February 15, 2005

Noncoherent Iterative Demodulation and Decoding for OFDM Systems

Abstract: In this report, a numerical comparison between a coherent and a noncoherent receiver algorithm in an OFDM system is undertaken. For noncoherent reception, differential QPSK modulation is employed at the transmitter. The noncoherent receiver relies on the principles of iterative demodulation and decoding to recover the transmitted information bits. For coherent reception, pilot symbols are inserted at regular time intervals at the transmitter and used to obtain channel estimates at the receiver. The performance of an iterative coherent receiver is also explored in the report, with receiver iterating on the channel estimates in a data-directed fashion. Noncoherent transmission and reception techniques could potentially significantly improve the throughput of communication systems, as they eliminate the overhead of pilot symbol transmissions. However, to achieve performance comparable to that of coherent systems, noncoherent systems generally require noncoherent demodulation techniques that are of higher complexity relative to coherent reception. The intent of this report is to perform a side-by-side comparison of the two communication paradigms in the framework of an OFDM system and explore the tradeoff between complexity and throughput of noncoherent systems.

Version 0.1

Noah Jacobsen, University of California, Santa Barbara
Eugene Visotsky, Motorola Labs
Roger Peterson , Motorola Labs

Document Change Log:

Version	Change Description	Date	Author
0.1	Original DRAFT	February 24, 2005	Visotsky

1 Introduction

The purpose of this report is to present a performance evaluation of the noncoherent iterative demodulation/decoding scheme recently proposed by Jacobsen and Madhow in [1] for differentially modulated coded systems. The evaluation is to be performed in the context of the 4G OFDM concept system under development at Motorola Labs. The scheme is to be evaluated via a direct throughput comparison with a coherently demodulated OFDM system of comparable complexity. For comparison, the details of the OFDM system, such as the number of sub-carriers, sub-carrier spacing, and cyclic prefix length and so forth, will be the same for both the differential demodulator and the coherent receiver.

The reported simulation results extend the work of Noah Jacobsen that he has performed in the summer of 2002 while an intern at Motorola Labs. The work is extended by relying on more sophisticated iterative demodulation/decoding techniques, developed since the summer of 2002 during the course of his Ph.D. research. The benchmark coherently demodulated OFDM system uses known symbols (pilots) for channel estimation, whereas the differentially modulated system employs convolutional coding with serially concatenated symbol-by-symbol differential modulator, per recommendation in [1]. The coherent system employs turbo-coding and an iterative technique for joint data and channel estimation. An effort will be made to keep the decoding/demodulation complexity of the two systems roughly equal, so as to facilitate a fair comparison. In particular, as both the coherent and differential systems rely on the MAP algorithm as the basis for soft-in/soft-out demodulation and decoding, the complexity of both systems will be estimated in terms of the number of max-star (log-MAP) or max-log-MAP kernel operations required for complete decoding of one information word. Detailed descriptions of the OFDM system model, channel model and of differentially modulated and coherent systems are given below.

2 Noncoherent OFDM

We consider the design of a noncoherent wireless transceiver for a 4G OFDM system. By definition, noncoherent communication does not require the transmission of pilot symbols for the purposes of channel estimation and tracking. Instead, channel continuity in time and frequency is leveraged by a turbo-like receiver that iteratively demodulates and decodes the received data sequence. At the demodulator, channel uncertainty is handled with a block-fading model with (1) MMSE amplitude estimation and (2) Bayesian combination of coherent symbol probabilities that result from quantizing the unknown channel phase.

2.1 System model

We adopt a turbo architecture based on serial concatenation of an "outer" channel code and "inner" unit-rate differential modulation code. Figure 1 depicts the OFDM transmitter, with differential modulation and error correction coding, and the channel model. A non-systematic convolutional code is used as an "outer" channel code. The information bits $\boldsymbol{u}$ are coded and interleaved and passed to a differential modulator. The $\boldsymbol{t}^{th}$ absolute symbol $\boldsymbol{b}_t$ is given by (1):

$$b_t = d_t b_{t-1}, \quad t = 0,1,\ldots,T-1 \qquad (1)$$

where $d_t \in \left\{ \exp\left(j2\pi \frac{m}{M}\right) \right\}_{m=0}^{m=3}$ is a differentially encoded QPSK symbol and $\boldsymbol{T}$ is the block length. Letting $\widetilde{\boldsymbol{h}}_t$ denote $\boldsymbol{t}^{th}$ fading coefficient and $\boldsymbol{n}_t$ denote i.i.d. $\left(0, 2\sigma^2\right)$ complex Gaussian noise, the baseband received symbol can be expressed as $r_t = \widetilde{h}_t b_t + n_t$. In this model, it is assumed that $\widetilde{\boldsymbol{h}}_t$ is a zero-mean, unit-variance proper complex Gaussian random variable. For the numerical results, channel coefficients are determined by the wide sense stationary with uncorrelated scattering (WSSUS) channel model. The case of the additive White Gaussian Noise (AWGN) channel, where $\widetilde{h}_t = \exp(j\theta)$ for all $\boldsymbol{t}$ and θ is uniformly distributed on $\left[0, 2\pi\right]$ is also numerically investigated.

Demodulation is based on approximating the channel as constant across an appropriate number of time and frequency bins. Thus, the transmitted symbol sequence should be mapped to the time and frequency dimensions in such a manner so as to maximize the correlation between the channels $\{\tilde{h}_t\}$. For this purpose, the OFDM channel is conveniently represented by a time/frequency grid, as in Figure 2. In the figure, columns represent OFDM symbols (or bauds), and rows correspond to subcarriers. The arrows represent the mapping of the QPSK symbols onto the time/frequency grid. The depicted mapping is well matched to an OFDM channel that does not vary significantly over $L=4$ bauds. The receiver operates by approximating the time/frequency channel as constant over a block of T transmitted symbols, and is free to choose the number of subcarriers K for which such an approximation holds. For example, the typical urban (TU) WSSUS channel, which we employ for numerical investigations, is best approximated as constant over two subcarriers ($K=2$). Accounting for a one-symbol overlap required for differential demodulation, the resulting block length is $T=L\times K+1$.

The received symbol vector over one block can then be expressed as follows

$$\mathbf{r}=\tilde{h}\mathbf{b}+\mathbf{n}=\tilde{h}b_0\bar{\mathbf{b}}+\mathbf{n}=h\bar{\mathbf{b}}+\mathbf{n} \qquad (2)$$

where $\tilde{h}$ denotes the value of the channel gain constant over the block and $\mathbf{b}$ is a vector of absolute symbols transmitted in this block.

Absorbing reference symbol b_0 into the channel coefficient does not affect its circularly symmetric density and constrains the absolute symbol sequence to start in the zero-phase state. That is, h is also a zero-mean, unit-variance proper complex Gaussian random variable and $\bar{b}_0=1$. With reference to the path traversed on the modulator trellis, we refer to the vector $\bar{\mathbf{b}}$ as the normalized state sequence.

2.2 Noncoherent demodulation using phase quantization

Refer to [1] for a detailed treatment of noncoherent demodulation based on phase quantization. The main steps of this approach are as follows. Demodulation is performed at a block level. First, a simple averaging estimate of the channel amplitude, based on a block of received symbols $\mathbf{r}$ and the MMSE criterion, is computed via $A^2=\mathbf{r}^H\mathbf{r}/T-2\sigma^2$. With this estimate, unknown channel phase is quantized and the log-likelihood that the t^{th} differential symbol is d is approximately

$$\lambda_t^O(d)\approx\max_{q=0,\ldots,Q-1}\left\{\lambda_t^O(d\mid h=A\exp(j\pi q/2Q))\right\}, \qquad (3)$$

where Q denotes the number of quantization levels, and $\lambda_t^O(d\mid h)$ denotes the coherent, extrinsic (less prior probabilities) log-likelihood of the t^{th} differential symbol given the channel h, which is efficiently computed with the BCJR algorithm. Note that, due to modulo-$\pi/2$ rotational invariance of a QPSK alphabet, the phase needs only to be quantized in an interval of length $\pi/2$.

The phase quantization approach to noncoherent demodulation has been shown to perform near capacity for the block-fading channel. However, since each quantization level requires its own BCJR computation per iteration, the algorithm is quite complex. In an effort to reduce complexity without compromising performance, we propose a phase selection mechanism, based on the generalized ratio likelihood test (GLRT) that selects the two best quantized phase levels based on feedback from the channel decoder. All subsequent demodulation iterations consider only the two selected phase values per block, resulting in a demodulator complexity that is only twice that of coherent BCJR demodulation. This selection mechanism has been shown to yield negligible loss on the block fading channel [1] and is employed here to maintain a comparable level of complexity between the coherent and noncoherent systems.

2.3 Turbo processing

Demodulation and decoding are performed iteratively, similar to the iterative decoding algorithm for serial concatenated turbo codes. Turbo processing for noncoherent OFDM is illustrated in Figure 3. Since block

differential demodulation with the BCJR algorithm requires prior probabilities of the differential symbol sequence, the receiver is initialized with the standard two-symbol differential demodulation in what is referred to as the *bootstrap* procedure. In this procedure, the initial likelihoods $\lambda_t^O(d)$ of the differential symbols are computed based on the two dimensional density obtained with the standard two-symbol differential demodulation, then converted to bit-level statistics, deinterleaved and passed to the input of the outer channel decoder as prior code-bit probabilities. The channel decoder, in turn, runs the BCJR algorithm on the trellis of the "outer" convolutional code and outputs bit-level extrinsic posterior probabilities that are converted to differential symbol priors $\lambda_t^I(d)$ for the next iteration of *block* differential demodulation, which are then used as priors in (3). The algorithm iterates between the channel decoder and demodulator in this fashion. In the final stage of the algorithm, information bit LLRs collected at the output of the decoder are sliced to yield hard information bit estimates. A frame error is declared if an estimate of at least one information bit is erroneous. For the numerical results, perfect error detection at the receiver is assumed.

To summarize, the proposed iterative noncoherent demodulation/decoding algorithm is as follows. In the initial stage, bootstrap procedure is performed to initialize priors for the channel decoder. The channel decoder accepts these priors and in turn generates priors for the channel demodulator to be used in the second stage. In the intermediate stages, the noncoherent demodulator and decoder exchange code-bit and differential symbol priors. In the final stage, posterior information bit LLRs are sliced to obtain hard information bit estimates.

2.4 Complexity analysis of noncoherent detection

The following complexity analysis considers the number of kernel operations (log-MAP or max-log-MAP) required for a complete decoding attempt of one codeword. For simplicity, the analysis will omit the complexity of the initial bootstrapping procedure, as well as other trivial operations such as bit-to-symbol and symbol-to-bit mappings, as complexity counts of equivalent operations will be similarly omitted in the complexity analysis of the benchmark iterative coherent demodulation/decoding scheme. The baseline in the complexity analysis is defined as one *binary* kernel operation (*bkop*) accepting two real-valued inputs in the log-domain and producing one output log-domain value. The complexity of kernel operations involving more than two inputs is assumed to be that of complexity of the binary kernel times the number of inputs minus one. For example, the complexity of a 4-ary kernel is assumed to be three times that of the binary kernel. For the complexity analysis, the following notation is useful. Let M denote the size of the modulation alphabet, so that there are $m = \log_2 M$ modulating bits per symbol. Let C denote the number of coded bits and Q denote the number of phase quantization bins used in the first iteration of the algorithm. For the convolutional code, let $r_C = k/n$ define its rate and mem denote its memory. Thus, $V = C/n$ gives the number of trellis sections for the convolutional decoder, where $S = 2^{mem}$ denotes the number of states per trellis section.

With the preceding definitions, it follows that the complexity of the convolutional decoding step performed in each stage of the algorithm is $N_1 = 2C(S - 1) + 2VS$ *bkops*. This is also the total complexity of the initial stage of the algorithm as the complexity of the symbol-by-symbol demodulation procedure to bootstrap the algorithm is ignored. The complexity of the noncoherent demodulation step for a single quantized phase value is $N_2 = 2(C/m)M(M - 1) + 2C(M^2/2 - 1)$ *bkops.* Then, the complexity of the second stage of the algorithm is $N_1 + QN_2$ *bkops,* where Q is the number of phase quantization levels used in the second stage. The complexity of each of the subsequent stages, excluding the final stage of the algorithm, is $N_1 + 2N_2$ *bkops*, as demodulation is only performed for the two best values of quantized phase. In the final stage of the algorithm, soft output in the decoder is computed only for the information bits. The complexity of this operation is $N_3 = 2r_C C(S - 1) + 2VS$ *bkops*. Hence, the complexity is of the last stage of the algorithm is $N_3 + 2N_2$ *bkops*. The overall complexity is $N_1 + N_1 + QN_2 + (N - 3)(N_1 + 2N_2) + N_3 + 2N_2$ *bkops*, where N denotes the total number of stages in the algorithm.

3 Numerical Results

In this section, results of a numerical comparison between the proposed iterative noncoherent receiver and an iterative coherent OFDM receiver employing pilot symbols for channel estimation and a turbo-code for error correction are presented. The comparison was performed in terms of the corresponding frame error rates, and information throughputs attained by the receivers. A WSSUS channel model with a TU power-delay profile and an AWGN channel models were employed for the comparison.

3.1 Benchmark coherent receiver

To facilitate coherent reception, it is assumed that a pilot OFDM symbol is transmitted after every four data-carrying OFDM symbols. The transmitter accepts a frame of information bits, performs turbo-encoding, interleaving and maps the resulting coded bits to a standard QPSK constellation. The 3GPP-defined parallel concatenated turbo-code is used. The transmitted codeword extends over the four data-carrying OFDM symbols, with one QPSK symbol transmitted on each OFDM subcarrier. At the receiver, the pilot OFDM symbols that immediately precede and follow the four data-carrying OFDM symbols are used to estimate the channel for decoding the codeword. For numerical results, an iterative coherent receiver was used to benchmark the performance of the proposed noncoherent iterative algorithm. In the initial stage of the algorithm, iterations of the standard turbo-decoding algorithm is performed based on the channel estimates obtained from the pilot symbols. At the conclusion of the initial stage, the turbo-decoder outputs a posteriori probabilities (APPs) for each coded bit. These soft values are converted into APPs for the received QPSK symbols, which are then used to refine the channel estimates. The refined channel estimates are passed to the channel decoder, and the APPs are again re-computed for the next stage of the algorithm by performing turbo-decoding iterations. In the final stage of the algorithm, the turbo-decoder outputs APPs only for the information bits. The APPs are then sliced to obtain the hard bit estimates.

In the initial stage, channel estimation is performed solely based on the pilot symbols. In particular, as the pilot symbol is unmodulated, the received sample on each subcarrier provides a raw estimate of the channel gain for that subcarrier. The raw channel estimate is further smoothed by computing its weighted average together with the channel estimates in the surrounding frequency bins. This smoothing operation can be efficiently implemented by applying an FIR filter across subcarriers. The channel estimate for a subcarrier of a data-carrying OFDM symbol is obtained by linearly interpolating between the smoothed channel estimates on the corresponding subcarriers of the leading and following pilot symbols. The received sample on each subcarrier of the data-carrying symbol is multiplied by the conjugate of the channel estimate and passed to the decoder.

In the subsequent stages, data-carrying OFDM symbols are used to refine the initial channel estimates. This data-directed channel estimation procedure is as follows. Consider a single OFDM sub-carrier. Assuming no inter-carrier interference, the receive signal on the sub-carrier is simply given by $y = h \times s + n$, where h denotes the channel gain, s denotes transmitted symbol and n is noise. The channel estimate is obtained as the solution to the following optimization problem:

$$\hat{h} = \min_{h} E_{p(\tilde{s})}\left[\left| y - h \times \tilde{s} \right|^2\right] \tag{4}$$

where the expectation is computed with respect to the distribution $p(\tilde{s})$ obtained by converting the coded bit APPs supplied by the turbo-decoder into the QPSK symbol probabilities. The solution is

$$\hat{h} = y \times E_{p(\tilde{s})}\left[s^*\right] / E_{p(\tilde{s})}\left[|s|^2\right] \tag{5}$$

Similarly to the pilot symbol channel estimates, the data symbol channel estimates are smoothed across sub-carriers by an FIR filter. At this point of the algorithm, six channel estimates are obtained for each frequency bin, corresponding to the leading and following pilot symbols, and to the four data-carrying symbols. These channel estimates are effectively time samples of the channel fading process in each frequency bin. As a first order approximation, the channel fading process is assumed to be linear across the time span of the two pilot and four data-carrying symbols. Based on this approximation, the final channel estimate for each

subcarrier of the four data-carrying symbols is obtained by curve fitting the six channel estimates to a linear function and sampling it at the time instances corresponding to the four data-carrying symbols.

3.2 Complexity analysis of the coherent receiver

In the initial stage, the complexity of the channel estimation step is assumed to be negligible relative to the turbo-decoding step, as the channel estimation step requires simple FIR filtering using pre-computed FIR coefficients and trivial curve-fitting calculations to a linear function. For complexity analysis of turbo-decoding, let M denote the size of the modulation alphabet ($M=2$ for QPSK), let C denote the number of coded bits transmitted, let $r_C = k/n$ define the rate, let mem denote the memory of the constituent convolutional code, and let r_{turbo} define the overall rate of the turbo-code. Then, $S = 2^{mem}$ is the number of states in the trellis of the constituent code, and $V = r_{turbo}C$ gives the number of information bits transmitted and the number of trellis sections for each constituent code. In each stage of the algorithm, the turbo-decoder iterates on the information bit APPs. Let N_{turbo} denote the number of turbo iterations in each stage of the algorithm. Furthermore, in the last iteration of the initial or of an intermediate stage, the turbo-decoder outputs APPs for all coded bits in order to update the channel estimates for the next stage of the algorithm. Similarly to the decoding complexity of the convolutional code computed in Section 2.4, the complexity of a single pass of the turbo-decoding algorithm over a single constituent convolutional code to compute information bit APPs is $N_1 = 2V(S-1) + 2VS$ *bkops.* Likewise, the corresponding complexity to compute coded bit APPs is $N_2 = 2Vn(S-1) + 2VS$. Then, the complexity of the initial or of the intermediate stage of the algorithm is $N_3 = (2N_{turbo} - 1)N_1 + N_2$. The complexity of the final stage of the algorithm is $N_4 = 2N_{turbo}N_1$. The overall complexity of the algorithm is $N_3(N-1) + N_4$, where N denotes the number of stages in the algorithm.

3.3 Simulation comparison

The following OFDM system parameters were used in the simulations. FFT size (or total number of subcarriers) is 1024, sample rate 25.6 MHz, length of the cyclic prefix is 256. The corresponding OFDM symbol duration is $(256+1024)/25.6\times10^6 = 50\,\mu s$, corresponding to a symbol rate of 20 *ksps*. The number of subcarriers used for data transmission is 768, corresponding to a nominal system bandwidth of 19.2 MHz. The carrier frequency is 3.676 GHz. For both systems, the coding is applied across four contiguous OFDM symbols; the modulation is differential QPSK for the noncoherent receiver and standard QPSK for the coherent one. In both cases, C equals 6144 bits. For the benchmark coherent receiver, the overall turbo-coding rate is $r_{turbo} = 1/2$ ($r_C = 1/3$, $mem = 3$ for both constituent codes), corresponding to the information bit rate of 12.28 Mbps, where a factor of $4/5$ is applied to account for the information rate loss due to transmissions of the pilot symbols. For numerical results, the performance of the coherent receiver with four stages and $N_{turbo} = 8$ was simulated. Further iterations of the algorithm were determined to result in negligible performance gains. Using the expressions in Section 3.2, the complexity of the receiver is about 24.6×10^9 *bkops*. As a point of reference, the complexity of a standard coherent receiver that is comprised only of the first stage of the iterative channel estimation algorithm is only 6.0×10^9 *bkops.*

For simulating the noncoherent receiver, several choices of the convolutional code were explored. Numerical results are presented for the following code rate and memory pairs: $(mem = 4, r_C = 1/4)$, $(mem = 4, r_C = 1/2)$, $(mem = 6, r_C = 1/2)$. The receiver was simulated with four stages, including the initial stage to bootstrap the algorithm. Additional stages of the algorithm result in negligible performance gains. The complexity count and maximum throughput for the three choices of code parameters are $(11.2\times10^9\ bkops, 7.68\,\text{Mbps})$, $(12.4\times10^9\ bkops, 15.4\,\text{Mbps})$, and $(28.6\times10^9\ bkops, 15.4\,\text{Mbps})$.

Numerical results are displayed in Figure 6, Figure 7, and Figure 8. All results are displayed as a function of the information bit E_b/N_0 ratio. The performance of the receivers is shown in an AWGN and a WSSUS channel with a TU power delay profile. Each ray in the power delay profile exhibits independent fading at a Doppler rate of 160 Hz, roughly corresponding to a speed of 30 mph at the carrier frequency of the system. The frame error rate (FER) plots are displayed in Figure 6. Figure 7 and Figure 8 display the throughput attained by both receivers in AWGN and the WSSUS channel, respectively. The throughput as a function of SNR, $T(SNR)$, is defined as:

$$T(SNR) = (1 - FER(SNR)) \times R_{max} \quad (6)$$

where R_{max} is the maximum information bit rate that is supported by a given combination of modulation and code rate. Note that this definition assumes perfect error detection at the receiver and that no retransmission of erroneous frames is performed. For comparison, the performance of the standard single-stage coherent receiver and the performance of the noncoherent receiver with only two stages, the bootstrap stage and the full phase quantization stage, are also displayed in the figures.

3.4 Discussion

Based on the numerical results, several observations are in order. The performance of the single-stage coherent receiver is only slightly worse than that of the four-stage coherent receiver for most E_b/N_0 operating points in both channels. This slight gain in performance of the iterative channel estimation scheme does not justify its significant cost in complexity, roughly four-fold relative to the complexity of the single-stage coherent receiver. Among the three code parameter pairs used for the noncoherent receiver, the pair $(mem = 4, r_C = 1/2)$ appears to be the best choice to trade-off complexity of noncoherent detection against throughput performance.

The throughput performance of this code is just slightly worse than that of the more powerful, but significantly more complex, code used for the $(mem = 6, r_C = 1/2)$ pair. From the results for the fading channel, it is evident that increasing the number of stages of the noncoherent receiver from two to four does result in a significant performance improvement in terms of both the FER and throughput metrics. This gain is more pronounced than the performance improvement obtained by increasing the number of stages of the coherent receiver from one to four. Relative to the coherent scheme, the noncoherent scheme does offer a higher throughput; however, it is more susceptible to noise and is more complex. The sensitivity to noise is highlighted in Figure 6, where the FER curves of the coherent receiver utilizing a rate one-half turbo code, are below, for all values of E_b/N_0 in both channels, the FER curves of the noncoherent receiver utilizing a rate one-half convolutional code. The sensitivity to noise is also evident in Figure 7, where the throughput curves of the noncoherent receiver rise at a higher E_b/N_0 value relative to the corresponding E_b/N_0 value of the coherent receiver. This effect, however, seems to be less pronounced for the fading channel results. For a complexity comparison, consider the complexity of the four-stage noncoherent receiver with $(mem = 4, r_C = 1/2)$, given by 12.4 *bkops*, and that of the single-stage coherent receiver, at 6.0×10^9 *bkops*. It should be noted that the complexity of both receivers could be reduced by decreasing the number of stages in the former and the number of turbo iterations in the latter, with possibly minimal effect on the performance of both receivers.

4 Conclusion

Noncoherent iterative demodulation and decoding appears to be a viable alternative to standard receivers that rely on pilot symbols for channel estimation, demodulation and decoding. As demonstrated in this report, omission of pilot transmissions results in higher throughput of the noncoherent scheme relative to the pilot-based coherent receiver. As noncoherent iterative schemes require feedback of soft information from the channel decoder to the demodulator, they are inherently more complex than standard coherent receivers

that normally do not require such feedback. However, this increase in complexity is not prohibitive, as for instance a 25% increase in throughput was achieved with only a two-fold increase in complexity by the noncoherent scheme investigated in this report. The scheme exhibited higher sensitivity to noise as compared to the benchmark coherent receiver, which suggests that noncoherent transmissions might be more appropriate in high SNR regime. The results in this report are by no means exhaustive and should be regarded only as a first step towards a comprehensive evaluation of iterative noncoherent receivers in the framework of Motorola's OFDM systems. Overall, the results in this report are encouraging and point to iterative noncoherent receivers as a possible design choice for future wireless systems.

5 References

[1] Noah Jacobsen and Upamanyu Madhow, "Coded Noncoherent Communication with Amplitude/Phase Modulation: from Shannon Theory to Practical Turbo Architectures", *submitted for publication*.

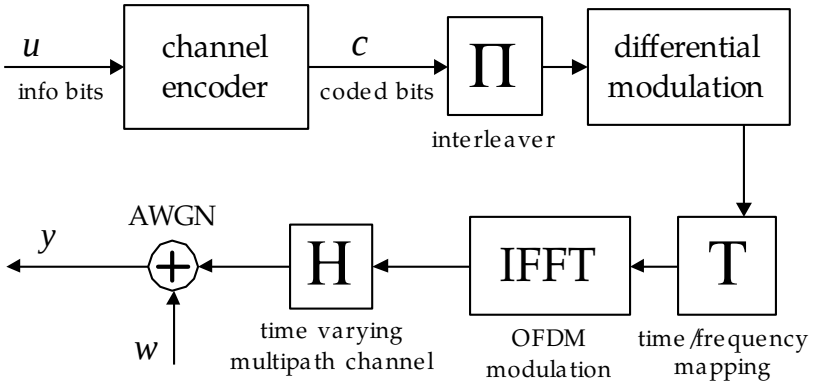

Figure 1. Model for the noncoherent OFDM transmitter.

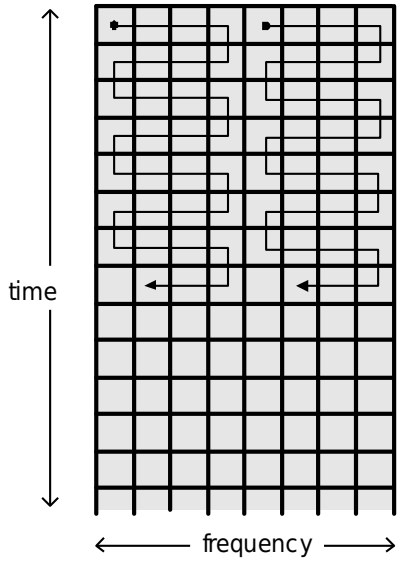

Figure 2. Differential symbol encoding in time and frequency.

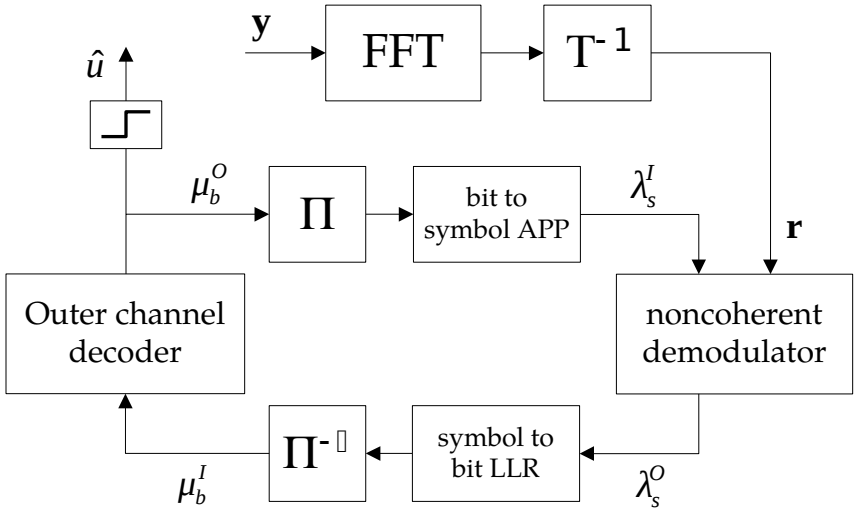

Figure 3. Differential demodulator/decoder.

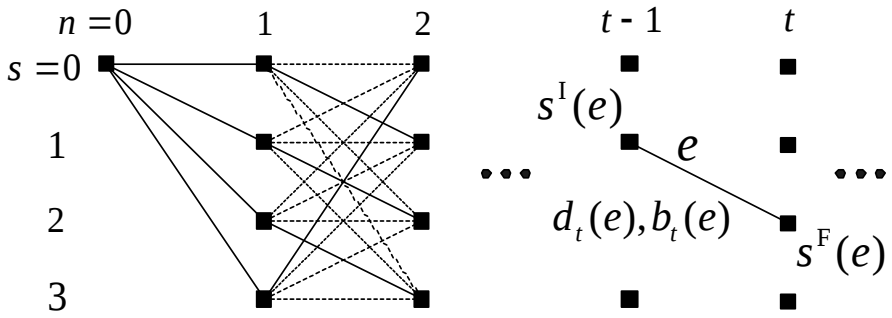

Figure 4. Differential demodulation trellis.

Figure 5. Demodulation module.

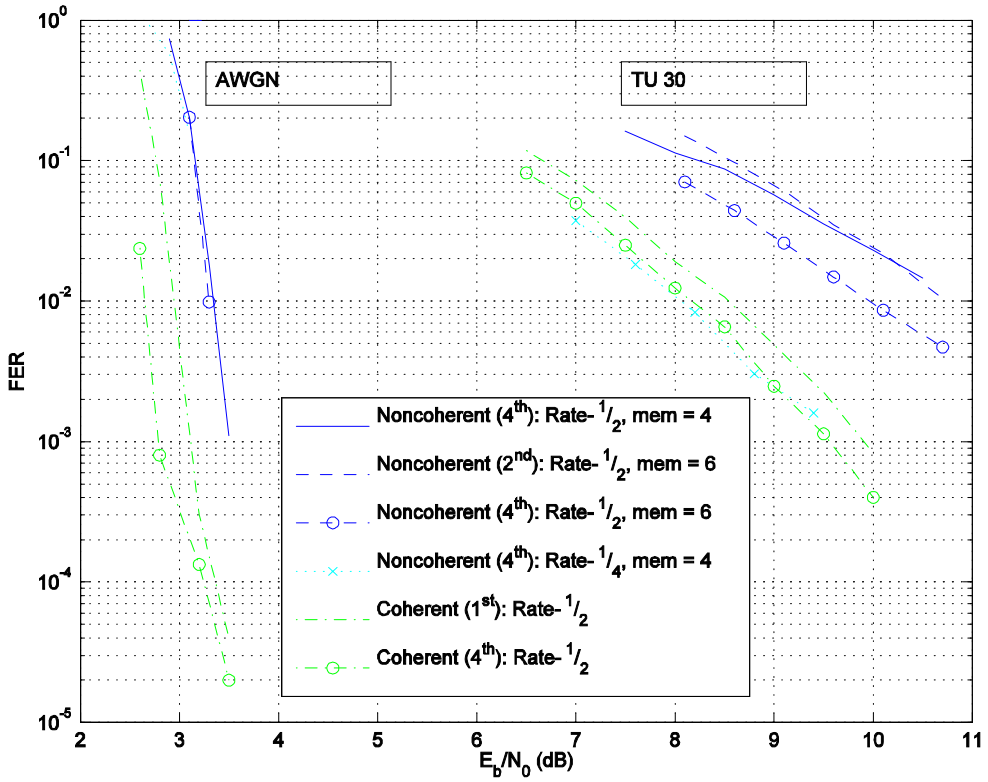

Figure 6. Frame error rates of the noncoherent and coherent receivers in AWGN and TU30 channel.

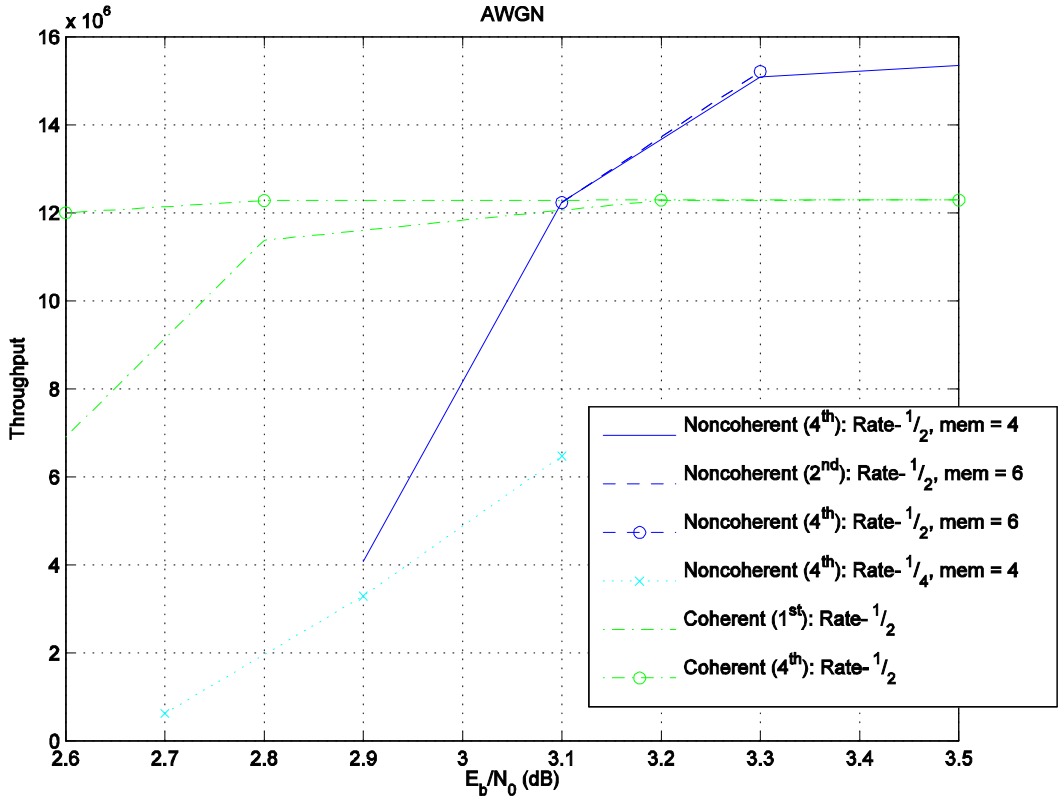

Figure 7. Attained throughput of the noncoherent and coherent receivers in AWGN.

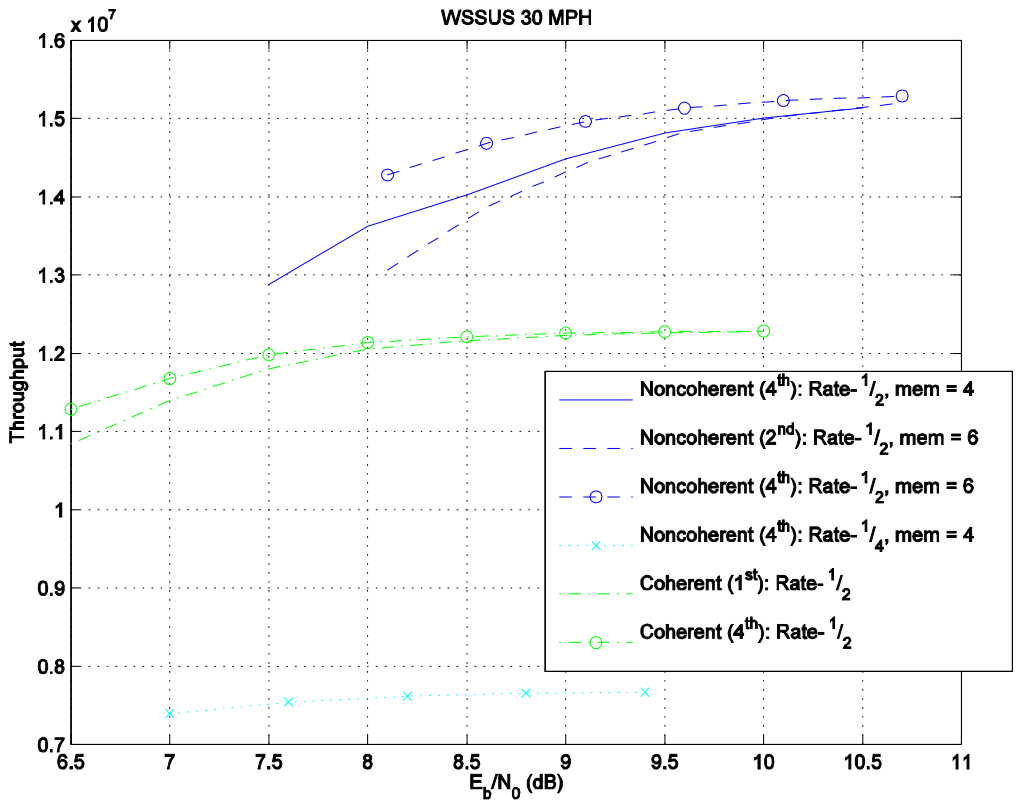

Figure 8. Attained throughput of the noncoherent and coherent receivers in TU30.