Implementation of a European Paging System Receiver Using CORDIC Algorithm

Jarkko Vuori *Jorma Skyttä*

Laboratory of Signal Processing and Computer Technology
Helsinki University of Technology
Otakaari 5A
FIN-02150 ESPOO, Finland

tel: +358-0-451 2452; fax: +358-0-460 224
email: Jarkko.Vuori@hut.fi

ABSTRACT

ERMES is a new European paging standard. The data transmission speed is higher than in older systems, e. g. POCSAG, and advanced new features are implemented including intelligent battery saving operation and country roaming. The higher speed is achieved using the more elaborate modulation method 4-PAM/FM which makes the demodulator implementation much harder than in older 2-FSK based paging systems.

The objective of this paper is to propose a novel ERMES signal demodulator structure utilising a complex digital phase-locked loop which is implemented using the CORDIC algorithm. Phase-locked loop demodulators have inherently better performance than the normally used discriminator type detectors. Implementation of those phase-locked loop structures using the CORDIC algorithm makes VLSI realizations very feasible.

1. INTRODUCTION

In this paper we consider non signal processor implementation of ERMES paging network receiver. ERMES (European Radio MEssage System) [1] is a pan-European paging system which is defined by ETSI/PS (European Telecommunications Standards Institute/Paging Systems) and is operating in the 169 MHz band using 16 RF (Radio Frequency) channels each being 25 kHz wide. Since the service area is usually quite extended, the coverage is achieved by operating several radio transmitters on the same RF channel simultaneously (*simulcast* environment).

Due to the physical limitations pagers antennas are inefficient. This means that pagers used in this network must have quite good sensitivity, 25 dBμV/m [2]. The power consumption of the pagers should also be very low in order to make battery lifetime long, about one year. This imposes tough limits to the receiver RF-front end and also to the digital VLSI-implementation. The performance of the digital demodulator should be good and the number of gates used must be low.

In the first part of this paper, we describe the ERMES signal format and the RF-part of the receiver. Then we discuss the FSK-demodulation in general and present a phase-locked loop detector for FSK-demodulation. After that a novel CORDIC (COordinate Rotation DIgital Computer) [5] realisation of a complex phase-locked loop is introduced. The final results show the obtained bit-error rates.

2. ERMES SIGNAL FORMAT

The RF-signal modulation format in ERMES is 4-PAM/FM with symbol rate of 3125 symbols/s and the data transmission rate is 6.25 kbit/s [1]. This modulation method allows an acceptable spectral efficiency to be obtained without using RF power amplifiers near their saturation point, being amplitude modulation free. The noncoherent detection in pagers allows simple implementation due to the absence of any carrier recovery circuitry.

A modulation index (defined as the ratio of the steady-state frequency deviation corresponding to the highest modulation level at the level generator output to the bit rate of the input signal [3]) equal to 0.75 was chosen as a trade off between spectral efficiency and performance in simulcast environment.

The signal to the FM-modulator is fed via a premodulation shaping filter in order to make a performance trade-off among different working conditions. The premodulation filter is a 10th-order Bessel filter with 3 dB bandwidth of 3.9 kHz.

The signal to be transmitted can be described as

$$s(t,\vec{a}) = A\cos\left(\omega_c t + \phi(t,\vec{a}) + \phi_0\right), \qquad (1)$$

where A is the amplitude of the signal depending on the energy of the signal during one symbol time T, ω_c is the carrier frequency, $\vec{a} = (a_1, a_2, \ldots, a_n)$ is an *uncorrelated symbol sequence* ($a_i = \pm 1, \pm 3$) and ϕ_0 is the phase of the carrier at the beginning of the symbol. In *non-coherent receiving*, the ϕ_0 is assumed to be a uniformly distributed

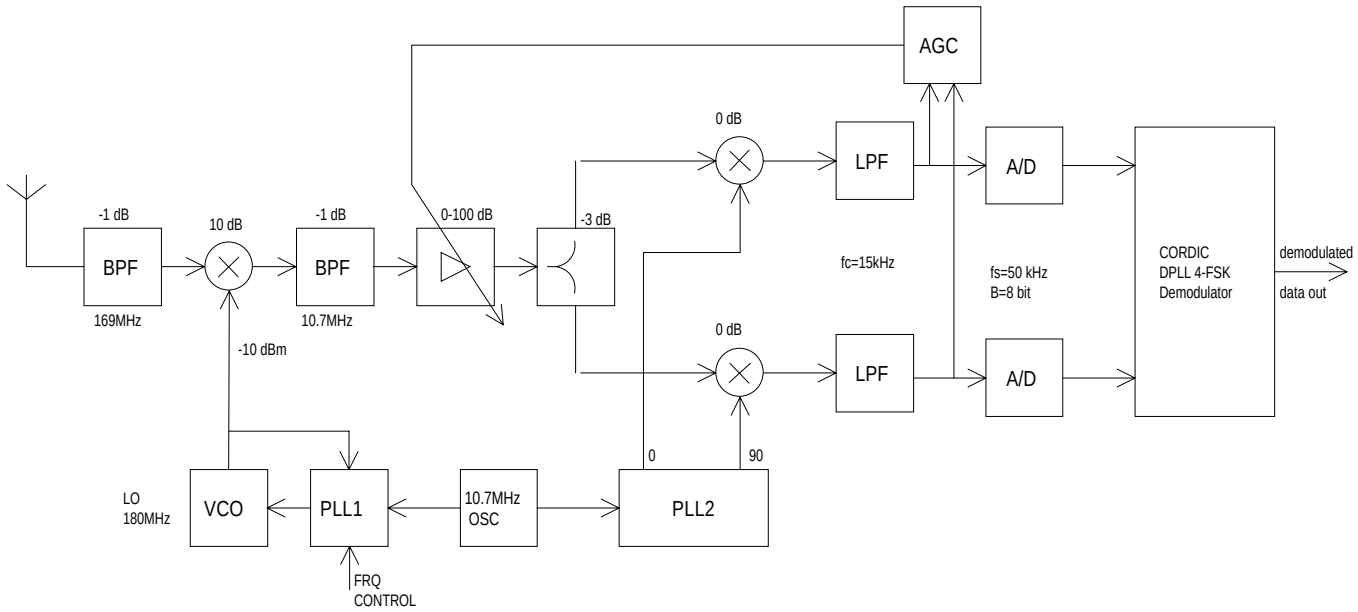

Figure 1: ERMES receiver RF-front end

random variable in range $[-\pi, \pi]$. The information bearing phase can be stated as in general FSK-case [7]

$$\phi(t, \vec{a}) = \pi a_i h \frac{t-(i-1)}{T} + \pi \sum_{r=1}^{i-1} a_r h, \qquad (2)$$

where $(i-1)T \le t \le iT$ and h is the modulation index.

3. STRUCTURE OF THE RECEIVER

FM-modulated signal can be demodulated with simple a discriminator type FM-receiver, and from this detected signal the decision can be made using only amplitude levels. This is the well-known *limiter-discriminator* detection (LD) of frequency modulated signals. *Digital phase locked loop* type detector (PLD) has some interesting benefits compared to the LD. PLD doesn't have a threshold effect and there is no need for the *automatic frequency control* (AFC) circuit [3]. In figure 1 is shown the RF front end block diagram of the PLD type ERMES receiver.

The radio front-end has an adjustable receiving frequency, so that the pager can automatically tune to the desired ERMES channel. This is done by a computer controlled single-loop synthesized local oscillator (LO). The first mixer is a typical modern, doubly balanced Gilbert-Cell type active mixer with third-point IP at -5 dBm. Amplified, downconverted and filtered intermediate frequency (IF) of 10.7 MHz is amplified more by the adjustable gain control block. This amplified IF-signal is splitted and down-converted to the baseband. Downconversion is done in quadrature in order to reject the undesired image frequency. Quadrature local oscillator is made by a crystal oscillator and phase lock-loop to obtain accurate 90° phase difference over wide frequency band. Both the LO synthesizer and baseband converter PLL has the same reference frequency in order to minimize undesired beat frequencies. Baseband signal is antialias filtered and converted to the digital form with a dual A/D converter. Automatic gain controller (AGC) block maintains constant drive level to the A/D converters in order to maximally utilise the dynamic ratio of the A/D converters.

Using modern standard IC-technology the RF-front end can be implemented with only three chips including the A/D-converters and LO synthesizer.

3.1 Phase-locked loop demodulator

We have implemented an FSK demodulator using digital phase-locked loop (DPLL) which is frequency locked to the received signal. It is quite easy from this frequency indication to make the decision of the transmitted symbol. Downconverted and filtered complex baseband signal, the received signal **s***(t)*, has two components: *real* and *imaginary* parts, and therefore can hold **both** the amplitude and phase of the sinusoidal signal at the same time. On the other hand, the real signal can have only amplitude component (we cannot say to which direction the signal is going by observing only one point of the sinusoidal signal). Using complex signals in phase-locked loops is useful, because in this case there are no image frequencies at the output of the phase comparator (mixer). This means that there is no need for the loop filter to reject this image frequency, and thus the loop filter can be of lower order. This makes the implementation easier and there is no group delay caused by the loop filter.

The complete time domain equation for the complex first-order DPLL demodulator is

$$\begin{aligned} \varepsilon_n &= \Re\{\mathbf{s}(n)\}\cos(t_n) - \Im\{\mathbf{s}(n)\}\sin(t_n) \\ c_n &= 2\pi K \varepsilon_n \\ t_{n+1} &= (t_n + c_n + \omega_c) \bmod 2\pi \end{aligned}, \qquad (3)$$

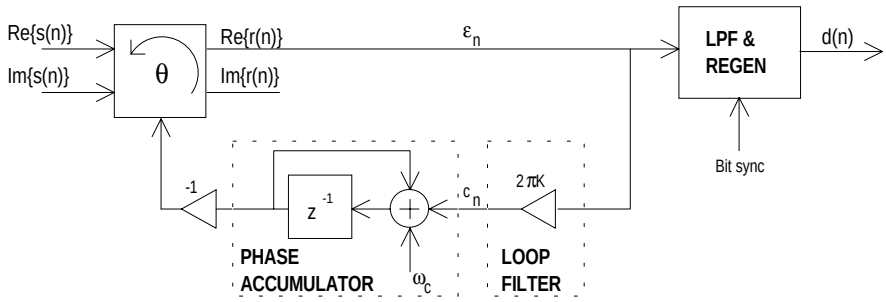

Figure 3: Complex DPLL using CORDIC algorithm

where $\omega_c = 2\pi f_c$ is the center frequency, K is the loop filter coefficient, $\Re\{\cdot\}$ and $\Im\{\cdot\}$ are real and imaginary part of the given signal. The first line of equation (3) is the phase comparator equation, second line is the loop filter equation and the final line is the phase accumulator equation.

3.2 Complex DPLL CORDIC Implementation

Implementing the first order complex DPLL requires adjustable local sinewave generation and phase detection. Direct sine and cosine term calculation using polynomial approximation, e.g. with Taylor series, needs quite a large amount of hardware. Table look-up using interpolation also needs a large amount of gates for interpolation and ROM memory.

By inspecting the complex DPLL equation we note that computationally the most difficult part of the equation can be stated as a *coordinate rotation operator*

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} \angle\theta: \quad \begin{cases} x_{i+1} = x_i \cos(\theta) - y_i \sin(\theta) \\ y_{i+1} = y_i \cos(\theta) + x_i \sin(\theta) \end{cases}, \quad x_i \in \Re, y_i \in \Re, \theta \in [0, 2\pi] \cdot \quad (4)$$

Using the rotator operator, the DPLL equation can be stated as

$$\begin{aligned} \varepsilon_n &= \Re\{\mathbf{s}(n) \angle -\theta_n\} \\ \theta_{n+1} &= (\theta_n + 2\pi K \varepsilon_n + \omega_c) \bmod 2\pi \end{aligned} \quad . \quad (5)$$

This rotator DPLL tries to adjust the phase rotation in such a way that the imaginary component from the rotator will be always zero. By setting $\Re\{s(n)\} = x_n$ and $\Im\{s(n)\} = y_n$ we can draw the block diagram of the DPLL using a rotation operator, figure 3.

One well-known way to calculate rotations and vector length functions is the CORDIC algorithm [5]. CORDIC algorithm performs vector coordinate rotations by using iterative binary shift and conditional add/subtract operations which are very easy to implement in hardware. When we limit the rotation angle to one of the followings, $\theta_i = \tan^{-1}(2^{-i}), \quad i = 0,1,2,\ldots$, the rotation can be made by using only binary shifts and additions/subractions. In this case the rotation equation can be stated as

$$\begin{aligned} x_{i+1} &= k_i \left(x_i \mp y_i 2^{-i} \right) \\ y_{i+1} &= k_i \left(y_i \pm x_i 2^{-i} \right) \end{aligned}, \quad (6)$$

where the scaling constant is given by $k_i = \sqrt{1 + 2^{-2i}}$. Note that the rotation is not a pure rotation but a rotation-extension. The direction of rotation is determined by the addition/subraction selection.

In order to rotate other angles than those special rotation angles θ_i, successive iterative clockwise or anticlockwise rotations depending on the remaining angle must be made. In this case the total rotation angle $\hat{\theta}$ is

$$\begin{aligned} x_1 &= -\mathrm{sgn}(\theta_0) \times y_0 \\ y_1 &= \mathrm{sgn}(\theta_0) \times x_0 \\ \theta_1 &= \theta_0 - \mathrm{sgn}(\theta_0) \times \pi/2 \\ &\text{for i = 1...(N-1)} \\ \xi_i &= \mathrm{sgn}(\theta_i) \\ \theta_{i+1} &= \theta_i - \xi_i \times \arctan(2^{-(i-1)}) \\ x_{i+1} &= x_i - \xi_i \times y_i \times 2^{-(i-1)} \\ y_{i+1} &= y_i + \xi_i \times x_i \times 2^{-(1-1)} \\ &\text{end} \end{aligned}$$

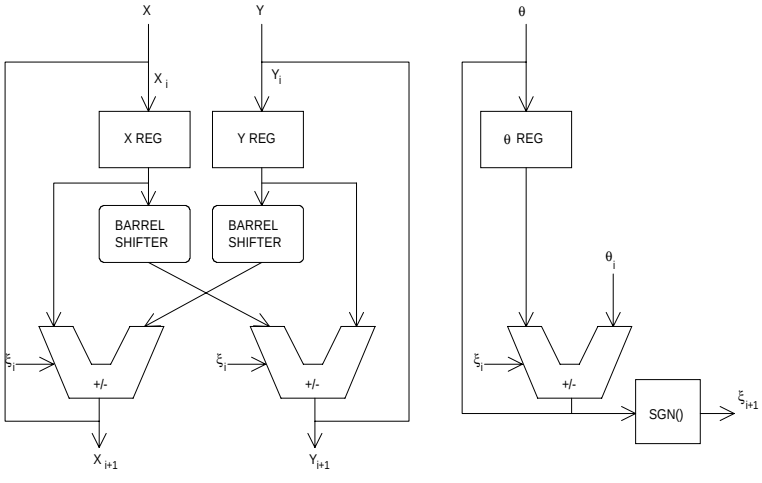

Figure 2: CORDIC algorithm and one hardware implementation

given by

$$\hat{\theta} = \sum_{i=0}^{N-1} \mathrm{sgn}(\xi_i) \times \theta_i , \quad (7)$$

where $\xi_i = \theta - \sum_{k=0}^{i-1} \mathrm{sgn}(\xi_k) \times \theta_k$ is the angle approximation error and θ is the desired rotation angle. The remaining angle ξ_i goes smaller on every interation (if N is large $\Rightarrow \xi_i \approx 0$), thus the total rotation angle converges to the desired angle θ. The total scaling factor is then

$$k = \prod_{i=0}^{N-1} \sqrt{1 + 2^{-2i}} . \quad (8)$$

The total number of iterations *N* is determined by the accuracy desired. CORDIC algorithm is a bit-recursive algorithm—each iteration increases the accuracy of the results approximately by one bit. Also the rounding error due to the finite precision in fixed point arithmetic causes approximation errors in the CORDIC algorithm. Both scaling error and rounding errors must be taken account when selecting the optimal number of iterations. The scaling is better done after the interations in fixed precision arithmetic, otherwise the scaling errors accumulate. The CORDIC algorithm and its hardware implementation block diagram are shown in figure 2.

The hardware implementation of the CORDIC algorithm requires adders, shifters, multiplexers and registers. The most speed-critical component will be the adder due to the carry propagation logic. The CORDIC algorithm could be accelerated by introducing a redundant number representation for the internal computation and therefore eliminating carry propagation from each addition/subtraction, but the required chip area increases [6].

The hardware implementation of the CORDIC algorithm is relatively slow because each iteration requires an addition. A solution is to use a redundant number representation [6] in order to get rid of carry propagation in addition operations.

By noting that this loop filter coefficient *K* depends on the sampling frequency and the number of iterations on the CORDIC algorithm, it is often possible to arrange the loop filter multiplication to be shift operations only.

The post detection filter after the DPLL rejects the unwanted noise from the signal before the decision making process. The post detection filter can be realised as a Boxcar filter [7], which contains only additions and therefore makes the implementation very easy.

3.3 Performance

Figure 4 shows the simulated bit error rate of the proposed ERMES demodulator structure. The test signal consists of uniformly distributed pseudo-random bit patterns being 4-FSK modulated with ERMES standard filters and mixed with additive gaussian noise. The measured bit error rate (crossmarks) corresponds closely to the theoretical upper

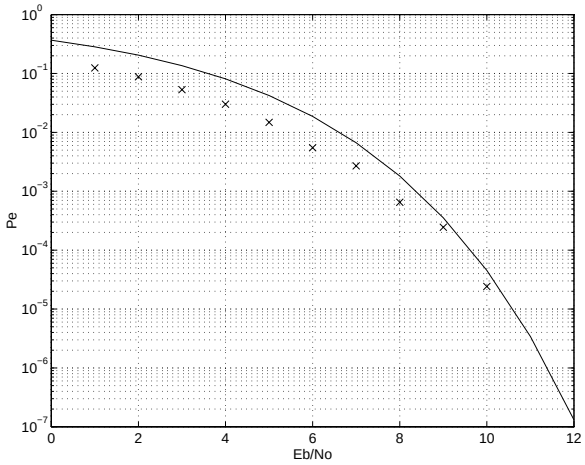

Figure 4: Simulated performance for the CORDIC DPLL ERMES demodulator

limit of the error rate (solid line) for a non-coherent demodulator, thus showing the usefulness of the proposed algorithm.

4. CONCLUSIONS

Normally the ERMES signal is demodulated with an analog discriminator type receiver. The ERMES signal has the annoying feature that idle batches contain a bit pattern which has strong frequency components near DC-level. This may distort the reception of the pages, especially in weak reception conditions.

In this paper we have presented the concept of directly demodulating the downconverted baseband signal with a DPLL where also DC-components of the signal can be handled. When the DPLL is implemented with complex phase comparator the loop filter is extremely simple and does not distort the received waveform. The digital VLSI implementation of this kind of complex DPLL is very simple due to the use of CORDIC computational unit.

REFERENCES

[1] "Air interface specification", European Telecommunications Standards Institute (ETSI), prETS 300 133-4, 1991

[2] "Receiver requirements", European Telecommunications Standards Institute (ETSI), prTBR 7, DTBR/PS-02003, 1993

[3] Iwanami, Y., "A DSP DPLL Demodulator with Sequence Estimation for MSK Signal in the Presence of Doppler Shift on Rician Fading Channel", PIMR'94, The Hague, The Netherlands, August 1994, p. 332–336

[4] Muratore, F., Palestini, V., Pattini, F., "4-PAM/FM with Noncoherent Detection for a Pan-European Paging System", IEEE Trans. on Communications, Vol. 41, No. 11, Nov. 1993

[5] Hu, Y., H., "CORDIC-Based VLSI Architectures for Digital Signal Processing", IEEE Signal Processing Magazine, July 1992, p. 16–35

[6] Lee, J., Lang, T., "Constant-Factor Redundant CORDIC for Angle Calculation and Rotation", IEEE Trans. on Computer, Vol. 41, No. 8, August 1992, p. 1016–1025

[7] Frerking, M., E., "Digital Signal Processing in Communication Systems", Van Nostrand Reinhold, 1994