

A SOFTWARE DEFINED RADIO TESTBED FOR MIMO SYSTEMS

David Palchak (University of Utah: Salt Lake City, UT, USA, palchak@ece.utah.edu);
Behrouz Farhang-Boroujeny (University of Utah: Salt Lake City, UT, USA, farhang@ece.utah.edu)

ABSTRACT

Multiple-input multiple-output communications promise to increase system capacities over traditional SISO systems and is an active area of research. In this paper we present a design and implementation results for a 4×4 MIMO testbed based on the GNU Radio software package and associated hardware. GNU Radio provides a full featured platform for developing quasi-realtime Software Defined Radio (SDR) algorithms. We propose methods for carrier recovery, timing recovery, channel estimation and noise variance estimation suitable for software implementation. Most of these methods rely on a specifically designed packet structure that combines SISO and MIMO formatting. For the channel estimation we suggest an optimized technique that relies on Golay Complementary Sequences. The performance of each component of the testbed is defined according to the project objectives and evaluated analytically.

1. INTRODUCTION

Multiple-input multiple-output (MIMO) communication refers to a system utilizing $N_t > 1$ transmit antennas and $N_r > 1$ receive antennas [1]. In a spatially multiplexed MIMO system a transmit stream is split into N_t components, each of which is transmitted on one antenna simultaneously. Each parallel transmission utilizes the same portion of the spectrum, thus increasing the bandwidth efficiency. When joint detection is performed by the receiver the increased efficiency translates into increased capacity. However, optimal joint detection methods [2] have exponential complexity with N_t , rendering them impractical in most circumstances. A number of sub-optimal methods have been proposed, and this is still an active area of research [3][4][5][6].

Most of the literature addressing MIMO detection relies on a block-fading channel model that can be expressed as:

$$\mathbf{y} = \mathbf{H}\mathbf{d} + \nu \quad (1)$$

where \mathbf{y} represents the $N_r \times 1$ received symbol vector, \mathbf{H} represents the $N_r \times N_t$ channel gain matrix, \mathbf{d} denotes the $N_t \times 1$ transmitted symbol vector, and ν corresponds to Additive White Gaussian Noise (AWGN) satisfying the property $E[\nu^H \nu] = \sigma_\nu^2 \mathbf{I}$. Systems modeled using (1) assume that the received symbols are perfectly timed and free from carrier frequency offsets. Furthermore, nearly all of the popular MIMO detectors

assume that the receiver has some knowledge or estimate of the channel matrix \mathbf{H} and noise variance σ_ν^2 .

The goal of the project described in this paper was to design and implement a MIMO system that would allow researchers to test the performance of various MIMO detectors in real channel environments. This required the design of a specific packet format and corresponding receiver structures that together satisfy the assumptions inherent in the block-fading model (1). It should be noted that this effort is not intended to produce a commercially competitive MIMO solution. While some portions of the architecture have been optimized, in general the aim is to introduce an inexpensive, viable laboratory testbed.

The rest of the document is organized as follows: Section 2 introduces the SDR platform used for the testbed and highlights important system parameters. The packet design is presented in Section 3, followed by detailed examinations of the carrier (Section 4) and timing (Section 5) synchronization methods. Section 6 introduces a reduced complexity technique for channel estimation based on Golay Complementary Sequences, and Section 7 addresses the problem of noise variance estimation. Finally, Section 8 ends with a brief conclusion.

2. SDR PLATFORM

The testbed is based around the Universal Software Radio Peripheral (USRP) hardware component offered by Ettus Designs. The USRP is an FPGA based device specifically designed for software radios. It features high-rate ADC and DAC components along with a USB2 transceiver all coupled to the FPGA. The USRP is designed to interface directly with the GNU Radio software package. The GNU Radio project provides a free and Open Source platform for developing software radio applications. It includes many of the signal processing blocks necessary to implement fully featured digital and analog transceivers. GNU Radio also provides a C++ API for the creation of custom signal processing blocks.

Two USRPs with customized FPGA code and a customized version of GNU Radio are used to form the testbed. Table 2 lists the important operational and design parameters of the system. The principle advantage offered by the USRP/GNU Radio system

in this application is the combination of low hardware cost and conceptual simplicity. However, the trade-off for these features is a relatively low data rate. This is reflected in several of the performance values shown in the table.

TABLE I
SYSTEM PARAMETERS

Parameter	Value
Antennas	4 Tx, 4 Rx
Modulation	BPSK,QPSK
Coding	Convolutional., rate $\frac{1}{2}$
Symbol Rate	125 kHz
Sample Rate	500 kHz
Throughput	500 kb/s

3. PACKET DESIGN

The testbed is designed to support a packet-type communications model, where signals arrive in short bursts similar to IEEE 802.11a systems. As previously mentioned, the packet format and receiver structure are closely related to facilitate the required functionality. Figure 1 shows the packet formatting, where the numbers under each field denote the length of the field in symbols.

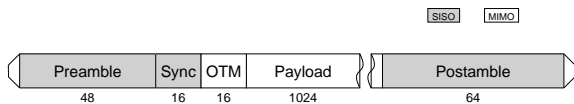


Fig. 1. Packet format

In Figure 1, the preamble and postamble fields represent sequences of alternating BPSK symbols. At the transmitter these symbols are interpolated and passed through a root-raised cosine transmit filter, producing a real-valued single frequency tone at half the symbol frequency. This tone is used for both timing recovery and noise variance estimation. The sync field represents a unique symbol sequence that is used to locate the start of the orthogonal training matrix (OTM) field. The OTM field represents a second unique symbol sequence used to estimate the channel parameters. During the SISO portions of the packet, each transmit antenna transmits identical symbols. During the MIMO portions, each antenna transmits different symbols.

4. CARRIER RECOVERY

Carrier frequency recovery is performed by frequency multiplexing a low power pilot signal with the data

signal. The decision to use a dedicated pilot, which requires additional transmit power, was made to eliminate the effects of carrier frequency estimation and tracking errors when testing various MIMO detectors. Figure 2 shows a plot of the spectrum for a signal packet. The single frequency tone, positioned at exactly 0.8π rad/sample, represents the pilot signal.

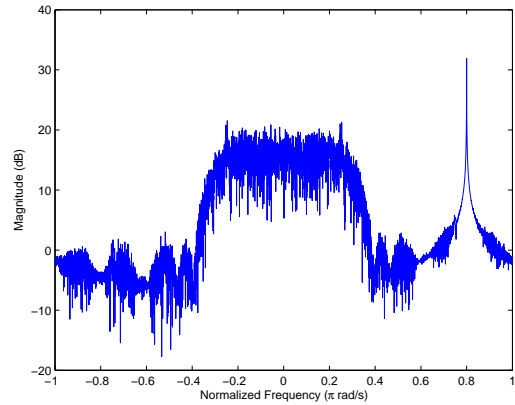


Fig. 2. Spectrum of a single packet showing frequency multiplexed carrier

As depicted in Figure 3, the pilot signal is used directly to compensate for frequency offsets. This method provides perfect frequency synchronization at the cost of noise enhancement. In practice, however, the lowpass filter applied to the pilot eliminates enough of the out-of-band noise that the final increase in noise variance is typically less than the noise variance estimation error.

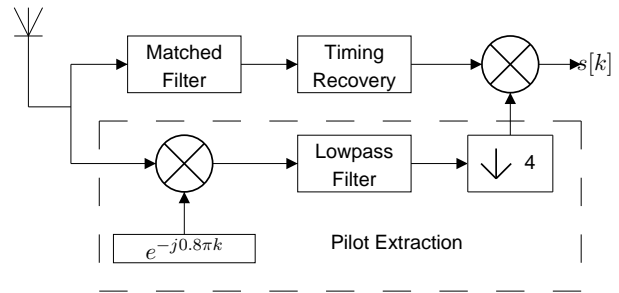


Fig. 3. Pilot recovery circuit

5. SYMBOL TIMING RECOVERY

Symbol timing recovery is handled at the receiver by applying the well-known power spectral line method [7] followed by a fractional delay FIR (FD-FIR) filter

[8]. The spectral line method computes the ideal sample offset, μ , as

$$\mu = -\frac{1}{2\pi} \angle \left(\sum_{i=1}^{N_r} \sum_{k=0}^{L-1} |r_i[k]|^2 e^{j \frac{2\pi k}{M}} \right) \quad (2)$$

where $r_i[k]$ represents the sampled signal received by the i^{th} antenna, M is the oversampling rate at the receiver, and $\angle(\cdot)$ computes the angle of its argument in radians. Figure 4 shows a graphical interpretation of μ .

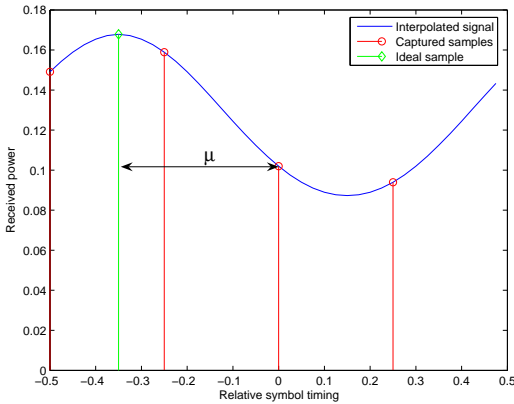


Fig. 4. Graphical representation of ideal timing parameter μ

Since the ADC sampling clock is driven by a fixed oscillator, fractional sample delays are achieved by applying a fractional delay FIR filter to the received signals. The complete timing recovery structure is shown in Figure 5. In this circuit, the ideal timing offset μ is broken down into integer and fractional sample delays, μ_I and μ_F , respectively. The fractional delay is quantized to 7 bits, allowing for fractional delays in range of $[-\frac{1}{2}, \frac{63}{64}]$, in increments of $\frac{1}{64}$.

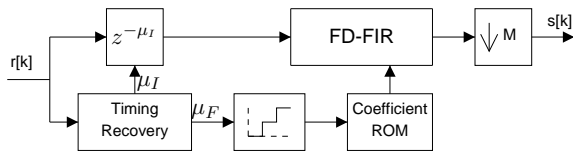


Fig. 5. Timing recovery structure

The performance of the FD-FIR is computed as the MSE of the filter output relative to a that of a perfect interpolator. This is shown in Figure 6 as a function of the fractional delay and the number of filter taps, T . It can be seen that even with a small number of taps

the FD-FIR filters offer sufficient performance over the entire delay range.

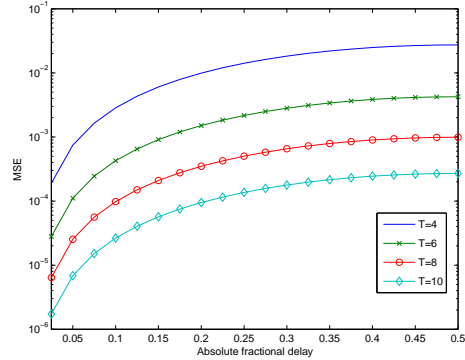


Fig. 6. Mean squared error of several fractional delay FIR filters

6. CHANNEL ESTIMATION

Estimation of the channel parameters is achieved through the use of a pilot symbol matrix, as proposed in [9]. Letting \mathbf{D} represent the pilot symbol matrix, and assuming the system can be modeled by (1), the least-squares (LS) estimate of the channel can be written as

$$\hat{\mathbf{H}} = \mathbf{Y}\mathbf{D}^H(\mathbf{D}\mathbf{D}^H)^{-1} \quad (3)$$

Recall from [10] that the MSE of the LS estimate can be expressed as

$$MSE_{\hat{\mathbf{H}}} = \frac{\sigma_n^2}{N_t} Tr[(\mathbf{D}\mathbf{D}^H)^{-1}] \quad (4)$$

where $Tr(\cdot)$ denotes the trace of its argument. It can be seen from (4) that the MSE of the LS estimate is minimized when $\mathbf{D}\mathbf{D}^H = \mathcal{E}_0\mathbf{I}$. Matrices satisfying this condition are referred to as orthogonal matrices. Returning again to equation (3) and assuming that \mathbf{D} is chosen to be orthogonal, we get

$$\hat{\mathbf{H}} = \frac{1}{\mathcal{E}_0} \mathbf{Y}\mathbf{D}^H \quad (5)$$

and

$$MSE_{\hat{\mathbf{H}}} = \frac{\sigma_v^2}{LN_t} \quad (6)$$

where L is the length of the training matrix. Equation (5) shows that when \mathbf{D} is an orthogonal matrix, the LS channel estimate can be computed using correlation.

To reduce the correlation complexity for the LS channel estimate, we construct an orthogonal training matrix (OTM) using Golay Complementary Sequences (GCS). In [9] GCS are applied to channel estimation in MIMO-OFDM systems. Golay sequences are frequently used

in OFDM systems because of their good aperiodic autocorrelation properties[11]. We choose GCS to exploit the efficient correlator structures presented in [12] and [13].

Construction of an orthogonal matrix for a system with 4 transmit antennas is achieved as follows: Define the vector $\mathbf{a} = [a_0 \ a_1 \ \dots \ a_{L-1}]$. Similarly define the vectors \mathbf{b} , \mathbf{x} , and \mathbf{y} . Let $\{\mathbf{a}, \mathbf{b}\}$ represent a polyphase Golay complementary pair (GCP) generated via the Golay-Rudin-Shapiro (GRuS) recursive technique[14]. Let $\{\mathbf{x}, \mathbf{y}\}$ also represent a polyphase GCP generated via the GRuS recursive technique using a different set of weights. An orthogonal matrix \mathbf{O} can be expressed as

$$\mathbf{O} = \begin{bmatrix} \mathbf{a} & \mathbf{x} \\ \mathbf{b} & \mathbf{y} \\ \mathbf{a} & -\mathbf{x} \\ \mathbf{b} & -\mathbf{y} \end{bmatrix} \quad (7)$$

To prove that \mathbf{O} constitutes an orthogonal matrix, we must first demonstrate that the members of a polyphase GCP generated via the GRuS recursion are orthogonal. This can be achieved by induction.

a) Base Case: Consider the GCP composed of the sequences $[w_0 \ w_1]$ and $[w_0 \ -w_1]$ where w_m is a complex number such that $|w_m|^2 = 1$. It is easy to see that these sequences are orthogonal.

b) Inductive Step: Let $\mathbf{p} = [p_0 \ p_1 \ \dots \ p_{L-1}]$ and $\mathbf{q} = [q_0 \ q_1 \ \dots \ q_{L-1}]$, where $\{\mathbf{p}, \mathbf{q}\}$ represent a GCP and $\mathbf{p}\mathbf{q}^H = 0$. Applying the GRuS technique to $\{\mathbf{p}, \mathbf{q}\}$ creates a new GCP of length $2L$:

$$\begin{aligned} \mathbf{u} &= [\mathbf{p} \quad w_i\mathbf{q}] \\ \mathbf{v} &= [\mathbf{p} \quad -w_i\mathbf{q}] \end{aligned} \quad (8)$$

Proving that \mathbf{u} and \mathbf{v} are orthogonal, we have

$$\begin{aligned} \mathbf{u}\mathbf{v}^H &= [\mathbf{p} \quad w_i\mathbf{q}] \begin{bmatrix} \mathbf{p}^H \\ -w_i^*\mathbf{q}^H \end{bmatrix} \\ &= \sum_{k=0}^{L-1} p_k p_k^* - w_i w_i^* \sum_{k=0}^{L-1} q_k q_k^* \\ &= 2L - (1)(2L) \\ &= 0 \end{aligned} \quad (9)$$

where the simplification used in the second step from the definition of Golay complementary sequences. Thus, by induction, the members of a GCP generated via the GRuS recursion are orthogonal. This also proves that \mathbf{O} is an orthogonal matrix.

For the training matrix \mathbf{O} , the computation of the LS channel estimate for a single receive antenna is efficiently computed using the structure shown in Figure 7. In this diagram, the components marked EGC-AB

and EGC-XY are Efficient Golay Correlator structures for the Golay pairs $\{\mathbf{a}, \mathbf{b}\}$ and $\{\mathbf{x}, \mathbf{y}\}$, respectively [13]. These structures exploit the recursive construction of the Golay sequences to reduce the number of required multiplications. For training sequences of length L , the circuit presented requires $4 \cdot \log_2(L)$ multiplies, in contrast to $4L$ multiplies required for standard correlation.

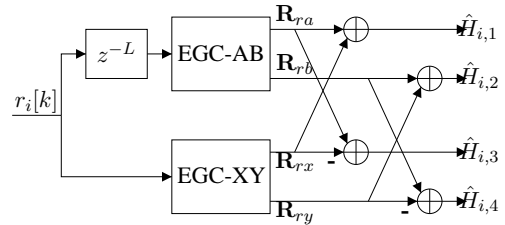


Fig. 7. Structure for efficient computation of least-squares channel estimate

7. NOISE VARIANCE ESTIMATION

The noise estimation technique used at the receiver relies on the periodicity of the packet preamble to isolate samples of the noise. Assuming perfect carrier frequency synchronization, the preamble signal can be expressed as $b[k] = A \cos(\frac{\pi k}{M} + \phi) + \nu[k]$. To extract only the noise portion of the preamble, $\nu[k]$, we delay each sample by M time units and then add it to the newest sample. This produces

$$\begin{aligned} n[k] &= b[k] + b[k - M] \\ &= A \left[\cos\left(\frac{\pi k}{M} + \phi\right) + \cos\left(\frac{\pi k}{M} - \frac{\pi M}{M} + \phi\right) \right] \\ &\quad + \nu[k] + \nu[k - M] \\ &= A \left[\cos\left(\frac{\pi k}{M} + \phi\right) - \cos\left(\frac{\pi k}{M} + \phi\right) \right] \\ &\quad + \nu[k] + \nu[k - M] \\ &= \nu[k] + \nu[k - M] \end{aligned} \quad (10)$$

Since the values $\nu[k]$ and $\nu[k - M]$ are uncorrelated, $n[k]$ is a white, Gaussian process with variance $2\sigma_\nu^2$. An unbiased estimation of σ_ν^2 is directly obtained from $n[k]$ as

$$\hat{\sigma}_\nu^2 = \frac{S^2}{2L} \quad (11)$$

where

$$S^2 = \sum_{k=0}^{L-1} n[k]n^*[k] \quad (12)$$

and L is the length of $n[k]$.

The random variable S^2 is known to have a χ^2 distribution with L degrees of freedom [15]. Let $\xi = \log(\hat{\sigma}_v^2) - \log(\sigma_v^2)$. Using a transformation of variables on S^2 , we obtain a probabilistic bound on ξ as a function of L :

$$P\{|\xi| \leq \log(\delta)\} = \frac{1}{\Gamma(\frac{L}{2})} \int_{\frac{L}{2}}^{\frac{\delta L}{2}} t^{\frac{L}{2}-1} e^{-t} dt \quad (13)$$

where $\Gamma(z)$ denotes the Gamma function and δ represents the accuracy of the estimate, subject to $\delta \geq 1$. The value computed in (13) is the probability that, given L samples of the noise, the noise variance estimation error is bounded by δ . Figure 8 shows a plot of (13) for several values of δ . From the curves it can be seen that for practical values of L , we can expect $\hat{\sigma}_v^2$ to be within 1 dB of the actual noise variance 90% of the time.

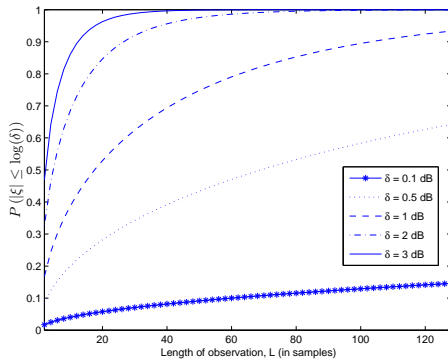


Fig. 8. Performance probabilities for noise variance estimator

8. CONCLUSION

This project provides a basic yet well-performing testbed for researchers to use in testing MIMO algorithms. Given the project objectives, performance is measured in terms of the degree to which the system is capable of characterizing and emulating the block-fading MIMO model presented as (1). Characterization performance refers to the precision of the channel and noise variance estimates. The emulation performance refers to the elimination of frequency and timing offsets.

Future work on the testbed will include refinement of each component of the receiver with the goal of supporting higher order modulations, such as multilevel QAM. This will necessitate the integration of an algorithm for tracking timing offset over the duration of a packet. Also, work is currently underway to eliminate the need for the pilot carrier. Alternative techniques being considered focus on estimating the carrier frequency offset directly using the preamble and postamble.

The relatively low cost of the USRP hardware and GNU Radio software makes the combination well suited for laboratory settings. The simple, extensible design of the GNU Radio API gives researchers the flexibility to easily reassign testbed nodes to different projects as needed. Such an idea is currently being applied to single antenna systems by the Flux Research Group at the University of Utah.

9. REFERENCES

- [1] E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Trans. Telecomm.*, vol. 10, no. 6, pp. 585–595, Nov-Dec 1999.
- [2] H. Vikalo and B. Hassibi, "Modified Fincke-Pohst algorithm for low-complexity iterative decoding over multiple antenna channels," in *Proceedings of IEEE International Symposium on Information Theory (ISIT 2002)*, 2002, p. 390.
- [3] P. W. Wolniansky, G. J. Foschini, G. D. Godlen, and R. A. Valenzuela, "V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel," in *invited paper, Proc. ISSSE-98*, Pisa, Italy, Sept. 1998.
- [4] H. Zhu, Z. Shi, and B. Farhang-Boroujeny, "MIMO detection using Markov chain Monte Carlo techniques for near capacity performance," in *Int. Conf. Acoustics, Speech and Signal Processing (ICASSP'05)*, Philadelphia, Mar. 2005.
- [5] Z. Shi, H. Zhu, and B. Farhang-Boroujeny, "Markov chain Monte Carlo techniques in iterative detectors: a novel approach based on Monte Carlo integration," in *Global Telecommunications Conference (GLOBECOM '04)*, vol. 1, no. 29, Nov. 2004, pp. 325–329.
- [6] B. Farhang-Boroujeny, H. Zhu, and Z. Shi, "Markov chain Monte Carlo algorithms for CDMA and MIMO communication systems," *IEEE Trans. Signal Processing*, accepted.
- [7] M. Oerder and H. Meyr, "Digital filter and square timing recovery," *Communications, IEEE Transactions on*, vol. 36, no. 5, pp. 605–612, 1988.
- [8] T. Laakso, V. Valimaki, M. Karjalainen, and U. Laine, "Splitting the unit delay [FIR/all pass filters design]," *Signal Processing Magazine, IEEE*, vol. 13, no. 1, pp. 30–60, 1996.
- [9] C. Tellambura, Y. J. Guo, and S. K. Barton, "Channel estimation using aperiodic binary sequences," *IEEE Comm. Lett.*, vol. 2, no. 5, pp. 140–142, May 1998.
- [10] C. Suh, C.-S. Hwang, and H. Choi, "Preamble design for channel estimation in MIMO-OFDM systems," in *IEEE Global Telecommunications Conference (GLOBECOM'03)*, vol. 1, Dec. 2003, pp. 317–321.
- [11] M. J. E. Golay, "Complementary series," *IRE Trans. Inform. Theory*, vol. IT-7, pp. 82–87, Apr. 1961.
- [12] S. Z. Budisin, "Efficient pulse compressor for Golay complementary sequences," *Elec. Letters*, vol. 27, no. 3, pp. 219–220, Jan. 1991.
- [13] B. Popovic, "Efficient Golay correlator," *Electronics Letters*, vol. 35, no. 17, pp. 1427–1428, 1999.
- [14] X. Huang and Y. Li, "Polyphase scalable complete complementary sets of sequences," in *Communication Systems, 2002. ICCS 2002. The 8th International Conference on*, vol. 2, 2002, pp. 810–814.
- [15] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. New York: McGraw-Hill, 1991.