# DIGITAL DOWN CONVERSION IN SOFTWARE RADIO TERMINALS

*Michael Löhning, Tim Hentschel, and Gerhard Fettweis*
Dresden University of Technology
Mannesmann Mobilfunk Chair for Mobile Communications Systems
D-01062 Dresden, Germany
`e-mail: loehning@ifn.et.tu-dresden.de`

## ABSTRACT

The idea of software radio requires an expansion of digital signal processing towards the antenna. Hence, for converting the received signal to baseband, the need of efficient high speed digital down converters arises. In [1] digital down conversion was identified as one of the 'critical functionalities' because it has to run at a relative high sample rate, and has to provide high resolution. The common approach for digital down conversion (DDC) is the so called ROM table approach where the samples of the input signal are multiplied with amplitude values of the sine- and cosine-function stored in ROM. To achieve high resolution this technique requires a large look-up table which means large chip area, high power consumption, lower speed, and increased costs.

In this paper a CORDIC-based digital down converter is described. It enables to reduce the size of the look-up table considerably. Additionally to previous publications, this paper provides an overall worst case quantization error estimation that facilitates the dimensioning of the CORDIC-DDC.

## 1 INTRODUCTION

DDC is conventionally performed by multiplying the digitized input signal with amplitude values of sine- and cosine-functions stored in a ROM table which could easily be addressed by the output of an overflowing phase accumulator (Figure 1). However, for high resolution ($n$ bit) this technique requires a large look-up table ($\sim 2^n \times n$ bit) resulting in large chip area, high power consumption, lower speed, and increased costs.

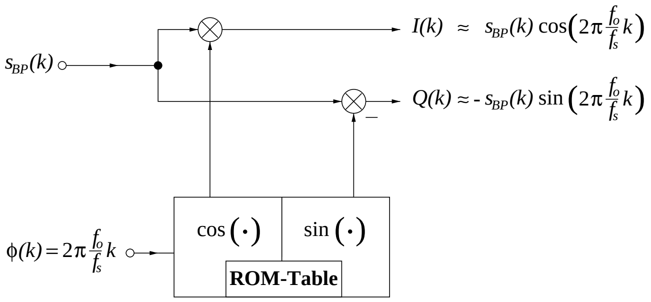

$I(k) \approx s_{BP}(k)\cos\left(2\pi \frac{f_o}{f_s} k\right)$
$Q(k) \approx - s_{BP}(k)\sin\left(2\pi \frac{f_o}{f_s} k\right)$
$s_{BP}(k)$
–
$\phi(k) = 2\pi \frac{f_o}{f_s} k$
$\cos(\cdot)$
$\sin(\cdot)$
ROM-Table

Figure 1: Conventional I/Q-DDC using the ROM table approach

An approach to overcome this drawback is the calculation of the corresponding sine- and cosine-values by means of CORDIC [2, 3, 4] with the main advantage of using only a small look up table ($\sim n \times n$ bit). The major drawback of the CORDIC approach is the increased circuit complexity. However, if used in the context of digital down conversion or frequency synchronization, the additional hardware effort is partly compensated because there is no need for explicit multipliers as will be shown in this paper.

## 2 THE CORDIC-ALGORITHM

The CORDIC (**CO**ordinate **R**otation **D**igital **C**omputer) was developed by Volder [2] in 1959 as an iterative algorithm to convert between polar and cartesian coordinates using shift, add, and subtract operations only.

In the circular rotation mode the CORDIC computes the cartesian coordinates of the target vector $\boldsymbol{v}_n = [x_n\ y_n]^T$ by rotating the input vector $\boldsymbol{v}_0 = [x_0\ y_0]^T$ by an arbitrary angle $\phi = z_0$. This vector rotation is realized by performing a sequence of successively decreasing elementary rotations with the basic rotation angles $\phi_i = \pm \arctan(2^{-i})$ for $i = 0, \ldots, n-1$. The resulting iterative process can be described by the following equations

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

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

$$z_{i+1} = z_i - d_i \arctan\left(2^{-i}\right) \tag{3}$$

where

$$d_i = \begin{cases} -1 & \text{if} \quad z_i < 0 \\ +1 & \text{otherwise} \end{cases} \quad . \tag{4}$$

specifies the direction for each elementary rotation. Eq. (3) describes an angle accumulator. After $n$ iterations the CORDIC equations result in

$$x_n \approx A_n[x_0 \cos(z_0) - y_0 \sin(z_0)] \tag{5}$$

$$y_n \approx A_n[y_0 \cos(z_0) + x_0 \sin(z_0)] \tag{6}$$

$$z_n \approx 0 \tag{7}$$

where

$$A_n = \prod_{i=0}^{n} \sqrt{1+2^{-2i}} \tag{8}$$

is the CORDIC scaling factor which depends on the total number of iterations $n$. $x_n$ and $y_n$ contain the coordinates of a scaled version of the $z_0$-rotated vector $[x_0\ y_0]^T$. The accuracy is determined by the number of iterations $n$ and the word length of the CORDIC processor (s. section 4).

It should be noted that Eqs. (1–3) converge for rotation angles between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ only. In order to increase the convergence range for all rotation angles $|z_0| \leq \pi$, Volder [2] proposes an initial iteration which rotates the input vector by $\pm\frac{\pi}{2}$:

$$x' = -d\,y \tag{9}$$

$$y' = d\,x \tag{10}$$

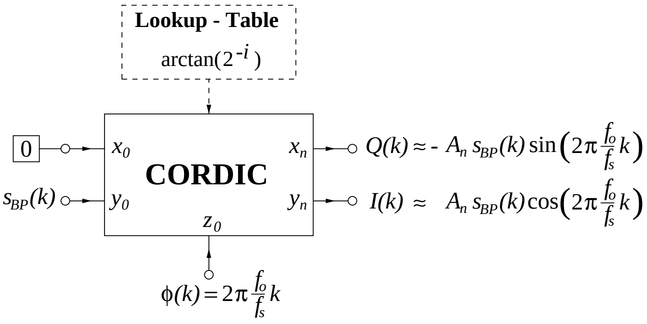
$$z' = z - d\frac{\pi}{2} \tag{11}$$

where

$$d = \begin{cases} -1 & \text{if} \quad z < 0 \\ +1 & \text{otherwise} \end{cases} \quad . \tag{12}$$

## 3 THE CORDIC-DDC

After the brief review of the CORDIC algorithm it will be shown how it can be used for digital down conversion. Substituting the output signals of the conventional DDC of Figure 1 into Eqs. (5) and (6) yields the structure of a CORDIC-based DDC. It is shown in Figure 2.



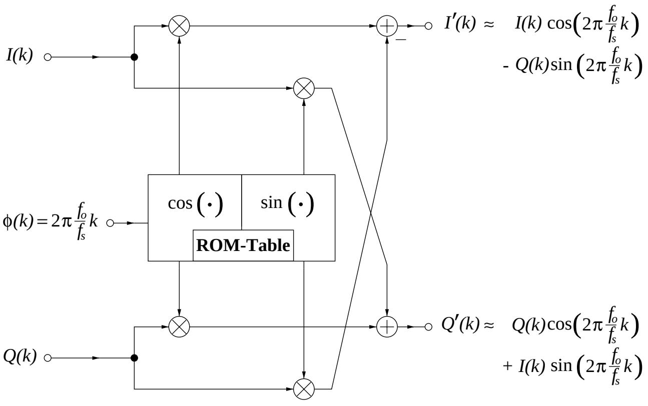
Lookup - Table
arctan(2^-i )
0
$s_{BP}(k)$
$x_0$
$y_0$
CORDIC
$x_n$
$y_n$
$z_0$
$Q(k) \approx -A_n\, s_{BP}(k) \sin\left(2\pi \frac{f_0}{f_s} k\right)$
$I(k) \approx A_n\, s_{BP}(k) \cos\left(2\pi \frac{f_0}{f_s} k\right)$
$\phi(k) = 2\pi \frac{f_0}{f_s} k$

Figure 2: CORDIC-based I/Q-DDC

At each clock interval, $y_0$ is loaded with the current sample $s_{BP}(k)$ of an IF-input signal, and $z_0$ with the current sample $\phi(k)$. The latter is supplied by an overflowing phase accumulator which generates the oscillator frequency $f_0$. $x_0$ is set to zero. After $n+1$ iterations the CORDIC provides the samples $I(k)$ and $Q(k)$ of the down-converted in-phase and quadrature-phase signal with a resolution of approximately $n$ bits. In order to achieve this, only a very small look-up table is needed. It contains the $(n+2)$ basic rotation angles.

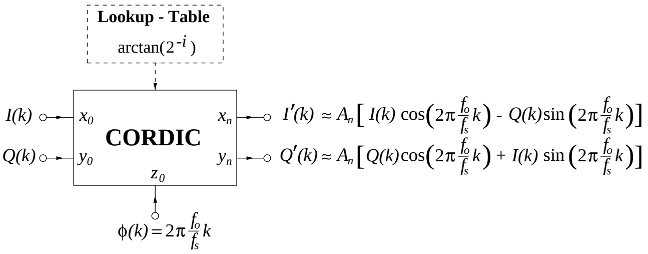
Still, the main problem of the CORDIC is that $n+1$ iterations have to be performed for each signal sample, requiring an internal clock rate being $(n+1)$-times higher than the sample rate of the signal. However, the CORDIC can be implemented by a pipelined architecture [5, 6]. Thus, the CORDIC-DDC becomes suitable for high speed applications. An additional advantage of such an implementation is that there is no need for a look-up table anymore, since the invariant elementary rotation angles can be hard-wired to each stage.

Further it should be noted that the hardware effort of the CORDIC is approximately that of three multipliers with the respective word length. This means that the CORDIC-DDC has only one and a half of the hardware complexity of the common DDC (two explicit multipliers), while saving a large amount of chip area because no ROM table is needed. In this context it is worth mentioning that without any additional hardware effort the CORDIC can easily be used as a complex image rejection mixer (s. Figure 4) e.g., for frequency synchronization purposes. Comparing with the conventional image rejection mixer of Figure 3 reveals that the CORDIC even replaces four multipliers and the look-up table.



$I(k)$
$\phi(k) = 2\pi \frac{f_0}{f_s} k$
$Q(k)$
cos(·)
sin(·)
ROM-Table
–
$I'(k) \approx I(k) \cos\left(2\pi \frac{f_0}{f_s} k\right) - Q(k) \sin\left(2\pi \frac{f_0}{f_s} k\right)$
$Q'(k) \approx Q(k) \cos\left(2\pi \frac{f_0}{f_s} k\right) + I(k) \sin\left(2\pi \frac{f_0}{f_s} k\right)$

Figure 3: Conventional image rejection mixer



Lookup - Table
arctan(2^-i )
$I(k)$
$Q(k)$
$x_0$
$y_0$
CORDIC
$x_n$
$y_n$
$z_0$
$I'(k) \approx A_n\left[I(k) \cos\left(2\pi \frac{f_0}{f_s} k\right) - Q(k) \sin\left(2\pi \frac{f_0}{f_s} k\right)\right]$
$Q'(k) \approx A_n\left[Q(k) \cos\left(2\pi \frac{f_0}{f_s} k\right) + I(k) \sin\left(2\pi \frac{f_0}{f_s} k\right)\right]$
$\phi(k) = 2\pi \frac{f_0}{f_s} k$

Figure 4: CORDIC-based image rejection mixer

## 4 QUANTIZATION ERROR BOUND

In order to dimension the CORDIC-DDC it is necessary to estimate the achievable accuracy. For that purpose we derive a worst case error bound of the overall quantization error of the CORDIC algorithm. With respect to the high-speed application as a digital down-converter, only fixed point implementations are considered. An approach to this problem including CORDIC processors with floating point arithmetic has been presented in [7], still, without considering the quantization error produced in the CORDIC angle accumulator (Eq. (3)).

After an initial iteration (Eqs. (9)-(12)), and $n$ further iterations (Eqs. (1)-(4)), the maximum overall quantization error of a fixed point CORDIC processor can be calculated as follows:

$$|\Delta \boldsymbol{v}_n| = |\boldsymbol{v}_n - \boldsymbol{v}^*| \ \leq\ |\boldsymbol{v}^*| \cdot \Big(|\delta_n| + |\theta_n|\Big) + |\boldsymbol{f}_n| \qquad (13)$$

where $\boldsymbol{v}^* = [x^*\ y^*]^T$ is the ideal output vector (theoretically produced by a CORDIC processor with infinite precision arithmetic after an infinite number of iterations), and $\boldsymbol{v}_n = [x_n\ y_n]^T$ is the output vector computed by the finite precision CORDIC. $\delta_n$ stands for the so-called *(angle) approximation error* and $\boldsymbol{f}_n$ for the *rounding error vector*, both derived in [7]:

$$|\delta_n| \leq 2^{-(n-1)} \qquad \text{for} \quad n \geq 1 \qquad (14)$$

$$|\boldsymbol{f}_n| \leq \sqrt{2}\left[\epsilon_0 \prod_{i=0}^{n-1} k(i) + \epsilon \left(1 + \sum_{j=1}^{n-1} \prod_{i=j}^{n-1} k(i)\right)\right] \qquad (15)$$

with

$$k(i) = \sqrt{1 + 2^{-2i}} \quad . \qquad (16)$$

Thereby, $\epsilon_0$ considers the maximum absolute input quantization error in $x_0$ and $y_0$, while $\epsilon = 2^{-b}$ stands for the maximum absolute rounding error in $x_i$ and $y_i$ due to bit truncation with fixed point arithmetic ($b$ fractional binary digits) which is done in each iteration.

As an extension to the results in [7], $\theta_n$ describes the resulting error component in $\boldsymbol{v}_n$ due to the rounding errors in $z_i$ that are caused by the input quantization of $z_0$ and the quantized fixed point representation of the basic rotation angles $\frac{\pi}{2}$ and $\arctan(2^{-i})$, respectively:

$$|\theta_n| \leq \begin{cases} \sigma_0 + n\,\sigma & \text{for} \quad 0 \leq n \leq b+1 \\ \sigma_0 + (b+2)\,\sigma & \text{for} \quad n > b+1 \end{cases} \qquad (17)$$

$\sigma_0$ represents the maximum absolute phase input quantization error, and $\sigma$ the maximum absolute rounding error of the basic rotation angles.

## 5 SIMULATION RESULTS

In this section some simulation results are presented in order to verify the quality of the derived maximum quantization error bound. Therefore, Eqs. (13)-(17) are evaluated for different values for $n$ and $b$. For simplification the input quantization error $\epsilon_0$ is set to zero, which models an ideal analog-to-digital converter (ADC). The input signal is set to constant *one*. With the so computed worst case error estimates the minimum SNR (signal to [quantization] noise ratio) can be estimated.

In Figure 5 these results are compared with the minimum SNR values obtained by means of Monte Carlo simulations. The latter have been made with a parameterizable model of the CORDIC-DDC considering all quantization effects. $10^5$ vector rotations with randomly generated rotation angles $\phi(k)$ (uniformly distributed in the interval $[-\pi, \pi]$) were performed in MATLAB for each set of parameters $(m, b)$. The worst result for each set $(m, b)$ has been taken as the "simulated worst case error" (in contrast to the analytically predicted worst case error of Eq. (13)). As the curves in Figure 5 suggest, Eq. (13) tends to overestimate the worst case quantization error. This means that the analytically predicted minimum SNR is lower than the simulated one. Nevertheless, although a bit pessimistic, the theoretically derived error bound can be used for dimensioning a CORDIC-DDC. For a given minimum SNR and a given number of iterations, the difference between the theoretically calculated minimum number of fractionally binary digits $b$ and that found by simulation is only about one.

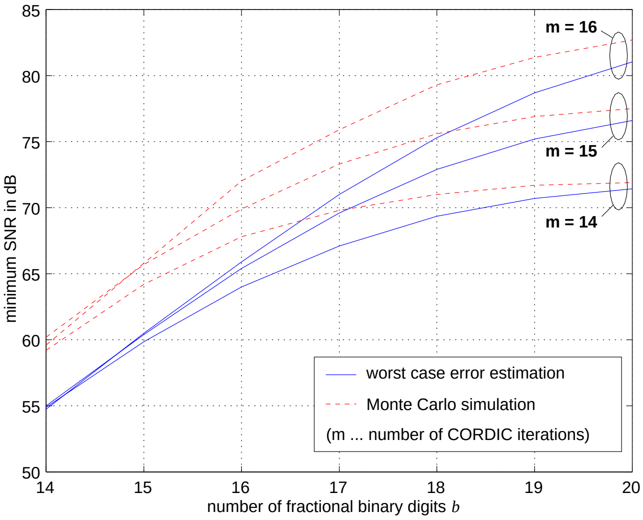


Figure 5: Minimum SNR predicted and simulated for a CORDIC-DDC with $m = n + 1$ iterations, different numbers of fractional binary digits $b$, and **ideal** ADC

Beside the overall quantization error power its spectral distribution is an important property, i.e., the spectral purity. A worst case assumption would be to concentrate the error in one discete spur. This would give a bound for the spurious-free dynamic range (SFDR) i.e., the ratio of the power of the desired signal, and the power of the strongest spur.

Still, simulations have shown that the error is distributed over frequency. Due to this distribution the strongest resulting spur is significantly lower than the overall error (worst case error bound). Figure 6 shows the result of downconverting a 20 MHz tone to DC (sample rate $f_S$ = 65 MSps). The SFDR is approximately 98 dB in contrast to the worst case estimation of 75 dB (analytically predicted) or 79 dB (simulated) in Figure 5.

The spurs in Figure 6 are a result of cross-mixing the spurs of the CORDIC with the input signal. If the input signal itself contains spurs, even more spurs will appear at the output of the CORDIC-DDC. In this case the spurs of the input signal are mixed with the desired CORDIC frequency, and with all the spurs of the CORDIC. Quantizing the input signal yields such spurs. If their power is larger than the power of the spurs of the CORDIC the SFDR is mainly determined by the quantization of the input signal rather than the quantization effects within the CORDIC-DDC. This is shown in Figure 7 where a 12 bit input signal is fed to the CORDIC. The SFDR is decreased to approximately 70 dB compared to Figure 6.

It should be noted that quantized harmonic signals produce an error spectrum which is much worse conditioned that that of typical signals e.g., in communication applications.

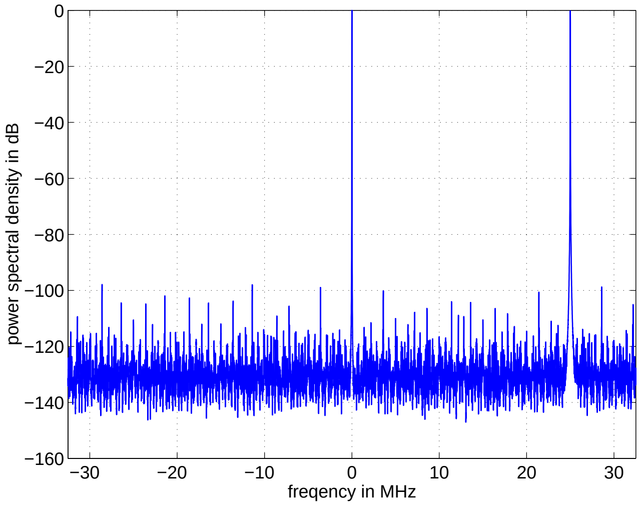


Figure 6: Power spectral density of the complex DDC output signal $I(k) + jQ(k)$ after down conversion of a 20 MHz harmonic tone using a CORDIC-DDC with 16 iterations, 18 fractional binary digits, and **ideal** ADC

## 6 CONCLUSIONS

In this paper a CORDIC-based method for digital down conversion was proposed which overcomes the drawback of the common DDC of requiring a very large ROM table to achieve high resolution. Therefore, it helps to save chip area, power consumption, and costs. Since this CORDIC-DDC can easily be implemented on a pipelined architecture, it is suitable for high speed applications as required for the task of digital down-conversion in software radio terminals. The CORDIC-DDC can realize any oscillator frequency by simply feeding it with the appropriate saw-tooth input signal. No coefficients must be changed. Thus, it empowers the software radio concept.

An analytically derived worst case quantization error bound was presented which allows an efficient design of CORDIC-based DDC.

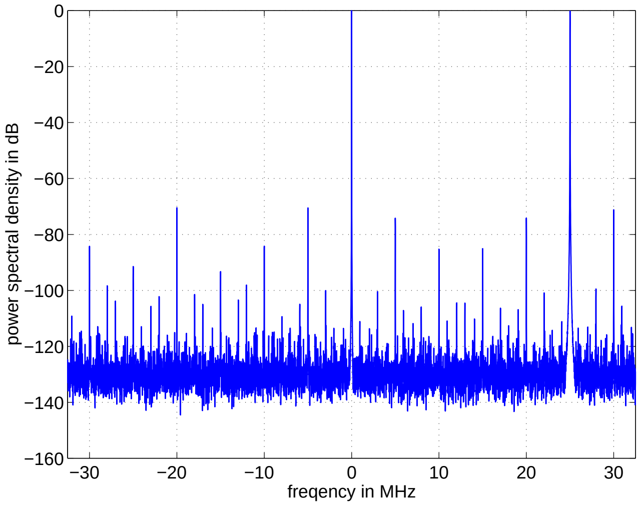


Figure 7: Power spectral density of the complex DDC output signal $I(k) + j\,Q(k)$ after down conversion of a 20 MHz harmonic tone using a CORDIC-DDC with 16 iterations, 18 fractional binary digits, and **12 bit** ADC

## References

[1] T. Hentschel and G. Fettweis, “Software radio receivers,” in *CDMA Techniques for Third Generation Mobile Systems* (F. Swarts, P. van Rooyen, I. Opperman, and M. P. Lötter, eds.), vol. 487 of *The Kluwer International Series in Engineering and Computer Science*, ch. 10, pp. 257–283, Kluwer Academic Publishers, 1999.

[2] J. E. Volder, “The CORDIC trigonometric computing technique,” *IRE Transactions on Electronic Computers*, vol. EC–8, pp. 330–334, Sept. 1959.

[3] J. S. Walther, “A unified algorithm for elementary functions,” in *Proceedings of the Joint Spring Computer Conference*, pp. 379–385, 1971.

[4] J. Vankka, “Methods of mapping from phase to sine amplitude in direct digital synthesis,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 44, pp. 526–534, Mar. 1997.

[5] R. Andraka, “A survey of CORDIC algorithms for FPGA based computers,” in *Proceedings of the 1998 ACM/SIGDA 6th international symposium on Field programmable gate arrays*, (Monterey, CA), pp. 191–200, Feb. 1998.

[6] S. Wang, V. Piuri, and E. E. Schwartzlander, Jr., “Granularly-pipelined CORDIC processors for sine and cosine generators,” in *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing*, (Atlanta, GA), pp. 3299–3302, May 1996.

[7] Y. H. Hu, “The quantization effects of the CORDIC algorithm,” *IEEE Transactions on Signal Processing*, vol. 40, pp. 834–844, Apr. 1992.