*This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.*

# Non-Data Aided Symbol Timing Estimation in MIMO Systems

Ketan Rajawat and Ajit K Chaturvedi
Department of Electrical Engineering
Indian Institute of Technology Kanpur
Kanpur, UP India 208016
Email: {ketanr, akc}@iitk.ac.in

***Abstract*—We present two Maximum Likelihood (ML) based estimators for Non-Data-Aided (NDA) symbol timing recovery in MIMO systems. These estimators are based on the classical Unconditional ML and the Stochastic ML (SML) methods. The proposed estimators utilize information about the particular space-time code used and give performance comparable to data aided estimators, though for a relatively higher complexity. An approximate version of the SML estimator which requires lower implementation complexity is also presented. The loss in SNR due to timing estimation error is also analyzed.**

## I. INTRODUCTION

Wireless communication systems employing multiple antennas promise vastly increased data rates compared to single antenna systems. However, in order to fully utilize the potential of these Multiple input/Multiple output (MIMO) systems, estimation of symbol timing is of crucial importance. Symbol timing estimation may be done either through Data Aided (DA) or Non-Data Aided (NDA) methods. The DA methods require training sequences to be sent from the transmit antennas, to aid the receiver in estimating the timing error. While DA algorithms achieve good performance, transmission of training sequences contributes to overheads and reduces overall data rate. Further, the receiver needs to know the starting point of the training sequences and hence frame synchronization is required even before symbol timing can be estimated, thus further complicating the receiver.

NDA algorithms work by extracting the timing estimate from the received signal without using any training sequence. Since NDA methods usually make use of second order statistics, they require longer observation lengths and are computationally intensive. However they provide several benefits, thus data rate is not compromised and the need for frame synchronization at the physical layer is obviated. Further, if the receiver is resourceful (eg. a base station), NDA methods allow us to trade-off receiver complexity with the performance of estimator simply by changing the observation length and without compromising data rate, unlike DA estimators. These factors have contributed to the popularity of NDA schemes and there has been considerable research for NDA methods in single antenna systems [1, and references therein]. However, in MIMO systems research effort is mainly focused on DA methods as in [2], [3], which rely upon orthogonal training sequences. The NDA problem is more difficult because the received signals are not orthogonal.

An NDA timing estimator for MIMO systems was first proposed in [4]. It employed a Conditional ML based method and was developed for a general case assuming no prior information about the space time code used. In this paper, we show that for a class of space time codes known as the Linear Dispersive Codes, it is possible to separate out the code matrix from the unknown parameters and hence obtain better performance. Since popular Orthogonal Space Time Block Codes (OSTBC) and Vertical Bell Labs Space-Time (V-BLAST), among many others, are simply special cases of LDC codes, the proposed schemes are applicable to a variety of scenarios. These schemes are based on the classical Unconditional Maximum Likelihood (UML) and Stochastic Maximum Likelihood (SML) estimation techniques. We show that, as expected, the proposed SML estimator performs better than the Conditional ML (CML) estimator of [4].

We also present a low complexity approximation to the SML algorithm. Finally we derive an approximate but simple expression for the loss in SNR due to timing errors and present guidelines for choosing the observation length given the SNR loss that can be tolerated.

The remainder of the paper is organized as follows: Section II presents the notations and the system model used. Section III presents the low-SNR UML and SML estimators. Section IV gives the Cramer Rao Bound and the approximate version of SML is presented in Section V. We analyze the SNR loss due to timing error in Section VI. Section VII gives the simulation results and a discussion of the trade-off parameters involved in the estimation process. A comparison with ML DA estimator is also provided. Finally we conclude in Section VIII.

## II. SYSTEM MODEL

We consider a Space Time Coded Modulation based modem containing N transmit and M receive antennas. Assuming a quasi-static (block fading) channel, the received signal at each of the M receivers is given by

$$r_j(t) = \sqrt{\frac{E_s}{N}} \sum_{i=1}^{N} h_{ij} \sum_{n} c_i(n) p(t - nT - \epsilon_0 T) + \eta_j(t) \quad (1)$$

where $j = 1, 2 \ldots M$; $E_s/N$ is the symbol energy; T is the symbol period; $h_{ij}$s are independent channel gains corresponding to the channel between $i$th transmit and $j$th receive antenna; $\epsilon_0$ is the timing error; $p(t)$ is transmit filter pulse (eg. square root raised cosine pulse or other improved Nyquist pulses like [5] and [6]) with bandwidth $(1+\alpha)/T$, $\alpha$ being the excess bandwidth factor; $\eta_j(t)$ is additive white Gaussian noise of power $N_0$; $c_i(n)$ are space time coded symbols sent

**1-4244-0353-7/07/$25.00 ©2007 IEEE**



Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on September 30, 2009 at 03:28 from IEEE Xplore. Restrictions apply.

from the $i$th antenna. This signal when oversampled at a rate of Q samples per symbol gives,

$$\mathbf{r}_j(m) = \sqrt{\frac{E_s}{N}} \sum_{i=1}^{N} h_{ij} \sum_{n} c_i(n) p(mT/Q - nT - \epsilon_0 T) + \eta_j(m) \quad (2)$$

where $\mathbf{r}_j(m)$ is the $m$th sample at $j$th receive antenna. We assume an observation period of $L_0 R$ symbols and a guard band of $L_g R$ symbols. Then (2) can be arranged in a matrix format as follows [4]:

$$\mathbf{r}_j = \sqrt{\frac{E_s}{N}} \mathbf{P}_{\epsilon_0} \mathbf{X} \mathbf{h}_j + \eta_{\mathbf{j}} \quad (3)$$

where

$$\begin{aligned}
\mathbf{r}_j &= \begin{bmatrix} r_j(0) & r_j(1) & \dots & r_j(L_0RQ-1) \end{bmatrix}^T \\
[\mathbf{P}_{\epsilon_0}]_{mn} &= p(mT/Q - (n - L_gR)T - \epsilon_0 T) \\
\mathbf{h}_j &= \begin{bmatrix} h_{1j} & h_{2j} & \dots & h_{Nj} \end{bmatrix}^T \\
[\mathbf{X}]_{in} &= c_i(n) \\
\eta_j &= \begin{bmatrix} \eta_j(0) & \eta_j(1) & \dots & \eta_j(L_0RQ-1) \end{bmatrix}^T
\end{aligned}$$

with $m = 0, 1 \dots L_0RQ-1$, $n = 0, 1 \dots (L_0+2L_g)R$ and $j = 1, 2 \dots M$. The notation $[\mathbf{A}]_{mn}$ refers to the $(m, n)$th element of the matrix $\mathbf{A}$, $\mathbf{A}^T$ denotes the transpose while $\mathbf{A}^H$ denotes the conjugate transpose. For a rate $K/R$ Linear Dispersive Code (LDC) [7], the matrix X can be further partitioned as

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{-L_g} & \mathbf{X}_{-L_g+1} & \dots & \mathbf{X}_{L_0+L_g-1} \end{bmatrix}^T \quad (4)$$

where each $R \times N$ sub matrix $X_n$ is a LDC satisfying

$$\mathbf{X}_n = \sum_{k=1}^{K} \mathbf{C}_k s_{nK+k} + \mathbf{C}_{k+K} s^*_{nK+k} \quad (5)$$

for each $n = -L_g,\ -L_g+1\ \dots\ L_0+L_g-1$. Straightforward manipulations lead to

$$\mathbf{r}_j = \mathbf{A}_{\epsilon_0} \theta_j + \eta_j \quad (6)$$

where

$$\begin{aligned}
\mathbf{A}_{\epsilon_0} &= \sqrt{E_s/N} \mathbf{P}_{\epsilon_0} (\mathbf{I}_{L_0+2L_g} \otimes \mathbf{C}) \\
[\mathbf{C}]_{ij} &= \begin{bmatrix} [C_1]_{ij} & [C_2]_{ij} & \dots & [C_{2K}]_{ij} \end{bmatrix} \\
\theta_j &= \begin{bmatrix} I_N \otimes \mathbf{S}^T_{-L_g} & I_N \otimes \mathbf{S}^T_{-L_g+1} & \dots & \mathbf{S}^T_{L_0+L_g-1} \end{bmatrix}^T \mathbf{h}_j \\
\mathbf{S}_n &= \begin{bmatrix} s_{nK+1} & \dots & s_{nK+K} & s^*_{nK+1} & \dots & s^*_{nK+K} \end{bmatrix}^T
\end{aligned}$$

where $\otimes$ stands for the kronecker delta product. We have thus been able to present our system as a linear model. The next section describes two approaches to estimate the timing error from this model. Henceforth we will drop the subscript from $\mathbf{A}$ and $\mathbf{P}$ for the sake of clarity.

## III. Symbol Timing Estimation

We aim to estimate $\epsilon_0$ from (6). The joint maximum likelihood function of $\epsilon$ and $\theta_j$, for each receive antenna $j$, can be formulated as

$$\Lambda(\mathbf{r}_j|\epsilon, \theta_j) = C \exp(-\frac{1}{N_0} \|\mathbf{r}_j - \mathbf{A}\theta_j\|^2) \quad (7)$$

whose minimization leads to joint estimation of $\theta_j$ and $\epsilon$ ($C$ is a constant irrelevant to the estimation). For the purposes of timing estimation however, $\theta_j$ is simply a nuisance parameter. The classical maximum likelihood (also called the Unconditional Maximum likelihood or UML) approach considers $\theta_j$ as random. The Unconditional likelihood function is thus obtained by computing the marginal likelihood function of wanted parameters,

$$\Lambda_{UML}(\mathbf{r}_j|\epsilon) = E_{\theta_j} \Lambda(\mathbf{r}_j|\epsilon, \theta_j) \quad (8)$$

where the expectation is with respect to $\theta_j$ only. As in our case, this expectation operation usually poses insuperable difficulties and leads to a complicated cost function. A usual approach to solve this problem is to assume low SNR at the receiver. The likelihood function then becomes [8],

$$\begin{aligned}
\Lambda_{UML}(\mathbf{r}_j|\epsilon, \theta_j) \cong C_a &+ \frac{1}{N_0}\left(2\Re\{\theta_j^H \mathbf{A}^T \mathbf{r}_j\} - \theta_j^H \mathbf{A}^T \mathbf{A}^T \theta_j\right) \\
&+ \frac{1}{2N_0^2}\left(2\Re\{\theta_j^H \mathbf{A}^T \mathbf{r}_j\} - \theta_j^H \mathbf{A}^T \mathbf{A}^T \theta_j\right)^2
\end{aligned} \quad (9)$$

As shown in the Appendix, the marginal likelihood function becomes,

$$\Lambda_{UML}(\epsilon) = \sum_{j=1}^{M} \mathbf{r}_j^H \mathbf{P} (\mathbf{I}_{L_0+2L_g} \otimes \mathbf{C}\mathbf{C}^T) \mathbf{P}^T \mathbf{r}_j \quad (10)$$

The estimate $\hat{\epsilon}_0$ is given by

$$\hat{\epsilon}_0 = arg \max_{0 \le \epsilon < 1} \Lambda_{UML}(\epsilon) \quad (11)$$

As shown later in the simulations, the main limitation of the low-SNR UML is the impact of approximations, especially at high SNRs. The SML approach suggests the following likelihood function

$$\Lambda_{SML}(\mathbf{r}_j|\epsilon) = C \exp(-\frac{1}{N_0} \left\| \mathbf{r}_j - \mathbf{A}\hat{\theta}_j \right\|^2) \quad (12)$$

where $\hat{\theta}_j$ is itself a linear estimate of $\theta_j$. The best linear estimate for general $\theta_j$ is given by the Linear Minimum Means Square Error (LMMSE) estimator [9],

$$\hat{\theta}_j = E(\theta_j) + \mathbf{C}_{\theta_j} \mathbf{A}^T (\mathbf{A}\mathbf{C}_{\theta_j}\mathbf{A}^T + \mathbf{C}_{\eta_j})^{-1} (\mathbf{r}_j - \mathbf{A}E(\theta_j)) \quad (13)$$

where $C_{\theta_j}$ and $C_{\eta_j}$ are covariance matrices of $\theta_j$ and $\eta_j$ respectively. For this paper we make the following assumptions about the complex symbols $s_i$ (which constitute $\theta_j$ as defined in (6)), $E(s_i) = E(s_i^2) = E(s_i^{*2}) = 0$ and $E(|s_i|^2) = 1$, although extensions to other schemes is straightforward. Therefore for our case, $E(\theta_j) = 0$, $\mathbf{C}_{\theta_j} = \mathbf{I}$ and $\mathbf{C}_{\eta_j} = N_0\mathbf{I}$ (see Appendix).

The LMMSE therefore simplifies to

$$\hat{\theta}_j = \mathbf{A}^H (\mathbf{A}\mathbf{A}^H + N_0\mathbf{I})^{-1} \mathbf{r}_j \quad (14)$$

We may now estimate $\epsilon$ in a least square sense. The log-likelihood function for $\epsilon$ becomes

$$\Lambda_{SML}(\mathbf{r}_j|\epsilon) = (\mathbf{r}_j - \mathbf{A}\hat{\theta}_j)^H (\mathbf{r}_j - \mathbf{A}\hat{\theta}_j) \quad (15)$$



Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on September 30, 2009 at 03:28 from IEEE Xplore. Restrictions apply.

Substituting $\hat{\theta}_j$ from (14), doing some straightforward simplifications and averaging over all receive antennas we get,

$$\Lambda_{SML}(\epsilon) = \sum_{j=1}^{M} \mathbf{r}_j^H \mathbf{M}_\epsilon \mathbf{r}_j \tag{16}$$

where the real matrix $\mathbf{M}_\epsilon$ is given by,

$$\mathbf{M}_\epsilon = ((\frac{\xi}{N}\mathbf{P}(\mathbf{I}_{L_0+2L_g} \otimes \mathbf{C}\mathbf{C}^T)\mathbf{P}^T + \mathbf{I})^2)^{-1} \tag{17}$$

where $\xi = E_s/N_0$, the signal to noise ratio (SNR) and the estimated $\epsilon$ is given by

$$\hat{\epsilon}_0 = arg \min_{0 \le \epsilon < 1} \Lambda(\epsilon) \tag{18}$$

Note that the estimation of $\epsilon$ from the log likelihood function $\Lambda(\epsilon)$ requires computation of the inverse of a matrix. A more efficient method to implement the estimator is to pre compute the matrix $\mathbf{M}_\epsilon$ at $\epsilon = 0, 1/L, \ldots (L-1)/L$. This will provide us with $L$ values of $\Lambda(\epsilon)$ which can then be sinc interpolated as in [4]. The final expression becomes

$$\hat{\epsilon}_0 = 1 - \frac{1}{2\pi} arg \left\{ -\sum_{k=0}^{L-1} \Lambda_{SML}(k/L) e^{-j2\pi k/L} \right\} \tag{19}$$

where arg(.) operator gives the phase of its argument in the range $(0, 1)$. Typical values of $L$ are 4 or 8, for which the expression is simple to evaluate.

The SML estimator also requires a prior estimate of SNR. However as shown later in Figure 8 the estimator is quite robust to this estimate and a value of $\xi = 20$ dB suffices for all SNRs in the range 0-30 dB.

For typical values of $L_0 = 16$, $Lg = 2$, $Q = 2$, 2 transmit and 4 receive antennas and using Alamouti code and using the implementation suggested above with $L = 4$, the number of multiplications required is in excess of $2 \times 10^5$ for each block, both in UML as well as SML. Section 4 therefore presents an approximate implementation which reduces the complexity albeit at the cost of performance.

## IV. The Cramer Rao Bound

The SML estimator described above is based on LMMSE estimator, which is the best possible linear estimator. SML is therefore the best quadratic unbiased estimator, but is not optimal unless the vector $\theta_j$ is itself Gaussian for each $j$ (in which case, LMMSE becomes the best estimator). Therefore, as shown in [10], we can bound the performance of quadratic unbiased estimators by the Gaussian Unconditional Cramer Rao Bound (UCRB).

Stacking the different vectors by defining $\mathbf{r} = [\mathbf{r}_1^T \mathbf{r}_2^T \ldots \mathbf{r}_M^T]^T$, $\theta = [\theta_1^T \theta_2^T \ldots \theta_M^T]^T$ and $\eta = [\eta_1^T \eta_2^T \ldots \eta_M^T]^T$, the system model becomes,

$$\mathbf{r} = (\mathbf{I}_M \otimes \mathbf{A})\theta + \eta \tag{20}$$

The expression for UCRB, when the noise power is known and the $\theta$ vector is assumed Gaussian is given by [10],

$$UCRB = \frac{1}{\Re\Big[tr\big(\mathbf{R}^{-1}\{\frac{\partial}{\partial\epsilon}\mathbf{R}\}\mathbf{R}^{-1}\{\frac{\partial}{\partial\epsilon}\mathbf{R}\}\big)\Big]} \tag{21}$$

where,

$$\mathbf{R} \triangleq E[\mathbf{r}\mathbf{r}^H] = \mathbf{I}_M \otimes \mathbf{A}\mathbf{A}^H + N_0\mathbf{I} \tag{22}$$

$$\frac{\partial}{\partial\epsilon}\mathbf{R} = \mathbf{I} \otimes (\mathbf{A}\mathbf{D}^H + \mathbf{D}\mathbf{A}^H) \tag{23}$$

where $\mathbf{D} = \frac{\partial}{\partial\epsilon}\mathbf{A}$.

## V. An Approximate Implementation of UML and SML Estimators

In this section we attempt to design a low complexity estimator by making some approximations. First we note that $\Lambda(\epsilon)$ may conveniently be expressed as

$$\Lambda(\epsilon) = \sum_{j=1}^{M} \sum_{m=1}^{L_0RQ} \sum_{n=1}^{L_0RQ} r_j^*(m)[\mathbf{M}_\epsilon]_{mn} r_j(n) \tag{25}$$

for both the estimators, so that $\mathbf{M}_\epsilon = \mathbf{A}\mathbf{A}^T$ for UML and (17) for SML estimators. Now, we can make the following general observations about the central portion of $\mathbf{M}_\epsilon$ (ie. excluding the outer square of $TQ$ elements),

$$\begin{aligned} [\mathbf{M}_\epsilon]_{m,m+l} &\approx [\mathbf{M}_\epsilon]_{m+2k,m+l+2k} \\ [\mathbf{M}_\epsilon]_{m,m+l+1} &\approx [\mathbf{M}_\epsilon]_{m+2k,m+l+2k+1} \end{aligned} \tag{26}$$

for all "in range" integers $m$, $n$ and $l = -\kappa, -\kappa+1 \ldots \kappa$. We also observe that the above values become small for large values of $|l|$, so they can be approximated to be zero. These approximations suggest that we can arrange (25) as follows,

$$\Lambda(\epsilon) = \sum_{l=0}^{2\kappa} m_l \rho(1; l) + \sum_{l=0}^{2\kappa} m_{2\kappa+l+1} \rho(2; l) \tag{27}$$

where,

$$\rho(m, l) = \begin{cases} \sum\limits_{j=1}^{M} \sum\limits_{k=0}^{L_0R-1} |r_j(2k+m)|^2 & \text{if } l = 0 \\ 2\Re\{\sum\limits_{j=1}^{M} \sum\limits_{k=0}^{L_0R-\lceil \frac{l+m}{2} \rceil} r_j^*(2k+m) r_j(2k+m+l)\} & \\ & \text{if } l \neq 0 \end{cases} \tag{28}$$

and

$$m_l = \begin{cases} [\mathbf{M}_\epsilon]_{2k+1,2k+l+1} & \text{if } l \le 2\kappa+1 \\ [\mathbf{M}_\epsilon]_{2k+2,2k+l-2\kappa} & \text{if } l > 2\kappa+1 \end{cases} \tag{29}$$

where $k = \lfloor L_0R \rfloor$ in the last equation (this is done to ensure that above values lie in the central portion of matrix $\mathbf{M}_\epsilon$). These $4\kappa+2$, $\epsilon$ dependent coefficients are fixed for a particular pulse shape and a given value of $\alpha$, the roll of factor. This means that we can approximate $m_l(\epsilon)$ by polynomials of suitable degree and simply store the coefficients. It also implies that we have been able to put $\Lambda(\epsilon)$ as a polynomial in $\epsilon$. This polynomial may now be calculated over a grid of values of $\epsilon$ and sinc interpolated as described in Section II.

As shown later in simulations, a value of $\kappa = 3$ and fifth degree polynomial for $m_l$s provides good approximation to $\Lambda(\epsilon)$. For typical values given in Section III, the approximate



Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on September 30, 2009 at 03:28 from IEEE Xplore. Restrictions apply.

implementation requires less than 2000 multiplications. Compared to the matrix based implementation, the computational complexity required is small. However as shown later in Section VII, the price paid is loss in performance, especially at high SNR.

## VI. Loss in SNR Due to Timing Error

Of the various timing estimators proposed so far, we have mainly concentrated on evaluating the performance through MSE. While MSE serves our purpose for comparison, we need to address the issue of choosing the observation length of a system with given constraints. Since the observation length is proportional to the number of multiplications (i.e. the computational complexity), we want to keep it as small as possible. On the other hand, non zero MSE of the timing estimate results in a loss of SNR and choosing a large $L_0$ is the only way to make the MSE go to zero. Thus for a system operating at a particular SNR, there is a trade-off between SNR loss and observation length.

To characterize this trade-off curve, we assume a simple system employing V-BLAST scheme (so that $c_i(n)$ in (1) are independent). Further we assume $E[c_i(n)] = 0$ and $E[c_i(n)^2] = 1$ for the sake of simplicity. More general cases are also expected to have a similar pattern of behavior though specific values may differ. After estimation of $\epsilon_0$, the matched filtered and sampled version of the received signal is reconstructed from the samples in (2),

$$r_j(m) = \sqrt{\frac{E_s}{N}} \sum_{i=1}^{N} h_{ij} \sum_n c_i(n) g(mT - nT) + n_j(m) \quad (30)$$

where $g(t) = p(t) \otimes p(t)$ and $n_j(m) = \eta_j(t) \otimes p(t)|_{t=mT}$ (convolution is denoted by $\otimes$). Note that this reconstruction is equivalent to sampling the matched filtered version of received signal (1) at $t = mT + \epsilon_0$. Conventionally we choose $p(t)$ to be an ISI free pulse so that $p(mT - nT) = 0 \ \forall m \neq n$ and $p(0) = 1$. Thus the reconstructed signal becomes,

$$r_j(m) = \sqrt{\frac{E_s}{N}} \sum_{i=1}^{N} h_{ij} c_i(m) + n_j(m) \quad (31)$$

The SNR of the received signal becomes,

$$\rho = \frac{E_s}{N} \frac{E\left[\left|\sum_{i=1}^{N} h_{ij} c_i(m)\right|^2\right]}{E[n_j(m)^2]} = \frac{E_s}{N_0} \quad (32)$$

where we have used $E[|h_{ij}|^2] = 1$. Now let us repeat the same exercise assuming the reconstructed samples used a value of $\hat{\epsilon_0}$ instead of $\epsilon_0$. In this case the signal will consist of a nonzero interference part which will contribute to the Signal-to-Interference-plus-noise ratio (SINR). The received samples are,

$$\begin{aligned} r_j(m) = \sqrt{\frac{E_s}{N}} \sum_{i=1}^{N} h_{ij} \big[ & c_i(m-1) g(1 - \Delta\epsilon) + c_i(m) g(\Delta\epsilon) \\ & + c_i(m+1) g(1 + \Delta\epsilon) \big] + n_j(m) \quad (33) \end{aligned}$$

where $\Delta\epsilon = \hat{\epsilon_0} - \epsilon_0$ and we have neglected the ISI contribution from all but the neighboring two symbols. The SINR $\rho_e$ becomes as shown in (24) at the bottom of this page. When the error $\Delta\epsilon$ is small, we can assume it to be uniformly distributed in the interval $[-\epsilon_l, \epsilon_l]$ for some $\epsilon_l << 1$. If we make the following two approximation,

$$\begin{aligned} g(\Delta\epsilon) &\approx 1 - a\Delta\epsilon^2 + O[\Delta\epsilon^3] \\ g(1 - \Delta\epsilon) &\approx b\Delta\epsilon + O[\Delta\epsilon^2] \quad (34) \end{aligned}$$

where $a = \frac{1}{6}(\pi^2 + 3\alpha^2(\pi^2 - 8))$ and $b = \frac{\cos(\alpha\pi)}{1 - 4\alpha^2}$ from the Taylor Series expansion for a raised cosine pulse. The approximate value of $E[\rho_e]$ may be obtained as,

$$\begin{aligned} E[\rho_e] &\approx \frac{\rho}{2\epsilon_l} \int_{-\epsilon_l}^{\epsilon_l} \frac{1 + 2ax^2}{1 + 2\rho b^2 x^2} dx \\ &\approx \rho - \frac{2}{3} \rho (a + \rho b^2) \epsilon_l^2 + O[\epsilon_l^3] \quad (35) \end{aligned}$$

Also, since the $MSE = E[\Delta\epsilon^2] = \epsilon_l^2/3$, we get,

$$\rho_e \approx \rho \left\{ 1 - 2(MSE)(a + \rho b^2) \right\} \quad (36)$$

Thus the loss in SNR in dB is given by,

$$\begin{aligned} \Delta\rho &= 10 \log \left( \frac{\rho_e}{\rho} \right) = 10 \log \left( 1 - 2(MSE)(a + \rho b^2) \right) \\ &\approx 20 \log(e)(a + \rho b^2) \times MSE \quad (37) \end{aligned}$$

From the above relation, the following observations can be made:

1) At low SNRs when MSE is high, the loss in SNR is also high.
2) Both DA and NDA timing estimation methods which are based on some kind of approximation (as in [4], [2], [3], to reduce system complexity) show an error floor at very high SNRs. Thus $\Delta\rho$ is expected to be high at large SNRs, since the product $\rho MSE$ eventually starts increasing as $\rho \to \infty$.

$$\begin{aligned} \rho_e &= \frac{E_s}{N} \frac{E\left[\left|\sum_{i=1}^{N} h_{ij} c_i(m) g(\Delta\epsilon)\right|^2\right]}{E\left[\left|\sum_{i=1}^{N} h_{ij} \left(c_i(m-1) g(1 - \Delta\epsilon) + c_i(m+1) g(1 + \Delta\epsilon)\right)\right|^2\right] + E[n_j(m)^2]} \\ &= \frac{E_s g(\Delta\epsilon)^2}{E_s(g(1 - \Delta\epsilon)^2 + g(1 + \Delta\epsilon)^2) + N_0} \quad (24) \end{aligned}$$



Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on September 30, 2009 at 03:28 from IEEE Xplore. Restrictions apply.

3) There must be an optimal value of SNR when the loss in minimum. In fact this point is observed to occur much before than the error floor in MSE.

In Fig. 1, where the MSEs are obtained through simulations, we illustrate the above points for SML, UML and CML methods. From the graph we can see that the SNR loss becomes minimum at some intermediate SNRs and starts to increase with higher SNR, especially for UML for which the error floor is high.

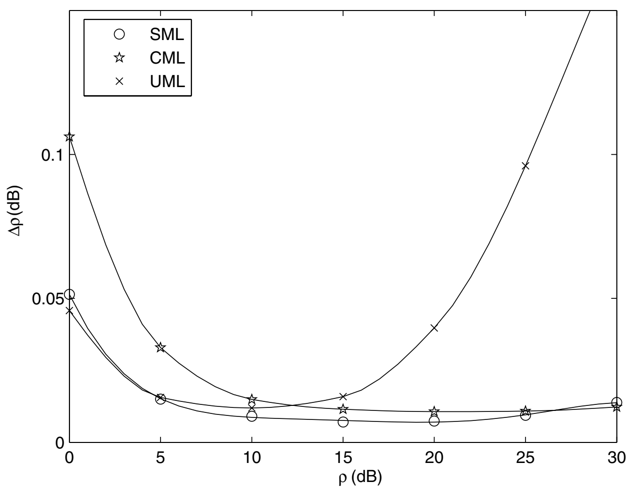


Fig. 1. Loss in SNR($\Delta\rho$) against SNR($\rho$) for SML, CML and UML methods.

## VII. Simulation Results and Discussion

We provide Monte Carlo simulations to determine the performance of the proposed NDA methods. We also compare SML NDA with ML DA method. In all the simulations, we use the mean square error (MSE) ie. $E(\hat{\epsilon}-\epsilon)^2$ as the performance measure. We assume QPSK modulated iid symbols $s_i$. The channel coefficients $h_{ij}$s are taken to be independent complex Gaussian distributed with mean zero and variance $0.5$ along each dimension. The elements of the noise vector $\eta_j$ are also assumed to be iid complex Gaussian with mean zero and variance $N_0/2$ in each dimension. Unless stated otherwise, $N = 2$, $M = 4$, $L_0 = 16$, $L_g = 2$ and Alamouti coding scheme (ie. $K = 2$ and $T = 2$) has been assumed.

Figure 2 shows the comparison of CML, UML and SML estimators for $L_0 = 16$ i.e. 32 symbols. We also show the low complexity version of SML estimator as described in Section V, for $\kappa = 3$ and using a fifth degree polynomial for approximation. Notice the better performance of SML at low and moderate SNRs and that of UML at low SNRs than the CML method of [4]. This is mainly because the proposed estimators utilize the information about the space time code used. The performance of UML and SML is almost same at low SNRs, while UML degrades rapidly at high SNRs. This is because at high SNRs the approximation in (9) does not hold. Also, the performance of the approximate SML is almost as good as SML but it develops an error floor with increasing SNR, mainly because of the approximation errors.

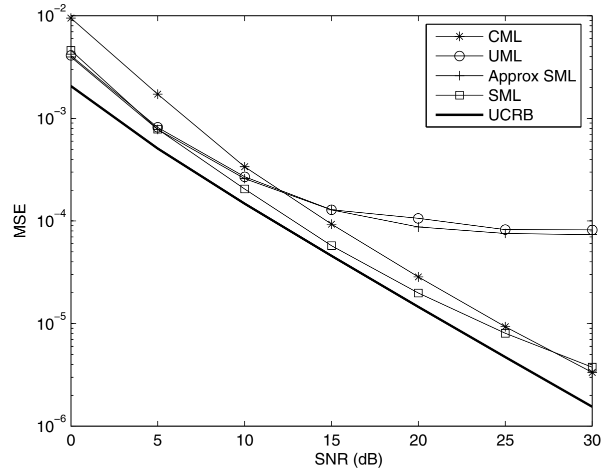


Fig. 2. Comparison of various schemes for $N = 2$, $M = 4$, $L_0 = 16$ and $Q = 2$ employing Alamouti coding scheme.

In Figure 3, we compare ML DA [4] and SML NDA algorithms. The DA scheme has been simulated for 16 training symbols while the SML is simulated for $L_0 = 32$. Both the schemes use a $2 \times 4$ system. We see that the proposed NDA scheme matches the DA scheme of [4] without any loss in data rate, although the complexity of NDA is approximately four-fold that of the DA scheme.

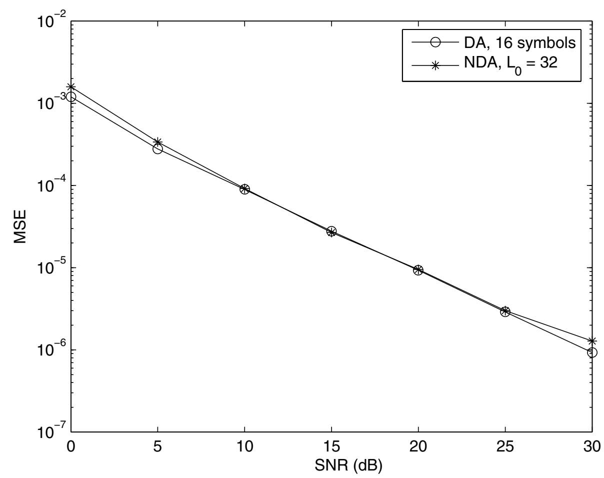


Fig. 3. Comparison of DA and NDA schemes $N = 2$, $M = 4$, $L_0 = 16$ for DA (CML of [4]) and $L_0 = 32$ for NDA (proposed SML method) and $Q = 2$ employing Alamouti coding scheme.

Figure 4 shows the performance of SML estimator against variation in observation lengths. We see that the performance of the estimator can be improved by increasing $L_0$. This can be handy for obtaining acceptable performance in low SNR scenarios. From the graph we can also observe the approximate 3dB fall in MSE each time the value of $L_0$ is doubled. This



Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on September 30, 2009 at 03:28 from IEEE Xplore. Restrictions apply.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

is a typical behavior of all ML based estimators whose error variance is usually inversely proportional to the observation interval $L_0$.

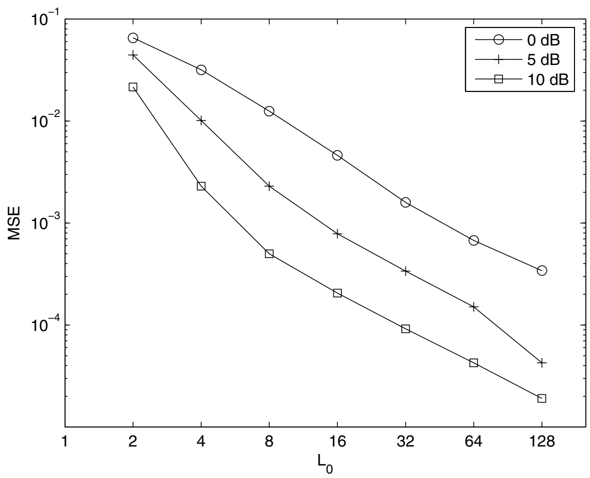


Fig. 4. Varying performance based on the observation length $L_0$ for $N = 2$, $M = 4$ and $Q = 2$ employing Alamouti coding scheme.

If we consider operation at a particular SNR, using Figure 4 and (37) we can draw a curve illustrating the variation of $\Delta\rho$ with $L_0$. Fig 5 shows these curves for SML at 0 dB, 5 dB and 10 dB. As expected, the loss in SNR becomes negligible as we increase $L_0$ beyond a certain point (depending on the operating SNR). Also note that this transition becomes sharper with increase in SNR because of the decrease in MSE.

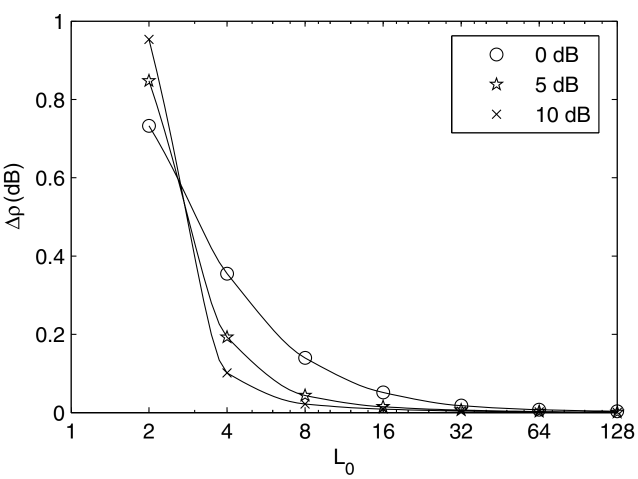


Fig. 5. Trade-off between $L_0$ and $\Delta\rho$ at 0 dB, 5 dB and 10 dB for SML.

Figure 6 shows comparison of SML scheme for varying number of receive antennas. The estimator's performance quickly degrades with decrease in the number of receive antennas. This is expected since the overall likelihood function is the sum of individual likelihood functions of various receive antennas, so reducing number of antennas is very similar to reducing the observation length.

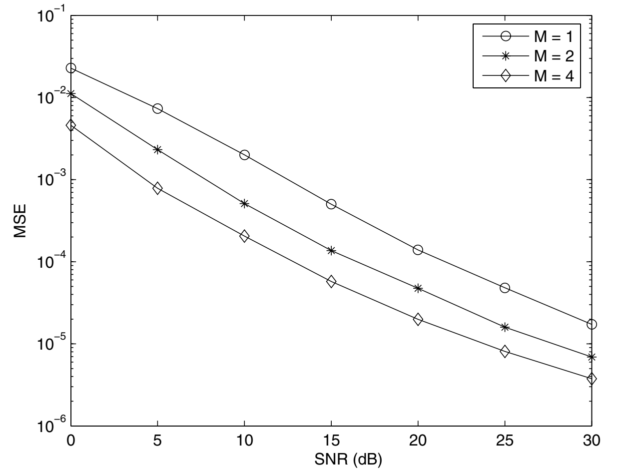


Fig. 6. Comparison of SML for varying number of receive antennas for $N = 2$, $L_0 = 16$ and $Q = 2$ employing Alamouti coding scheme.

Figure 7 shows the approximate SML estimator for varying degrees of approximation. From the graph, it can be seen that choosing $\kappa = 2$ and a fifth degree polynomial is good enough and there is no need to go for polynomials of higher degree.

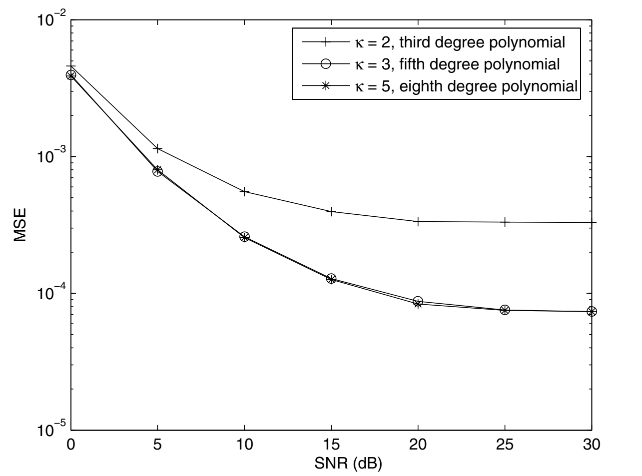


Fig. 7. Comparison of SML for varying degrees of approximation by varying $\kappa$ and the degree of the polynomial used for approximation for $N = 2$, $L_0 = 16$ and $Q = 2$ employing Alamouti coding scheme.

Figure 8 analyzes the sensitivity of SML method to the accuracy in the estimate of $\xi$, the estimated SNR. We plot the MSE of SML for the true value of $\xi$ and for the cases when $\xi$ is taken to be constant at 10 dB, 20 dB and 30 dB regardless of the true value of $\xi$. We see that the curves overlap and therefore SML is robust to inaccuracies in the estimate of $\xi$ used in (17). Thus we have used $\xi = 20$ dB for the range of SNRs considered.



Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on September 30, 2009 at 03:28 from IEEE Xplore. Restrictions apply.

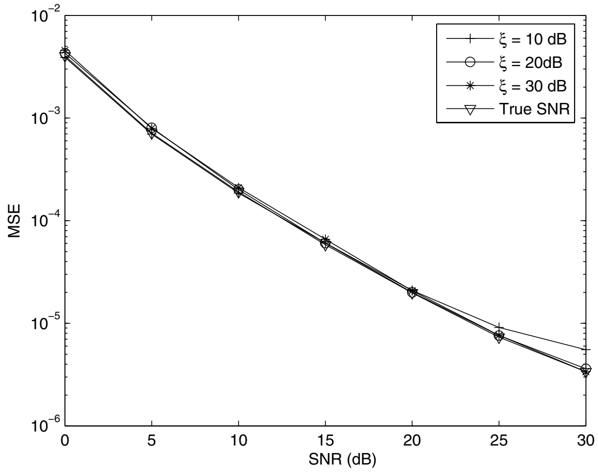


Fig. 8. Sensitivity of SML to estimated SNR for $N = 2$, $M = 4$, $L_0 = 16$ and $Q = 2$ employing Alamouti coding scheme.

## VIII. Conclusion

Two Non-Data Aided symbol timing estimators based on the classical Unconditional Maximum Likelihood (UML) and Stochastic ML (SML) have been proposed. By carefully arranging the system model in a matrix form, we have been able to utilize the information about the type of code used and hence obtain better performance than the existing CML method. The Cramer Rao Bound and an approximate version of the SML estimator have also been derived. Simulations show that at the expense of higher computational complexity, NDA schemes can perform as well as DA schemes. An approximate expression for the loss in SNR because of timing error has been derived and a trade-off curve between observation length and the loss in SNR is provided.

## Appendix

This section provides derivation of (10) assuming the approximation given in (9). First we note the structure of $\theta_j$,

$$[\theta_j]_{2mKN+2iK+n} = \begin{cases} s_{(m-L_g)K+n}h_{(i+1)j} & \text{if } 1 \leq n \leq K \\ s^*_{(m-L_g)K+n}h_{(i+1)j} & \text{if } K+1 \leq n \leq 2K \end{cases} \tag{38}$$

where $-L_g \leq m \leq L_0 + L_g - 1$ and $0 \leq i \leq N-1$. This means that $E(\theta_j) = 0$ and $\mathbf{C}_{\theta_j} = \mathbf{I}$. For evaluating $\Lambda(\mathbf{r}_j|\epsilon) = E[\Lambda(\mathbf{r}_j|\epsilon, \theta_j)]$, we consider the each of the terms in (9),

$$E[\Re(\theta_j^H \mathbf{A}^T \mathbf{r}_j)] = 0 \tag{39}$$

since $E(\theta_j) = 0$. Also using the identity $E(\mathbf{x}^H \mathbf{A} \mathbf{x}) = tr(\mathbf{A}\mathbf{C}_x)$ where $tr(.)$ denotes the trace operator,

$$E[\theta_j^H \mathbf{A}^T \mathbf{A}^T \theta_j] = tr(\mathbf{A}^T \mathbf{A}) \tag{40}$$

Also note that $E([\theta_j]_m[\theta_j]_n) = 0$, $\forall m, n$. It can therefore be easily verified by expanding and using the quadratic identity stated above that,

$$E[\Re^2(\theta_j^H \mathbf{A}^T \mathbf{r}_j)] = \frac{1}{2}(\mathbf{r}_j^H \mathbf{A}\mathbf{A}^T \mathbf{r}_j) \tag{41}$$

As shown in [9, Chap. 15],

$$E[\theta_j^H \mathbf{A}^T \mathbf{A} \theta_j \theta_j^H \mathbf{A}^T \mathbf{A} \theta_j] = tr(\mathbf{A}^T \mathbf{A})^2 + tr(\mathbf{A}^T \mathbf{A}\mathbf{A}^T \mathbf{A}) \tag{42}$$

Lastly since $h_{ij}$s are all Gaussian, third order moment is zero, which gives,

$$E[\Re(\theta_j^H \mathbf{A}^T \mathbf{r}_j)\theta_j^H \mathbf{A}^T \mathbf{A} \theta_j] = 0 \tag{43}$$

It can be seen through numerical computations that $tr(\mathbf{A}^T \mathbf{A})$ and $tr(\mathbf{A}^T \mathbf{A}\mathbf{A}^T \mathbf{A})$ are both independent of $\epsilon$. Plugging back the terms in (9) and ignoring these terms, we get

$$\Lambda(\mathbf{r}_j|\epsilon) = \mathbf{r}_j^H \mathbf{A}\mathbf{A}^T \mathbf{r}_j \tag{44}$$

Further, we may remove the multiplicative term $\xi$ from the likelihood function as it does not effect the maximization :

$$\Lambda(\mathbf{r}_j|\epsilon) = \mathbf{r}_j^H \mathbf{P}(\mathbf{I}_{L_0+2L_g} \otimes \mathbf{C}\mathbf{C}^T)\mathbf{P}^T \mathbf{r}_j \tag{45}$$

Finally the $\Lambda(\mathbf{r}_j|\epsilon)$ is averaged over all receive antennas to give the overall likelihood function in (10).

## References


[1] U. Mengali and A. N. D'Andrea, *"Synchronization Techniques for Digital Receivers"* , Plenium Publishers, 1997. 233 Spring Street, New York, N.Y. 10013.

[2] Yik-Chung Wu, S. C. Chan and Erchin Serpedin, *"Symbol-Timing Estimation in SpaceTime Coding Systems Based on Orthogonal Training Sequences"* , IEEE Trans. on Wireless Commun., Vol. 4, No. 2, pp. 603-13, Mar. 2005.

[3] K. Rajawat and A. K. Chaturvedi, *"A low complexity symbol timing estimator for MIMO systems using two samples per symbol"* , IEEE Commun. Lett., Vol. 10, No. 7, pp. 525-27, July. 2006.

[4] Yik-Chung Wu and E. Serpedin, *"Symbol Timing Estimation in MIMO Correlated Flat-Fading Channels"*, Wireless Communications and Mobile Computing (WCMC) Journal, Special Issue on "Multiple-Input Multiple-Output (MIMO) Communications", Wiley, vol. 4, Issue 7, pp. 773-790, Nov. 2004.

[5] N. C. Beaulieu, C. C. Tan, and M. O. Damen, *"A "better than" Nyquist pulse",* IEEE Commun. Lett., vol. 5, pp. 367-68, Sept. 2001.

[6] Antonio Assalini and Andrea M. Tonello, *"Improved Nyquist Pulses"* , IEEE Commun. Lett., Vol. 8, No. 2, pp. 87-89, Feb. 2004.

[7] Babak Hassibi and Bertrand M. Hochwald, *"High-Rate Codes That Are Linear in Space and Time"* , IEEE Info. Theory., Vol. 48, No. 7, pp. 1804-24, July. 2002.

[8] G. Vzquez, J. Riba, G. B. Giannakis, Y. Hua, P. Stoica, and L. Tong, *"Non-data-aided digital synchronization,"* in *Signal Process. Adv. Wireless Mobile Commun. Vol. 2, Trends in Single-and Multi-User Systems, July 2000.*

[9] Steven M. Kay, *"Fundamentals of Statistical Signal Processing: Estimation Theory"* PTR Prentice-Hall 1993.

[10] B. Ottersten, M. Viberg P. Stoica, *"Radar Array Processing"* , Springer-Verlag, 1993. Chapter: Exact and Large Sample Maximum Likelihood Techniques for Parameter Estimation and Detection.




Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on September 30, 2009 at 03:28 from IEEE Xplore. Restrictions apply.