + \theta_{n-1} > \theta_i \left(2^{-1} + 2^{-2} + \dots + 2^{-(n-1-i)} \right). \quad (9.53)$$

Pošto je

$$2^{-1} + 2^{-2} + \dots + 2^{-(n-1-i)} = \sum_{j=1}^{n-i-1} 2^{-j} = 1 - 2^{-(n-i-1)}, \quad (9.54)$$

dalje imamo

$$\sum_{j=i+1}^{n-1} \theta_j > \theta_i \left(1 - 2^{-(n-i-1)} \right). \quad (9.55)$$

Korišćenjem (9.51) možemo dobiti nejednakost:

$$\theta_{n-1} > \theta_i 2^{-(n-i-1)}. \quad (9.56)$$

Sabiranjem levih i desnih strana (9.55) i (9.56) dobijamo:

$$\theta_{n-1} + \sum_{j=i+1}^{n-1} \theta_j > \theta_i 2^{-(n-i-1)} + \theta_i \left(1 - 2^{-(n-i-1)} \right) = \theta_i, \quad (9.57)$$

$$\theta_{n-1} > \theta_i - \sum_{j=i+1}^{n-1} \theta_j. \quad (9.58)$$

CORDIC algoritam je konvergentan ako se u svakoj iteraciji vrednost u akumulatoru ugla z_i može predstaviti preostalim elementarnim uglovima θ_i sa greškom manjom od θ_{n-1} za proizvoljno n , odnosno ako važi

$$|z_i| < \theta_{n-1} + \sum_{j=i}^{n-1} \theta_j.$$

Tvrđenje važi za z_0 pošto je $z_0 < z_{\max}$. Pretpostavimo da nejednakost važi za z_i i pokažemo da važi i za z_{i+1} . Oduzimanjem θ_i sa obe strane dobijamo

$$|z_i| - \theta_i < \theta_{n-1} + \sum_{j=i+1}^{n-1} \theta_j. \quad (9.59)$$

Dalje imamo

$$- \left[\theta_{n-1} + \sum_{j=i+1}^{n-1} \theta_j \right] < -\theta_i < |z_i| - \theta_i < \theta_{n-1} + \sum_{j=i+1}^{n-1} \theta_j. \quad (9.60)$$

Uzimanjem apsolutne vrednosti dobijamo

$$||z_i| - \theta_i| < \theta_{n-1} + \sum_{j=i+1}^{n-1} \theta_j. \quad (9.61)$$

Pošto je CORDIC iteracija za akumulator ugla z_i :

$$z_{i+1} = z_i - \sigma_i \theta_i. \quad (9.62)$$

možemo napisati:

$$|z_{i+1}| = ||z_i| - \theta_i|. \quad (9.63)$$

Zamenom (9.63) u (9.61) dobijamo

$$|z_{i+1}| < \theta_{n-1} + \sum_{j=i+1}^{n-1} \theta_j, \quad (9.64)$$

čime je pokazano da nejednakost važi za svako i . Nejednakost (9.64) pokazuje da je u svakoj iteraciji moguće aproksimirati ugao z_i sa greškom manjom od θ_{n-1} . Za $i = n - 1$ dobijamo maksimalnu grešku određivanja ugla

$$|z_{n-1}| < \theta_{n-1} < 2^{-n}. \quad (9.65)$$

Greška određivanja ugla može se proizvoljno smanjiti izborom broja iteracija n .

Za veliki broj iteracija $i \gg 1$ možemo napisati

$$\theta_i = \arctan 2^{-i} \approx 2^{-i},$$

$$\cos \theta_i \approx \cos 2^{-i} \approx 1 - 2^{-(2n+1)},$$

$$\sin \theta_i \approx \sin 2^{-i} \approx 2^{-i}.$$

Tačnost rezultata se povećava za jedan bit izvršavanjem jedne CORDIC iteracije.

♣ Broj iteracija CORDIC-a = Dužina binarne reči.

Greška zaokruživanja može biti manja od LSB dodavanjem zaštitnih bitova:

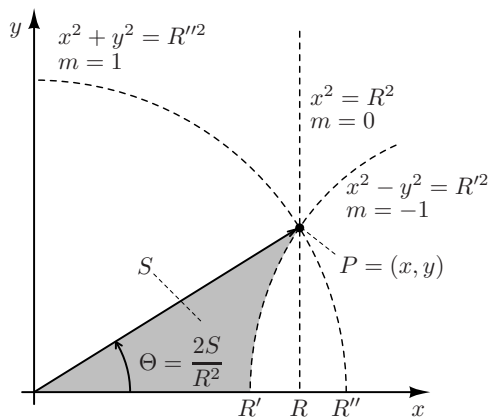
$$\clubsuit n_{\text{guard}} < \log_2 n.$$

9.7 Generalizovani CORDIC algoritam

Generalizacija CORDIC algoritma [55] se može izvršiti formulacijom u različitim koordinatnim sistemima, kao što je prikazano na slici 9.6. Ugao θ vektora $P = [x \ y]^T$ se može definisati u svim koordinatnim sistemima kao

$$\Theta = \frac{2S}{R^2}, \quad (9.66)$$

gde je S površina ograničena vektorom P , x osom i konturom $R = \sqrt{x^2 + my^2}$, m je parametar kojim se bira koordinatni sistem: $m = 1$ odgovara polarnom, $m = 0$ linearnom i $m = -1$ hiperboličkom koordinatnom sistemu.



Slika 9.6: Koordinatni sistemi u generalizovanom CORDIC algoritmu. Osenčena površina S je prikazana za slučaj hiperboličkog koordinatnog sistema $m = -1$.

Modul R i ugao Θ vektora P se mogu izraziti kao:

$$\begin{aligned} R &= \sqrt{x^2 + my^2} \\ \Theta &= m^{-\frac{1}{2}} \arctan \left(\frac{\sqrt{m}y}{x} \right). \end{aligned} \quad (9.67)$$

Iteracija generalizovanog CORDIC algoritma se može formulisati kao:

$$\begin{aligned} x_{i+1} &= x_i - \sigma_i m y_i \delta_i \\ y_{i+1} &= y_i + \sigma_i x_i \delta_i, \\ z_{i+1} &= z_i - \sigma_i \theta_i \end{aligned} \quad (9.68)$$

gde su θ_i su elementarni uglovi i δ_i je proizvoljna vrednost koja zadovoljava uslove konvergentnosti CORDIC algoritma. U (9.68) se pojavljuju proizvodi $x_i \delta_i$ i $y_i \delta_i$, tako da izbor $\delta_i = 2^{-i}$ rezultuje jednostavnom implementacijom pošto se množenja svode na binarno pomeranje u desno, i zadovoljava uslove konvergentnosti. Usvajanjem $\delta_i = 2^{-i}$ izrazi za iteraciju generalizovanog CORDIC algoritma su:

$$\begin{aligned} x_{i+1} &= x_i - m \sigma_i y_i \gg i \\ y_{i+1} &= y_i + \sigma_i x_i \gg i. \\ z_{i+1} &= z_i - \sigma_i \theta_i \end{aligned} \quad (9.69)$$

Ugao i modul transformisanog vektora su

$$\begin{aligned} \Theta_{i+1} &= \Theta_i + \sigma_i \theta_i \\ R_{i+1} &= K_i R_i, \end{aligned} \quad (9.70)$$

gde su

$$\begin{aligned} \theta_i &= m^{-\frac{1}{2}} \arctan \left(\sqrt{m} 2^{-i} \right) \\ K_i &= \sqrt{1 + m 2^{-2i}}. \end{aligned} \quad (9.71)$$

Posle n iteracija ugao i modul vektora P_n su

$$\begin{aligned}\Theta_n &= \Theta_0 + \theta = \Theta_0 + \sum_{i=0}^{n-1} \sigma_i \theta_i \\ R_n &= R_0 K(n)^{-1} = R_0 \prod_{i=0}^{n-1} K_i^{-1}\end{aligned}\quad (9.72)$$

odnosno

$$\begin{aligned}x_n &= K(n)^{-1} [x_0 \cos(\sqrt{m}\theta) - y_0 \sqrt{m} \sin(\sqrt{m}\theta)] \\ y_n &= K(n)^{-1} [y_0 \cos(\sqrt{m}\theta) + x_0 \sqrt{m} \sin(\sqrt{m}\theta)] \\ z_n &= z_0 + \theta = z_0 + \sum_{i=0}^{n-1} \theta_i\end{aligned}\quad (9.73)$$

Parametri generalizovanog CORDIC algoritma - multiplikativni faktor K_i , elementarni uglovi θ_i , brojevi iteracija koje se izvršavaju i i maksimalni ugao za koji algoritam konvergira Θ_{\max} u zavisnosti od izbora koordinatnog sistema su dati u tabeli 9.1. U hiperboličkim koordinatama iteracije $r_{i+1} = 3r_i + 1$, $r_1 = 4$, $r = 4, 13, 40, \dots$ je potrebno izvršiti dva puta da bi algoritam konvergirao.

Tabela 9.1: Parametri generalizovanog CORDIC algoritma

m	Koordinatni sistem	K_i^{-1}	θ_i	i	Θ_{\max}
1	Polarni	$\sqrt{1 + 2^{-2i}}$	$\arctan 2^{-i}$	$0, 1, 2, 3, 4, \dots$	≈ 1.74
0	Linearni	1	2^{-i}	$1, 2, 3, 4, 5, \dots$	1
-1	Hiperbolički	$\sqrt{1 - 2^{-2i}}$	$\operatorname{arctanh} 2^{-i}$	$1, 2, 3, 4, 4, \dots$	≈ 1.13

Za vrednost parametra $m = 1$ CORDIC algoritam se izvršava u polarnom koordinatnom sistemu - svodi se na slučaj razmatran u prethodnim odeljcima. Slučajevi izvršavanja u linearnom ($m = 0$) i hiperboličkom koordinatnom sistemu su razmatrani u narednim odeljcima.

9.7.1 CORDIC algoritam u linearnim koordinatama

CORDIC iteracija u linearnom koordinatnom sistemu se dobija za $m = 0$:

$$\begin{aligned}x_{i+1} &= x_i \\ y_{i+1} &= y_i + \sigma_i x_i \gg i \\ z_{i+1} &= z_i - \sigma_i 2^{-i}\end{aligned}\quad (9.74)$$

Za $m = 0$ faktor skaliranja je $K(n) = 1$, tako da nema potrebe za množačem konstantom. Prva iteracija koja se izvršava u linearnom modu je $i = 1$.

CORDIC algoritam u linearnom koordinatnom sistemu se može izvršavati u rotacionom ili vektorskom modu, u zavisnosti od izbora σ_i . Za $\sigma_i = \operatorname{sgn}(z_i)$ algoritam se izvršava

u rotacionom modu, i vrednosti posle n iteracija su:

$$\begin{aligned}x_n &= x_0 \\y_n &= y_0 + x_0 z_0. \\z_n &= 0\end{aligned}\tag{9.75}$$

Za $\sigma_i = -\text{sgn}(y_i)$ algoritam se izvršava u vektorskom modu, i vrednosti posle n iteracija su:

$$\begin{aligned}x_n &= x_0 \\y_n &= 0 \\z_n &= z_0 - \frac{y_0}{x_0}.\end{aligned}\tag{9.76}$$

9.7.2 CORDIC algoritam u hiperboličkim koordinatama

CORDIC iteracija u hiperboličkom modu se dobija za $m = -1$:

$$\begin{aligned}x_{i+1} &= x_i + \sigma_i y_i \gg i \\y_{i+1} &= y_i + \sigma_i x_i \gg i \\z_{i+1} &= z_i - \sigma_i \text{arctanh} 2^{-i}.\end{aligned}\tag{9.77}$$

Iteracije $r_{i+1} = 3r_i + 1$, $r_1 = 4$, $r = 4, 13, 40, \dots$ je potrebno izvršiti dva puta da bi algoritam konvergirao. Za $m = -1$ faktor skaliranja je

$$H(n)^{-1} = \prod_n \sqrt{1 - 2^{-2i}} \approx 0.80.\tag{9.78}$$

Pri izračunavanju multiplikativnog faktora potrebno je voditi računa o iteracijama koje se izvršavaju dva puta.

Za izbor $\sigma_i = \text{sgn}(z_i)$ CORDIC algoritam se izvršava u rotacionom modu, i vrednosti posle izvršavanja n iteracija su:

$$\begin{aligned}x_n &= H(n)^{-1} [x_0 \cosh z_0 + y_0 \sinh z_0] \\y_n &= H(n)^{-1} [y_0 \cosh z_0 + x_0 \sinh z_0]. \\z_n &= 0\end{aligned}\tag{9.79}$$

Za izbor $\sigma_i = -\text{sgn}(y_i)$ CORDIC algoritam se izvršava u vektorskom modu, i vrednosti posle izvršavanja n iteracija su:

$$\begin{aligned}x_n &= H(n)^{-1} \sqrt{x_0^2 - y_0^2} \\y_n &= 0 \\z_n &= z_0 + \text{arctanh} \frac{y_0}{x_0}.\end{aligned}\tag{9.80}$$

9.8 Pregled CORDIC operacija

Pregled izračunavanja elementarnih funkcija CORDIC algoritmom je dat u tabeli 9.2 [56].

Tabela 9.2: Izračunavanje elementarnih funkcija CORDIC algoritmom

$f(x)$	m	σ	x_0, y_0, z_0	x_n, y_n, z_n
$\cos \theta$ $\sin \theta$	1	$\text{sgn}(z_i)$	$x_0 = K(n)$ $y_0 = 0$ $z_0 = \theta$	$x_n = \cos \theta$ $y_n = \sin \theta$
$\cosh \theta$ $\sinh \theta$	-1	$\text{sgn}(z_i)$	$x_0 = H(n)$ $y_0 = 0$ $z_0 = \theta$	$x_n = \cosh \theta$ $y_n = \sinh \theta$
$\log(a)$ \sqrt{a}	-1	$-\text{sgn}(y_i)$	$x_0 = a + 1$ $y_0 = a - 1$ $z_0 = 0$	$x_n = H(n)^{-1} \sqrt{a}$ $z_n = \frac{H(n)^{-1}}{2} \log a$
$\arctan(a)$	1	$-\text{sgn}(y_i)$	$x_0 = a$ $y_0 = 1$ $z_0 = 0$	$z_n = \arctan(a)$
$\frac{b}{a}$	0	$-\text{sgn}(y_i)$	$x_0 = a$ $y_0 = b$ $z_0 = 0$	$z_n = \frac{b}{a}$
$(\rho, \theta) \rightarrow (x, y)$	1	$\text{sgn}(z_i)$	$x_0 = R$ $y_0 = 0$ $z_0 = \theta$	$x_n = K(n)^{-1} R \cos \theta$ $y_n = K(n)^{-1} R \sin \theta$
$(x, y) \rightarrow (\rho, \theta)$	1	$-\text{sgn}(y_i)$	$x_0 = x$ $y_0 = y$ $z_0 = 0$	$x_n = K(n)^{-1} \sqrt{x^2 + y^2}$ $z_n = \arctan(y/x)$

9.9 Hardverska implementacija CORDIC algoritma

CORDIC algoritam je pogodan za implementaciju u hardveru pošto se koriste samo sabirači, promena znaka, pomerači, registri i mala tabela konstanti [36]. U zavisnosti od primene i raspoloživih resursa može se napraviti iterativna ili protočna verzija algoritma.

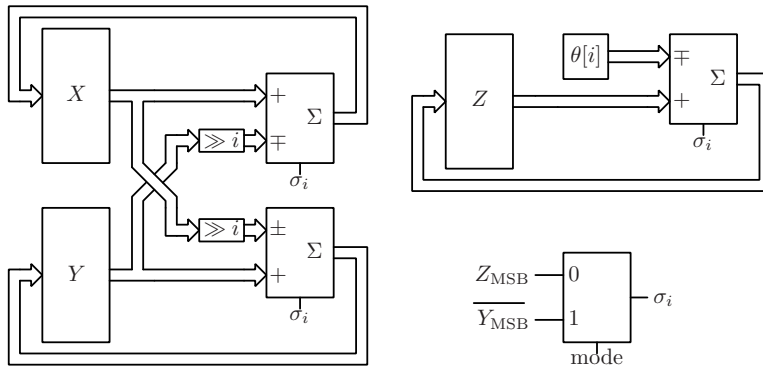
Iterativna realizacija CORDIC algoritma je prikazana na slici 9.7. Izračunavanje vrednosti se izvršava u mestu i registri X , Y i Z čuvaju trenutne vrednosti promenljivih. Pre

početka izvršavanja algoritma početne vrednosti se učitavaju u registre i posle N iteracija se rezultat nalazi u registrima.

Kontrolnim signalom mode vrši se izbor σ_i

$$\sigma_i = \begin{cases} Z_{\text{MSB}} & , \text{mode} = 0 \\ Y_{\text{MSB}} & , \text{mode} = 1 \end{cases} . \quad (9.81)$$

Pošto najviši bit (MSB) predstavlja znak u reprezentaciji komplementa dvojke, gde je broj pozitivan ako je najviši bit 0, izbor (9.81) odgovara CORDIC algoritmu u rotacionom modu za vrednost kontrolnog signala $\text{mode} = 0$ i vektorskom modu za $\text{mode} = 1$. Kontrolnim signalom σ_i se kontrolišu znakovi sabiraka u sabiračima, čime se realizuje CORDIC iteracija.

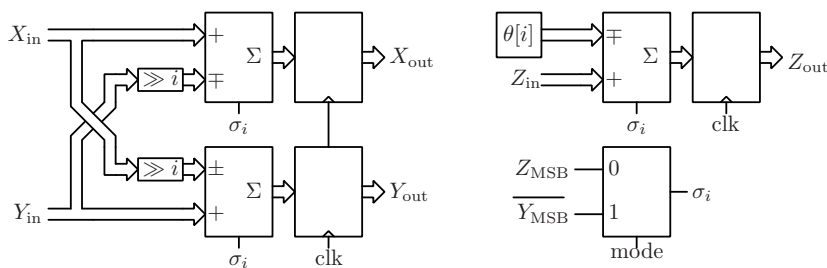


Slika 9.7: Hardverska implementacija iterativnog CORDIC algoritma.

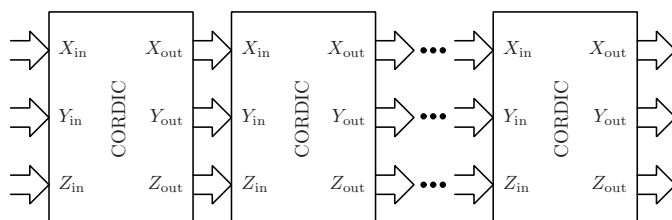
Iterativna implementacija CORDIC algoritma za N bita tačnosti se izvršava u N iteracija, i u i -toj iteraciji se vrši pomeranje za i bita u desno. Greške izračunavanja usled zaokruživanja se mogu svesti na zanemarljive ukoliko se širina reči registara, sabirača i pomerača poveća na $N + \log_2 N$.

Ukoliko se iterativnom implementacijom CORDIC algoritma ne može postići željena brzina izračunavanja, potrebno je koristiti protočnu implementaciju. U protočnoj implementaciji svaka iteracija se izvršava u posebnom stepenu, prikazanom na slici 9.8, i posle kašnjenja od N taktova rezultat se dobija u svakom taktu. Pošto se u svakom taktu dobija novi rezultat, protočna implementacija je N puta brža od iterativne, ali zauzima N puta više resursa. Pomerači u protočnoj implementaciji su fiksni i mogu se realizovati povezivanjem, tako da ne povećavaju zauzeće resursa.

Protočna implementacija CORDIC algoritma je prikazana na slici 9.9, i sastoji se iz N stepena prikazanih na slici 9.8. Pojedinačni stepeni su povezani međusobno i prosleđuju međurezultate u svakom taktu. Širina reči ne mora biti ista u svim stepenima, već se može povećavati do širine $N + \log_2 N$, kako bi se greške zaokruživanja svele na prihvatljiv nivo, odnosno kako bi se postigla tačnost izračunavanja od N bita.



Slika 9.8: Stepen protočne implementacije CORDIC algoritma.



Slika 9.9: Protočna implementacija CORDIC algoritma.

9.10 Numerički kontrolisani oscilator

Numerički kontrolisani oscilator (NCO) generiše odbirke sinusoide učestanosti ω_0 sa periodom odabiranja T . Koristi se za generisanje sinusoide proizvoljne učestanosti, npr. u sistemima za direktnu digitalnu sintezu (DDS) signala ili u kompleksnom mešaču za pomeranje spektra signala u osnovni opseg učestanosti.

Faza signala $\phi(t)$ se može izračunati integracijom trenutne učestanosti signala $\omega(t)$. Za konstantnu učestanost signala $\omega(t) = \omega_0$ trenutna vrednost faze je:

$$\phi(t) = \int_0^t \omega_0 dt = \omega_0 t. \quad (9.82)$$

U sistemu sa periodom odabiranja T diskretizovana vrednost faze je:

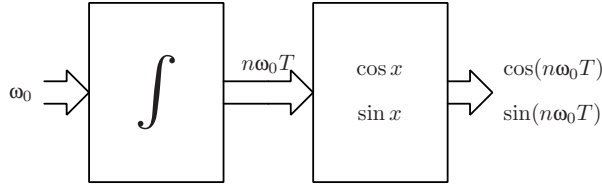
$$\phi_n = \phi(nT) = n\omega_0 T, \quad (9.83)$$

dok su odbirci izlaznog signala:

$$y_n = \cos \phi_n = \cos(n\omega_0 T). \quad (9.84)$$

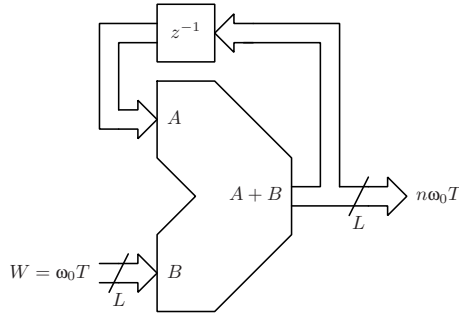
Integracija faze se može realizovati faznim akumulatorom dužine reči L bita, prikazanim na slici 9.11. Pogodno je usvojiti predstavu da maksimalna vrednost akumulatora faze 2^L predstavlja ugao od 2π , pošto je u tom slučaju:

$$\cos\left(2\pi \frac{N \bmod 2^L}{2^L}\right) = \cos\left(2\pi \frac{N}{2^L}\right), \quad (9.85)$$



Slika 9.10: Blok dijagram numerički kontrolisanog oscilatora.

odnosno sabiranje po modulu 2^L se poklapa sa periodom $\cos(x)$.



Slika 9.11: Fazni akumulator.

Priraštaj faze koji odgovara učestanosti signala ω_0 za vreme jedne periode odabiranja T je

$$\Delta\phi = \omega_0 T. \quad (9.86)$$

Pošto je $\omega_0 = 2\pi f_0$ i $T = f_s^{-1}$ imamo i

$$\Delta\phi = \omega_0 T = \frac{2\pi f_0}{f_s} \quad (9.87)$$

Ukoliko pretpostavimo da fazni akumulator dužine L bita predstavlja ugao od 2π , priraštaj faze za kontrolnu reč W je

$$\Delta\phi = W \frac{2\pi}{2^L} \quad (9.88)$$

Korišćenjem prethodnih izraza dobijamo

$$W \frac{2\pi}{2^L} = \frac{2\pi f_0}{f_s}, \quad (9.89)$$

odakle je kontrolna reč W za učestanost NCO-a f_0 :

$$W = 2^L \frac{f_0}{f_s}, \quad (9.90)$$

odnosno učestanost signala f_0 za kontrolnu reč W :

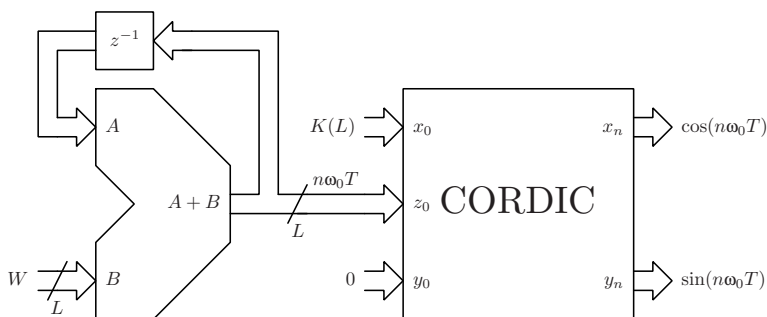
$$f_0 = \frac{W}{2^L} f_s = W f_{\min}, \quad (9.91)$$

gde je minimalna učestanost izlaznog signala

$$f_{\min} = \frac{f_s}{2^L}. \quad (9.92)$$

Minimalna učestanost izlaznog signala f_{\min} je ujedno i frekvencijska rezolucija NCO-a, pošto su učestanosti koje se mogu generisati celobrojni umnošci f_{\min} .

Numerički kontrolisani oscilator se može napraviti faznim akumulatorom dužine L bita i upotrebom CORDIC algoritma za izračunavanje odbiraka $\cos(x)$ i $\sin(x)$, kao što je prikazano na slici 9.12.



Slika 9.12: CORDIC implementacija numerički kontrolisanog oscilatora.

Primer 9.5: Frekvencijska rezolucija NCO-a

Frekvencijska rezolucija numerički kontrolisanog oscilatora za učestanost odabiranja $f_s = 100$ MHz i dužinu reči akumulatora ugla $L = 16$ bita je:

$$f_{\min} = \frac{100 \cdot 10^6}{2^{16}} \approx 1525.9 \text{ Hz.}$$

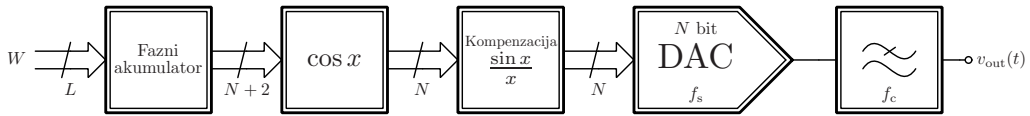
Za $L = 32$ bita frekvencijska rezolucija je

$$f_{\min} = \frac{100 \cdot 10^6}{232} \approx 23.3 \text{ mHz.}$$

Frekvencijska rezolucija od ≈ 1.5 kHz je prevelika za praktične primene. U većini praktičnih primena koristi se dužina reči faznog akumulatora od 32 ili više bita, i učestanosti odabiranja od nekoliko desetina MHz do nekoliko GHz. U prikazanom primeru NCO-a sa CORDIC algoritmom tačnost izračunavanja odbiraka $\cos(x)$ je jednakta dužini reči faznog akumulatora, što za slučaj od $L \geq 32$ bita nije praktično, a ni potrebno pošto je rezolucija digitalno-analognog konvertora u većini slučajeva manja od 16 bita. U sistemima za direktnu digitalnu sintezu signala se vrši kvantizacija faze kako bi se dužina reči faze svela na potrebnu tačnost.

Blok dijagram sistema za direktnu digitalnu sintezu (DDS) učestanosti je prikazan na slici 9.13. Sistem radi na učestanosti odabiranja f_s i sastoji se iz faznog akumulatora,

generatora odbiraka $\cos(x)$, FIR filtra za kompenzaciju $\sin(x)/x$ frekvencijske karakteristike D/A konvertora, N -to bitnog D/A konvertora i analognog rekonstrukcionog filtra granične učestanosti f_c . Rezolucija D/A konvertora od N bita određuje širinu reči korekcionog $\sin(x)/x$ FIR filtra i odbiraka $\cos(x)$. Za generisanje odbiraka $\cos(x)$ tačnosti N bita u opsegu $[0, 2\pi]$ potrebno je da širina reči ugla bude $N + 2$: N bita za generisanje odbiraka tačnosti N bita u jednom kvadrantu i još dva bita za određivanje kvadranta. Interna širina reči faznog akumulatora je L i u svim slučajevima od praktičnog značaja je $L > N + 2$, tako da se najnižih $L - N - 2$ bita odbacuju.



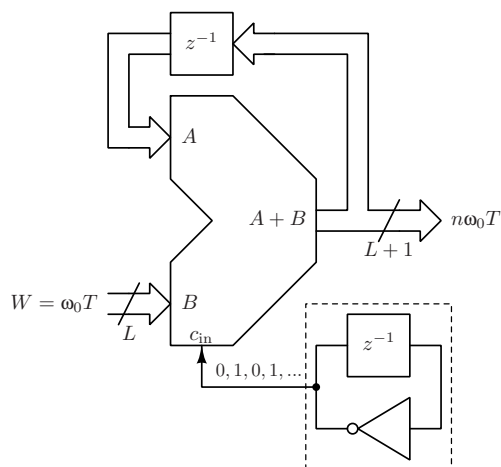
Slika 9.13: Blok dijagram sistema za direktnu digitalnu sintezu signala.

Kvantizacija faze odbacivanjem najnižih $L - N - 2$ bita za posledicu ima neželjene spektralne komponente - spurove. Teorijska analiza [62, 63] pokazuje da je amplituda najvećeg spura A_{spur} u odnosu na amplitudu pune skale A_{FS} posle skraćivanja širine reči faze sa L na N bita odbacivanjem $T = L - N$ bita za vrednost kontrolne reči W :

$$\frac{A_{\text{spur}}}{A_{\text{FS}}} = 2^{-N} \frac{\sin\left(\frac{\pi \text{NZD}(W, 2^T)}{2^T}\right)}{\frac{\pi \text{NZD}(W, 2^T)}{2^T}}, \quad (9.93)$$

gde je $\text{NZD}(a, b)$ najveći zajednički delilac a i b . Iz (9.93) se može zaključiti da spurovi usled odsecanja bita faze ne postoje ukoliko je $\text{NZD}(W, 2^T) = 2^T$, pošto je u tom slučaju $W = n2^T$ i najnižih T bita su nule, tako da odsecanje bita faze nema uticaja. Spurovi usled odsecanja bitova faze se mogu potisnuti modifikacijom faznog akumulatora prikazanom na slici 9.14 [62, 63]. Modifikacija faznog akumulatora se sastoji iz generatora povorke naizmeničnih nula i jedinica koja se dovodi na ulaz za prenos prekoračenja (carry in) c_{in} , usled čega je dužina reči faze $L + 1$ bit.

Pored spurova koji potiču od skraćivanja reči faznog akumulatora, mogu se javiti i spurovi usled greške izračunavanja $\cos(x)$ i usled kvantizacije amplitude. Greške izračunavanja se mogu svesti na proizvoljno malu vrednost usložnjavanjem kola za izračunavanje odbiraka $\cos(x)$, ili u krajnjem slučaju korišćenjem tabeliranih vrednosti. Ukoliko se koristi CORDIC algoritam, potrebno je uvesti zaštitne bite kako bi se greška izračunavanja svela na prihvatljiv nivo. Spurovi usled kvantizacije amplitude se mogu smanjiti izborom učestanosti signala i odabiranja koje zadovoljavaju uslove koherentnog odabiranja i/ili ditheringom.



Slika 9.14: Modifikovani fazni akumulator.

9.11 Kompleksni mešač

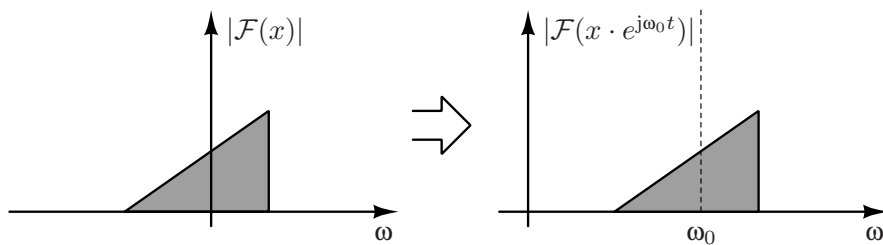
Kompleksni mešač se koristi za pomeranje spektra signala bez generisanja bočnih opsega. Kompleksni signal u osnovnom opsegu učestanosti

$$x = I + jQ \quad (9.94)$$

se može translirati u spektru množenjem sa

$$e^{j\omega_0 t} = \cos \omega_0 t + j \sin \omega_0 t \quad (9.95)$$

Translacija signala u spektru je prikazana na slici 9.15, gde se može videti da spektar transliranog signala nema simetrični bočni opseg.



Slika 9.15: Translacija spektra signala.

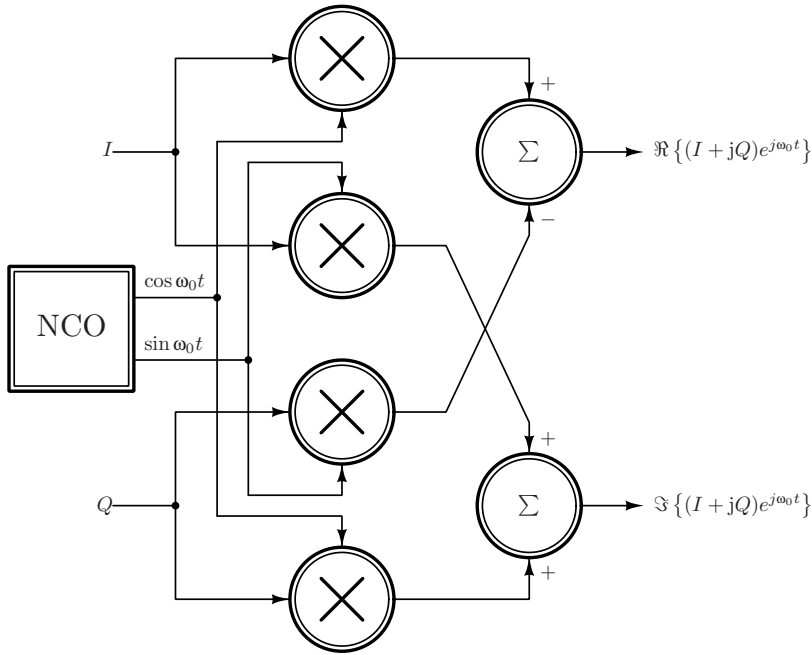
Signal transliran na centralnu učestanost ω_0 je

$$\begin{aligned} x e^{j\omega_0 t} &= (I + jQ)(\cos \omega_0 t + j \sin \omega_0 t) \\ &= I \cos \omega_0 t - Q \sin \omega_0 t + j(I \sin \omega_0 t + Q \cos \omega_0 t) \end{aligned} \quad (9.96)$$

Razdvajanjem realnog i imaginarnog dela x dobijamo

$$\begin{aligned} x e^{j\omega_0 t} &= \Re \{ x e^{j\omega_0 t} \} + j \Im \{ x e^{j\omega_0 t} \} \\ \Re \{ x e^{j\omega_0 t} \} &= I \cos \omega_0 t - Q \sin \omega_0 t \\ \Im \{ x e^{j\omega_0 t} \} &= I \sin \omega_0 t + Q \cos \omega_0 t \end{aligned} \quad (9.97)$$

Izraz (9.97) se može realizovati sa četiri množača, dva sabirača i numerički kontrolisanim oscilatorom.

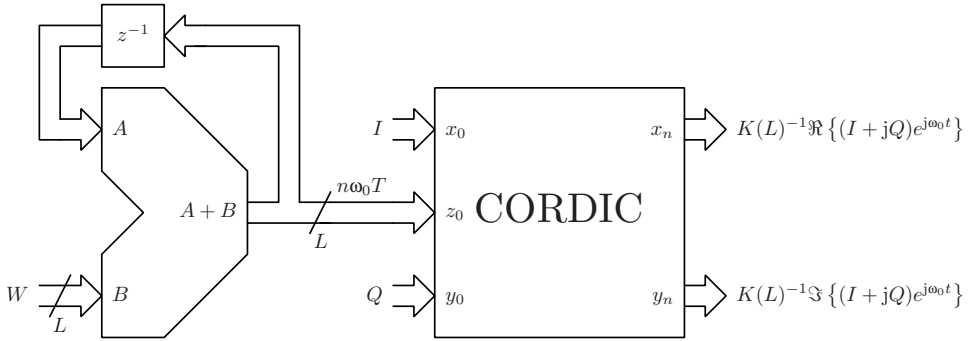


Slika 9.16: Kompleksni mešač.

Upoređivanjem (9.97) i izraza za krajnju vrednost iteracija CORDIC algoritma u rotacionom modu

$$\begin{aligned} x_n &\stackrel{n \gg 1}{\approx} K(n)^{-1} [x_0 \cos \theta - y_0 \sin \theta] \\ y_n &\stackrel{n \gg 1}{\approx} K(n)^{-1} [x_0 \sin \theta + y_0 \cos \theta] \end{aligned}$$

možemo videti da se kompleksni mešač može implementirati CORDIC algoritmom, kao što je prikazano na slici 9.17, čime se izbegava upotreba četiri množača. Dužina reči akumulatora faze se može povećati ukoliko je potrebna bolja frekvencijska rezolucija, kao što je pokazano u prethodnom odeljku. U tom slučaju je potrebno izvršiti modifikaciju akumulatora faze kako bi se potisnuli spurovi koji nastaju usled kvantizacije faze.



Slika 9.17: Implementacija kompleksnog mešača korišćenjem CORDIC algoritma.

9.12 QR dekompozicija matrice CORDIC algoritmom

QR dekompozicija matrice se koristi u MIMO-OFDM sistemima velikih brzina prenosa podataka [64, 65, 66] i postoji praktična potreba za hardverskom implementacijom. Kvadratna matrica \mathbf{A} se može dekomponovati kao [67]:

$$\mathbf{A} = \mathbf{Q}\mathbf{R}, \quad (9.98)$$

gde je \mathbf{Q} ortogonalna matrica

$$\mathbf{Q}^T \mathbf{Q} = \mathbf{Q} \mathbf{Q}^T = \mathbf{I}, \quad (9.99)$$

i $\mathbf{R} = [r_{i,j}]$ je gornje trougaona matrica

$$r_{i,j} = 0, \quad i < j. \quad (9.100)$$

Dekompozicija matrice se može izvršiti korišćenjem Givens rotacija za anuliranje elemenata ispod glavne dijagonale. Matrica Givens rotacije je [67]

$$\mathbf{G}(i, j, \theta)|_{i > j} = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & \cos(\theta) & \cdots & -\sin(\theta) & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & \sin(\theta) & \cdots & \cos(\theta) & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \quad (9.101)$$

gde je element $g_{i,j}$ matrice $\mathbf{G}(i, j, \theta)$

$$g_{k,l} = \begin{cases} 1 & , k = l \neq \{i, j\} \\ \cos \theta & , k = l = \{i, j\} \\ \sin \theta & , k = i, l = j \\ -\sin \theta & , k = j, l = i \\ 0 & , \text{inače} \end{cases} \quad (9.102)$$

Iz definicije matrice Givens rotacije (9.101) se može videti da ima iste članove kao i rotaciona matrica \mathbf{ROT} , ali je proširena sa 2×2 na $N \times N$. Matrica \mathbf{G} ima ista svojstva kao i rotaciona matrica - između ostalog i da je ortogonalna:

$$\mathbf{G}\mathbf{G}^T = \mathbf{G}^T\mathbf{G} = \mathbf{I}. \quad (9.103)$$

Proizvod Givens rotacije $\mathbf{G}(i, j, \theta)$ i matrice \mathbf{A}

$$\mathbf{B} = \mathbf{G}(i, j, \theta)\mathbf{A}, \quad (9.104)$$

ne menja elemente van redova i i j

$$b_{k,l} = a_{k,l} \Big|_{k \neq \{i,j\}}, \quad (9.105)$$

dok su elementi u redovima i i j rotirani

$$\begin{bmatrix} b_{j,1} & b_{j,2} & \cdots & b_{j,n} \\ b_{i,1} & b_{i,2} & \cdots & b_{i,n} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} a_{j,1} & a_{j,2} & \cdots & a_{j,n} \\ a_{i,1} & a_{i,2} & \cdots & a_{i,n} \end{bmatrix}. \quad (9.106)$$

Izraz za $[b_{j,l} \ b_{i,l}]^T$ (9.106) se može napisati kao:

$$\begin{bmatrix} b_{j,l} \\ b_{i,l} \end{bmatrix} = \mathbf{ROT}(\theta) \begin{bmatrix} a_{j,l} \\ a_{i,l} \end{bmatrix}. \quad (9.107)$$

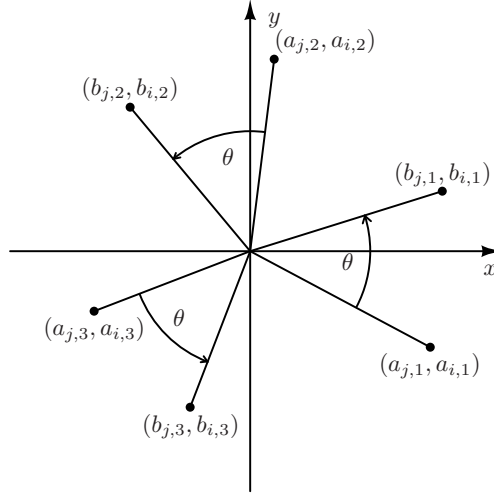
Množenjem matrice \mathbf{A} sa $\mathbf{G}(i, j, \theta)$ se elementi van redova i i j ne menjaju, dok se od elemenata redova i i j formiraju vektori $[a_{j,l} \ a_{i,l}]^T$ koji se rotiraju za ugao θ , kao što je prikazano na slici 9.18.

Proizvoljni element matrice se može anulirati Givens rotacijom

$$\begin{bmatrix} b_{j,l} \\ b_{i,l} \end{bmatrix} = \begin{bmatrix} a_{j,l} \cos \theta - a_{i,l} \sin \theta \\ a_{j,l} \sin \theta + a_{i,l} \cos \theta \end{bmatrix} \quad (9.108)$$

Izborom ugla rotacije θ

$$\theta = -\arctan \frac{a_{i,l}}{a_{j,l}} \quad (9.109)$$



Slika 9.18: Rotacija vektora Givens rotacijom.

element $b_{i,l}$ se anulira

$$\begin{bmatrix} b_{j,l} \\ b_{i,l} \end{bmatrix} = \begin{bmatrix} \sqrt{a_{j,l}^2 + a_{i,l}^2} \\ 0 \end{bmatrix}, \quad (9.110)$$

dok se ostali elementi redova i i j se rotiraju za ugao θ .

QR dekompozicija se izvršava u mestu anuliranjem elemenata dok se ne dobije gornje trougaona matrica **R** i ortogonalna matrica **Q**. Početne vrednosti matrica **Q** i **R** su:

$$\mathbf{R} = \mathbf{A} \quad \mathbf{Q}^T = \mathbf{I} \quad (9.111)$$

U svakoj iteraciji **QR** dekompozicije se izvršava Givens rotacija $\mathbf{G}(i, j, \theta)$ nad matricama **R** i **Q**:

$$\mathbf{R}_{i+1} = \mathbf{G}(i, j, \theta) \mathbf{R}_i \quad \mathbf{Q}_{i+1}^T = \mathbf{G}(i, j, \theta) \mathbf{Q}_i^T \quad (9.112)$$

koja menja redove i i j rotiranjem vektora za ugao θ . Gornje trougaona matrica se može dobiti anuliranjem elemenata ispod glavne dijagonale korišćenjem (9.109)

$$\mathbf{G}(i_1, j_1, \theta_1) \mathbf{G}(i_2, j_2, \theta_2) \cdots \mathbf{G}(i_k, j_k, \theta_k) \mathbf{A} = \mathbf{R} \quad (9.113)$$

Pošto je

$$\mathbf{A} = \mathbf{QR} \stackrel{\mathbf{Q}^T \mathbf{Q} = \mathbf{I}}{\Rightarrow} \mathbf{Q}^T \mathbf{A} = \mathbf{R} \quad (9.114)$$

ortogonalna matrica **Q** je proizvod Givens matrica koje anuliraju elemente ispod glavne dijagonale:

$$\mathbf{Q}^T = \mathbf{G}(i_1, j_1, \theta_1) \mathbf{G}(i_2, j_2, \theta_2) \cdots \mathbf{G}(i_k, j_k, \theta_k) \quad (9.115)$$

Opisani postupak dekompozicije matrice se može izvršiti CORDIC algoritmom [36]. Inicijalne vrednosti matrica su:

$$\mathbf{R}_0 = \mathbf{A} \quad \mathbf{Q}_0 = \mathbf{I}. \quad (9.116)$$

U svakoj iteraciji od elemenata redova i i j matrica \mathbf{R} i \mathbf{Q}^T se formiraju vektori

$$\begin{bmatrix} x_l \\ y_l \end{bmatrix} = \begin{bmatrix} a_{j,l} \\ a_{i,l} \end{bmatrix} \quad (9.117)$$

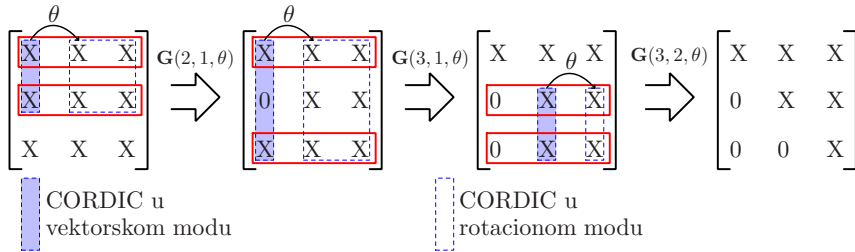
i anulira se jedan element matrice \mathbf{R} . Anuliranje elementa matrice odgovara izvršavanju CORDIC algoritma u vektorskom modu:

$$x_0 = a_{j,l}, y_0 = a_{i,l}, z_0 = 0 \quad (9.118)$$

$$x_n \approx \sqrt{a_{j,l}^2 + a_{i,l}^2}, y_n \approx 0, z_n \approx \theta \quad (9.119)$$

Ostali elementi redova i i j se rotiraju za ugao θ , koji je određen u vektorskom modu, izvršavanjem CORDIC algoritma u rotacionom modu.

Dekompozicija 3x3 matrice CORDIC algoritmom je ilustrovana na slici 9.19. U prvom koraku se nad vektorom $[a_{1,1} \ a_{2,1}]^T$ izvršava CORDIC algoritam u vektorskom modu, čime se element $a_{2,1}$ anulira i određuje ugao θ za koji se rotiraju vektori formirani od ostalih elemenata prvog i drugog reda matrice. Izvršavanje CORDIC algoritma u vektorskom modu i dva puta u rotacionom modu odgovara množenju matrice \mathbf{A} sa $\mathbf{G}(2, 1, \theta)$, odnosno $\mathbf{R}_1 = \mathbf{G}(2, 1, \theta)\mathbf{R}_0$. Isto matricom je potrebno pomnožiti i matricu $\mathbf{Q}_1 = \mathbf{G}(2, 1, \theta)\mathbf{Q}_0$. Isti postupak se može ponoviti još dva puta, kao što je prikazano na slici 9.19, čime se dobija gornje trougaona matrica \mathbf{R} i ortogonalna matrica \mathbf{Q} .



Slika 9.19: \mathbf{QR} dekompozicija matrice CORDIC algoritmom.

U toku izvršavanja jedne Givens rotacije potrebno je izvršiti CORDIC algoritam u vektorskom modu kako bi se anulirao jedan element matrice, i za ugao određen u vektorskom modu izvršiti CORDIC algoritam u rotacionom modu nad nekoliko vektora - preostalim elementima redova i i j u matrici \mathbf{R} i nad svim elementima redova i i j matrice \mathbf{Q} . CORDIC algoritam se izvršava iterativno u oba moda - vektorskom i rotacionom - i ugao se određuje i koristi za rotacije iterativno. Usled toga je moguće započeti sa rotacijama čim se odredi prva vrednost σ , odnosno nije potrebno čekati da se završi određivanje ugla u vektorskom modu. Ova optimizacija se može iskoristiti u protočnoj implementaciji CORDIC algoritma za eliminaciju ciklusa čekanja.