AEÜ Int. J. Electron. Commun.
51 (2003) No. 1, 1–9

Multipath–Resistant Time of Arrival Estimation for Satellite Positioning

R. Bischoff, R. Häb-Umbach, N. Sai Ramesh

Dedicated to Prof. Klaus Meerkötter on the occasion of his 60th birthday

Abstract **Satellite positioning systems, such as GPS or the future European system Galileo, employ direct-sequence spread-spectrum signals. The positioning accuracy is strongly affected by the quality of the pseudo range measurements. These measurements necessitate code and carrier synchronization of the received signal with the internally generated reference signals. In this type of systems one major error source is the multipath phenomenon, which results in a sum of delayed and weighted copies of the original signal to be present at the receiver input. This can result in a systematic error of the code tracking loop resulting in range errors in the order of several tens of meters.**

In this paper we propose an extension of the standard code tracking loop capable of estimating the parameters of the line-of-sight (LOS) signal and separating the LOS from the reflected signal portions. It is based on an analysis of the cross correlation of the received signal with a locally generated code sequence in the vicinity of the tracking point of a Delay-Locked Loop (DLL). For this reason, we call this method Cross Correlation Function (CCF)–Analysis. The proposed method achieves considerably more accurate estimates than a DLL. Its performance is comparable to the Multipath Estimating Delay-Locked Loop (MEDLL) which is considered to be the best method for reducing multipath–induced errors, so far. However, the computational complexity of the CCF–Analysis is by a factor of three smaller compared to the MEDLL. Extensive simulations have been conducted for the proposed method and the MEDLL in order to assess the robustness of the two approaches under various signal constellations.

Keywords **satellite navigation, multipath mitigation, synchronization.**

1. Introduction

Satellite positioning is based on the principle that one's position can be determined from distances measured to objects with known positions. From the propagation time measurements to at least four satellites the user's coordinates in three-dimensional space can be determined including an estimate of the clock offset between user and system clock [1].

With the spread spectrum signals used in the *Global Positioning System* (GPS) and the future European *Galileo* system time of arrival measurements can be obtained with high precision by conducting code and carrier phase synchronization of the received signal. While many sources of error (e.g. ionospheric and atmospheric propagation delays, ephemeris errors) can be eliminated by differential techniques, i.e. by employing a reference station close to the user's location which sends a correction signal to the user, this does not hold for errors due to multipath and receiver noise [2]. Errors in the code tracking loop due to reception of the direct signal from the satellite and one or more reflections from the ground or structures in the area, result in pseudorange errors at the meter or tens of meter level and thus need attention for high precision positioning. Influence of multipath on the carrier phase is at the cm-level and is therefore not considered in this paper [2].

Many approaches in literature address multipath at the signal processing level. Code synchronization is typically done by a DLL. The peak tracking error due to multipath depends on the spacing between the early and late correlator. Reducing this spacing from one chip period to one tenth of it in a "narrow correlator" DLL thus effectively combats multipath [4], at the expense of increased computational complexity, though.

Other approaches include the strobe and edge correlator [5], special correlator reference waveform design [6], and the Multipath Estimating Delay-Locked Loop (MEDLL) [8, 9]. Among those the MEDLL is considered to be the best method to compensate multipath for satellite positioning. However, its computational complexity is much higher than that of a DLL.

In this paper we propose a new scheme named cross correlation function (CCF)–Analysis for reducing multipath errors in a positioning receiver with reduced complexity and with performance comparable to the MEDLL method. It is based upon the analysis of cross correlation values of the received signal with the local code sequence in an interval around the DLL tracking point.

The paper is organized as follows: In Section 2 we describe the satellite channel models used in our simulations. Section 3 outlines DLL-based code synchronization and illustrates the multipath effect on tracking. In the following section the MEDLL is described. It serves as a reference method for comparison to our proposed solution, which is presented in Section 5. Section 6 presents simulation results for performance evaluation.

2. Channel Models

2.1 Simple Multipath Model

In this paper we will consider two channel models. The first one is a simple $(M+1)$–path model, in which the received signal is the sum of the direct, i.e. line-of-sight (LOS) signal and M reflected rays. The received signal in

Received May 16, 2003.

R. Bischoff, R. Häb-Umbach, N. Sai Ramesh, University of Paderborn, Department of Communications Engineering, Pohlweg 47–49, D-33098 Paderborn, Germany.

AEÜ Int. J. Electron. Commun.
51 (2003) No. 1, 1–9

complex baseband has the form

$$r(t) = \sum_{i=0}^{M} a_i e^{j\theta_i} \cdot c(t-\tau_i) \cdot d(t-\tau_i) + z(t), \qquad (1)$$

where (a_i, θ_i, τ_i) are the amplitude, phase and delay of the i-th signal component, which are all assumed to be constant. $c(t)$ and $d(t)$ are the spreading code and the data signal, respectively. In this paper we assume for simplicity that $d(t) \equiv 1$ In Eq.(1), $z(t)$ denotes additive complex–valued white Gaussian noise. In case of an infinite transmission bandwidth $c(t)$ is a pulse train consisting of rectangular pulses of duration T_C, where T_C is the chip period, and can be expressed as

$$c(t) = \sum_i B_i \cdot \mathrm{rect}\left(\frac{t - i \cdot T_C}{T_C}\right). \qquad (2)$$

Here, $B_i \in \{\pm 1\}$ is the pseudo-noise (PN) sequence. Throughout this paper we will assume a chip period of $T_C = 977$ ns ($1.023 \cdot 10^6$ chips/second), which is the value used for the SPS–Code of GPS.

Because of its simplicity, this channel model is often used for theoretical performance evaluations [12].

2.2 LMS Channel Model

The *Land to Mobile Satellite* (LMS) channel model is a complex and very realistic channel model, because it is based upon extensive helicopter measurements [10].

The received signal consists of a direct path and five reflected paths, each of them with additional diffuse multipath originating from shadowing and diffraction. The amplitudes of the diffuse multipath are Rayleigh distributed and can be approximated by a linear decrease as shown in [10] and their delays are exponentially distributed. The relative delays and relative amplitudes of the reflected paths, as well as their Doppler bandwidths were taken from [11].

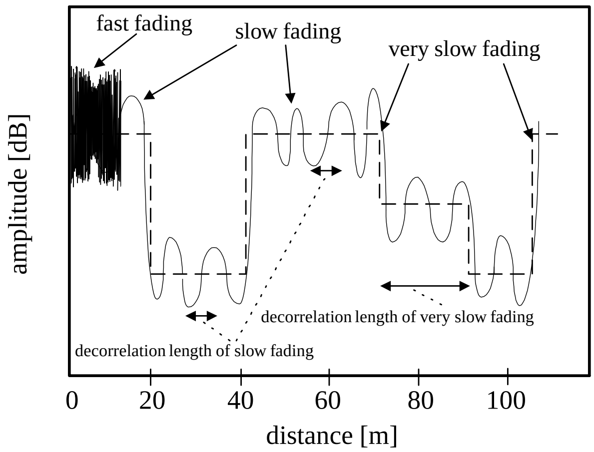


Fig. 1: Amplitude of the received signal versus driven distance.

The amplitudes of the respective paths exhibit three kinds of fading, as it is illustrated schematically in Fig. 1. For reasons of clarity the fast fading is only drawn for the first 15 meters.

3. Code Synchronization with DLL

3.1 Principle Operation

In spread spectrum systems code synchronization is typically a two-stage process consisting of acquisition and tracking. In the acquisition circuit code phase synchronization up to an error of $\pm T_C/2$ is achieved.

The fine synchronization or tracking is often done with a DLL with the block diagram depicted in Fig. 2. Here, the received signal is correlated with the two shifted versions $c(t - \tau_0' \pm \Delta \cdot T_C/2)$ of the code sequence. Δ is called the correlator spacing, which is equal to 1 in a standard DLL and 0.1 for a narrow correlator. Here, τ_0' is the delay of the locally generated code sequence.

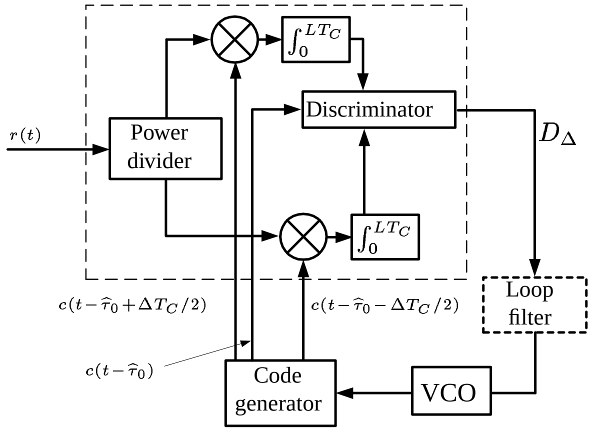


Fig. 2: Block diagram of a coherent DLL.

Let us first assume that the received signal is simply a delayed version of the code sequence, i.e. $r(t) = c(t - \tau_0)$. The purpose of the DLL is to estimate the unknown delay τ_0. The cross correlation $R_x(\tau)$ of the received signal with the early and late version of the code sequence yields

$$R_x\left(\tau_0' \mp \frac{\Delta}{2} T_C\right) = \frac{1}{LT_C} \int_0^{LT_C} c\left(t - \tau_0' \pm \frac{\Delta}{2} T_C\right) r(t) dt$$
$$= R_c\left(\tau_0' - \tau_0 \pm \frac{\Delta}{2} T_C\right). \qquad (3)$$

The integration interval has a length of LT_C and $R_c(\tau)$ is the autocorrelation function of the code sequence which is assumed to be ideal (i.e. we neglect effects due to the finite length of the PN sequence):

$$R_c(\tau) = \begin{cases} 1 - \frac{|\tau|}{T_C} & : \quad |\tau| \le T_C \\ 0 & : \quad \text{otherwise} \end{cases} \qquad (4)$$

Due to this symmetry it follows that the difference D_Δ between early and late correlation is zero for $\tau_0' = \tau_0$, i.e. if the delay τ_0' of the local code sequence is equal to the unknown τ_0. As a consequence, in a DLL the estimate $\widehat{\tau}_0$ for the unknown delay is set to be the zero crossing of the above discriminator D_Δ, i.e. in this case: $\widehat{\tau}_0 = \tau_0$.

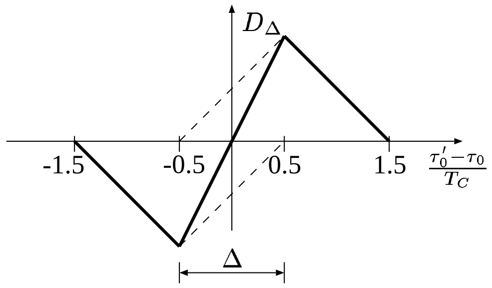


Fig. 3: S–curve of a coherent DLL with correlator spacing of $\Delta = 1.0$.

Fig. 3 shows the typical S-curve shape of the discriminator. The discriminator signal $D_\Delta(\tau_0' - \tau_0)$ is used in a feedback loop to track the unknown code phase τ_0.

3.2 Effect of Multipath

The presence of multipath degrades the tracking performance of the DLL because the local signal is correlated with the composite signal (LOS plus multipath) instead of the LOS signal only. To illustrate the effect of multipath on the DLL we assume that the received signal consists of the direct LOS path and an additional reflected path:

$$r_{2p}(t) = a_0 e^{j\theta_0} \cdot c(t - \tau_0) + a_1 e^{j\theta_1} \cdot c(t - \tau_1). \quad (5)$$

For positioning systems estimates for code and carrier phases τ_0 and θ_0 of the LOS-path have to be determined. For ease of illustration we assume $\theta_0 = \theta_1 = 0$. The correlations (3) now yield

$$a_0 R_c\left(\tau_0' - \tau_0 \mp \frac{\Delta}{2} T_C\right) + a_1 R_c\left(\tau_0' - \tau_1 \mp \frac{\Delta}{2} T_C\right) \quad (6)$$

As a result the discriminator signal consists of the superposition of two S–curves, one with zero crossing at $\tau_0' = \tau_0$ and the other with zero crossing at $\tau_0' = \tau_1$. Fig. 4 shows a degenerated S–curve (thick solid line) obtained as the superposition of the S–curve from the LOS-path (thin solid line) and the S–curve from the reflected path (dashed line).

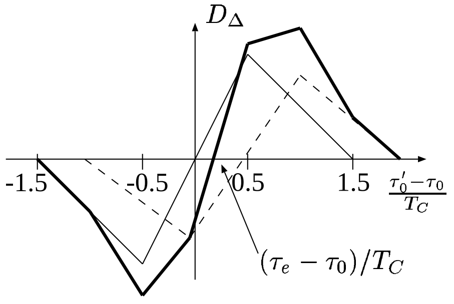


Fig. 4: Degenerated S–curve.

The tracking curve D_Δ does not have a zero crossing at $\tau_0' = \tau_0$, i.e. if the local delay is equal to the delay of the LOS-path, but rather at some delay $\tau_0' = \tau_e$. The estimate $\hat{\tau}_0$, which is defined as the zero crossing of D_Δ is thus $\hat{\tau}_0 = \tau_e$. The value of this systematic estimation error depends on the delay of the reflected path compared to the direct path $(\tau_1 - \tau_0)$, on the amplitude ratio a_1/a_0, and on the phases θ_0 and θ_1. Note that this is a **systematic** error due to multipath, which even exists in the absence of additive noise.

Fig. 9 in Section 6.1 shows the so–called *error envelope* of a narrow correlator DLL, i.e. the systematic range error $d_e = \pm\bar{c} \cdot (\tau_e - \tau_0)$ as a function of $(\tau_1 - \tau_0)$ for an amplitude ratio of $a_1/a_0 = 0.5$ ($\bar{c}$: speed of light). The upper curve results if the direct and reflected path add constructively ($\theta_1 = \theta_0$), and the lower corresponds to destructive composition. If $|a_1| = |a_0|$, the maximum error is $|\tau_e - \tau_0| = 0.05115 \cdot T_C$, corresponding to $d_e \approx \pm 15\, m$ [2, 4].

4. MEDLL

The MEDLL, which was introduced by Van Nee [8, 9], computes the Maximum Likelihood estimates of the parameters (a_i, θ_i, τ_i) of the individual components of the received signal. The likelihood function is given by

$$L(a_i', \theta_i', \tau_i') = \int_0^{LT_C} |r(t) - s(t)|^2 \, dt, \quad (7)$$

where

$$s(t) = \sum_{i=0}^{\widehat{M}-1} a_i' \cdot e^{j\theta_i'} \cdot c\left(t - \tau_i'\right). \quad (8)$$

In van Nee's approach the number of paths $\widehat{M}$ of the received signal is estimated once at the beginning and periodically after specified time intervals, whereas the parameters (a_i, θ_i, τ_i), $i = 0, \ldots, \widehat{M}$, are estimated in each tracking step. Setting the partial derivatives of (7) w.r.t. all unknowns to zero yields a set of coupled nonlinear equations [8]. With the received signal according to Eq. (1) and assuming absence of noise we can write for the CCF

$$R_x(\tau) = \sum_{i=0}^{M} R_i(\tau) \quad (9)$$

where $R_i(\tau)$ is the component of $R_x(\tau)$ coming from the i-th ray, c.f. Eq. (6). Knowledge of $R_x(\tau)$ is required to solve the ML-equations. In the MEDLL approach $R_x(\tau')$ is computed at delays $\tau' = k\Delta\tau$ in a parallel bank of correlators, as shown in Fig. 5. Note that in practice the input signal is sampled with some sampling period T and the integral has to be replaced by a sum. The cross correlation values $R_x(k\Delta\tau)$ are the input to the DSP processor, which solves the MEDLL equations.

The iterative algorithm proposed by van Nee attempts to estimate all unknowns (a_i, θ_i, τ_i), $i = 0, \ldots, \widehat{M}$, such that R_x can be decomposed into its constituents according to Eq. (9). Since the analytic form of $R_c(\tau)$ is known from Eq. (4), $R_x(\tau)$ can be reconstructed and thus the location

AEÜ Int. J. Electron. Commun.
51 (2003) No. 1, 1–9

of the maximum of $R_x(\tau)$ can be obtained, as opposed to just the delay $\widehat{k\Delta\tau}$ corresponding to the maximum of the $R_x(k\Delta\tau)$ sequence.

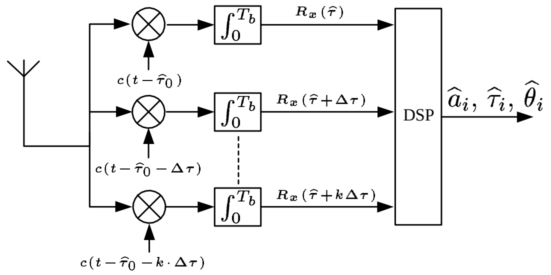


Fig. 5: Block diagram of MEDLL.

However, very small delays $(\tau_1-\tau_0)$ cannot be discerned, resulting in a residual code tracking error. This will be analyzed in more detail in Section 6.

5. CCF–Analysis

5.1 Preliminary Remarks and Definitions

In this section we present a novel method, called CCF–Analysis, for code synchronization in the presence of multipath. Its performance is similar to the MEDLL, however at reduced computational complexity. The proposed method is based upon an analysis of the cross correlation function of the received signal and the internally generated code sequence around the zero crossing of the tracking curve of a narrow correlator DLL. It is assumed that the received signal of a navigation receiver can be modeled as consisting of two paths [11]: $R_x(\tau) = R_0(\tau)+R_1(\tau)$. Assuming $|a_0| \geq |a_1|$, we have a maximum delay error of $\tau_e - \tau_0 \approx \pm 0.05115\cdot T_C$ according to section 3.2. Furthermore, we will always assume that $\tau_1 > \tau_0$. The CCF–Analysis is carried out in the interval

$$I_G := [\tau_e - 2.5\cdot\Delta\tau, \tau_e + 2.5\cdot\Delta\tau]\ , \qquad (10)$$

which is symmetric to τ_e. Subsequently, the interval I_G of width $5\cdot\Delta\tau$ is subdivided into the five intervals

$$I_k := \left[\tau_e+\left(k-\frac{1}{2}\right)\Delta\tau,\ \tau_e+\left(k+\frac{1}{2}\right)\Delta\tau\right]; k=-2,\ldots,2. \qquad (11)$$

Thus, τ_e lies in the middle of I_0. Furthermore, we choose a spacing of $\Delta\tau = 0.1023\cdot T_C$. Due to this choice and the maximum error $|\tau_e - \tau_0|$ derived above it is guaranteed that τ_0 lies in I_0. At the edges of the intervals I_k we compute the correlation coefficients (CC)

$$R_x^{(k)} := R_x\left(\tau_e+\left(k-\frac{1}{2}\right)\Delta\tau\right);\ \ k=-2,\ldots,3 \qquad (12)$$

Finally, the complex normalized slopes are computed by

$$S_k := \left(R_x^{(k+1)} - R_x^{(k)}\right)\cdot w; \quad k=-2,\ldots,2\ , \qquad (13)$$

where w denotes the normalizing factor $\frac{T_C}{w_{\text{Gold}}\cdot\Delta\tau}$ and w_{Gold} depends on the value of the ACF of the applied Gold sequence at $\pm T_C$. Fig. 6 illustrates the defined values.

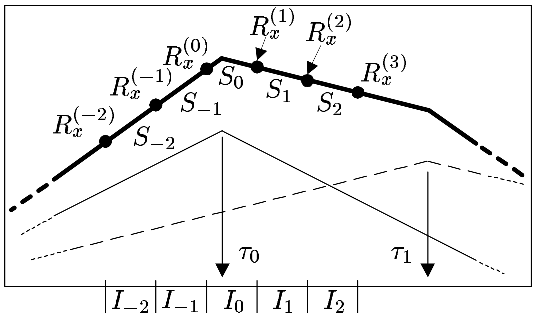


Fig. 6: Example illustrating the used notation.

Due to noise and the finite signal bandwidth two slopes S_i and S_j are usually not totally identical,. For this reason, we introduced a tolerance value δ_{tol}. In the rest of this section, two slopes S_i and S_j are said to be (un)equal if

$$|S_i - S_j| \underset{(>)}{\leq} \delta_{\text{tol}}\ . \qquad (14)$$

From the above definitions we can deduce the following four facts:

i) The ACF of the i-th signal component has slope changes at τ_i and $\tau_i \pm T_C$. Since τ_0 lies in I_0, there is no other slope change of R_0 in I_G. Likewise, there can be at most one slope change of the ACF of the second path, R_1, in I_G.

ii) If the interval I_i contains no slope change the respective slope S_i must be one of the following: $S_i \in \{\pm a_0, \pm a_1, \pm(a_0+a_1), \pm(a_0-a_1)\}$. Fig. 7 shows a typical CCF for two arriving paths with the corresponding slopes in the different intervals.

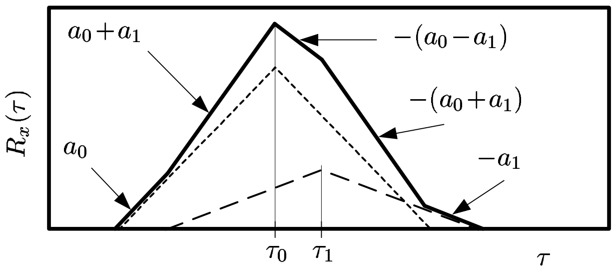


Fig. 7: Illustration of the different slopes of the CCF.

iii) If the slopes in two neighboring intervals are different, at least one of two intervals must contain a slope change of one of the ACFs.

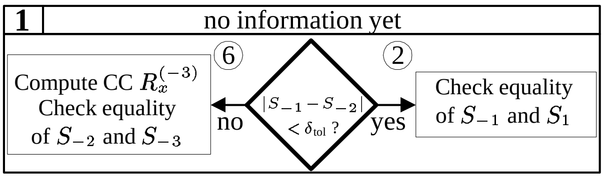
iv) If $\tau_1 - \tau_0$ is less than T_C, the slope of the CCF in the interval $[\tau_1 - T_C; \tau_0]$ has the negative value of the slope in the interval $[\tau_1; \tau_0 + T_C]$. The reverse is also valid: If for two slopes S_i and S_j with $i<j$ we obtain $-S_i = S_j$, then I_i must lie left and I_j right from all the maxima of the ACFs.

Next, the CCF–Analysis is presented as a stepwise procedure. For simplicity we assume real–valued signals (and correlations). The generalization to complex signals is done later on. In each step two slopes S_i and S_j are compared and conclusions are drawn from equality or inequality of these slopes. A maximum of five comparisons is required to obtain two linearly independent equations for the two unknowns a_0 and a_1.

In the following, each step is introduced by a diagram. The line at the top summarizes the conclusions drawn so far. The rhombus in the middle shows the investigated slope comparison. The boxes at the left and the right show the consequences in case of an equality or inequality, respectively. If another comparison is required, the number in the circle denotes the number of the step with which the CCF–Analysis continues. Since it has to be admitted that the reasoning is fairly complicated (though not really difficult), the method will be further illustrated by an example in Sec. 5.3.

5.2 Algorithm for CCF–Analysis

The CCF–Analysis starts with a comparison of the slopes in the intervals I_{-1} and I_{-2}, which lie left from τ_0.



Equality: In case of equality of S_{-1} und S_{-2} one of the following three cases is possible.

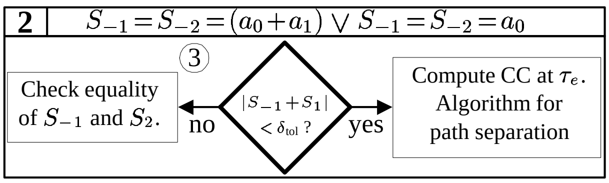
(I) There is only one relevant path[1]: $a_1 = 0$. It follows: $\boxed{S_{-1} = S_{-2} = a_0}$

(II) Both I_{-1} and I_{-2} are due to a rising edge of R_0 and R_1: $\boxed{S_{-1} = S_{-2} = (a_0 + a_1)}$

(III) The slope is due to the rising edge of R_0 only, because the second path is delayed such that $\tau_1 - T_C$ is right from I_{-1} $\boxed{S_{-1} = S_{-2} = a_0}$

After this, the CCF–Analysis continues with step 2 and a comparison of $-S_{-1}$ and S_1.

Inequality: Since τ_1 lies right from τ_0, in case of inequality $\tau_1 - T_C$ must be contained in I_{-1} or I_{-2}. The CCF–Analysis continues with step 6.



Equality: In case of an equality of $-S_{-1}$ and S_1 the interval I_1 must be right from the maxima of the ACFs of all contributing paths, i.e. there is only one path or two paths are very close together. We now only have the linearly independent equation

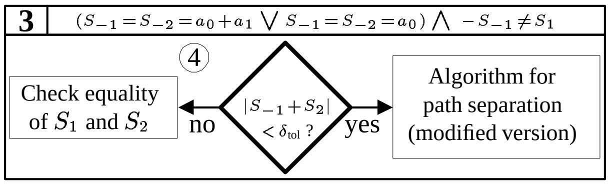
$$S_{-1} = S_{-2} = a_0 + a_1 \quad \text{or} \quad S_{-1} = S_{-2} = a_0 \,. \tag{15}$$

All other equations, e.g. $S_1 = -(a_0 + a_1)$ or $S_1 = -a_0$, respectively, are linearly dependent on Eq. (15). In order to check, if there is only one path or two paths with very small relative delay, a special separation algorithm was developed. This algorithm at first computes an additional CC $R_x^{(e)}$ at τ_e and the two additional slopes

$$\begin{aligned} S_0^{(1)} &:= \left(R_x^{(e)} - R_x^{(0)}\right) \cdot 2 \cdot w \quad \text{and} \\ S_0^{(2)} &:= \left(R_x^{(1)} - R_x^{(e)}\right) \cdot 2 \cdot w \,. \end{aligned} \tag{16}$$

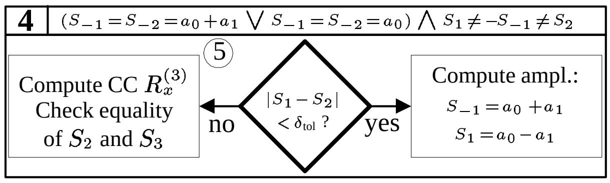
Based on these two slopes the algorithm calculates estimates for the amplitudes and the delay τ_0 of the direct component. In case of only one arriving path it yields $a_1 = 0$ and $\hat{\tau}_0$ is set to τ_e, since there is no multipath error. In this case the CCF–Analysis terminates here.

Inequality: In case of inequality of S_1 and $-S_{-1}$ we can draw the conclusion that there exists a reflected path (case **(I)** of step 1 can be excluded) and that I_1 can not be right from the maximum of the ACF of the reflected path. This means, that τ_1 lies right from the interval I_0. In case that τ_1 lies in I_1, the slope S_2 would have the same value as $-S_{-1}$. In order to check this, the CCF–Analysis continues with step 3, comparing the slopes S_2 and $-S_{-1}$.



Equality: An equality of S_2 and $-S_1$ means, that I_2 is right from all maxima of the ACFs. Because of the inequality of S_1 and $-S_{-1}$ in step 2, τ_1 can not be left from I_1 and, hence, I_1 must contain the slope change at τ_1. Again, there is only one independent equation and a slightly modified version of the separation algorithm is applied. This algorithm again yields an estimation value for τ_0 and the CCF–Analysis is terminated.

Inequality: An inequality can on one hand mean that (**I**) τ_1 lies in I_2 or right from I_2. Another possibility is that (**II**) $\tau_1 - T_C$ lies in I_0, I_1 or I_2. In order to check, which of the cases is valid, a comparison of S_1 and S_2 is performed in step 4.



Equality: In case of equality of S_1 and S_2 and due to the additional inequality of S_2 and $-S_{-1}$ we can conclude that either **(I)** I_{-1} to I_2 lie between τ_0 and τ_1 or **(II)** $(\tau_1 - T_C)$ lies in I_0. In both cases there is no slope jump in I_1 and I_2 and we obtain the equation

$$S_1 = a_1 - a_0 \,. \tag{17}$$

In order to determine whether **(I)** or **(II)** holds we compute an additional CC at $t_{\text{add}} = \tau_e + 0,9 \cdot T_C$ and the corresponding slope

$$S_{\text{add}} = \left(R_x^{(\text{add})} - R_x^{(1)}\right) \cdot \frac{w}{8.5}. \tag{18}$$

[1] There may exist a second path, which is, however, sufficiently delayed ($\tau_1 - \tau_0 > 1.05 T_C$) and therefore doesn't introduce an error, see Fig. 9.

If S_{add} is not equal to S_1 there must be a slope change between $R_x^{(1)}$ and $R_x^{(\text{add})}$ and $\tau_1 - T_C$ must be left from I_{-2}. We obtain

$$S_{-1} = a_0 + a_1 \tag{19}$$

and the amplitudes can be computed by

$$\mathbf{a} := \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} \frac{S_{-1}-S_1}{2} \\ \frac{S_{-1}+S_1}{2} \end{pmatrix} . \tag{20}$$

If $S_{\text{add}} = S_1$, then there can be no slope change between t_1 and t_{add} and $\tau_1 - T_C$ must lie in I_0. In this case we have

$$S_{-1} = a_0 \tag{21}$$

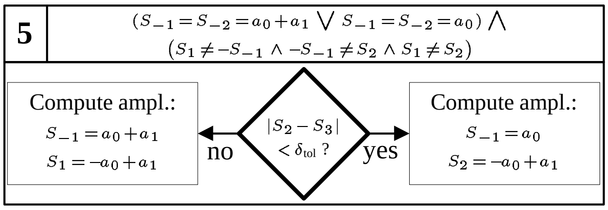
and the amplitudes can be computed according to

$$\mathbf{a} := \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} S_{-1} \\ S_{-1} + S_1 \end{pmatrix} . \tag{22}$$

Inequality: If S_1 is not equal to S_2, it follows that either **(I)** I_2 must contain the slope jump at τ_1 or **(II)** I_1 must contain the slope change at $\tau_1 - T_C$. $\tau_1 - T_C$ can not lie in I_2, because in this case we would have an equality of $-S_{-1}$ and S_1. Furthermore, the interval I_1 can not contain τ_1, since if it would contain τ_1, we would have an equality of $-S_{-1}$ and S_2. In order to find out, which of these both cases is valid, an additional CC $R_x^{(3)}$ at $\tau_e + 3.5 \cdot T_C$ and an additional slope

$$S_3 = \left(R_x^{(3)} - R_x^{(2)}\right) \cdot w \tag{23}$$

must be computed.



Equality: In case of an equality of S_2 and S_3, we can conclude that $\tau_1 - T_C$ must lie in I_1. Additionally to $S_{-1} = a_0$ we obtain the equation

$$S_2 = -a_0 + a_1 \tag{24}$$

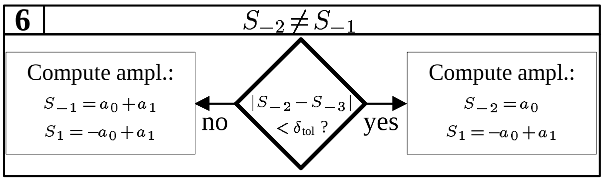
and the amplitudes can be computed by

$$\begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} S_{-1} \\ S_{-1} + S_2 \end{pmatrix} \tag{25}$$

Inequality: In case of inequality τ_1 must lie in I_2. Thus, the interval I_1 lies between τ_0 and τ_1 and we obtain

$$S_1 = -a_0 + a_1 \quad \text{and} \tag{26}$$
$$S_{-1} = a_0 + a_1 \, . \tag{27}$$

Again we have two linearly independent equations and are able to compute the amplitudes according to Eq. (20).



If S_{-1} and S_{-2} are unequal, either I_{-1} or I_{-2} must contain the slope jump at $\tau_1 - T_C$. In order to check, which of these two intervals contains the slope jump, the additional correlation coefficient $R_x^{(-3)}$ at

$$t_{-3} = \tau_e - 3,5 \cdot T_C = t_{-2} - \Delta\tau \tag{28}$$

and the additional slope

$$S_{-3} := \left(R_x^{(-2)} - R_x^{(-3)}\right) \cdot w \tag{29}$$

are computed. Then S_{-3} is compared to S_{-2}.

Equality: In case of an equality of S_{-2} and S_{-3}, the interval I_{-1} must contain the slope jump at $\tau_1 - T_C$. Thus, both I_{-2} and I_{-3} are left from $\tau_1 - T_C$, such that both intervals only contain a portion of the ACF of the direct path and $S_{-1} = S_{-2} = a_0$ is valid. Using Eq. (17) we obtain

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} S_{-2} \\ S_1 \end{pmatrix} \tag{30}$$

and the amplitude vector

$$\mathbf{a} = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} S_{-2} \\ S_1 - S_{-2} \end{pmatrix} . \tag{31}$$

Inequality: In case of an inequality, $\tau_1 - T_C$ must lie in I_{-2}. Then $\tau_1 - T_C$ lies left from I_{-1} and

$$S_{-1} = a_0 + a_1 \tag{32}$$

is valid. From the equations (17) and (32) we obtain

$$\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} S_{-1} \\ S_1 \end{pmatrix} \tag{33}$$

and the amplitude vector according to Eq. (20).

Now, for all possible constellations the amplitudes of both paths are determined. The procedure can be readily extended to complex-valued signals, where we compute the slopes (a_{0r}, a_{1r}) from the real and (a_{0i}, a_{1i}) from the imaginary part of the correlation function. In the next step the phases θ_i, $i = 0, 1$ and the phase difference θ_{diff} are calculated:

$$\theta_{\text{diff}} = \theta_0 - \theta_1 = \arctan\left[\frac{a_{0i}}{a_{0r}}\right] - \arctan\left[\frac{a_{1i}}{a_{1r}}\right] . \tag{34}$$

Now, the parameters (a_i, θ_i), $i = 0, 1$ and $\tau_{\text{diff}} := (\tau_1 - \tau_0)$ have been determined (at least approximately), and the final step is to obtain an estimate of the bias $(\tau_e - \tau_0)$ from the error envelope:

$$\widehat{\tau}_e - \tau_0 = 2 \cdot \frac{a_1}{a_0} \cdot F(\tau_{\text{diff}}, \theta_{\text{diff}}) \cdot \cos(\theta_{\text{diff}}) \, . \tag{35}$$

$F(\tau_{\text{diff}}, \theta_{\text{diff}})$ is defined by

$$F(\tau_{\text{diff}}, \theta_{\text{diff}}) = \begin{cases} \text{upper part of the error envelope} \\ \quad \text{if } \operatorname{sgn}(\cos(\theta_{\text{diff}})) > 0 \\ \text{lower part of the error envelope} \\ \quad \text{if } \operatorname{sgn}(\cos(\theta_{\text{diff}})) \leq 0 \end{cases} . \tag{36}$$

Because the DLL error envelope is constant to a large extent, we only need to know the exact delay difference τ_{diff} for very small values and for values in the vicinity of T_C. For $\tau_{\text{diff}} \approx 0$ the separation algorithm directly computes an estimate for τ_0 so that this case is not critical. In the other case we computed the additional CC $R_x^{(3)}$ in step 4 and obtained an equality of S_1 and S_{add}. τ_{diff} is then between $0.9 \cdot T_C$ and $1.1 \cdot T_C$. In this case we assume an average value of $\tau_{\text{diff}} = T_C$ and read the corresponding bias $(\tau_e - \tau_0)$ from the error envelope. More exact results could be obtained if more correlations were computed.

In the following section the CCF–Analysis is illustrated by means of an example.

5.3 Example for the CCF–Analysis

For the following example we assume real–valued signals for the direct path and the reflected path with amplitudes $a_0 = 1$ and $a_1 = 0.5$. $(\tau_1 - \tau_0)$ was chosen to $0.225 \cdot T_C$. The resulting CCF and the computed slopes are shown in Fig. 8. The slopes are normalized for clarity. Under the given assumptions the narrow-correlator DLL multipath error is $0.025575 \cdot T_C$, which corresponds to $\Delta\tau/4$.

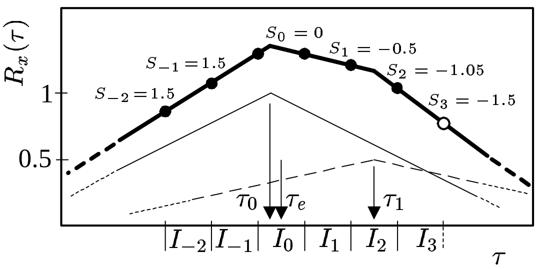


Fig. 8: Sample CCF with assumed values.

First (step 1) the slopes $S_{-1} = 1.5$ and $S_{-2} = 1.5$ are compared. From the equality we can follow that there is no jump in I_{-1} or I_{-2}, i.e. one of the three cases given above in step 1 is given. This means there is **(I)** only one path or **(II)** τ_1 lies right from I_{-1} and $\tau_1 - T_C$ is left from I_{-2} or **(III)** $\tau_1 - T_C$ lies right from I_{-1}.

In the next step (step 2) $-S_{-1} = -1.5$ and $S_1 = -0.5$ are compared. In the case of only one path or $\tau_1 \in I_0$ or if the peak of the second path lies right from I_1 both slopes would be equal. In our example we have an inequality of both slopes so that there is at least one reflected path and either **(I)** I_{-1} and I_{-2} must lie between $\tau_1 - T_C$ and τ_0 or **(II)** $\tau_1 - T_C$ is right from I_{-1}.

Now (step 3) $-S_{-1} = -1.5$ and $S_2 = -1.05$ are compared. An equality would mean that the interval I_2 must lie right from all the maxima of the ACFs (see fact **(iV)** in Sec.5.1). However, in our example we obtain an inequality of $-S_{-1}$ and S_2. This means that either **(I)** I_{-2}, I_{-1} and I_1 lie between $\tau_1 - T_C$ and τ_1 or **(II)** $\tau_1 - T_C$ must lie in I_0 or right from I_0.

In order to find out which of these cases is valid, a comparison of the slopes $S_1 = -0.5$ and $S_2 = -1.05$ is performed (step 4). In our example this comparison yields an inequality and we can conclude that there must be a slope jump in either I_1 or I_2, so that only two cases are possible. Either **(I)** $\tau_1 - T_C$ lies in I_1 or **(II)** τ_1 lies in I_2. The other cases can be ruled out. For example, $\tau_1 - T_C$ can not lie right from I_1, since if it lied right from I_1, the slopes $-S_{-1}$ and S_1 would be equal. Moreover, $\tau_1 - T_C$ can not be contained in I_0, because otherwise there would be no slope change in I_1 or I_2. Note that the constellation with $\tau_1 - T_C$ right from I_2 is considered as situation with only one path, because the second path does not affect the DLL in this case.

An additional CC $R_x^{(3)}$ at $\tau_e + 3.5 \cdot T_C$ and the additional slope S_3 must be computed. The slope $S_3 = -1.5$ is subsequently compared with $S_2 = -1.05$. In our example this comparison yields an inequality. We can conclude that there is a slope jump in one of these two intervals. From the inequality of S_1 and S_2 we already know that either I_1 or I_2 must contain a slope jump. Because there are at most two jumps in I_G, the interval I_2 must contain the slope jump at τ_1.

We can now conclude that I_1 lies right from τ_0 and left from τ_1, such that $S_1 = -0.5$ must be equal to $-a_0 + a_1$. Furthermore, we know that $S_{-1} = 1.5$ must be equal to $a_0 + a_1$, because I_{-1} lies between τ_0 and $\tau_1 - T_C$. We now have the two equations

$$\begin{aligned} a_0 + a_1 &= 1.5 \\ -a_0 + a_1 &= -0.5 \end{aligned} \tag{37}$$

and can compute the amplitude vector

$$\mathbf{a} = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0.5 \end{pmatrix} \tag{38}$$

After that, the phases (θ_0, θ_1) would have to be computed. In this example we assumed real-valued signals, so $\theta_{\text{diff}} = 0$. Furthermore, we know that the delay is greater than $\Delta\tau$ and less than $3 \cdot \Delta\tau$. In this region the error envelope is constant and using Eq. (35) we can estimate the multipath error $\tau_e - \tau_0$ by

$$\begin{aligned} \widehat{\tau}_e - \tau_0 &= 2 \cdot 0.5 \cdot F(0.2 \cdot T_C, 0) \cdot \cos(0) \\ &= 0.025575 \cdot T_C \end{aligned} \tag{39}$$

6. Performance Analysis

6.1 Error Envelopes

Fig. 9 compares the systematic range error $d_e = \bar{c} \cdot (\hat{\tau}_0 - \tau_0)$ as a function of the delay $(\tau_1 - \tau_0)$ of the reflected path to the direct path for a narrow correlator DLL ($\Delta = 0.1$), MEDLL, and CCF–Analysis. Here we assumed a 2–ray channel model (i.e. $M = 1$) without additive noise. The curves above the abscissa correspond to $\theta_1 = \theta_0$ and the curves below the abscissa to $\theta_1 = \theta_0 + \pi$, respectively. For all other phase values the error lies within this "error envelope". In this figure we assumed a signal-to-multipath (SMR) ratio of $a_0/a_1 = 2$, i.e. 6 dB. For the MEDLL and CCF results we chose $\Delta\tau$ to $0.1023 \cdot T_C$. The signal bandwidth was chosen to 2.046 MHz corresponding to the bandwidth of the C/A–code signal and a chip rate of 1.023 Mchip/s was assumed. For a wide range of delay differences $(\tau_1 - \tau_0)$, both MEDLL and CCF–Analysis perform virtually biasfree. The error envelope of the CCF–Analysis exhibits two regions of residual systematic errors. On one hand reflected paths with delays less than $0.1 \cdot T_C$ can not be perfectly separated by the separation algorithm. The other region is around $1 \cdot T_C$. If both paths are about one chip duration apart (decreasing/increasing edge of the error curve), we can only determine that the delay is in the interval $[0.9 \cdot T_C; 1.1 \cdot T_C]$.

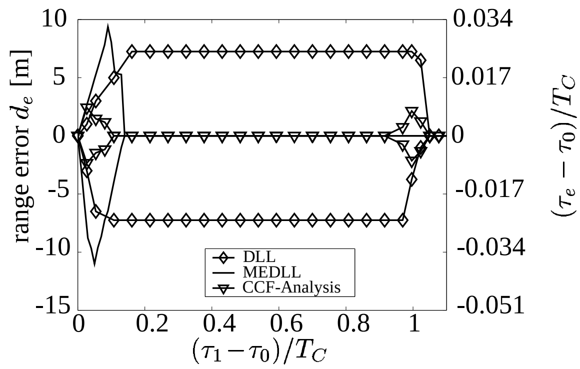


Fig. 9: Error envelopes for the CCF–Analysis, the MEDLL and the conventional DLL.

For this reason, the estimated multipath error $\widehat{\tau}_e$ is afflicted with an error. Overall, the performance of the CCF–Analysis is similar to the performance of the MEDLL. For small delays the CCF–Analysis even outperforms the MEDLL if the same number of samples per chip is used. For both methods, errors for small relative delays can be reduced if $\Delta\tau$ is reduced, i.e. if more values of the correlation function R_x are computed. For the MEDLL performance given in Fig. 9 ten correlation values per chip were used.

6.2 Performance in AWGN

Fig. 10 compares the performance of MEDLL and CCF–Analysis in the presence of additive white gaussian noise (AWGN). For both methods we assumed an integration time of 600 ms, which corresponds to the duration of 30 data bit in GPS. This long correlation time yields to a SNR higher than 30 dB after correlation, such that the correlation coefficients and, thus, the slopes can be computed with a sufficient accuracy. For the multipath-free case both methods deliver a bias free estimation. Fig. 10 shows the variance $\sigma^2_{d_e}$ of the estimated tracking error $d_e = \tau_e - \tau_0$ as a function of the signal-to-noise ratio at the input of a satellite navigation receiver. SNRs between -30 and -14 dB are typical values at the input of a satellite navigation receiver [2], the SNR becomes positive after despreading. It can be seen that the variances for both methods are nearly identical for all SNR values. The two other curves show the tracking error variance of both methods in case of an additional reflected path delayed by $0.25 \cdot T_C$ and an SMR of 6 dB. The variance obtained from the CCF–Analysis is again in the same region as the variance for the MEDLL. If the more complex and realistic LMS channel model and AWGN is employed, the tracking error variance is increased for both methods, see Fig. 11.

One reason for this is the fast fading which affects the performance of both methods. Another reason is the diffuse multipath spread, which can not be compensated by means of either MEDLL or CCF–Analysis.

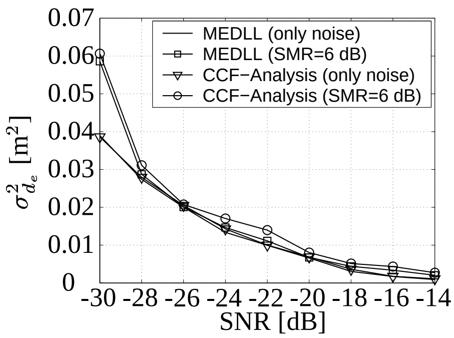


Fig. 10: Tracking error variances for different scenarios.

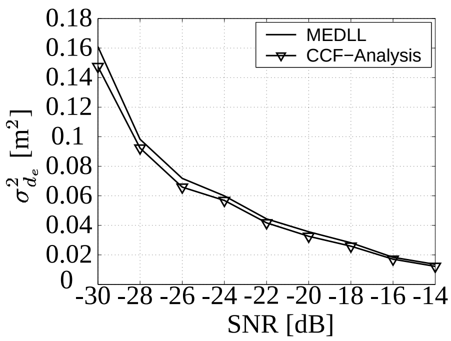


Fig. 11: Tracking error variances for the realistic LMS channel model.

For this realistic scenario both methods show approximately the same performance. The good performance of the CCF–Analysis is remarkable, since we assumed in the derivation, that the received signal consists of a direct and one reflected component only. Also for the MEDLL we had set $\widehat{M} = 1$ in all simulations. Overall, the performance of the CCF–Analysis is comparable to the MEDLL performance. In some scenarios it even outperforms the MEDLL.

6.3 Computational Costs

The computational complexity of the MEDLL is mainly determined by the number of correlations, since the iterative solution of the ML equations can be done on a DSP and is not the computational bottleneck. Note that the correlator spacing $\Delta\tau$ is independent of the sampling period T, although it may often be chosen equal. The systematic error seen in Fig. 9 can be reduced by increasing N_τ, the number of correlation values per chip, however, at the cost of increased computational complexity.

Correlation values have to be computed for the whole delay range where $R_x(\tau)$ has values different from zero. This range is of size $3.05 \cdot T_C$ ($1 \cdot T_C$ left from τ_0 and

$2.05 \cdot T_C$ to the right of τ_0). This means approximately $3 \cdot N_\tau$ values. Because of the large number of correlations the MEDLL is only implemented in monitoring stations and in DGPS–stations.

One correlation value is computed by multiplying the sampled input signal with the locally generated code sequence over an interval of L chips. The number of multiplications and additions is thus $L \cdot N_s$, where $N_s = T_s/T$ is the number of samples per chip. For MEDLL an integration time $L \cdot T_C$ exceeding one second should be used [9].

Table 1: Comparison of the computational costs.

	DLL	MEDLL	CCF–Anal.
# correlations	3	>30	≤ 9
Integration Time $L \cdot T_C$	0.1 s	>1 s	<0.5 s

For the CCF–Analysis the computational complexity is also mainly determined by the number of correlations. At first, two correlations are required to find the zero crossing of the S-curve. Furthermore, six or seven correlations are needed to determine the amplitudes and the delays (if required) of the respective paths. Hence, at most nine correlations are needed to obtain the performance given in the figures. In comparison to this, a conventional DLL requires two correlations (early, late) and, depending on the kind of discriminator, a third (on-time) one. Table 1 compares the number of correlation values to be computed to obtain an estimate of the code delay. The table also contains typical values for the integration times used for the computation of the correlation values.

7. Conclusions and Outlook

In this paper we proposed a new method capable of compensating for multipath errors in satellite navigation systems. It is based upon an analysis of the cross correlation function of the received signal and the internally generated code sequence in an interval around the DLL tracking point. In comparison to a MEDLL, which is nowadays regarded as the method with the best performance, the computational cost is reduced by approximately a factor of three. Simultaneously, simulation results showed that the performance of the CCF–Analysis is comparable to and in some cases even better than the performance of the MEDLL. This is also valid for a realistic LMS channel model, comprising six paths and additional diffuse multipath spread. In future we have to examine, in how far the bit resolution of the applied ADC will affect the method.

References

[1] P. Misra, B.P. Burke, M.M. Pratt, “GPS performance in Navigation”, *Proc. of the IEEE*, Vol. 87, No.1, Jan. 99, pp. 65–85.

[2] M.S. Braasch, A.J. van Dierendonck, “GPS Receiver Architectures and Measurements”, *Proc. of the IEEE*, vol. 87, no.1, Jan. 99, pp. 48–64.

[3] C.C. Counselman, “Multipath-Rejecting GPS Antennas”, *Proc. of the IEEE*, vol. 87, no.1, Jan. 99, pp. 86–91.

[4] A. J. van Dierendonck, P. Fenton, T. Ford, ”Theory and Performance of Narrow Correlator Spacing in a GPS Receiver”, *NAVIGATION, Journal of the Institute of Navigation*, Vol. 39, No. 3, Fall 1992, pp. 265–283.

[5] L. Garin, J.-M. Rousseau, ”Enhanced Strobe Correlator Multipath Mitigation for Code and Carrier”, *Proc. of the* 10^{th} *Int. Technical Meeting of the Satellite Division of the Institute of Navigation, ION-GPS-97*, Vol. 1, pp. 559–568.

[6] L. Weill, “GPS Multipath Mitigation by Means of Correlator Reference Waveform Design”, *Proc. Institute of Navigation Nat. Tech. Meeting*, Jan. 1997, pp. 197-206.

[7] C. Macabiau, B. Roturier, A. Benhallam, ”Performance of GPS Receivers with more than one Multipath”, *Proc. of the ION–GPS99*, Nashville, 1999.

[8] R. van Nee, “The Multipath Estimating Delay Locked Loop”, *Proc. of 2nd IEEE Symp. on Spread Spectrum Techniques and Applications*, Yokohama, 1992, pp. 39-42.

[9] R. van Nee, *Multipath and Multi-Transmitter Interference in Spread-Spectrum Communication and Navigation Systems*, Ph.D. Dissertation, Delft Univ. Techn., Delft, 1995.

[10] F. P. Fontan, M. Vazquez-Castro, C. E. Cabado, J. P. Garcia, and E. Kubista, ”Statistical Modeling of the LMS Channel”, *IEEE Transactions on Veh. Techn.*, Vol. 50, No. 6, Nov. 2001.

[11] Galileo Phase B2, ”User Segment Requirements Document”, DRD 033, RQS–GAL–0017-GLI, Issue, March 2002.

[12] Parkinson, Bradford W. and James J. Spilker. eds. 1996. *GPS: Theory and Practice*. Volumes I and II. Washington DC: American Institute of Aeronautics and Astronautics, Inc.

Renke Bischoff received his Dipl.-Math degree in Technomathematics in 1999 from the University of Paderborn. Since 1999 he works as a research scientist at the Dept. of Communications Engineering at the University of Paderborn, Germany. He works in the area for receiver structures of modern satellite navigation systems. His special interest is the multipath mitigation in satellite navigation receivers. He is now working towards his PHD.

Reinhold Häb–Umbach obtained a Dipl.-Ing. and Dr.-Ing. degree in Electrical Engineering from Aachen University of Technology in 1983 and 1988, respectively. From 1983 to 1988 he was with the Lehrstuhl fuer Elektrische Regelungstechnik, Aachen University of Technology, working on digital receiver design. From 1988 to 1989 he was a postdoctoral fellow at the IBM Almaden Research Center, San Jose, CA, conducting research on coding and signal processing for recording channels. As a research staff member of Philips Research he worked on various aspects of automatic speech recognition from 1990 to 2001. Since 2001 he is a full professor in communications engineering at the University of Paderborn, Germany.

Sai Ramesh Nammi received his Master of Technology in Communication Systems from Indian Institute of Technology, Madras in 2000. He was with Motorola India Electronics Limted Hyderabad India for 2.5 years. Since 2002 he is working as a research staff member at the Dept. of Communications Engineering at the University of Paderborn, Germany. His research interests are mainly in reducing multipath mitigation for navigation receivers and reducing multi transmitter interference for communication receivers.