Low Delay Spread Multi-Path Cancellation for 3G WCDMA

Daniel Iancu*, John Glossner†*, and Mayan Moudgill*

*Sandbridge Technologies, Inc.
1 North Lexington Ave.
White Plains, NY 10601 USA
{diancu, jglossner, mayan}@SandbridgeTech.com

†Delft University of Technology
Computer Engineering
Delft, The Netherlands

Abstract In a multi-path environment where the path delay of multi-path structures are less
than a chip delay, multi-path interference contributions significantly impair the received signal.
Even in high signal-to-noise ratio situations if the time delay between two paths is one quarter of
a chip the error rate is significantly higher. In this paper we describe a multi-path interference
cancellation method for closely spaced multi-path delays suitable for software
implementation [1]. For closely spaced path delays, we derive a system of equations that
improves the symbol estimate. For simplicity, but without loss of generality, the mathematical
description is derived for a single user, with AWGN. The noise term includes inter-cell
interference and thermal noise coming from the front end of the receiver. The described multi-
path cancellation method is applicable to closed spaces or very dense urban areas at 2 Mbps
data rate.

I. INTRODUCTION

One of the most common signaling schemes in current wireless services is the spread spectrum
technique in the form of direct-sequence (DS), frequency hopping (FH) or orthogonal frequency-
division multiplexing (OFDM). In these signaling schemes, multi-path, multi-user and in-band
interference plays a major role. Many techniques targeting interference cancellation have
emerged in the last years [2][3][4][5]. The majority of them deal with various kinds of
interference but few of them having any practical implementation possibilities using current
technology. A smaller number of papers have been dedicated to a special type of interference
where two or more paths are subject to a fraction of a chip delay [6],[7] A common practice,
attractive to real time implementation, is to employ filtering to minimize the effect of the
interference [8] . In this paper sub-chip delay interference is cancelled by solving a set of linear
matrix equations for the symbols on each path. The resulting symbols are then combined. The
entries for matrices are calculated and stored in nonvolatile memory and used as required.

II. THEORETICAL DESCRIPTION

A block of user data, in vector form, is described by

\[ x = [x[0] \hspace{10pt} x[T_s] \hspace{10pt} x[2T_s] \hspace{10pt} \ldots \hspace{10pt} x[kT_s] \hspace{10pt} \ldots]^T \]  (1)

In (1) \( T_s \) represents the symbol rate. A block of data can be a slot, a frame or a fraction of a slot
or frame. Blocks of data are processed and transmitted successively at the transmitter.

At the transmitter side the processing consists mainly of spreading, scrambling, and shape
filtering. Applied to the input vector \( x \), the output vector is described by
\[ y = [y[0] \ y[Tc] \ y[2Tc] \ \cdots \ y[kTc] \ \cdots]^T \] (2)

where \( Tc \) is the chip rate. In matrix representation, (2) becomes:
\[ y = (x \ s \ S) F \] (3)

where:
\[
S = \begin{bmatrix}
   s_1 & s_2 & \cdots & s_M & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
   0 & 0 & \cdots & 0 & s_1 & s_2 & \cdots & s_M & 0 & 0 & 0 & 0 \\
   \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
   0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & s_1 & s_2 & \cdots & s_M \\
   \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
   \end{bmatrix}
\] (4)

\( s \) (4) is the spreading matrix, \( M \) is the spreading factor length,
\[
S = \begin{bmatrix}
   S_1 & 0 & 0 & \cdots & 0 \\
   0 & S_2 & 0 & \cdots & 0 \\
   0 & 0 & S_3 & \cdots & 0 \\
   \vdots & \vdots & \vdots & \cdots & \vdots \\
   0 & 0 & 0 & \cdots & S_Q \\
   \end{bmatrix}
\] (5)

\( S \) (5) is the scrambling matrix of size \( Q \times Q \) with complex entries \( S_i \) with the number of symbols being \( Q/M \), and \( F \) (6) is the FIR shaping filter matrix (block Toeplitz matrix) of dimensions \( Q \times (Q+L-1) \) with elements \( f[kTc] \), where \( Tc \) is the sampling rate (or chip rate at one sample per chip), and \( L \) is the number of samples of the sampled shaping filter derived by sampling the continuous function \( f(t) \):
\[
F = \begin{bmatrix}
   f[1 \cdot Tc] & f[2 \cdot Tc] & \cdots & f[L \cdot Tc] & 0 & 0 \\
   0 & f(1 \cdot Tc) & \cdots & f[(L-1)Tc] & f[L \cdot Tc] & 0 \\
   \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
   0 & 0 & f(1 \cdot Tc) & \cdots & 0 & 0 \\
   0 & 0 & 0 & \cdots & f(1 \cdot Tc) & f[2 \cdot Tc] \cdot f[(L-1)Tc] & f[L \cdot Tc] \\
   \end{bmatrix}
\] (6)

The channel model is described by:
\[
h(t) = \sum_{i=1}^{Np} c_i(t) \delta(t - \tau_i) \] (7)

where \( \delta \) is the delta function, \( c_i(t) \) are the time varying complex channel coefficients, and \( \tau_i \) are the path delays. The received base-band signal is described as:
\[
\int_{-\infty}^{+\infty} y(t) \sum_{i=0}^{Np-1} c_i(t) \delta(t - \tau_i) dt + n(t) \] (8)

where \( n(t) \) is the noise term including the thermal noise and the inter-cell interference, and \( Np \) is the number of paths. After integration, (8) becomes:
\[
\mathbf{r}(\tau) = \sum_{i=0}^{N_d-1} \mathbf{x}(\tau_i)\mathbf{s}(\tau_i)\mathbf{S}(\tau_i)c_i(\tau_i)\mathbf{F}(\tau_i) + \mathbf{n}(\tau) \tag{9}
\]

A Rake search will estimate the time delays \(\hat{\tau}_i\), while the channel estimation procedure will estimate the channel complex coefficients \(\widehat{c}_i\), where the hat indicates estimated value. A method to track the multipath coefficients, for example, is described in [8].

At the receiver side, the received signal \(\mathbf{r}(\tau)\) is filtered using the same shaping filter \(f(t)\) used at the transmitter side.

\[
\mathbf{\psi}(\tau) = \mathbf{r}(\tau) \otimes f(\tau_0) = \left[ \sum_{i=\tau_i}^{N_d-1} \mathbf{c}_i(\tau_i)\mathbf{x}(\tau_i)\mathbf{s}(\tau_i)\mathbf{S}(\tau_i)\mathbf{F}(\tau_i) \right] \otimes f(\tau_0) + \mathbf{n}(\tau)
\]

\[
= \sum_{i=\tau_i}^{N_p} \mathbf{c}_i(\tau_i)\mathbf{x}(\tau_i)\mathbf{s}(\tau_i)\mathbf{S}(\tau_i)\mathbf{R}_{gg}(\tau_i - \tau_0) + \mathbf{n}(\tau) \tag{10}
\]

In equation (10), \(\mathbf{n}(\tau) = \mathbf{n}(\tau)\mathbf{F}(\tau_0)\) and \(\mathbf{R}_{gg}(\tau_i - \tau_0)\) is a double convolution matrix similar to \(\mathbf{F}\), with the entries being the autocorrelation sequence of \(\mathbf{F}\), and \(\otimes\) represents convolution. Using the notation \(\mathbf{A}_{ss}(\tau_i) = \mathbf{s}(\tau_i)\mathbf{S}(\tau_i)\), (10) can be rewritten as

\[
\mathbf{\psi}(\tau) = \sum_{i=\tau_i}^{N_p} \mathbf{c}_i(\tau_i)\mathbf{x}(\tau_i)\mathbf{A}_{ss}(\tau_i)\mathbf{R}_{gg}(\tau_i - \tau_0) + \mathbf{n}(\tau) \tag{11}
\]

Assuming the \(\mathbf{c}_i(\tau_i)\) coefficients are constant for the duration of a data block, and multiplying both sides of (11) by \(\mathbf{R}_{gg}^{-1}(\tau_0 - \tau_i)\mathbf{A}_{ss}^H(\hat{\tau}_0)\) we obtain (12):

\[
\mathbf{\psi}(\tau)\mathbf{R}_{gg}^{-1}(\tau_0 - \tau_i)\mathbf{A}_{ss}^H(\hat{\tau}_0) = \sum_{i=\tau_i}^{N_p} \mathbf{c}_i(\tau_i)\mathbf{x}(\tau_i)\mathbf{A}_{ss}(\tau_i)\mathbf{R}_{gg}(\tau_i - \tau_0)\mathbf{R}_{gg}^{-1}(\tau_0 - \tau_i)\mathbf{A}_{ss}^H(\hat{\tau}_0) + \mathbf{n}(\tau)\mathbf{R}_{gg}^{-1}(\tau_0 - \tau_i)\mathbf{A}_{ss}^H(\hat{\tau}_0)
\]

\[
= \mathbf{x}(\tau_i)\mathbf{c}_i^T(\tau_0)\mathbf{A}_{ss}(\tau_i)\mathbf{A}_{ss}^H(\tau_0) + 
\]

\[
+ \sum_{i=\tau_i}^{N_p} \mathbf{c}_i(\tau_i)\mathbf{x}(\tau_i)\mathbf{A}_{ss}(\tau_i)\mathbf{R}_{gg}(\tau_i - \tau_0)\mathbf{R}_{gg}^{-1}(\tau_0 - \tau_i)\mathbf{A}_{ss}^H(\hat{\tau}_0) + \mathbf{n}(\tau) \tag{12}
\]

where, \(\mathbf{n}(\tau)\mathbf{R}_{gg}^{-1}(\tau_0 - \tau_i)\mathbf{A}_{ss}^H(\hat{\tau}_0) = \mathbf{n}(\tau)\) and \(\mathbf{R}_{gg}^{-1}(\tau_0 - \tau_i)\) is the pseudo inverse of \(\mathbf{R}_{gg}(\tau_0 - \tau_i)\). The last sum in the right hand side of (12) is the multi-path interference term and is shown in (13):

\[
\sum_{i=\tau_i}^{N_p} \mathbf{c}_i(\tau_i)\mathbf{x}(\tau_i)\mathbf{A}_{ss}(\tau_i)\mathbf{R}_{gg}(\tau_i - \tau_0)\mathbf{R}_{gg}^{-1}(\tau_0 - \tau_i)\mathbf{A}_{ss}^H(\hat{\tau}_0) \tag{13}
\]
If the path delay is larger than a chip, the last sum in (12) is negligible due to the correlation properties of the scrambling matrix S. For path delays less than a chip, (13) is not negligible and including it in the calculation provides a better symbol estimate.

For the zero path, (12) reduces to (14):

$$\hat{x}(\tau_0)c_0(\tau_0)c_0(\tau_0)M = \psi(\tau)R_{rf}^H(\tau_0 - \tau_0)\Lambda_H^H(\tau_0)c_0(\tau_0) - c_0(\tau_0)\sum_{l=1}^{N_{OSV}} c_l(\tau_l)x(\tau_l)A_{ss}(\tau_l)R_{rf}(\tau_l - \tau_0)R_{rf}^{-1}(\tau_0 - \tau_0)\Lambda_H^H(\tau_0) + n(\tau)$$

(14)

If the path delays are larger than the duration of a chip, the last sum in (14) is negligible and reduces to:

$$\hat{x}(\tau_0) = \frac{\psi(\tau)R_{rf}^{-1}(\tau_0 - \tau_0)\Lambda_H^H(\tau_0)c_0(\tau_0)}{\alpha M}$$

where $\alpha = c_0(\tau_0)c_0(\tau_0)$ is constant for the duration of a data block.

The same calculations can be performed for the other paths, and the results combined with some weighting factors, called Rake combining. Theoretically, the number of visible paths will be less than or equal to the number of over samples $N_{OSV}$, therefore the maximum number of $R_{rf}^{-1}(\tau_l - \tau_0)$ matrices will be $N_{OSV}$, where the values $\tau_l - \tau_0 = \Delta \tau$ is equal to the sampling period.

Equation (14) for different path delays results in the following system of equations (15):

$$\psi(\tau)R_{rf}^{-1}(\tau_0 - \tau_0)\Lambda_H^H(\tau_0)c_0(\tau_0)M + c_0(\tau_0)\sum_{l=0}^{N_{OSV}} c_l(\tau_l)x(\tau_l)A_{ss}(\tau_l)R_{rf}(\tau_l - \tau_0)R_{rf}^{-1}(\tau_0 - \tau_0)\Lambda_H^H(\tau_0) + n(\tau)$$

(15)

$$k = 0..N_{OSV} - 1$$

III. IMPLEMENTATION

In order to simplify the writing the following notation will be used:

$$R_{rf}^{-1}(\tau_l - \tau_0) = R(l), \quad \Lambda(\tau_l) = \Lambda(l), \quad \Lambda(\tau) = \Lambda(l), \quad c_l(\tau_l) = c(l), \quad x(\tau_l) = x(k).$$

For two path delays less than a chip apart, (15) becomes:

$$\begin{bmatrix}
\psi(0)R^{-1}(0)\Lambda^H(0) = c_0x(0)\Lambda(0)R(0)R^{-1}(0)\Lambda^H(0) + c_1x(1)\Lambda(1)R(1)R^{-1}(0)\Lambda^H(0) + n_1 \\
\psi(0)R^{-1}(1)\Lambda^H(1) = c_0x(0)\Lambda(0)R(0)R^{-1}(1)\Lambda^H(1) + c_1x(1)\Lambda(1)R(1)R^{-1}(1)\Lambda^H(1) + n_2
\end{bmatrix}$$

(16)

where, $\psi(0)$ is $\psi(\tau)$ aligned at $\tau_l = \tau_0$.

Ignoring the noise term, $\hat{x}(0)$ and $\hat{x}(1)$ result from solving the system of equations (16).
For $x(0)$ it follows:

\[
x(0) = \psi(0) \frac{1}{c_1} \begin{bmatrix}
    \Lambda(0)R(0)R^{-1}(0)\Lambda^H(0) & \Lambda(1)R(1)R^{-1}(0)\Lambda^H(0) \\
    \Lambda(0)R(0)R^{-1}(1)\Lambda^H(1) & \Lambda(1)R(1)R^{-1}(1)\Lambda^H(1)
\end{bmatrix}
\]

(17)

Using the well known formula:

\[
\begin{bmatrix} A & B \\ C & D \end{bmatrix} = A - CD^{-1}B
\]

the numerator of (17) becomes:

\[
\begin{bmatrix}
    \Lambda(0)R(0)R^{-1}(0)\Lambda^H(0) & \Lambda(1)R(1)R^{-1}(0)\Lambda^H(0) \\
    \Lambda(0)R(0)R^{-1}(1)\Lambda^H(1) & \Lambda(1)R(1)R^{-1}(1)\Lambda^H(1)
\end{bmatrix} = 
\]

(18)

\[
R^{-1}(0)\Lambda^H(0) - R^{-1}(1)\Lambda^H(1)\left[\Lambda(1)R(1)R^{-1}(1)\Lambda^H(1)\right]^{-1} \Lambda(1)R(1)R^{-1}(0)\Lambda^H(0) =
\]

\[
R^{-1}(0)\Lambda^H(0) - R^{-1}(1)\Lambda^H(1)\Lambda(1)R(1)R^{-1}(0)\Lambda^H(0) = N
\]

In the same way, for the denominator:

\[
\begin{bmatrix}
    \Lambda(0)R(0)R^{-1}(0)\Lambda^H(0) & \Lambda(1)R(1)R^{-1}(0)\Lambda^H(0) \\
    \Lambda(0)R(0)R^{-1}(1)\Lambda^H(1) & \Lambda(1)R(1)R^{-1}(1)\Lambda^H(1)
\end{bmatrix} = 
\]

(19)

\[
= \Lambda(0)R(0)R^{-1}(0)\Lambda^H(0) - \Lambda(0)R(0)R^{-1}(1)\Lambda^H(1)\left[\Lambda(1)R(1)R^{-1}(1)\Lambda^H(1)\right]^{-1} \Lambda(1)R(1)R^{-1}(0)\Lambda^H(0)
\]

\[
= I - \Lambda(0)R(0)R^{-1}(1)\Lambda^H(1)\Lambda(1)R(1)R^{-1}(0)\Lambda^H(0) = D
\]

From (17), (18) and (19)

\[
x(0) = \psi(0) \frac{1}{c_1} ND^{-1} = \psi(0) \frac{1}{c_1} H
\]

(20)

where, $H = ND^{-1}$.

$x(1)$ is calculated in a similar fashion.

Since all the entries in $N$ and $D$ are known, the matrix $H$ can be pre calculated and stored in the memory.

The transmitted symbols are calculated by multiplying the vector $\psi$ by the appropriate matrix $H$.
IV. RESULTS

To validate our approach we simulated two paths at one quarter chip apart, for a single user. The first path was twice as strong as the second path. We did not add noise. We used a 16 tap FIR filter with 16 bits coefficients, four samples per chip, and a spreading factor 4.

Using the above simulation conditions, without error correction, we measured a bit error rate of 0.18 on path zero and a 0.23 bit error rate on path one. Using our technique we were able to completely annul the interference using our low delay spread multi-path cancellation technique.

V. CONCLUSION

We have presented a method of reducing multipath interference for low delay spreads. Our analysis reduces to previously published results when the delay is greater than one chip. When the delay is less than one chip our technique can cancel interference between multiple paths.

VI. REFERENCES


