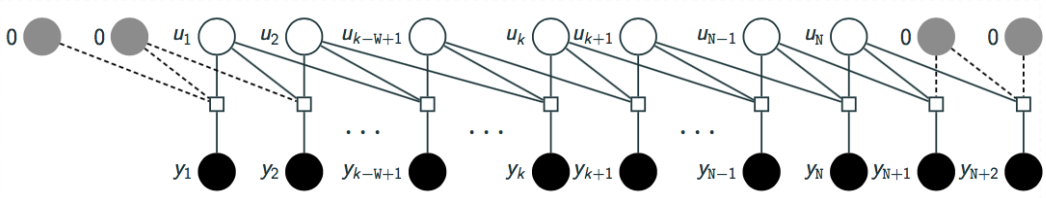
Doctoral Thesis
PhD in Automatic, Electronics and Telecommunication Engineering

Expectation Propagation as a Solution for Digital Communication Systems



Author: Irene Santos Velázquez
Advisors: Juan José Murillo Fuentes
Eva María Arias de Reyna Domínguez

Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2018

Doctoral Thesis
PhD in Automatic, Electronics and Telecommunication Engineering

Expectation Propagation as a Solution for Digital Communication Systems

Author:

Irene Santos Velázquez

Advisors:

Juan José Murillo Fuentes

Profesor Catedrático

Eva María Arias de Reyna Domínguez

Profesor Contratado Doctor

Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

2018

Tesis Doctoral: Expectation Propagation as a Solution for Digital Communication Systems

Autor: Irene Santos Velázquez
Directores: Juan José Murillo Fuentes
Eva María Arias de Reyna Domínguez

El tribunal nombrado para juzgar la Tesis arriba indicada, compuesto por los siguientes doctores:

Presidente:

Vocales:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

Me gustaría comenzar agradeciendo a aquellas personas sin las cuales esta tesis no hubiera sido posible.

Juanjo, creo que no podría haber encontrado un mejor tutor/director de la tesis que tú. Quiero agradecerte la oportunidad que me diste hace cuatro años para iniciarme en el mundo de la investigación contigo. Sin tus consejos, enseñanza y dedicación no hubiera podido sacar adelante este trabajo.

Eva, tenerte como co-directora ha sido un privilegio y una pieza fundamental para desarrollar esta tesis. Gracias por tu tiempo y minuciosidad. Han sido clave para que la tesis sea lo más clara y amigable posible.

No pueden faltar tampoco mis responsables de estancia, Petar y Pablo. Gracias por enseñarme nuevas líneas de investigación y por toda vuestra ayuda durante mi estancia.

También quiero agradecer a todos mis compañeros de Departamento, en especial a los fieles de la hora del desayuno: Rafa, Irene, Uxi y Javier (aunque sólo desayunes los viernes). Por supuesto, también a todos mis compis de sala, porque me están haciendo una experta en música y hacen mi día a día mucho más divertido.

Y no puedo terminar sin agradecer a mis padres, mi hermano y Javi. Porque sin vuestro apoyo incondicional sé que no lo hubiera conseguido.

Abstract

In the context of digital communications, a digital receiver is required to provide an estimation of the transmitted symbols. Nowadays channel decoders highly benefit from soft (probabilistic) estimates for each transmitted symbol rather than from hard decisions. For this reason, digital receivers must be designed to provide the probability that each possible symbol was transmitted based on the received corrupted signal. Since exact inference might be unfeasible in terms of complexity for high-order scenarios, it is necessary to resort to approximate inference, such as the linear minimum mean square error (LMMSE) criterion. The LMMSE approximates the discrete prior information of the transmitted symbols with a Gaussian distribution, which causes a degradation in its performance. In this thesis, an alternative approximate statistical technique is applied to the design of a digital probabilistic receiver in digital communications. Specifically, the expectation propagation (EP) algorithm is investigated to find the Gaussian posterior probability density function (pdf) that minimizes the Kullback-Leibler (KL) divergence with respect to the true posterior pdf.

Two different communication system scenarios are studied: a single-input single-output (SISO) digital communication system with memory channel and a multiple-input multiple-output (MIMO) system with memoryless channel. In the SISO scenario, three different designs of a soft standalone and turbo equalizer based on the EP algorithm are developed: the block or batch approach, the filter-type version that emulates the Wiener filter behavior and the smoothing equalizer which proceeds similarly to a Kalman smoother. Finally, the block EP implementation is also adapted to MIMO scenarios with feedback from the decoder. In both scenarios, the EP is applied iteratively, including a damping mechanism and a control to avoid negative values of variances, which would lead to instabilities (specially for high-order constellations). Experimental results included through

the thesis show that the EP algorithm applied to communication systems greatly improves the performance of previous approaches found in the literature with a complexity slightly increased but still proportional to that of the LMMSE. These results also show the robustness of the algorithm even for high-order modulations, large memory channels and high number of antennas.

Major contributions of this dissertation have been published in four journal (one of them is still under review) and two conference papers. One more paper will be submitted to a journal soon. All these papers are listed below:

- Irene Santos, Juan José Murillo-Fuentes, Rafael Boloix-Tortosa, Eva Arias de Reyna and Pablo M. Olmos, "Expectation Propagation as Turbo Equalizer in ISI Channels," IEEE Transactions on Communications, vol. 65, no.1, pp. 360-370, Jan 2017.
- Irene Santos, Juan José Murillo-Fuentes, Eva Arias de Reyna and Pablo M. Olmos, "Turbo EP-based Equalization: a Filter-Type Implementation," IEEE Transactions on Communications, Sep 2017, Accepted. [Online] Available: https://ieeexplore.ieee.org/document/8353388/
- Irene Santos, Juan José Murillo-Fuentes, Eva Arias-de-Reyna and Pablo M. Olmos, "Probabilistic Equalization With a Smoothing Expectation Propagation Approach," IEEE Transactions on Wireless Communications, vol. 16, no. 5, pp. 2950-2962, May 2017.
- Irene Santos, Juan José Murillo-Fuentes and Eva Arias-de-Reyna, "Equalization with Expectation Propagation at Smoothing Level," To be submitted. [Online] Available: https://arxiv.org/abs/1809.00806
- Irene Santos and Juan José Murillo-Fuentes, "EP-based turbo detection for MIMO receivers and large-scale systems," IEEE Transactions on Vehicular Technology, May 2018, Under review. [Online] Available: https://arxiv.org/abs/1805.05065
- Irene Santos, Juan José Murillo-Fuentes, and Pablo M. Olmos, "Block expectation propagation equalization for ISI channels," 23rd European Signal Processing Conference (EUSIPCO 2015), Nice, 2015, pp. 379-383.
- Irene Santos, and Juan José Murillo-Fuentes, "Improved probabilistic EP-based receiver for MIMO systems and high-order modulations," XXXIII Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2018), Granada, 2018.

Resumen

En el ámbito de las comunicaciones digitales, es necesario un receptor digital que proporcione una estimación de los símbolos transmitidos. Los decodificadores de canal actuales se benefician enormemente de estimaciones suaves (probabilísticas) de cada símbolo transmitido, en vez de estimaciones duras. Por este motivo, los receptores digitales deben diseñarse para proporcionar la probabilidad de cada posible símbolo que fue transmitido en base a la señal recibida y corrupta. Dado que la inferencia exacta puede no ser posible en términos de complejidad para escenarios de alto orden, es necesario recurrir a inferencia aproximada, como por ejemplo el criterio de linear minimum-mean-square-error (LMMSE). El LMMSE aproxima la información a priori discreta de los símbolos transmitidos con una distribución Gaussiana, lo cual provoca una degradación en su resultado. En esta tesis, se aplica una técnica alternativa de inferencia estadística para diseñar un receptor digital probabilístico de comunicaciones digitales. En concreto, se investiga el algoritmo expectation propagation (EP) con el objetivo de encontrar la función densidad de probabilidad (pdf) a posteriori Gaussiana que minimiza la divergencia de Kullback-Leibler (KL) con respecto a la pdf a posteriori verdadera.

Se estudian dos escenarios de comunicaciones digitales diferentes: un sistema de comunicaciones single-input single-output (SISO) con canales con memoria y un sistema multiple-input multiple-output (MIMO) con canales sin memoria. Para el escenario SISO se proponen tres diseños diferentes para un igualador probabilístico, tanto simple como turbo, que está basado en el algoritmo EP: una versión bloque, una versión filtrada que emula el comportamiento de un filtro Wiener y una versión smoothing que funciona de forma similar a un Kalman smoother. Finalmente, la implementación del EP en bloque se adapta también para escenarios MIMO con realimentación desde el decodificador. En ambos escenarios, el EP se aplica de forma iterativa, incluyendo un mecanismo de damping y un control para evitar

valores de varianzas negativas, que darían lugar a inestabilidades (especialmente, en constelaciones de alto orden). Los resultados experimentales que se incluyen en la tesis muestran que, cuando el algoritmo EP se aplica a sistemas de comunicaciones, se mejora notablemente el resultado de otras propuestas anteriores que existen en la literatura, con un pequeño incremento de la complejidad que es proporcional a la carga del LMMSE. Estos resultados también demuestran la robustez del algoritmo incluso para modulaciones de alto orden, canales con bastante memoria y un gran número de antenas.

Las principales contribuciones de esta tesis se han publicado en cuatro artículos de revista (uno de ellos todavía bajo revisión) y dos artículos de conferencia. Otro artículo adicional se encuentra en preparación y se enviaría próximamente a una revista. Estos se citan a continuación:

- Irene Santos, Juan José Murillo-Fuentes, Rafael Boloix-Tortosa, Eva Arias de Reyna and Pablo M. Olmos, "Expectation Propagation as Turbo Equalizer in ISI Channels," IEEE Transactions on Communications, vol. 65, no.1, pp. 360-370, Jan 2017.
- Irene Santos, Juan José Murillo-Fuentes, Eva Arias de Reyna and Pablo M. Olmos, "Turbo EP-based Equalization: a Filter-Type Implementation," IEEE Transactions on Communications, Sep 2017, Aceptado. [Online] Disponible: https://ieeexplore.ieee.org/document/8353388/
- Irene Santos, Juan José Murillo-Fuentes, Eva Arias-de-Reyna and Pablo M. Olmos, "Probabilistic Equalization With a Smoothing Expectation Propagation Approach," IEEE Transactions on Wireless Communications, vol. 16, no. 5, pp. 2950-2962, May 2017.
- Irene Santos, Juan José Murillo-Fuentes and Eva Arias-de-Reyna, "Equalization with Expectation Propagation at Smoothing Level," En preparación. [Online] Disponible: https://arxiv.org/abs/1809.00806
- Irene Santos and Juan José Murillo-Fuentes, "EP-based turbo detection for MIMO receivers and large-scale systems," IEEE Transactions on Vehicular Technology, May 2018, En revisión. [Online] Disponible: https://arxiv.org/abs/1805.05065
- Irene Santos, Juan José Murillo-Fuentes, and Pablo M. Olmos, "Block expectation propagation equalization for ISI channels," 23rd European Signal Processing Conference (EUSIPCO 2015), Nice, 2015, pp. 379-383.
- Irene Santos, and Juan José Murillo-Fuentes, "Improved probabilistic EP-based receiver for MIMO systems and high-order modulations," XXXIII

Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2018), Granada, 2018.

Contents

Notation

$\mathcal{R}(\cdot)$	real part
$\mathcal{I}(\cdot)$	imaginary part
u	scalar
$\mathbf{u}$	column vector
$\mathbf{U}$	matrix
u_i or $u(i)$	i-th element of the vector $\mathbf{u}$
$\mathbf{u}_{i:j}$	column vector with the entries of $\mathbf{u}$, in the range $[i,j]$ in increasing order if $i < j$, or in the range $[j,i]$ in decreasing order otherwise
$\mathbf{u}^{\setminus i}$	vector $\mathbf{u}$ without the i-th component
$\mathbf{U}^{\setminus i}$	submatrix obtained by removing from $\mathbf{U}$ the i-th row and column
$\mathrm{diag}(\mathbf{u})$	returns a diagonal matrix with diagonal given by $\mathbf{u}$
$\mathcal{CN}(\mathbf{u} : \mathbf{m}, \mathbf{C})$	normal distribution of a random proper complex vector $\mathbf{u}$ with mean $\mathbf{m}$ and covariance matrix $\mathbf{C}$
$(\cdot)^{\mathrm{H}}$	hermitian
$(\cdot)^*$	complex conjugate
$(\cdot)^{-1}$	inverse
$\propto$	proportional to
$\sim$	distributed according to
$\mathbf{I}$	identity matrix
$\mathbb{I}$	indicator function
$\mathcal{O}(\cdot)$	computational cost in operations
$\mathrm{Proj}_G[\cdot]$	projection of the distribution given as an argument into the family of Gaussians

$\delta(\cdot)$	delta function
$E(\cdot)$	expected value of the variable in $(\cdot)$
$\mathtt{K}$	length of the transmitted bit sequence
$\mathtt{V}$	length of the coded bit vector
$\mathtt{N}$	length of the transmitted frame
$\mathtt{R}=\mathtt{K}/\mathtt{V}$	code rate
$\mathtt{M}$	modulation/constellation size
$Q=\log_2(\mathtt{M})$	required bits for each symbol of the constellation
$\mathcal{A}$	set of symbols of the constellation
$\mathtt{L}$	length (number of taps) of the channel state information (CSI)
$\tilde{\mathtt{L}}=\mathtt{L}-1$	memory of the CSI
N_t	number of transmit antennas
N_r	number of receive antennas
$\mathbf{a}=[a_1,...,a_{\mathtt{K}}]^\top$	information bit sequence where $a_i \in \{0,1\}$
$\widehat{\mathbf{a}}=[\widehat{a}_1,...,\widehat{a}_{\mathtt{K}}]^\top$	estimated information bit sequence
$\mathbf{b}=[b_1,...,b_{\mathtt{V}}]^\top$	coded bit vector where $b_t \in \{0,1\}$
$\mathbf{c}=[c_1,...,c_{\mathtt{V}}]^\top$	coded bit vector after permuting the bits with an interleaver where $c_t \in \{0,1\}$
$\mathbf{c}=[\mathbf{c}_1,...,\mathbf{c}_{\mathtt{N}}]^\top$	permuted coded bit vector after partitioning it into $\mathtt{N}$ blocks where $\mathbf{c}_k=[c_{k,1},\cdots,c_{k,Q}]$
$\mathbf{u}$	modulated (or transmitted) symbols where $u_k \in \mathcal{A}$
$\widehat{\mathbf{u}}$	estimated transmitted symbols
$\mathbf{h}=[h_1,...,h_{\mathtt{L}}]^\top$	CSI
$\mathbf{h}_{\mathtt{L}:1}=[h_{\mathtt{L}},...,h_1]^\top$	flipped CSI
$\mathbf{y}$	received signal
σ_w^2	noise variance
$\mathbf{w}$	additive white Gaussian noise (AWGN) where $w_k \sim \mathcal{CN}\big(w_k : 0, \sigma_w^2\big)$
$\mathbf{H}$	channel matrix
E_s	the mean transmitted symbol energy
E_b	the energy per bit
S	number of EP iterations
ℓ	index of the EP iteration, $\ell \in [1,S]$
β	smoothing parameter used in the EP algorithm
ε	minimum allowed variance used in the EP algorithm
T	number of turbo iterations
t	index of the turbo iteration, $t \in [0,T]$

$\mathtt{W}$	length of the window in filtered or windowed algorithms
$p(\mathbf{u}\|\mathbf{y})$	true posterior pdf
$p(u_k\|\mathbf{y})$	k-th marginal of the true posterior pdf
$q(\mathbf{u})$	approximated posterior pdf
$q(u_k)$	k-th marginal of the full approximated posterior pdf
$p_E(u_k)$	true extrinsic pdf for the k-th symbol
$q_E(u_k)$	approximated extrinsic pdf for the k-th symbol
$p_D(\cdot)$	true probabilistic information returned by the channel decoder
$L(\cdot)$	posterior log-likelihood ratio (LLR) at the input of the decoder
$L_E(\cdot)$	extrinsic LLR at the input of the decoder
$L_D(\cdot)$	extrinsic LLR at the output of the decoder
$q^{[\ell]}(u_k)$	k-th marginal of the full approximated posterior pdf at ℓ-th EP iteration, $q^{[\ell]}(u_k) \sim \mathcal{CN}\left(u_k : \mu_k^{[\ell]}, s_k^{2[\ell]}\right)$
$q_E^{[\ell]}(u_k)$	approximated extrinsic pdf at ℓ-th EP iteration for the k-th symbol, $q_E^{[\ell]}(u_k) \sim \mathcal{CN}\left(u_k : z_k^{[\ell]}, v_k^{2[\ell]}\right)$
$\tilde{p}^{[\ell]}(u_k)$	factors of an approximated prior at ℓ-th EP iteration, $\tilde{p}^{[\ell]}(u_k) \sim \mathcal{CN}\left(u_k : m_k^{[\ell]}, \eta_k^{[\ell]}\right)$
$\varphi_j(u)$	j-th bit associated to the demapping of symbol u

1 Introduction

1.1 Motivation and objectives

Current digital communication systems need to transmit sequences of information bits over communication channels, providing high-speed data and high-quality services. During the propagation process, the digital transmission is corrupted by additive white Gaussian noise (AWGN), intersymbol interference (ISI) and interference from other users, channels of antennas, among others. All these events negatively affect the received signal. In many scenarios, the above problem can be expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{u} + \mathbf{w}, \tag{1.1}$$

where $\mathbf{y}$, $\mathbf{u}$, $\mathbf{H}$ and $\mathbf{w}$ are the received and transmitted signals, the channel matrix and the AWGN noise, respectively. Through this thesis, the channel matrix will be perfectly known and the noise will be distributed according to $\mathbf{w} \sim \mathcal{CN}(\mathbf{w} : \mathbf{0}, \mathbf{C_w})$, i.e., a circular complex Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\mathbf{C_w}$. In (1.1), $\mathbf{w}$ accounts for the noise while $\mathbf{H}$ models the interference.

The goal of a digital receiver is to provide an estimation of the transmitted sequence of symbols, $\widehat{\mathbf{u}}$, from the received corrupted signal, as shown in Figure 1.1. To further improve the estimation, a forward error correction (FEC) is usually introduced. At the transmitter, the channel encoder includes some redundancy to protect the data, by means of an error correction code (ECC). At the receiver, the channel decoder exploits the redundancy to significantly improve the estimation of the transmitted sequence, reducing the resulting bit error rate (BER). Furthermore, the estimation provided by the digital receiver can be further refined by feeding it back with the output of the channel decoder, as it is shown with dashed lines in

Figure 1.1. This exchange of information between the receiver and the channel decoder is usually referred to as turbo scheme.

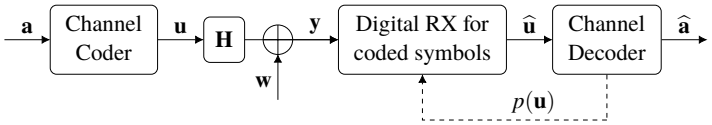


Figure 1.1 System model of a digital communication system.

The most straightforward way to estimate the transmitted symbols is by making hard decisions on the sequence of channel symbols that were transmitted. Nowadays channel decoders highly benefit from soft (probabilistic) estimates for each transmitted symbol, rather than from hard decisions. For this reason, digital receivers must be designed to provide to the channel decoder the probability of transmitting each possible symbol based on the received corrupted signal, i.e., the posterior probability. One commonly used technique to obtain this probability at the receiver is based on *Bayesian inference*.

One optimal criterion in Bayesian inference, known as maximum a posteriori (MAP) criterion, finds the estimation of a variable of interest given some observations by minimizing the probability of error. In the context of digital communications, the variable of interest and the observations are the transmitted and received signals, respectively. To compute the estimation of the transmitted sequence that was most probably transmitted, the MAP algorithm maximizes the a posteriori probabilities (APP) given the received signal, i.e.,

$$\widehat{\mathbf{u}} = \arg\max_{\mathbf{u}} p(\mathbf{u}|\mathbf{y}) \tag{1.2}$$

where, in case of the linear model of the digital communication system in (1.1),

$$p(\mathbf{u}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{u})p(\mathbf{u})}{p(\mathbf{y})} \propto \mathcal{CN}(\mathbf{y} : \mathbf{Hu}, \mathbf{C_w})\, p(\mathbf{u}) \tag{1.3}$$

and the prior information of the transmitted symbols, $p(\mathbf{u})$, follows a discrete distribution[1].

An alternative Bayesian estimator is the minimum mean square error (MMSE), that computes an estimation of the transmitted signal by minimizing the mean

[1] This prior information is given by the output of the channel decoder. When no feedback is available from the decoder and the transmitted symbols are assumed to be equiprobable, this prior is normally set to a uniform distribution.

square error (MSE) between the transmitted signal and its estimation or, equivalently,

$$\widehat{\mathbf{u}} = E(\mathbf{u}|\mathbf{y}) = \int \mathbf{u} p(\mathbf{u}|\mathbf{y}) \mathrm{d}\mathbf{u}. \tag{1.4}$$

The covariance matrix of this estimator can be computed as

$$\mathbf{C}_{\mathbf{u}|\mathbf{y}} = \int (\mathbf{u} - E(\mathbf{u}|\mathbf{y}))(\mathbf{u} - E(\mathbf{u}|\mathbf{y}))^{\mathrm{H}} p(\mathbf{u}|\mathbf{y}) \mathrm{d}\mathbf{u}. \tag{1.5}$$

Both estimators require the computation of the posterior pdf in (1.3), whose close-form analytical expression may not be obtained (except for the simplest types of probabilistic models, such as Gaussians), or it might be unfeasible in terms of complexity. These two approaches in (1.2) and (1.4)-(1.5), where the exact posterior is employed, are usually referred to as *exact inference* [4, 31, 26].

When the analytical expression of the posterior cannot be obtained, it is necessary to resort to approximation schemes, referred to as *approximate inference*. One common assumption is that the transmitted and received signals are jointly Gaussian,

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{CN}\left(\begin{bmatrix} \mathbf{u} \\ \mathbf{y} \end{bmatrix} : \begin{bmatrix} \mathbf{m}_{\mathbf{u}} \\ \mathbf{m}_{\mathbf{y}} \end{bmatrix}, \begin{bmatrix} \mathbf{C}_{\mathbf{u}} & \mathbf{C}_{\mathbf{uy}} \\ \mathbf{C}_{\mathbf{yu}} & \mathbf{C}_{\mathbf{y}} \end{bmatrix} \right), \tag{1.6}$$

yielding a Gaussian and tractable posterior pdf whose mean vector and covariance matrix are given by

$$E(\mathbf{u}|\mathbf{y}) = \mathbf{m}_{\mathbf{u}} + \mathbf{C}_{\mathbf{uy}} \mathbf{C}_{\mathbf{y}}^{-1} (\mathbf{y} - \mathbf{m}_{\mathbf{y}}), \tag{1.7}$$

$$\mathbf{C}_{\mathbf{u}|\mathbf{y}} = \mathbf{C}_{\mathbf{u}} - \mathbf{C}_{\mathbf{uy}} \mathbf{C}_{\mathbf{y}}^{-1} \mathbf{C}_{\mathbf{yu}}. \tag{1.8}$$

This approximation implies that $\mathbf{u} \sim \mathcal{CN}(\mathbf{u} : \mathbf{m}_{\mathbf{u}}, \mathbf{C}_{\mathbf{u}})$. Hence, it follows that the approximation to the posterior in (1.3) is distributed as

$$q(\mathbf{u}) \propto \mathcal{CN}(\mathbf{y} : \mathbf{H}\mathbf{u}, \mathbf{C}_{\mathbf{w}})\, \mathcal{CN}(\mathbf{u} : \mathbf{m}_{\mathbf{u}}, \mathbf{C}_{\mathbf{u}}) \sim \mathcal{CN}\big(\mathbf{u} : \widehat{\mathbf{u}}, \mathbf{C}_{\mathbf{u}|\mathbf{y}}\big) \tag{1.9}$$

and the estimator yields

$$\widehat{\mathbf{u}} = \mathbf{m}_{\mathbf{u}} + \mathbf{C}_{\mathbf{u}} \mathbf{H}^{\mathrm{H}} \left(\mathbf{H} \mathbf{C}_{\mathbf{u}} \mathbf{H}^{\mathrm{H}} + \mathbf{C}_{\mathbf{w}} \right)^{-1} (\mathbf{y} - \mathbf{H}\mathbf{m}_{\mathbf{u}}), \tag{1.10}$$

$$\mathbf{C}_{\mathbf{u}|\mathbf{y}} = \mathbf{C}_{\mathbf{u}} - \mathbf{C}_{\mathbf{u}} \mathbf{H}^{\mathrm{H}} \left(\mathbf{H} \mathbf{C}_{\mathbf{u}} \mathbf{H}^{\mathrm{H}} + \mathbf{C}_{\mathbf{w}} \right)^{-1} \mathbf{H} \mathbf{C}_{\mathbf{u}}. \tag{1.11}$$

This estimation is commonly referred to as linear minimum mean square error (LMMSE).

As mentioned before, in the context of digital communications, the prior information of the transmitted symbols follows a discrete distribution. Therefore, assuming a Gaussian prior distribution for these symbols degrades their estimation. The goal of this thesis lies on the design of alternative approximate statistical techniques that will be applied for the design of digital probabilistic receivers for communication systems to provide a good compromise between performance and computational complexity. Specifically, the expectation propagation (EP) algorithm is investigated to find the Gaussian posterior pdf that minimizes the Kullback-Leibler (KL) divergence with respect to the true posterior pdf [33, 32, 53, 4, 59].

In this thesis, different novel designs of a probabilistic EP-based receiver are proposed for both SISO and MIMO scenarios. These receivers greatly outperform the performance of the LMMSE and previous approaches found in the literature, with a complexity slightly increased but still proportional to the LMMSE one. These novel receivers show a robust behavior regardless of the order of the modulation, the length of the channel and the number of antennas. The scope of this thesis encompasses time domain receivers and linear time invariant (LTI) systems. The proposed EP algorithms are part of the first subsystem of the digital receiver of Figure 1.1. In the SISO scenario, this subsystem is an equalizer while in the MIMO scenario it is a detector.

1.2 Thesis overview

In this thesis, we deal with two different scenarios of a communication system: a SISO system with memory channels and a MIMO system with memoryless channels. The literature about digital receivers employed in both scenarios is investigated in Chapter 2. The main part of this thesis is focused on the SISO scenario where several EP-based equalizers are proposed. In Chapter 3 we focus on the standalone equalizer to later extended it for turbo equalization in Chapter 4. Specifically, three different designs of a soft equalizer based on the EP algorithm are developed: the block or batch approach, the filter-type version that emulates the Wiener filter behavior and the smoothing equalizer that proceeds similarly to a Kalman filter. These designs were developed as part of a collection of published scientific articles listed below, including the Appendix where they can be found in this thesis:

- Appendix A: Irene Santos, Juan José Murillo-Fuentes, Rafael Boloix-Tortosa, Eva Arias de Reyna and Pablo M. Olmos, "Expectation Propagation as Turbo Equalizer in ISI Channels," IEEE Transactions on Communications, vol. 65, no.1, pp. 360-370, Jan 2017. [50]

In this paper, the block/batch approach for the EP equalizer is developed. Since the computational complexity of the algorithm is dominated by the inversion of a matrix, its structure is exploited to reduce this computation. Specifically, its complexity is $\mathcal{O}(S\mathtt{N}^2\mathtt{L})$, where $\mathtt{N}$ is the length of the transmitted frame, $\mathtt{L}$ the length of the channel and S is the number of iterations of the EP algorithm.

- Appendix B: Irene Santos, Juan José Murillo-Fuentes, Eva Arias de Reyna and Pablo M. Olmos, "Turbo EP-based Equalization: a Filter-Type Implementation," IEEE Transactions on Communications, Sep 2017, Accepted. [Online] Available: https://ieeexplore.ieee.org/document/8353388/ [48]

  In this paper, the previous block proposal is reviewed and optimized for turbo equalization. A Wiener filter-type EP-based equalizer is also developed, whose complexity is dominated by the inversion of a matrix with size given by the length of the window, $\mathtt{W}$, i.e., $\mathcal{O}(S\mathtt{N}\mathtt{W}^2)$. This solution reduces the computational complexity compared to the block design at the cost of a degradation in performance. Finally, the EP parameters are revised to reduce the number of EP iterations in a turbo scheme.

- Appendix C: Irene Santos, Juan José Murillo-Fuentes, Eva Arias-de-Reyna and Pablo M. Olmos, "Probabilistic Equalization With a Smoothing Expectation Propagation Approach," IEEE Transactions on Wireless Communications, vol. 16, no. 5, pp. 2950-2962, May 2017. [49]

  In this paper, a smoothing Kalman implementation of the EP-based equalizer is proposed. Its computational complexity is dominated by the inversion of a matrix whose size is given by the length of the window used. It achieves the same performance as the block design with lower complexity given by $\mathcal{O}(S\mathtt{N}\mathtt{L}^3)$.

These three different designs have one characteristic in common: they use the EP algorithm to approximate with Gaussians the posterior pdf at the output of the equalizer. This strategy is different to what can be found in the literature [58, 24, 66] where the EP is used to pass the discrete information from the channel decoder to the LMMSE equalizer, not improving the solution of a standalone equalizer without decoder. To take the advantages of both approaches, in Chapter 4 they are combined into a novel double EP-based approach. Chapter 5 focuses on the MIMO scenario and develops a block EP-based turbo receiver following the guidelines of our previous works on turbo equalization. This novel receiver outperforms previous approaches found in the literature [5, 8, 54]. It has also been submitted to a journal, where it is currently under review:

- Appendix D: Irene Santos and Juan José Murillo-Fuentes, "EP-based turbo detection for MIMO receivers and large-scale systems," IEEE Transactions on Vehicular Technology, May 2018, Submitted. [Online] Available: https://arxiv.org/abs/1805.05065 [46]

  In this paper, the EP algorithm is applied to MIMO systems in a block form and a turbo scheme. Based on our previous articles in equalization, we employed the revised EP parameters to reduce the computational complexity of the algorithm.

Conclusions and future lines of research are included in Chapter 6. Finally, Appendix A to Appendix D contain the main scientific contributions of this dissertation [50, 48, 49, 46] and Appendix E includes some related conference papers.

2 State of the art

2.1 Standalone and turbo equalization

Soft equalizers provide a posterior pdf of the transmitted symbols given a set of observations, from which nowadays channel decoders highly benefit. In addition, this soft information can be further refined by feeding the output of the channel decoder back to the probabilistic equalizer. This new information is considered by the equalizer as an updated prior information about the transmitted symbols. This exchange of information between the equalizer and the channel decoder is known as turbo equalization. If no turbo equalization is performed, we refer to the equalization step as standalone equalization.

2.1.1 Optimal equalization

Optimal equalization methods must minimize the symbol error rate (SER), which can be based on MAP estimations. One efficient MAP method is the BCJR algorithm [1], which requires perfect knowledge of the channel state information (CSI). The BCJR works on a trellis representation defined by states and computes the transition probability between consecutive trellis states in three different steps [29]. First, the forward filtering step computes the probability from a state to the next one, given a new observation through the trellis. Second, a similar filtering step is performed backwards. Finally, a smoothing step computes the APP for each symbol from the transition probabilities of both filtering procedures. However, the BCJR becomes unfeasible for multilevel modulations and/or few taps of the channel since its computational complexity is proportional to the number of trellis states, $\mathtt{M}^{\mathtt{L}}$, where $\mathtt{M}$ is the constellation size and $\mathtt{L}$ the number of taps of the channel.

2.1.2 Approximations to the BCJR

A variety of solutions can be found in the literature to reduce the complexity of the trellis diagram used by the BCJR algorithm, at the expense of performance loss [10, 19, 18, 63, 55]. Instead of considering the full trellis, they perform a simplified trellis search with only $M_e \ll \mathtt{M}^{\mathtt{L}}$ states. They can be divided into two different families. The first one is based on truncating the number of trellis states by keeping only the states with highest forward (or backward) metric or whose forward (or backward) metric is above a predefined threshold, which is the case of the M-BCJR and T-BCJR algorithms [19], respectively. Basically, they perform a reduced search on the original full trellis. On the other hand, the second family consists on truncating the effective length of the CSI by cancelling the last channel taps (retaining just $\mathtt{L}' < \mathtt{L}$ taps) and therefore reducing the number of states to $M_e = \mathtt{M}^{\mathtt{L}'}$. In contrast to the first family, this one performs a full search on a reduced-state trellis. This is the case of the channel shortening (CS) approaches, such as [43], and the reduced-state BCJR (RS-BCJR) algorithm [55], originally inspired by the reduced-state sequence detection (RSSD) [16, 15, 9]. To enhance the results of both trends, a posterior approach mixes both families into the M*-BCJR algorithm [10], which keeps the states with highest forward metric, as the M-BCJR does, but instead of deleting the remaining states, they are merged into the surviving states, similarly to the RS-BCJR strategy.

The previous approaches have an important limitation: the surviving paths cannot be computed by the forward and backward procedures independently, since these paths generally do not match in both procedures. To overcome this situation, they give a predominant role to the forward recursion, so that the backward stage just follows the subset of paths previously selected in the forward stage. For this reason, they might not succeed when equalizing maximum-phase channels. On the contrary, if they give a predominant role to the backward recursion, instead of the forward one, they do not succeed with minimum-phase channels [18]. To solve the previous limitation, a different state selection criterion is proposed in [63]. Instead of selecting the states with largest metric as the M-BCJR algorithm, they keep the states with largest values of its marginal APP and repeat the forward/backward procedure allowing different sets of surviving states at each iteration. The performance dependence of these approaches with the type of channel is deeply studied in [18]. The authors propose equalizing according to the minimum, maximum or mixed phase nature of the channel, resorting to the forward-trellis (FT), backward-trellis (BT) or a mixture of them named doubled-trellis (DT), respectively. The authors also mention that all the approximated BCJR methods select the most promising paths while performing the reduced search over the trellis, i.e., they ignore some paths in the trellis. For this reason, the estimation of their decisions are normally

considered more reliable than they really are. To mitigate this overestimation, they introduce an output saturation to their previous proposals [18].

All these approximated BCJR approaches suffer from two major drawbacks. Firstly, they are usually designed and tested for just some specific channels and the parameters are selected according to those channels and the simulated scenario. In general, no information is given about how to tune these parameters for other channels. Secondly, they are intractable for large trellises since their performance degrades if the number of surviving states, M_e, does not grow accordingly with the total number of states, i.e, if the ratio $\mathtt{M}^{\mathtt{L}-1}/M_e$ is not kept constant at a relatively high value. To the best of our knowledge, this issue has not been studied and the value for this ratio to ensure an accurate enough performance remains as an open question. For these reasons, these approximated BCJR approaches are only interesting for a moderate number of states. For large number of states, it is preferable to envisage filter-based equalizers of the MMSE type [2] or message-passing approaches [12, 11, 21, 24, 58, 66].

2.1.3 The LMMSE as equalizer

The LMMSE algorithm is commonly used for soft equalization since it provides a suboptimal but low-cost and robust estimation even for multidimensional modulations. It is derived for a model where outputs and inputs are jointly Gaussian, as discussed in Section 1.1. Hence, the LMMSE is an approximated solution that minimizes the MSE between the transmitted and detected symbol. This solution is actually the same for any observation, i.e., it exhibits the same linear filter for any given observation, as long as the channel response is fixed. Since the LMMSE considers the inputs as Gaussians, it implies that the marginals for the inputs (the priors) are also Gaussians and, in turbo equalization, they are updated with statistics according to the channel decoder output. If no information is available from the channel decoder, equiprobable a priori probabilities (priors) are commonly assumed for the symbols in the constellation, which is equivalent to setting the mean and variance for the initial Gaussian priors to the mean and energy of the modulation used. The computational complexity of its block or batch implementation is quadratic in the frame length and linear in the length of the channel, $\mathcal{O}(\mathtt{N}^2\mathtt{L})$, which requires huge computational resources for large frames [61, 34]. To reduce this complexity, a filter-based LMMSE approach that emulates a Wiener-type implementation is proposed in [61, 62, 60]. It works as a forward LMMSE filter that processes the observations within the observed sliding window, rather than the complete sequence as in its batch implementation. In [30], a different recursive procedure is proposed by replacing the sliding windows with an extending one. The complexity of these filter-based LMMSE approaches is linear in the frame

length and quadratic in the length of the window, $\mathcal{O}(\mathtt{N}\mathtt{W}^2)$. However, their performance degrade in comparison with their block counterpart. The complexity of the filter-based LMMSE can be further reduced to be linear in the length of the window by relying on some approximations, at the expense of deteriorating its performance [61, 62, 60]. To solve this degradation in the performance and, at the same time, reduce the complexity of the block LMMSE, a Kalman smoothing implementation can be employed [38]. This Kaman smoother achieves the same performance than the block LMMSE with computational complexity given by $\mathcal{O}(\mathtt{N}\mathtt{L}^3)$. It can be seen as a BCJR with Gaussian inputs.

All these three implementations of the LMMSE algorithm, i.e., the block, the Wiener filter and the Kalman smoother, are quite popular due to their low complexity, which is also independent of the constellation order, allowing multidimensional constellations without increasing the complexity. However, their performance are usually far from the optimal solution.

2.1.4 Message-passing equalization

Message-passing solutions have also been applied to equalization. In [12], the authors develop an equalizer based on the sum-product algorithm (SPA) that is applied to a factor graph that represents the joint APP of the transmitted symbols. This factor graph leads to cycles of length four unless certain conditions are met that allow a girth of at least six and converges to a good approximation. However, its complexity is still exponential in the number of non-zero interferers, solving the computational issue just for sparse ISI channels. Inspired by the previous algorithm, in [11] the loopy belief propagation (BP) is applied to the graphical model of the system, which is rewritten to end with a graph with girth equal or larger than six that leads to good results, and reduces the complexity to be linear in the number of interferers, instead of exponential as in [12]. How complexity scales with the modulation size is not addressed in these works, but it seems to be quadratic. The complexity also depends linearly with the number of iterations of the SPA, but this number is not specified. A different approach is proposed in [21], where Gaussian message passing (GMP) is applied to a tree-structured graph that avoids the short cycle problem explained above. They prove that the algorithm is equivalent to the LMMSE from a GMP point of view with quadratic complexity in the length of the channel.

2.1.5 EP-based equalization

The previous GMP is improved in [24, 58, 66] where the authors use EP [33, 32, 53] to incorporate the non-Gaussian messages from the channel decoder into an improved Gaussian approximated message. Then, equalization is performed by a

forward-backward Gaussian message passing, i.e., a Kalman smoothing. In other words, the approaches in [24, 58, 66] share the following characteristics:

- They apply GMP to the factor graph of the system.
- They use the EP algorithm at the output of the decoder to improve the Gaussian approximate messages given back to the equalizer.
- The equalization stage is based on a Kalman smoothing implementation.

At this point, it is interesting to point out that [24, 58] do not improve the equalization step by itself. They are turbo equalization approaches which boil down to the LMMSE for standalone equalization, i.e., if no turbo equalization is carried out their performance is that of the LMMSE. These approaches differ in the procedure to tackle the instabilities during the EP updates. In [58, 66] the authors solve the problem of negative variances by taking their absolute values while in [24] they use a damping procedure to avoid instabilities. This damping mechanism is based on a geometrical mixture of the standard Gaussian approximation and the EP Gaussian approximation. The approach [66] is an improved version of [58], where a discrete distribution instead of a Gaussian one is sent from the equalizer to the decoder. However, its complexity increases exponentially with the constellation size. It is also important to remark that approaches in [21, 58, 24, 43, 11, 66] are tested just with BPSK transmissions and some specific channels, i.e., no information is available on how to extend these methods to non-binary constellations or the performance for random channels. Since these approaches are formulated just for BPSK transmissions, they do not include any mechanism to control instabilities for multidimensional constellations, then their performance is expected to degrade when using high-order modulations. As a result, the EP-based approaches in [58, 24, 66] are expected to suffer from instabilities and convergence problems [8, 5, 37].

In this thesis different novel schemes of an EP-based probabilistic standalone equalizer (Chapter 3) and turbo equalizer (Chapter 4) are proposed to greatly outperform the performance of the LMMSE and previous approaches found in the literature, with a complexity slightly increased but still proportional to that of the LMMSE. These EP-based equalizers show a quite robust behavior regardless of the order modulation and the channel response.

2.2 MIMO detection

In a MIMO system, where N_t transmit antennas and N_r receive antennas are employed, the soft detector provides the posterior distribution of the transmitted

symbols given the received signal. This soft estimation is then driven to the channel decoder. In turbo detection, the output of the channel decoder feeds the detector back and is used as a new updated prior for the transmitted symbols.

One possible optimal detection method is based on the MAP algorithm, which finds the set of transmitted symbols that maximizes the APP. However, its complexity is proportional to the set of possible transmitted words, i.e., it suffers from an exponential complexity in the number of transmit antennas and the size of the constellation used. Therefore it becomes unfeasible for medium or large scale systems. In this scenario, suboptimal approaches with lower complexity are employed. The sphere decoding (SD) method provides an approximation to the marginal posterior pdf in a subspace of the whole set of possible transmitted words [57, 64]. However, its performance deteriorates if the dimension of the subspace does not grow accordingly with the constellation size and the number of antennas [56]. Hence, this approach is just interesting for moderate scenarios, remaining intractable for large scale systems. Another alternative to approximate the posterior distribution is the use of Markov chain Monte Carlo (MCMC) algorithms [13], but it also requires a sufficiently large number of samples to obtain an accurate enough performance.

In this situation of large complexity scenarios, the LMMSE is a low-cost and commonly employed technique with robust estimation even for high-order constellations. Its complexity is dominated by the inversion of a matrix with size N_t, i.e., the computational cost does not depend on the size of the constellation. It computes an approximated APP by minimizing the MSE between the transmitted and detected symbol. However, its performance is far from optimal and alternative approaches with improved performance can be found in the literature. The Gaussian tree approximation (GTA) algorithm [20] constructs a tree-factorized Gaussian approximation to the APP and then estimates the marginals with the BP algorithm. The channel hardening-exploiting message passing (CHEMP) [35] is a message-passing approach where all the exchanged messages are approximated by Gaussian distributions. These three approaches have one characteristic in common: they assume a Gaussian prior for the transmitted symbols.

The assumption of Gaussian priors limits the performance of the above approximated approaches since they do not consider the discrete nature of symbols when estimating the APP. This restriction can be incorporated by means of the EP algorithm [33, 53], that obtains a Gaussian approximated posterior pdf conditioned to the received signal by matching its moments with the ones of the true APP. This algorithm has been already applied to standalone and turbo equalization [50, 48, 49] (as explained in Subsection 2.1.5) and detection in flat-fading channels [41, 40]. It has been also applied to MIMO detection, initially for hard detection

[6] and then extended for soft detection [7, 8]. In these works, it was shown that the EP detector was able to improve the performance of LMMSE, GTA and CHEMP with complexity proportional to the one of the LMMSE algorithm. Preliminary results on turbo MIMO detection can be found in [5, 54]. In [5] the same EP detector proposed for standalone soft detection in [8] is reused and applied to turbo detection, where the information from the decoder is just used to initialize the moments of the Gaussian prior used in the detector. However, it keeps uniform priors in the to-be-approximated posterior during the moment matching step. This EP application is done iteratively within the detector and a damping procedure is included to control the speed and the convergence of the algorithm. On the other hand, [54] characterizes the true priors used in the moment matching procedure with non-uniform distributions but no EP iterations within the detector are considered. Since a non-uniform prior for the symbols describes better the information returned by the decoder, this approach improves the performance in [5]. In this thesis (Chapter 5), the performance of both proposals is improved by considering non-uniform priors within the EP application in the detector and repeating this procedure iteratively.

3 Standalone equalization

As introduced in Subsection 2.1.5, previous equalization solutions based on the EP algorithm reduce to the LMMSE for standalone equalization [24, 58]. In addition, they have been only implemented from a GMP point of view and a Kalman strategy, not dealing with other designs such as a block or filter one. In this thesis the EP algorithm is investigated to propose several equalizer designs that share the structure of the LMMSE conditioned to the received symbol. The proposed designs follow the three different approaches described in Subsection 2.1.3: a block, Wiener filter and Kalman smoother type. This chapter is focused on the behavior of an isolated probabilistic equalizer with no prior information provided by the decoder. Then, in Chapter 4, we study how to improve this solution with the aid of turbo equalization.

3.1 Problem to solve

Nowadays digital communication systems are required to provide high-speed data and high-quality services when transmitting over communication channels. Along the propagation process, the digital transmission is corrupted by additive white Gaussian noise (AWGN) which negatively affects the received signal. In addition, the dispersive nature of channels and the multiple propagation paths of wireless communications degrade the transmission and introduce ISI [39, 22]. The process of mitigating/reversing the effect of ISI is known as channel *equalization* [52]. Channel equalizers provide an estimation of the transmitted sequence of symbols from a received corrupted signal. To protect the transmitted sequence from errors, a controlled amount of redundancy is introduced by means of an *encoder/decoder*. Hence, the receiver task is double: firstly, it processes the received corrupted signal to combat the effects of ISI and provides an estimation of the transmitted sequence

of symbols (equalization) and secondly, it recovers the data from the equalized symbols by exploiting the structure of the code used (decoding). Nowadays channel decoders highly benefit from *soft* (probabilistic) estimates for each transmitted symbol, improving the BER performance significantly [45].

3.1.1 System model

The model of a SISO digital communication system is represented in Figure 3.1. It can be divided into three main blocks:

- Transmitter: The information sequence of bits, $\mathbf{a} = [a_1,...,a_{\mathtt{K}}]^\top$ with $a_i \in \{0,1\}$, is encoded into the coded bit vector $\mathbf{b} = [b_1,...,b_{\mathtt{V}}]^\top$ with a code rate equal to $\mathtt{R} = \mathtt{K}/\mathtt{V}$. After permuting the bits with an interleaver, if needed, the codeword $\mathbf{c} = [c_1,...,c_{\mathtt{V}}]^\top$ is obtained. This codeword is partitioned into $\mathtt{N}$ blocks of length $Q = \log_2(\mathtt{M})$, $\mathbf{c} = [\mathbf{c}_1,...,\mathbf{c}_{\mathtt{N}}]^\top$ where $\mathbf{c}_k = [c_{k,1},...,c_{k,Q}]$, and each $\mathbf{c}_k$ is modulated with a complex $\mathtt{M}$-ary constellation with alphabet $\mathcal{A}$. Then, the modulated symbols, $\mathbf{u} = [u_1,...,u_{\mathtt{N}}]^\top$, where each component $u_k = \mathcal{R}(u_k) + j\mathcal{I}(u_k) \in \mathcal{A}$, are transmitted over the channel.

- Channel: The channel is completely specified by the CSI, i.e., $\mathbf{h} = [h_1,...,h_{\mathtt{L}}]^\top$, where $\mathtt{L}$ is the number of taps, and is corrupted with AWGN whose noise variance, σ_w^2, is known. Each k-th entry of the complex received signal, $\mathbf{y} = [y_1,...,y_{\mathtt{N}+\mathtt{L}-1}]^\top$, is given by

$$y_k = \sum_{j=1}^{\mathtt{L}} h_j u_{k-j+1} + w_k \tag{3.1}$$

  where h_j does not change during the transmission of the $\mathtt{N}$ symbols, $w_k \sim \mathcal{CN}\left(w_k : 0, \sigma_w^2\right)$ and $u_k = 0$ for $k < 1$ and $k > \mathtt{N}$. Equivalently, the system can be expressed in block form as

$$\underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_{\mathtt{N}+\mathtt{L}-1} \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} h_1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ h_{\mathtt{L}} & & \ddots & 0 \\ 0 & \ddots & & h_1 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & h_{\mathtt{L}} \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} u_1 \\ \vdots \\ u_{\mathtt{N}} \end{bmatrix}}_{\mathbf{u}} + \underbrace{\begin{bmatrix} w_1 \\ \vdots \\ w_{\mathtt{N}+\mathtt{L}-1} \end{bmatrix}}_{\mathbf{w}}, \tag{3.2}$$

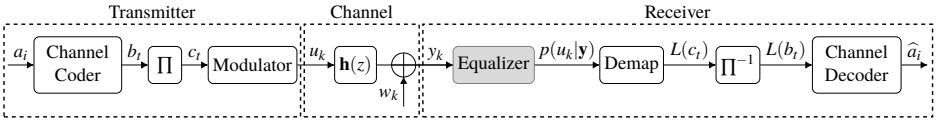


Figure 3.1 System model of a SISO communication system.

that fits the model already introduced in (1.1). In this thesis, the channel response is assumed to be perfectly known.

- Receiver: As in (1.3), the posterior probability of the transmitted symbol vector $\mathbf{u}$ given the whole vector of observations $\mathbf{y}$ yields

$$p(\mathbf{u}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{u})p(\mathbf{u})}{p(\mathbf{y})} \propto \mathcal{CN}\big(\mathbf{y} : \mathbf{Hu}, \sigma_w^2\mathbf{I}\big) \prod_{k=1}^{\mathtt{N}} p(u_k) \tag{3.3}$$

where, assuming equiprobable symbols, the prior is given by

$$p(u_k) = \frac{1}{\mathtt{M}} \sum_{u\in\mathcal{A}} \delta(u_k - u) = \frac{1}{\mathtt{M}} \mathbb{I}_{u_k\in\mathcal{A}} \tag{3.4}$$

and $\mathbb{I}_{u_k\in\mathcal{A}}$ is the indicator function that takes value 1 if $u_k \in \mathcal{A}$ and zero in other case. The posterior in (3.3) is then demapped, deinterleaved if needed and given to the channel decoder. Finally, the channel decoder computes an estimation for the transmitted symbols, $\widehat{a}_i$.

3.2 EP-based equalizers

As stated in (3.3), the true posterior $p(\mathbf{u}|\mathbf{y})$ (or its marginal $p(u_k|\mathbf{y})$) is clearly non-Gaussian and analytically intractable due to the discrete nature of the transmitted symbols. Specifically, when these symbols are equiprobable and no information from the channel decoder is available, their priors are given by (3.4). Instead of using this true discrete prior as the optimal algorithm does, the LMMSE allows for a low-cost solution by approximating it with Gaussians as introduced in (1.9). In other words, it computes a Gaussian approximate posterior as

$$q(\mathbf{u}) = \mathcal{CN}\big(\mathbf{y} : \mathbf{Hu}, \sigma_w^2\mathbf{I}\big) \prod_{k=1}^{\mathtt{N}} \tilde{p}(u_k) \tag{3.5}$$

where $\tilde{p}(u_k)$ is a Gaussian approximation of the true prior $p(u_k)$ in (3.4), whose moments are set to the ones of the prior in (3.4). This is denoted as

$$\tilde{p}(u_k) = \text{Proj}_G[p(u_k)], \tag{3.6}$$

where $\text{Proj}_G[p(u_k)]$ is the projection of $p(u_k)$ into the family of Gaussians, which means that $\tilde{p}(u_k)$ has the same mean and variance than $p(u_k)$. Note that (3.5) is a Gaussian distribution whose mean vector and covariance matrix are given by (1.10) and (1.11), respectively, since the prior has been assumed to be Gaussian.

However, the LMMSE does not use the available information of the discrete nature of symbols. This restriction can be incorporated with an approximate inference method used in Bayesian machine learning, named EP algorithm [33, 32, 53, 4, 59]. This algorithm is based on approximating an analytically intractable or prohibitively complex (non-exponential) distribution, such as (3.3), with a Gaussian distribution by minimizing the KL divergence. Since the approximated distribution is Gaussian, the process of minimizing the KL divergence is equivalent to matching its expected sufficient statistics to the corresponding moments of the true distribution. This process is known as *moment matching*.

As the LMMSE, the EP algorithm replaces the true prior with Gaussian factors, denoted as $\tilde{p}^{[\ell]}(u_k)$, to iteratively obtain a Gaussian approximate posterior given by

$$q^{[\ell]}(\mathbf{u}) = \mathcal{CN}\big(\mathbf{y} : \mathbf{H}\mathbf{u}, \sigma_w^2\mathbf{I}\big) \prod_{k=1}^{\mathtt{N}} \tilde{p}^{[\ell]}(u_k) \sim \mathcal{CN}\big(\mathbf{u} : \boldsymbol{\mu}^{[\ell]}, \mathbf{S}^{[\ell]}\big)\,. \tag{3.7}$$

Then, it optimizes each factor $\tilde{p}^{[\ell]}(u_k)$ in turn independently and in the context of all the remaining ones by matching the moments of the discrete posterior,

$$\widehat{p}^{[\ell]}(u_k) \propto q_E^{[\ell]}(u_k) p(u_k), \tag{3.8}$$

to the ones of its approximation, $q_E^{[\ell]}(u_k)\tilde{p}^{[\ell+1]}(u_k)$, i.e.,

$$\overbrace{q_E^{[\ell]}(u_k)p(u_k)}^{\widehat{p}^{[\ell]}(u_k)} \quad \begin{matrix}\text{moment}\\ \text{matching}\\ \longleftrightarrow\end{matrix} \quad q_E^{[\ell]}(u_k)\tilde{p}^{[\ell+1]}(u_k) \tag{3.9}$$

resulting

$$\tilde{p}^{[\ell+1]}(u_k) = \frac{\text{Proj}_G[\widehat{p}^{[\ell]}(u_k)]}{q_E^{[\ell]}(u_k)} = \frac{\mathcal{CN}\Big(u_k : \mu_{\widehat{p}_k}^{[\ell]}, s_{\widehat{p}_k}^{2[\ell]}\Big)}{\mathcal{CN}\Big(u_k : z_k^{[\ell]}, v_k^{2[\ell]}\Big)} \sim \mathcal{CN}\Big(u_k : m_k^{[\ell+1]}, \eta_k^{2[\ell+1]}\Big)\,, \tag{3.10}$$

where $\mu_{\widehat{p}_k}^{[\ell]}$ and $s_{\widehat{p}_k}^{2[\ell]}$ are the mean and variance of the discrete posterior $\widehat{p}^{[\ell]}(u_k)$. The Gaussian distribution $q_E^{[\ell]}(u_k)$ used in (3.10) is known as *extrinsic distribution* and it is the resulting distribution after removing from the posterior distribution the factor to be updated,

$$q_E^{[\ell]}(u_k) = \frac{q^{[\ell]}(u_k)}{\tilde{p}^{[\ell]}(u_k)} = \frac{\mathcal{CN}\left(u_k : \mu_k^{[\ell]}, s_k^{2[\ell]}\right)}{\mathcal{CN}\left(u_k : m_k^{[\ell]}, \eta_k^{2[\ell]}\right)} \sim \mathcal{CN}\left(u_k : z_k^{[\ell]}, v_k^{2[\ell]}\right), \tag{3.11}$$

where $q^{[\ell]}(u_k)$ is the k-th marginal of the Gaussian approximate posterior in (3.7) and

$$z_k^{[\ell]} = \frac{\mu_k^{[\ell]} \eta_k^{2[\ell]} - m_k^{[\ell]} s_k^{2[\ell]}}{\eta_k^{2[\ell]} - s_k^{2[\ell]}}, \tag{3.12}$$

$$v_k^{2[\ell]} = \frac{s_k^{2[\ell]} \eta_k^{2[\ell]}}{\eta_k^{2[\ell]} - s_k^{2[\ell]}}. \tag{3.13}$$

We include Table 3.1 to summarize the distributions defined above.

Table 3.1 Gaussian distributions in an EP-based equalizer.

Distribution	Mean	Variance	Description
$\widehat{p}^{[\ell]}(u_k)$	$\mu_{\widehat{p}_k}^{[\ell]}$	$s_{\widehat{p}_k}^{2[\ell]}$	Discrete posterior defined in (3.8)
$q_E^{[\ell]}(u_k)$	$z_k^{[\ell]}$	$v_k^{2[\ell]}$	Approximated extrinsic distribution defined in (3.11)
$\tilde{p}^{[\ell]}(u_k)$	$m_k^{[\ell]}$	$\eta_k^{2[\ell]}$	EP Gaussian factor defined in (3.10)
$q^{[\ell]}(u_k)$	$\mu_k^{[\ell]}$	$s_k^{2[\ell]}$	k-th marginal of the approximated full posterior defined in (3.7)

At this point it is interesting to remark that the EP uses a Gaussian approximate posterior, as the LMMSE does. However, while the LMMSE assumes a Gaussian pdf that best fits the priors, i.e., the constellation, the EP uses Gaussian approximations to better fit the posterior given the observations, hence being non-linear. It can be seen as an LMMSE with an iterative feedback provided by the EP algorithm to exploit the discrete nature of the transmitted symbols, as represented in Figure 3.2 (the superscript $[\ell]$ has been removed from the figure for simplicity).

The Gaussian factor in (3.10) is given back to the LMMSE that will compute

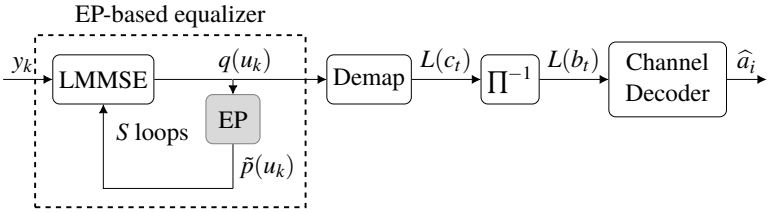


Figure 3.2 EP-based receiver diagram.

a new approximate posterior distribution following (3.7). The previous moment matching procedure is done iteratively along $\ell = 1,...,S$ iterations, to let the algorithm achieve a more accurate solution for the posterior in (3.7) before handling it to the channel decoder. The EP updates in (3.10) can lead to the assignment of negative values to the variances that must be controlled. Following the guidelines in [8, 5, 37], the variance of (3.10) is not updated (keeping its previous value) whenever the algorithm provides negative variance. In addition, the EP algorithm suffers from instabilities, specially with multidimensional modulations and large values of E_b/N_0. For this reason, some parameters are introduced to control the accuracy and the speed of the algorithm. Specifically, these parameters are:

- A minimum allowed variance, 'ε'.
- A damping procedure based on averaging the EP update at the previous iteration with the current update with a damping factor, 'β'.
- A maximum number of EP iterations, 'S'.

Details about how to choose the parameters of the EP algorithm can be found in Appendix A and Appendix B. Following their guidelines, in this chapter we set the parameters to the values shown in Table 3.2.

Table 3.2 Values of the EP parameters for standalone equalization.

ε	β	S
10^{-8}	0.1	10

The moment matching in (3.10) and the damping procedure (MMD) are described in Algorithm 3.1, which also includes the previously described parameters to avoid performance degradation and instabilities. This EP mechanism is performed iteratively along $\ell = 1,...,S$ iterations, to let the algorithm achieve a more

Algorithm 3.1 Moment Matching and Damping (MMD) for EP equalizer

Given inputs: The extrinsic distribution $q_E^{[\ell]}(u_k)$ with moments $z_k^{[\ell]}$ and $v_k^{2[\ell]}$, the true prior $p(u_k)$, the Gaussian EP approximation $\tilde{p}^{[\ell]}(u_k)$ with moments $m_k^{[\ell]}$ and $\eta_k^{2[\ell]}$, β and ε

1) Run moment matching:

   1.1) Estimate the mean $\mu_{\widehat{p}_k}^{[\ell]}$ and variance $s_{\widehat{p}_{k,aux}}^{2[\ell]}$ of the discrete posterior distribution $\widehat{p}^{[\ell]}(u_k)$ defined in (3.8) and set a minimum allowed variance $s_{\widehat{p}_k}^{2[\ell]} = \max(\varepsilon, s_{\widehat{p}_{k,aux}}^{2[\ell]})$.

   2.2) Set them to the mean and variance of the unnormalized Gaussian distribution $q_E^{[\ell]}(u_k)\mathcal{CN}\Big(u_k : m_{k,new}^{[\ell+1]}, \eta_{k,new}^{2[\ell+1]}\Big)$, to get the solution

$$\eta_{k,new}^{2[\ell+1]} = \frac{s_{\widehat{p}_k}^{2[\ell]} v_k^{2[\ell]}}{v_k^{2[\ell]} - s_{\widehat{p}_k}^{2[\ell]}}, \qquad m_{k,new}^{[\ell+1]} = \eta_{k,new}^{2[\ell+1]} \left(\frac{\mu_{\widehat{p}_k}^{[\ell]}}{s_{\widehat{p}_k}^{2[\ell]}} - \frac{z_k^{[\ell]}}{v_k^{2[\ell]}} \right). \quad (3.14)$$

2) Run the damping procedure: Update the values as

$$\eta_k^{2[\ell+1]} = \left(\beta \frac{1}{\eta_{k,new}^{2[\ell+1]}} + (1-\beta) \frac{1}{\eta_k^{2[\ell]}} \right)^{-1}, \quad (3.15)$$

$$m_k^{[\ell+1]} = \eta_k^{2[\ell+1]} \left(\beta \frac{m_{k,new}^{[\ell+1]}}{\eta_{k,new}^{2[\ell+1]}} + (1-\beta) \frac{m_k^{[\ell]}}{\eta_k^{2[\ell]}} \right). \quad (3.16)$$

3) Control of negative variances:

   if $\eta_k^{2[\ell+1]} < 0$ **then**

$$\eta_k^{2[\ell+1]} = \eta_k^{2[\ell]}, \quad m_k^{[\ell+1]} = m_k^{[\ell]}. \quad (3.17)$$

   end if

Output: $\eta_k^{2[\ell+1]}, m_k^{[\ell+1]}$

accurate solution for the posterior in (3.7) before handling it to the channel decoder, as summarized in Algorithm 3.2, where the input $p(u_k)$ is given by (3.4). In order to use the Algorithm 3.2, a closed-form expression must be provided for the mean and variance of the extrinsic distribution in (3.11), i.e., $z_k^{[\ell]}$ and $v_k^{2[\ell]}$. These

Algorithm 3.2 Generic standalone EP-based Equalizer

Given inputs: y_k for $k = 1,\ldots,\mathtt{N}+\mathtt{L}-1$ and $p(u_k)$ for $k = 1,\ldots,\mathtt{N}$.
1) Obtain the mean $m_k^{[1]}$ and variance $\eta_k^{2[1]}$ of $\tilde{p}^{[1]}(u_k) = \mathrm{Proj}_G[p(u_k)]$.
for $\ell = 1$,...,S **do**
  for $k = 1$,...,$\mathtt{N}$ **do**
    2) Compute the k-th extrinsic distribution, $q_E^{[\ell]}(u_k)$, in (3.11).
    3) Run the moment matching procedure in Algorithm 3.1 to obtain $\tilde{p}^{[\ell+1]}(u_k) \sim \mathcal{CN}\Big(u_k : m_k^{[\ell+1]}, \eta_k^{2[\ell+1]}\Big)$.
  end for
end for
4) With the distribution $\tilde{p}^{[S+1]}(u_k)$ obtained after the EP algorithm, calculate the extrinsic distribution $q_E^{[S+1]}(u_k)$ as in (3.11).
Output: Deliver $q_E^{[S+1]}(u_k)$ to the channel decoder for $k = 1,\ldots,\mathtt{N}$.

expressions depend on how the equalizer is implemented. In this thesis we propose three different strategies: a block (or batch) approach that uses the whole vector of observations to obtain the estimation of the transmitted symbols, a Wiener-type filtered approach that only uses the observations included in a predefined window and a forward/backward Kalman-type filtered approach and smoothing approach that merges both forward and backward estimations, emulating the BCJR behavior. Every strategy leads to different expressions for $z_k^{[\ell]}$ and $v_k^{2[\ell]}$ defined in (3.12) and (3.13), respectively, that will be incorporated at steps 2) and 4) of the Algorithm 3.2 to derive the final estimation at the output of the equalizer. The performance of these approaches will be illustrated in Section 3.3.

3.2.1 Block Expectation Propagation (BEP)

The block (or batch) EP-based equalizer approximates the APP of the transmitted symbols conditioned to the complete sequence of observations, $p(\mathbf{u}|\mathbf{y})$, in a centralized manner. Note that the Gaussian approximate posterior, $q^{[\ell]}(\mathbf{u})$, in (3.7) follows the same structure than the already explained LMMSE in (1.9), i.e., the mean vector $\boldsymbol{\mu}^{[\ell]}$ and the covariance matrix $\mathbf{S}^{[\ell]}$ in (3.7) are given by expressions like (1.10) and (1.11). Accordingly, the marginal of the Gaussian approximate posterior is obtained as

$$q^{[\ell]}(u_k) \sim \mathcal{CN}\Big(u_k : \mu_k^{[\ell]}, s_k^{2[\ell]}\Big) \tag{3.18}$$

where

$$\mu_k^{[\ell]} = m_k^{[\ell]} + \eta_k^{2[\ell]} \mathbf{h}_\mathbf{k}^{\mathrm{H}} \left(\sigma_w^2 \mathbf{I} + \mathbf{H}\mathrm{diag}(\boldsymbol{\eta}^{2[\ell]})\mathbf{H}^{\mathrm{H}}\right)^{-1} (\mathbf{y} - \mathbf{H}\mathbf{m}^{[\ell]}), \tag{3.19}$$

$$s_k^{2[\ell]} = \eta_k^{2[\ell]} - \eta_k^{4[\ell]} \mathbf{h}_\mathbf{k}^{\mathrm{H}} \left(\sigma_w^2 \mathbf{I} + \mathbf{H}\mathrm{diag}(\boldsymbol{\eta}^{2[\ell]})\mathbf{H}^{\mathrm{H}}\right)^{-1} \mathbf{h}_\mathbf{k}, \tag{3.20}$$

with $\mathbf{m}^{[\ell]} = [m_1^{[\ell]}, \ldots, m_{\mathtt{N}}^{[\ell]}]^\top$, $\boldsymbol{\eta}^{2[\ell]} = [\eta_1^{2[\ell]}, \ldots, \eta_{\mathtt{N}}^{2[\ell]}]^\top$ in (3.10) and $\mathbf{h}_\mathbf{k}$ is the k-th column of the matrix $\mathbf{H}$, previously defined in (3.2). This implementation of the EP-based equalizer is called block expectation propagation (BEP) and the whole procedure is described in Algorithm 3.3, where the true prior $p(u_k)$ is given by (3.4). Note that Algorithm 3.3 was obtained from Algorithm 3.2 by just particularizing (3.11) in Step 2) and 4) with the expressions in (3.19) and (3.20). Results for real-valued systems can be found in [51] (Section E.1) and then they were extended for complex-systems in Appendix A.

Algorithm 3.3 BEP standalone Equalizer

Given inputs: y_k for $k = 1, \ldots, \mathtt{N}+\mathtt{L}-1$ and $p(u_k)$ given by (3.4) for $k = 1, \ldots, \mathtt{N}$.
Run Algorithm 3.2 with $q_E^{[\ell]}(u_k)$ in (3.11) computed by replacing $\mu_k^{[\ell]}$ and $s_k^{2[\ell]}$ in (3.12) and (3.13) with the expressions in (3.19) and (3.20).
Output: Deliver $q_E^{[S+1]}(u_k)$ to the channel decoder for $k = 1, \ldots, \mathtt{N}$.

The computational complexity of this implementation is dominated by the inversion of the matrix in (3.19) and (3.20), whose size is $(\mathtt{N}+\mathtt{L}-1) \times (\mathtt{N}+\mathtt{L}-1)$. Note that equations (3.19) and (3.20) can be also computed as the k-th element of

$$\boldsymbol{\mu}^{[\ell]} = \mathbf{S}^{[\ell]} \left(\sigma_w^{-2} \mathbf{H}^{\mathrm{H}} \mathbf{y} + \mathrm{diag}(\boldsymbol{\eta}^{2[\ell]})^{-1} \mathbf{m}^{[\ell]}\right), \tag{3.21}$$

$$\mathbf{S}^{[\ell]} = \left(\sigma_w^{-2} \mathbf{H}^{\mathrm{H}} \mathbf{H} + \mathrm{diag}(\boldsymbol{\eta}^{2[\ell]})^{-1}\right)^{-1}, \tag{3.22}$$

respectively. The proof of this equivalence can be found in Appendix B. The matrix to invert in (3.21) and (3.22) has size $\mathtt{N} \times \mathtt{N}$. Then, the inversion can be computed with cubic complexity in the frame length, $\mathcal{O}(\mathtt{N}^3)$. To reduce the complexity of the matrix inversion, in Appendix A the banded structure of the matrix to invert along with the short length of the channel compared to $\mathtt{N}$ is exploited. This yields a quadratic complexity in the frame length and linear with the length of the channel. This computation is repeated along S iterations, resulting in a final complexity of the BEP of $\mathcal{O}((S+1)\mathtt{L}\mathtt{N}^2)$, i.e., the computational complexity of the LMMSE plus the one of repeating the EP algorithm along S iterations. A full comparison with the most relevant approximated BCJR approaches is also provided in Appendix A.

3.2.2 Filter expectation propagation (FEP)

For long frames, the computational complexity of block approaches may become unfeasible and some filtered-based solutions are preferred [2]. The proposed filter-based EP equalizer, that emulates a Wiener implementation, works forwards by processing just the observations within an observed sliding window of size $\mathtt{W} = \mathtt{W}_1 + \mathtt{W}_2 + 1$, instead of the whole sequence as in its batch implementation. In other words, instead of solving the system in (3.2), it solves

$$\mathbf{y_k} = \mathbf{H}_{\mathtt{W}}\mathbf{u_k} + \mathbf{w_k} \tag{3.23}$$

where $\mathbf{y_k} = [y_{k-\mathtt{W}_2}, ..., y_{k+\mathtt{W}_1}]^\top$, $\mathbf{u_k} = [u_{k-\mathtt{L}-\mathtt{W}_2+1}, ..., u_{k+\mathtt{W}_1}]^\top$, $\mathbf{w_k} = [w_{k-\mathtt{W}_2}, ..., w_{k+\mathtt{W}_1}]^\top$ and

$$\mathbf{H}_{\mathtt{W}} = \begin{bmatrix} h_{\mathtt{L}} & \dots & h_1 & & \mathbf{0} \\ & \ddots & & \ddots & \\ & & \ddots & & \ddots \\ \mathbf{0} & & & h_{\mathtt{L}} & \dots & h_1 \end{bmatrix} \tag{3.24}$$

is the $\mathtt{W} \times (\mathtt{W} + \mathtt{L} - 1)$ channel matrix. Hence, instead of computing the marginal posterior conditioned to the whole sequence of observations, as the block implementation does, it obtains an estimated APP of the transmitted symbol conditioned to just the observations within the window, $p(u_k|\mathbf{y}_{k-\mathtt{W}_2:k+\mathtt{W}_1})$. Under these constraints, the extrinsic distribution defined in (3.11) is determined by the following moments

$$z_k^{[\ell]} = \frac{\mathbf{c_k^{[\ell]}}^{\mathrm{H}}(\mathbf{y_k} - \mathbf{H}_{\mathtt{W}}\mathbf{m_k^{[\ell]}} + m_k^{[\ell]}\mathbf{h}_{\mathtt{W}})}{\mathbf{c_k^{[\ell]}}^{\mathrm{H}}\mathbf{h}_{\mathtt{W}}}, \tag{3.25}$$

$$v_k^{2[\ell]} = \frac{E_s(1 - \mathbf{h}_{\mathtt{W}}{}^{\mathrm{H}}\mathbf{c_k^{[\ell]}})}{\mathbf{c_k^{[\ell]}}^{\mathrm{H}}\mathbf{h}_{\mathtt{W}}}, \tag{3.26}$$

where

$$\mathbf{c_k^{[\ell]}} = \left(\mathbf{\Sigma_k^{[\ell]}} + (E_s - \eta_k^{2[\ell]})\mathbf{h}_{\mathtt{W}}\mathbf{h}_{\mathtt{W}}^{\mathrm{H}}\right)^{-1} E_s\mathbf{h}_{\mathtt{W}}, \tag{3.27}$$

$$\mathbf{m_k^{[\ell]}} = [m_{k-\mathtt{L}-\mathtt{W}_2+1}^{[\ell]}, ..., m_{k+\mathtt{W}_1}^{[\ell]}]^\top, \tag{3.28}$$

$$\boldsymbol{\eta}_{\mathbf{k}}^{2[\ell]} = [\eta_{k-\mathtt{L}-\mathtt{W}_2+1}^{2[\ell]}, ..., \eta_{k+\mathtt{W}_1}^{2[\ell]}]^\top, \tag{3.29}$$

$$\mathbf{\Sigma_k^{[\ell]}} = \sigma_w^2\mathbf{I} + \mathbf{H}_{\mathtt{W}}\mathrm{diag}(\boldsymbol{\eta}_{\mathbf{k}}^{2[\ell]})\mathbf{H}_{\mathtt{W}}^{\mathrm{H}}, \tag{3.30}$$

E_s is the mean transmitted symbol energy and $\mathbf{h}_{\mathtt{W}}$ is the $(\mathtt{W}_2+\mathtt{L})$-th column of $\mathbf{H}_{\mathtt{W}}$ defined in (3.24).

This approach is known as filter expectation propagation (FEP) and its implementation is described in Algorithm 3.4, where the true prior $p(u_k)$ is given by (3.4). Note that the only difference between Algorithm 3.4 and Algorithm 3.3 lies in the moments of the extrinsic distribution, $q_E^{[\ell]}(u_k)$, computed in steps 2) and 4) where running Algorithm 3.2. A detailed description of this implementation can be found in Appendix B.

Algorithm 3.4 FEP standalone Equalizer

Given inputs: y_k for $k=1,\ldots,\mathtt{N}+\mathtt{L}-1$ and $p(u_k)$ given by (3.4) for $k=1,\ldots,\mathtt{N}$. Run Algorithm 3.2 with $q_E^{[\ell]}(u_k)$ in (3.11) computed by replacing $z_k^{[\ell]}$ and $v_k^{2[\ell]}$ with the expressions in (3.25) and (3.26), respectively.

Output: Deliver $q_E^{[S+1]}(u_k)$ to the channel decoder for $k=1,\ldots,\mathtt{N}$.

The computational complexity is dominated now by the inverse of the matrix in (3.27), whose size is the length of the window, $\mathtt{W}$. This yields a cubic complexity in the length of window. As explained in [62], a fast recursive solution can be employed by exploiting the structured time dependence of the matrix to invert, reducing the complexity to be quadratic in the length of the window. Since this computation is repeated along the number of EP iterations, the final complexity is $\mathcal{O}((S+1)\mathtt{N}\mathtt{W}^2)$, i.e., proportional to the one of LMMSE in its filter implementation.

3.2.3 Kalman smoothing expectation propagation (KSEP)

The previous filtered implementation of the EP-based equalizer does not use the whole information about the observations when estimating the posterior probability of each transmitted symbol. For this reason, and in absence of feedback from the decoder, the performance of the FEP is usually far from its block implementation. In this situation, a Kalman smoother type implementation that keeps the same performance as the BEP with linear complexity in the frame length is preferred. This smoothing proposal applies the EP algorithm after the forward and backward steps. Hence, information from the whole vector of observations is exploited and involved in the estimation. Initially, a forward procedure is run, computing at every k-th iteration the following distribution

$$q_k^F(\mathbf{u}_{k-\tilde{\mathtt{L}}:k}) \propto p(y_k|\mathbf{u}_{k-\tilde{\mathtt{L}}:k}) q_{k-1}^F(\mathbf{u}_{k-\tilde{\mathtt{L}}:k-1}) \tilde{p}^{[\ell]}(u_k) \tag{3.31}$$

where $\tilde{\mathtt{L}}=\mathtt{L}-1$ and $q_{k-1}^F(\mathbf{u}_{k-\tilde{\mathtt{L}}:k-1})$ is the marginal of the forward distribution at the previous $(k-1)$-th window, $q_{k-1}^F(\mathbf{u}_{k-\tilde{\mathtt{L}}-1:k-1})$, over $u_{k-\tilde{\mathtt{L}}-1}$ and $\tilde{p}^{[\ell]}(u_k) \sim$

$\mathcal{CN}\left(u_k : m_k^{[\ell]}, \eta_k^{2[\ell]}\right)$. This distribution in (3.31) belongs to the family of Gaussians, i.e., it follows

$$q_k^F(\mathbf{u}_{k-\tilde{\mathrm{L}}:k}) \sim \mathcal{CN}\big(\mathbf{u}_{k-\tilde{\mathrm{L}}:k} : \boldsymbol{\mu}_k^F, \mathbf{S}_k^F\big) \tag{3.32}$$

where

$$\mathbf{S}_k^F = \left(\sigma_w^{-2}\mathbf{h}_{\mathrm{L}:1}\mathbf{h}_{\mathrm{L}:1}^{\mathrm{H}} + \begin{bmatrix} \left(\mathbf{S}_{k-1}^{\backslash 1}\right)^{-1} & \mathbf{0}_{\tilde{\mathrm{L}}\times 1} \\ \mathbf{0}_{1\times\tilde{\mathrm{L}}} & 1/\eta_k^{2[\ell]} \end{bmatrix}\right)^{-1}, \tag{3.33}$$

$$\boldsymbol{\mu}_k^F = \mathbf{S}_k^F\left(\sigma_w^{-2}\mathbf{h}_{\mathrm{L}:1}^{*}y_k + \begin{bmatrix} \left(\mathbf{S}_{k-1}^{\backslash 1}\right)^{-1}\boldsymbol{\mu}_{k-1}^{\backslash 1} \\ m_k^{[\ell]}/\eta_k^{2[\ell]} \end{bmatrix}\right), \tag{3.34}$$

and $\boldsymbol{\mu}_{k-1}^{\backslash 1}$ is the vector $\boldsymbol{\mu}_{k-1}^F$ defined in (3.34) without the first element and $\mathbf{S}_{k-1}^{\backslash 1}$ is the submatrix obtained by removing the first row and column from $\mathbf{S}_{k-1}^F$ defined in (3.33). A similar backward procedure is run to obtain

$$q_k^B(\mathbf{u}_{k-\tilde{\mathrm{L}}:k}) \sim \mathcal{CN}\big(\mathbf{u}_{k-\tilde{\mathrm{L}}:k} : \boldsymbol{\mu}_k^B, \mathbf{S}_k^B\big)\,. \tag{3.35}$$

Note that only a Kalman LMMSE forward and backward has been run, but the EP algorithm has not been applied at that point.

Once the forward and backward distribution is obtained (which can be run in parallel), a smoothed posterior distribution is obtained as:

$$q_k^{FB}(\mathbf{u}_{k-\tilde{\mathrm{L}}:k}) = \frac{q_k^F(\mathbf{u}_{k-\tilde{\mathrm{L}}:k})q_k^B(\mathbf{u}_{k-\tilde{\mathrm{L}}:k})}{p(y_k|\mathbf{u}_{k-\tilde{\mathrm{L}}:k})\prod_{i=k-\tilde{\mathrm{L}}}^{k}\tilde{p}^{[\ell]}(u_i)} \sim \mathcal{CN}\big(\mathbf{u}_{k-\tilde{\mathrm{L}}:k} : \boldsymbol{\mu}_k^{FB}, \mathbf{S}_k^{FB}\big) \tag{3.36}$$

where

$$\boldsymbol{\mu}_k^{FB} = \mathbf{S}_k^{FB}\left(\left(\mathbf{S}_k^F\right)^{-1}\boldsymbol{\mu}_k^F + \left(\mathbf{S}_k^B\right)^{-1}\boldsymbol{\mu}_k^B - \sigma_w^{-2}\mathbf{h}_{\mathrm{L}:1}^{*}y_k - \begin{bmatrix} m_{k-\tilde{\mathrm{L}}}^{[\ell]}/\eta_{k-\tilde{\mathrm{L}}}^{2[\ell]} \\ \vdots \\ m_k^{[\ell]}/\eta_k^{2[\ell]} \end{bmatrix}\right), \tag{3.37}$$

$$\mathbf{S}_k^{FB} = \left(\left(\mathbf{S}_k^F\right)^{-1} + \left(\mathbf{S}_k^B\right)^{-1} - \sigma_w^{-2}\mathbf{h}_{\mathrm{L}:1}\mathbf{h}_{\mathrm{L}:1}^{\mathrm{H}} - \begin{bmatrix} 1/\eta_{k-\tilde{\mathrm{L}}}^{2[\ell]} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & 1/\eta_k^{2[\ell]} \end{bmatrix}\right)^{-1}. \tag{3.38}$$

By marginalizing the expression in (3.36), we have an estimation for the following transmitted symbols: $u_{k-\tilde{\mathtt{L}}},...,u_k$. To latter apply the EP, we decided to use the estimation of u_k, i.e.,

$$q_k^{FB}(u_k) = \int q_k^{FB}(\mathbf{u}_{k-\tilde{\mathtt{L}}:k})\mathrm{d}\mathbf{u}_{k-\tilde{\mathtt{L}}:k-1} \sim \mathcal{CN}\left(u_k : \mu_k, s_k^2\right). \tag{3.39}$$

Any other choice of u_i within the vector $\mathbf{u}_{k-\tilde{\mathtt{L}}:k}$ yields an equivalent approach. An extrinsic distribution can be computed following (3.11) as

$$q_E^{[\ell]}(u_k) = \frac{q_k^{FB}(u_k)}{\tilde{p}^{[\ell]}(u_k)}. \tag{3.40}$$

Up to this point, this procedure is equivalent to a LMMSE equalizer from a Kalman smoothing point of view. To refine this LMMSE estimation, the EP algorithm is applied over the distribution in (3.39), which already has information from all the observations, to obtain an updated value for $m_k^{[\ell]}$ and $\eta_k^{2[\ell]}$. This proposal is repeated along S iterations and it is denoted as Kalman smoothing expectation propagation (KSEP). Its implementation is described in Algorithm 3.5, where the true prior $p(u_k)$ is given by (3.4). Note again that the only difference between Algorithm 3.5 and Algorithm 3.3 or Algorithm 3.4 lies in the way of computing the extrinsic distribution, $q_E^{[\ell]}(u_k)$, computed in steps 2) and 4) of the algorithms when running Algorithm 3.2. A detailed description of this implementation can be found in [47].

Algorithm 3.5 KSEP standalone Equalizer

Given inputs: y_k for $k = 1,\ldots,\mathtt{N}+\mathtt{L}-1$ and $p(u_k)$ given by (3.4) for $k = 1,\ldots,\mathtt{N}$.
Run Algorithm 3.2 with $q_E^{[\ell]}(u_k)$ in (3.11) computed with (3.40).
Output: Deliver $q_E^{[S+1]}(u_k)$ to the channel decoder for $k = 1,\ldots,\mathtt{N}$.

The complexity of the KSEP is driven by the inversion of matrices in (3.38), with size $\mathtt{L}\times\mathtt{L}$. This operation has to be recomputed along S iterations of the EP algorithm and with all the transmitted symbols. Hence the complexity is given by $\mathcal{O}((S+1)\mathtt{N}\mathtt{L}^3)$, i.e., the one of the LMMSE Kalman smoother plus the one of the S iterations of the EP algorithm. This approach is equivalent to the BEP equalizer proposed in Subsection 3.2.1, since both proposals can be seen as an LMMSE whose Gaussian inputs have been refined by means of the EP algorithm, as shown in Figure 3.2. The only difference lies in the way the LMMSE is implemented: the BEP uses a block strategy for the LMMSE while in KSEP a Kalman smoother LMMSE is employed. Since both LMMSE implementations are equivalent, it

yields that both BEP and KSEP are also equivalent, where the KSEP has the advantage of a computational complexity linear in the frame length. As it will be seen in Section 3.3, the performances of both proposals are exactly the same.

At this point it is important to remark that a preliminar smoothing proposal was proposed in Appendix C, named smoothing expectation propagation (SEP), but it is not totally equivalent to the one developed in this subsection. In the SEP proposal, the EP algorithm is used during the forward and backward recursion, only using the past or future information, respectively. Then, both approximations are joined into a smoothing proposal. During the forward recursion, the EP does not take into account the future observations when updating the estimated priors, while during the backward procedure it does not use the past observations. Since not all the observations are used when running the EP algorithm, the minimum allowed variance, 'ε', needed to be set to a relatively high value (specifically, $\varepsilon = 0.5$ as explained in Appendix C) to avoid very confident estimates with errors. Note that in this situation, the error can propagate when moving forward and backward, negatively affecting the estimation at subsequent windows and leading to instabilities. Note also that the SEP procedure is not exactly a Kalman smoothing implementation of the block EP approach, since the EP algorithm is run without taking into account all the observations.

In contrast, in the KSEP approach, rather than applying EP in the backward and forward procedures, it is applied within the smoothing estimation which already has information of all the observations (the future and the past ones). In this approach, the Kalman LMMSE algorithm is run during the forward and backward procedure and finally joined into a smoothing estimation, where the EP is applied. This process is repeated along S iterations and it is equivalent to the BEP equalizer from a Kalman smoothing point of view rather than a block one. On the other hand, the window used by the SEP approach in Appendix C is $\mathtt{W} = 2\mathtt{L} - 1$ while in the current KSEP proposal it is reduced to $\mathtt{W} = \mathtt{L}$. In other words, the SEP uses a higher window than KSEP to counteract errors caused by the absence of the whole vector of observations. Hence, KSEP improves the computational complexity and the convergence of the SEP.

Finally, we also need to point out here the differences between the current KSEP and the proposals in [58, 24]. The KSEP proposes an EP-based equalizer for single equalization, i.e., when no feedback information is available from the channel decoder. This novel equalizer uses the EP algorithm to include the discrete nature of the transmitted symbols to obtain an approximation for the whole posterior distribution at the output of the equalizer. This equalizer is implemented from a Kalman smoothing point of view. On the contrary, the proposals in [58, 24] are just thought for turbo equalization, boiling down to the LMMSE for standalone

equalization. They apply the EP algorithm at the output of the channel decoder to obtain a better Gaussian prior that feeds back the equalizer. The equalizer is implemented as a GMP approach following a Kalman smoothing strategy. Also, they are tested just with BPSK transmissions, not including any mechanism to control instabilities for multidimensional constellations.

3.3 Experimental results

In this section we show the performance of the previous implementations of the proposed EP-based equalizers. The modulator uses a Gray mapping[1]. The results are averaged over 100 random channels and 10^4 random encoded words of length $\mathtt{V} = 4096$ per channel realization. Each channel tap is Gaussian independent and identically distributed (i.i.d.) with zero mean and variance equal to $1/\mathtt{L}$. The absolute value of LLRs given to the decoder is limited to 5 in order to avoid very confident probabilities. A (3,6)-regular low-density parity-check (LDPC) of rate $1/2$ is used, for a maximum of 100 iterations. We set the EP parameters to values in Table 3.2, following the guidelines in Appendix A and Appendix B.

The simulated algorithms are the following:

- LMMSE ($\triangledown$) in its block (or batch) implementation. The LMMSE filter [61, 62, 60] has not been included in the simulations since its block implementation exhibits equal or better performance than any filter-based approach based on the LMMSE algorithm.
- BEP ($\square$). This block equalizer was explained in Subsection 3.2.1 and the corresponding paper can be found in Appendix A.
- FEP ($\star$). This filtering EP-based equalizer was explained in Subsection 3.2.2 and its detailed implementation can be found in Appendix B. The size of the window is set to $\mathtt{W} = \mathtt{W}_1 + \mathtt{W}_2 + 1$ where $\mathtt{W}_1 = 2\mathtt{L}$ and $\mathtt{W}_2 = \mathtt{L} + 1$.
- KSEP ($\times$). This Kalman smoothing implementation was proposed in Subsection 3.2.3 and it is a revised and improved version of the SEP approach in Appendix C.

The comparison with approximated BCJR proposals, such as [19, 43, 55, 10, 18], has been omitted since this study can already be found in Appendix A and Appendix C. They all showed poorer performance than our EP-based equalizers. In Table 3.3, we include a detailed complexity comparison with all the simulated EP-based algorithms.

[1] The results shown in this thesis do not match exactly with the simulations in Appendix A and Appendix C because in these papers we used a non-Gray mapping.

Table 3.3 Complexity comparison between EP-based standalone equalizers.

Algorithm	Description	Complexity	Publication
BEP	Block EP-based equalizer	$\mathcal{O}(11\mathtt{N}^2\mathtt{L})$	Appendix A
FEP	Filter EP-based equalizer	$\mathcal{O}(11\mathtt{N}\mathtt{W}^2)$	Appendix B
KSEP	Smoothing EP-based equalizer	$\mathcal{O}(11\mathtt{N}\mathtt{L}^3)$	-

In Figure 4.4, we depict the BER curves considering random channels of $\mathtt{L} = 7$ complex-valued taps and three different modulations: 16-QAM (a), 64-QAM (b) and 128-QAM (c). It can be observed that both BEP and KSEP approaches have the same performance and improve the results of the classical LMMSE. The improvement increases with the complexity (constellation order) of the scenario. Specifically, the EP implementations outperform the LMMSE in about 2 dB for a 16-QAM, 3 dB for a 64-QAM and 4 dB for a 128-QAM.

The filtered version explained in Subsection 3.2.2 exhibits a degraded performance since it just considers the observations within a small window when estimating the transmitted symbols. For this reason, it is preferred to implement it when feedback from the channel decoder is available since this extra information helps to compensate for the lack of knowledge on the observations out of the window. This turbo feedback will be explained in detail in Chapter 4 and the performance of the filtered version will be shown in Section 4.6.

3.4 Conclusions

From the review of the state of the art in Section 2.1, we conclude that no EP-based standalone equalizer has been proposed in the literature. The works in [58, 24, 66] apply the EP to better approximate with Gaussians the discrete information from the channel decoder, not improving the equalizer step by itself and boiling down to the LMMSE for standalone equalization. Furthermore, only a *Kalman smoothing* architecture has been considered, not dealing with different implementations such as a block or filter EP-based equalizer.

In this chapter, we show how the EP algorithm can be exploited in a standalone equalizer, i.e., when no feedback is available from the channel decoder. The proposed novel EP-based equalizers quite improve the LMMSE performance since they include the discrete nature of the transmitted symbols when computing the Gaussian posterior pdf while the LMMSE just approximates the priors with Gaussians. Specifically, we developed the block (BEP), filtered (FEP) and smoothing (KSEP) proposals:

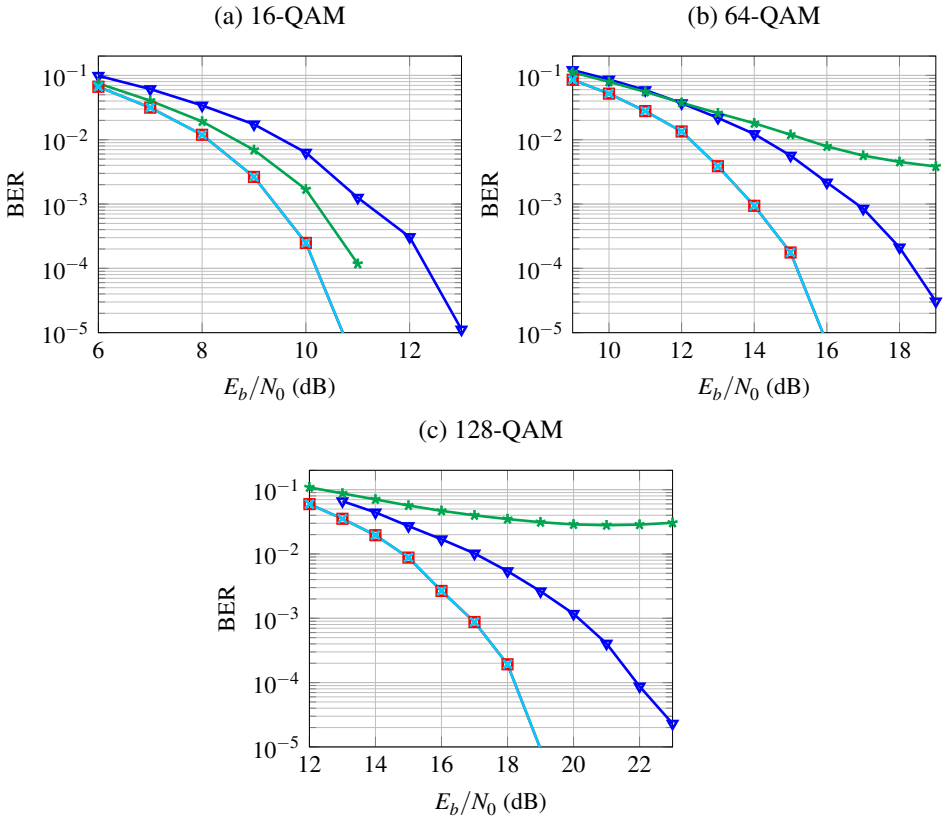


Figure 3.3 BER along E_b/N_0 for block-LMMSE ($\triangledown$), BEP [50] ($\square$), FEP ($\star$) and KSEP ($\times$) standalone equalizers, codewords of $\mathtt{V} = 4096$, 100 random channels with $\mathtt{L} = 7$ and 16-QAM (a), 64-QAM (b) and 128-QAM (c).

- The BEP has already been published in a journal paper (see Appendix A), where simulations of diverse scenarios show its robustness, clearly outperforming the LMMSE and other proposals found in the literature. However, its complexity is quadratic in the frame length, which may be unfeasible for long frames. In this situation, filtered implementations are preferred.

- For this reason, we proposed FEP that only uses the observations within a window to obtain the estimation of the transmitted symbols, reducing the complexity to be quadratic in the length of the window used. This proposal has been also published in a journal paper that can be found in Appendix B. Typically, the window used is set to a small value to keep a low complexity. However, if the window size is not large enough, the algorithm may present a degraded performance (as showed in Section 3.3). This degradation can

be mitigated by increasing the length of the window, at the cost of also increasing the complexity. Another solution is to use a turbo scheme where the extra information from the channel decoder helps to combat the absence of knowledge about the observations not included in the window. This solution will be addressed in Section 4.6.

- To solve the problems of the FEP while keeping a reduction in complexity with respect to BEP, we finally proposed the KSEP. This approach is equivalent to the BEP from a Kalman smoothing point of view and with complexity cubic with the length of the channel. Since it is equivalent to BEP, it showed exactly its same performance in Section 3.3. We recall here that this novel proposal is completely new and has not been published in any journal paper yet. A preliminar, but different, implementation was published in a journal paper (see Appendix C) with the name of SEP. This proposal uses the EP algorithm during the forward and backward recursion, only using the past or future information, respectively. Note the difference with KSEP, where the EP algorithm is applied in the smoothing step, involving the whole vector of observations as it works with the result of joining the forward and backward Kalman LMMSE estimations.

4 Turbo equalization

4.1 Problem to solve

The soft information provided by the probabilistic equalizer can be further iteratively refined by feeding back the output of the channel decoder to the equalizer [14, 28, 61]. This method is called *turbo equalization* and was originally developed for turbo codes [2, 23, 3]. In this scheme, the equalizer and channel decoder iteratively exchange information in terms of log-likelihood ratios (LLRs). Two alternatives are proposed for this exchange of information: extrinsic or posterior feedback [36, 27]. The extrinsic feedback is the most commonly used in the literature, although some previous works show that posterior feedback can outperform the usual choice of extrinsic feedback in some scenarios [27, 54, 65].

This chapter is focused on the improvement of the previous EP-based SISO equalizer in Chapter 3, yielding different time-domain implementations of a soft turbo equalizer that quite outperforms traditional and current approaches in the literature. In all the implementations, the channel response is assumed to be perfectly known.

4.1.1 System model

The model of a digital communication SISO turbo equalization system is represented in Figure 4.1. This system was already introduced in Subsection 3.1.1. The only difference lies in the receiver, which now includes a feedback from the channel decoder to improve the estimation of the posteriors computed by the soft equalizer. Note that, in turbo equalization, the information handled from the equalizer to the channel decoder is usually the extrinsic distribution, rather than the

APP as in Subsection 3.1.1. For this reason, the equalizer must provide an extrinsic distribution, $p_E(u_k|\mathbf{y})$, rather than the APP, $p(u_k|\mathbf{y})$, defined in (3.3).

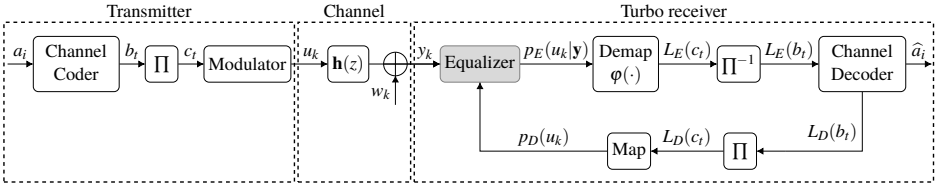


Figure 4.1 System model for turbo equalization.

After demapping the extrinsic distribution at the output of the equalizer and deinterleaving (if needed), the resulting extrinsic LLR is given to the channel decoder. The latter computes an extrinsic LLR on the coded bits as

$$L_D(b_t|L_E(\mathbf{b})) = \log \frac{p(b_t = 0|L_E(\mathbf{b}))}{p(b_t = 1|L_E(\mathbf{b}))} - L_E(b_t), \tag{4.1}$$

which is conditioned to the previous output of the equalizer. This extrinsic LLR is interleaved (if needed) and mapped again to obtain an updated prior for the transmitted symbols, $p_D(u_k|L_E(\mathbf{b}))$, which is fed back to the equalizer. This prior is computed as

$$p_D(u_k|L_E(\mathbf{b})) = \sum_{u \in \mathcal{A}} \delta(u_k - u) \prod_{j=1}^{Q} p_D(c_{k,j} = \varphi_j(u)|L_E(\mathbf{b})), \tag{4.2}$$

where $\varphi_j(u)$ denotes the j-th bit associated to the demapping of symbol u, $c_{k,j}$ is the j-th bit associated to the symbol u_k and $p_D(c_{k,j} = \varphi_j(u)|L_E(\mathbf{b}))$ is the probability that the j-th bit of the symbol u_k was equal to the j-th bit associated to the demapping of symbol u, given $L_E(\mathbf{b})$ as input to the channel decoder. Note that the updated prior in (4.2) clearly differs from the uniform prior in (3.4) used when there is no feedback from the decoder.

This updated prior is used to compute a new posterior distribution at the output of the equalizer as

$$p(\mathbf{u}|\mathbf{y}, L_E(\mathbf{b})) \propto p(\mathbf{y}|\mathbf{u}) \prod_{k=1}^{\mathtt{N}} p_D(u_k|L_E(\mathbf{b})). \tag{4.3}$$

Again, note the prior used in (4.3) is different to the one in (3.3).

This process is repeated iteratively for a given maximum number of iterations, T, or until convergence. To ease the reading, the dependence with $L_E(\mathbf{b})$ will be

omitted from equations in the following. It has been also removed from Figure 4.1.

4.2 EP-based turbo equalizers

This chapter improves the EP-based standalone equalizer proposed in Chapter 3 by introducing feedback from the channel decoder. Note that the information provided by the channel decoder, $p_D(u_k)$, is discrete and that the first step in the EP-based equalizers is to find a Gaussian approximation. This Gaussian approximation, that will be denoted as $\tilde{p}^{[1]}(u_k)$, can be computed by using two different strategies, as explained below.

4.2.1 Single EP turbo equalization

The most straightforward way is to project $p_D(u_k)$ into the family of Gaussians, as the turbo LMMSE does, i.e.,

$$\tilde{p}^{[1]}(u_k) = \text{Proj}_G[p_D(u_k)]. \tag{4.4}$$

The resulting proposed EP-based turbo equalizer can be cast as a double turbo LMMSE equalizer:

- One outer loop ($t = 1,...,T$) for the turbo scheme provides the discrete and non-uniform information from the channel decoder to the equalizer.
- A second inner loop ($\ell = 1,...,S$), due to the iterative behavior of the EP algorithm, corresponding to the standalone EP-based equalizers proposed in Chapter 3. Note that the information provided by the channel decoder follows the discrete and non-uniform distribution described in (4.2), which differs from the uniform prior defined in (3.4) and used in the EP-based equalizers proposed in Chapter 3. For this reason, the true prior used during the moment matching procedure in the EP-based equalizers (Step 1.1 of Algorithm 3.1) must be changed accordingly.

These outer/inner loops are represented in Figure 4.2.

4.2.2 Double EP turbo equalization

A different Gaussian approximation for the discrete information from the channel decoder can be obtained by applying a second EP at that point, as introduced by

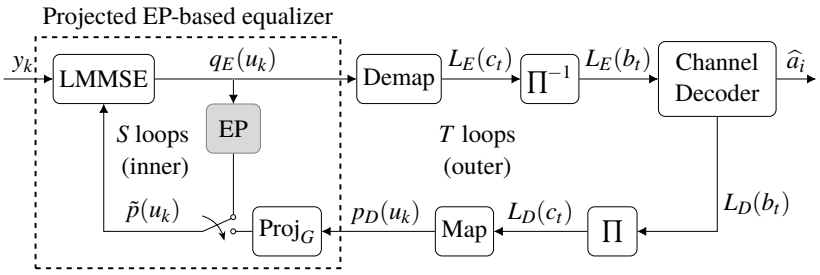


Figure 4.2 Turbo EP-based receiver diagram (option 1).

[58, 24, 66], yielding

$$\tilde{p}^{[1]}(u_k) = \frac{\text{Proj}_G[q_E^{[S+1]}(u_k)p_D(u_k)]}{q_E^{[S+1]}(u_k)} = \frac{\mathcal{CN}\left(u_k : \mu_{\widehat{p}_k}^{[S+1]}, s_{\widehat{p}_k}^{2[S+1]}\right)}{\mathcal{CN}\left(u_k : z_k^{[S+1]}, v_k^{2[S+1]}\right)} \sim \mathcal{CN}\left(u_k : m_k^{[1]}, \eta_k^{2[1]}\right), \tag{4.5}$$

where $q_E^{[S+1]}(u_k)$ is the extrinsic distribution at the output of the equalizer (after S EP iterations) that was given to the channel decoder in the previous turbo iteration and

$$\eta_k^{2[1]} = \frac{s_{\widehat{p}_k}^{2[S+1]} v_k^{2[S+1]}}{v_k^{2[S+1]} - s_{\widehat{p}_k}^{2[S+1]}}, \tag{4.6}$$

$$m_k^{[1]} = \eta_k^{2[1]} \left(\frac{\mu_{\widehat{p}_k}^{[S+1]}}{s_{\widehat{p}_k}^{2[S+1]}} - \frac{z_k^{[S+1]}}{v_k^{2[S+1]}} \right). \tag{4.7}$$

Note the difference with (3.10), where the extrinsic distribution used was the one computed by the EP-based equalizer at the current EP iteration, 'ℓ'. Also, this EP is applied just once, not iteratively along S iterations as in the inner loop. The outer and inner loops are represented[1] in Figure 4.3. The only difference between Figure 4.3 and Figure 4.2 lies in the way they compute the approximation of the discrete information provided by the decoder, i.e., in the block after $p_D(u_k)$. Note also that the computational complexity of both options are the same.

At this point, we need to remark that the variance in (4.6) may return a negative value. In this sense, [24] does not propose any mechanism to deal with a non-valid pdf in case of negative variances, since it states that $\tilde{p}^{[1]}(u_k)$ is then multiplied by

[1] Note that the proposals in [58, 24, 66] just introduced the EP in the context of Kalman smoothing and after the channel decoder, i.e., in the outer loop, while we also introduced a second EP in the inner loop, as it is described in Figure 4.3.

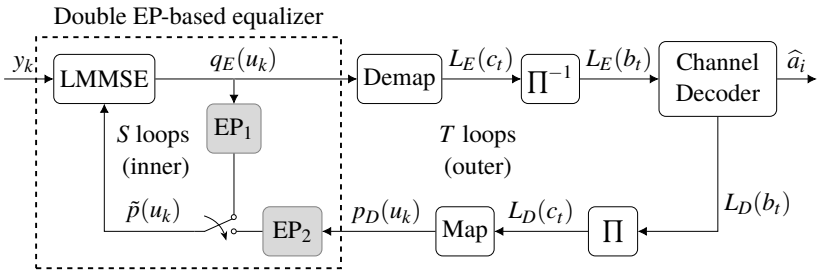


Figure 4.3 Turbo EP-based receiver diagram (option 2).

the extrinsic distribution, which tries to compensate the over confident $\tilde{p}^{[1]}(u_k)$ and this usually results in a valid pdf. The proposals in [58, 66] improve the EP update of [24] by taking the absolute value of the variance in (4.6), not allowing it to become negative. We propose to use a more sophisticated mechanism to control the negative variances. In case (4.6) returns a negative value, we set $\tilde{p}^{[1]}(u_k)$ to (4.4).

4.2.3 Implementation of the proposed generic EP-based equalizer

Once the Gaussian approximation of the discrete information from the channel decoder is obtained, following (4.4) or (4.5), it is handled to the inner loop, i.e., to the standalone EP-based equalizer described in Chapter 3, as the initial value for the EP factor, $\tilde{p}^{[1]}(u_k)$. Iteratively, this equalizer optimizes these factors by matching the moments of the discrete posterior, $\widehat{p}^{[\ell]}(u_k) \propto q_E^{[\ell]}(u_k)p_D(u_k)$, to the ones of its approximation, $q_E^{[\ell]}(u_k)\tilde{p}^{[\ell+1]}(u_k)$, i.e.,

$$\overbrace{q_E^{[\ell]}(u_k)p_D(u_k)}^{\widehat{p}^{[\ell]}(u_k)} \overset{\text{moment matching}}{\longleftrightarrow} q_E^{[\ell]}(u_k)\tilde{p}^{[\ell+1]}(u_k) \tag{4.8}$$

resulting

$$\tilde{p}^{[\ell+1]}(u_k) = \frac{\text{Proj}_G[\widehat{p}^{[\ell]}(u_k)]}{q_E^{[\ell]}(u_k)} = \frac{\mathcal{CN}\left(u_k : \mu_{\widehat{p}_k}^{[\ell]}, s_{\widehat{p}_k}^{2[\ell]}\right)}{\mathcal{CN}\left(u_k : z_k^{[\ell]}, v_k^{2[\ell]}\right)} \sim \mathcal{CN}\left(u_k : m_k^{[\ell+1]}, \eta_k^{2[\ell+1]}\right). \tag{4.9}$$

Note that the difference between (4.9) and (3.10) lies in the true prior used. While in (3.10) the true prior is the uniform distribution defined in (3.4), in (4.9) a non-uniform distribution defined in (4.2) is considered, since it better characterizes the information provided by the decoder. Therefore, we can call Algorithm 3.2 by just introducing the following two changes:

- Replace $p(u_k)$ with $p_D(u_k)$ as input parameter.
- Replace $\tilde{p}^{[1]}(u_k)$ computed in Step 1 with (4.4) or (4.5).

To sum up, when a turbo scheme is employed we first approximate the discrete and non-uniform distribution from the channel decoder with Gaussians, by means of (4.4) or (4.5), and then we run any of the implementations of the EP-based equalizer proposed in Chapter 3 where the true prior used in the moment matching follows (4.2) rather than (3.4). This procedure is described in Algorithm 4.1 for an iteration of the turbo loop. This procedure is then repeated along $t = 1,...,T$ iterations. Note that the only difference with Algorithm 3.2 lies in the computation of $\tilde{p}^{[1]}(u_k)$ in Step 1 and the true prior used in Step 3 as input to Algorithm 3.1. The given input $q_E^{[S+1]}(u_k)$ in Algorithm 4.1 is just needed if $\tilde{p}^{[1]}(u_k)$ in Step 1 is computed with (4.5).

Algorithm 4.1 Generic t-th iteration of the EP-based turbo Equalizer

---

Given inputs: y_k for $k = 1,\ldots,\mathtt{N}+\mathtt{L}-1$, $p_D(u_k)$ given by (4.2) and $q_E^{[S+1]}(u_k)$ defined in (3.11) at the previous $(t-1)$-th turbo iteration for $k = 1,\ldots,\mathtt{N}$.
Run Algorithm 3.2 with:

- $p_D(u_k)$ defined in (4.2) in the place of the input parameter $p(u_k)$.
- $\tilde{p}^{[1]}(u_k)$ computed in Step 1 replaced with (4.4) or (4.5) for single or double EP turbo equalization, respectively. If $\eta_k^{2[1]} < 0$, compute $\tilde{p}^{[1]}(u_k)$ with (4.4).

Output: Deliver $q_E^{[S+1]}(u_k)$ at the current t-th turbo iteration to the channel decoder for $k = 1,\ldots,\mathtt{N}$.

Also, when feedback from the channel decoder is available, the EP parameters can be optimized to reduce the number of EP-iterations in the inner loop. As explained in Appendix B in detail, the computational complexity can be reduced to roughly a third part in comparison with the inner EP-based equalizers in Chapter 3, since only 3 iterations (instead of 10) are needed for the algorithm to achieve the same performance. Specifically, these parameters are set to the values in Table 4.1, where $t = [0,T]$ is the number of the current turbo iteration.

In the rest of the chapter, the three different designs (block, filtered and Kalman smoothing) proposed in Chapter 3 are extended for turbo equalization following the guidelines explained above. The chosen design will determine the specific equations to compute the extrinsic distribution needed for Steps 2 and 4 of Algorithm 3.2 when called from Algorithm 4.1.

Table 4.1 Values of the EP parameters for turbo equalization.

ε	β	S
10^{-8}	$\min(e^{t/1.5}/10, 0.7)$	3

4.3 Turbo block expectation propagation (T-BEP)

This proposal is based on the previous BEP equalizer developed in Subsection 3.2.1. Its detailed implementation is described in Algorithm 4.2. Note that the only difference with Algorithm 3.3 lies in the feedback from the channel decoder, the computation of $\tilde{p}^{[1]}(u_k)$ and the replacement of the true prior $p(u_k)$ with $p_D(u_k)$.

Algorithm 4.2 t-th iteration of the Projected/Double BEP turbo Equalizer

Given inputs: y_k for $k = 1, \ldots, \mathtt{N}+\mathtt{L}-1$, $p_D(u_k)$ given by (4.2) and $q_E^{[S+1]}(u_k)$ at the previous $(t-1)$-th turbo iteration for $k = 1, \ldots, \mathtt{N}$.

Run Algorithm 4.1 with $q_E^{[\ell]}(u_k)$ in (3.11) computed by replacing $\mu_k^{[\ell]}$ and $s_k^{2[\ell]}$ in (3.12) and (3.13) with the expressions in (3.19) and (3.20).

Output: Deliver $q_E^{[S+1]}(u_k)$ at the current t-th turbo iteration to the channel decoder for $k = 1, \ldots, \mathtt{N}$.

In this thesis, we named projected block expectation propagation (P-BEP) the proposal in Algorithm 4.2 if $\tilde{p}^{[1]}(u_k)$ is computed with (4.4). This approach can be found in Appendix B, where it was named non-uniform block expectation propagation (nuBEP) to emphasize the non-uniform nature of the true prior, $p_D(u_k)$, used in Algorithm 4.2. A preliminar result in turbo equalization is described in Appendix A, where rather than using (4.2) as the true priors of Algorithm 4.2, they are set to the uniform ones defined in (3.4) and the information from the channel decoder is just used to initialize $\tilde{p}^{[1]}(u_k)$. Since a non-uniform prior better describes the nature of the information returned by the channel decoder, its performance is clearly outperformed by nuBEP, as shown in Appendix B. On the other hand, we called double block expectation propagation (D-BEP) the proposal in Algorithm 4.2 where $\tilde{p}^{[1]}(u_k)$ is computed with (4.5).

As explained in Subsection 3.2.1, the computational complexity of this implementation is dominated by the inversion of a matrix whose size is given by $\mathtt{N} \times \mathtt{N}$. This inversion can be efficiently computed by exploiting the banded structure of the matrix to invert (see Appendix A). Hence, the final complexity of P-BEP and D-BEP is $\mathcal{O}((S+1)\mathtt{L}\mathtt{N}^2)$ per turbo iteration.

4.4 Turbo filter expectation propagation (T-FEP)

As introduced in Subsection 3.2.2, this filtered Wiener implementation is particularly useful for long frames, where the computational complexity of the block implementation can become infeasible. As illustrated in Section 3.3, the performance of this proposal degrades when no feedback is available from the channel decoder. However, as it will be showed in Section 4.6, the extra information from the channel decoder provided by a turbo scheme helps to counteract the absence of knowledge about the observations not included in the window and the performance is quite close to the block implementation. Its detailed implementation is described in Algorithm 4.3. Note that the only difference with Algorithm 3.4 lies in the feedback from the channel decoder, the computation of $\tilde{p}^{[1]}(u_k)$ and the replacement of the true prior $p(u_k)$ with $p_D(u_k)$.

Algorithm 4.3 t-th iteration of the Projected/Double FEP turbo Equalizer

Given inputs: y_k for $k = 1,\ldots,\mathtt{N}+\mathtt{L}-1$, $p_D(u_k)$ given by (4.2) and $q_E^{[S+1]}(u_k)$ at the previous $(t-1)$-th turbo iteration for $k = 1,\ldots,\mathtt{N}$.
Run Algorithm 4.1 with $q_E^{[\ell]}(u_k)$ in (3.11) computed by replacing $z_k^{[\ell]}$ and $v_k^{2[\ell]}$ with the expressions in (3.25) and (3.26), respectively.
Output: Deliver $q_E^{[S+1]}(u_k)$ at the current t-th turbo iteration to the channel decoder for $k = 1,\ldots,\mathtt{N}$.

In this thesis, we named projected filter expectation propagation (P-FEP) the proposal in Algorithm 4.3 if $\tilde{p}^{[1]}(u_k)$ is computed with (4.4). This approach can be found in Appendix B, where it was named expectation propagation filter (EP-F). On the other hand, we called double filter expectation propagation (D-FEP) the proposal in Algorithm 4.3 where $\tilde{p}^{[1]}(u_k)$ is computed with (4.5).

Regarding complexity, as introduced in Subsection 3.2.2 and Appendix B, the computational complexity is dominated by the inverse of a matrix in (3.27), whose size is $\mathtt{W}\times\mathtt{W}$. By exploiting the structured time dependence of this matrix, a fast recursive solution can be employed as explained in [62]. With this improvement, the final complexity of both P-FEP and D-FEP reduces to $\mathcal{O}((S+1)\mathtt{N}\mathtt{W}^2)$, per outer loop.

4.5 Turbo Kalman smoothing expectation propagation (T-KSEP) approach

The block design in Section 4.3 can be also delivered from a Kalman smoothing point of view, as explained in Subsection 3.2.3. Both proposals (block and smoothing) differ in the formulation but they are equivalent, as it will be shown in Section 4.6. The advantages of the smoothing proposal with respect to block and filtered designs are the following:

- It reduces the computational complexity with respect to the block approach, being linear in the frame length and cubic in the number of taps of the channel.
- It improves the performance of the filtered proposal, at the cost of increasing the complexity in roughly one order of magnitude. Specifically, the filtered approach is quadratic in the length of the window while the Kalman proposal is cubic in the length of channel. In practice, the chosen window for the filtered approach is higher than the length of the channel, and the difference in computational complexity reduces.

Its detailed implementation is described in Algorithm 4.4. Note that the only difference with Algorithm 3.5 lies in the feedback from the channel decoder, the computation of $\tilde{p}^{[1]}(u_k)$ and the replacement of the true prior $p(u_k)$ with $p_D(u_k)$.

Algorithm 4.4 t-th iteration of the Projected/Double KSEP turbo Equalizer

---

Given inputs: y_k for $k = 1, \ldots, \mathtt{N}+\mathtt{L}-1$, $p_D(u_k)$ given by (4.2) and $q_E^{[S+1]}(u_k)$ at the previous $(t-1)$-th turbo iteration for $k = 1, \ldots, \mathtt{N}$.

Run Algorithm 4.1 with $q_E^{[\ell]}(u_k)$ in (3.11) computed with (3.40).

Output: Deliver $q_E^{[S+1]}(u_k)$ at the current t-th turbo iteration to the channel decoder for $k = 1, \ldots, \mathtt{N}$.

In this thesis, we named projected Kalman smoothing expectation propagation (P-KSEP) the proposal in Algorithm 4.4 where $\tilde{p}^{[1]}(u_k)$ is computed with (4.4). This approach can be found in [47], where it was named KSEP. Also, and as introduced in Subsection 3.2.3, a preliminar smoothing proposal is described in Appendix C, named as SEP, where the EP algorithm is applied during the forward and backward procedure rather than over the smoothing distribution as in P-KSEP. A preliminar result in turbo equalization for the SEP is shown in Appendix C where, rather than using (4.2) as the true priors in Algorithm 4.4, they are set to the uniform ones defined in (3.4). Since a non-uniform prior better describes

the nature of the information returned by the channel decoder, its performance is outperformed by P-KSEP. On the other hand, we called double Kalman smoothing expectation propagation (D-KSEP) the proposal in Algorithm 4.4 where $\tilde{p}^{[1]}(u_k)$ is computed with (4.5). Note that if we set $S = 0$ in D-KSEP, it yields the proposal in [58] where they apply the EP algorithm at the output of the channel decoder (outer loop) but no EP is run in the inner loop. Note that the control of negative variances introduced in [58] is different, where the authors just take the absolute value of the variances in case of negative values.

As explained in Subsection 3.2.3, the computational complexity is driven by the inversion of matrices in (3.38), whose size is $\mathtt{L} \times \mathtt{L}$. Then, it requires a cubic complexity in $\mathtt{L}$. This operation has to be recomputed along S iterations of the EP algorithm and with all the transmitted symbols. Hence the final complexity is given by $\mathcal{O}((S+1)\mathtt{N}\mathtt{L}^3)$, per outer loop.

4.6 Experimental results

In this section we show the performance of all proposed schemes for turbo EP-based equalizers and also compare their performance with other EP-based equalizers in the literature. The modulator uses a Gray mapping[2]. The results are averaged over 100 random channels and 10^4 random encoded words of length $\mathtt{V} = 4096$ (per channel realization). A maximum number of $T = 5$ turbo iterations were run. Each channel tap is i.i.d. Gaussian distributed with zero mean and variance equal to $1/\mathtt{L}$. The absolute value of LLRs given to the decoder is limited to 5 in order to avoid very confident probabilities. A (3,6)-regular LDPC of rate $1/2$ is used, for a maximum of 100 iterations. The EP parameters are set to the values in Table 4.1, as explained in Appendix B.

The simulated algorithms are the following:

- LMMSE ($\triangledown$) in its block (or batch) implementation. The LMMSE filter [61, 62, 60] has not been included in the simulations since its block implementation exhibits equal or better performance than any filter-based approach based on the LMMSE algorithm.
- P-BEP ($\circ$). This block equalizer was introduced in Section 4.3 and presented in Appendix B. It is a revised version of BEP in Subsection 3.2.1, optimized for turbo equalization and where the complexity has been reduced to a third part.
- D-BEP ($\bullet$). This block approach was also developed in Section 4.3.

[2] The results shown in this thesis do not match exactly with the simulations in Appendix A and Appendix C because in these papers we used a non-Gray mapping.

- P-FEP ($\star$). This filtering EP-based equalizer was explained in Section 4.4 and published in Appendix B. It is also optimized for turbo equalization. The size of the window is set to $\mathtt{W} = \mathtt{W}_1 + \mathtt{W}_2 + 1$ where $\mathtt{W}_1 = 2\mathtt{L}$ and $\mathtt{W}_2 = \mathtt{L} + 1$.
- D-FEP ($\diamond$). Another filter proposal explained in Section 4.4. The size of the window is set to $\mathtt{W} = \mathtt{W}_1 + \mathtt{W}_2 + 1$ where $\mathtt{W}_1 = 2\mathtt{L}$ and $\mathtt{W}_2 = \mathtt{L} + 1$.
- P-KSEP ($\times$). This Kalman smoothing implementation was proposed in Section 4.5. The length of the window is set to $\mathtt{W} = \mathtt{L}$.
- D-KSEP ($\square$). Another Kalman smoothing proposal explained in Section 4.5. The length of the window is set to $\mathtt{W} = \mathtt{L}$.
- BP-EP ($+$). The EP-based equalizer proposed in [58]. Its performance is shown after 30 turbo iterations, as proposed in [58].
- BCJR ($-$). It is the optimal algorithm in equalization [1]. Due to its exponential complexity, it will be only simulated when possible.

In Table 4.2, we include a detailed complexity comparison (per outer loop) with all the simulated EP-based turbo equalizers.

Table 4.2 Complexity comparison between EP-based turbo equalizers.

Algorithm	Description	Complexity (per outer loop)	Publication
P-BEP	Block Single EP-based turbo equalizer	$\mathcal{O}(4\mathtt{N}^2\mathtt{L})$	Appendix B
D-BEP	Block Double EP-based turbo equalizer	$\mathcal{O}(4\mathtt{N}^2\mathtt{L})$	-
P-FEP	Filter Single EP-based turbo equalizer	$\mathcal{O}(4\mathtt{N}\mathtt{W}^2)$	Appendix B
D-FEP	Filter Double EP-based turbo equalizer	$\mathcal{O}(4\mathtt{N}\mathtt{W}^2)$	-
P-KSEP	Smoothing Single EP-based turbo equalizer	$\mathcal{O}(4\mathtt{N}\mathtt{L}^3)$	-
D-KSEP	Smoothing Double EP-based turbo equalizer	$\mathcal{O}(4\mathtt{N}\mathtt{L}^3)$	-
BP-EP	Smoothing GMP EP-based turbo equalizer	$\mathcal{O}(\mathtt{N}\mathtt{L}^2)$	[58]

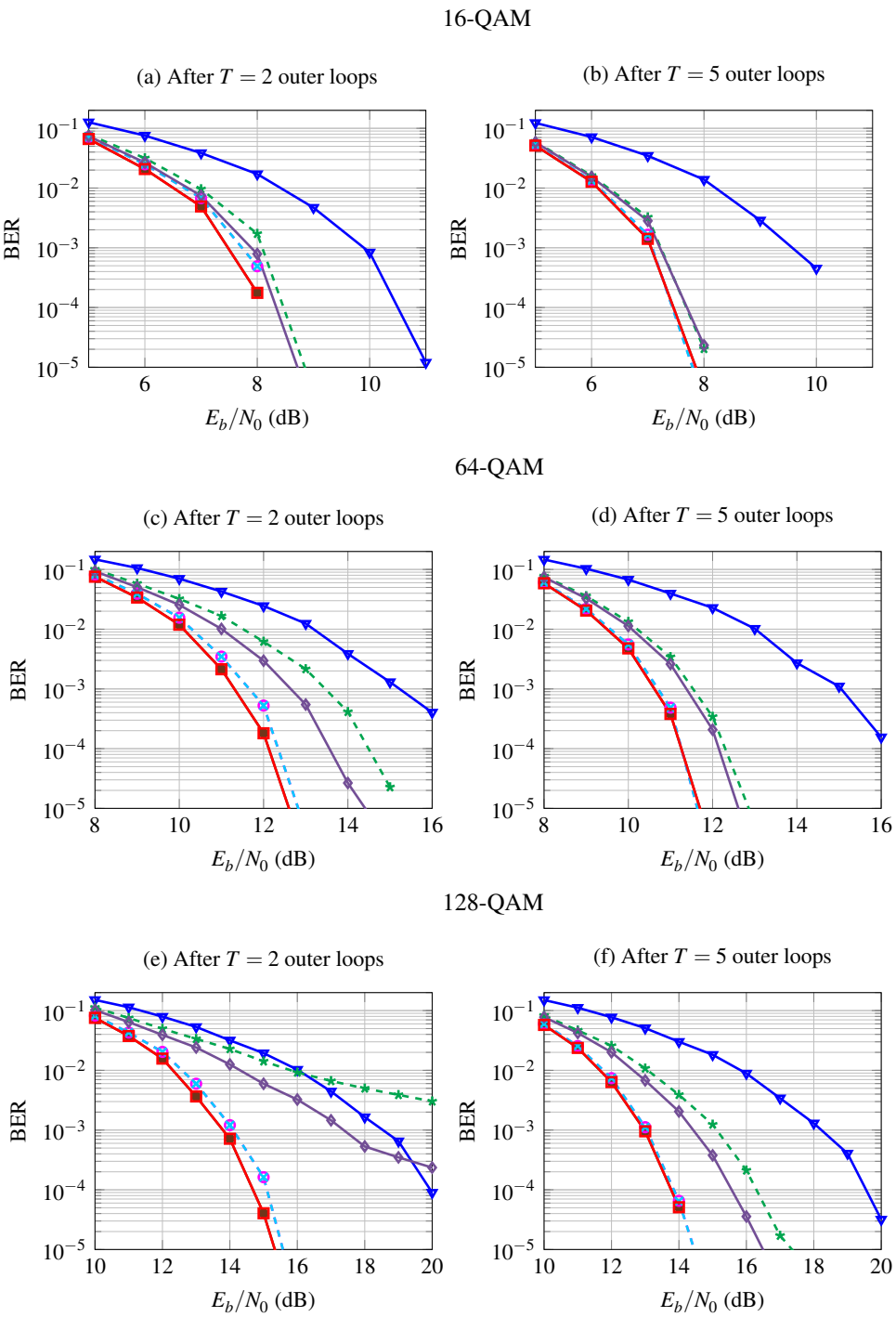


Figure 4.4 BER along E_b/N_0 for turbo block-LMMSE (▽), P-BEP [49] (○), P-FEP [49] (⋆), P-KSEP (×), D-BEP (●), D-FEP (◇) and D-KSEP (□) equalizers, codewords of $\mathtt{V} = 4096$, 100 random channels with $\mathtt{L} = 7$, $S = 3$ and 16-QAM (a)-(b), 64-QAM (c)-(d) and 128-QAM (e)-(f).

4.6.1 Comparison of all the proposed schemes for turbo EP-based equalizers

In Figure 4.4, we depict the BER curves considering random channels of $\mathtt{L} = 7$ complex-valued taps and three different modulations: 16-QAM (a)-(b), 64-QAM (c)-(d) and 128-QAM (e)-(f). We observe that all the EP designs exhibit significant improvements with respect to the LMMSE solution for all constellations (except P-FEP for 128-QAM and $T = 2$). As already introduced, the block and Kalman smoothing proposals are equivalent but developed from a different point of view. That is the reason why they share exactly the same performance. Specifically, P-BEP matches the results of P-KSEP and D-BEP matches the ones of D-KSEP. Also, the batch approaches, P-BEP and D-BEP, have very similar performance for $T = 5$. For a lower number of iterations, the performance of D-BEP slightly improves the one of the P-BEP . The same behavior takes place with the Kalman smoothing proposals, P-KSEP and D-KSEP. This means that the performance can be improved by introducing a second EP in the block or smoothing proposals after the channel decoder, specially when few turbo iterations are performed. A different behavior is obtained in the filtered proposals, P-FEP and D-FEP. The performance of the filtered P-FEP starts deteriorating with respect to other EP-based proposals once the scenario increases its complexity, i.e., with the order of the constellation. The cause of this degradation lies in the nature of the filter behavior that only takes into the account the observations within the window, requiring higher lengths of the considered window when the complexity of the scenario increases to keep an accurate performance. The gap between this filter implementation and the other EP-based implementations decreases when increasing the number of turbo iterations, due to the help of the information provided by the decoder. Its double implementation, D-FEP, exhibits an improvement with respect its P-FEP counterpart, since the second EP introduced at the output of the channel decoder helps to combat the absence of knowledge about the whole vector of observations.

The results are remarkable. For a 16-QAM modulation, all the EP-based implementations have 2 dB and 3 dB gains, for $T = 2$ and $T = 5$, respectively, with respect the LMMSE. Also, all EP-based implementations have quite similar performance. In a 64-QAM scenario, P-FEP improves the LMMSE performance in 2 dB for $T = 2$ and 4 dB for $T = 5$. This improvement is higher for D-FEP, that achieves 3 dB gains for $T = 2$ with respect to the LMMSE, while P-FEP only has gains of 2 dB. These gains can be outperformed with the proposals P-BEP, D-BEP, P-KSEP and D-KSEP, improving the D-FEP in 1.5 dB for $T = 2$ and 1 dB for $T = 5$. Finally, for a 128-QAM, the P-FEP only improves the LMMSE once a sufficient number of turbo iterations has been run. This is due to the complexity of the scenario, which would require to use a higher length of the window to get

an accurate enough performance. After 5 turbo iterations, the information from the decoder has helped the filtered implementation to achieve better performance, improving the LMMSE in 3 dBs. These results can be improved with the use of its double implementation, D-FEP. It obtains 4dB gains for $T = 2$ and 0.75 dB for $T = 5$ with respect to the P-FEP. Regarding P-BEP, D-BEP, P-KSEP and D-KSEP they have gains of 5-6 dB compared to the LMMSE and improve the P-FEP and D-FEP performances in 2 dB after 5 turbo iterations.

Note that all double proposals (D-BEP, D-FEP and D-KSEP) have better or equal performance than their projected counterparts (P-BEP, P-FEP and P-KSEP), where the improvements reduce when increasing the number of turbo iterations. Since both proposals have the same complexity, it is preferable to envisage double proposals than projected ones.

4.6.2 Comparison with other EP-based approaches in the literature

It can be proven that the GMP algorithms in [58, 66, 24] can be viewed as a particularization of D-KSEP when $S = 0$, with a different control of negative variances. In this subsection, we compare the performance (in terms of BER) of BP-EP [58] with the designs for the EP-based equalizers developed in this thesis, showing that BP-EP fails when the size of the constellation grows. Since we already showed in Figure 4.4 that double proposals have better or similar results than projected proposals and they share complexity, we just included our double proposals (D-BEP, D-FEP and D-KSEP) in this section. Figure 4.5 shows the BER curves after $T = 5$ iterations of outer loop and considering encoded words of length $\mathtt{V} = 1024$, random channels of $\mathtt{L} = 5$ real-valued taps and two different modulations: BPSK (a) and 4-PAM (b).

In Figure 4.5 (a), it can be observed again that the D-BEP and D-KSEP have the same performance. The D-FEP is quite close to D-BEP and D-KSEP. Note that BP-EP [58] slightly improves the performance of D-BEP, D-FEP and D-KSEP at low-E_b/N_0 but it has some errors around $E_b/N_0 = 8$ dB due to the lack of a good damping mechanism, while the behavior of the rest of EP-based approaches is quite robust. In Figure 4.5 (b), the performance of BP-EP [58] degrades even when compared to LMMSE turbo equalization, while the EP-based equalizers proposed in this thesis exhibit a gain of 1-2 dB. The BP-EP [58] performance degrades when increasing the size of the constellation since it does not properly control negative variances and does not include any damping mechanism. Since BP-EP [58] degrades when increasing the complexity of the scenario, we do not repeat the modulations in Figure 4.4.

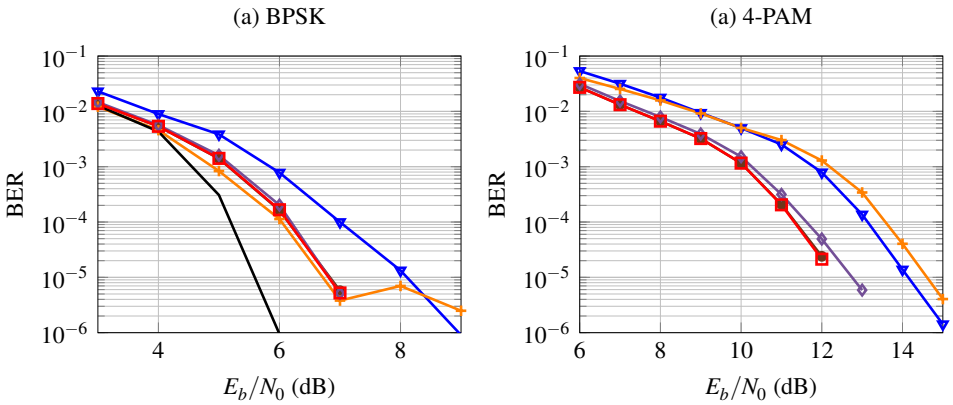


Figure 4.5 BER along E_b/N_0 for turbo block-LMMSE (▽), BP-EP [58] (+), D-BEP (•), D-FEP (◇), D-KSEP (□) and BCJR (−) turbo equalizers, codewords of $\mathtt{V} = 1024$, 100 random channels with $\mathtt{L} = 5$, $S = 3$, $T = 5$ and BPSK (a) and 4-PAM (b) modulations.

4.7 Conclusions

In this chapter, we have applied the EP algorithm to different designs of a turbo equalizer, i.e., when feedback is available from the channel decoder. To the best of our knowledge, the related literature [58, 24, 66] has just used the EP algorithm at the output of the channel decoder to better approximate its output discrete messages with Gaussians, i.e., in the outer loop in Figure 4.3. Then, this Gaussian is used as the prior for an LMMSE that computes a Gaussian posterior (or extrinsic) distribution at the output of the equalizer. Also, they just proposed a Kalman smoothing implementation from a GMP point of view, not dealing with different designs such as a block or a Wiener filter equalizer.

We have improved these proposals in [58, 24, 66]. On the one hand, we used an improved mechanism to control negative variances that reduces the instabilities of the algorithm. On the other hand, we replaced the LMMSE used in [58, 24, 66] by a second EP (yielding an inner loop), that was already introduced by us for standalone equalization (see Chapter 3). It is important to remark that, after the turbo procedure starts, the true priors are given by the information at the output of the channel decoder, that are distributed according to a discrete and non-uniform distribution, rather than an uniform one as stated for single equalization in Chapter 3. Accordingly, we changed the true priors used in the proposed equalizers in Chapter 3 for turbo equalization. As it is shown in Section 4.6, this proposal clearly outperforms the performance of the approach in [58], which is the most

representative solution out of [58, 24, 66]. In addition, we developed the block (Section 4.3), filtered (Section 4.4) and smoothing (Section 4.5) proposals, instead of focusing on only one implementation as in [58, 24, 66].

Finally, for each design (block, filter and smoothing) we developed two proposals regarding how the information at the output of the channel decoder is approximated by a Gaussian. We first proposed to project the information from the decoder directly into a Gaussian distribution, yielding the P-BEP, P-FEP and P-KSEP approaches. In the following, we will refer to them as projected proposals. The other proposed Gaussian approximation is computed by applying EP following the guidelines in [58, 24, 66], yielding the D-BEP, D-FEP and D-KSEP approaches. In the following, we will refer to them as double proposals. These proposals have been introduced for the first time in this thesis and have not been published anywhere yet.

We showed in Section 4.6 that, for the first turbo (outer) iterations, the double proposals slightly improve the projected proposals. Once some turbo iterations are performed, they share the same performance. Since both projected and double proposals share the same computational complexity, it is preferable to run the double approaches than the projected ones. Note also that the performance of the filtered approach (P-FEP and D-FEP) greatly improves with the number of turbo (outer) iterations since, as introduced in Section 3.4, its degradation is due to the absence of knowledge about the whole vector of observations and can be mitigated with the help of extra information provided by the channel decoder. In addition, the block and smoothing EP-based proposals show the same performance since both proposals are equivalent but formulated from a different point of view. All these proposals show a great robustness regardless of the scenario used.

5 Application to turbo MIMO detection

5.1 Problem to solve

Nowadays, there is a great interest in MIMO systems since, among other advantages, they are more spectrally efficient than the traditional SISO systems and provide higher channel capacity [42]. In MIMO, N_t transmit antennas send the symbols over a channel which is corrupted by AWGN and this corrupted signal is received by N_r receive antennas. Then, the received signal is processed in the detector to obtain an estimation of the transmitted symbols. This decision made by the detector can be hard or soft (probabilistic), although the latter is preferred since it results in a high benefit for modern channel decoding [2]. In addition, the performance can be improved with a *turbo detection* scheme, i.e., by iteratively exchanging information between the decoder and the soft detector.

The aim of this chapter is to develop an EP-based turbo detector by adapting previous batch proposals for turbo equalization in Chapter 4 to MIMO systems. These novel detectors are then compared with other EP-based receivers found in the literature, clearly outperforming their results.

5.1.1 System model

The model of a digital communication MIMO system with feedback from the decoder is represented in Figure 5.1. Three different parts can be distinguished: transmitter, channel model and turbo receiver.

- Transmitter: The information bit sequence, $\mathbf{a} = [a_1,...,a_{\mathtt{K}}]^\top$ where $a_i \in \{0,1\}$, is encoded into the codeword $\mathbf{b} = [b_1,...,b_{\mathtt{V}}]^\top$ with a code rate $\mathtt{R} = \mathtt{K}/\mathtt{V}$. The

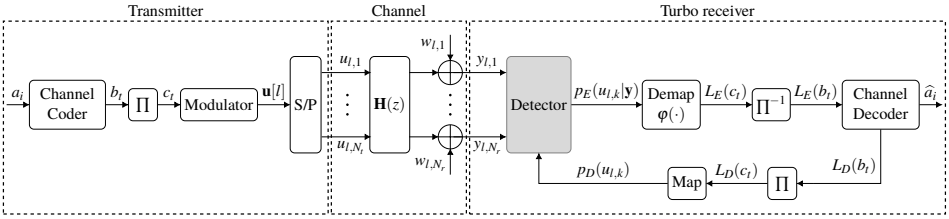


Figure 5.1 System model for a turbo MIMO architecture.

coded bits are interleaved, if needed, and partitioned into $\mathtt{N} = \lceil \mathtt{V}/\log_2 \mathtt{M} \rceil$ blocks, $\mathbf{c} = [\mathbf{c}_1, ..., \mathbf{c}_\mathtt{N}]^\top$, where $\mathbf{c}_n = [c_{n,1}, ..., c_{n,Q}]$ and $Q = \log_2(\mathtt{M})$. This codeword is sent to the M-ary modulator, that obtains the block frames $\mathbf{u} = [u_1, ..., u_\mathtt{N}]^\top$, where each component $u_n = \mathcal{R}(u_n) + j\mathcal{I}(u_n) \in \mathcal{A}$. As in previous chapters, $\mathcal{A}$ denotes the set of symbols of the constellation of order $|\mathcal{A}| = \mathtt{M}$. These symbols are partitioned into $\mathtt{P} = \lceil \mathtt{N}/N_t \rceil$ blocks of length N_t, i.e., $\mathbf{u} = [\mathbf{u}[1], ..., \mathbf{u}[\mathtt{P}]]^\top$, where $\mathbf{u}[l] = [u_{l,1}, ..., u_{l,N_t}]$. Each block is demultiplexed into N_t substreams through the serial to paralell (S/P) converter. Then, the block frames are transmitted over the channel.

- Channel model: The channel is completely specified by the noise variance, σ_w^2, and the weights between each transmitting and receiving antenna, $h_{k,j}$ where $k = 1, ..., N_r$ and $j = 1, ..., N_t$, with $N_r \geq N_t$. The received signal $\mathbf{y} = [\mathbf{y}[1], ..., \mathbf{y}[\mathtt{P}]]^\top$, where $\mathbf{y}[l] = [y_{l,1}, ..., y_{l,N_r}]$, is given by

$$\underbrace{\begin{bmatrix} y_{l,1} \\ \vdots \\ y_{l,N_r} \end{bmatrix}}_{\mathbf{y}[l]} = \underbrace{\begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,N_t} \\ h_{2,1} & h_{2,2} & \ddots & h_{2,N_t} \\ \vdots & \ddots & \ddots & \vdots \\ h_{N_r,1} & \cdots & \cdots & h_{N_r,N_t} \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} u_{l,1} \\ \vdots \\ u_{l,N_t} \end{bmatrix}}_{\mathbf{u}[l]} + \underbrace{\begin{bmatrix} w_{l,1} \\ \vdots \\ w_{l,N_r} \end{bmatrix}}_{\mathbf{w}[l]}, \tag{5.1}$$

  where $\mathbf{w}[l] \sim \mathcal{CN}\big(\mathbf{w}[l] : \mathbf{0}, \sigma_w^2 \mathbf{I}\big)$ is a complex-valued AWGN vector.

- Turbo receiver: Similarly to (4.3), the posterior probability of the transmitted symbol vector $\mathbf{u}[l]$ given the whole vector of observations $\mathbf{y}[l]$ yields

$$p(\mathbf{u}[l]|\mathbf{y}[l]) \propto \mathcal{CN}\big(\mathbf{y}[l] : \mathbf{H}\mathbf{u}[l], \sigma_w^2 \mathbf{I}\big) \prod_{k=1}^{N_t} p_D(u_{l,k}|L_E(\mathbf{b})), \tag{5.2}$$

  where $\mathbf{H}$ is the full matrix with no structure defined in (5.1) and the true prior returned by the decoder, $p_D(u_{l,k}|L_E(\mathbf{b}))$, is clearly non-Gaussian but a

non-uniform discrete distribution that follows (4.2), i.e.,

$$p_D(u_{l,k}|L_E(\mathbf{b})) = \sum_{u\in\mathcal{A}} \delta(u_{l,k}-u) \prod_{j=1}^{Q} p_D(c_{l,k,j} = \varphi_j(u)|L_E(\mathbf{b})), \qquad (5.3)$$

where $\varphi_j(u)$ denotes the j-th bit associated to the demapping of symbol u, $c_{l,k,j}$ is the j-th bit associated to the symbol $u_{l,k}$ and $p_D(c_{l,k,j} = \varphi_j(u)|L_E(\mathbf{b}))$ is the probability that the j-th bit of the symbol $u_{l,k}$ was equal to the j-th bit associated to the demapping of symbol u, given $L_E(\mathbf{b})$ as input to the channel decoder.

In the following, the index l and the dependence on $L_E(\mathbf{b})$ will be omitted to simplify the notation and ease the reading.

In a turbo architecture, the exchange of information between the receiver and the channel decoder is usually done in terms of extrinsic distributions. For this reason, the detector must provide an extrinsic distribution, $p_E(u_k|\mathbf{y})$. After demapping the extrinsic distribution at the output of the detector and deinterleaving (if needed), the resulting extrinsic LLR is given to the channel decoder. The latter computes an extrinsic LLR on the coded bits, $L_D(b_t)$, as in (4.1). This extrinsic LLR is interleaved if needed, mapped again and given to the detector as updated priors, $p_D(u_k)$, that follows the expression in (4.2). This process is repeated iteratively for a given maximum number of iterations, T, or until convergence.

5.2 EP-based turbo detectors

From the results in Chapter 3 and Chapter 4, the block or batch approach based on the EP algorithm can be applied to any system of the form

$$\mathbf{y} = \mathbf{H}\mathbf{u} + \mathbf{w} \qquad (5.4)$$

where $\mathbf{y}$ is a vector of observations, $\mathbf{H}$ is a known matrix, $\mathbf{u}$ is a vector whose elements are discrete and unknown variables and $\mathbf{w} \sim \mathcal{CN}\big(\mathbf{w} : \mathbf{0}, \sigma_w^2\mathbf{I}\big)$.

Detection of MIMO systems follows the model in (5.4) (see (5.1)), where $\mathbf{H}$ is a full $N_r \times N_t$ matrix that represents a memoryless MIMO channel with N_t transmitters and N_r receivers, $\mathbf{u}$ is a vector with the N_t transmitted (and unknown) symbols and $\mathbf{y}$ is a vector with the observed N_r elements. This model is equivalent to the one in (3.2) for equalization in SISO systems. Since both SISO equalizers and MIMO detectors follow the system structure described in (5.4), the batch

implementations proposed in Chapter 3 and Chapter 4 can be also applied to a MIMO system if the channel matrix, transmitted symbols and received signal are replaced accordingly. Hence, the aim of this section is to adapt the block P-BEP and D-BEP equalizers proposed in Section 4.3 to MIMO systems.

5.2.1 Turbo block expectation propagation (T-BEP) for MIMO

This proposal is based on the previous P-BEP and D-BEP equalizers developed in Section 4.3. Since they are block proposals, the moments of the approximated posterior, $q^{[\ell]}(u_k)$, computed by the EP algorithm at the output of the decoder are given by (3.19) and (3.20), respectively, i.e.,

$$\mu_k^{[\ell]} = m_k^{[\ell]} + \eta_k^{2[\ell]} \mathbf{h}_{\mathbf{k}}^{\mathrm{H}} \left(\sigma_w^2 \mathbf{I} + \mathbf{H}\mathrm{diag}(\boldsymbol{\eta}^{2[\ell]})\mathbf{H}^{\mathrm{H}}\right)^{-1} (\mathbf{y} - \mathbf{H}\mathbf{m}^{[\ell]}), \tag{5.5}$$

$$s_k^{2[\ell]} = \eta_k^{2[\ell]} - \eta_k^{4[\ell]} \mathbf{h}_{\mathbf{k}}^{\mathrm{H}} \left(\sigma_w^2 \mathbf{I} + \mathbf{H}\mathrm{diag}(\boldsymbol{\eta}^{2[\ell]})\mathbf{H}^{\mathrm{H}}\right)^{-1} \mathbf{h}_{\mathbf{k}}, \tag{5.6}$$

where `N` must be replaced by N_t, $\mathbf{m}^{[\ell]} = [m_1^{[\ell]}, \ldots, m_{N_t}^{[\ell]}]^\top$, $\boldsymbol{\eta}^{2[\ell]} = [\eta_1^{2[\ell]}, \ldots, \eta_{N_t}^{2[\ell]}]^\top$ and $\mathbf{h}_{\mathbf{k}}$ is the k-th column of the matrix $\mathbf{H}$, previously defined in (5.1). The detailed implementation of this turbo EP-based detector is described in Algorithm 5.1. Note that the differences with Algorithm 4.2 lie in the channel matrix, $\mathbf{H}$, used when computing the extrinsic distribution, $q_E^{[\ell]}(u_k)$, in step 2 and 4 and the replacement of `N` by N_t.

In this thesis, we named P-BEP detector the proposal in Algorithm 5.1 where $\tilde{p}^{[1]}(u_k)$ in step 1 is computed with (4.4). This approach can be found in Appendix D, where it was named nuBEP to emphasize the non-uniform nature of the true prior used during Step 3 in Algorithm 5.1. On the other hand, we called D-BEP detector the proposal in Algorithm 5.1 where $\tilde{p}^{[1]}(u_k)$ in Step 1 is computed with (4.5). The EP parameters of both proposals in Algorithm 5.1 are set to the ones in Table 5.1, where $t = [0,T]$ is the number of the current turbo iteration, as explained in Appendix D in detail. The values are the same as in Table 4.1.

Table 5.1 Values of the EP parameters for turbo MIMO detection.

ε	β	S
10^{-8}	$\min(e^{t/1.5}/10, 0.7)$	3

The computational complexity of the algorithm is dominated by the inversion of the matrices in (5.5) and (5.6), whose size is $N_r \times N_r$. As explained in Subsection 3.2.1, these equations can be also computed with expressions of the form of

Algorithm 5.1 t-th iteration of the Projected/Double BEP MIMO turbo Detector

Given inputs: y_k for $k = 1,\ldots,N_r$, $p_D(u_k)$ given by (5.3) and $q_E^{[S+1]}(u_k)$ at the previous $(t-1)$-th turbo iteration for $k = 1,\ldots,N_t$.

1) Compute $\tilde{p}^{[1]}(u_k)$ with (4.4) or (4.5) and obtain its mean $m_k^{[1]}$ and variance $\eta_k^{2[1]}$. If $\eta_k^{2[1]} < 0$, compute $\tilde{p}^{[1]}(u_k)$ with (4.4).

for $\ell = 1,...,S$ **do**

  for $k = 1,...,N_t$ **do**

    2) Compute the k-th extrinsic distribution, $q_E^{[\ell]}(u_k)$, in (3.11) with $\mu_k^{[\ell]}$ and $s_k^{2[\ell]}$ given by (5.5) and (5.6), respectively.

    3) Run the moment matching procedure in Algorithm 3.1 with $p(u_k)$ replaced by the input parameter $p_D(u_k)$ to obtain $\eta_k^{2[\ell+1]}$ and $m_k^{[\ell+1]}$.

  end for

end for

4) With the moments $\eta_k^{2[S+1]}$ and $m_k^{[S+1]}$ obtained after the EP algorithm, calculate the extrinsic distribution $q_E^{[S+1]}(u_k)$ as in (3.11) with $\mu_k^{[S+1]}$ and $s_k^{2[S+1]}$ given by (5.5) and (5.6), respectively.

Output: Deliver $q_E^{[S+1]}(u_k)$ at the current t-th turbo iteration to the channel decoder for $k = 1,\ldots,N_t$.

(3.21) and (3.22), where the matrix to invert has size $N_t \times N_t$. Since we assumed that $N_r \geq N_t$ (see Subsection 5.1.1), it is preferable to invert the matrix in (3.21) and (3.22) with size $N_t \times N_t$ in terms of complexity. Note that now $\mathbf{H}$ is a full matrix with no structure, as defined in (5.1), then the computational complexity of this inversion is cubic in N_t. This computation has to be repeated along S iterations, then the final complexity of P-BEP and D-BEP is given by $\mathcal{O}((S+1){N_t}^3)$, per turbo iteration.

5.2.2 Related approaches

As discussed in Section 2.2, the EP algorithm has been already successfully applied to MIMO detectors. In [7, 8, 6] a block and iterative approach, named expectation propagation detector (EPD), is developed for detection without considering any feedback from the decoder, i.e., without turbo. These latter approaches are reviewed in [5] and a first trial to include feedback from the decoder is also proposed. The difference between [5] and the P-BEP detector in Subsection 5.2.1 lies on the definition of the true prior, $p_D(u_k)$, used in the moment matching procedure of the

EP algorithm, i.e.,

$$\tilde{p}^{[\ell+1]}(u_k) = \frac{\text{Proj}_G[q_E^{[\ell]}(u_k)p_D(u_k)]}{q_E^{[\ell]}(u_k)}. \tag{5.7}$$

While in [5] $p_D(u_k)$ is set to a uniform and discrete distribution given by

$$p(u_k) = \frac{1}{\mathtt{M}} \sum_{u \in \mathcal{A}} \delta(u_k - u), \tag{5.8}$$

in the current P-BEP approach, it is set to the non-uniform and discrete distribution returned by the channel decoder defined in (5.3). Note that (5.8) and (5.3) match with the definition of the true priors in (3.4) and (4.2), respectively.

As already explained in Chapter 4, the distribution in (5.8) only characterizes the true prior properly before the turbo procedure, but once the feedback from the decoder is available, a non-uniform distribution as in (5.3) is a more suitable option than the uniform one in (5.8). In addition, the EP parameters used by [5] and P-BEP are different. In [5] they are set to $\varepsilon = 2^{-\max(\ell-3,1)}$, $\beta = 0.95$ and $S = 10$ while P-BEP set them to the values in Table 5.1. Since P-BEP reduces the number of EP iterations from 10 to 3, the computational complexity of the algorithm is also a third part of the one in [5]. The advantages in terms of performance between both approaches will be shown in Section 5.3. Note that EPD can be implemented as P-BEP in Algorithm 5.1 by setting the priors used in Step 3 to (3.4) rather than (4.2).

In [54] the authors also apply the EP in a block approach, considering the feedback from the decoder and setting the true prior to (5.3), i.e., their approach follows the implementation of P-BEP. They named it EP IC-LMMSE. The main difference with P-BEP lies in the number of EP iterations. While P-BEP runs the EP algorithm $S = 3$ times per each turbo iteration, EP IC-LMMSE executes just one ($S = 1$). In addition, no damping procedure or minimum allowed variance is employed in EP IC-LMMSE. As will be shown in Section 5.3, an improvement in the performance is obtained (specially for high-order modulations and after some outer iterations) when the EP procedure is repeated along some iterations, as it is the case for P-BEP. The control of negative variances is also implemented differently. While P-BEP keeps the previous value of the EP update in case of negative variances, as in (3.17), in the EP IC-LMMSE they are set to the moments of $\widehat{p}(u_k)$ defined in (3.8), i.e.,

$$m_k = \mu_{\widehat{p}_k}, \quad \eta_k^2 = s_{\widehat{p}_k}^2 \quad \text{if } \eta_k^2 \text{ is negative} \tag{5.9}$$

where the superscript $[\ell]$ has been removed since EP IC-LMMSE only runs one iteration of the EP algorithm. Hence, the EP IC-LMMSE can be run with our proposed P-BEP by setting $S = 1$, $\beta = 1$, changing the control of negative variances in (3.17) by (5.9) and removing the minimum allowed variance set in Step 1.1 of Algorithm 3.1.

5.3 Experimental results

In this section we show the performance of the proposed P-BEP and D-BEP detectors and compare their performance with other EP-based detectors in the literature. The modulator uses a Gray mapping. The results are averaged over 100 random channels and 10^4 random encoded words of length $\mathtt{V} = 4096$ (per channel realization). A maximum number of $T = 5$ turbo iterations were run. Each channel tap is i.i.d. Gaussian distributed with zero mean and variance equal to $1/\mathtt{L}$. The absolute value of LLRs given to the decoder is limited to 5 in order to avoid very confident probabilities. A (3,6)-regular LDPC of rate $1/2$ is used, for a maximum of 100 iterations. The EP parameters of P-BEP and D-BEP are set to the values in Table 5.1.

The simulated algorithms are the following:

- LMMSE ($\triangledown$) in its block (or batch) implementation.
- EPD ($\square$). The block MIMO detector proposed in [8, 5]. It assumes a true prior for the symbols that is uniformly distributed.
- P-BEP ($\circ$). This block MIMO detector was introduced in Subsection 5.2.1 and presented in Paper IV (Appendix D). It is optimized for turbo detection and reduces the complexity of EPD to a third part.
- D-BEP ($\bullet$). This approach was also developed in Subsection 5.2.1.
- EP IC-LMMSE ($+$). The EP-based detector proposed in [54].

In Table 5.2, we include a detailed complexity comparison with all the simulated EP-based turbo MIMO detectors.

In Figure 5.2, we depict the BER curves for the algorithms above, considering $N_t = N_r = 6$ antennas and two different modulations: 64-QAM (a)-(b) and 128-QAM (c)-(d). It is observed that all EP-based detectors greatly outperform the LMMSE approach. Specifically, EPD [5] has gains of 8 dB and 4 dB for 64-QAM and 128-QAM, respectively. These gains are far from the ones obtained with EP IC-LMMSE [54], P-BEP and D-BEP since the latter approaches better characterize the true prior used during the moment matching procedure with non-uniform

Table 5.2 Complexity comparison between EP-based turbo detectors for MIMO.

Algorithm	Description	Complexity (per outer loop)	Publication
P-BEP	Block Single EP-based turbo decoder	$\mathcal{O}(4{N_t}^3)$	Appendix D
D-BEP	Block Double EP-based turbo decoder	$\mathcal{O}(4{N_t}^3)$	-
EPD	Block Single EP-based turbo decoder	$\mathcal{O}(11{N_t}^3)$	[5]
EP IC-LMMSE	Block Single EP-based turbo decoder	$\mathcal{O}({N_t}^3)$	[54]

distributions. Also, EPD shows problems of instabilities at large E_s/N_0 as can be seen from Figure 5.2 (b) and (d). The EP IC-LMMSE is in between the EPD and the proposals in this thesis (P-BEP and D-BEP). This is due to EP IC-LMMSE just computing one iteration of the EP algorithm and not using any damping procedure. Regarding the proposals in this thesis, the D-BEP outperforms P-BEP at the first turbo iterations since the information provided by the decoder can be improved with a second EP at that point. After some turbo iterations, the prior information returned by the decoder is so accurate that a second EP application at that point does not further improve this information. This is the reason why after 5 turbo loops, i.e., Figure 5.2 (b) and (d), P-BEP and D-BEP share the same performance. Specifically, they have gains around 14-15 dB with respect to the LMMSE and 8-10 dB with respect to EPD.

In Figure 5.3 the same scenario is simulated, increasing the number of antennas to $N_t = N_r = 32$. The problem of convergence of EPD is found again in Figure 5.3 (c)-(d) and is more pronounced than in Figure 5.2. In contrast, the proposed P-BEP and D-BEP show a robust behavior in all the simulated scenarios. As in Figure 5.2, the EP IC-LMMSE is in between EPD and the proposals in this thesis and D-BEP is slightly better than P-BEP when few turbo iterations are used. For $T = 5$, P-BEP and D-BEP have gains of 5 dB and 7 dB with respect to the LMMSE for 64-QAM and 128-QAM, respectively, and gains of 4 dB and 5 dB with respect to the EPD.

5.4 Conclusions

In this chapter, we have applied the EP algorithm to a block design of a turbo MIMO detector. We have developed two different versions of this EP-based turbo

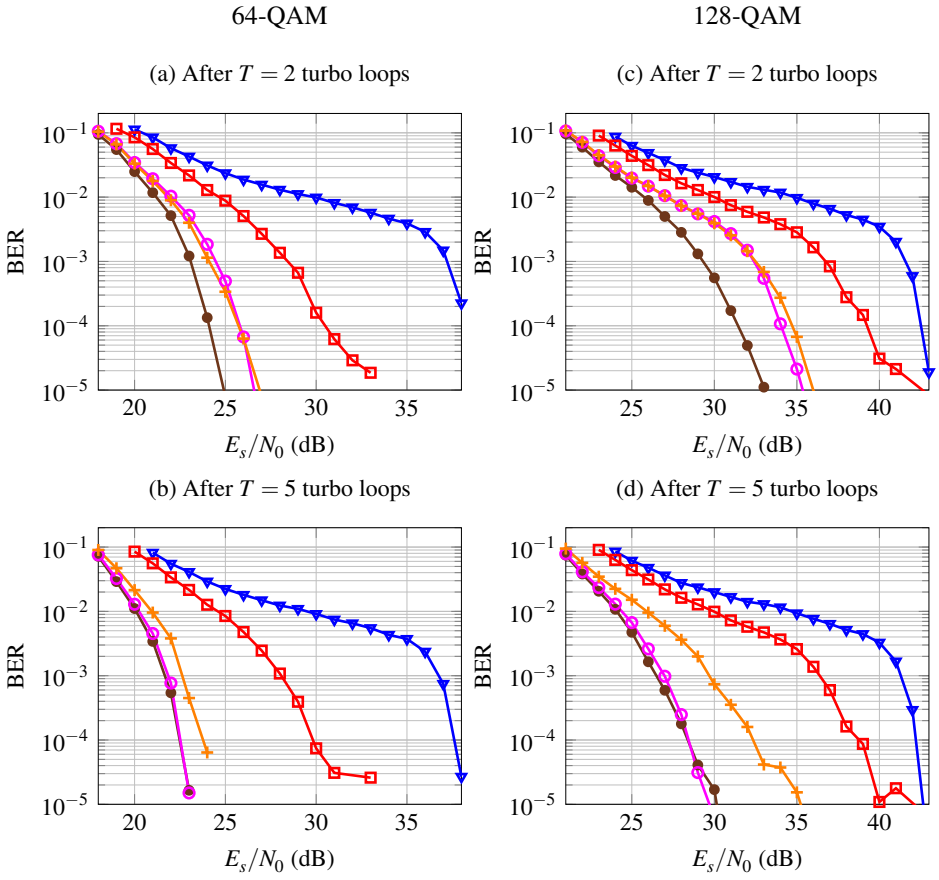


Figure 5.2 BER along E_s/N_0 for block-LMMSE (▽), EPD [5] (□), P-BEP (○), D-BEP (●) and EP IC-LMMSE [54] (+) detectors, codewords of $\mathtt{V} = 4096$, 6×6 systems, 100 random channels and 64-QAM (a)-(b) and 128-QAM (c)-(d) modulations.

detector, based on how the information at the output of the channel decoder is approximated with a Gaussian:

- The first one projects the information from the decoder directly into a Gaussian distribution. We called it P-BEP.
- The second one obtains the approximated Gaussian at the output of the channel decoder by applying EP, yielding a double EP application. We named it D-BEP. This proposal has been introduced for the first time in this thesis and has not been published anywhere yet.

We have reused our previous proposals developed in Chapter 4 for turbo equalization, accordingly replacing the channel matrix, transmitted and received vector.

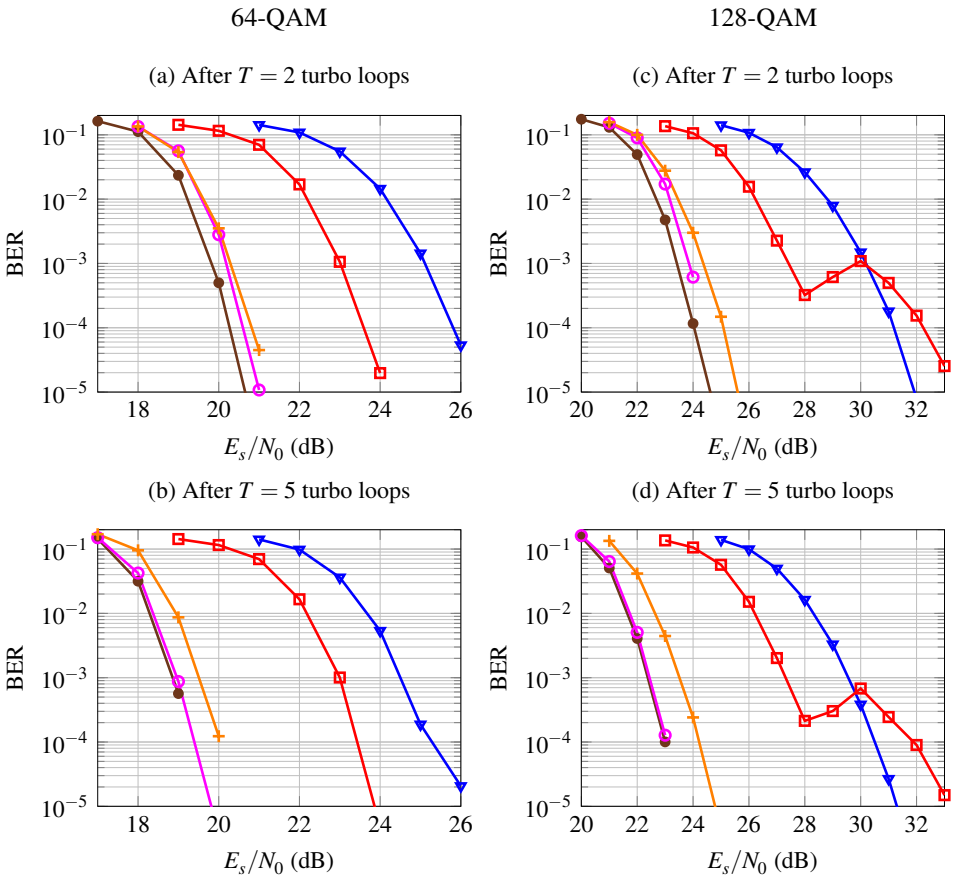


Figure 5.3 BER along E_s/N_0 for block-LMMSE (▽), EPD [5] (□), P-BEP (○), D-BEP (●) and EP IC-LMMSE [54] (+) detectors, codewords of $\mathtt{V} = 4096$, 32×32 systems, 100 random channels and 64-QAM (a)-(b) and 128-QAM (c)-(d) modulations.

To the best of our knowledge, the related literature [5, 54] have just approximated the information from the channel decoder by projecting this information into a Gaussian distribution, i.e., following a a similar structure that our P-BEP proposal. Hence, our D-BEP MIMO detector is completely novel and has not been published yet. In addition, [5, 54] present the following differences with our P-BEP proposal. On the one hand, [5] uses a uniform distribution as the true prior, rather than a non-uniform as P-BEP or [54]. Since the true priors are determined by the information at the output of the channel decoder once the turbo procedure starts and these priors are better characterized by a non-uniform distribution, P-BEP, D-BEP and [54] greatly outperform the proposal in [5]. On the other hand, [54]

does not apply EP iteratively in an inner loop, running the EP algorithm just once, no damping procedure is used, the control of negative variances is also different and no minimum allowed variance value is imposed. As it is shown in Section 5.3, our approaches also improve the one in [54].

6 Conclusions

In this thesis, we focus on the design of probabilistic receivers for digital communications systems. For high-complexity scenarios, optimal inference presents intractable computational complexity and it is necessary to resort to approximate inference, such as the LMMSE. We exploit an alternative approximate technique based on Bayesian inference named EP. Two different scenarios are studied: a SISO system with memory channels and a MIMO system with memoryless channel. In both scenarios, the EP is applied to build a Gaussian approximated posterior distribution at the output of the equalizer/detector. This approximate posterior is found by matching its moments with the ones of the true posterior, to better exploit the discrete nature of the symbols. When feedback from the channel decoder is available, a second EP can be also applied at the output of the channel decoder to better approximate its discrete information with Gaussians.

Through the thesis, three different designs based on the EP algorithm are developed:

- The block or batch approach (see Subsection 3.2.1 and Section 4.3 for equalization and Subsection 5.2.1 for MIMO detection).
- The filter-type version that emulates the Wiener filter behavior (see Subsection 3.2.2 and Section 4.4 for equalization).
- The smoothing equalizer that proceeds as a Kalman forward and backward filter and then joins both estimations into an smoothing distribution (see Subsection 3.2.3 and Section 4.5 for equalization).

All these designs have complexities of the same order than their LMMSE counterparts, which do not depend at all on the modulation size.

These designs were developed as part of a collection of published scientific articles listed below:

- Appendix A: Irene Santos, Juan José Murillo-Fuentes, Rafael Boloix-Tortosa, Eva Arias de Reyna and Pablo M. Olmos, "Expectation Propagation as Turbo Equalizer in ISI Channels," IEEE Transactions on Communications, vol. 65, no.1, pp. 360-370, Jan 2017. [50]
- Appendix B: Irene Santos, Juan José Murillo-Fuentes, Eva Arias de Reyna and Pablo M. Olmos, "Turbo EP-based Equalization: a Filter-Type Implementation," IEEE Transactions on Communications, Sep 2017, Accepted. [Online] Available: https://ieeexplore.ieee.org/document/8353388/ [48]
- Appendix C: Irene Santos, Juan José Murillo-Fuentes, Eva Arias-de-Reyna and Pablo M. Olmos, "Probabilistic Equalization With a Smoothing Expectation Propagation Approach," IEEE Transactions on Wireless Communications, vol. 16, no. 5, pp. 2950-2962, May 2017. [49]
- Appendix D: Irene Santos and Juan José Murillo-Fuentes, "EP-based turbo detection for MIMO receivers and large-scale systems," IEEE Transactions on Vehicular Technology, May 2018, Submitted. [Online] Available: https://arxiv.org/abs/1805.05065 [46]

Experimental simulations show that the proposed receivers quite improve the performance of the well-known LMMSE and other EP-based proposals found in the literature [58, 54, 5]. In addition, they present a quite robust behavior for any type of channel, constellation and length of the transmitted word. Finally, but not less important, the EP parameters have been chosen to work in any scenario, in contrast to other approaches, such as [10, 19, 18, 63, 55], where the parameters were tuned according to a specific channel and/or modulation used.

6.1 Summary of results

In this section we review a summary of the results of this thesis for all the considered scenarios: standalone equalization, turbo equalization and turbo MIMO detection.

- Regarding *standalone equalization*, we have developed three different designs. These are the block BEP, the filtering FEP and the Kalman smoothing KSEP. In terms of complexity, the BEP can be unfeasible for long frames since its complexity is quadratic with the length of the frame. In this situation of long frames, the other two proposals are preferable since their complexity is linear in the frame length. The FEP, which is based on a Wiener filter, uses a window whose size determines the observations that will be considered when estimating the transmitted symbols. The size of the window determines the computational complexity of the algorithm. By exploiting the

structured time dependence of the matrix to invert, a quadratic complexity in the size of the window can be achieved. However, this window size does not only affect the complexity, but also the performance. If the window is not large enough, the absence of knowledge about the observations out of the window usually degrades the performance. In contrast, if the window is large enough, the performance of FEP can be close to its block counterpart. This degradation of the performance can be avoided with the KSEP. The complexity of KSEP is also dominated by the length of the window, whose value is set to the length of the channel. Due to the smoothing behavior, no structure can be exploited to reduce the computational complexity of the matrix to invert, hence the complexity is cubic in the length of the window. The KSEP achieves the performance of the BEP with linear complexity in the length of the frame, N. To sum up, BEP and KSEP will always achieve a performance that is better or equal that the one of the FEP. In terms of complexity, KSEP is an order of magnitude lower than BEP in terms of N. This complexity can be further reduced to be quadratic with the length of the windows by using the FEP approach at the cost of deteriorating the performance.

- These three implementations are then adapted to *turbo equalization*, naming them P-BEP, P-FEP and P-KSEP. These proposals introduce the discrete information from the channel decoder by projecting it into a Gaussian distribution directly. We also proposed a different approach where a second EP is applied at the output of the channel decoder to better approximate the information from the channel decoder. These proposals are called D-BEP, D-FEP and D-KSEP. The complexity of the block (P-BEP and D-BEP), the Wiener filtering (P-FEP and D-FEP) and Kalman smoothing (P-KSEP and D-KSEP) proposals is dominated by the same factors than their standalone equalization counterparts. The only difference is the number of EP iterations, which can be reduced due to the additional information provided by the channel decoder. Specifically, the complexity of the inner EP algorithm is reduced to a third part in comparison with their standalone equalization counterparts, although it is then repeated along all the turbo iterations. In terms of performance, the double proposals (D-BEP, D-FEP and D-KSEP) slightly improve the projected proposals (P-BEP, P-FEP and P-KSEP) for the first turbo (outer) iterations. Once some turbo iterations are performed, they share the same performance. Since both projected and double proposals have the same computational complexity, it is preferable to run the double approaches than the projected ones. Focusing on these double proposals and as we already stated for standalone equalization, the block D-BEP and

Kalman smoothing D-KSEP are equivalent in terms of performance. Also, the D-FEP greatly improves its performance in comparison with standalone equalization, since the absence of knowledge about the observations is mitigated with the help of the extra information provided by the channel decoder. Anyway, its performance is always improved by the D-BEP and D-KSEP since they use the whole vector of observations. To sum up, in terms of performance, it is better to run double than projected proposals and, specifically, D-BEP or D-KSEP rather than D-FEP. In terms of complexity D-KSEP and D-FEP are an order of magnitude lower in N than D-BEP. Other EP-based approaches in the literature, such as [58], were proposed for BPSK constellations and their performance degrades when high-order modulations are employed. Hence, our proposals greatly outperform the approach in [58].

- Finally, the block approaches P-BEP and D-BEP are adapted to turbo *MIMO detection* by changing the channel matrix and transmitted/received vectors accordingly. The results are similar to the ones showed in turbo equalization. The double proposal D-BEP shows slightly better performance than P-BEP for the first turbo (outer) iterations. Since both approaches have the same performance, it is preferable to run D-BEP. When comparing with other EP-based turbo detectors found in the literature, such as [54, 5], our approaches are able to improve both of them.

6.2 Future lines

The following improvements for the work presented in this thesis remain for the future:

- Since frequency domain has received a lot of attention [17, 61, 25], frequency domain counterparts for the proposed EP based turbo-receivers could be derived. In [44], results from this thesis have been employed to develop a frequency domain GMP EP-based equalizer.
- The formulation of all the implementations proposed in this thesis for a MIMO system with memory.
- The design of an optimum channel code in terms of capacity for the channel equalization response to optimize the turbo stage.
- Reduction of the computational complexity of the algorithms, which is dominated by the inversion of a matrix. How an approximated inverse with lower complexity, rather than the exact one, may influence the performance of the algorithm could be also investigated.

- The digital receivers proposed in this thesis assume that the CSI is perfectly known and static. An extension of this work could introduce error in the CSI and analyze how this uncertainty affects the behavior of all the proposals.
- A successive interference cancellation (SIC) implementation for the MIMO detector could be also developed. The computational complexity of this type of detectors is an order of magnitude higher than the complexity of the block implementation, i.e., it is quartic in the frame length. Hence, a reduction of complexity should be considered.

Appendix A
Paper I

Authors: Irene Santos, Juan José Murillo-Fuentes, Rafael Boloix-Tortosa, Eva Arias de Reyna and Pablo M. Olmos.
Title: Expectation Propagation as Turbo Equalizer in ISI Channels.
Journal: IEEE Transactions on Communications.
Volume: 65.
Number: 1.
Pages: 360-370.
Date: January 2017.
Abstract: In probabilistic equalization of channels with inter- symbol interference, the BCJR algorithm and its approximations become intractable for high-order modulations, even for moderate channel dispersions. In this paper, we introduce a novel soft equalizer to approximate the symbol a posteriori probabilities (APP), where the expectation propagation (EP) algorithm is used to provide an accurate estimation. This new soft equalizer is presented as a block solution, denoted as block-EP (BEP), where the structure of the matrices involved is exploited to reduce the complexity order to $\mathcal{O}(\mathtt{LN}^2)$, i.e., linear in the length of the channel, $\mathtt{L}$, and quadratic in the frame length, $\mathtt{N}$. The solution is presented in complex-valued formulation within a turbo equalization scheme. This algorithm can be cast as a linear minimum-mean-squared-error (LMMSE) turbo equalization with double feedback architecture, where constellations being discrete is a restriction exploited by the EP that provides a first refinement of the APP. In the experiments included, the BEP exhibits a robust performance, regardless of the channel response, with gains in the range 1.5-5 dB compared with the LMMSE equalization.

Expectation Propagation as Turbo Equalizer in ISI Channels

Irene Santos, Juan José Murillo-Fuentes, Rafael Boloix-Tortosa, Eva Arias-de-Reyna, and Pablo M. Olmos

***Abstract*—In probabilistic equalization of channels with inter-symbol interference, the BCJR algorithm and its approximations become intractable for high order modulations, even for moderate channel dispersions. In this paper we introduce a novel soft equalizer to approximate the symbol a posteriori probabilities (APP) where the expectation propagation (EP) algorithm is used to provide an accurate estimation. This new soft equalizer is presented as a block solution, denoted as block-EP (BEP), where the structure of the matrices involved is exploited to reduce the complexity order to $\mathcal{O}(LN^2)$, i.e., linear in the length of the channel, L, and quadratic in the frame length, N. The solution is presented in complex-valued formulation within a turbo equalization scheme. This algorithm can be cast as a linear minimum-mean-squared-error (LMMSE) turbo equalization with double feedback architecture where constellations being discrete is a restriction exploited by the EP that provides a first refinement of the APP. In the experiments included, the BEP exhibits a robust performance, regardless of the channel response, with gains in the range 1.5-5 dB compared to the LMMSE equalization.**

***Index Terms*—Expectation propagation (EP), BCJR, complex-valued, turbo equalization, ISI.**

I. Introduction

SOFT or probabilistic channel equalization [1] is a technique to mitigate the interference between symbols (ISI) provoked by the dispersive nature of the channel [2], [3]. It provides the posterior probabilities of the estimated transmitted symbols given the observation, from which nowadays decoders highly benefit [4]. These two tasks, equalization and decoding, were initially considered separately, but the performance was remarkably improved by joining them into a turbo equalization scheme [5]–[7]. In turbo equalization, an equalizer and a decoder exchange information in terms of log-likelihood ratios (LLRs). After one or more iterations, the channel decoder generates the LLRs which are delivered back to the equalizer as updated a priori information.

The optimal BCJR algorithm [8] computes the a posteriori probability (APP) for each transmitted symbol providing maximum a posteriori (MAP) probabilistic decisions. It works on a trellis representation, assuming perfect knowledge of the channel impulse response (CIR) and a channel with finite memory [9]. The BCJR complexity is proportional to the number of trellis branches, M^L, increasing with the number of taps of the channel, L, and the size of the constellation used, M. The BCJR memory requirements per step also grow with the number of states. Therefore, for a few taps and a multilevel constellation the complexity becomes intractable.

To reduce the complexity of the BCJR some suboptimal algorithms, based on performing a simplified trellis search with only M_e states, have been proposed in the literature. They can be divided into two different families. The first one consists in reducing the effective length of the CIR, such as the reduced-state BCJR (RS-BCJR) algorithm [10], which is based on the reduced-state sequence detection (RSSD) [11]–[13]. The key idea is to cancel the final channel taps, by truncating the memory of the channel, to reduce the number of states. On the other hand, other algorithms only keep the states with highest APP, i.e., unlike the previous algorithms, they perform a reduced search on the original full trellis, instead of a full search on a reduced-state trellis. This is the case of the M-BCJR algorithm [14]. Some approaches try to join both families to improve the results. We mention the M*-BCJR in [15], that outperforms both RS-BCJR and M-BCJR algorithms. These approaches have some important limitations. A first issue is that they are usually designed for some types of channels [10], [14], [15]. Secondly, they are unable to merge the paths determined by forward and backward trellises, since these paths generally do not match in both procedures. To overcome these problems, a variation of the M-BCJR algorithm is proposed in [16], where the authors use a different active state selection criterion. In [17] the channels are equalized differently according to their minimum, maximum or mixed phase nature [18]. Depending on the channel realization, they resort to the forward recursion or forward-trellis, backward-trellis or an optimized mixture of them. This approach is referred to as nonzero (NZ) completion. Thirdly, all approximated methods above ignore some paths in the trellis, which means that the explored paths tend to be considered much more reliable than they really are. To reduce this overestimation, the NZ with output saturation (NZ-OS) is proposed in [17]. The above issues can be mitigated with channel shortening approaches, such as [19], where the authors perform a full search over an optimized and reduced trellis. However, it is important to remark that the performance of this approach and approximated BCJR solutions degrades if the number of survivor paths, M_e, does not grow accordingly

I. Santos, J.J. Murillo-Fuentes, R. Boloix-Tortosa and E. Arias-de-Reyna are with the Dept. Teoría de la Señal y Comunicaciones, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Camino de los Descubrimiento s/n, 41092 Sevilla, Spain. E-mail: {irenesantos,murillo,rboloix,earias}@us.es.

P. M. Olmos is with the Dept. Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid, Avda. de la Universidad 30, 28911, Leganés (Madrid), Spain. E-mail: olmos@tsc.uc3m.es.

The final version of the manuscript can be found at https://ieeexplore.ieee.org/document/7587428/. This work was partially funded by Spanish government (Ministerio de Economía y Competitividad TEC2016-78434-C3-{2-3}-R and Juan de la Cierva Grant No. IJCI-2014-19150), by the European Union (FEDER and Marie Curie Initial Training Network "Machine Learning for Personalized Medicine" MLPM2012, Grant No. 316861), and by Comunidad de Madrid in Spain (project 'CASI-CAM-CM', id. S2013/ICE-2845).

with the total number of states. Hence they are computationally unfeasible for large trellises and equalizers of the MMSE type are preferred [20].

Message passing approaches have been also investigated, see [21]–[25] and references therein. In [21], an equalizer based on the belief propagation (BP) algorithm is developed to reduce the inference complexity in sparse channels. However, the complexity of the method still grows exponentially with the size of the modulation and the number of nonzero channel interferers. In [24] the graphical model of the system is rewritten to end with a graph with loops equal or larger than 6, and the loopy BP is then applied. Its output is an approximation reported to provide good results in ISI channel equalization for BPSK modulations, although the method can be applied to other modulations. Its complexity scales linearly with the frame length and the channel memory, but quadratically with the constellation order. A different approach is proposed in [23], where a successive interference canceler applied to equalization is developed by considering the interference plus noise as Gaussian distributed. In [22] the authors develop an approach that introduces expectation propagation (EP) approximate inference [26]–[28] to incorporate into the BP algorithm the information from the BPSK symbol estimates coming from the channel decoder. In [25] the authors develop a different EP implementation for the equalizer in [22]. These two works develop particular instances of EP to project the BP messages into the right distribution (Gaussian/discrete), allowing feasible updates of the BP messages. The main difference between the EP method proposed by [22] and [25] is the procedure to tackle numerical instabilities during the EP updates related to variance parameters taking extremely small values or even negative ones. While in [22] the authors introduce a damping approach, in [25] negative variance parameters are replaced by their absolute values. At this point it is important to remark that [22], [23], [25] consider BPSK transmissions and the EP is just used to better approximate messages from the decoder to the equalizer.

Soft linear equalization, such as the linear minimum-mean-squared-error (LMMSE) [29], is a suboptimal but low-cost alternative. Its complexity is dominated by the inversion of a matrix of size N when all observations are processed in a block or batch approach. When no turbo scheme is used the complexity can be reduced to $\mathcal{O}(N \log N)$ by exploiting the circulant nature of the involved matrix and the fast Fourier transform (FFT). However, after the feedback in the turbo equalization the matrix is no longer circulant and we can not use the FFT, yielding a cubic complexity in N [30]. In [31], some approximations are introduced to lower the complexity. Some windowed versions have also been developed in order to reduce this complexity. Specifically, in [32] a sliding window LMMSE algorithm is proposed and in [33] these results are improved by replacing the sliding window with an extending window. In [5], [32], [34] the authors also propose some approximated windowed solutions to further reduce the complexity.

In this work we focus on a novel block solution to improve the equalization algorithms above when dealing with multidimensional constellations, with linear complexity in the constellation size and for any channel realization. We present an EP-based algorithm that approximates the joint posterior probability of the transmitted bits in a centralized manner, i.e., we do not use EP to project the BP messages into a different distribution, as [22], [25] would do. EP has been already successfully applied to multiple-input multiple-output (MIMO) detection [35]–[37], low-density parity-check (LDPC) channel decoding [38], [39], tracking of flat-fading channels [40] and equalization of BPSK transmissions with message passing BP approaches [22], [25]. We exploit the key idea in [36], where compared to [35] the EP is used to better approximate the full posterior rather than to improve message passing algorithms at some points. Preliminary results for equalization of real-valued systems were discussed in [41]. In this paper we propose a turbo equalization scheme where we implement EP to obtain a complex-valued Gaussian approximation to the probability of the transmitted symbols conditioned to the received signal. We denote this approach as turbo BEP equalizer (T-BEP). We discuss the interpretation of this T-BEP as a double turbo LMMSE equalizer, where the discrete nature of the transmitted symbols is used as a first feedback. To deal with negative variance we avoid updates whenever EP provides negative variance. The computational complexity per step of the T-BEP is dominated by the inversion of an N-dimensional banded covariance matrix, which has exactly the same structure as in the turbo LMMSE scheme [30]. We exploit the structure of the matrices to reduce it to $\mathcal{O}(LN^2)$, i.e., quadratic complexity in N and independent of M. In addition, we address the estimation of the mutual information between the detected and transmitted symbols to explain the obtained gain. Finally, we compare it to the most relevant approximated BCJR approaches discussed above. As a result, at low dimensional scenarios we achieve a performance close to the optimal BCJR solution, regardless of the channel realization. For multilevel constellations the BEP outperforms the turbo LMMSE equalizer (T-LMMSE). This performance is further improved with the T-BEP. Gains in the 2-5 dB range are reported for 16 and 64-QAM constellations.

The paper is organized as follows. We first describe in Section II the structure and model of the communication system at hand and review the formulation for the block LMMSE turbo equalizer. In Section III the EP algorithm is introduced. Section IV is devoted to describe the novel proposed T-BEP equalizer and develop its formulation for complex numbers. We also study the convergence of the algorithm to propose values for the parameters of the EP equalizer and exploit the structure of the matrices involved to propose efficient computations. In Section V, we include several experiments to show the good performance of the T-BEP. We end with some conclusions.

The following specific notation is used throughout the paper. If $\mathbf{u}$ is a vector, $\mathbf{u}_{i:j}$ refers to a column vector with the entries of vector $\mathbf{u}$ indexed by the set $\{i, i-1, i-2, ..., j\}$. We use $\mathbf{u}^*$ to denote the complex conjugate of $\mathbf{u}$. To denote a normal distribution of a random proper complex vector $\mathbf{u}$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ we use $\mathcal{CN}(\mathbf{u} : \boldsymbol{\mu}, \boldsymbol{\Sigma})$.

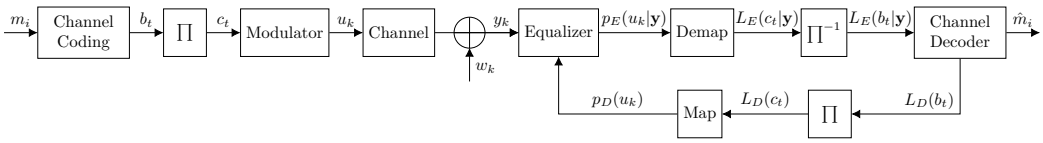


Fig. 1: System model and turbo equalization.

II. System model and LMMSE solution

A. System model

In Fig. 1 we represent the discrete-time communication system with turbo equalization. It can be divided into four parts: transmitter, channel model, equalizer and turbo equalization.

1) Transmitter: A block of K message bits, $\mathbf{m} = [m_1, ..., m_K]^\top$, is encoded with a rate $R = K/V$ into the codeword $\mathbf{b} = [b_1, ..., b_V]^\top$ and permuted with an interleaver to $\mathbf{c} = [c_1, ..., c_V]^\top$. An M-ary modulation is considered to obtain $N = \lfloor V/\log_2 M \rfloor$ symbols, in $\mathbf{u}$. Then, the block frames $\mathbf{u} = [u_1, ..., u_N]^\top = \mathcal{R}(\mathbf{u}) + \mathrm{j}\mathcal{I}(\mathbf{u})$ are transmitted over the channel, where each component $u_k = \mathcal{R}(u_k) + \mathrm{j}\mathcal{I}(u_k) \in \mathcal{A}$. Hereafter, $\mathcal{A}$ denotes the set of symbols of the constellation of order $|\mathcal{A}| = M$. The mean transmitted symbol energy and energy per bit are denoted by E_s and E_b, respectively.

2) Channel model: The channel is completely specified by the CIR, i.e., $\mathbf{h} = [h_1, ..., h_L]^\top$, where L is the length of the CIR, and the noise variance σ_w^2, that we assume known at the receiver. The received signal $\mathbf{y} = [y_1, ..., y_{N+L-1}]^\top \in \mathbb{C}^{N+L-1}$ is given by

$$\begin{bmatrix} y_1 \\ \\ \vdots \\ \\ y_{N+L-1} \end{bmatrix} = \begin{bmatrix} h_1 & & \mathbf{0} \\ \vdots & \ddots & \\ h_L & \ddots & h_1 \\ & \ddots & \vdots \\ \mathbf{0} & & h_L \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} + \begin{bmatrix} w_1 \\ \\ \vdots \\ \\ w_{N+L-1} \end{bmatrix} \tag{1}$$

or

$$\mathbf{y} = \mathbf{H}\mathbf{u} + \mathbf{w} \tag{2}$$

in matrix form, where

$$y_k = \sum_{l=1}^{L} h_l u_{k-l+1} + w_k = \mathbf{h}^\top \mathbf{u}_{k:k-L+1} + w_k, \tag{3}$$

$u_k = 0$ for $k < 1$ and $k > N$, and $\mathbf{w} \sim \mathcal{CN}(\mathbf{w} : \mathbf{0}, \sigma_w^2 \mathbf{I})$. In this case, it is circular complex AWGN due to its zero mean [42].

3) Equalization: Given the model above, the posterior probability of the transmitted symbol vector $\mathbf{u}$ yields

$$p(\mathbf{u}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{u})p(\mathbf{u})}{p(\mathbf{y})} \propto \mathcal{CN}(\mathbf{y} : \mathbf{H}\mathbf{u}, \sigma_w^2 \mathbf{I}) \prod_{k=1}^{N} \mathbb{I}_{u_k \in \mathcal{A}}, \tag{4}$$

where $\mathbb{I}_{u_k \in \mathcal{A}}$ is the indicator function that takes value one if $u_k \in \mathcal{A}$ and zero otherwise. The optimal equalizer to estimate $p(u_k|\mathbf{y})$, $k = 1, \ldots, N$, is the BCJR algorithm, unaffordable for multilevel constellations when the channel memory, L, grows.

4) Turbo Equalization: The key point of turbo equalization is the iterative exchange of information between the equalizer and the decoder for the same set of received symbols [5], [7], [34]. Given the extrinsic log-likelihood ratios (LLR) from the equalizer to the decoder, $L_E(b_t|\mathbf{y})$, the latter computes (after one or more iterations) an estimation of the information bits, $\hat{\mathbf{m}}$, and an extrinsic LLR on the coded bits

$$L_D(b_t) = \log \frac{p(b_t = 0|L_E(\mathbf{b}|\mathbf{y}))}{p(b_t = 1|L_E(\mathbf{b}|\mathbf{y}))} - L_E(b_t|\mathbf{y}). \tag{5}$$

These LLRs are mapped again and delivered back to the equalizer as updated a priori probability, with a slight abuse of notation we denote it by $p_D(\mathbf{u})$. This process is repeated iteratively for a given maximum number of iterations, T, or until convergence.

B. Turbo LMMSE Equalization Algorithm

We include next the description of the turbo equalization with LMMSE, that we denote by T-LMMSE, since we will use it as benchmark and its structure is a good starting point to develop our solution. Given the CIR, the posterior approximation provided by the LMMSE equalizer is obtained by replacing the discrete uniform prior $p(\mathbf{u})$ in (4) by a product of independent Gaussian distributions with mean $\mathbb{E}[u_k] \in \mathbb{C}$ and variance $\mathbb{V}[u_k] \in \mathbb{R}_+$,

$$q_{MMSE}(\mathbf{u}) = \mathcal{CN}(\mathbf{y} : \mathbf{H}\mathbf{u}, \sigma_w^2 \mathbf{I}) \prod_{k=1}^{N} \mathcal{CN}(u_k : \mathbb{E}[u_k], \mathbb{V}[u_k]). \tag{6}$$

This distribution is a proper complex Gaussian [42]

$$q_{MMSE}(\mathbf{u}) = \mathcal{CN}(\mathbf{u} : \boldsymbol{\mu}_{MMSE}, \boldsymbol{\Sigma}_{MMSE}) \tag{7}$$

where

$$\boldsymbol{\Sigma}_{MMSE} = \left(\sigma_w^{-2} \mathbf{H}^{\mathrm{H}} \mathbf{H} + \boldsymbol{\Sigma}_{\mathbf{u}}^{-1}\right)^{-1}, \tag{8}$$

$$\boldsymbol{\mu}_{MMSE} = \boldsymbol{\Sigma}_{MMSE} \left(\sigma_w^{-2} \mathbf{H}^{\mathrm{H}} \mathbf{y} + \boldsymbol{\Sigma}_{\mathbf{u}}^{-1} \boldsymbol{\mu}_{\mathbf{u}}\right) \tag{9}$$

and $\boldsymbol{\mu}_{\mathbf{u}} = [\mathbb{E}[u_1], ..., \mathbb{E}[u_N]]^\top$, $\boldsymbol{\Sigma}_{\mathbf{u}} = \mathrm{diag}\left(\mathbb{V}[u_1], ..., \mathbb{V}[u_N]\right)$. The complexity of this solution is dominated by the matrix inversion in (8). For the first iteration, we set $\mathbb{E}[u_k] = 0$ and $\mathbb{V}[u_k] = E_s$, and the matrix yields a circulant matrix whose inverse can be computed with complexity of order $\mathcal{O}(N \log N)$ by means of the FFT. Then, the symbol probability of each entry is computed by independently deciding on each component.

In turbo equalization, the extrinsic information is passed to the channel decoder. The extrinsic information for symbol k is computed from (7), assuming equally probable symbols

for the a priori information of the k-th one. By using this turbo scheme in the receiver, the equalizer is fed back with the statistics $\mathbb{E}[u_k]$ and $\mathbb{V}[u_k]$, which are obtained from the updated a priori probability $p_D(u_k)$ in Fig. 1 as

$$\mathbb{E}(u_k) = \sum_{u \in \mathcal{A}} u \cdot p_D(u_k = u), \tag{10}$$

$$\mathbb{V}(u_k) = \sum_{u \in \mathcal{A}} (u - \mathbb{E}[u_k])^* (u - \mathbb{E}[u_k]) \cdot p_D(u_k = u). \tag{11}$$

At this point, it is important to remark that since the variances are no longer equal, the matrix to invert in (8) is not circulant. Therefore, the complexity of its inversion is of order $\mathcal{O}(N^2 L)$, as discussed later in this paper.

III. Expectation Propagation

Expectation propagation [26]–[28], [43] is a technique in Bayesian machine learning to approximate an intractable probability distribution, in which inference is unfeasible, by exponential family distributions. Suppose we are given some statistical distribution with hidden variables $\mathbf{x}$ and observables $\mathcal{D}$ that factors as

$$p(\mathbf{x}|\mathcal{D}) \propto f(\mathbf{x}) \prod_{i=1}^{\mathcal{I}} t_i(\mathbf{x}), \tag{12}$$

where $f(\mathbf{x})$ belongs to an exponential family $\mathcal{F}$ with sufficient statistics $\mathbf{\Phi}(\mathbf{x})$, and $t_i(\mathbf{x})$ are nonnegative factors that do not belong to $\mathcal{F}$, making direct inference over (12) not possible. EP provides a feasible approximation to $p(\mathbf{x}|\mathcal{D})$ by an exponential distribution $q(\mathbf{x})$ from $\mathcal{F}$ which factorizes as

$$q(\mathbf{x}) \propto f(\mathbf{x}) \prod_{i=1}^{\mathcal{I}} \tilde{t}_i(\mathbf{x}), \tag{13}$$

where factors $\tilde{t}_i(\mathbf{x}) \in \mathcal{F}$ are optimized to achieve an accurate global approximation $q(\mathbf{x}) \leftarrow p(\mathbf{x}|\mathcal{D})$, which optimally satisfies $\mathbb{E}_{q(\mathbf{x})}[\mathbf{\Phi}(\mathbf{x})] = \mathbb{E}_{p(\mathbf{x}|\mathcal{D})}[\mathbf{\Phi}(\mathbf{x})]$. This is known as the moment matching solution. A feasible algorithm to approximate this solution is the sequential EP algorithm [26], [27], which optimizes each factor $\tilde{t}_i(\mathbf{x})$ in turns independently in the context of all of the remaining factors. A sketch of the EP algorithm is given in Algorithm 1 where $q^{[\ell]}(\mathbf{x})$ is the approximation to $q(\mathbf{x})$ in (13) at iteration ℓ.

IV. Block-EP turbo equalizer

The EP is endowed with a great flexibility, given by the model in (13). In order to improve the accuracy of the EP solution, it is important to retain as much structure as possible from the true distribution and separate it from the latent (unknown) factors, $t_i(\mathbf{x})$ [26]. Bearing this in mind, we develop an EP approximation to (4), namely the posterior distribution of the transmitted symbols given the channel outcome $\mathbf{y}$.

A. The BEP equalizer

The following Gaussian exponential family will be considered to find a suitable approximation to (4):

$$q(\mathbf{u}) \propto \mathcal{CN}\big(\mathbf{y} : \mathbf{H}\mathbf{u}, \sigma_w^2 \mathbf{I}\big) \prod_{k=1}^{N} \exp\left(u_k^* \gamma_k + \gamma_k^* u_k - \Lambda_k u_k^* u_k\right), \tag{15}$$

Algorithm 1 The EP algorithm

Initialize approximating factors $\tilde{t}_i(\mathbf{x})$ and then $q(\mathbf{x})$ in (13).
repeat
  for $i = 1, ..., \mathcal{I}$ **do**
    1) Compute the distribution

$$\tilde{p}_i(\mathbf{x}) \propto t_i(\mathbf{x}) q^{[\ell]\setminus i}(\mathbf{x}) = t_i(\mathbf{x}) q^{[\ell]}(\mathbf{x}) / \tilde{t}_i^{[\ell]}(\mathbf{x}) \tag{14}$$

    and its moments, where $q^{[\ell]\setminus i}(\mathbf{x})$ is the so called *cavity* function.
    2) Compute the refined factor $\tilde{t}_i^{[\ell+1]}(\mathbf{x})$ by setting the moments of the distribution $\tilde{t}_i^{[\ell+1]}(\mathbf{x}) q^{[\ell]\setminus i}(\mathbf{x})$ equal to the moments of $\tilde{p}_i(\mathbf{x})$.
  end for
until convergence (or stopped criterion)

where the product of indicator functions is replaced by a product of univariate proper complex Gaussians, each parameterized by a (γ_k, Λ_k) pair, $k = 1, \ldots, N$. For any value $\gamma_k \in \mathbb{C}$ and $\Lambda_k \in \mathbb{R}_+$, $q(\mathbf{u})$ is also proper complex Gaussian $\mathcal{CN}(\mathbf{u} : \boldsymbol{\mu}, \mathbf{\Sigma})$ with

$$\mathbf{\Sigma} = \mathbf{R}^{-1} = (\sigma_w^{-2} \mathbf{H}^{\mathrm{H}} \mathbf{H} + \mathrm{diag}(\mathbf{\Lambda}))^{-1}, \tag{16}$$

$$\boldsymbol{\mu} = \mathbf{\Sigma}(\sigma_w^{-2} \mathbf{H}^{\mathrm{H}} \mathbf{y} + \boldsymbol{\gamma}), \tag{17}$$

exhibiting a similar structure to the LMMSE in (7). Based on these definitions in mind, we adapt the EP algorithm in Algorithm 1 to our setting. We denote the resulting algorithm block-EP (BEP) soft equalizer, and it can be found in Algorithm 2. A few remarks:

- Step 1) is the "LMMSE" step of the algorithm, as it requires computing $\mathbf{\Sigma}$ and $\boldsymbol{\mu}$ in (16) and (17) for the current configuration of $(\mathbf{\Lambda}, \boldsymbol{\gamma})$. Note that this step is equivalent to the LMMSE method in (10) and (11).
- Steps 2.1)-2.3) can be done in parallel for $k = 1, \ldots, N$. They can be seen as a refinement to the estimate computed at Step 1) by enforcing a discrete distribution. Given the factorization in (15), marginal $q^{[\ell]}(u_k)$, $k = 1, \ldots, N$ is proportional to $\exp\left(u_k^* \gamma_k + \gamma_k^* u_k - \Lambda_k u_k^* u_k\right)$. This term is canceled out in $\hat{p}^{[\ell]}(u_k)$ and the"true" discrete factor is introduced. Parameters canceled are recomputed by moment matching in (22)-(23). Eqn. (22) and (23) are proposed following the guidelines in [36, Eq. 35-36]. The parameter update in (22) may return a negative value for some k's. For those k's, we keep the values from the previous iteration. We introduce a smoothing parameter $\beta \in [0, 1]$ and a small constant ϵ. To avoid numerical instabilities, constant ϵ is the minimum allowed variance at each iteration, i.e., $\sigma_{p_k}^{2[\ell]} = \max(\epsilon, \sigma_{p_k}^{2[\ell]})$.

A block diagram of the BEP equalizer is included in Fig. 2 (a). The LMMSE block corresponds to Step 1). The grey block represents the refinement of the current marginals $q^{[\ell]}(u_k)$ through projection over discrete alphabet, whose output is fed back to the LMMSE block.

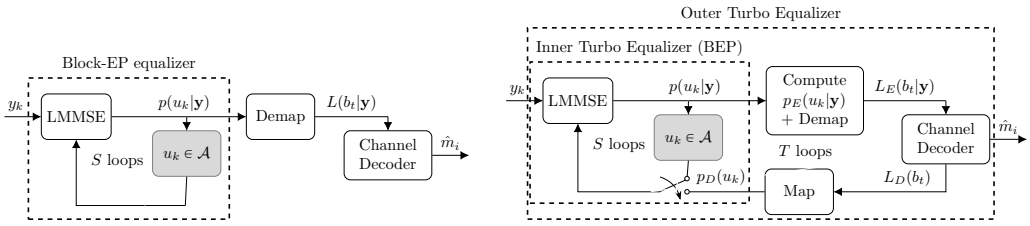


Fig. 2: In (a), BEP equalizer block diagram. In (b), Turbo BEP block diagram.

B. The Turbo BEP equalizer

The BEP equalizer can be further improved by using a turbo scheme. The resulting proposed equalizer, denoted as T-BEP, can be easily described as the T-LMMSE in Subsection II-B by just replacing (7) with the result of the BEP, Algorithm 2, in (15). A detailed implementation of the T-BEP is included in Algorithm 3. Also, a block diagram is included in Fig. 2(b). Note that the T-BEP can actually be seen as a turbo equalizer with two loops. First we run an *inner* turbo scheme, i.e. BEP. After S iterations of the BEP iterative procedure, the extrinsic LLR is given to the decoder in an *outer* loop, which is repeated T times. We propose to approximate the extrinsic probabilities by the cavity functions at the end of the EP algorithm, see (18). In this second stage the restrictions from the channel coding are exploited. The output of the channel decoder is used to initialize the BEP iterative procedure, whose outputs are then fed forward to the channel decoder. This is a major difference with respect to the previous scheme used in [5], [32], [34], where the estimates were refined only using the output of the decoder, as shown in Fig. 1, and the proposed inner loop was not present.

Algorithm 2 Block-EP equalizer (BEP)

Input: $(\gamma_k^{[1]}, \Lambda_k^{[1]})$ initialization.
for $\ell = 1, ..., S$ **do**
1) Calculate the moments of $q^{[\ell]}(\mathbf{u})$ in (15) for the current values of $\gamma_k \leftarrow \gamma_k^{[\ell]}$ and $\Lambda_k \leftarrow \Lambda_k^{[\ell]}$.
for $k = 1, ..., N$ **do**
2.1) Compute the k-th marginal of the distribution $q^{[\ell]}(\mathbf{u})$, denoted as $q^{[\ell]}(u_k) = \mathcal{CN}\Big(u_k : \mu_k^{[\ell]}, \sigma_k^{2[\ell]}\Big)$, and the distribution

$$q^{[\ell]\backslash k}(u_k) = \frac{q^{[\ell]}(u_k)}{\exp\left(u_k^* \gamma_k^{[\ell]} + \gamma_k^{[\ell]^*} u_k - \Lambda_k^{[\ell]} u_k^* u_k\right)} \sim \sim \mathcal{CN}\Big(u_k : z_k^{[\ell]}, v_k^{2[\ell]}\Big), \tag{18}$$

namely *cavity* marginal function, where

$$v_k^{2[\ell]} = \frac{\sigma_k^{2[\ell]}}{1 - \sigma_k^{2[\ell]} \Lambda_k^{[\ell]}}, \quad z_k^{[\ell]} = v_k^{2[\ell]} \left(\frac{\mu_k^{[\ell]}}{\sigma_k^{2[\ell]}} - \gamma_k^{[\ell]}\right).$$

2.2) Obtain the distribution $\hat{p}^{[\ell]}(u_k) \propto q^{[\ell]\backslash k}(u_k) \mathbb{I}_{u_k \in \mathcal{A}}$ and estimate its mean $\mu_{p_k}^{[\ell]}$ and variance $\sigma_{p_k}^{2[\ell]}$.
2.3) Set the mean and variance of the unnormalized Gaussian distribution

$$q^{[\ell]\backslash k}(u_k) \exp\left(u_k^* \gamma_{k,new}^{[\ell+1]} + \gamma_{k,new}^{[\ell+1]^*} u_k - \Lambda_{k,new}^{[\ell+1]} u_k^* u_k\right) \tag{19}$$

equal to $\mu_{p_k}^{[\ell]}$ and $\sigma_{p_k}^{2[\ell]}$. To this end, compute:

$$\Lambda_{k,new}^{[\ell+1]} = \left(\sigma_{p_k}^{-2[\ell]} - v_k^{-2[\ell]}\right), \tag{20}$$

$$\gamma_{k,new}^{[\ell+1]} = \left(\mu_{p_k}^{[\ell]} \sigma_{p_k}^{-2[\ell]} - z_k^{[\ell]} v_k^{-2[\ell]}\right). \tag{21}$$

2.4) Update the values as

$$\Lambda_k^{[\ell+1]} = \beta \Lambda_{k,new}^{[\ell+1]} + (1-\beta)\Lambda_k^{[\ell]}, \tag{22}$$

$$\gamma_k^{[\ell+1]} = \beta \gamma_{k,new}^{[\ell+1]} + (1-\beta)\gamma_k^{[\ell]}. \tag{23}$$

end for
end for
With the values $\boldsymbol{\gamma}^{[S+1]}, \boldsymbol{\Lambda}^{[S+1]}$ obtained after EP algorithm, calculate the final distribution $q(\mathbf{u})$ in (15).

Algorithm 3 Block-EP turbo equalizer (T-BEP)

0) Initialize $(\gamma_k^{[1]}, \Lambda_k^{[1]}) = (0, E_s^{-1})$.
for $t = 1, \ldots, T$ **do**
1) With the current initialization to $(\gamma_k^{[1]}, \Lambda_k^{[1]})$, run BEP equalizer in Algorithm 2.
2) Compute an estimate to the extrinsic LLRs, $L_E(b_t|\boldsymbol{y})$, by feeding

$$p_E(u_k|\mathbf{y}) = q^{[S+1]\backslash k}(u_k) \tag{24}$$

to the demapper. Feed $L_E(b_t|\boldsymbol{y})$ to the channel decoder.
3) From the channel decoder per-bit soft output, recompute a probability distribution for each symbol $p_D(u_k)$ from the decoder and compute its mean $\mathbb{E}[u_k]$ and variance $\mathbb{V}[u_k]$ given by (10) and (11).
4) Re-initialize $(\gamma_k^{[1]}, \Lambda_k^{[1]})$ to $(\mathbb{E}[u_k]\mathbb{V}[u_k]^{-1}, \mathbb{V}[u_k]^{-1})$.
end for

C. Efficient implementation

All $(\gamma_k^{[\ell]}, \Lambda_k^{[\ell]})$ pairs for $k = 1, ..., N$ can be updated in parallel. The most involved step is the computation of an N-dimensional inverse matrix in (16) for each ℓ-iteration, whose complexity is dominated by its size, i.e., $\mathcal{O}(N^3)$. Once this inverse is computed, the parallel update of all pairs $(\gamma_k^{[\ell]}, \Lambda_k^{[\ell]}) \leftarrow (\gamma_k^{[\ell+1]}, \Lambda_k^{[\ell+1]})$ by means of step 2 and 3 in

Algorithm 3 has a smaller computational complexity, linear in NM. To reduce the complexity of the matrix inversion, we propose to exploit the banded structure of the channel matrix along with the short length of the channel compared to N. The matrix $\mathbf{R}$ in (16) is a symmetric, positive-definite and banded matrix with bandwidth $2L-1$. We can decompose $\mathbf{R}$ using the band Cholesky factorization [44] such that

$$\mathbf{R} = \mathbf{G}\mathbf{G}^{\top}, \tag{25}$$

where $\mathbf{G}$ is a lower triangular banded matrix with bandwidth L that can be computed with NL^2 operations. Then, the inverse of the covariance matrix can be rewritten as

$$\mathbf{\Sigma} = \mathbf{R}^{-1} = \mathbf{G}^{-1^{\top}}\mathbf{G}^{-1}. \tag{26}$$

We invert matrix $\mathbf{G}$ by Gauss-Jordan elimination. For every diagonal element, say $G_{k,k}$, we divide row k of less than N non-null elements by $G_{k,k}$ and cancel all the $L-1$ lower elements of its column. Repeated for the whole diagonal yields a complexity of $\mathcal{O}(N^2L)$.

D. Convergence

Although the convergence is not guaranteed, we concluded empirically that in about $S = 10$ iterations the distribution $q(\mathbf{u})$ constructed in (15) tipically reaches a stationary value. Experimental results show that controlling numerical instabilities in the parameter updates in the turbo case ($T > 0$) is simply done by setting $\epsilon = 10^{-9}$ and $\beta = 0.1$. This solution is robust regardless the constellation order, SNR or channel realization. In the equalization case ($T = 0$), we have found that results can be improved if ϵ is first kept constant to a relatively high constant (0.5) to then reduce it. More precisely, we have used $\epsilon = 2^{-\max(\ell-5,1)}$. For equalizing 64-QAM constellations, the best performance has been reported by setting fixed to $\epsilon = 0.9$. We have selected the previous values after extensive experimentation as a trade off between convergence speed and accuracy. We emphasize again that the latter heuristics to control stability in the equalization case are not needed if the turbo scheme is used, as the feedback loop naturally stabilizes the BEP output. In Fig. 3, we include a representative example of the convergence by depicting the evolution of some components of the mean vector $\boldsymbol{\mu}$ in (17) and covariance matrix $\mathbf{\Sigma}$ in (16) along different values of S in the low-E_b/N_0 regime (specifically, $E_b/N_0 = 3$ dB) for a given observation, $\mathbf{y}$, with a 16-PAM modulation, $L = 4$ and $T = 0$. As shown in Fig. 3, approximately after $S = 10$ iterations the EP equalizer reaches a stationary value for the mean and variance, equal to the value provided by the optimal BCJR approach.

E. Performance analysis

We compare probabilistic equalizers using the mutual information between the transmitted symbol u_k and detected symbol $\hat{u}_k$, distributed according to the estimation of the posterior distribution of u_k given $\mathbf{y}$:

$$I(u_k,\hat{u}_k) = \sum_{u_k}\sum_{\hat{u}_k} p(u_k,\hat{u}_k)\log_2\frac{p(\hat{u}_k|u_k)}{p(\hat{u}_k)}, \tag{27}$$

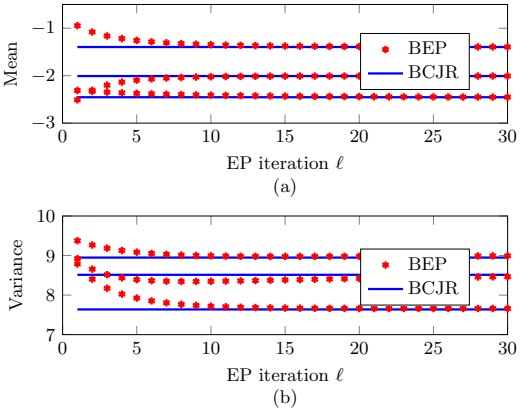


Fig. 3: Evolution of the mean (a) and variance (b) of 3 randomly chosen entries of the approximate posterior as EP iterates with 16-PAM, $L = 4$ and $E_b/N_0 = 3$ dB.

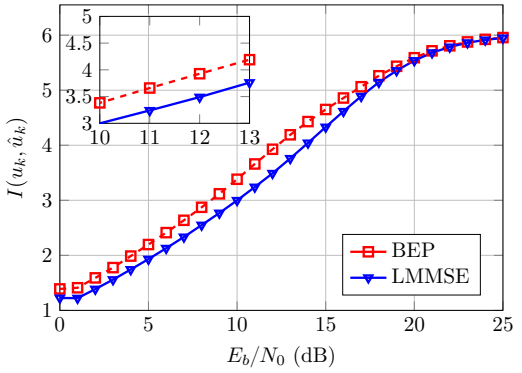


Fig. 4: Mutual information for BEP and LMMSE for 64-QAM and random channels with $L = 7$.

where note that $p(u_k,\hat{u}_k)$ is the joint probability distribution of u_k and $\hat{u}_k$ after marginalizing the channel output $\mathbf{y}$, the channel impulse response (in the case we consider it random) and the rest of symbols in the sequence $\mathbf{u}$. We resort to Monte Carlo estimates to generate samples from $p(u_k,\hat{u}_k)$ and then evaluate $I(u_k,\hat{u}_k)$ from these samples. We assume that the channel taps are Gaussian distributed, but are perfectly known at the receiver. First, we collect $N \in \mathbb{Z}_+$ samples from the joint distribution of $\mathbf{u},\mathbf{h},\mathbf{y}$ and $\hat{u}_k$ using standard sampling techniques in directed graphical models [43]. The key aspect is to note that, during the sampling procedure, $p(\hat{u}_k|\mathbf{y})$ is dependent on the considered probabilistic detection method. After N samples of possible $(u_k,\hat{u}_k)$ pairs have been collected, we use these samples to estimate $p(\hat{u}_k)$, $p(\hat{u}_k|u_k)$ and, finally, $I(u_k,\hat{u}_k)$ in (27). The higher $I(u_k,\hat{u}_k)$ is for each probabilistic equalizer, the closer we perform to channel capacity. In Fig. 4 we depict the mutual information in (27)

computed for $N = 10^6$ samples per E_b/N_0 point and averaged over all the symbols in the transmitted sequence, for the BEP and LMMSE algorithms, considering a 64-QAM constellation and channels of $L = 7$ complex-valued taps. Observe that, for very small E_b/N_0, the noise is so large that both methods achieve a small mutual information. On the other hand, for very large E_b/N_0 values, both methods converge to 6 bits, i.e., the number of bits transmitted per QAM symbol. Any probabilistic method will eventually saturate to this value. However, the key aspect is to be able to design a probabilistic equalizer able to improve the mutual information for intermediate E_b/N_0 values. Observe that before the saturation, the BEP achieves a gain w.r.t. LMMSE of around 1.5dB.

F. Computational Complexity

A detailed comparison of the complexity for the T-BEP and the T-LMMSE is included in Table I. We also include the computational complexity of the T-BCJR and its approximated approaches with turbo. From the computational point of view, as M and/or L grow, the BCJR and approximated approaches are unaffordable.

Algorithm	Complexity
T-BEP	$S'(LN^2+NM)+\alpha+S'T(LN^2+\alpha)$
T-LMMSE	$N\log N+\alpha+T(LN^2+\alpha)$
T-BCJR	$(T+1)(NM^L+\alpha)$
Approx. T-BCJR	$(T+1)(NM_eM+\alpha)$

TABLE I: Complexity comparison between algorithms, where $S' = S+1$ and α is the complexity of the LDPC decoder.

V. Experimental results

In this section, we illustrate the good performance of the BEP equalizer for different scenarios. Each channel tap is i.i.d. complex circular Gaussian distributed with zero mean and variance equal to $1/L$. The channel response is normalized. We average the BER over 1000 random frames per channel realization. We limit to 5 the absolute value of LLRs given to the decoder in order to avoid very confident probabilities which negatively affect its estimations. In the following, when mentioning approximated BCJR algorithms we refer to the M-BCJR [14], M*-BCJR [15], RS-BCJR [10], NZ and NZ-OS [17] solutions. We denote by NZ the approach consisting in running FT, BT or DT-NZ in [17] depending on the phase of the channel. If output saturation is used we refer to them as NZ-OS. We use a (3,6)-regular LDPC of rate 1/2, for a maximum of 200 iterations using the belief propagation as decoder [45], [46]. The codes were generated using the progressive edge-growth algorithm [47]. The maximum number of iterations of the LDPC decoder is set to 100 in every iteration of the turbo equalizer.

A. BEP soft equalizer

In Fig. 5 and Fig. 6 we show a comparison for some typical channels found in the literature using codewords of $V = 1024$ bits. In Fig. 5 we simulate the following scenario in [17]:

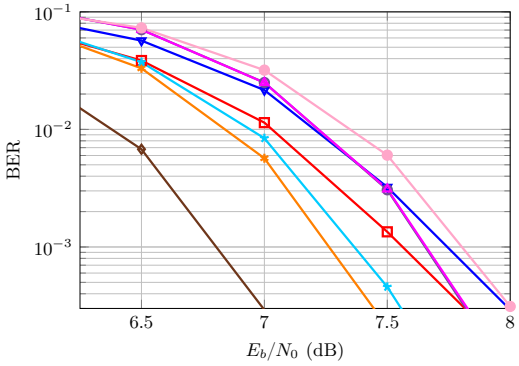


Fig. 5: BER for LMMSE (▽), BEP (□), BCJR (◇), M-BCJR (○), M*-BCJR (✳), RS-BCJR (★), NZ (△) and NZ-OS (●) equalizers for BPSK and the minimum phase channel $\mathbf{h} = \frac{1}{\sqrt{140}}[7\ 6\ 5\ 4\ 3\ 2\ 1]^\top$.

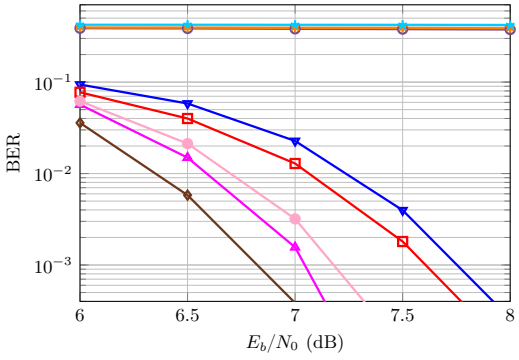


Fig. 6: BER for LMMSE (▽), BEP (□), BCJR (◇), M-BCJR (○), M*-BCJR (✳), RS-BCJR (★), NZ (△) and NZ-OS (●) equalizers for BPSK and the maximum phase channel $\mathbf{h} = \frac{1}{\sqrt{140}}[1\ 2\ 3\ 4\ 5\ 6\ 7]^\top$.

minimum phase channel $\mathbf{h} = [7\ 6\ 5\ 4\ 3\ 2\ 1]^\top/\sqrt{140}$, BPSK symbols and $M_e = 4$ states out of 64. In Fig. 6 we simulate the maximum phase channel $\mathbf{h} = [1\ 2\ 3\ 4\ 5\ 6\ 7]^\top/\sqrt{140}$ and BPSK symbols with $M_e = 8$ states for the approximated solutions as in [17]. To study the performance for other channels, in Fig. 7 and Fig. 8 we include the averaged BER over 100 random channels with $L = 5$ real-valued taps. We fix to $M_e = 8$ the number of states for the approximated solutions. In Fig. 7, we consider BPSK modulation, hence the BCJR has 16 states per step, while in Fig. 8 a 4-PAM is used, increasing the number of states to 256.

In Fig. 5 and Fig. 6 we can observe that approximated methods M-BCJR, M*-BCJR, RS-BCJR are quite sensitive to the channel realization. This is not only caused because they are based on just a forward or a backward strategy, but also because their parameters need to be tuned according to the particular channel. For this reason, the approximations M-BCJR, M*-BCJR and RS-BCJR fail with maximum phase

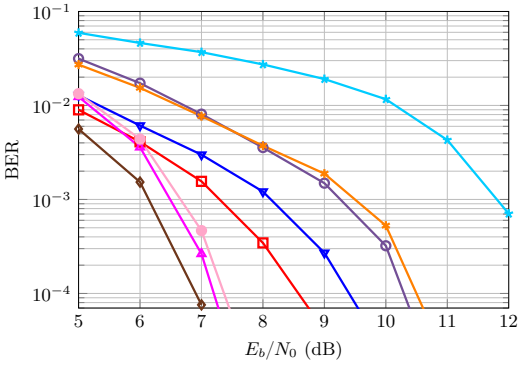


Fig. 7: BER for LMMSE (▽), BEP (□), BCJR (◇), M-BCJR (○), M*-BCJR (∗), RS-BCJR (⋆), NZ (△) and NZ-OS (•) equalizers for BPSK and 100 random channels with $L = 5$.

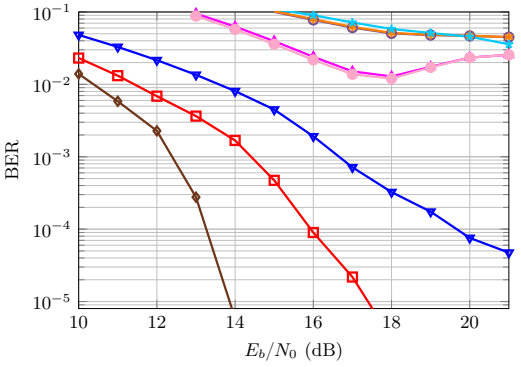


Fig. 8: BER for LMMSE (▽), BEP (□), BCJR (◇), M-BCJR (○), M*-BCJR (∗), RS-BCJR (⋆), NZ (△) and NZ-OS (•) equalizers for 4-PAM and 100 random channels with $L = 5$.

channels, as observed in Fig. 6. For averaged channels, see Fig. 7, NZ and NZ-OS exhibit a good performance at a low computational complexity due to the low state reduction ratio (M^L/M_e). However, these solutions fail if M_e does not grow accordingly, as can be observed in Fig. 8. This means that if the number of states, M_e, is kept as a fraction of the total number, M^L, they may exhibit a good performance. But if a large number of states is unaffordable, a fraction of it becomes also intractable. The LMMSE in these experiments does not fail as the approximated approaches do, but its performance degrades significantly compared to the BER of the BCJR in Fig. 8. Finally, the BEP exhibits a quite robust behavior, closer to the BCJR performance. Note here that the number of states is low, and that the BCJR can be used with optimal results at a low complexity.

For multilevel constellations and a channel with a few taps, the BCJR or their approximations are no longer computationally affordable. In the case of large dimensions, filter-based equalizers of the MMSE type are preferred [20]. In Fig. 9,

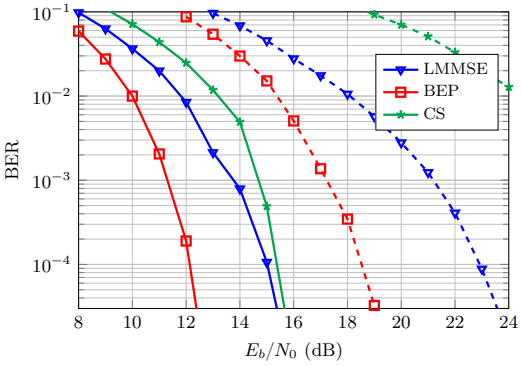


Fig. 9: BER for LMMSE (▽), BEP (□) and CS (⋆) equalizers for 16-QAM (solid lines) and 64-QAM (dashed lines) and 100 random channels with $L = 7$.

we depict the BER curves for BEP, LMMSE equalization and the algorithm in [19] that we denote as channel shortening (CS), considering codewords of $V = 4096$ bits, 100 random channels of $L = 7$ complex-valued taps and two different modulations: 16-QAM (solid lines) and 64-QAM (dashed lines). The CS algorithm has been simulated with a reduced memory $\nu = 2$ ($\nu = 1$) for the 16-QAM (64-QAM) case. We observe in this experiment that the BEP exhibits significant improvements with respect to the LMMSE and CS solution for both constellations.

B. Turbo BEP

To improve the estimates the turbo-equalization can be used. We simulate scenarios with large dimensions, where the only computationally feasible algorithms are the BEP and LMMSE solutions. In Fig. 10 we simulate a 16-QAM constellation, averaging over 100 random channels with a large memory, $L = 20$ complex-valued taps and codewords of $V = 1024$ bits. We represent the first 3 iterations of the turbo scheme, since we found no further improvement for $T \geqslant 3$. In Fig. 11 and Fig. 12, we depict the BER curves for turbo BEP and LMMSE equalization, averaged over 100 random channels of $L = 7$ complex-valued taps, 64-QAM modulation and codewords of $V = 1024$ and $V = 4096$ bits, respectively, for $T = 3$.

The results are remarkable. The BEP with no feedback from the decoder is 1.5-4.5 dB away from the LMMSE equalizer for BER=10^{-4}, depending on the channel length, L, and the constellation size, M. It even outperforms the turbo LMMSE equalizer. The BEP estimation can be further improved using the turbo scheme, in about 1 or 1.5 dB depending on the codeword length.

VI. Conclusions

When the number of states involved in the BCJR equalization or its approximations is high, their computational complexity is unaffordable. Approximated strategies are interesting whenever the number of states is not too high. Therefore,

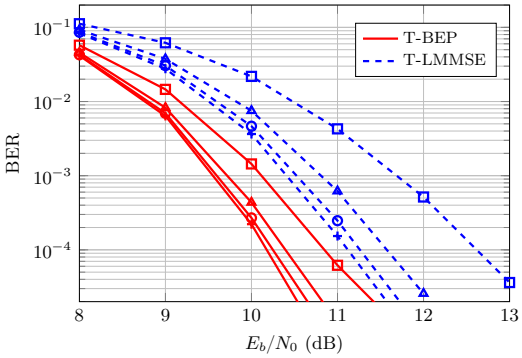


Fig. 10: BER for T-BEP (red lines, solid) and T-LMMSE (blue lines, dashed) equalizers using the outer loop for 16-QAM and 100 random channels with $L = 20$. No feedback (▫), one loop (▵), two loops (○) and three loops (+).

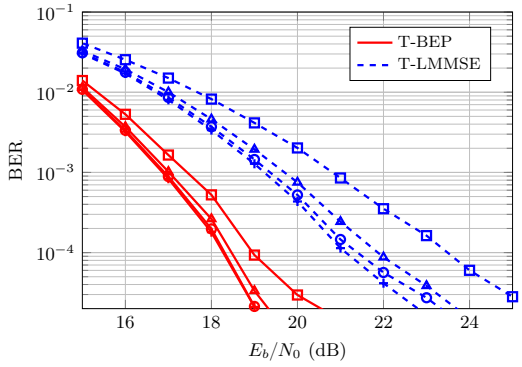


Fig. 11: BER for T-BEP (red lines, solid) and T-LMMSE (blue lines, dashed) equalizers using the outer loop for 64-QAM, 100 random channels with $L = 7$ and codewords of $V = 1024$ bits. No feedback (▫), one loop (▵), two loops (○) and three loops (+).

in the case of multilevel constellations with moderate or large channel lengths, it is preferable to resort to MMSE type equalizers. The turbo LMMSE equalization improves the estimations by feeding back the output of the channel decoder to the equalizer. In this paper we propose the EP as a novel alternative, where the whole posterior probabilities are approximated in a complex-valued formulation. This equalizer, denoted as BEP, can be cast as an LMMSE with an inner turbo scheme. The BEP exploits the fact that the transmitted symbols belong to a known constellation to improve the estimations. This soft equalizer outperforms the turbo LMMSE equalization. The BEP equalization can be further improved by exploiting the channel decoder output to propose the turbo BEP equalizer. This can be interpreted as an outer turbo equalization as compared to the proposed EP based inner turbo equalization. In the included experiments, we report gains in the range 1.5-5 dB with respect to the LMMSE. We have

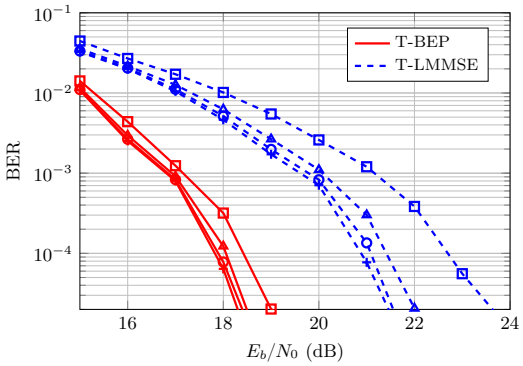


Fig. 12: BER for T-BEP (red lines, solid) and T-LMMSE (blue lines, dashed) equalizers using the outer loop for 64-QAM, 100 random channels with $L = 7$ and codewords of $V = 4096$ bits. No feedback (▫), one loop (▵), two loops (○) and three loops (+).

focused on block or batch solutions, to report computational complexities of order $\mathcal{O}(N^2L)$. The development of windowed or filtered solutions remains as future research, to reduce the computational complexity to be linear in the frame length.

References

[1] R. Schober, "Detection and estimation of signals in noise," University of British Columbia, Vancouver, Tech. Rep., 2010.

[2] J. G. Proakis, *Digital Communications*, 5th ed. New York, NY: McGraw-Hill, 2008.

[3] S. Haykin, *Communication Systems*, 5th ed. Wiley Publishing, 2009.

[4] L. Salamanca, J. J. Murillo-Fuentes, and F. Pérez-Cruz, "Bayesian equalization for LDPC channel decoding," *IEEE Trans. on Signal Processing*, vol. 60, no. 5, pp. 2672–2676, May 2012.

[5] M. Tüchler and A. Singer, "Turbo equalization: An overview," *IEEE Trans. on Information Theory*, vol. 57, no. 2, pp. 920–952, Feb 2011.

[6] C. Douillard, M. Jezequel, C. Berrou, P. Didier, and A. Picart, "Iterative correction of intersymbol interference: turbo-equalization," *European Trans. on Telecommunications*, vol. 6, no. 5, pp. 507–512, Sep 1995.

[7] R. Koetter, A. Singer, and M. Tüchler, "Turbo equalization," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 67–80, Jan 2004.

[8] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Trans. on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[9] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 498 –519, Feb 2001.

[10] G. Colavolpe, G. Ferrari, and R. Raheli, "Reduced-state BCJR-type algorithms," *IEEE Journal on Sel. Areas in Communications*, vol. 19, no. 5, pp. 848–859, May 2001.

[11] M. Eyuboglu and S. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback," *IEEE Trans. on Communications*, vol. 36, no. 1, pp. 13–20, Jan 1988.

[12] A. Duel-Hallen and C. Heegard, "Delayed decision-feedback sequence estimation," *IEEE Trans. on Communications*, vol. 37, no. 5, pp. 428–436, May 1989.

[13] P. Chevillat and E. Eleftheriou, "Decoding of trellis-encoded signals in the presence of intersymbol interference and noise," *IEEE Trans. on Communications*, vol. 37, no. 7, pp. 669–676, Jul 1989.

[14] V. Franz and J. Anderson, "Concatenated decoding with a reduced-search BCJR algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 186–195, Feb 1998.

[15] M. Sikora and D. Costello, "A new SISO algorithm with application to turbo equalization," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Sep 2005, pp. 2031–2035.

[16] C. Vithanage, C. Andrieu, and R. Piechocki, "Approximate inference in hidden Markov models using iterative active state selection," *IEEE Signal Processing Letters*, vol. 13, no. 2, pp. 65–68, Feb 2006.
[17] D. Fertonani, A. Barbieri, and G. Colavolpe, "Reduced-complexity BCJR algorithm for turbo equalization," *IEEE Trans. on Communications*, vol. 55, no. 12, pp. 2279–2287, Dec 2007.
[18] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 2nd ed. Prentice Hall, 1996.
[19] F. Rusek and A. Prlja, "Optimal channel shortening for MIMO and ISI channels," *IEEE Trans. on Wireless Communications*, vol. 11, no. 2, pp. 810–818, Feb 2012.
[20] C. Berrou, *Codes and turbo codes*, ser. Collection IRIS. Springer Paris, 2010.
[21] G. Colavolpe and G. Germi, "On the application of factor graphs and the sum-product algorithm to ISI channels," *IEEE Trans. on Communications*, vol. 53, no. 5, pp. 818–825, May 2005.
[22] J. Hu, H. Loeliger, J. Dauwels, and F. Kschischang, "A general computation rule for lossy summaries/messages with examples from equalization," in *Proc. 44th Allerton Conf. Communication, Control, and Computing*, Sep 2006, pp. 27–29.
[23] Q. Guo and L. Ping, "LMMSE turbo equalization based on factor graphs," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 2, pp. 311–319, Feb 2008.
[24] G. Colavolpe, D. Fertonani, and A. Piemontese, "SISO detection over linear channels with linear complexity in the number of interferers," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 8, pp. 1475–1485, Dec 2011.
[25] P. Sun, C. Zhang, Z. Wang, C. Manchon, and B. Fleury, "Iterative receiver design for ISI channels using combined belief- and expectation-propagation," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1733–1737, Oct 2015.
[26] T. P. Minka, "A family of algorithms for approximate Bayesian inference," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.
[27] T. Minka, "Expectation propagation for approximate Bayesian inference," in *Proc. 17th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2001, pp. 362–369.
[28] M. Seeger, "Expectation propagation for exponential families," *Univ. Calif., Berkeley, CA, USA, Tech. Rep.*, 2005.
[29] K. Muranov, "Survey of MMSE channel equalizers," University of Illinois, Chicago, Tech. Rep.
[30] J. Wu and Y. Zheng, "Low complexity soft-input soft-output block decision feedback equalization," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, Nov 2007, pp. 3379–3383.
[31] S. Kim, J. Lee, and Y. Kim, "Adaptive Cholesky based MMSE equalizer in GSM," in *Proc. IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, Nov 2008, pp. 886–889.
[32] M. Tüchler, A. Singer, and R. Koetter, "Minimum mean squared error equalization using a priori information," *IEEE Trans. on Signal Processing*, vol. 50, no. 3, pp. 673–683, Mar 2002.
[33] L. Liu and L. Ping, "An extending window MMSE turbo equalization algorithm," *IEEE Signal Processing Letters*, vol. 11, no. 11, pp. 891–894, Nov 2004.
[34] M. Tüchler, R. Koetter, and A. Singer, "Turbo equalization: principles and new results," *IEEE Trans. on Communications*, vol. 50, no. 5, pp. 754–767, May 2002.
[35] M. Senst and G. Ascheid, "How the framework of expectation propagation yields an iterative IC-LMMSE MIMO receiver," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, Dec 2011, pp. 1–6.
[36] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Pérez-Cruz, "Expectation propagation detection for high-order high-dimensional MIMO systems," *IEEE Trans. on Communications*, vol. 62, no. 8, pp. 2840–2849, Aug 2014.
[37] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Pérez-Cruz, "Improved performance of LDPC-coded MIMO systems with EP-based soft-decisions," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Jun 2014, pp. 1997–2001.
[38] P. M. Olmos, J. J. Murillo-Fuentes, and F. Pérez-Cruz, "Tree-structure expectation propagation for LDPC decoding over the BEC," *IEEE Trans. on Information Theory*, vol. 59, no. 6, pp. 3354–3377, 2013.
[39] L. Salamanca, P. M. Olmos, F. Pérez-Cruz, and J. J. Murillo-Fuentes, "Tree-structured expectation propagation for LDPC decoding over BMS channels," *IEEE Trans. on Communications*, vol. 61, no. 10, pp. 4086–4095, Oct 2013.
[40] Y. Qi and T. Minka, "Window-based expectation propagation for adaptive signal detection in flat-fading channels," *IEEE Trans. on Wireless Communications*, vol. 6, no. 1, pp. 348–355, Jan 2007.
[41] I. Santos, J. J. Murillo-Fuentes, and P. M. Olmos, "Block expectation propagation equalization for ISI channels," in *Proc. 23rd European Signal Processing Conference (EUSIPCO)*, Sep 2015, pp. 379–383.
[42] P. J. Schreier and L. L. Scharf, *Statistical Signal Processing of Complex-Valued Data. The Theory of Improper and Noncircular Signals*. Cambridge, UK: Cambridge University Press, 2010.
[43] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag, New York, 2006.
[44] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: Johns Hopkins University Press, 1996.
[45] T. J. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, Mar. 2008.
[46] L. Salamanca, J. J. Murillo-Fuentes, P. M. Olmos, F. Pérez-Cruz, and S. Verdú, "Approaching the DT bound using linear codes in the short blocklength regime," *IEEE Communications Letters*, vol. 19, no. 2, pp. 123–126, Feb 2015.
[47] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. on Information Theory*, vol. 51, no. 1, pp. 386–398, 2005.

Appendix B
Paper II

Authors: Irene Santos, Juan José Murillo-Fuentes, Eva Arias de Reyna and Pablo M. Olmos.
Title: Turbo EP-based Equalization: a Filter-Type Implementation.
Journal: IEEE Transactions on Communications.
Date: September 2017.
Status: Accepted.
Abstract: We propose a novel filter-type equalizer to improve the solution of the linear minimum-mean squared-error (LMMSE) turbo equalizer, with computational complexity constrained to be quadratic in the filter length. When high-order modulations and/or large memory channels are used the optimal BCJR equalizer is unavailable, due to its computational complexity. In this scenario, the filter-type LMMSE turbo equalization exhibits a good performance compared to other approximations. In this paper, we show that this solution can be significantly improved by using expectation propagation (EP) in the estimation of the a posteriori probabilities. First, it yields a more accurate estimation of the extrinsic distribution to be sent to the channel decoder. Second, compared to other solutions based on EP the computational complexity of the proposed solution is constrained to be quadratic in the length of the finite impulse response (FIR). In addition, we review previous EP-based turbo equalization implementations. Instead of considering default uniform priors we exploit the outputs of the decoder. Some simulation results are included to show that this new EP-based filter remarkably outperforms the turbo approach of previous versions of the EP algorithm and also improves the LMMSE solution, with and without turbo equalization.

Turbo EP-based Equalization: a Filter-Type Implementation

Irene Santos, Juan José Murillo-Fuentes, Eva Arias-de-Reyna, and Pablo M. Olmos

***Abstract*—We propose a novel filter-type equalizer to improve the solution of the linear minimum-mean squared-error (LMMSE) turbo equalizer, with computational complexity constrained to be quadratic in the filter length. When high-order modulations and/or large memory channels are used the optimal BCJR equalizer is unavailable, due to its computational complexity. In this scenario, the filter-type LMMSE turbo equalization exhibits a good performance compared to other approximations. In this paper, we show that this solution can be significantly improved by using expectation propagation (EP) in the estimation of the a posteriori probabilities. First, it yields a more accurate estimation of the extrinsic distribution to be sent to the channel decoder. Second, compared to other solutions based on EP the computational complexity of the proposed solution is constrained to be quadratic in the length of the finite impulse response (FIR). In addition, we review previous EP-based turbo equalization implementations. Instead of considering default uniform priors we exploit the outputs of the decoder. Some simulation results are included to show that this new EP-based filter remarkably outperforms the turbo approach of previous versions of the EP algorithm and also improves the LMMSE solution, with and without turbo equalization.**

***Index Terms*—Expectation propagation (EP), linear MMSE, low-complexity, turbo equalization, ISI, filter-type equalizer.**

I. Introduction

MANY digital communication systems need to transmit over channels that are affected by inter-symbol interference (ISI). The equalizer produces a probabilistic estimation of the transmitted data given the vector of observations [1]. Significant improvements are found when the previous estimation is given to a probabilistic channel decoder [2]. Equalization can be done in the frequency domain to avoid complexity problems associated with the inverse of covariance matrices [3]. In addition, feeding the equalizer back again with the output of the decoder, iteratively, yields a turbo-equalization scheme that significantly reduces the overall error rate [4]–[6].

The BCJR algorithm [7] performs optimal turbo equalization under the maximum a posteriori (MAP) criterion. It provides a posteriori probability (APP) estimations given some *a priori* information about the transmitted data. However, its complexity grows exponentially with the length of the channel and the constellation size, becoming intractable for few taps and/or multilevel constellations. In this situation, approximated BCJR solutions, such as [8]–[11], can be used. They are based on a search over a simplified trellis with only M_e states, yielding a complexity which is linear in this number of states. However, the performance of these approaches is quite dependent on the channel realization and the order of the constellation used. In addition, these approximated BCJR solutions degrade rapidly if the number of survivor paths does not grow according to the total number of states. For these reasons, filter-based equalizers are preferred [12].

A quite extended filter type equalizer in the literature is based on the well-known linear minimum-mean squared-error (LMMSE) algorithm [5], [13], [14]. This LMMSE filter is an appealing alternative where the BCJR is computationally unfeasible due to its robust performance with linear complexity in the frame length, `N`, and quadratic dependence with the window length, `W`. From a Bayesian point of view, the LMMSE algorithm obtains a Gaussian extrinsic distribution by replacing the discrete prior distribution of the transmitted symbols with a Gaussian prior.

A more accurate estimation for the extrinsic distribution can be obtained by replacing the prior distributions with approximations of the probability distribution. This can be done by means of the expectation propagation (EP) algorithm. The EP approach projects the approximated posterior distribution into the family of Gaussians by matching its moments iteratively with the ones of the true posterior. This algorithm has been already successfully applied to multiple-input multiple-output (MIMO) systems [15] and low-density parity-check (LDPC) channel decoding [16], [17], among others. It has been also applied to turbo equalization in a message passing approach as a way to incorporate into the BP algorithm the discrete information coming from the channel decoder [18], [19]. These message passing methods reduce to the LMMSE estimation if no turbo equalization is employed. A different approach is proposed in [20], [21] under the name of block EP (BEP) where, rather than applying EP after the channel decoder, it is used within the equalizer to better approximate the posterior, outperforming previous solutions.

The computational complexity of previous EP-based equalizers is large for long frame lengths or memories of the channel. Due to its block implementation, the complexity of the BEP is quadratic in the frame length, becoming intractable for large frames [21]. To overcome this drawback, a smoothing EP (SEP) implementation is proposed in [22], but its complexity

I. Santos, J.J. Murillo-Fuentes and E. Arias-de-Reyna are with the Dept. Teoría de la Señal y Comunicaciones, Escuela T. Superior de Ingeniería, Universidad de Sevilla, Camino de los Descubrimiento s/n, 41092 Sevilla, Spain. E-mail: {irenesantos,murillo,earias}@us.es

P. M. Olmos is with the Dept. Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid, Avda. de la Universidad 30, 28911, Leganés (Madrid), Spain. He is also with the Instituto de Investigación Sanitaria Gregorio Marañón (IiSGM). E-mail: olmos@tsc.uc3m.es.

The final version of the manuscript can be found at https://ieeexplore.ieee.org/document/8353388/. This work was partially funded by Spanish government (Ministerio de Economía y Competitividad TEC2016-78434-C3-{2-3}-R and Juan de la Cierva Grant No. IJCI-2014-19150) and by the European Union (FEDER).

is cubic with the memory of the channel. Furthermore, due to their iterative procedure, their computational load is roughly S times the one of the LMMSE counterparts, where S is the number of iterations used in the EP algorithm, typically around 10 [15], [20], [21]. Besides, in both, BEP and SEP, uniform discrete priors are assumed for the constellation of the modulations when computing the EP approximations, even within the turbo equalization iterations, while the use of information from the decoder remains unexplored.

The results developed in this paper focus on improving these previous EP-based equalizers [21], [22] both in computational complexity and performance. First, we improve the prior information used in the equalizer once the turbo procedure starts, forcing the true discrete prior to be non-uniform in contrast to the uniform priors used by previous EP-based approaches. As a result, we achieve a performance improvement. Second, the computational complexity of the EP algorithm is reduced to roughly a third part of that in [21], by optimizing the choice of EP parameters. Third, and most important, a new filter-type EP solution is designed. This solution is constrained to have linear complexity in the frame length and quadratic in the filter length, i.e., it is endowed with the same complexity order than the LMMSE filter.

The novel EP-based filter proposed outperforms the LMMSE algorithm with a robust behavior to changes in the constellation size and the channel realization, as the BEP and SEP approaches do [21], [22]. In the experiments included, we show that the EP filter solution greatly improves the LMMSE solution with and without turbo equalization, specifically we have 2 dB gains for a BPSK, 5 dB for the 8-PSK and 6-13 dB for 16 and 64-QAM, respectively. In comparison with previous EP approaches, the EP filter matches their performance with BPSK constellations, and outperforms them with gains of 2 dBs for 8-PSK and 4-5 dBs for 16 and 64-QAM. We study the extrinsic information transfer (EXIT) charts [5], [23] of our proposal for a BPSK, where the EP-based filter achieves the same performance as the BEP.

The scope of this paper encompasses time domain equalization. Frequency domain equalization has received a lot of attention as it usually achieves a complexity reduction for the block-wise processing [3], [14], [24], [25]. For this reason, derivation of a frequency domain counterpart for the proposed EP based turbo-equalizer remains as a future research line. Another promising research route is the application to MIMO with channels with memory [3], [26], [27].

The paper is organized as follows. We first describe in Section II the model of the communication system at hand. Section III is devoted to develop a new implementation of the EP-based equalizer considering non-uniform priors and studies the optimal values for the parameters. In Section IV, we review the formulation for the LMMSE filter in turbo equalization and describe the novel EP filter-type solution proposed. In Section VI, we include several simulations to compare both EP and LMMSE approaches. We end with conclusions.

Through the paper, we denote the i-th entry of a vector $\mathbf{u}$ as u_i, its complex conjugate as $\mathbf{u}^*$ and its Hermitian transpose as $\mathbf{u}^{\mathrm{H}}$. We define $\delta(u_i)$ as the delta function that takes value one if $u_i = 0$ and zero in other case. We use $\mathcal{CN}(\mathbf{u} : \boldsymbol{\mu}, \boldsymbol{\Sigma})$ to denote a normal distribution of a random proper complex vector $\mathbf{u}$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

II. System Model

The model of the communication system is depicted in Fig. 1, including turbo equalization at the receiver. There are three main blocks: transmitter, channel and turbo receiver.

A. Transmitter

The information bit sequence, $\mathbf{a} = [a_1, ..., a_K]^\top$ where $a_i \in \{0, 1\}$, is encoded into the coded bit vector $\mathbf{b} = [b_1, ..., b_V]^\top$ with a code rate equal to $R = K/V$. After permuting the bits with an interleaver, the codeword $\mathbf{c} = [c_1, ..., c_V]^\top$ is partitioned into $\mathtt{N}$ blocks of length $Q = \log_2(\mathtt{M})$, $\mathbf{c} = [\mathbf{c}_1, ..., \mathbf{c}_\mathtt{N}]^\top$ where $\mathbf{c}_k = [c_{k,1}, ..., c_{k,Q}]$, and modulated with a complex $\mathtt{M}$-ary constellation $\mathcal{A}$ of size $|\mathcal{A}| = \mathtt{M}$. These modulated symbols, $\mathbf{u} = [u_1, ..., u_\mathtt{N}]^\top$, where each component $u_k = \mathcal{R}(u_k) + j\mathcal{I}(u_k) \in \mathcal{A}$, are transmitted over the channel. Hereafter, transmitted symbol energy and energy per bit are denoted as E_s and E_b, respectively.

B. Channel

The channel is completely specified by the CIR, i.e., $\mathbf{h} = [h_1, ..., h_\mathtt{L}]^\top$, where $\mathtt{L}$ is the number of taps, and is corrupted with AWGN whose noise variance, σ_w^2, is known. Each k-th entry of the complex received signal $\mathbf{y} = [y_1, ..., y_{\mathtt{N}+\mathtt{L}-1}]^\top$ is given by

$$y_k = \sum_{j=1}^{\mathtt{L}} h_j u_{k-j+1} + w_k = \mathbf{h}^\top \mathbf{u}_{k:k-\mathtt{L}+1} + w_k, \qquad (1)$$

where $w_k \sim \mathcal{CN}(w_k : 0, \sigma_w^2)$ and $u_k = 0$ for $k < 1$ and $k > \mathtt{N}$.

C. Turbo receiver

When no information is available from the channel decoder, the posterior probability of the transmitted symbol vector $\mathbf{u}$ given the whole vector of observations $\mathbf{y}$ yields

$$p(\mathbf{u}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{u})p(\mathbf{u}) \qquad (2)$$

where, assuming equiprobable symbols, the prior would be given by

$$p(\mathbf{u}) = \frac{1}{\mathtt{M}} \prod_{k=1}^{\mathtt{N}} \sum_{u \in \mathcal{A}} \delta(u_k - u). \qquad (3)$$

This prior matches with the definition given in [21] but, as explained below, it is just valid before the turbo procedure.

In a turbo architecture the equalizer and decoder iteratively exchange information for the same set of received symbols [5], [14]. Traditionally, this exchange of information is done in terms of extrinsic probabilities in order to improve convergence and avoid instabilities. The extrinsic information at the output of the equalizer (see Fig. 1), $p_E(u_k|\mathbf{y})$, is computed so as to meet the turbo principle [13]. These probabilities, $p_E(u_k|\mathbf{y})$, are approximated when the optimal solution is intractable. We will denote the approximation by $q_E(u_k)$.

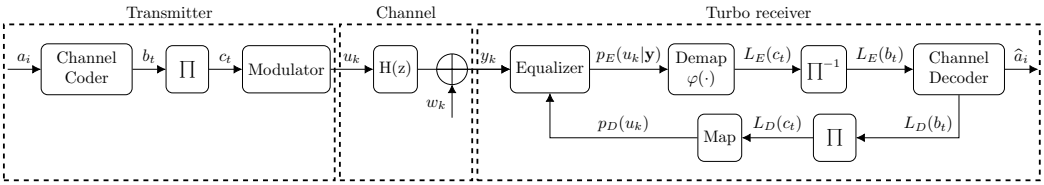


Fig. 1: System model.

The extrinsic distributions are demapped,

$$L_E(c_{k,j}) = \log \frac{\sum_{u_k \in \mathcal{A} | c_{k,j}=0} p_E(u_k|\mathbf{y})}{\sum_{u_k \in \mathcal{A} | c_{k,j}=1} p_E(u_k|\mathbf{y})}, \tag{4}$$

deinterleaved and given to the decoder as extrinsic log-likelihood ratios, $L_E(b_t)$. The channel decoder computes an estimation of the information bits, $\widehat{\mathbf{a}}$, along with the extrinsic LLRs on the coded bits, computed as

$$L_D(b_t|L_E(\mathbf{b})) = \log \frac{p(b_t=0|L_E(\mathbf{b}))}{p(b_t=1|L_E(\mathbf{b}))} - L_E(b_t). \tag{5}$$

These extrinsic LLRs are interleaved, mapped again and given to the equalizer as updated priors, $p_D(\mathbf{u}|L_E(\mathbf{b}))$, which are computed as

$$p_D(u_k|L_E(\mathbf{b})) = \sum_{u \in \mathcal{A}} \delta(u_k - u) \prod_{j=1}^{Q} p_D(c_{k,j} = \varphi_j(u)|L_E(\mathbf{b})), \tag{6}$$

with $\varphi_j(u)$ denoting the j-th bit associated to the demapping of symbol u. This process is repeated iteratively for a given maximum number of iterations, T, or until convergence. Note that in Fig. 1 we have included the computation of the extrinsic information within the equalization and channel decoding blocks. Note also that, once the turbo procedure starts, the prior in (2) is conditioned on the input at the channel decoder and the symbols are not equiprobable anymore. In this situation, the posterior distribution computed by the equalizer is given by

$$p(\mathbf{u}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{u}) \prod_{k=1}^{\mathtt{N}} p_D(u_k|L_E(\mathbf{b})), \tag{7}$$

The true posterior distribution in (2) and (7) has complexity proportional to $\mathtt{M}^{\mathtt{L}}$. When this complexity becomes intractable, we will approximate it, denoting it as $q(\mathbf{u})$. In the following, we omit the dependence on the input at the decoder, $L_E(\mathbf{b})$, to keep the notation uncluttered in the rest of the paper. It is also uncluttered in Fig. 1.

III. Non-uniform BEP turbo equalizer

EP [28]–[32] is a technique in Bayesian machine learning that approximates a (non-exponential) distribution with an exponential distribution whose moments match the true ones. In this paper, we focus on computing a Gaussian approximation for the posterior in (7), which is clearly non Gaussian due to the product of discrete priors in (6). As introduced in [21], this is done by iteratively updating an approximation within the Gaussian exponential family by replacing the non Gaussian prior terms in (2) by a product of Gaussians[1], i.e.,

$$\begin{aligned} q^{[\ell]}(\mathbf{u}) &\propto p(\mathbf{y}|\mathbf{u}) \prod_{k=1}^{\mathtt{N}} \tilde{p}_D^{[\ell]}(u_k) \\ &= \mathcal{CN}(\mathbf{y} : \mathbf{H}\mathbf{u}, \sigma_w^2 \mathbf{I}) \prod_{k=1}^{\mathtt{N}} \mathcal{CN}\Big(u_k : m_k^{[\ell]}, \eta_k^{[\ell]}\Big) \end{aligned} \tag{8}$$

The marginalization of the resulting approximated Gaussian posterior distribution for the k-th transmitted symbol and ℓ-th EP iteration yields

$$q^{[\ell]}(u_k) \sim \mathcal{CN}\Big(u_k : \mu_k^{[\ell]}, s_k^{2[\ell]}\Big) \tag{9}$$

where

$$\begin{aligned} \mu_k^{[\ell]} &= m_k^{[\ell]} + \\ &+ \eta_k^{[\ell]} \mathbf{h_k}^{\mathrm{H}} \left(\sigma_w^2 \mathbf{I} + \mathbf{H} \mathrm{diag}(\boldsymbol{\eta}^{[\ell]}) \mathbf{H}^{\mathrm{H}}\right)^{-1} (\mathbf{y} - \mathbf{H}\mathbf{m}^{[\ell]}), \end{aligned} \tag{10}$$

$$s_k^{2[\ell]} = \eta_k^{[\ell]} - \eta_k^{2[\ell]} \mathbf{h_k}^{\mathrm{H}} \left(\sigma_w^2 \mathbf{I} + \mathbf{H} \mathrm{diag}(\boldsymbol{\eta}^{[\ell]}) \mathbf{H}^{\mathrm{H}}\right)^{-1} \mathbf{h_k}, \tag{11}$$

$\mathbf{H}$ is the $\mathtt{N} + \mathtt{L} - 1 \times \mathtt{N}$ channel matrix given by

$$\mathbf{H} = \begin{bmatrix} h_1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ h_{\mathtt{L}} & & \ddots & 0 \\ 0 & \ddots & & h_1 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & h_{\mathtt{L}} \end{bmatrix} \tag{12}$$

and $\mathbf{h_k}$ is the k-th column of $\mathbf{H}$ (see Appendix A for the demonstration). At this point it is interesting to remark that (9)-(11) are completely equivalent to equations (15)-(17) in [21]. Here we developed the values of the mean and variance for each symbol while in [21] they were computed in block form. The current description is simpler because we only include the elements of the covariance matrix that are used during the execution of the algorithm, excluding the non-diagonal elements.

[1]Note that in [21] we used an alternative expression for (8) (an exponential distribution with parameters $\gamma_k = m_k/\eta_k$ and $\Lambda_k = 1/\eta_k$).

The mean and variance parameters in (8) are initialized with the statistics from the channel decoder as

$$m_k^{[1]} = \sum_{u\in\mathcal{A}} u \cdot p_D(u_k = u), \tag{13}$$

$$\eta_k^{[1]} = \sum_{u\in\mathcal{A}} (u - m_k^{[1]})^*(u - m_k^{[1]}) \cdot p_D(u_k = u). \tag{14}$$

Then, they are updated in parallel and iteratively by matching the moments of the following distributions

$$q_E^{[\ell]}(u_k)p_D(u_k) \overset{\text{moment matching}}{\longleftrightarrow} q_E^{[\ell]}(u_k)\mathcal{CN}\Big(u_k : m_k^{[\ell+1]}, \eta_k^{[\ell+1]}\Big) \tag{15}$$

where $q_E^{[\ell]}(u_k)$ is an extrinsic marginal distribution computed as[2]

$$q_E^{[\ell]}(u_k) = q^{[\ell]}(u_k)/\tilde{p}_D^{[\ell]}(u_k) = \mathcal{CN}\Big(u_k : z_k^{[\ell]}, v_k^{2[\ell]}\Big) \tag{16}$$

where

$$z_k^{[\ell]} = \frac{\mu_k^{[\ell]}\eta_k^{[\ell]} - m_k^{[\ell]}s_k^{2[\ell]}}{\eta_k^{[\ell]} - s_k^{2[\ell]}}, \tag{17}$$

$$v_k^{2[\ell]} = \frac{s_k^{2[\ell]}\eta_k^{[\ell]}}{\eta_k^{[\ell]} - s_k^{2[\ell]}}. \tag{18}$$

Note that this equalizer differs from the one in [21] because we used different definitions for the true prior, $p_D(u_k)$. In the current manuscript, we considered non-uniform and discrete priors, given by (6), during the moment matching procedure in the equalizer, while in [21] uniform priors as in (3) were considered by default even after the turbo procedure. To increase the accuracy of the algorithm, a damping procedure follows the moment matching in (15). We have defined an algorithm, described in Algorithm 1, called *Moment Matching and Damping* that runs these two procedures.

A. The nuBEP algorithm

Algorithm 2 contains a detailed description of the whole EP procedure, where S is the number of EP iterations while T is the number of turbo iterations. Note that the difference with the approach in [21] lies in the definition of the prior distribution used during the moment matching procedure. In [21], we use an uniform distribution (denoted with the indicator function), forcing the same *a priori* probability for the symbols, regardless of the information fed back from the decoder, during the moment matching employed in the equalizer even after the turbo procedure starts. In this paper, we refine the definition of the prior used in the moment matching of EP algorithm as in (6), considering non-uniform priors once the turbo procedure starts. For this reason, we named this algorithm non-uniform BEP (nuBEP) turbo equalizer.

[2]Note that in this paper we used $q_E^{[\ell]}(u_k)$ to denote the extrinsic marginal distribution, while in [21] we denoted as $q^{[\ell]\setminus k}(u_k)$ and called it cavity marginal function.

Algorithm 1 Moment Matching and Damping

Given inputs: $\mu_{p_k}^{[\ell]}, \sigma_{p_k}^{2[\ell]}, z_k^{[\ell]}, v_k^{2[\ell]}, m_k^{[\ell]}, \eta_k^{[\ell]}$

1) Run moment matching: Set the mean and variance of the unnormalized Gaussian distribution

$$q_E^{[\ell]}(u_k) \cdot \mathcal{CN}\Big(u_k : m_{k,new}^{[\ell+1]}, \eta_{k,new}^{[\ell+1]}\Big) \tag{19}$$

equal to $\mu_{p_k}^{[\ell]}$ and $\sigma_{p_k}^{2[\ell]}$, to get the solution

$$\eta_{k,new}^{[\ell+1]} = \frac{\sigma_{p_k}^{2[\ell]}v_k^{2[\ell]}}{v_k^{2[\ell]} - \sigma_{p_k}^{2[\ell]}}, \tag{20}$$

$$m_{k,new}^{[\ell+1]} = \eta_{k,new}^{[\ell+1]}\left(\frac{\mu_{p_k}^{[\ell]}}{\sigma_{p_k}^{2[\ell]}} - \frac{z_k^{[\ell]}}{v_k^{2[\ell]}}\right). \tag{21}$$

2) Run damping: Update the values as

$$\eta_k^{[\ell+1]} = \left(\beta\frac{1}{\eta_{k,new}^{[\ell+1]}} + (1-\beta)\frac{1}{\eta_k^{[\ell]}}\right)^{-1}, \tag{22}$$

$$m_k^{[\ell+1]} = \eta_k^{[\ell+1]}\left(\beta\frac{m_{k,new}^{[\ell+1]}}{\eta_{k,new}^{[\ell+1]}} + (1-\beta)\frac{m_k^{[\ell]}}{\eta_k^{[\ell]}}\right). \tag{23}$$

if $\eta_k^{[\ell+1]} < 0$ **then**

$$\eta_k^{[\ell+1]} = \eta_k^{[\ell]}, \quad m_k^{[\ell+1]} = m_k^{[\ell]}. \tag{24}$$

end if

Output: $\eta_k^{[\ell+1]}, m_k^{[\ell+1]}$

B. On the election of EP parameters

The moment matching condition explained in (15) determines the optimal operation point found by the EP approximation. By repeating this procedure, we allow to find a stationary solution for the operation point. In order to avoid instabilities and control the accuracy and speed of convergence, some EP parameters are introduced. These parameters are the number of EP iterations (S), a minimum allowed variance (ϵ) and a damping factor (β). Based on recent studies, these EP parameters can be further optimized [33]–[35]. Following the guidelines in those papers and after extensive experimentation, in the general case we found out that instabilities can be controlled by setting[3] $\epsilon = 1e^{-8}$. Regarding the accuracy of the algorithm, it is convenient to start with a conservative value of the damping parameter β in Algorithm 2. The value $\beta = 0.1$ forces our algorithm to move slowly towards the EP solution. Once the turbo procedure starts, we let the damping parameter grow in order to speed up the achievement of the EP solution, reducing the value of S from 10 in [21] to 3. A simple rule for determination of β that fulfills this requirements and leads to good performance is an exponential growth with a saturation value of 0.7, i.e., $\beta = \min(\exp^{t/1.5}/10), 0.7)$, where $t \in [0, T]$ is the number of the current turbo iteration. With this criterion the number of EP iterations after the turbo procedure starts is reduced to $S = 3$, hence reducing the computational complexity by more than a third.

[3]Parameters have been chosen to optimize turbo equalization [35].

Algorithm 2 nuBEP Turbo Equalizer

Initialization: Set $p_D(u_k) = \frac{1}{\mathtt{M}} \sum_{u \in \mathcal{A}} \delta(u_k - u)$ for $k = 1, \ldots, \mathtt{N}$
for $t = 1, ..., T$ **do**
1) Compute the mean $m_k^{[1]}$ and variance $\eta_k^{[1]}$ given by (13) and (14), respectively.
for $\ell = 1, ..., S$ **do**
for $k = 1, ..., \mathtt{N}$ **do**
2) Compute the k-th extrinsic distribution as in (16), i.e.,

$$q_E^{[\ell]}(u_k) = \mathcal{CN}\Big(u_k : z_k^{[\ell]}, v_k^{2[\ell]}\Big) \tag{25}$$

where $z_k^{[\ell]}$ and $v_k^{2[\ell]}$ are given by (17) and (18), respectively.
3) Obtain the distribution $\widehat{p}^{[\ell]}(u_k) \propto q_E^{[\ell]}(u_k) p_D(u_k)$ and estimate its mean $\mu_{p_k}^{[\ell]}$ and variance $\sigma_{p_k}^{2[\ell]}$. Set a minimum allowed variance as $\sigma_{p_k}^{2[\ell]} = \max(\epsilon, \sigma_{p_k}^{2[\ell]})$.
4) Run the moment matching and damping procedures by executing Algorithm 1.
end for
end for
5) With the values $m_k^{[S+1]}, \eta_k^{[S+1]}$ obtained after the EP algorithm, calculate the extrinsic distribution $q_E(u_k)$.
6) Demap the extrinsic distribution and compute the extrinsic LLR, $L_E(c_{k,j})$, by means of (4).
7) Run the channel decoder to output $p_D(u_k)$
end for
Output: Deliver $L_E(c_{k,j})$ to the channel decoder for $k = 1, \ldots, \mathtt{N}$ and $j = 1, \ldots, Q$

IV. Filter-type turbo equalization

A. LMMSE filter

In this subsection we review the formulation of the LMMSE-based filter [5], [13], [14], modified to allow for unnormalized transmitted energy and a different computation of the extrinsic distribution. The LMMSE-based filter [5], [13], [14] estimates one symbol per k-th iteration, u_k, given a W-size window of observations, $\mathbf{y_k} = [y_{k-\mathtt{W}_2}, ..., y_{k+\mathtt{W}_1}]^\top$, where $\mathtt{W} = \mathtt{W}_1 + \mathtt{W}_2 + 1$. This procedure differs from [36], where each transmitted symbol is estimated given the whole vector of observations, $\mathbf{y}$. The LMMSE equalizer approximates the prior for each symbol, $p_D(u_k)$, as a Gaussian

$$p_D(u_k) \approx \tilde{p}_D(u_k) = \mathcal{CN}(u_k : m_k, \eta_k)\,, \tag{26}$$

where the mean, m_k, and variance, v_k, are *a priori* statistics for each transmitted symbol, given by (13) and (14), respectively. For the first iteration of the turbo equalization no *a priori* information is available and a suitable initialization is $m_k = 0$, $\eta_k = E_s$, which boils down to $m_k = 0$, $\eta_k = 1$ when normalizing the energy [5], [13], [14]. Given the current prior and the channel impulse response (CIR), the LMMSE filter computes a Gaussian approximation of the *posterior* probability of each symbol. When a turbo scheme is used, the equalizer and decoder exchange *extrinsic* information [6]. Through the turbo equalization iterations, the *a priori* statistics in (26) are updated with the information fed back from the channel decoder.

Rather than computing the posterior distribution as in (7), the LMMSE filter [5] considers the a posteriori probabilities with respect to the estimated transmitted symbol, $\widehat{u}_k$. For this reason, and to keep the same notation than in [5], we will denote the approximated posterior as $q(u_k|\widehat{u}_k)$. With this posterior distribution in mind, the extrinsic probability at the output of the LMMSE filter can be computed as

$$q_E(u_k|\widehat{u}_k) = \frac{q(u_k|\widehat{u}_k)}{\tilde{p}_D(u_k)}. \tag{27}$$

This distribution is Gaussian and can be derived from the extrinsic distribution of the estimated symbol computed in [5], as shown in Appendix B, yielding

$$q_E(u_k|\widehat{u}_k) = \mathcal{CN}\big(u_k : z_k, v_k^2\big) \tag{28}$$

where

$$z_k = \frac{\mathbf{c_k}^{\mathrm{H}}(\mathbf{y_k} - \mathbf{H_W}\mathbf{m_k} + m_k \mathbf{h_W})}{\mathbf{c_k}^{\mathrm{H}}\mathbf{h_W}}, \tag{29}$$

$$v_k^2 = \frac{\mathbf{c_k}^{\mathrm{H}}\mathbf{h_W} E_s (1 - \mathbf{h_W}^{\mathrm{H}}\mathbf{c_k})}{(\mathbf{c_k}^{\mathrm{H}}\mathbf{h_W})^2}, \tag{30}$$

and, in turn,

$$\mathbf{c_k} = \Big(\mathbf{\Sigma_k} + (E_s - \eta_k)\mathbf{h_W}\mathbf{h_W}^{\mathrm{H}}\Big)^{-1} E_s \mathbf{h_W}, \tag{31}$$

$$\mathbf{H_W} = \begin{bmatrix} h_{\mathtt{L}} & \ldots & h_1 & & \mathbf{0} \\ & \ddots & & \ddots & \\ & & \ddots & & \ddots \\ \mathbf{0} & & & h_{\mathtt{L}} & \ldots \; h_1 \end{bmatrix} \tag{32}$$

is the $\mathtt{W} \times (\mathtt{W} + \mathtt{L} - 1)$ channel matrix, $\mathbf{h_W}$ is the $(\mathtt{W}_2 + \mathtt{L})$-th column of $\mathbf{H_W}$ and

$$\mathbf{m_k} = [m_{k-\mathtt{L}-\mathtt{W}_2+1}, ..., m_{k+\mathtt{W}_1}]^\top, \tag{33}$$

$$\mathbf{V_k} = \mathrm{diag}(\eta_{k-\mathtt{L}-\mathtt{W}_2+1}, ..., \eta_{k+\mathtt{W}_1}), \tag{34}$$

$$\mathbf{\Sigma_k} = \sigma_w^2 \mathbf{I} + \mathbf{H_W}\mathbf{V_k}\mathbf{H_W}^{\mathrm{H}}. \tag{35}$$

The computational complexity is dominated by (31), which has to be recomputed every k-th iteration. Hence, the complexity is $\mathcal{O}(\mathtt{NW}^2)$. This complexity can be further reduced by relying on some approximations proposed in [5], [14].

B. EP filter (EP-F)

A novel EP filter-type is developed in this subsection to improve the accuracy and performance of the LMMSE-based filter explained above. As explained in Subsection IV-A, if the LMMSE filter is run, the prior of each symbol is approximated by a Gaussian with the statistics given by the decoder, i.e., with mean and variance given by (13) and (14), respectively. By using the EP algorithm we approximate the posterior distribution with a Gaussian family. Since the posterior distribution includes the true discrete priors, we take into account the discrete nature of symbols.

At every iteration of the EP algorithm, ℓ, we approximate the product of priors of individual symbols in (7) as a product

Algorithm 3 EP-F

Initialization: Set $p_D(u_k) = \frac{1}{\mathtt{M}} \sum_{u \in \mathcal{A}} \delta(u_k - u)$ for $k = 1, \ldots, \mathtt{N}$

for $t = 1, ..., T$ **do**

1) Compute the mean $m_k^{[1]}$ and variance $\eta_k^{[1]}$ given by (13) and (14), respectively.

for $\ell = 1, ..., S$ **do**

for $k = 1, ..., \mathtt{N}$ **do**

2) Compute the k-th extrinsic distribution as in (28), i.e.,

$$q_E^{[\ell]}(u_k|\widehat{u}_k) = \mathcal{CN}\Big(u_k : z_k^{[\ell]}, v_k^{2[\ell]}\Big) \tag{37}$$

where $z_k^{[\ell]}$ and $v_k^{2[\ell]}$ are given by (29) and (30), respectively.

3) Obtain the distribution $\widehat{p}^{[\ell]}(u_k) \propto q_E^{[\ell]}(u_k|\widehat{u}_k) p_D(u_k)$ and estimate its mean $\mu_{p_k}^{[\ell]}$ and variance $\sigma_{p_k}^{2[\ell]}$. Set a minimum allowed variance as $\sigma_{p_k}^{2[\ell]} = \max(\epsilon, \sigma_{p_k}^{2[\ell]})$.

4) Run the moment matching and damping procedures by executing Algorithm 1.

end for

end for

5) With the values $m_k^{[S+1]}, \eta_k^{[S+1]}$ obtained after the EP algorithm, calculate the extrinsic distribution $q_E(u_k|\widehat{u}_k)$ in (28).

6) Demap the extrinsic distribution and compute the extrinsic LLR, $L_E(c_{k,j})$, by means of (4).

7) Run the channel decoder to output $p_D(u_k)$

end for

Output: Deliver $L_E(c_{k,j})$ to the channel decoder for $k = 1, \ldots, \mathtt{N}$ and $j = 1, \ldots, Q$

of $\mathtt{N}$ Gaussians, $\tilde{p}_D^{[\ell]}(u_k) = \mathcal{CN}\Big(u_k : m_k^{[\ell]}, \eta_k^{[\ell]}\Big)$, whose parameters (means and variances) are adjusted to find a better approximation, $q^{[\ell]}(\mathbf{u}) \propto p(\mathbf{y}|\mathbf{u}) \prod_{k=1}^{\mathtt{N}} \tilde{p}_D^{[\ell]}(u_k)$, to the true posterior. Similarly to (27)-(28), for each k-th symbol, we first compute the current extrinsic distribution,

$$q_E^{[\ell]}(u_k|\widehat{u}_k) = \frac{q^{[\ell]}(u_k|\widehat{u}_k)}{\tilde{p}_D^{[\ell]}(u_k)}. \tag{36}$$

Now, a more accurate posterior distribution can be obtained by finding a new Gaussian approximation, $\tilde{p}_D^{[\ell+1]}(u_k)$, to match the moments of $q_E^{[\ell]}(u_k|\widehat{u}_k)\tilde{p}_D^{[\ell+1]}(u_k)$ and $q_E^{[\ell]}(u_k|\widehat{u}_k)p_D(u_k)$, as in (15). With these new values for the mean, $m_k^{[\ell+1]}$, and variance, $\eta_k^{[\ell+1]}$, we can recompute a new extrinsic distribution $q_E^{[\ell+1]}(u_k|\widehat{u}_k)$, which is more accurate than the one in (28). The final extrinsic distribution delivered to the decoder is the one obtained after the last iteration of the EP algorithm, following (36).

We denote this new algorithm as EP-filter (EP-F). Algorithm 3 is a detailed description of its implementation. Note that the main difference between Algorithm 2 and Algorithm 3 lies in the computation of the extrinsic distribution, i.e., equations (25) and (37). The computational complexity is also dominated by (31), which has to be computed for each symbol and each ℓ-th iteration. Hence, the complexity is S times the LMMSE complexity, i.e. $\mathcal{O}(S\mathtt{NW}^2)$, where S is the number of iterations of the EP-F. At this point, it is interesting to remark that the approximations proposed in [5], [14] to further reduce the complexity cannot be applied when the EP is used. The reason is that these approximations remove (at some points) the prior variance computed by the decoder, setting it to one.

V. Relation to previous approaches

A. Update of the priors

We improve the prior information used in the equalizer once the turbo procedure starts, forcing the true discrete prior to be non-uniform in contrast to the uniform priors used by previous EP-based approaches.

In previous proposals [20]–[22], the probabilities from the channel decoder, $p_D(u_k)$, were used to initialize, at the beginning of every iteration of the turbo-equalization, the product of Gaussians that in the EP approximation replaces the product of priors, $\tilde{p}_D^{[1]}(u_k)$. But when the moment matching was performed in the EP algorithm, i.e.,

$$q_E^{[\ell]}(u_k)\mathbb{I}_{u_k \in \mathcal{A}} \overset{\text{moment matching}}{\longleftrightarrow} q_E^{[\ell]}(u_k)\tilde{p}_D^{[\ell+1]}(u_k), \tag{38}$$

the true priors used were uniformly distributed following

$$\mathbb{I}_{u_k \in \mathcal{A}} = \frac{1}{\mathtt{M}} \sum_{u \in \mathcal{A}} \delta(u_k - u). \tag{39}$$

In the current proposal, we keep the initialization of the Gaussians in every step of the turbo-equalization, $\tilde{p}_D^{[1]}(u_k)$, but also *propose to replace the uniform priors in* (39) *by non-uniform ones* in the moment matching step, as explained in (15), i.e.,

$$q_E^{[\ell]}(u_k)p_D(u_k) \overset{\text{moment matching}}{\longleftrightarrow} q_E^{[\ell]}(u_k)\tilde{p}_D^{[\ell+1]}(u_k), \tag{40}$$

where

$$p_D(u_k) = \sum_{u \in \mathcal{A}} \delta(u_k - u) \prod_{j=1}^{Q} p_D(c_{k,j} = \varphi_j(u)), \tag{41}$$

Note that the different definition of priors -(39) in previous proposals, (41) in this manuscript- is the difference between the currently proposed nuBEP algorithm and the BEP in [21], with remarkable improvements.

B. Parameter Optimization

The computational complexity of the EP algorithm is reduced to roughly a third part of that in [21], by optimizing the choice of EP parameters. In particular, we propose some new values for ϵ and β, that control numerical instabilities in the EP updates, and S, the number of iterations of the EP equalizer. The parameters proposed in this paper reduce the number of iterations in turbo equalization to $S = 3$, rather than the $S = 10$ iterations that were used in [21].

C. Filter-type solution

The new filter-type EP solution proposed is constrained to have linear complexity in the frame length and quadratic in the filter length, i.e., it is endowed with the same complexity order than the LMMSE filter. This complexity is not quadratic with the block length as the one of the BEP [21] nor cubic with the window length as complexity of the SEP [22].

D. Equalization solved with EP

Regarding the EP-based equalizers proposed by other authors, the approach in [18], [19] should be mentioned. These proposals deal just with how to pass information between the channel decoder and the LMMSE equalizer. Our proposal first focuses on the EP based equalization, performed independently of the turbo iterations. Therefore the approaches are quite different. Issues such as how to use the priors in the moment matching within the EP equalizer or the damping do not arise in these proposals where the improvement is related only to the handling of probabilities between blocks.

VI. Simulation results

In this section, we compare the performance of both the block LMMSE and EP-F equalizers for different scenarios. We also include the performance of the BEP [21] and the AWGN bound as references. Note that the MMSE filter [5] has not been included in the simulations since the block LMMSE exhibits equal or better performance than any filtering approaches based on the LMMSE algorithm. We did not include the SEP algorithm since it exhibits the same performance as the block implementation, as shown in [22]. We also include the nuBEP approach to illustrate the quite improved behavior when using non-uniform priors at each EP iteration, even reducing from 10 to 3 the number of iterations of the EP approach. The EP parameters have been selected as explained in Subsection III-B, both for the nuBEP and EP-F methods. For a full performance comparison with BCJR approximations, such as M-BCJR [8], M*-BCJR [10], RS-BCJR [9], NZ and NZ-OS [11], please see [21]. In Table I we include a detailed comparison of the complexity of all the simulated algorithms. Above we include the computational complexity of previous algorithms in [21] (BEP) and [22] (SEP), the block and filter implementation of the LMMSE and BCJR approaches. Below we provide the complexity for the new approaches in this paper, i.e., the proposed nuBEP and EP-F. Parameter $\mathtt{W}$ is typically around two times the length of the channel, $\mathtt{L}$. Here, we simulate the scenarios in [13], [14], using the same channel responses and modulations. Other modulations are also considered. The absolute value of LLRs given to the decoder is limited to 5 in order to avoid very confident probabilities. We use a (3,6)-regular LDPC of rate 1/2, and belief propagation as decoder with a maximum of 100 iterations. The window length in the filtered approach is set to $\mathtt{W} = \mathtt{W}_1 + \mathtt{W}_2 + 1$, where $\mathtt{W}_1 = 2\mathtt{L}$ and $\mathtt{W}_2 = \mathtt{L} + 1$ as suggested in [14].

In the following, we first include a section to analyze the performance of our approach in a low complexity scenario with BPSK modulation, similarly to [14]. The optimal BCJR algorithm can be run in this scenario with a low enough computational complexity and is used as bound. Next, we include a section to analyze the behavior of the algorithms in a large complexity scenario, where we use high-order modulations such as 8-PSK, 16-QAM and 64-QAM.

TABLE I: Complexity comparison between algorithms.

Algorithm	Complexity per turbo iteration
BCJR	$\mathtt{NM}^{\mathtt{L}}$
BEP	$10\mathtt{LN}^2$
block-LMMSE	$\mathtt{LN}^2$
SEP	$10\mathtt{NW}^3$
LMMSE filter	$\mathtt{NW}^2$
nuBEP	$3\mathtt{LN}^2$
EP-F	$3\mathtt{NW}^2$

A. BPSK scenario

In Fig. 2 we include the BER, averaged over 10^4 random frames, for the LMMSE, BEP [21], nuBEP, EP-F and BCJR equalizers with a BPSK modulation and two different channel responses and lengths of encoded words: $\mathbf{h} = [0.227\ 0.46\ 0.688\ 0.46\ 0.227]^\top$ and $V = 4096$ bits in Fig. 2 (a)-(c) and $h = [0.407\ 0.815\ 0.407]^\top$ and $V = 1024$ bits in Fig. 2 (d)-(f). The channel responses were selected following the simulations in [13], [14]. The performances of block-algorithms, BEP and nuBEP, are very similar to the equivalent forward filtering approach. When the nuBEP algorithm is applied, 2 and 1.5 dBs gains are obtained compared to LMMSE approach in the turbo scenario, for the two simulated scenarios, respectively. The EP-F exhibits a performance similar to that of the nuBEP.

In Fig. 3 we include the EXIT charts of the BEP [21], nuBEP, EP-F, LMMSE and BCJR for the channel response $\mathbf{h} = [0.227\ 0.46\ 0.688\ 0.46\ 0.227]^\top$ as in [5], [13], BPSK modulation with $E_b/N_0 = 9$ (solid) and 7 dB (dashed). The EXIT chart of the LDPC encoder of $2048/4096$ and $R = 1/2$ used is also depicted (solid). The horizontal and vertical axis depict the mutual information at the input, I_i, and the output, I_o, respectively. We use arrows to show the evolution of the mutual information along the turbo iterations for $E_b/N_o = 9$ dB. Vertical (horizontal) arrows indicate the improvement in the mutual information each time the equalizer (channel decoder) is executed. When no *a priori* information is given to the decoder, i.e., $I_i = 0$, both BEP and EP-F provide a higher value for the mutual information at the output, I_o, than the LMMSE approach, i.e., they start from a more accurate estimation even before the turbo equalization. This greatly improves the performance as it enlarges the gap between the equalizer and the channel decoder EXIT curves. It can be seen that the LMMSE approach will fail when $E_b/N_o = 7$ dB, because both curves intersect.

Note that the wide EXIT tunnel from the equalizer to the LDPC decoder is suggesting that the code is not optimum in terms of capacity [37]. An optimal code in this sense would

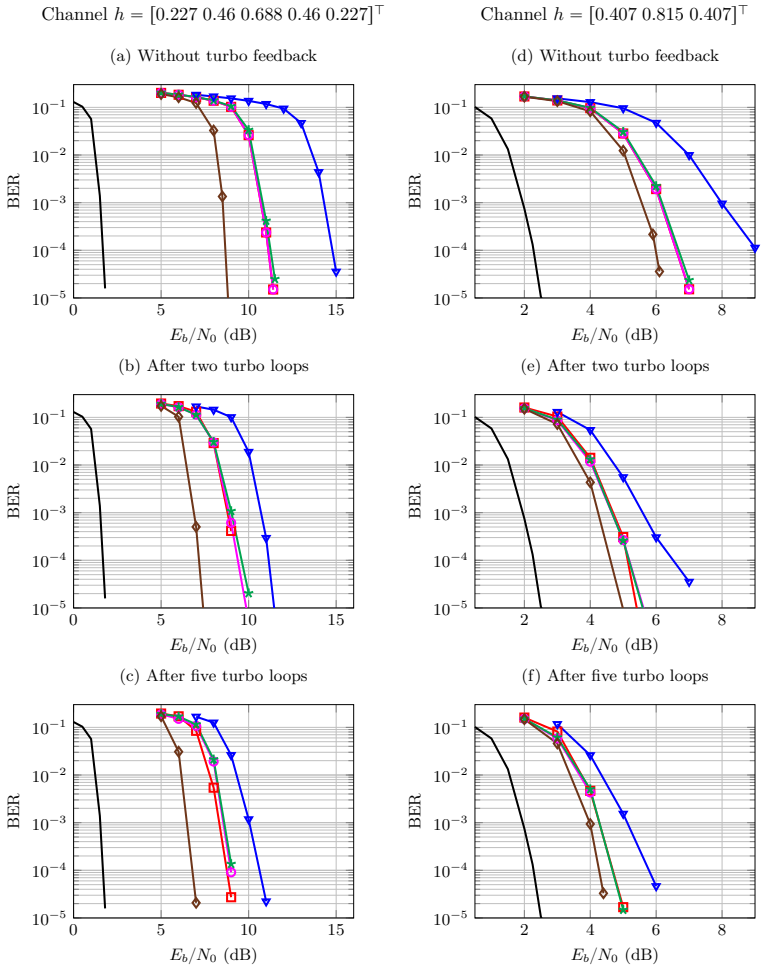


Fig. 2: BER along E_b/N_0 for BEP [21] (▫), nuBEP (○), EP-F (∗), block-LMMSE (▽) and BCJR (◇) turbo equalizers, BPSK, codewords of $V = 4096$ bits (a)-(c) and $V = 1024$ bits (d)-(f) and two different channel responses. Black lines represent the AWGN bound.

exhibit an EXIT chart near below the one of the equalizer and above the curve of the $1/2$ rate LDPC code used. The design of this code for the channel equalization response is out of the scope of this paper and remains as a future line of research.

B. Large complexity scenario

In Fig. 4 we simulate the same scenario of Fig. 2, but using an 8-PSK modulation rather than a BPSK. It can be observed that after increasing the order of the modulation, the EP-F approach presented in this work performs identically as its block counterpart, greatly improving the performance of the LMMSE algorithm before and after the turbo procedure. An improvement of the BER of the EP-F with respect to the one of the BEP approach in [21], after turbo equalization, can also be observed.

In Fig. 5 we depict the BER performance after five turbo loops for channels $\mathbf{h} = [0.227\ \ 0.46\ \ 0.688\ \ 0.46\ \ 0.227]^\top$ in (a) and $\mathbf{h} = [0.407\ \ 0.815\ \ 0.407]^\top$ in (b) with different modulations. We use solid lines to represent a 64-QAM constellation and dashed lines for a 16-QAM. We sent codewords of length $V = 4096$ in both scenarios. It can be observed that the performance of the EP-F matches with the one of its block implementation proposed in this paper (nuBEP) when a 16-QAM is used. However, the EP-F approach slightly degrades with a 64-QAM, where the block nuBEP gets the most accurate performance. Note that the behavior of the EP-F could be improved by increasing the length of the filter,

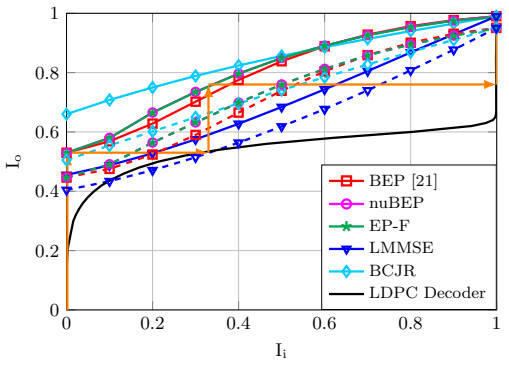


Fig. 3: EXIT charts for the decoder, the BEP [21], nuBEP, EP-F, LMMSE and BCJR equalizers with 7 (dashed) and 9 (solid) dB of E_b/N_0, BPSK modulation and codewords of $V = 4096$.

yielding the performance of its block implementation. We have a remarkable improvement of 3-5 dB with respect to the BEP in [21] and of 7-13 dB compared to the LMMSE algorithm.

For the sake of completeness, we include Fig. 6 to show how the BER changes along the turbo iterations and different block lengths at $E_b/N_0 = 13$ dB for an 8-PSK and $\mathbf{h} = [0.227\ 0.46\ 0.688\ 0.46\ 0.227]^\top$. The nuBEP algorithm is represented in (a) and the filter approach EP-F in (b). It can be observed that BER is higher for shorter codes and it improves when the code length is increased, as expected. Also, the BER does not significantly improve after the fifth turbo iteration. In the view of these results we stopped running our turbo equalizers after five turbo iterations in the experiments presented above.

VII. Conclusion

In a previous work, we presented a novel equalizer based on expectation propagation (EP) [21]. This solution presents quite an improved performance compared to previous approaches in the literature, both for hard, soft and turbo detection. The solution was presented as a block-wise solution and it was therefore denoted as block-EP (BEP). The major advantage of the BEP lies in the fact that its computational complexity does not grow exponentially with the constellation size and channel memory, as opposed to most equalizers, which are unfeasible for moderate values of these parameters. However, it exhibits a quadratic increase with the size of the transmitted word, V. To avoid this problem, filter-type equalizers are usually preferred [12]. For this reason, we proposed a smoothing EP (SEP) equalizer in [22]. However, the SEP has a computational complexity cubic in the channel length, L. Both BEP and SEP equalizers make use of a moderate feedback in the sense that an initial uniform discrete prior is assumed at the beginning of each execution of the EP algorithm, even after the turbo procedure has started. In this paper, we first propose a design to include the non-uniform discrete nature of the priors from the decoder in the EP algorithm, which amounts to a stronger feedback, quite outperforming the previous BEP and SEP approaches. Second, we develop a reduced-complexity approach by proposing better values of the EP parameters. The resulting algorithm has been denoted as nuBEP, and it significantly outperforms the BEP reducing the computational complexity to less than the third part. Finally, we adapt the EP block equalizer to the filter-type form, emulating the Wiener MMSE filter-type [14]. Therefore, we mimic the structure of the filter-type MMSE equalizer. The EP is used to better approximate the posteriors of a windowed version of the inputs, shifted for every new output estimate. As a result, we present a novel solution dealing with W inputs at a time and with quadratic computational complexity in W. This novel solution, the EP-F, despite the reduction in the computational complexity, exhibits a performance in terms of BER quite close to that of its block counterpart, the nuBEP. Furthermore, it remarkably improves the performance of the LMMSE turbo-equalizer, with same complexity order in terms of L and V. In the included experiments, for channels usually used as benchmarks in the literature, gains in the range 5-13 dB are reported for 8-PSK, 16-QAM and 64-QAM modulations.

One of the main benefits of this new proposal is to reduce the computational complexity, reducing it to be of quadratic order with the filter length. Other approaches, such as those solutions working on the frequency domain [3], could be investigated to achieve this goal. In this paper we face the equalization in single-input single-output channels, the application to MIMO channels with memory [27] remains unexplored.

Appendix A
Proof of (10) and (11)

In [21], the posterior distribution used for BEP is

$$q^{[\ell]}(\mathbf{u}) \sim \mathcal{CN}\Big(\mathbf{u} : \boldsymbol{\mu}^{[\ell]}, \boldsymbol{\Sigma}^{[\ell]}\Big) \tag{42}$$

where

$$\boldsymbol{\mu}^{[\ell]} = \boldsymbol{\Sigma}^{[\ell]}(\sigma_w^{-2}\mathbf{H}^{\mathrm{H}}\mathbf{y} + \mathrm{diag}(\boldsymbol{\eta}^{[\ell]})^{-1}\mathbf{m}^{[\ell]}), \tag{43}$$

$$\boldsymbol{\Sigma}^{[\ell]} = \Big(\sigma_w^{-2}\mathbf{H}^{\mathrm{H}}\mathbf{H} + \mathrm{diag}(\boldsymbol{\eta}^{[\ell]})^{-1}\Big)^{-1}. \tag{44}$$

By a direct application of the Woodbury identity, equation (44) can be rewritten as

$$\boldsymbol{\Sigma}^{[\ell]} = \mathrm{diag}(\boldsymbol{\eta}^{[\ell]}) - \mathrm{diag}(\boldsymbol{\eta}^{[\ell]})\mathbf{H}^{\mathrm{H}}\mathbf{C}^{-1}\mathbf{H}\mathrm{diag}(\boldsymbol{\eta}^{[\ell]}) \tag{45}$$

where

$$\mathbf{C} = \mathbf{H}\mathrm{diag}(\boldsymbol{\eta}^{[\ell]})\mathbf{H}^{\mathrm{H}} + \sigma_w^2\mathbf{I}. \tag{46}$$

The k-th diagonal element of (45) yields (11). Regarding (43), it can be divided into two terms

$$\boldsymbol{\mu}^{[\ell]} = \underbrace{\boldsymbol{\Sigma}^{[\ell]}\sigma_w^{-2}\mathbf{H}^{\mathrm{H}}\mathbf{y}}_{T_1} + \underbrace{\boldsymbol{\Sigma}^{[\ell]}\mathrm{diag}(\boldsymbol{\eta}^{[\ell]})^{-1}\mathbf{m}^{[\ell]}}_{T_2}. \tag{47}$$

We apply the following identity [31],

$$(\mathbf{A}^{-1}+\mathbf{B}^{\mathrm{H}}\mathbf{D}^{-1}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{H}}\mathbf{D}^{-1} = \mathbf{A}\mathbf{B}^{\mathrm{H}}(\mathbf{B}\mathbf{A}\mathbf{B}^{\mathrm{H}}+\mathbf{D})^{-1} \tag{48}$$

to the first term, T_1, in (47), yielding

$$T_1 = \mathrm{diag}(\boldsymbol{\eta}^{[\ell]})\mathbf{H}^{\mathrm{H}}\mathbf{C}^{-1}\mathbf{y}. \tag{49}$$

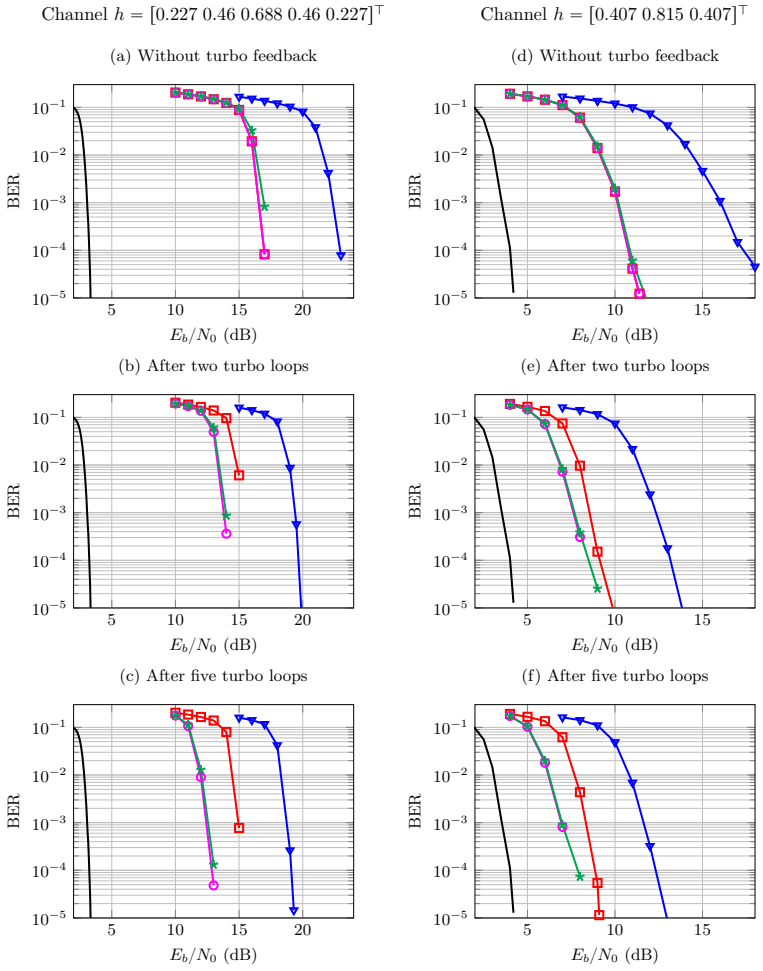


Fig. 4: BER along E_b/N_0 for BEP [21] (□), nuBEP (○), EP-F (⋆) and block-LMMSE (▽) turbo equalizers, 8-PSK, codewords of $V = 4096$ (a)-(c) and $V = 1024$ (d)-(f) and two different channel responses. Black lines represent the AWGN bound.

Now, we replace (45) into the second term, T_2, in (47), obtaining

$$T_2 = \mathbf{m}^{[\ell]} - \text{diag}(\boldsymbol{\eta}^{[\ell]})\mathbf{H}^{\text{H}}\mathbf{C}^{-1}\mathbf{H}\mathbf{m}^{[\ell]}. \tag{50}$$

By replacing (49) and (50) into (47), we finally get

$$\boldsymbol{\mu}^{[\ell]} = \mathbf{m}^{[\ell]} + \text{diag}(\boldsymbol{\eta}^{[\ell]})\mathbf{H}^{\text{H}}\mathbf{C}^{-1}(\mathbf{y} - \mathbf{H}\mathbf{m}^{[\ell]}), \tag{51}$$

whose k-th element is given by (10).

Appendix B
Proof of (29) and (30)

In [5], the extrinsic distribution of the estimated symbol is computed as

$$q_E(\widehat{u}_k|u_k) \sim \mathcal{CN}\big(\widehat{u}_k : u_k {\mathbf{c_k}}^{\text{H}}\mathbf{h}_{\text{W}}, \sigma_k^2\big) \tag{52}$$

where

$$\widehat{u}_k = {\mathbf{c_k}}^{\text{H}}(\mathbf{y_k} - \mathbf{H}_{\text{W}}\mathbf{m_k} + m_k\mathbf{h}_{\text{W}}), \tag{53}$$

$$\sigma_k^2 = {\mathbf{c_k}}^{\text{H}}\mathbf{h}_{\text{W}}E_s(1 - {\mathbf{h}_{\text{W}}}^{\text{H}}\mathbf{c_k}), \tag{54}$$

$\mathbf{c_k}$ is given by (31) and $\mathbf{h}_{\text{W}}$ is the $(\text{W}_2 + \text{L})$-th column of $\mathbf{H}_{\text{W}}$ defined in (32). Note that we generalized the expressions in [5] to consider a symbol energy of E_s. If we set $E_s = 1$, we obtain exactly the formulation in [5]. Instead of the extrinsic distribution of the estimated symbol, we use in our formulation the extrinsic distribution of the true symbol, which can be computed from (52) as

$$q_E(u_k|\widehat{u}_k) \sim \mathcal{CN}\big(u_k : z_k, v_k^2\big) \tag{55}$$

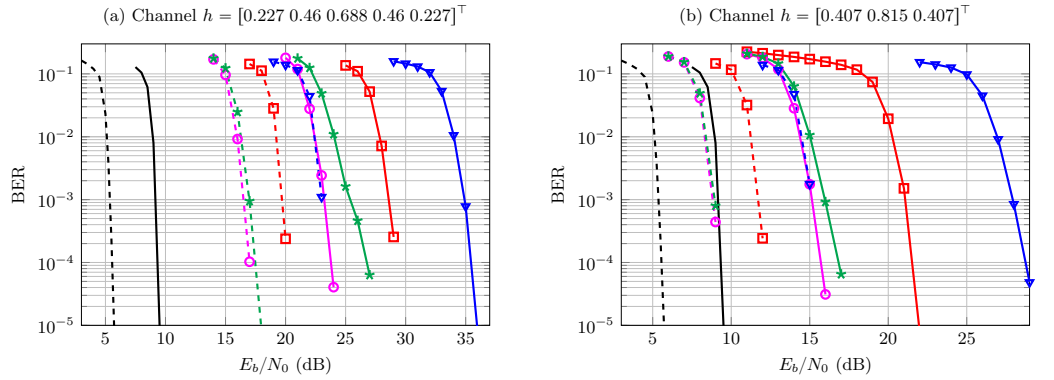


Fig. 5: BER along E_b/N_0 for BEP [21] (▫), nuBEP (○), EP-F (∗) and block-LMMSE (▽) turbo equalizers after five turbo loops, 64-QAM (solid lines) and 16-QAM (dashed lines), codewords of $V = 4096$ and two different channel responses. Black lines represent the AWGN bound.

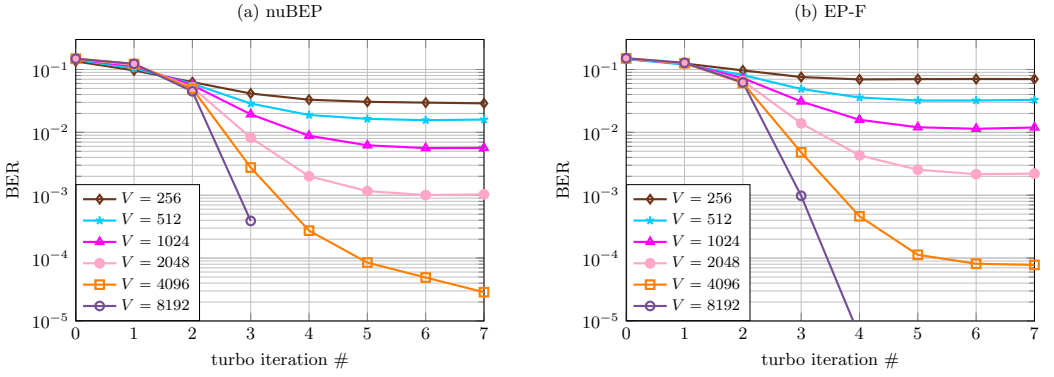


Fig. 6: BER of nuBEP (a) and EP-F (b) turbo equalizers at $E_b/N_0 = 13$ dB for several turbo iterations and lengths of encoded words. 8-PSK modulation and the channel response $\mathbf{h} = [0.227\ 0.46\ 0.688\ 0.46\ 0.227]^\top$ were used.

where

$$z_k = \frac{\widehat{u}_k}{\mathbf{c_k}^{\mathrm{H}}\mathbf{h}_{\mathrm{W}}}, \tag{56}$$

$$v_k^2 = \frac{\sigma_k^2}{(\mathbf{c_k}^{\mathrm{H}}\mathbf{h}_{\mathrm{W}})^2}, \tag{57}$$

yielding the formulation in (29) and (30).

References

[1] S. Haykin, *Communication Systems*, 5th ed. Wiley Publishing, 2009.

[2] L. Salamanca, J. J. Murillo-Fuentes, and F. Pérez-Cruz, "Bayesian equalization for LDPC channel decoding," *IEEE Trans. on Signal Processing*, vol. 60, no. 5, pp. 2672–2676, May 2012.

[3] J. Karjalainen, N. Veselinovic, K. Kansanen, and T. Matsumoto, "Iterative frequency domain joint-over-antenna detection in multiuser mimo," *IEEE Transactions on Wireless Communications*, vol. 6, no. 10, pp. 3620–3631, October 2007.

[4] C. Douillard, M. Jezequel, C. Berrou, P. Didier, and A. Picart, "Iterative correction of intersymbol interference: turbo-equalization," *European Trans. on Telecommunications*, vol. 6, no. 5, pp. 507–512, Sep 1995.

[5] M. Tüchler, R. Koetter, and A. Singer, "Turbo equalization: principles and new results," *IEEE Trans. on Communications*, vol. 50, no. 5, pp. 754–767, May 2002.

[6] R. Koetter, A. Singer, and M. Tüchler, "Turbo equalization," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 67–80, Jan 2004.

[7] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Trans. on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[8] V. Franz and J. Anderson, "Concatenated decoding with a reduced-search BCJR algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 186–195, Feb 1998.

[9] G. Colavolpe, G. Ferrari, and R. Raheli, "Reduced-state BCJR-type algorithms," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 5, pp. 848–859, May 2001.

[10] M. Sikora and D. Costello, "A new SISO algorithm with application to turbo equalization," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Sep 2005, pp. 2031–2035.

[11] D. Fertonani, A. Barbieri, and G. Colavolpe, "Reduced-complexity BCJR algorithm for turbo equalization," *IEEE Trans. on Communications*, vol. 55, no. 12, pp. 2279–2287, Dec 2007.

[12] C. Berrou, *Codes and turbo codes*, ser. Collection IRIS. Springer Paris, 2010.

[13] M. Tüchler, A. Singer, and R. Koetter, "Minimum mean squared error equalization using a priori information," *IEEE Trans. on Signal*

Processing, vol. 50, no. 3, pp. 673–683, Mar 2002.

[14] M. Tüchler and A. Singer, "Turbo equalization: An overview," *IEEE Trans. on Information Theory*, vol. 57, no. 2, pp. 920–952, Feb 2011.

[15] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Pérez-Cruz, "Improved performance of LDPC-coded MIMO systems with EP-based soft-decisions," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Jun 2014, pp. 1997–2001.

[16] P. M. Olmos, J. J. Murillo-Fuentes, and F. Pérez-Cruz, "Tree-structure expectation propagation for LDPC decoding over the BEC," *IEEE Trans. on Information Theory*, vol. 59, no. 6, pp. 3354–3377, 2013.

[17] L. Salamanca, P. M. Olmos, F. Pérez-Cruz, and J. J. Murillo-Fuentes, "Tree-structured expectation propagation for LDPC decoding over BMS channels," *IEEE Trans. on Communications*, vol. 61, no. 10, pp. 4086–4095, Oct 2013.

[18] J. Hu, H. Loeliger, J. Dauwels, and F. Kschischang, "A general computation rule for lossy summaries/messages with examples from equalization," in *Proc. 44th Allerton Conf. Communication, Control, and Computing*, Sep 2006, pp. 27–29.

[19] P. Sun, C. Zhang, Z. Wang, C. Manchon, and B. Fleury, "Iterative receiver design for ISI channels using combined belief- and expectation-propagation," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1733–1737, Oct 2015.

[20] I. Santos, J. J. Murillo-Fuentes, and P. M. Olmos, "Block expectation propagation equalization for ISI channels," in *Proc. 23rd European Signal Processing Conference (EUSIPCO)*, Sep 2015, pp. 379–383.

[21] I. Santos, J. J. Murillo-Fuentes, R. Boloix-Tortosa, E. Arias-de-Reyna, and P. M. Olmos, "Expectation propagation as turbo equalizer in ISI channels," *IEEE Trans. on Communications*, vol. 65, no. 1, pp. 360–370, Jan 2017.

[22] I. Santos, J. J. Murillo-Fuentes, E. Arias-de-Reyna, and P. M. Olmos, "Probabilistic equalization with a smoothing expectation propagation approach," *IEEE Trans. on Wireless Communications*, vol. 16, no. 5, pp. 2950–2962, May 2017.

[23] K. Kansanen and T. Matsumoto, "An analytical method for mmse mimo turbo equalizer exit chart computation," *IEEE Transactions on Wireless Communications*, vol. 6, no. 1, pp. 59–63, Jan 2007.

[24] D. Falconer, S. L. Ariyavisitakul, A. Benyamin-Seeyar, and B. Eidson, "Frequency domain equalization for single-carrier broadband wireless systems," *IEEE Communications Magazine*, vol. 40, no. 4, pp. 58–66, Apr 2002.

[25] M. Tüchler and J. Hagenauer, "Turbo equalization using frequency domain equalizers," in *Proc. of the Allerton Conference. on Communication, Control, and Computing*, Oct. 2000, pp. 1234–1243.

[26] N. Veselinovic, T. Matsumoto, and M. Juntti, *Iterative PDF Estimation-Based Multiuser Diversity Detection and Channel Estimation with Unknown Interference*. Hindawi Publishing Corporation: EURASIP Journal on Advances in Signal Processing, 2005. [Online]. Available: https://doi.org/10.1155/ASP.2005.872

[27] L. Xiao, Y. Xiao, Y. Zhao, P. Yang, M. D. Renzo, S. Li, and W. Xiang, "Time-domain turbo equalization for single-carrier generalized spatial modulation," *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 5702–5716, Sept 2017.

[28] T. P. Minka, "A family of algorithms for approximate Bayesian inference," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[29] T. Minka, "Expectation propagation for approximate Bayesian inference," in *Proc. 17th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2001, pp. 362–369.

[30] M. Seeger, "Expectation propagation for exponential families," *Univ. Calif., Berkeley, CA, USA, Tech. Rep.*, 2005.

[31] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag, New York, 2006.

[32] K. Takeuchi, "Rigorous dynamics of expectation-propagation-based signal recovery from unitarily invariant measurements," in *IEEE International Symposium on Information Theory (ISIT)*, June 2017, pp. 501–505.

[33] M. Opper and O. Winther, "Expectation consistent approximate inference," *Journal of Machine Learning Research*, vol. 6, pp. 2177–2204, Dec 2005.

[34] J. Céspedes, "Approximate inference in massive mimo scenarios with moment matching techniques," Ph.D. dissertation, Universidad Carlos III de Madrid, Jan 2017.

[35] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Pérez-Cruz, "Probabilistic MIMO symbol detection with expectation consistency approximate inference," *IEEE Trans. on Vehicular Technology*, Sep 2017. Accepted.

[36] J. Wu and Y. Zheng, "Low complexity soft-input soft-output block decision feedback equalization," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, Nov 2007, pp. 3379–3383.

[37] Q. Xie, Z. Yang, J. Song, and L. Hanzo, "EXIT-chart-matching-aided near-capacity coded modulation design and a BICM-ID design example for both gaussian and rayleigh channels," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 3, pp. 1216–1227, March 2013.

Appendix C
Paper III

Authors: Irene Santos, Juan José Murillo-Fuentes, Eva Arias de Reyna and Pablo M. Olmos.
Title: Probabilistic Equalization With a Smoothing Expectation Propagation Approach.
Journal: IEEE Transactions on Wireless Communications.
Volume: 16.
Number: 5.
Pages: 2950-2962.
Date: May 2017.
Abstract: In this paper, we face the soft equalization of channels with inter-symbol interference for large constellation sizes, `M`. In this scenario, the optimal BCJR solution and most of their approximations are intractable, as the number of states they track grows fast with `M`. We present a probabilistic equalizer to approximate the posterior distributions of the transmitted symbols using the expectation propagation (EP) algorithm. The solution is presented as a recursive sliding window approach to ensure that the computational complexity is linear with the length of the frame. The estimations can be further improved with a forward–backward approach. This novel soft equalizer, denoted as smoothing EP (SEP), is also tested as a turbo equalizer, with a low-density parity-check (LDPC) channel decoder. The extensive results reported reveal remarkably good behavior of the SEP. In low dimensional cases, the bit error rate (BER) curves after decoding are closer than 1 dB from those of the BJCR, robust to the channel response. For large `M`, the SEP exhibits gains in the range of 3-5 dB compared to the linear minimum mean square error algorithm.

Probabilistic Equalization with a Smoothing Expectation Propagation Approach

Irene Santos, Juan José Murillo-Fuentes, Eva Arias-de-Reyna, and Pablo M. Olmos

***Abstract*—In this paper we face the soft equalization of channels with inter-symbol interference for large constellation sizes, $\mathtt{M}$. In this scenario the optimal BCJR solution and most of their approximations are intractable, as the number of states they track grows fast with $\mathtt{M}$. We present a probabilistic equalizer to approximate the posterior distributions of the transmitted symbols using the expectation propagation (EP) algorithm. The solution is presented as a recursive sliding window approach to ensure that the computational complexity is linear with the length of the frame. The estimations can be further improved with a forward-backward approach. This novel soft equalizer, denoted as smoothing expectation propagation (SEP), is also tested as turbo equalizer, with a low-density parity-check (LDPC) channel decoder. The extensive results reported reveal remarkably good behavior of the SEP. In low dimensional cases, the bit error rate (BER) curves after decoding are closer than 1 dB from those of the BJCR, robust to the channel response. For large $\mathtt{M}$, the SEP exhibits gains in the range 3-5 dB compared to the linear minimum mean square error algorithm.**

***Index Terms*—Expectation propagation (EP), BCJR, low-complexity, turbo channel equalization, ISI.**

I. Introduction

EQUALIZATION provides an estimation for the transmitted symbols [1], by canceling out the inter-symbol interference (ISI) [2], [3]. The probability of each possible transmitted symbol can be also supplied, from which nowadays channel decoders highly benefit [4]. This soft information can be further refined by using turbo equalization, i.e., by feeding back the output of the channel decoder to the probabilistic equalizer [5]–[7]. If the channel decoder works iteratively, the feedback is performed after one or more iterations.

The BCJR algorithm [8] can be used as *symbol* maximum a posteriori (MAP) soft equalizer in schemes where a posteriori probabilities (APP) are suitable as inputs to the channel decoder. The BCJR assumes perfect channel state information (CSI) to work on a trellis representation [4], [9]. It exploits a doubly terminated frame where the first and last symbols are known. The BCJR is based on the definition of states and the computation of their probabilities from previous estimations and new observations through the trellis, in a forward *filtering* step. A similar procedure is performed backwards. Finally, the estimation at a given time is computed from both filtering results, as a *smoothing* approach. The BCJR has a complexity $\mathcal{O}(\mathtt{N}\mathtt{M}^{\mathtt{L}})$, i.e., linear with the frame length, $\mathtt{N}$, and proportional to the number of states, $\mathtt{M}^{\mathtt{L}}$. The BCJR memory requirements per step also grow exponentially, because it stores $\mathtt{M}^{\mathtt{L}-1}$ variables and the $\mathtt{M}$ possible transitions from each variable to the next one.

The complexity of the optimal BCJR solution becomes intractable in high dimensional scenarios, in particular when transmitting multilevel constellations. To reduce its complexity, approximate algorithms in the literature perform computations over a simplified trellis [10]–[14]. They usually follow one out of the next approaches. A first method consists in truncating the length of the CSI. The key idea is to cancel the last channel taps, retaining $\mathtt{L}'$, and reducing the number of states to $M_e = \mathtt{M}^{\mathtt{L}'}$. Then, a reduced-state sequence detection (RSSD) is performed [15]–[17]. The reduced-state BCJR (RS-BCJR) algorithm [10] belongs to this group. On the other hand, other algorithms just keep the most important sequence of states for which the APP does not differ significantly from that of the complete state sequence. Note that they perform a reduced search on the original full trellis, while the previous approach does a full search on a reduced-state trellis. The M-BCJR and the T-BCJR algorithms [11] are representative examples of this group. The M-BCJR uses a subset of fixed M_e states (given by the M_e strongest metrics) while the T-algorithm uses a variable number of states at different moments.

However, when implementing smoothing solutions, the approaches above present the following problem: they usually give a predominant role to the forward recursion, so that the subset of paths used in the backward stage is conditioned to the paths previously selected. For this reason, they do not usually succeed in equalizing maximum-phase channels [12]. A variation of the M-BCJR algorithm is proposed in [13], where the authors use a different active state selection criterion. Instead of selecting the M_e states with the largest forward metric, they keep the M_e states with the largest values of the marginal posterior probabilities of the state variables. Some posterior approaches try to join both trends to enhance the results, as the M*-BCJR proposed in [14]. This algorithm improves the performance of both RS-BCJR and M-BCJR but still fails with maximum-phase channels. In [12] the authors propose an algorithm that resorts either to the forward or to

I. Santos, J.J. Murillo-Fuentes and E. Arias-de-Reyna are with the Dept. Teoría de la Señal y Comunicaciones, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Camino de los Descubrimiento s/n, 41092 Sevilla, Spain. E-mail: {irenesantos,murillo,earias}@us.es.

P. M. Olmos is with the Dept. Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid, Avda. de la Universidad 30, 28911, Leganés (Madrid), Spain. E-mail: olmos@tsc.uc3m.es.

The final version of the manuscript can be found at https://ieeexplore.ieee.org/document/7880668/. This work was partially funded by Spanish government (Ministerio de Economía y Competitividad TEC2016-78434-C3-{2-3}-R and Juan de la Cierva Grant No. IJCI-2014-19150), by the European Union (FEDER and Marie Curie Initial Training Network "Machine Learning for Personalized Medicine" MLPM2012, Grant No. 316861), and by Comunidad de Madrid in Spain (project 'CASI-CAM-CM', id. S2013/ICE-2845).

the backward approach, depending on whether the channel is minimum or maximum phase, respectively. In the case of mixed-phase channels the authors propose an optimization stage where subsets of paths in the trellises from both approaches are involved into a smoothing approach. We will denote all these methods as nonzero (NZ) completion. The NZ output saturation (NZ-OS) method is also introduced in [12] to mitigate the overestimation induced by only considering a small subset of paths in the trellis.

All these approximations to BCJR have two major drawbacks. Firstly, they are usually designed and tested for just some specific channels. In particular, the parameters of the methods in [12] are tuned according to the CSI. But, in general, no information on how to select the parameters is given. Secondly, these solutions are intractable for large trellises. An open question in these approaches is the number of paths to track, M_e. In the experiments we include in this paper, the performance of these approaches degrades rapidly as the size of the constellation increases, i.e., if we do not keep the ratio $\mathtt{M}^{\mathtt{L}-1}/M_e$ constant. Hence, these approximated solutions are interesting for a moderate number of states. For large number of states, it is preferable to envisage filter-based equalizers of the minimum-mean-square-error (MMSE) type [6].

The soft linear minimum-mean-square-error (LMMSE) equalizer [7], [18]–[20], is a low-cost alternative for multilevel constellations. If no information on the priors is available we may assume equiprobable a priori probabilities for the symbols in the constellation. In this scenario, and assuming zero mean constellations, LMMSE estimation is based on assuming zero mean prior for every symbol, with variance equal to its energy. However, the LMMSE solution is far from being an accurate approximation to the posterior distribution of the symbols. Turbo schemes can somehow overcome this drawback, as the choice of the prior statistics is iteratively improved according to the channel decoder output. Forward filtering approaches developed for the LMMSE equalizer process a $\mathtt{W}$-length sliding window of observations, rather than the complete sequence [7], [19]–[21]. A modified recursive procedure is developed in [22], where the sliding window is replaced with an extending one. These methods are filtering approaches that do not use all future observations. Their complexity is $\mathcal{O}(\mathtt{N}\mathtt{W}^2)$, where $\mathtt{N}$ is the frame length, and it is independent of the constellation order. The complexity can be further reduced by relying on some approximations [20].

Since the LMMSE equalizer uses only the prior statistics, there is available information that this equalizer is wasting: the fact that the symbols are discrete variables. In this paper we propose to force the soft equalizer to include this restriction, before handing the estimation to the decoder. This new soft equalization paradigm is implemented by applying the expectation propagation (EP) algorithm [23], [24]. The EP, that returns an approximation for the posterior probabilities in the form of an exponential distribution, has been already successfully applied to multiple-input multiple-output (MIMO) detection [25], [26], low-density parity-check (LDPC) channel decoding [27], [28] and tracking of flat-fading channels [29]. In [30] the EP was applied to approximate messages in iterative receivers for a binary transmission. In [31], we tested the EP algorithm in [25], [26] to develop an equalizer that obtained a Gaussian approximation to the probability of the transmitted symbols conditioned to the received signal. This approach, applied to soft equalization, is developed as a block or batch method, and although it achieves a good performance, it requires the inversion of a $\mathtt{N}$-dimensional covariance matrix. When $\mathtt{N}$ is large, even an efficient matrix inversion requires huge computational resources.

To avoid the inversion of high-dimensional matrices, we propose to fully exploit the factorization of the posterior distribution of the transmitted symbols to develop a smoothing expectation propagation (SEP) approximation that proceeds recursively estimating the APP of the symbols in a sliding window of fixed length $\mathtt{W}$. As a first approach, in each iteration k, we estimate $\mathtt{W}$ symbols from $\mathtt{W}$ received ones, then we move forward to the next window, including a new received symbol in the previous set and marginalizing over the oldest one to remove it. We next rewrite this forward algorithm as a backward approach, so for each transmitted symbol we have two different approximations given by both procedures. We solve how to merge both estimations to obtain an improved single APP smoothing estimate for each k transmitted symbol. This new soft equalizer has linear complexity with the frame length, regardless of the constellation size. In addition, we show as preliminary result that the use of the proposed EP-based equalizer in a turbo scheme achieves an extra gain, improving the performance of the well-known turbo LMMSE equalizer [7]. Exploration of how some techniques especially aimed for turbo equalization, such as the family of soft Decision Feedback Equalization (DFE) methods [32]–[34], can benefit from the EP paradigm, remains as future work.

The paper is organized as follows. We first describe in Section II the model and notation of the communication system at hand. In Section III the EP algorithm used in the proposed equalizers is developed. Section IV and Section V are devoted to describe the novel sliding windowed EP forward and backward approaches, respectively, and describe their formulation for a fixed $\mathtt{W}$-length window. In Section VI, both methods are merged into a smoothing approach to include the whole vector of observations in the estimation of the APP for every transmitted symbol. Through the simulations in Section VII, we show that the EP soft equalizer outperforms the LMMSE equalization, and that these APP estimates can be further improved in a turbo architecture. Gains in the 3-5 dB range are reported for 16 and 64-QAM constellations.

The following specific notation is used throughout the paper. We use $i{:}j$ to denote indexes in the range $[i,j]$ if $j > i$ or $[j,i]$ otherwise and $\backslash i{:}j$ to denote all indexes except those in the range $i{:}j$. If $\mathbf{u}$ is a vector, $\mathbf{u}_{i:j}$ is a vector containing the entries of vector $\mathbf{u}$ indexed by $i{:}j$ and $\mathbf{u}_{\backslash i}$ is the vector $\mathbf{u}$ without the i-th component. Similarly, if $\mathbf{\Sigma}$ is a matrix, then $\mathbf{\Sigma}_{\backslash i,\backslash j}$ is the submatrix obtained by removing from $\mathbf{\Sigma}$ the i-th row and the j-th column. We denote a Gaussian probability density function of a random real vector $\mathbf{u}$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$ by $\mathcal{N}(\mathbf{u} : \boldsymbol{\mu}, \mathbf{\Sigma})$. We use the expression $\mathbb{E}_p[\phi(\cdot)]$ to denote the moments with respect to the distribution $p(\cdot)$ with statistics $\phi(\cdot)$. Finally, we denote the true probability

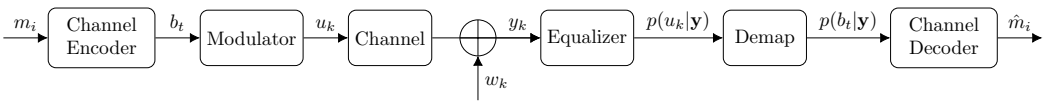


Fig. 1: System model.

density or mass functions as $p(\cdot)$ while we use $q(\cdot)$ for their approximations.

II. System Model

In Fig. 1 we represent the discrete-time communication system. A block of $\mathtt{k}$ message bits, $\mathbf{m} = [m_1, ..., m_{\mathtt{k}}]^\top$, is encoded with a rate $\mathtt{r} = \mathtt{k}/\mathtt{v}$ channel code into $\mathbf{b} = [b_1, ..., b_{\mathtt{v}}]^\top$. An $\mathtt{M}$-ary modulation is considered to obtain $\mathtt{N} = \lceil \mathtt{v}/\log_2 \mathtt{M} \rceil$ symbols, in $\mathbf{u}$. Then, the block frame $\mathbf{u} = [u_1, ..., u_{\mathtt{N}}]^\top = \mathcal{R}(\mathbf{u}) + \mathrm{j}\mathcal{I}(\mathbf{u})$ is transmitted over the channel, where each component $u_k = \mathcal{R}(u_k) + \mathrm{j}\mathcal{I}(u_k) \in \mathcal{A}$, and $\mathcal{A}$ denotes the set of symbols of the constellation of order $|\mathcal{A}| = \mathtt{M}$. The mean symbol energy transmitted is denoted by E_s. The channel is completely specified by the noise variance, σ_w^2, and the CSI, $\mathbf{h} = [h_1, ..., h_{\mathtt{L}}]^\top$, where $\mathtt{L}$ is the length of the channel impulsive response, that we assume known at the receiver. The received signal $\mathbf{y} = [y_1, ..., y_{\mathtt{N}+\mathtt{L}-1}]^\top \in \mathbb{C}^{\mathtt{N}+\mathtt{L}-1}$ is given by

$$y_k = \sum_{l=1}^{\mathtt{L}} h_l u_{k-l+1} + w_k = \mathbf{h}^\top \mathbf{u}_{k:k-\mathtt{L}+1} + w_k, \qquad (1)$$

where $u_i = 0\ \forall i \leqslant 0$ and $\forall i > \mathtt{N}$, and $w_k \sim \mathcal{CN}(w_k : 0, \sigma_w^2)$ $\forall k \in [1, \mathtt{N} + \mathtt{L} - 1]$ are i.i.d. samples of the circular complex-valued additive white Gaussian noise (AWGN) with zero mean and variance σ_w^2. Without loss of generality, in the following we focus on the solution for the real case. The complex-valued case can be easily tackled by using the real-valued formulation where real and imaginary parts are stacked into a real-valued vector for observations, symbols and noise, and the channel matrix is rewritten into an equivalent double-sized real-valued channel matrix.

Given the model above, the posterior probability of the transmitted symbol vector $\mathbf{u}$ yields

$$p(\mathbf{u}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{u})p(\mathbf{u})}{p(\mathbf{y})} \propto \mathcal{N}\left(\mathbf{y} : \mathbf{H}\mathbf{u}, \tfrac{\sigma_w^2}{2}\mathbf{I}\right) \prod_{k=1}^{\mathtt{N}} \mathbb{I}_{u_k \in \mathcal{A}}, \quad (2)$$

where $\mathbf{H}$ is an $(\mathtt{N} + \mathtt{L} - 1) \times (\mathtt{N} + 2\mathtt{L} - 2)$ matrix formed by the $(\mathtt{N} + \mathtt{L} - 1)$ first rows of a circulant matrix with first row $[\mathbf{h}_{\mathtt{L}:1}{}^\top, \mathbf{0}_{1\times(\mathtt{N}+\mathtt{L}-2)}]$ and up to a constant, the prior $p(u_k)$ is the a priori information about u_k, which is given by $\mathbb{I}_{u_k \in \mathcal{A}}$, the indicator function that takes value one if $u_k \in \mathcal{A}$ and zero otherwise.

III. Expectation Propagation

Expectation propagation [23], [24], [35], [36] is a technique in Bayesian machine learning to approximate the true posterior distribution by exponential family distributions. Suppose we are given some statistical distribution with latent vector $\mathbf{x}$ and observables $\mathcal{D}$ that factors as

$$p(\mathbf{x}|\mathcal{D}) \propto f(\mathbf{x}) \prod_{i=1}^{\mathtt{I}} t_i(\mathbf{x}), \qquad (3)$$

where $f(\mathbf{x})$ belongs to an exponential family $\mathcal{F}$ with sufficient statistics $\mathbf{\Phi}(\mathbf{x})$ and $t_i(\mathbf{x})$ are known nonnegative factors that do not belong to the exponential family $\mathcal{F}$. EP provides a feasible approximation to $p(\mathbf{x}|\mathcal{D})$ by an exponential distribution $q(\mathbf{x})$ from $\mathcal{F}$, which factorizes as

$$q(\mathbf{x}) \propto f(\mathbf{x}) \prod_{i=1}^{\mathtt{I}} \tilde{t}_i(\mathbf{x}), \qquad (4)$$

where the parameters of each factor $\tilde{t}_i(\mathbf{x}) \in \mathcal{F}$ (for instance mean vector and covariance matrix if $\mathcal{F}$ corresponds to the Gaussian family) are selected to achieve $\mathbb{E}_q[\mathbf{\Phi}(\mathbf{x})] = \mathbb{E}_p[\mathbf{\Phi}(\mathbf{x})]$. Each factor $\tilde{t}_i(\mathbf{x})$ is optimized independently in the context of all of the remaining factors.

A. Gaussian likelihood model with independent priors

The EP is endowed with a great flexibility, given by the model in (4). In order to improve the accuracy of the EP solution, it is important to retain as much structure as possible from the distribution $p(\mathbf{x}|\mathcal{D})$ and separate it from the latent factors [35]. In this paper, finding an EP approximation for the next model will be a recurrent problem

$$p(\mathbf{x}|\mathcal{D}) \propto \mathcal{N}(\mathbf{x} : \mathbf{z}, \mathbf{C}) \prod_{i=1}^{\mathtt{I}} p(x_i), \qquad (5)$$

where $\mathbf{x} \in \mathbb{R}^{\mathtt{I}}$, the first factor is the likelihood function with known mean vector, $\mathbf{z}$, and covariance matrix, $\mathbf{C}$, and the second factor is the product of priors of the latent variables, x_i. Using EP, we will approximate (5) with a member of the following family of Gaussian pdfs,

$$q(\mathbf{x}) \propto \mathcal{N}(\mathbf{x} : \mathbf{z}, \mathbf{C}) \prod_{i=1}^{\mathtt{I}} \mathrm{e}^{\left(\gamma_i x_i - \frac{1}{2}\Lambda_i x_i^2\right)}, \qquad (6)$$

where $\gamma_i \in \mathbb{R}$ and $\Lambda_i \in \mathbb{R}_+$. Hence, $q(\mathbf{x})$ is a Gaussian $\mathcal{N}(\mathbf{x} : \boldsymbol{\mu}, \mathbf{\Sigma})$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$ as follows

$$\mathbf{\Sigma} = \left(\mathbf{C}^{-1} + \mathrm{diag}(\mathbf{\Lambda})\right)^{-1}, \qquad (7)$$

$$\boldsymbol{\mu} = \mathbf{\Sigma}(\mathbf{C}^{-1}\mathbf{z} + \boldsymbol{\gamma}). \qquad (8)$$

where $\boldsymbol{\gamma} = [\gamma_1, ..., \gamma_{\mathtt{I}}]^\top$ and $\mathbf{\Lambda} = [\Lambda_1, ..., \Lambda_{\mathtt{I}}]^\top$. A detailed implementation of the EP algorithm to learn with this model is included in Algorithm 1 where (13) and (14) are proposed following the guidelines in [25, Eq. 35-36]. In Algorithm 1 $q^{(\ell)}(\mathbf{x})$ is the approximation to $q(\mathbf{x})$ in (4) at iteration ℓ

of the EP algorithm. Also, we use $q^{(\ell)\setminus i}(\mathbf{x})$ to refer to the distribution $q(\mathbf{x})$ at iteration ℓ divided by the i-th factor, $\tilde{t}_i(x)$. At iteration (ℓ) we use the pair $\mu_i^{(\ell)}$ and $\sigma_i^{2^{(\ell)}}$ to denote, respectively, the mean and variance of the distribution $q_i^{(\ell)}(x_i)$ in step 1. Similarly, we use $\mu_{\hat{p}_i}^{(\ell)}$ and $\sigma_{\hat{p}_i}^{2^{(\ell)}}$ to denote the mean and variance of the distribution $\hat{p}^{(\ell)}(x_i)$ in step 2. The initial values of parameters $(\gamma_i^{(1)}, \Lambda_i^{(1)})$ will be set according to the problem at hand. The parameter update in (11) may return a negative value $\Lambda_{i,new}^{(\ell+1)}$ for some i's which means that there is no pair $(\Lambda_{i,new}^{(\ell+1)}, \gamma_{i,new}^{(\ell+1)})$ that sets the variance of the Gaussian in (10). For those i's, we keep the values of the previous iteration. We introduce a smoothing parameter $\beta \in [0,1]$ and a small constant ϵ, both of them set before running the algorithm. Constant ϵ is the minimum allowed variance at each iteration in step 2, $\sigma_{\hat{p}_i}^{2^{(\ell)}} = \max(\epsilon, \sigma_{\hat{p}_i}^{2^{(\ell)}})$, to avoid numerical instabilities. It is interesting to note that the $\mathtt{I}$ approximating factors can be learned in parallel at every iteration ℓ. But before updating them, at the beginning of every iteration we need to re-compute (7) that, in the general case, involves a computational complexity of order $\mathcal{O}(\mathtt{I}^3)$ due to the computation of the inverse matrix. After S iterations, the complexity yields $\mathcal{O}((S+1)\mathtt{I}^3)$, where S is the number of iterations of the EP algorithm (see Algorithm 1). This value should be chosen to ensure convergence, that depends on the problem at hand.

Algorithm 1 EP for Gaussian Posterior

Initialize $\left(\gamma_i^{(1)}, \Lambda_i^{(1)}\right) \ \forall\ i = 1, ..., \mathtt{I}$.
for $\ell = 1, ..., S$ **do**
Calculate the distribution $q^{(\ell)}(\mathbf{x})$ in (6) with $\gamma_i \leftarrow \gamma_i^{(\ell)}$ and $\Lambda_i \leftarrow \Lambda_i^{(\ell)}$.
for $i = 1, ..., \mathtt{I}$ **do**
1) Compute the i-th marginal of $q^{(\ell)}(\mathbf{x})$, denoted as $q_i^{(\ell)}(x_i) = \mathcal{N}\left(x_i : \mu_i^{(\ell)}, \sigma_i^{2^{(\ell)}}\right)$, and the distribution

$$q^{(\ell)\setminus i}(x_i) = q_i^{(\ell)}(x_i)/\mathrm{e}^{\left(\gamma_i^{(\ell)} x_i - \frac{1}{2}\Lambda_i^{(\ell)} x_i^2\right)} \sim \sim \mathcal{N}\left(x_i : \eta_i^{(\ell)}, \nu_i^{2^{(\ell)}}\right), \quad (9)$$

namely *cavity* marginal function, where

$$\nu_i^{2^{(\ell)}} = \frac{\sigma_i^{2^{(\ell)}}}{1 - \sigma_i^{2^{(\ell)}}\Lambda_i^{(\ell)}}, \quad \eta_i^{(\ell)} = \nu_i^{2^{(\ell)}}\left(\frac{\mu_i^{(\ell)}}{\sigma_i^{2^{(\ell)}}} - \gamma_i^{(\ell)}\right).$$

2) Obtain the distribution $\hat{p}^{(\ell)}(x_i) \propto q^{(\ell)\setminus i}(x_i)\mathbb{I}_{x_i \in \mathcal{A}}$ and estimate its mean $\mu_{\hat{p}_i}^{(\ell)}$ and variance $\sigma_{\hat{p}_i}^{2^{(\ell)}}$.
3) Setting the mean and variance of the unnormalized Gaussian distribution

$$q^{(\ell)\setminus i}(x_i)\mathrm{e}^{\left(\gamma_{i,new}^{(\ell+1)} x_i - \frac{1}{2}\Lambda_{i,new}^{(\ell+1)} x_i^2\right)} \quad (10)$$

equal to $\mu_{\hat{p}_i}^{(\ell)}$ and $\sigma_{\hat{p}_i}^{2^{(\ell)}}$, we get the solution

$$\Lambda_{i,new}^{(\ell+1)} = 1/\sigma_{\hat{p}_i}^{2^{(\ell)}} - 1/\nu_i^{2^{(\ell)}}, \quad (11)$$
$$\gamma_{i,new}^{(\ell+1)} = \mu_{\hat{p}_i}^{(\ell)}/\sigma_{\hat{p}_i}^{2^{(\ell)}} - \eta_i^{(\ell)}/\nu_i^{2^{(\ell)}}. \quad (12)$$

4) Update the values as

$$\Lambda_i^{(\ell+1)} = \beta\Lambda_{i,new}^{(\ell+1)} + (1-\beta)\Lambda_i^{(\ell)}, \quad (13)$$
$$\gamma_i^{(\ell+1)} = \beta\gamma_{i,new}^{(\ell+1)} + (1-\beta)\gamma_i^{(\ell)}. \quad (14)$$

end for
end for
With the values $\boldsymbol{\gamma}^{(S+1)}, \boldsymbol{\Lambda}^{(S+1)}$ obtained after EP algorithm, calculate the final values of the mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ given by (8) and (7), respectively.

IV. Forward Windowed EP Equalizer

We first introduce an approximated EP algorithm that proceeds forward iteratively processing the observations. To keep the notation uncluttered, we define the memory of the channel as $\tilde{\mathtt{L}} = \mathtt{L} - 1$, the variables in the window at iteration k as $\mathbf{s}_k = \mathbf{u}_{k-\mathtt{W}+1:k}$ and the variables in the window except for the newest one as $\tilde{\mathbf{s}}_k = \mathbf{u}_{k-\mathtt{W}+1:k-1}$ for some value $\mathtt{W} \geqslant \mathtt{L}$. From the definitions, note that we can write $\mathbf{s}_k = [\tilde{\mathbf{s}}_k^\top,\ u_k]^\top = [u_{k-\mathtt{W}+1},\ \tilde{\mathbf{s}}_{k+1}^\top]^\top$.

A. Description

At iteration k we have observations $\mathbf{y}_{1:k}$, that are related to the to-be-estimated transmitted symbols by the following probability mass function (pmf):

$$p(\mathbf{u}_{-\tilde{\mathtt{L}}+1:k}|\mathbf{y}_{1:k}) \propto p(\mathbf{y}_{1:k}|\mathbf{u}_{-\tilde{\mathtt{L}}+1:k}) \prod_{i=-\tilde{\mathtt{L}}+1}^{k} p(u_i), \quad (15)$$

where $\mathbf{u}_{-\tilde{\mathtt{L}}+1:0}$ and $\mathbf{u}_{\mathtt{N}+1:\mathtt{N}+\mathtt{L}}$ are set to zero[1]. The pmf in (15), by splitting $p(\mathbf{y}_{1:k}|\mathbf{u}_{-\tilde{\mathtt{L}}+1:k})$ into $p(y_k|\mathbf{u}_{k-\tilde{\mathtt{L}}:k})p(\mathbf{y}_{1:k-1}|\mathbf{u}_{-\tilde{\mathtt{L}}+1:k-1})$, can be written as follows:

$$p(\mathbf{u}_{-\tilde{\mathtt{L}}+1:k}|\mathbf{y}_{1:k}) \propto p(y_k|\mathbf{u}_{k-\tilde{\mathtt{L}}:k})p(\mathbf{u}_{-\tilde{\mathtt{L}}+1:k-1}|\mathbf{y}_{1:k-1})p(u_k). \quad (16)$$

[1]For simplicity, we assume that at $k < 1$ and $k > \mathtt{N}$ nothing is transmitted and we use $p(u_k = 0) = 1$ and zero otherwise in the formulation.

Given the joint posterior distribution in (16), consider the joint posterior distribution of the last $\mathtt{W}$ consecutive transmitted symbols, $\mathbf{s}_k$. From (16) it is easy to check that

$$p(\mathbf{s}_k|\mathbf{y}_{1:k}) = \int p(\mathbf{u}_{-\tilde{\mathtt{L}}+1:k}|\mathbf{y}_{1:k})\mathrm{d}\mathbf{u}_{-\tilde{\mathtt{L}}+1:k-\mathtt{W}} \propto p(y_k|\mathbf{u}_{k-\tilde{\mathtt{L}}:k})p(\tilde{\mathbf{s}}_k|\mathbf{y}_{1:k-1})p(u_k). \quad (17)$$

As derived from (17), the term $p(\tilde{\mathbf{s}}_k|\mathbf{y}_{1:k-1})$ is the result of the forward algorithm in the previous iteration by marginalization. Given $p(\tilde{\mathbf{s}}_k|\mathbf{y}_{1:k-1})$, the goal of the forward equalizer at iteration k is to provide $p(\mathbf{s}_k|\mathbf{y}_{1:k})$ via (17). This posterior distribution is then marginalized over $u_{k-\mathtt{W}+1}$ to compute $p(\tilde{\mathbf{s}}_{k+1}|\mathbf{y}_{1:k})$, that is used to estimate $p(\mathbf{s}_{k+1}|\mathbf{y}_{1:k+1})$, following (17) again, in the next iteration. It is straightforward to show that for $\mathtt{W} = \mathtt{L}$ this process yields a recursive Bayesian estimation [9], [37]. If we increase $\mathtt{W}$, the system can be seen as a forward approach where we are allowed to check

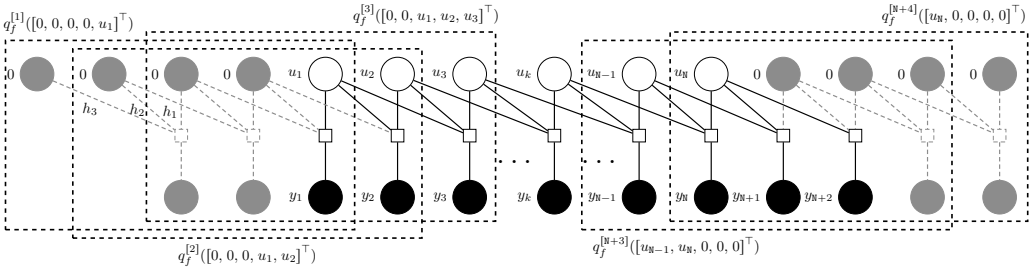


Fig. 2: Factor graph associated to (1) for $\mathtt{L}=3$, $\mathtt{W}=5$ and FWEP performance.

for the $\mathtt{W}-\mathtt{L}$ future observations, improving the estimations for the first symbols in the window. The main advantage of the whole procedure is that the complexity is linear with the number of iterations. However, marginalizing (17) to compute $p(\tilde{\mathbf{s}}_{k+1}|\mathbf{y}_{1:k})$ has complexity $\mathcal{O}(\mathtt{M}^{\mathtt{W}})$, that increases rapidly with the length of the window $\mathtt{W}$, and/or the size of the constellation, $\mathtt{M}$. At this point we rely on an EP-based Gaussian approximation to (17). First, we rewrite the term $p(\tilde{\mathbf{s}}_k|\mathbf{y}_{1:k-1})$ in (17) as follows

$$p(\tilde{\mathbf{s}}_k|\mathbf{y}_{1:k-1}) \propto p(\mathbf{y}_{1:k-1}|\tilde{\mathbf{s}}_k) \prod_{i=k-\mathtt{W}+1}^{k-1} p(u_i) \tag{18}$$

and introduce it into (17), yielding

$$p(\mathbf{s}_k|\mathbf{y}_{1:k}) \propto p(y_k|\mathbf{u}_{k-\tilde{\mathtt{L}}:k}) p(\mathbf{y}_{1:k-1}|\tilde{\mathbf{s}}_k) \prod_{i=k-\mathtt{W}+1}^{k} p(u_i). \tag{19}$$

This expression is appealing, since it fits the model (5) for which an EP approximation is described. Assume for now that the previous EP forward iteration provided a Gaussian approximation to $p(\mathbf{y}_{1:k-1}|\tilde{\mathbf{s}}_k)$, that we denote by $g^{[k-1]}(\tilde{\mathbf{s}}_k)$. We propose to use the following family $\mathcal{F}$ of Gaussian pdfs to find an EP-based approximation to (19)

$$q_f^{[k]}(\mathbf{s}_k) \overset{\text{def}}{=} p(y_k|\mathbf{u}_{k-\tilde{\mathtt{L}}:k}) g^{[k-1]}(\tilde{\mathbf{s}}_k) \prod_{i=k-\mathtt{W}+1}^{k} \mathrm{e}^{\left(\gamma_i^{[k]} u_i - \frac{1}{2}\Lambda_i^{[k]} u_i^2\right)}. \tag{20}$$

We use the superscript $^{[k]}$ to explicitly indicate at what iteration the different terms are updated. For instance, $g^{[k-1]}(\tilde{\mathbf{s}}_k)$ was computed at iteration $k-1$, while $q_f^{[k]}(\mathbf{s}_k)$, $\gamma_i^{[k]}$ and $\Lambda_i^{[k]}$ were computed at iteration k. Also, the subscript f in $q_f^{[k]}(\mathbf{s}_k)$ is included to avoid confusion with the equivalent estimation provided by the backward counterpart of the algorithm explained in Section V. Note that $q_f^{[k]}(\mathbf{s}_k)$ in (20) is a function of parameters $\gamma_i^{[k]}$ and $\Lambda_i^{[k]}$ for $i=k-\mathtt{W}+1,...,k$. Details about how to compute these parameters with EP to ensure that (20) has mean and covariance matrix close to those in (19) are given in the next subsection. After the EP approximation in (20) has been computed, we can easily estimate the marginal over the symbol $u_{k-\mathtt{W}+1}$, namely $q_f^{[k]}(u_{k-\mathtt{W}+1})$, which is fed to the channel decoder. Also, recall that $\mathbf{s}_k=[u_{k-\mathtt{W}+1},\ \tilde{\mathbf{s}}_{k+1}]$ and thus after integrating out $u_{k-\mathtt{W}+1}$ in (20) we obtain $q_f^{[k]}(\tilde{\mathbf{s}}_{k+1})$, the current EP approach to $p(\tilde{\mathbf{s}}_{k+1}|\mathbf{y}_{1:k})$. Finally, the k-th iteration of the forward windowed EP equalizer ends by computing the moments of the Gaussian distribution

$$g^{[k]}(\tilde{\mathbf{s}}_{k+1}) = \frac{q_f^{[k]}(\tilde{\mathbf{s}}_{k+1})}{\prod_{i=k-\mathtt{W}+2}^{k} \mathrm{e}^{\left(\gamma_i^{[k]} u_i - \frac{1}{2}\Lambda_i^{[k]} u_i^2\right)}}, \tag{21}$$

which by (18) is the current approximation to $p(\mathbf{y}_{1:k}|\tilde{\mathbf{s}}_{k+1})$ and will be used in the next iteration of the forward windowed EP equalizer. Table I summarizes the distributions defined to describe each iteration of the algorithm.

TABLE I: Gaussian distributions in the FWEP equalizer.

Distribution	Approximates	Mean	Cov. Matrix
$\mathrm{g}^{[k-1]}(\tilde{\mathbf{s}}_k)$	$p(\mathbf{y}_{1:k-1}\|\tilde{\mathbf{s}}_k)$	$\boldsymbol{\mu}^{\mathrm{g},[k-1]}$	$\boldsymbol{\Sigma}^{\mathrm{g},[k-1]}$
$q_f^{[k]}(\mathbf{s}_k)$	$p(\mathbf{s}_k\|\mathbf{y}_{1:k})$	$\boldsymbol{\mu}^{q,[k]}$	$\boldsymbol{\Sigma}^{q,[k]}$
$q_f^{[k]}(\tilde{\mathbf{s}}_{k+1})$	$p(\tilde{\mathbf{s}}_{k+1}\|\mathbf{y}_{1:k})$	$\boldsymbol{\mu}_{\setminus 1}^{q,[k]}$	$\boldsymbol{\Sigma}_{\setminus 1,\setminus 1}^{q,[k]}$
$\mathrm{g}^{[k]}(\tilde{\mathbf{s}}_{k+1})$	$p(\mathbf{y}_{1:k}\|\tilde{\mathbf{s}}_{k+1})$	$\boldsymbol{\mu}^{\mathrm{g},[k]}$	$\boldsymbol{\Sigma}^{\mathrm{g},[k]}$

In the following, to keep the notation uncluttered, we will refer to $\gamma_i^{[k]}$ and $\Lambda_i^{[k]}$ simply as γ_i and Λ_i. In Fig. 2 we depict the factor graph associated to (1) for the case $\mathtt{L}=3$ and $\mathtt{W}=5$ where the sliding window is drawn with dashed lines. It can be observed that in each iteration the window is moved forward, including a new transmitted symbol and a new observation, and marginalizing over the oldest transmitted symbol. As a result, we have a linear complexity with the number of iterations, $\mathtt{N}$, and cubic complexity with the window length, $\mathtt{W}$. We denote this method as *forward windowed expectation propagation* (FWEP).

B. Formulation

In the development of the FWEP algorithm we assume that the initial and final transmitted symbols are known. In particular, and without loss of generality, $\mathbf{u}_{-\mathtt{W}+2:0}$ are assumed zero. Hence, the pmf for $\tilde{\mathbf{s}}_k$ at the first iteration, $k=1$, is $p(\tilde{\mathbf{s}}_1=[0...0]^\top)=1$ and zero otherwise. This pmf is approximated in the EP approach by defining Gaussians with zero mean and a low enough variance, i.e.,

$q_f^{[0]}(\tilde{\mathbf{s}}_1) = \mathcal{N}\left(\tilde{\mathbf{s}}_1 : [0\dots 0]^\top, \sigma_{min}^2 \mathbf{I}\right)$. In a similar way, recall that for the last $k = \mathtt{N}+1, ..., \mathtt{N}+\mathtt{W}-1$ iterations we assume that the final transmitted symbols $\mathbf{u}_{\mathtt{N}+1:\mathtt{N}+\mathtt{W}-1}$ are zero. For these symbols, we force $(\gamma_k, \Lambda_k) = (0, \sigma_{min}^{-2})$ and keep them fixed.

In the following, we present the formulation of the FWEP update rules focusing on learning the approximated distributions in (20) and (21). We assume that, as a result of the previous FWEP iteration, the moments of $g^{[k-1]}(\tilde{\mathbf{s}}_k)$, i.e. $\boldsymbol{\mu}^{g,[k-1]}$ and $\boldsymbol{\Sigma}^{g,[k-1]}$ are known. The Gaussian likelihood factor $p(y_k|\mathbf{u}_{k-\tilde{\mathtt{L}}:k})$ in (19) is given by the channel model in (1). If we define $\mathbf{Q} = \sigma_w^{-2}\mathbf{h}_{\mathtt{L}:1}\mathbf{h}^\top{}_{\mathtt{L}:1}$ and $\mathbf{r} = \sigma_w^{-2}\mathbf{h}_{\mathtt{L}:1}y_k$, it is possible to merge the first two terms in (20) into just one Gaussian distribution and rewrite it as

$$q_f^{[k]}(\mathbf{s}_k) \overset{\text{def}}{=} \mathcal{N}\left(\mathbf{s}_k : \mathbf{z}^{[k]}, \mathbf{C}^{[k]}\right) \prod_{i=k-\mathtt{W}+1}^{k} \mathrm{e}^{\left(\gamma_i u_i - \frac{1}{2}\Lambda_i u_i^2\right)} \quad (22)$$

where

$$\mathbf{C}^{[k]^{-1}} = \\ = \begin{bmatrix} \mathbf{0}_{(\mathtt{W}-\mathtt{L})\times(\mathtt{W}-\mathtt{L})} & \mathbf{0}_{(\mathtt{W}-\mathtt{L})\times\mathtt{L}} \\ \mathbf{0}_{\mathtt{L}\times(\mathtt{W}-\mathtt{L})} & \mathbf{Q} \end{bmatrix} + \begin{bmatrix} \left(\boldsymbol{\Sigma}^{g,[k-1]}\right)^{-1} & \mathbf{0}_{(\mathtt{W}-1)\times 1} \\ \mathbf{0}_{1\times(\mathtt{W}-1)} & 0 \end{bmatrix}, \quad (23)$$

$$\mathbf{C}^{[k]^{-1}}\mathbf{z}^{[k]} = \left(\begin{bmatrix} \mathbf{0}_{(\mathtt{W}-\mathtt{L})\times 1} \\ \mathbf{r} \end{bmatrix} + \begin{bmatrix} \left(\boldsymbol{\Sigma}^{g,[k-1]}\right)^{-1}\boldsymbol{\mu}^{g,[k-1]} \\ 0 \end{bmatrix}\right). \quad (24)$$

Note that the goal of the EP approximation at iteration k is to find $\boldsymbol{\Lambda}_{k-\mathtt{W}+1:k}$ and $\boldsymbol{\gamma}_{k-\mathtt{W}+1:k}$ in (22) so that the moments of $q_f^{[k]}(\mathbf{s}_k)$ approximate well the moments of $p(\mathbf{s}_k|\mathbf{y}_{1:k})$ in (19). Further, note that the model in (22) is equivalent to that described in Subsection III-A and thus to learn $\boldsymbol{\Lambda}_{k-\mathtt{W}+1:k}$ and $\boldsymbol{\gamma}_{k-\mathtt{W}+1:k}$ we can resort directly to Algorithm 1. Parameters (γ_i, Λ_i) $\forall i \in [k-\mathtt{W}+1, k-1]$ are initialized to the values obtained in the previous k-th iteration while $(\gamma_k, \Lambda_k) = (0, E_s)$. The mean and covariance of the resulting pdf for $q_f^{[k]}(\mathbf{s}_k)$, $\mathcal{N}\left(\mathbf{s}_k : \boldsymbol{\mu}^{q,[k]}, \boldsymbol{\Sigma}^{q,[k]}\right)$, are given by (8) and (7),

$$\boldsymbol{\Sigma}^{q,[k]} = \left(\mathbf{C}^{[k]^{-1}} + \mathrm{diag}(\boldsymbol{\Lambda}_{k-\mathtt{W}+1:k})\right)^{-1}, \quad (25)$$

$$\boldsymbol{\mu}^{q,[k]} = \boldsymbol{\Sigma}^{q[k]}\left(\mathbf{C}^{[k]^{-1}}\mathbf{z}^{[k]} + \boldsymbol{\gamma}_{k-\mathtt{W}+1:k}\right). \quad (26)$$

Finally, given $\boldsymbol{\Sigma}^{q,[k]}$ and $\boldsymbol{\mu}^{q,[k]}$, the moments of $g^{[k]}(\tilde{\mathbf{s}}_{k+1})$ are straightforward to compute. From (21) we obtain:

$$\left(\boldsymbol{\Sigma}^{g,[k]}\right)^{-1} = \left(\boldsymbol{\Sigma}^{q,[k]}_{\setminus 1,\setminus 1}\right)^{-1} - \mathrm{diag}(\boldsymbol{\Lambda}_{k-\mathtt{W}+2:k}) \quad (27)$$

$$\boldsymbol{\mu}^{g,[k]} = \boldsymbol{\Sigma}^{g,[k]}\left(\left(\boldsymbol{\Sigma}^{q,[k]}_{\setminus 1,\setminus 1}\right)^{-1}\boldsymbol{\mu}^{q,[k]}_{\setminus 1} - \boldsymbol{\gamma}_{k-\mathtt{W}+2:k}\right), \quad (28)$$

where $\boldsymbol{\mu}^{q,[k]}_{\setminus 1}$ is the vector $\boldsymbol{\mu}^{q,[k]}$ in (26) without the first element and $\boldsymbol{\Sigma}^{q,[k]}_{\setminus 1,\setminus 1}$ is the submatrix obtained by removing from $\boldsymbol{\Sigma}^{q,[k]}$ in (25) the first row and column. A detailed implementation of the FWEP iterative algorithm is included in Algorithm 2. Since we use a window of size $\mathtt{W}$, the transmitted symbol u_k is involved in $\mathtt{W}$ iterations, i.e., from iteration k to $k+\mathtt{W}-1$, and we have $\mathtt{W}$ different estimations for each symbol. We use the approximation $q_f^{[k+\mathtt{W}-1]}(\mathbf{s}_{k+\mathtt{W}-1})$ to estimate u_k since it is the one that includes more observations, specifically it exploits the received observations until time $k+\mathtt{W}-1$, i.e., $\mathbf{y}_{1:k+\mathtt{W}-1}$. At every iteration, k, the computational complexity of the FWEP is dominated by the inversion of the covariance matrix in (25), of dimension $\mathtt{W}$. Therefore, we have a quite efficient implementation with computational complexity $\mathcal{O}((S+1)\mathtt{N}\mathtt{W}^3)$, where S is the number of iterations of the EP algorithm. Note that after the EP algorithm, a last inversion is needed to compute the APP. A lower value for $\mathtt{W}$ reduces the computational complexity, however a larger size involves more future observations and improves the estimation. A value $\mathtt{W} \geqslant \mathtt{L}$, implies that all observations directly connected to the to be-estimated variable are included. Note that $\mathtt{W} = \mathtt{L}$ defines the state for the recursive Bayesian estimator.

Algorithm 2 The FWEP approximation

Set the value of the parameter $\mathtt{W}$.
Initialize $\boldsymbol{\Sigma}^{q,[0]} = \mathrm{diag}(\sigma_{min}^2[1\cdots 1])$, $\boldsymbol{\mu}^{q,[0]} = [0\cdots 0]^\top$, $\boldsymbol{\gamma}_{-\mathtt{W}+2:0} = [0\cdots 0]^\top$ and $\boldsymbol{\Lambda}_{-\mathtt{W}+2:0} = \sigma_{min}^{-2}[1\cdots 1]^\top$.
for $k = 1, ..., \mathtt{N}+\mathtt{W}-1$ **do**
1) Initialize Algorithm 1 with

$$(\gamma_k, \Lambda_k) = \begin{cases} (0, E_s^{-1}) & \text{if } 1 \leqslant k \leqslant \mathtt{N} \\ (0, \sigma_{min}^{-2}) & \text{if } \mathtt{N} < k \leqslant \mathtt{N}+\mathtt{W}-1 \end{cases} \quad (29)$$

and (γ_i, Λ_i) for $i \in [k-\mathtt{W}+1, k-1]$ with the values obtained in the previous k-th iteration.
2) Apply Algorithm 1 with S, $\mathtt{I} = \mathtt{N}$, $\mathbf{x} = \mathbf{s}_k = \mathbf{u}_{k-\mathtt{W}+1:k}$, and $\mathbf{z}$ and $\mathbf{C}$ in (6) given by (24) and (23), respectively. Any estimate (γ_i, Λ_i) with $i \in [-\mathtt{W}+2, 0]$ or $i \in [\mathtt{N}+1, \mathtt{N}+\mathtt{W}-1]$ is not updated.
end for

V. Backward Windowed Expectation Propagation

As already discussed, the approximation used for FWEP exploits the received observations until time $k+\mathtt{W}-1$, i.e., $\mathbf{y}_{1:k+\mathtt{W}-1}$ to estimate u_k. However, it does not use the information of the following ones, $\mathbf{y}_{k+\mathtt{W}:\mathtt{N}+\mathtt{W}-1}$. We next propose the backward recursion as a first step to include these observations in the estimation. The key idea is just to use the same formulation for the FWEP but from the end of the sequence of observations towards the begining. This algorithm can be easily formulated by left-right flipping the channel, observation and symbol vectors, and applying the FWEP algorithm to them, as illustrated in Fig. 3. We denote this method *backward windowed expectation propagation* (BWEP). In Fig. 3 we include the factor graph in Fig. 2 flipped. The FWEP is applied to this graph to get the BWEP solution. In Appendix A we describe the change of variables associated to this flipping needed to run the BWEP algorithm as an instance of the FWEP. Note that there is an $\mathtt{L}-1$ offset in the relative position of the observations, y_k, and transmitted symbols, u_k, with respect to Fig. 2. This way we ensure that, at each iteration k, the sliding window is shifted to include a new observation y_k, and all its directly connected symbols, $\mathbf{u}_{k-\mathtt{L}+1:k}$. As a result,

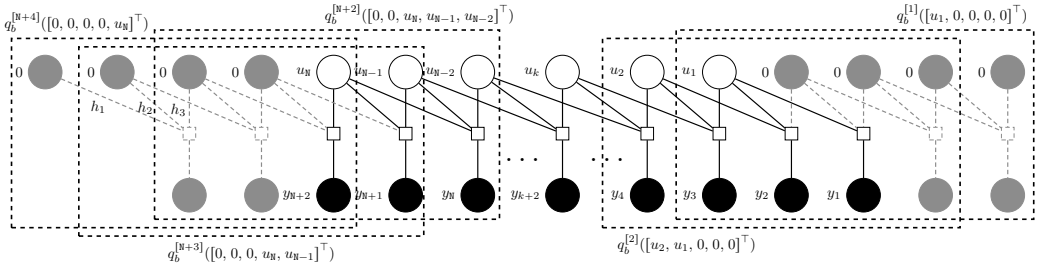


Fig. 3: Factor graph associated to (1) for $\mathtt{L} = 3$, $\mathtt{W} = 5$ and BWEP performance. The sliding window, dashed, shifts from left to right.

the BWEP will provide a Gaussian approximation, $q_b^{[k]}(\mathbf{s}_k)$, to $p(\mathbf{s}_k|\mathbf{y}_{k-\mathtt{W}+\mathtt{L}:\mathtt{N}+\mathtt{W}-1})$, $k = 1, ..., \mathtt{N}+\mathtt{W}-1$, from which we estimate by marginalization a pmf for the symbol u_k (recall that in the FWEP case, we marginalize $q_f^{[k]}(\mathbf{s}_k)$ to estimate symbol $u_{k-\mathtt{W}+1}$).

VI. Smoothed Expectation Propagation

The estimation in the FWEP ignores the future observations while the BWEP estimation does not exploit the previous ones. In this section we introduce the *smoothing expectation propagation* (SEP) to improve the estimation by merging both procedures into one. In the forward bayesian recursive approach, the optimal solution can be estimated with a window size of L. This window size can be adopted for the FWEP, to estimate the posterior of u_k from $\mathbf{u}_{k:k+\tilde{\mathtt{L}}}$. Following the same reasoning, to estimate the posterior of u_k with the BWEP we would use a window of L spanning $\mathbf{u}_{k-\tilde{\mathtt{L}}:k}$. In order to deal with both estimations, we implement both algorithms with a window size $2\tilde{\mathtt{L}}-1$ and combine their estimates. In the following we show how both estimations can be merged and describe the SEP algorithm.

A. Description

As in the forward approach, we first rewrite the wanted probabilities into a suitable form for the EP algorithm. The following lemma summarizes this first result.

Lemma 6.1: The posterior probabilites, $p(\mathbf{u}_{k-\tilde{\mathtt{L}}:k+\tilde{\mathtt{L}}}|\mathbf{y})$, up to a constant, can be written as

$$p(\mathbf{u}_{k-\tilde{\mathtt{L}}:k+\tilde{\mathtt{L}}}|\mathbf{y}) \propto \frac{p(\mathbf{u}_{k-\tilde{\mathtt{L}}:k+\tilde{\mathtt{L}}}|\mathbf{y}_{1:k+\tilde{\mathtt{L}}})p(\mathbf{u}_{k-\tilde{\mathtt{L}}:k+\tilde{\mathtt{L}}}|\mathbf{y}_{k:\mathtt{N}+\tilde{\mathtt{L}}})}{\prod_{i=k}^{k+\tilde{\mathtt{L}}} p(y_i|\mathbf{u}_{i-\tilde{\mathtt{L}}:i}) \prod_{i=k-\tilde{\mathtt{L}}}^{k+\tilde{\mathtt{L}}} p(u_i)}. \tag{30}$$

Proof: From (2), the exact posterior probability distribution is

$$p(\mathbf{u}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{u})p(\mathbf{u}) = \prod_{k=1}^{\mathtt{N}+\tilde{\mathtt{L}}} p(y_k|\mathbf{u}_{k-\tilde{\mathtt{L}}:k}) \prod_{k=-\tilde{\mathtt{L}}+1}^{\mathtt{N}+\tilde{\mathtt{L}}} p(u_k) \tag{31}$$

where $\mathbf{u}_{-\tilde{\mathtt{L}}+1:0}$ and $\mathbf{u}_{\mathtt{N}+1:\mathtt{N}+\tilde{\mathtt{L}}}$ are known. We aim at estimating u_k, by using $q_f^{[k+\tilde{\mathtt{L}}]}(\mathbf{s}_{k+\tilde{\mathtt{L}}})$ in the forward step and $q_b^{[k+\tilde{\mathtt{L}}]}(\mathbf{s}_{k+\tilde{\mathtt{L}}})$ in the backward step. Accordingly, we split (31) in two parts: received symbols until time $k+\tilde{\mathtt{L}}$ and after time $k+\tilde{\mathtt{L}}$,

$$\begin{aligned} p(\mathbf{u}|\mathbf{y}) &= \frac{1}{p(\mathbf{y})} p(\mathbf{y}_{1:k+\tilde{\mathtt{L}}}|\mathbf{u}_{-\tilde{\mathtt{L}}+1:k+\tilde{\mathtt{L}}}) p(\mathbf{u}_{-\tilde{\mathtt{L}}+1:k+\tilde{\mathtt{L}}}) \\ &\quad \prod_{i=k+\tilde{\mathtt{L}}+1}^{\mathtt{N}+\tilde{\mathtt{L}}} p(y_i|\mathbf{u}_{i-\tilde{\mathtt{L}}:i}) p(u_i) \\ &\propto p(\mathbf{u}_{-\tilde{\mathtt{L}}+1:k+\tilde{\mathtt{L}}}|\mathbf{y}_{1:k+\tilde{\mathtt{L}}}) \prod_{i=k+\tilde{\mathtt{L}}+1}^{\mathtt{N}+\tilde{\mathtt{L}}} p(y_i|\mathbf{u}_{i-\tilde{\mathtt{L}}:i}) p(u_i). \end{aligned} \tag{32}$$

We now marginalize $p(\mathbf{u}|\mathbf{y})$ over $\mathbf{u}_{-\tilde{\mathtt{L}}+1:k-\tilde{\mathtt{L}}-1}$ simplifying the first factor, from the FWEP description in (17),

$$p(\mathbf{u}_{k-\tilde{\mathtt{L}}:\mathtt{N}+\tilde{\mathtt{L}}}|\mathbf{y}) \propto p(\mathbf{u}_{k-\tilde{\mathtt{L}}:k+\tilde{\mathtt{L}}}|\mathbf{y}_{1:k+\tilde{\mathtt{L}}}) \prod_{i=k+\mathtt{L}}^{\mathtt{N}+\tilde{\mathtt{L}}} p(y_i|\mathbf{u}_{i-\tilde{\mathtt{L}}:i}) p(u_i). \tag{33}$$

To rewrite the last term of (33) into the posterior $p(\mathbf{u}_{k-\tilde{\mathtt{L}}:\mathtt{N}+\tilde{\mathtt{L}}}|\mathbf{y}_{k:\mathtt{N}+\tilde{\mathtt{L}}})$, the following factors are missing

$$\prod_{i=k}^{k+\tilde{\mathtt{L}}} p(y_i|\mathbf{u}_{i-\tilde{\mathtt{L}}:i}) \prod_{i=k-\tilde{\mathtt{L}}}^{k+\tilde{\mathtt{L}}} p(u_i). \tag{34}$$

By multiplying and dividing (33) by (34), we get the equivalent distribution

$$p(\mathbf{u}_{k-\tilde{\mathtt{L}}:\mathtt{N}+\tilde{\mathtt{L}}}|\mathbf{y}) \propto \frac{p(\mathbf{u}_{k-\tilde{\mathtt{L}}:k+\tilde{\mathtt{L}}}|\mathbf{y}_{1:k+\tilde{\mathtt{L}}}) p(\mathbf{u}_{k-\tilde{\mathtt{L}}:\mathtt{N}+\tilde{\mathtt{L}}}|\mathbf{y}_{k:\mathtt{N}+\tilde{\mathtt{L}}})}{\prod_{i=k}^{k+\tilde{\mathtt{L}}} p(y_i|\mathbf{u}_{i-\tilde{\mathtt{L}}:i}) \prod_{i=k-\tilde{\mathtt{L}}}^{k+\tilde{\mathtt{L}}} p(u_i)}. \tag{35}$$

Then after marginalizing $p(\mathbf{u}_{k-\tilde{\mathtt{L}}:\mathtt{N}+\tilde{\mathtt{L}}}|\mathbf{y})$ over the last symbols $\mathbf{u}_{k+\mathtt{L}:\mathtt{N}+\tilde{\mathtt{L}}}$, it yields (30). ■

Assume we run the FWEP and BWEP algorithms with $\mathtt{W} = 2\tilde{\mathtt{L}}-1$. Note that with this window size $\mathbf{s}_{k+\tilde{\mathtt{L}}} = \mathbf{u}_{k-\tilde{\mathtt{L}}:k+\tilde{\mathtt{L}}}$. We propose to approximate the numerator in (30) by using the forward and backward approximations developed in Sections IV and V as follows,

$$p(\mathbf{u}_{k-\tilde{\mathtt{L}}:k+\tilde{\mathtt{L}}}|\mathbf{y}_{1:k+\tilde{\mathtt{L}}}) \to q_f^{[k+\tilde{\mathtt{L}}]}(\mathbf{s}_{k+\tilde{\mathtt{L}}}) \tag{36}$$

$$p(\mathbf{u}_{k-\tilde{\mathtt{L}}:k+\tilde{\mathtt{L}}}|\mathbf{y}_{k:\mathtt{N}+\tilde{\mathtt{L}}}) \to q_b^{[k+\tilde{\mathtt{L}}]}(\mathbf{s}_{k+\tilde{\mathtt{L}}}), \tag{37}$$

where $\rightarrow$ denotes that the term on its left is replaced by the one to its right. Regarding the priors in the denominator of (30), we approximate them by exponentials as we did in (5)-(6),

$$\prod_{i=k-\tilde{\mathtt{L}}}^{k+\tilde{\mathtt{L}}} p(u_i) \rightarrow \prod_{i=k-\tilde{\mathtt{L}}}^{k+\tilde{\mathtt{L}}} \mathrm{e}^{\left(\gamma_i u_i - \frac{1}{2}\Lambda_i u_i^2\right)}. \tag{38}$$

We assign to the pairs (γ_i, Λ_i) the values obtained with the $k+\tilde{\mathtt{L}}$ forward approximation for $k < i \leqslant k+\tilde{\mathtt{L}}$ (denoted as $\boldsymbol{\gamma}_{\mathbf{f}k+1:k+\tilde{\mathtt{L}}}$, $\boldsymbol{\Lambda}_{\mathbf{f}k+1:k+\tilde{\mathtt{L}}}$) and with the $k+\tilde{\mathtt{L}}$ backward approximation for $k-\tilde{\mathtt{L}} \leqslant i \leqslant k$ (denoted as $\boldsymbol{\gamma}_{\mathbf{b}k-\tilde{\mathtt{L}}:k}$, $\boldsymbol{\Lambda}_{\mathbf{b}k-\tilde{\mathtt{L}}:k}$). The SEP approximation yields

$$p(\mathbf{s}_{k+\tilde{\mathtt{L}}}|\mathbf{y}) \approx \frac{q_f^{[k+\tilde{\mathtt{L}}]}(\mathbf{s}_{k+\tilde{\mathtt{L}}}) q_b^{[k+\tilde{\mathtt{L}}]}(\mathbf{s}_{k+\tilde{\mathtt{L}}})}{\prod_{i=k}^{k+\tilde{\mathtt{L}}} p(y_i|\mathbf{u}_{i-\tilde{\mathtt{L}}:i}) \prod_{i=k-\tilde{\mathtt{L}}}^{k+\tilde{\mathtt{L}}} \mathrm{e}^{\left(\gamma_i u_i - \frac{1}{2}\Lambda_i u_i^2\right)}}, \tag{39}$$

where the r.h.s. is already a Gaussian distribution and, thus, there is no need for further EP approximations. First, the two factors in the numerator are Gaussian distributions computed by the FWEP/BWEP algorithms:

$$q_f^{[k+\tilde{\mathtt{L}}]}(\mathbf{s}_{k+\tilde{\mathtt{L}}}) = \mathcal{N}\left(\mathbf{s}_{k+\tilde{\mathtt{L}}} : \boldsymbol{\mu}_f^{q,[k+\tilde{\mathtt{L}}]}, \boldsymbol{\Sigma}_f^{q,[k+\tilde{\mathtt{L}}]}\right), \tag{40}$$

$$q_b^{[k+\tilde{\mathtt{L}}]}(\mathbf{s}_{k+\tilde{\mathtt{L}}}) = \mathcal{N}\left(\mathbf{s}_{k+\tilde{\mathtt{L}}} : \boldsymbol{\mu}_b^{q,[k+\tilde{\mathtt{L}}]}, \boldsymbol{\Sigma}_b^{q,[k+\tilde{\mathtt{L}}]}\right). \tag{41}$$

Using the Gaussian channel likelihood, it is straightforward to show that the r.h.s. of (39) is a Gaussian distribution with covariance matrix

$$\begin{aligned}(\boldsymbol{\Sigma}_{fb})^{-1} = &\left(\boldsymbol{\Sigma}_f^{q,[k+\tilde{\mathtt{L}}]}\right)^{-1} + \left(\boldsymbol{\Sigma}_b^{q,[k+\tilde{\mathtt{L}}]}\right)^{-1} \\ &- \sigma_w^{-2}\mathbf{H}_{\mathtt{L}}^{\top}\mathbf{H}_{\mathtt{L}} - \mathrm{diag}(\boldsymbol{\Lambda}_{\mathbf{fb}}),\end{aligned} \tag{42}$$

and mean

$$\begin{aligned}\boldsymbol{\mu}_{fb} = \boldsymbol{\Sigma}_{fb}\Bigg[&\left(\boldsymbol{\Sigma}_f^{q,[k+\tilde{\mathtt{L}}]}\right)^{-1}\boldsymbol{\mu}_f^{q,[k+\tilde{\mathtt{L}}]} + \left(\boldsymbol{\Sigma}_b^{q,[k+\tilde{\mathtt{L}}]}\right)^{-1}\boldsymbol{\mu}_b^{q,[k+\tilde{\mathtt{L}}]} \\ &- \left(\sigma_w^{-2}\mathbf{H}_{\mathtt{L}}^{\top}\mathbf{y}_{k:k+\tilde{\mathtt{L}}} + \boldsymbol{\gamma}_{\mathbf{fb}}\right)\Bigg],\end{aligned} \tag{43}$$

where $\mathbf{H}_{\mathtt{L}}$ is a $\mathtt{L} \times (2\mathtt{L}-1)$ matrix formed by the $\mathtt{L}$ first rows of a circulant matrix with first row $\left[\mathbf{h}_{\mathtt{L}:1}{}^{\top}, \mathbf{0}_{1\times\tilde{\mathtt{L}}}\right]$ and

$$\boldsymbol{\Lambda}_{\mathbf{fb}} = \begin{bmatrix}\boldsymbol{\Lambda}_{\mathbf{b}k-\tilde{\mathtt{L}}:k} \\ \boldsymbol{\Lambda}_{\mathbf{f}k+1:k+\tilde{\mathtt{L}}}\end{bmatrix}, \quad \boldsymbol{\gamma}_{\mathbf{fb}} = \begin{bmatrix}\boldsymbol{\gamma}_{\mathbf{b}k-\tilde{\mathtt{L}}:k} \\ \boldsymbol{\gamma}_{\mathbf{f}k+1:k+\tilde{\mathtt{L}}}\end{bmatrix}. \tag{44}$$

Using the obtained Gaussian approximation to $p(\mathbf{s}_{k+\tilde{\mathtt{L}}}|\mathbf{y})$, we marginalize to obtain a probabilistic estimation to the posterior probability of the u_k symbol. The main steps of the SEP iterative algorithm are summarized in Algorithm 3. Since the SEP algorithm needs the forward and backward approximations computed by FWEP and BWEP, its final complexity is dominated by the computational cost of these algorithms, i.e., $\mathcal{O}(\mathtt{N}(S+1)(2\mathtt{L}-1)^3)$.

B. Windowed Linear MMSE and SEP turbo equalization

In the light of this result, it is interesting to point out that a windowed linear (recursive) MMSE can be easily derived from the formulation of the SEP by just running the SEP algorithm without entering in the EP iterative loop, i.e., forcing $S = 0$. We will denote this approach, the SEP with $S = 0$, as WLMMSE. Note that if no iterations of EP are performed we have $\gamma_i^{[k]} = \gamma_i^{[k-1]}$ and $\Lambda_i^{[k]} = \Lambda_i^{[k-1]}$. Accordingly, (20) can be reduced to

$$q_f^{[k]}(\mathbf{s}_k) = p(y_k|\mathbf{u}_{k-\tilde{\mathtt{L}}:k}) q_f^{[k-1]}(\tilde{\mathbf{s}}_k) \mathrm{e}^{\left(\gamma_k u_k - \frac{1}{2}\Lambda_k u_k^2\right)}, \tag{45}$$

where we removed the superindexes of γ_k and Λ_k. This expression, up to a constant, equals (17) particularized to the scenario where $p(u_k)$ is given and assumed to be Gaussian distributed, $p(u_k) \propto \exp\left(\gamma_k u_k - \frac{1}{2}\Lambda_k u_k^2\right)$, and we have a Gaussian likelihood that can be easily computed, $q_f^{[k-1]}(\tilde{\mathbf{s}}_k) = p(\tilde{\mathbf{s}}_k|\mathbf{y}_{1:k-1})$. Therefore, in this scenario the FWEP with $S = 0$ is a Kalman filtering and the SEP yields the smoothing version.

The performance of soft-output equalizers can be improved by using a turbo equalization approach [7], [19], [38]. The extrinsic information at the output of the decoder, $L_D(b_t)$, after mapping, $p_D(u_k)$, is handed back to the SEP equalizer. Then $p_D(u_k)$ is approximated by a Gaussian whose mean and variance are used to initialize the EP parameters, $(\gamma_i^{(1)}, \Lambda_i^{(1)})$, in Algorithm 1, when called in Algorithm 2. The extrinsic information handed to the decoder consists in the cavity functions (see Algorithm 1) at the end of the SEP algorithm, once demapped. This process can be repeated T times or until convergence.

Algorithm 3 The SEP estimator

1) Compute forward approximations with the FWEP algorithm detailed in Algorithm 2 with $\mathtt{W} = 2\tilde{\mathtt{L}} + 1$.
2) Compute backward approximations using Algorithm 2 as described in Section V with $\mathtt{W} = 2\tilde{\mathtt{L}} + 1$.
for $k = 1, ..., \mathtt{N}$ **do**
  Compute (42) and (43) and marginalize to estimate the APP $p(u_k|\mathbf{y})$.
end for

TABLE II: Complexity comparison between algorithms.

Algorithm	Complexity
SEP	$\mathtt{N}(S+1)(2\mathtt{L}-1)^3$
BEP	$(S+1)\mathtt{L}\mathtt{N}^2$
WLMMSE	$\mathtt{N}\mathtt{W}^2$
BCJR	$\mathtt{N}\mathtt{M}^{\mathtt{L}}$
Approx. BCJR	$\mathtt{N}M_e\mathtt{M}$

C. Computational Complexity

In Table II we include a detailed comparison of the complexity of the SEP, the algorithm in [31] (BEP), WLMMSE, BCJR, and approximated approaches, where M_e is the number of survivor states and $\mathtt{W}$ depends linearly with $\mathtt{L}$. As $\mathtt{M}$ and/or $\mathtt{L}$ grow, the BCJR and approximate approaches become computationally unaffordable. Due to the quadratic complexity of BEP in the frame length, the only feasible solutions for large $\mathtt{N}$ are the SEP and the WLMMSE, where the performance of the SEP quite outperforms that of the WLMMSE, as illustrated in the next section.

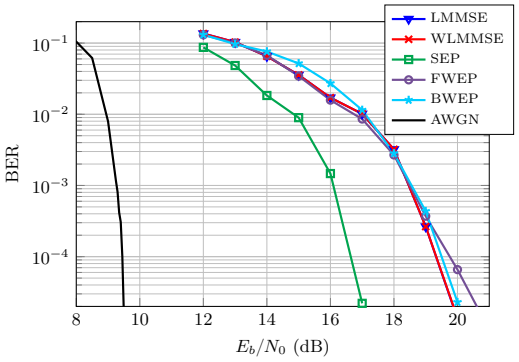


Fig. 4: BER for 64-QAM and 10 random channels with $\mathtt{L} = 7$.

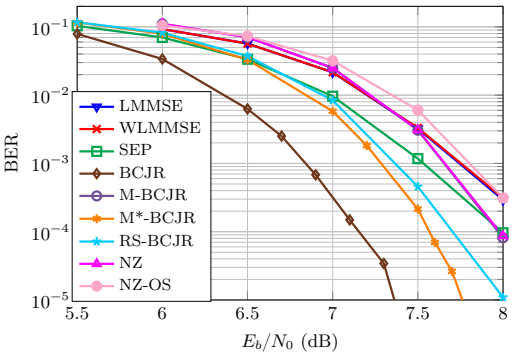


Fig. 5: BER for BPSK and a specific minimum phase channel $\mathbf{h} = \frac{1}{\sqrt{140}}[7\ 6\ 5\ 4\ 3\ 2\ 1]^\top$.

VII. Experimental results

In this section, we illustrate the good performance of the SEP equalizer for different scenarios. We use a regular LDPC of rate 1/2, with a maximum number of 200 iterations at the decoder. The codes used in this paper are (3,6)-regular LDPC generated using the progressive-edge growth (PEG) algorithm [39], and the belief propagation was used as channel decoder [40], [41]. The PEG algorithm ensures the largest possible girth according to the design parameters, i.e. the block length $\mathtt{N}$ and the degree distribution. We limit to 5 the maximum absolute value of the LLR at its input in order to avoid very confident probabilities which negatively affect the decoder. We average the BER over 1000 different frames per channel realization. When random channels are used, each tap is i.i.d. Gaussian distributed, circular when complex, with zero mean and equal variance, $1/\mathtt{L}$. The channel response is normalized. In all experiments in this paper we set $\mathtt{W} = 2\mathtt{L} - 1$, $\beta = 0.1$, $\epsilon = 0.5$ and $\sigma^2_{min} = 1e^{-3}$. $S = 10$ except for the case after the turbo procedure has started, where $S = 1$. These values were selected after extensive experimentation as a trade off between convergence speed and accuracy. In the BER curves, we include the block implementation of the LMMSE algorithm as reference. We also show the WLMMSE performance in the first figures to illustrate that it exhibits the same performance as the block implementation. For low complexity scenarios, the BCJR and its approximated solutions, the M-BCJR [11], M*-BCJR [14], RS-BCJR [10], NZ and NZ-OS [12] approaches are also included. The AWGN bound is also added in some of the figures as reference.

A. FWEP and BWEP

In Fig. 4 we illustrate the performance of the FWEP and the BWEP for $T = 0$. We depict the BER average for $\hat{m}_i$, see Fig. 1, over 10 random complex-valued channels for $\mathtt{L} = 7$, a 64-QAM modulation and codewords of 4098 bits. We also include the performance of the SEP, merging the estimation from both the FWEP and BWEP approaches, i.e. exploiting the whole vector of observations. Note that the LMMSE/WLMMSE need the whole observation vector to achieve a similar performance to the one of the FWEP (or the BWEP).

B. Low complexity scenario

In Fig. 5 and Fig. 6 we study the performance for typical channels used in the literature [12] to compare approximations to the BCJR algorithm. We use codewords of block length 1024 bits. In Fig. 5 we consider the minimum phase channel, $\mathbf{h} = \frac{1}{\sqrt{140}}[7\ 6\ 5\ 4\ 3\ 2\ 1]^\top$ and BPSK symbols. The BCJR has 64 states per step, while its approximations are computed with $M_e = 4$ states, as described in [12][2]. In Fig. 6 we simulate the maximum phase channel $\mathbf{h} = \frac{1}{\sqrt{140}}[1\ 2\ 3\ 4\ 5\ 6\ 7]^\top$ and BPSK symbols. Now, we use $M_e = 8$ states as in [12] for the approximated solutions. In Fig. 7 and Fig. 8 we average the BER over 100 random channels with 5 real-valued taps. We set the number of states for the approximated solutions to $M_e = 8$. In Fig. 7, we consider BPSK modulation, the BCJR has 16 states per step, while in Fig. 8 a 4-PAM is used, increasing the number of states to 256 in the BCJR algorithm.

In view of these results we may conclude that many algorithms approximating the optimal solution in the literature are quite dependent on the channel realization. Not only because they are based in just a forward or a backward strategy, but also because the parameters for some approximations need to be tuned for the particular channel. In this sense, the approximations M-BCJR, M*-BCJR and RS-BCJR fail, as observed in Fig. 6. On the other hand, we analyze how approximations based on tracking just a subset of paths or states, M_e, behave as the number of states grows. If the number of subsets, M_e, is kept as a fraction of the total number, they may exhibit a good performance. But if a large number of states is unaffordable, a fraction of it is also intractable. In the first cases, Fig. 5, Fig. 6 and Fig. 7, both NZ and NZ-OS algorithms have quite good performance because the state reduction ratio ($\mathtt{M}^\mathtt{L}/M_e$) is low. However, when we increase the number of states keeping the

[2]The number of states in approximated solutions are chosen according to the ones in [12]. We run FT, BT or DT depending on the phase of the channel and denote all of them in the results by NZ (and NZ-OS).

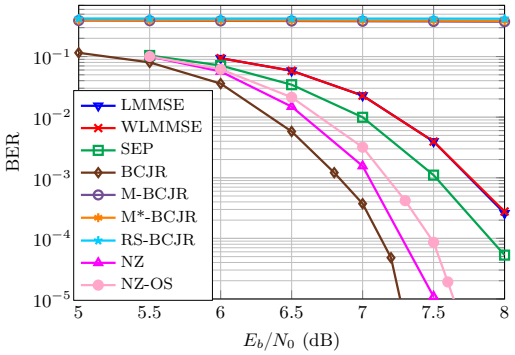


Fig. 6: BER for BPSK and a specific maximum phase channel $\mathbf{h} = \frac{1}{\sqrt{140}}[1\ 2\ 3\ 4\ 5\ 6\ 7]^\top$.

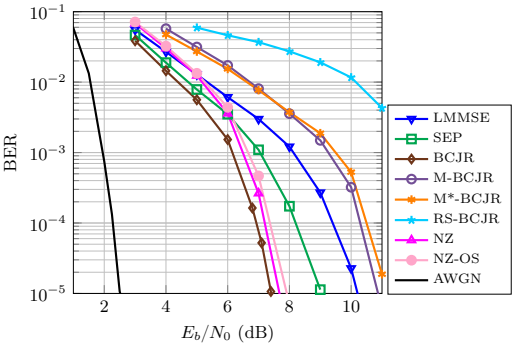


Fig. 7: BER for BPSK and 100 random channels with $\mathtt{L} = 5$.

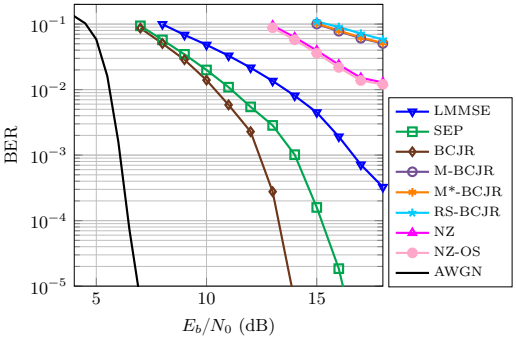


Fig. 8: BER for 4-PAM and 100 random channels with $\mathtt{L} = 5$.

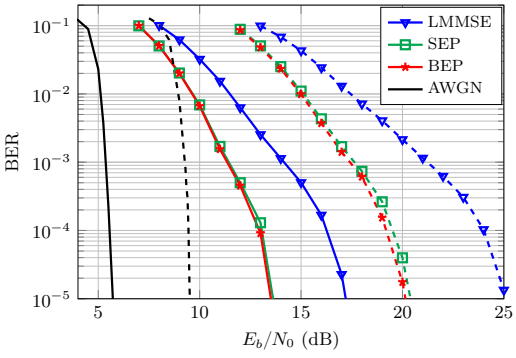


Fig. 9: BER after the channel decoder for 16-QAM (solid lines) and 64-QAM (dashed lines), 500 random channels with $\mathtt{L} = 7$ and codewords of 4096 bits.

number M_e fixed, the solution degrades, as can be observed in Fig. 8. The performance of the SEP is close to the one of the BCJR, within 0.75-2 dB in the four scenarios, regardless of the channel. The LMMSE also exhibits a robust behavior, but it quite deviates as the number of states grows. Note that the BCJR algorithm is computationally affordable in these scenarios, but its complexity increases exponentially with the dimension of the scenario, similarly to its approximations. Out of the approaches compared in these experiments, just the SEP and the LMMSE soft equalizers are available for large constellations. In the following, we study their performance.

C. Large complexity scenario

In Fig. 9, we depict the BER curves after the LDPC decoder for SEP, BEP [31] and LMMSE soft equalization, considering codewords of 4096 bits, channels of $\mathtt{L} = 7$ complex-valued taps with 16-QAM modulation (solid lines) and 64-QAM modulation (dashed lines). By including the BEP approach, we show that the SEP approach reduces the complexity of our previous work, with no degradation in the performance. The SEP exhibits a good performance compared to the LMMSE. For 16-QAM we have a gain of 3 dB, while for 64-QAM we improve it in 5 dB.

D. Performance analysis

We compare probabilistic equalizers using a information-theoretical metric: the mutual information between the transmitted symbol u_k and $\hat{u}_k$, which is a random variable distributed according to the estimation to the posterior distribution of u_k given $\mathbf{y}$ (the observation for the whole sequence):

$$I(u_k, \hat{u}_k) = \sum_{u_k} \sum_{\hat{u}_k} p(u_k, \hat{u}_k) \log_2 \frac{p(\hat{u}_k|u_k)}{p(\hat{u}_k)}, \qquad (46)$$

where note that $p(u_k, \hat{u}_k)$ is the joint probability distribution of u_k and $\hat{u}_k$ after marginalizing the channel output $\mathbf{y}$, the channel impulse response (in the case we consider it random) and the rest of symbols in the sequence $\mathbf{u}$. As the above expression cannot be evaluated in closed-form, we resort to Monte Carlo estimates to generate samples from $p(u_k, \hat{u}_k)$ and then evaluate $I(u_k, \hat{u}_k)$ from these samples. Assume that the channel taps are Gaussian distributed, but are perfectly known at the receiver. First, we collect $N \in \mathbb{Z}_+$ samples from the joint

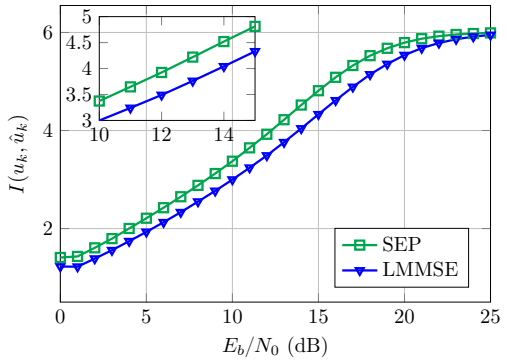


Fig. 10: Mutual information for SEP and LMMSE for 64-QAM and random channels with $\mathtt{L} = 7$.

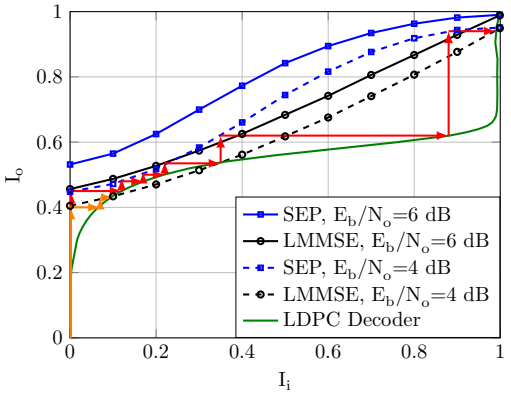


Fig. 11: EXIT charts for the decoder, the SEP and LMMSE equalizers with 4 and 6 dB of E_b/N_o. BPSK modulation was used.

distribution of $\mathbf{u}, \mathbf{h}, \mathbf{y}$ and $\hat{u}_k$ using *ancestral sampling* [36]. Each sample of the joint distribution is obtained as follows:

1) Sample $\mathbf{u}$ from a uniform distribution in the constellation.
2) Sample the channel coefficients $\mathbf{h}$.
3) Sample $\mathbf{y}$ according to $p(\mathbf{y}|\mathbf{u}, \mathbf{h})$.
4) Sample $\hat{u}_k$ according to $p(\hat{u}_k|\mathbf{y})$.

Second, after N samples of possible $(u_k, \hat{u}_k)$ pairs have been collected, we use these samples to estimate $p(\hat{u}_k)$, $p(\hat{u}_k|u_k)$ and, finally, $I(u_k, \hat{u}_k)$ in (46). It is important to observe that $p(\hat{u}_k|\mathbf{y})$ is dependent on the probabilistic detection method considered. The higher $I(u_k, \hat{u}_k)$ is for each probabilistic equalizer, the closer we perform to channel capacity. In Fig. 10 we depict the mutual information in (46) computed for $N = 10^6$ samples per E_b/N_0 point and averaged over all the symbols in the transmitted sequence, for the SEP and LMMSE algorithms, considering a 64-QAM constellation and channels of $\mathtt{L} = 7$ complex-valued taps. A magnification for intermediate E_b/N_0 is shown in the upper left corner. Observe that, for very small E_b/N_0, the noise is so large that both methods achieve a small mutual information. On the other hand, for very large E_b/N_0 values, both methods converge to 6 bits, i.e., the number of bits transmitted per QAM symbol. Any probabilistic method will eventually saturate to this value. However, the key aspect is to be able to design a probabilistic equalizer able to improve the mutual information for intermediate E_b/N_0 values. Observe that before the saturation, the SEP achieves a gain w.r.t. LMMSE of around 2 dB.

E. EXIT Charts and Turbo Equalization

The performance of the SEP equalization and turbo equalization can be analyzed with extrinsic information transfer (EXIT) charts [19], [42], [43]. The EXIT charts for the equalizer represent the improvement of the mutual information between the estimation at the output and the true symbols. By also including the EXIT chart of the channel decoder, we can predict the evolution of the interchange of extrinsic information from the equalizer to the channel decoder and back to the equalizer in turbo equalization.

In Fig. 11 we include the EXIT charts of the SEP (▫) and LMMSE (○) equalizers for the channel response $\mathbf{h} = [0.227\ 0.46\ 0.688\ 0.46\ 0.227]^\top$ [19], BPSK transmission and $E_b/N_o = 6$ (solid) and 4 dB (dashed). The EXIT chart of the LDPC decoder with $\mathtt{N} = 1024$ used is also depicted (solid). The horizontal axis is the mutual information of the LLR at the input, I_i. The vertical axis is the same measure for the output, I_o. The evolution of the mutual information for $E_b/N_o = 4$ dB along iterations of the SEP and LMMSE turbo equalization are indicated with arrows. An arrow is included each time the equalizer (vertical arrow) or the channel decoder (horizontal arrow) is run. Note that I_o at the output of the equalizer is also the value at the input of the channel decoder. In the figure, it is shown that the trajectory of the SEP turbo approach is able to successfully equalize and decode. In the LMMSE, at the second iteration of the turbo scheme the trajectory gets stuck, at $I_i \approx 0.1$, where the curve of the LMMSE intersects the curve of the channel decoder. Since the EXIT chart for the SEP and $E_b/N_o = 4$ dB is close to the one of the LMMSE for $E_b/N_o = 6$ dB, a 2 dB gain is expected in this scenario.

In Fig. 12 we depict the BER curves after the LDPC decoder for SEP turbo equalization, considering channels of $\mathtt{L} = 7$ complex-valued taps, 64-QAM modulation and codewords of $\mathtt{N} = 1024$ bits. The number of EP iterations once the turbo procedure starts is set to $S = 2$. Due to the iterative procedure of the feedback scheme, we reduced the maximum number of iterations in the LDPC decoder to 100. We represent the first 2 iterations and fifth iteration of the turbo scheme. It can be observed that the SEP without turbo already finds a very good estimate, still outperforming in 1.75 dB at a BER of 10^{-4} the performance of the LMMSE turbo equalizer after 5 loops. Therefore, the SEP receiver brings a remarkable gain in performance, but also a drastic latency reduction, since no feedback loop is required at least to improve the turbo-LMMSE scheme. In any case, if SEP is incorporated into the turbo receiver, we obtain an extra gain of about 0.75 dB after the first turbo iteration, which can be improved in 0.25 dB in

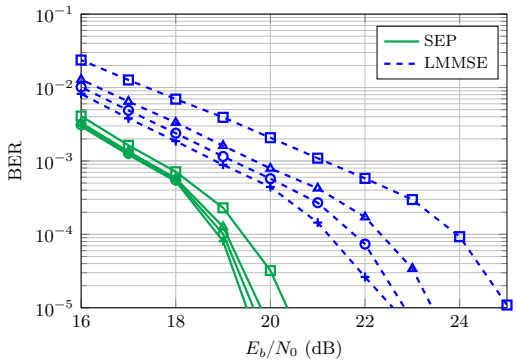


Fig. 12: BER for SEP (green lines, solid) and LMMSE (blue lines, dashed) using the feedback loop for 64-QAM, 500 random channels with $\mathtt{L} = 7$ and codewords of $\mathtt{v} = 4096$ bits. No feedback (▫), one loop (▵), two loops (○) and five loops (+).

successive turbo iterations.

VIII. Conclusion

In large dimension scenarios, optimal turbo and soft channel equalization become intractable. In particular, large constellations cannot be addressed with the BCJR solution or their approximations. They need to compute and store a huge number of states, and the performance of the approximations is quite dependent on the channel realizations. Classical methods such as LMMSE equalization can be used as an affordable alternative, with poorer performance. Its performance can be improved with turbo equalization. The complexity is linear in $\mathtt{N}$ and quadratic in the length of the window used. It can be reduced by using approximations.

In this paper, we propose the SEP as novel EP based soft equalizer with the following advantages. Firstly, the order of its computational complexity is linear with the frame length, $\mathtt{N}$. Secondly, this new solution outperforms the LMMSE equalizer in terms of BER. Finally, based on the results of the experiments included, we get improvements with respect to the LMMSE in the range 2-5 dB exhibiting a good performance regardless of the channel response.

Appendix A
Backward Windowed EP Equalizer

In this appendix we introduce the change of variables that allows to run the BWEP algorithm as a particular instance of FWEP. Define

$$u_i^{\leftarrow} = u_{\mathtt{N}+1-i} \quad i = -\mathtt{W}+2, ..., \mathtt{N}+\mathtt{W}-1 \tag{47}$$
$$\mathbf{s}_k^{\leftarrow} = \mathbf{u}^{\leftarrow}{}_{k-\mathtt{W}+1:k} \quad k = 1, ..., \mathtt{N}+\mathtt{W}-1 \tag{48}$$
$$y_k^{\leftarrow} = y_{\mathtt{N}+\mathtt{W}-\mathtt{L}+1-k} \quad k = 1, ..., \mathtt{N}+\mathtt{W}-1 \tag{49}$$
$$h_l^{\leftarrow} = h_{\mathtt{L}+1-l} \quad l = 1, ..., \mathtt{L}. \tag{50}$$

These definitions have been proposed to allow using (20). We obtain the backward approximated distribution for $k = 1, ..., \mathtt{N}+\mathtt{W}-1$, by applying the FWEP with the vectors in (47)-(50),

$$q'^{[k]}_b(\mathbf{s}_k^{\leftarrow}) \overset{\text{def}}{=} q_f^{[k]}(\mathbf{s}_k^{\leftarrow}) =$$
$$= p(y_k^{\leftarrow}|\mathbf{u}^{\leftarrow}{}_{k-\tilde{\mathtt{L}}:k}) g^{[k-1]}(\tilde{\mathbf{s}}_k^{\leftarrow}) \prod_{i=k-\mathtt{W}+1}^{k} \mathrm{e}^{\left(\gamma_i u_i^{\leftarrow} - \frac{1}{2}\Lambda_i u_i^{\leftarrow 2}\right)}. \tag{51}$$

To later ease the combination of the forward and backward algorithms we propose to change the indexing, and define $q_b^{[k]}(\mathbf{s}_k) \overset{\text{def}}{=} q'^{[\mathtt{N}+\mathtt{W}-k]}_b(\mathbf{s}^{\leftarrow}_{\mathtt{N}+\mathtt{W}-k})$. By using this change of variable in the superscript, in the first iteration we estimate $q_b^{[\mathtt{N}+\mathtt{W}-1]}(\mathbf{u}_{\mathtt{N}+\mathtt{W}-1:\mathtt{N}})$ while in the last one $q_b^{[1]}(\mathbf{u}_{1:-\mathtt{W}+2})$, as described in Fig. 3.

References

[1] R. Schober, "Detection and estimation of signals in noise," University of British Columbia, Vancouver, Tech. Rep., 2010.
[2] J. G. Proakis, *Digital Communications*, 5th ed. New York, NY: McGraw-Hill, 2008.
[3] S. Haykin, *Communication Systems*, 5th ed. Wiley Publishing, 2009.
[4] L. Salamanca, J. J. Murillo-Fuentes, and F. Pérez-Cruz, "Bayesian equalization for LDPC channel decoding," *IEEE Trans. on Signal Processing*, vol. 60, no. 5, pp. 2672–2676, May 2012.
[5] C. Douillard, M. Jezequel, C. Berrou, P. Didier, and A. Picart, "Iterative correction of intersymbol interference: turbo-equalization," *European Trans. on Telecommunications*, vol. 6, no. 5, pp. 507–512, Sep 1995.
[6] C. Berrou, *Codes and turbo codes*, ser. Collection IRIS. Springer Paris, 2010.
[7] M. Tüchler and A. Singer, "Turbo equalization: An overview," *IEEE Trans. on Information Theory*, vol. 57, no. 2, pp. 920–952, Feb 2011.
[8] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Trans. on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.
[9] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 498 –519, Feb 2001.
[10] G. Colavolpe, G. Ferrari, and R. Raheli, "Reduced-state BCJR-type algorithms," *IEEE Journal on Sel. Areas in Communications*, vol. 19, no. 5, pp. 848–859, May 2001.
[11] V. Franz and J. Anderson, "Concatenated decoding with a reduced-search BCJR algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 186–195, Feb 1998.
[12] D. Fertonani, A. Barbieri, and G. Colavolpe, "Reduced-complexity BCJR algorithm for turbo equalization," *IEEE Trans. on Communications*, vol. 55, no. 12, pp. 2279–2287, Dec 2007.
[13] C. Vithanage, C. Andrieu, and R. Piechocki, "Approximate inference in hidden Markov models using iterative active state selection," *IEEE Signal Processing Letters*, vol. 13, no. 2, pp. 65–68, Feb 2006.
[14] M. Sikora and D. Costello, "A new SISO algorithm with application to turbo equalization," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Sep 2005, pp. 2031–2035.
[15] M. Eyuboglu and S. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback," *IEEE Trans. on Communications*, vol. 36, no. 1, pp. 13–20, Jan 1988.
[16] A. Duel-Hallen and C. Heegard, "Delayed decision-feedback sequence estimation," *IEEE Trans. on Communications*, vol. 37, no. 5, pp. 428–436, May 1989.
[17] P. Chevillat and E. Eleftheriou, "Decoding of trellis-encoded signals in the presence of intersymbol interference and noise," *IEEE Trans. on Communications*, vol. 37, no. 7, pp. 669–676, Jul 1989.
[18] K. Muranov, "Survey of MMSE channel equalizers," University of Illinois, Chicago, Tech. Rep.
[19] M. Tüchler, R. Koetter, and A. Singer, "Turbo equalization: principles and new results," *IEEE Trans. on Communications*, vol. 50, no. 5, pp. 754–767, May 2002.
[20] M. Tüchler, A. Singer, and R. Koetter, "Minimum mean squared error equalization using a priori information," *IEEE Trans. on Signal Processing*, vol. 50, no. 3, pp. 673–683, Mar 2002.
[21] R. Drost and A. Singer, "Factor-graph algorithms for equalization," *IEEE Trans. on Signal Processing*, vol. 55, no. 5, pp. 2052–2065, May 2007.

[22] L. Liu and L. Ping, "An extending window MMSE turbo equalization algorithm," *IEEE Signal Processing Letters*, vol. 11, no. 11, pp. 891–894, Nov 2004.
[23] T. Minka, "Expectation propagation for approximate Bayesian inference," in *Proc. 17th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2001, pp. 362–369.
[24] M. Seeger, "Expectation propagation for exponential families," *Univ. Calif., Berkeley, CA, USA, Tech. Rep.*, 2005.
[25] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Pérez-Cruz, "Expectation propagation detection for high-order high-dimensional MIMO systems," *IEEE Trans. on Communications*, vol. 62, no. 8, pp. 2840–2849, Aug 2014.
[26] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Pérez-Cruz, "Improved performance of LDPC-coded MIMO systems with EP-based soft-decisions," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Jun 2014, pp. 1997–2001.
[27] P. M. Olmos, J. J. Murillo-Fuentes, and F. Pérez-Cruz, "Tree-structure expectation propagation for LDPC decoding over the BEC," *IEEE Trans. on Information Theory*, vol. 59, no. 6, pp. 3354–3377, 2013.
[28] L. Salamanca, P. M. Olmos, F. Pérez-Cruz, and J. J. Murillo-Fuentes, "Tree-structured expectation propagation for LDPC decoding over BMS channels," *IEEE Trans. on Communications*, vol. 61, no. 10, pp. 4086–4095, Oct 2013.
[29] Y. Qi and T. Minka, "Window-based expectation propagation for adaptive signal detection in flat-fading channels," *IEEE Trans. on Wireless Communications*, vol. 6, no. 1, pp. 348–355, Jan 2007.
[30] P. Sun, C. Zhang, Z. Wang, C. Manchon, and B. Fleury, "Iterative receiver design for ISI channels using combined belief- and expectation-propagation," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1733–1737, Oct 2015.
[31] I. Santos, J. J. Murillo-Fuentes, and P. M. Olmos, "Block expectation propagation equalization for ISI channels," in *Proc. 23rd European Signal Processing Conference (EUSIPCO)*, Sep 2015, pp. 379–383.
[32] R. R. Lopes and J. R. Barry, "The soft-feedback equalizer for turbo equalization of highly dispersive channels," *IEEE Trans. on Communications*, vol. 54, no. 5, pp. 783–788, May 2006.
[33] J. Jiang, C. He, E. M. Kurtas, and K. R. Narayanan, "Performance of soft feedback equalization over magnetic recording channels," in *Proc. IEEE International Magnetics Conference (INTERMAG)*, May 2006, p. 795.
[34] J. Tao, "On low-complexity soft-input soft-output decision-feedback equalizers," *IEEE Communications Letters*, vol. 20, no. 9, pp. 1737–1740, Sep 2016.
[35] T. P. Minka, "A family of algorithms for approximate Bayesian inference," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.
[36] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag, New York, 2006.
[37] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2002.
[38] R. Koetter, A. Singer, and M. Tüchler, "Turbo equalization," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 67–80, Jan 2004.
[39] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. on Information Theory*, vol. 51, no. 1, pp. 386–398, 2005.
[40] T. J. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, Mar. 2008.
[41] L. Salamanca, J. J. Murillo-Fuentes, P. M. Olmos, F. Pérez-Cruz, and S. Verdú, "Approaching the DT bound using linear codes in the short blocklength regime," *IEEE Communications Letters*, vol. 19, no. 2, pp. 123–126, Feb 2015.
[42] S. Ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE transactions on communications*, vol. 49, no. 10, pp. 1727–1737, 2001.
[43] J. Choi, *Adaptive and Iterative Signal Processing in Communications*. Cambdirge University Press, 2006.

Appendix D
Paper IV

Authors: Irene Santos and Juan José Murillo-Fuentes.
Title: EP-based turbo detection for MIMO receivers and large-scale systems.
Journal: IEEE Transactions on Vehicular Technology
State: Submitted.
Abstract: We investigate a turbo soft detector based on the expectation propagation (EP) algorithm for large-scale multiple- input multiple-output (MIMO) systems. Optimal detection in MIMO systems becomes computationally unfeasible for high-order modulations and/or large number of antennas. In this situation, the low-complexity linear minimum-mean squared- error (LMMSE) is quite employed but its performance is far from the optimal. To improve the performance, the EP algorithm can be used. In this paper, we review previous EP-based turbo detectors and improve their estimation in terms of complexity and performance. Specifically, we replace the uniform prior used by previous approaches by a non-uniform one, which better characterizes the information returned by the decoder once the turbo procedure starts. We also review the EP parameters to avoid instabilities when using high-order modulations and to reduce the complexity. Simulations results are included to show the robustness and improvement in performance of our proposed detector in comparison with previous approaches found in the literature.

EP-based turbo detection for MIMO receivers and large-scale systems

Irene Santos and Juan José Murillo-Fuentes

***Abstract*—We investigate a turbo soft detector based on the expectation propagation (EP) algorithm for large-scale multiple-input multiple-output (MIMO) systems. Optimal detection in MIMO systems becomes computationally unfeasible for high-order modulations and/or large number of antennas. In this situation, the low-complexity linear minimum-mean squared-error (LMMSE) is quite employed but its performance is far from the optimal. To improve the performance, the EP algorithm can be used. In this paper, we review previous EP-based turbo detectors and improve their estimation in terms of complexity and performance. Specifically, we replace the uniform prior used by previous approaches by a non-uniform one, which better characterizes the information returned by the decoder once the turbo procedure starts. We also review the EP parameters to avoid instabilities when using high-order modulations and to reduce the complexity. Simulations results are included to show the robustness and improvement in performance of our proposed detector in comparison with previous approaches found in the literature.**

***Index Terms*—Expectation propagation (EP), MMSE, low-complexity, MIMO, turbo detection, feedback**

I. Introduction

The use of MIMO systems is of great interest nowadays due to the need of transmitting at high rates. They are more spectrally efficient than the traditional single-input single-output (SISO) systems and provide higher channel capacity [1]. Once the symbols are sent over the transmit antennas, the received signal is processed in the detector to obtain an estimation of the transmitted symbols. This estimation can be probabilistic, resulting in a high benefit for modern channel decoders [2]. In addition, the performance can be improved with a turbo-detection scheme, i.e., by exchanging information between the decoder and the soft detector iteratively.

The optimal algorithm to perform soft detection is the maximum a posteriori (MAP) algorithm that finds the transmitted word that maximices the a posteriori probabilities (APP). However, its computational complexity is intractable for high order constellations and/or large number of transmitting antennas. In this situation, non-optimal approaches are employed. The sphere decoding (SD) method provides an approximated marginal posterior probability density function (pdf) in a subspace of the whole set of possible transmitted words given by the constellation [3], [4]. However, its performance deteriorates if the dimension of the subspace does not grow accordingly with the constellation size and the number of antennas. Hence it is computationally unfeasible for large scale scenarios. Another alternative to approximate the posterior distribution is the use of Markov chain Monte Carlo (MCMC) algorithms [5], but it also requires a sufficiently large number of samples to obtain an accurate enough performance.

In this situation of large complexity scenarios, the LMMSE is commonly employed due to its low complexity. It computes an approximated APP by minimizing the mean-square-error (MSE) between the transmitted and detected symbol. Since its performance is far from the optimal, alternative approaches can be found in the literature. The Gaussian tree approximation (GTA) algorithm [6] constructs a tree-factorized Gaussian approximation to the APP and then estimate the marginals with the belief propagation (BP) algorithm. The channel hardening-exploiting message passing (CHEMP) [7] is a message-passing algorithm where all the exchanged messages are approximated by Gaussian distributions.

The above approximated approaches assume a Gaussian prior for the transmitted symbols, i.e., they do not consider the discrete nature of symbols when estimating the APP. This restriction can be included by means of the EP algorithm [8], [9], that obtains a Gaussian approximated posterior pdf by matching its moments with the ones of the true APP. This algorithm has been already applied to equalization [10]–[12] and detection in flat-fading channels [13], [14]. It has been also applied to MIMO detection, initially for hard detection [15] and then extended for soft detection [16], [17]. In these works, it was shown that the EP detector was able to improve the performance of LMMSE, GTA and CHEMP with complexity proportional to the LMMSE algorithm. Preliminary results on turbo detection can be found in [18], [19].

In this paper, we focus on turbo detection based on the EP algorithm, improving the prior information used in the detection stage once the turbo procedure starts. Rather than consider uniform priors during the moment matching procedure in the EP algorithm as proposed in [18], we force the true discrete prior to be non-uniform distributed according to the output of the channel decoder, similarly to [19]. Since a non-uniform prior for the symbols better characterizes the information returned by the decoder once the turbo procedure starts, we obtain large improvements in terms of performance. We also optimize the EP parameters with a double purpose: avoid instabilities that appears in high-order modulations and reduce the computational complexity of the algorithm once the turbo procedure starts, following the guidelines in [12] and improving the results in [19].

The paper is organized as follows. In Section II we describe the structure of the communication system. The proposed EP detector is explained in Section III, where we also detail the

I. Santos and J.J. Murillo-Fuentes are with the Dept. Teoría de la Señal y Comunicaciones, Universidad de Sevilla, Camino de los Descubrimiento s/n, 41092 Sevilla, Spain. E-mail: {irenesantos,murillo}@us.es

This work was partially funded by Spanish government (Ministerio de Economía y Competitividad TEC2016-78434-C3-R) and by the European Union (FEDER).

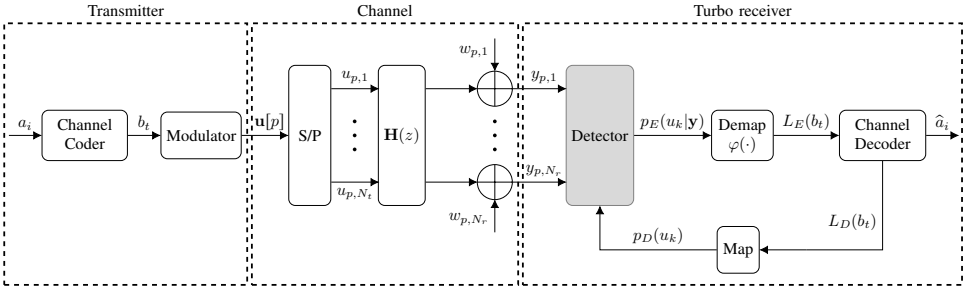


Fig. 1: System model for a MIMO architecture.

parameters used. Some simulations results are included in Section IV. Finally, we end with conclusions.

II. System Model

The model of the communication system is included in Fig. 1 where N_t transmit antennas communicate to a receiver with N_r antennas. A turbo architecture of a soft MIMO detector is proposed. It can be divided into three parts: transmitter, channel model and turbo receiver.

A. Transmitter

The information bit vector, $\mathbf{a} = [a_1, ..., a_K]^\top$, is encoded into the codeword $\mathbf{b} = [b_1, ..., b_V]^\top$ with a code rate $R = K/V$. The codeword is partitioned into $\mathtt{N} = \lceil V/\log_2 \mathtt{M} \rceil$ blocks, $\mathbf{b} = [\mathbf{b}_1, ..., \mathbf{b}_\mathtt{N}]^\top$, where $\mathbf{b}_k = [b_{k,1}, ..., c_{k,Q}]$ and $Q = \log_2(\mathtt{M})$. This codeword is sent to the $\mathtt{M}$-ary modulator, that obtains the block frames $\mathbf{u} = [u_1, ..., u_\mathtt{N}]^\top$, where each component $u_k = \mathcal{R}(u_k) + j\mathcal{I}(u_k) \in \mathcal{A}$. Hereafter, $\mathcal{A}$ denotes the set of symbols of the constellation of order $|\mathcal{A}| = \mathtt{M}$. These symbols are partitioned into $\mathtt{P}$ blocks of length N_t, $\mathbf{u} = [\mathbf{u}[1], ..., \mathbf{u}[\mathtt{P}]]^\top$, where $\mathbf{u}[p] = [u_{p,1}, ..., u_{p,N_t}]$. Each block is demultiplexed into N_t substreams through the serial to paralell (S/P) converter. Then, the block frames are transmitted over the channel.

B. Channel model

The channel is completely specified by the weights between each transmitting and receiving antenna, $h_{k,j}$ where $k = 1, ..., N_r$ and $j = 1, ..., N_t$, with $N_r \geqslant N_t$, and the known noise variance, σ_w^2. The received signal $\mathbf{y} = [\mathbf{y}[1], ..., \mathbf{y}[\mathtt{P}]]^\top$, where $\mathbf{y}[p] = [y_{p,1}, ..., y_{p,N_r}]$, is given by

$$\mathbf{y}[p] = \mathbf{H}\mathbf{u}[p] + \mathbf{w}[p] \tag{1}$$

where

$$\mathbf{H} = \begin{bmatrix} h_{1,1} & \dots & h_{1,N_t} \\ h_{2,1} & \ddots & h_{2,N_t} \\ \vdots & \ddots & \vdots \\ h_{N_r,1} & \dots & h_{N_r,N_t} \end{bmatrix}, \tag{2}$$

and $\mathbf{w}[p] \sim \mathcal{N}\left(\mathbf{w}[p] : \mathbf{0}, \sigma_w^2\mathbf{I}\right)$ is a complex-valued additive white Gaussian noise (AWGN) vector. The channel matrix, $\mathbf{H}$, in (1) could evolve with time but, for simplicity, we will assume that the system is linear time invariant (LTI).

C. Turbo receiver

The posterior probability of the transmitted symbol vector $\mathbf{u}[p]$ given the whole vector of observations $\mathbf{y}[p]$ yields

$$\begin{aligned} p(\mathbf{u}[p]|\mathbf{y}[p]) &= \frac{p(\mathbf{y}[p]|\mathbf{u}[p])p(\mathbf{u}[p])}{p(\mathbf{y}[p])} \\ &\propto \mathcal{CN}\big(\mathbf{y}[p] : \mathbf{H}\mathbf{u}[p], \sigma_w^2\mathbf{I}\big) \prod_{k=1}^{N_t} p_D(u_{p,k}), \end{aligned} \tag{3}$$

where the true prior returned by the decoder, $p_D(u_{p,k})$, is clearly non-Gaussian, but a non-uniform discrete distribution. If no information is available from the decoder, then the true prior is assumed to be equiprobable. In the following, the index p will be omitted to simplify the notation and ease the reading.

The extrinsic distribution computed by the detector is demapped and given to the decoder as extrinsic log-likelihood ratios,

$$L_E(b_{k,j}) = \log \frac{\sum_{u_k \in \mathcal{A}|b_{k,j}=0} p_E(u_k|\mathbf{y})}{\sum_{u_k \in \mathcal{A}|b_{k,j}=1} p_E(u_k|\mathbf{y})}. \tag{4}$$

The channel decoder computes an estimation of the information bit vector, $\hat{\mathbf{a}}$, and an extrinsic log-likelihood ratio (LLR) on the coded bits

$$L_D(b_t) = \log \frac{p(b_t = 0|L_E(\mathbf{b}))}{p(b_t = 1|L_E(\mathbf{b}))} - L_E(b_t). \tag{5}$$

These LLRs are again mapped and given to the detector as updated priors, $p_D(\mathbf{u})$. This process is repeated iteratively for a given maximum number of iterations, T, or until convergence.

III. The nuBEP detector

The EP algorithm provides a feasible approximation to the posterior distribution, $p(\mathbf{u}|\mathbf{y})$, in (3) by a Gaussian approximation, $q^{[\ell]}(\mathbf{u})$, where the non-Gaussian terms, $p_D(u_k)$, are replaced by Gaussians denoted as $\tilde{p}_D^{[\ell]}(u_k)$. This approximated posterior factorizes as

$$q^{[\ell]}(\mathbf{u}) \propto p(\mathbf{y}|\mathbf{u}) \prod_{k=1}^{N_t} \tilde{p}_D^{[\ell]}(u_k) = \tag{6}$$

$$= \mathcal{CN}\big(\mathbf{y} : \mathbf{H}\mathbf{u}, \sigma_w^2\mathbf{I}\big) \prod_{k=1}^{N_t} \mathcal{CN}\Big(u_k : m_k^{[\ell]}, \eta_k^{[\ell]}\Big) \tag{7}$$

and it is distributed according to a Gaussian given by

$$q^{[\ell]}(\mathbf{u}) \propto \mathcal{CN}\Big(\mathbf{u} : \boldsymbol{\mu}^{[\ell]}, \mathbf{S}^{[\ell]}\Big) \tag{8}$$

where

$$\mathbf{S}^{[\ell]} = \left(\sigma_w^{-2}\mathbf{H}^{\mathrm{H}}\mathbf{H} + \mathrm{diag}(\boldsymbol{\eta}^{[\ell]})^{-1}\right)^{-1}, \tag{9}$$

$$\boldsymbol{\mu}^{[\ell]} = \mathbf{S}^{[\ell]}\left(\sigma_w^{-2}\mathbf{H}^{\mathrm{H}}\mathbf{y} + \mathrm{diag}(\boldsymbol{\eta}^{[\ell]})^{-1}\mathbf{m}^{[\ell]}\right), \tag{10}$$

and $\boldsymbol{\eta}^{[\ell]} = [\eta_1^{[\ell]}, \ldots, \eta_{N_t}^{[\ell]}]^\top$, $\mathbf{m}^{[\ell]} = [m_1^{[\ell]}, \ldots, m_{N_t}^{[\ell]}]^\top$. Along S iterations, this algorithm obtains the value of the parameters $m_k^{[\ell]}$ and $\eta_k^{[\ell]}$ that allows the moments of the true posterior in (3) match the moments of its approximation in (8), i.e.,

$$\widehat{p}^{[\ell]}(u_k) = q_E^{[\ell]}(u_k)p_D(u_k) \quad \overset{\text{moment matching}}{\longleftrightarrow} \quad q_E^{[\ell]}(u_k)\tilde{p}_D^{[\ell+1]}(u_k) \tag{11}$$

where $q_E^{[\ell]}(u_k)$ is an extrinsic marginal distribution computed as

$$q_E^{[\ell]}(u_k) = \frac{q^{[\ell]}(u_k)}{\tilde{p}_D^{[\ell]}(u_k)} = \mathcal{CN}\Big(u_k : z_k^{[\ell]}, v_k^{2[\ell]}\Big), \tag{12}$$

$q^{[\ell]}(u_k) \sim \mathcal{CN}\Big(u_k : \mu_k^{[\ell]}, s_k^{2[\ell]}\Big)$ is the k-th marginal of (8) and

$$z_k^{[\ell]} = \frac{\mu_k^{[\ell]}\eta_k^{[\ell]} - m_k^{[\ell]}s_k^{2[\ell]}}{\eta_k^{[\ell]} - s_k^{2[\ell]}}, \tag{13}$$

$$v_k^{2[\ell]} = \frac{s_k^{2[\ell]}\eta_k^{[\ell]}}{\eta_k^{[\ell]} - s_k^{2[\ell]}}. \tag{14}$$

The moment matching procedure in (11) is detailed in Algorithm 1, where the damping procedure and the control of negative variances proposed in [16]–[18] is also included. The whole EP procedure for a turbo detector is detailed in Algorithm 2, where S is the number of EP iterations and T the number of turbo iterations. Unlike [16]–[18], this algorithm uses the non uniform probability mass function (pmf) of the output of the decoder as priors in the moment matching. For this reason, we denote this approach as non-uniform block expectation propagation (nuBEP) detector.

A. EP parameters

The update of the EP solution is a critical issue due to instabilities, particularly for high-order modulations. In this subsection, we review the EP parameters used in related approaches [17], [19] and explain the ones used in this approach. Following [17], these parameters are: the minimum allowed variance (ϵ) and a control of negative variances, a damping procedure (β) and the number of EP iterations (S). The first two parameters determine the speed of the algorithm when arriving to an stationary solution and control instabilities. The computational complexity of the algorithm depends linearly with S.

In [17], the authors set $S = 10$ and introduced fast updates of EP solution by setting $\beta = 0.95$. To avoid instabilities due to the fast updates, they set a gradual decrease for the minimum variance, setting it to a high value during the first 4 iterations and then decreasing it exponentially, i.e., $\epsilon = 2^{-\max(\ell-4,1)}$. However, we found that for large-size modulations and turbo schemes, the fast updates can provoque instabilities, as will be shown in Section IV. For this reason and following our proposal in [12], we let β start with a conservative value and increases it exponentially with the number of turbo iterations, $\beta = \min(\exp^{t/1.5}/10), 0.7)$, where $t \in [0, T]$ is the number of the current turbo iteration. This growth of β allows to reduce the number of EP iterations once the turbo procedure starts. We propose to reduce it from 10 in [17] to $S = 3$. We also set $\epsilon = 1e^{-8}$. Regarding the control of negative variances, we just update the EP solution when the computed variance is positive (see (20)), as proposed in [17].

On the other hand, in [19] they only compute one iteration of the EP procedure, i.e., they set $S = 1$. They did not introduce any damping or control of minimum variances. Regarding the control of negative variances, rather that keeping the previous estimate of the EP algorithm as in (20), they propose a different procedure: in case of negative variances they update the EP solution with the moments computed in step 3 of Algorithm 2, i.e.,

$$\eta_k^{[\ell+1]} = \sigma_{p_k}^{2[\ell]}, \quad m_k^{[\ell+1]} = \mu_{p_k}^{[\ell]}. \tag{23}$$

In Table I, we describe the values of the EP parameters used in the current proposal (nuBEP) and the other EP-based detectors in the literature.

Algorithm 1 Moment Matching and Damping (MMD)

Given inputs: $\mu_{p_k}^{[\ell]}, \sigma_{p_k}^{2[\ell]}, z_k^{[\ell]}, v_k^{2[\ell]}, m_k^{[\ell]}, \eta_k^{[\ell]}$

1) Run moment matching: Set the mean and variance of the unnormalized Gaussian distribution

$$q_E^{[\ell]}(u_k) \cdot \mathcal{CN}\Big(u_k : m_{k,new}^{[\ell+1]}, \eta_{k,new}^{[\ell+1]}\Big) \tag{15}$$

equal to $\mu_{p_k}^{[\ell]}$ and $\sigma_{p_k}^{2[\ell]}$, to get the solution

$$\eta_{k,new}^{[\ell+1]} = \frac{\sigma_{p_k}^{2[\ell]}v_k^{2[\ell]}}{v_k^{2[\ell]} - \sigma_{p_k}^{2[\ell]}}, \tag{16}$$

$$m_{k,new}^{[\ell+1]} = \eta_{k,new}^{[\ell+1]}\left(\frac{\mu_{p_k}^{[\ell]}}{\sigma_{p_k}^{2[\ell]}} - \frac{z_k^{[\ell]}}{v_k^{2[\ell]}}\right). \tag{17}$$

2) Run damping: Update the values as

$$\eta_k^{[\ell+1]} = \left(\beta\frac{1}{\eta_{k,new}^{[\ell+1]}} + (1-\beta)\frac{1}{\eta_k^{[\ell]}}\right)^{-1}, \tag{18}$$

$$m_k^{[\ell+1]} = \eta_k^{[\ell+1]}\left(\beta\frac{m_{k,new}^{[\ell+1]}}{\eta_{k,new}^{[\ell+1]}} + (1-\beta)\frac{m_k^{[\ell]}}{\eta_k^{[\ell]}}\right). \tag{19}$$

if $\eta_k^{[\ell+1]} < 0$ **then**

$$\eta_k^{[\ell+1]} = \eta_k^{[\ell]}, \quad m_k^{[\ell+1]} = m_k^{[\ell]}. \tag{20}$$

end if

Output: $\eta_k^{[\ell+1]}, m_k^{[\ell+1]}$

IV. Simulation Results

In this section we illustrate the performance of the proposed nuBEP turbo detector and compare its performance with

Algorithm 2 nuBEP Turbo Decoder for MIMO

Initialization: Set $p_D(u_k) = \frac{1}{\mathtt{M}} \sum_{u \in \mathcal{A}} \delta(u_k - u)$ for $k = 1, \ldots, N_t$

for $t = 1, ..., T$ **do**

1) Compute the mean $m_k^{[1]}$ and variance $\eta_k^{[1]}$ as

$$m_k^{[1]} = \sum_{u \in \mathcal{A}} u \cdot p_D(u_k = u), \quad (21)$$

$$\eta_k^{[1]} = \sum_{u \in \mathcal{A}} (u - m_k^{[1]})^*(u - m_k^{[1]}) \cdot p_D(u_k = u). \quad (22)$$

for $\ell = 1, ..., S$ **do**

for $k = 1, ..., N_t$ **do**

2) Compute the k-th extrinsic distribution, $q_E^{[\ell]}(u_k)$, as in (12).

3) Obtain the distribution $\widehat{p}^{[\ell]}(u_k) \propto q_E^{[\ell]}(u_k) p_D(u_k)$ and estimate its mean $\mu_{p_k}^{[\ell]}$ and variance $\sigma_{p_k}^{2[\ell]}$. Set a minimum allowed variance as $\sigma_{p_k}^{2[\ell]} = \max(\epsilon, \sigma_{p_k}^{2[\ell]})$.

4) Run the moment matching procedure in Algorithm 1 to obtain $m_k^{[\ell+1]}$ and $\eta_k^{[\ell+1]}$.

end for

end for

5) With the values $m_k^{[S+1]}, \eta_k^{[S+1]}$ obtained after the EP algorithm, calculate the extrinsic distribution $q_E^{[S+1]}(u_k)$ as in (12).

6) Demap the extrinsic distribution and compute the extrinsic LLR, $L_E(b_{k,j})$, as in (4).

7) Run the channel decoder to output $p_D(u_k)$.

end for

Output: Deliver $L_E(b_{k,j})$ to the channel decoder for $k = 1, \ldots, N_t$ and $j = 1, \ldots, Q$

Algorithm	ϵ	β	S
nuBEP	$1e^{-8}$	$\beta = \min(\exp^{t/1.5}/10), 0.7)$	3
EPD [17]	$2^{-\max(\ell-4,1)}$	0.95	10
EP [19]	-	-	1

TABLE I: Values for the EP parameters after the turbo procedure.

the EP-based detector proposed in [17] and [19], that we will named EPD and MPEP, respectively. We also show the LMMSE performance. We do not include the SD [3], MCMC [5], GTA [6] or CHEMP [7] algorithms in the simulations because it has already been shown that EPD [17] quite outperforms these three approaches [15], [18]. The modulator uses a Gray mapping and a 128-QAM constellation. The results are averaged over 100 random channels and $1e^4$ random encoded words of length 4096 (per channel realization). A number of $T = 5$ turbo iterations were run. Each channel tap is independent and identically distributed (IID) Gaussian distributed with zero mean and variance equal to $1/\mathrm{L}$. The absolute value of LLRs given to the decoder is limited to 5 in order to avoid very confident probabilities. A (3,6)-regular low-density parity-check (LDPC) of rate $1/2$ is used, for a maximum of 100 iterations.

In Fig. 2 we show a system with $N_t = N_r = 6$ antennas. It

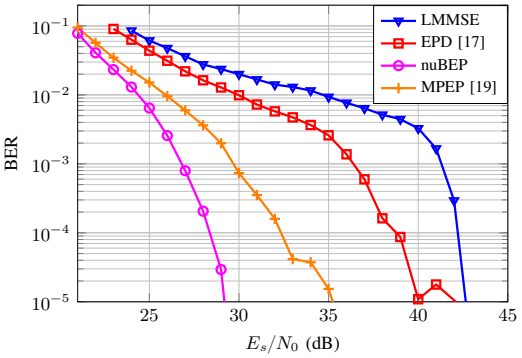


Fig. 2: BER along E_s/N_0 for nuBEP, EPD [17], MPEP [19] and LMMSE detectors, 128-QAM and averaged over 100 randomly channels in a 6×6 system after turbo 5 loops.

can be seen that EPD [17] improves the LMMSE performance but is far from nuBEP and MPEP. The reason is that the true prior used in the moment matching procedure is set to an uniform distribution, while we use a non-uniform one given by the output of the decoder, which better characterize the prior after the turbo feedback. The MPEP approach [19] quite outperforms both LMMSE and EPD because it uses a non-uniform prior. However, it does not achieve the performance of nuBEP because it just computes one iteration of the EP algorithm and does not use any damping procedure. Our approach, nuBEP, shows the more accurate and robust performance, due to its carefully chosen EP parameters, having gains of 14 dB and 6 dB with respect to the LMMSE and MPEP, respectively.

In Fig. 3 we increase the number of antennas to $N_t = N_r = 32$. In this scenario, the EPD approach shows instabilities at large E_s/N_0 since its parameters are not optimized for large-scale constellations and turbo schemes. Again, the best performance is obtained with our proposal, that has a remarkable improvement of 8 dB with respect to the LMMSE and of 1.5 dB compared to the MPEP algorithm.

V. Conclusion

We have proposed an EP-based turbo detector (nuBEP) for MIMO systems and large-size modulations where the optimal MAP algorithm is computationally unfeasible. The nuBEP detector quite outperforms the classical LMMSE and other EP-based detectors found in the literature. Specifically, it uses a non-uniform prior, rather than an uniform one as in [17]. This prior better characterizes the true prior used during the moment matching procedure of the EP algorithm once the turbo procedure has started. The proposed detector also optimizes its parameters to avoid some instabilities that appear at large E_s/N_0 and to reduce its complexity. Specifically, it reduces the number of EP iterations from 10 (used in [17]) to 3 after the feedback from the decoder. It also outperforms the EP detector in [19] since we include a different control of negative variances, a damping procedure and more EP iterations that let the algorithm achieve a more accurate solution. Simulations

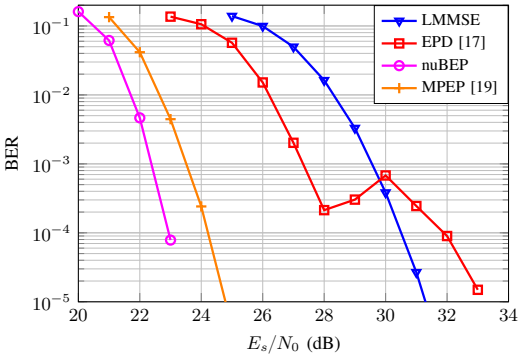


Fig. 3: BER along E_s/N_0 for nuBEP, EPD [17], MPEP [19] and LMMSE detectors, 128-QAM and averaged over 100 randomly channels in a 32×32 system after turbo 5 loops.

results show that the proposed nuBEP turbo detector has gains of 5-11 dB with respect to the EPD [17] and 1.5-6 dB compared to the MPEP [19].

REFERENCES

[1] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up mimo: Opportunities and challenges with very large arrays," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 40–60, Jan 2013.

[2] C. Berrou, *Codes and turbo codes*, ser. Collection IRIS. Springer Paris, 2010.

[3] C. Studer, A. Burg, and H. Bolcskei, "Soft-output sphere decoding: algorithms and vlsi implementation," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 2, pp. 290–300, February 2008.

[4] R. Wang and G. B. Giannakis, "Approaching mimo channel capacity with soft detection based on hard sphere decoding," *IEEE Transactions on Communications*, vol. 54, no. 4, pp. 587–590, April 2006.

[5] T. Datta, N. A. Kumar, A. Chockalingam, and B. S. Rajan, "A novel monte-carlo-sampling-based receiver for large-scale uplink multiuser mimo systems," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 7, pp. 3019–3038, Sept 2013.

[6] J. Goldberger and A. Leshem, "Mimo detection for high-order qam based on a gaussian tree approximation," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 4973–4982, Aug 2011.

[7] T. L. Narasimhan and A. Chockalingam, "Channel hardening-exploiting message passing (chemp) receiver in large-scale mimo systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 847–860, Oct 2014.

[8] T. P. Minka, "A family of algorithms for approximate Bayesian inference," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[9] M. Seeger, "Expectation propagation for exponential families," *Univ. Calif., Berkeley, CA, USA, Tech. Rep.*, 2005.

[10] I. Santos, J. J. Murillo-Fuentes, R. Boloix-Tortosa, E. Arias-de-Reyna, and P. M. Olmos, "Expectation propagation as turbo equalizer in ISI channels," *IEEE Trans. on Communications*, vol. 65, no. 1, pp. 360–370, Jan 2017.

[11] I. Santos, J. J. Murillo-Fuentes, E. Arias-de-Reyna, and P. M. Olmos, "Probabilistic equalization with a smoothing expectation propagation approach," *IEEE Trans. on Wireless Communications*, vol. 16, no. 5, pp. 2950–2962, May 2017.

[12] ——, "Turbo EP-based equalization: a filter-type implementation," *IEEE Trans. on Communications*, Sep 2017. Accepted.

[13] Y. Qi and T. Minka, "Window-based expectation propagation for adaptive signal detection in flat-fading channels," *IEEE Trans. on Wireless Communications*, vol. 6, no. 1, pp. 348–355, Jan 2007.

[14] ——, "Expectation propagation for signal detection in flat-fading channels," in *IEEE International Symposium on Information Theory, 2003. Proceedings.*, June 2003, pp. 358–358.

[15] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Pérez-Cruz, "Expectation propagation detection for high-order high-dimensional MIMO systems," *IEEE Trans. on Communications*, vol. 62, no. 8, pp. 2840–2849, Aug 2014.

[16] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Pérez-Cruz, "Improved performance of LDPC-coded MIMO systems with EP-based soft-decisions," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Jun 2014, pp. 1997–2001.

[17] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Pérez-Cruz, "Probabilistic MIMO symbol detection with expectation consistency approximate inference," *IEEE Trans. on Vehicular Technology*, Sep 2017. Accepted.

[18] J. Céspedes, "Approximate inference in massive MIMO scenarios with moment matching techniques," Ph.D. dissertation, Universidad Carlos III de Madrid, Jan 2017.

[19] M. Senst and G. Ascheid, "How the framework of expectation propagation yields an iterative IC-LMMSE MIMO receiver," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, Dec 2011, pp. 1–6.

Appendix E
Conference publications

E.1 Paper I

Authors: Irene Santos, Juan José Murillo-Fuentes, and Pablo M. Olmos.
Title: Block expectation propagation equalization for ISI channels.
Conference: 23rd European Signal Processing Conference (EUSIPCO 2015).
Pages: 379-383.
Date: September 2015.
Abstract: Actual communications systems use high-order modulations and channels with memory. However, as the memory of the channels and the order of the constellations grow, optimal equalization such as BCJR algorithm is computationally intractable, as their complexity increases exponentially with the number of taps and size of modulation. In this paper, we propose a novel low-complexity hard and soft output equalizer based on the Expectation Propagation (EP) algorithm that provides high-accuracy posterior probability estimations at the input of the channel decoder with similar computational complexity than the linear MMSE. We experimentally show that this quasi-optimal solution outperforms classical solutions reducing the bit error probability with low complexity when LDPC channel decoding is used, avoiding the curse of dimensionality with channel memory and constellation size.

BLOCK EXPECTATION PROPAGATION EQUALIZATION FOR ISI CHANNELS

Irene Santos, Juan José Murillo-Fuentes*, Pablo M. Olmos*†

* ETSI, Universidad de Sevilla
Dept. Teoría de la Señal y Comunicaciones
Camino de los Descubrimientos s/n,
41092 Sevilla, Spain.
E-mail: {irenesantos,murillo}@us.es

† Universidad Carlos III de Madrid
Dept. Teoría de la Señal y Comunicaciones
Avda. de la Universidad 30,
28911, Leganés (Madrid), Spain.
E-mail: olmos@tsc.uc3m.es

ABSTRACT

Actual communications systems use high-order modulations and channels with memory. However, as the memory of the channels and the order of the constellations grow, optimal equalization such as BCJR algorithm is computationally intractable, as their complexity increases exponentially with the number of taps and size of modulation. In this paper, we propose a novel low-complexity hard and soft output equalizer based on the Expectation Propagation (EP) algorithm that provides high-accuracy posterior probability estimations at the input of the channel decoder with similar computational complexity than the linear MMSE. We experimentally show that this quasi-optimal solution outperforms classical solutions reducing the bit error probability with low complexity when LDPC channel decoding is used, avoiding the curse of dimensionality with channel memory and constellation size.

Index Terms— Expectation propagation, BCJR algorithm, low complexity, channel equalization, ISI.

1. INTRODUCTION

Single input single output (SISO) communication channels are corrupted by additive white Gaussian noise (AWGN) and introduce inter-symbol interference (ISI) between transmitted symbols, due to its dispersive nature and the multiple paths of wireless communications [1]. Channel equalization is a solution to this problem, which provides estimations of the transmitted symbols and exploit diversity. Furthermore, rather than hard decision on the received symbols, nowadays channel decoders highly benefit from probabilistic estimates for each transmitted symbol given the received sequence [2].

Consider a discrete-time dispersive digital communication system, where the channel is completely defined by the channel state information (CSI) which is known at the receiver. Assuming perfect CSI and a channel with finite memory, linear equalization, such as the linear minimum-mean-squared-error (LMMSE) [1], is a low-cost alternative based on the minimization of the signal error. However, its results are far from the optimal solution provided by symbol maximum a posteriori (MAP) BCJR algorithm. The BCJR algorithm [3] computes the a posteriori probabilities (APP) for each transmitted symbol providing optimal decisions

$$p(u_k = \mathcal{A}|\mathbf{h}, \mathbf{y}) \quad \forall\, k = 1, ..., N \tag{1}$$

where $\mathbf{u}$ is the block frame transmitted taken from an N-dimensional alphabet $\mathcal{A}^N$ (of a M-ary constellation, i.e., of order $|\mathcal{A}| = M$), $\mathbf{h}$ is the CSI of a channel with L taps and $\mathbf{y}$ is the received sequence.

The BCJR algorithm works on a trellis representation and its complexity is proportional to the number of states. This number increases with the number of taps of the channel and the size of the constellation used. Specifically, for each symbol we have M^{L-1} possible states whose transition to the next state depends on each M possible received bit, so the final complexity of each step of the BCJR algorithm is $\mathcal{O}(M^L)$, which becomes intractable for the actual communications systems. The memory needed by this algorithm also grows exponentially, because it stores M^{L-1} variables. For all these reasons, in this paper we focus on an approximated solution whose complexity and memory are computationally realizable for the actual communications systems.

In this paper, we propose the EP algorithm as a low-complexity and high-accuracy solution for equalization in SISO systems and channels with memory. This approach has been successfully already applied to MIMO detection [4] and channel decoding [5]. The EP algorithm [6–8] can naturally and efficiently work with continuous distributions by moment matching and it powerfully deals with complex and versatile approximating functions. This novel solution exhibits a performance close to the optimal, as illustrated in the experiments included, with linear complexity similar to the one of the LMMSE. Using EP, we construct a Gaussian approximation to the posterior distribution of the transmitted symbol vector, i.e., $q_{EP}(\mathbf{u}) \approx p(\mathbf{u}|\mathbf{y})$. Iteratively, EP finds $q_{EP}(\mathbf{u})$

This work was partially funded by Spanish government (Ministerio de Economía y Competitividad TEC2012-38800-C03-01/02 and FEDER) and by Comunidad de Madrid in Spain (project 'CASI-CAM-CM', id. S2013/ICE-2845).

that aims to match the first two moments for each dimension (in parallel), whose direct computation from $p(\mathbf{u}|\mathbf{y})$ becomes computationally prohibitive for large N. The computational complexity of the algorithm per iteration is dominated by the inversion of a N-dimensional matrix, i.e., $\mathcal{O}(N^3)$. In addition, EP is a soft-output algorithm that provides a posterior probability estimation for each received symbol, which can be naturally fed to modern channel decoders.

The following notation is used throughout the paper. If $\mathbf{u}$ is a vector, u_i denotes the entry i of the vector $\mathbf{u}$ and $\mathbf{u_{i:j}}$ is a vector with the entries of $\mathbf{u}$ in the range i to j. The operator $\mathrm{diag}(\cdot)$ when applied to a vector, e.g. $\mathrm{diag}(\mathbf{u})$, returns a diagonal matrix with diagonal given by $\mathbf{u}$. To denote a normal distribution of a random variable u with mean μ and variance σ^2 we use the notation $\mathcal{N}\left(u : \mu, \sigma^2\right)$. In case of a random vector $\mathbf{u}$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ we use $\mathcal{N}\left(\mathbf{u} : \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$.

2. SYSTEM MODEL AND SOLUTIONS

We consider the discrete-time dispersive communication system depicted in Figure 1. A block of I message bits, $\mathbf{m} = [m_1, ..., m_I]^\top$, is encoded with a rate $R = I/T$ code into $\mathbf{b} = [b_1, ..., b_T]^\top$. An M-ary modulation is considered to obtain $N = \lceil T/\log_2 M \rceil$ symbols, $\mathbf{u}$. Then, the block frame $\mathbf{u} = [u_1, ..., u_N]^\top = \mathcal{R}(\mathbf{u}) + j\mathcal{I}(\mathbf{u})$ is transmitted over the channel, where each component $u_k = \mathcal{R}(u_k) + j\mathcal{I}(u_k) \in \mathcal{A}$. Here $\mathcal{A}$ denotes the set of symbols of the constellation of order $|\mathcal{A}| = M$, hence the alphabet of $\mathbf{u}$ symbols has size $|\mathcal{A}|^N$. The mean symbol energy transmitted is denoted by E_s. The channel is completely specified by the CSI, i.e., $\mathbf{h} = [h_1, ..., h_L]^\top$, where L is the length of the channel impulsive response. The received signal $\mathbf{y} = [y_1, ..., y_{N+L-1}]^\top \in \mathbb{C}$ is given by

$$\begin{bmatrix} y_1 \\ \\ \vdots \\ \\ y_{N+L-1} \end{bmatrix} = \begin{bmatrix} h_1 & & \mathbf{0} \\ \vdots & \ddots & \\ h_L & \ddots & h_1 \\ & \ddots & \vdots \\ \mathbf{0} & & h_L \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} + \begin{bmatrix} w_1 \\ \\ \vdots \\ \\ w_{N+L-1} \end{bmatrix} \tag{2}$$

or more compactly

$$\mathbf{y} = \mathbf{H}\mathbf{u} + \mathbf{w} \tag{3}$$

where $\mathbf{H}$ is a $(N+L-1) \times N$ matrix, the k-th received entry is given by

$$y_k = \sum_{i=1}^{L} h_i u_{k-i+1} + w_k = \mathbf{h}^\top \mathbf{u_{k:k-L+1}} + w_k \tag{4}$$

and $\mathbf{w} \sim \mathcal{N}\left(\mathbf{w} : \mathbf{0}, \sigma_w^2 \mathbf{I}\right)$ is a AWGN vector. In (2) we consider a transmission of N symbols where $u_i = 0\ \forall i \leq 0$ and $\forall i > N$.

Inference is typically presented using real-valued random variables, instead of complex-valued variables used in signal processing for communications. The system model in (3) can be translated into an equivalent double-sized real-valued representation that is obtained by considering the real and imaginary parts separately. Therefore, without loss of generality, in the following we adopt the real-valued channel model.

Given the model above, the posterior probability of the transmitted symbol vector $\mathbf{u}$ has the following expression:

$$p(\mathbf{u}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{u})p(\mathbf{u})}{p(\mathbf{y})} \propto \mathcal{N}\left(\mathbf{y} : \mathbf{H}\mathbf{u}, \sigma_w^2 \mathbf{I}\right) \prod_{k=1}^{N} \mathbb{I}_{u_k \in \mathcal{A}} \tag{5}$$

where $\mathbb{I}_{u_k \in \mathcal{A}}$ is the indicator function that takes value one if $u_k \in \mathcal{A}$ and zero otherwise.

Note that we are using simple equalization (see Figure 1). However, to improve the results, we could iteratively feed the soft detector with the output probabilities of the decoder, as in turbo equalization [9].

2.1. LMMSE algorithm

Given the CSI, the LMMSE equalizer [1] first proceeds by computing

$$\boldsymbol{\mu}_{MMSE} = \left(\mathbf{H}^\top \mathbf{H} + \frac{\sigma_w^2}{E_s}\mathbf{I}\right)^{-1} \mathbf{H}^\top \mathbf{y} \tag{6}$$

and then, it performs a component-wise hard decision by projecting each component of $\boldsymbol{\mu}_{MMSE}$ into the corresponding constellation

$$\hat{u}_{k\ MMSE} = \arg\min_{u_k \in \mathcal{A}} |u_k - \mu_{k\ MMSE}|^2. \tag{7}$$

The complexity of this solution is dominated by the matrix inversion in (6). The posterior approximate provided by the LMMSE algorithm is a Gaussian distribution with mean $\boldsymbol{\mu}_{MMSE}$ and covariance $\boldsymbol{\Sigma}_{MMSE}$

$$q_{MMSE}(\mathbf{u}) = \mathcal{N}\left(\mathbf{u} : \boldsymbol{\mu}_{MMSE}, \boldsymbol{\Sigma}_{MMSE}\right) \tag{8}$$

where

$$\boldsymbol{\Sigma}_{MMSE} = \sigma_w^2 \left(\mathbf{H}^\top \mathbf{H} + \frac{\sigma_w^2}{E_s}\mathbf{I}\right)^{-1}. \tag{9}$$

The symbol probability of each entry is computed by independently deciding on each component

$$q_{MMSE}(u_k = \mathcal{A}_i) \propto \mathcal{N}\left(\mathcal{A}_i : \mu_{k\ MMSE}, \Sigma_{k,k\ MMSE}\right). \tag{10}$$

3. EXPECTATION PROPAGATION

Expectation propagation or EP [6–8, 10] is a technique in Bayesian machine learning for approximating the true posterior distribution with exponential family distributions. It

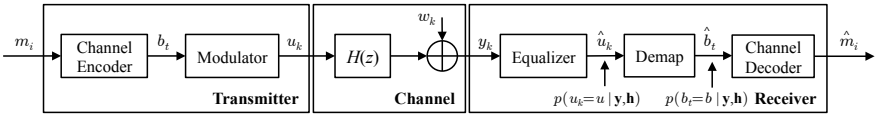


Fig. 1: System model.

is based on the minimization of the Kullback-Leibler divergence.

Suppose we are given some statistical distribution with hidden $\mathbf{x}$ and observables $\mathcal{D}$[1] that factors as follows

$$p(\mathbf{x}|\mathcal{D}) \propto f(\mathbf{x}) \prod_{i}^{\mathcal{I}} t_i(\mathbf{x}), \tag{11}$$

where $f(\mathbf{x})$ belongs to an exponential family $\mathcal{F}$ with sufficient statistics $\mathbf{\Phi}(\mathbf{x})$ and $t_i(\mathbf{x})$ are nonnegative factors that do not belong to the exponential family $\mathcal{F}$. When the true posterior $p(\mathbf{x})$ in (11) is analytically intractable or prohibitively complex, EP provides a feasible approximation to $p(\mathbf{x})$ by an exponential distribution $q(\mathbf{x})$ from $\mathcal{F}$ which factorizes as

$$q(\mathbf{x}) \propto f(\mathbf{x}) \prod_{i} \tilde{t}_i(\mathbf{x}) \tag{12}$$

where each factor $\tilde{t}_i(\mathbf{x}) \in \mathcal{F}$ is an approximation of the factor $t_i(\mathbf{x})$ in the true posterior (11). The approximation $q(\mathbf{x})$ is obtained by minimizing the Kullback-Leibler divergence with respect to $p(\mathbf{x})$, i.e. $q(\mathbf{x}) = \arg\min_{q'(\mathbf{x})\in\mathcal{F}} D_{KL}(p(\mathbf{x})\|q'(\mathbf{x}))$. This solution is equivalent to matching the expected sufficient statistics

$$\mathbb{E}_{q(\mathbf{x})}[\mathbf{\Phi}(\mathbf{x})] = \mathbb{E}_{p(\mathbf{x})}[\mathbf{\Phi}(\mathbf{x})] \tag{13}$$

where $\mathbb{E}_{q(\mathbf{x})}[\cdot]$ denotes expectation with respect to the distribution $q(\mathbf{x})$. Equation (13) is called $moment\ matching$ condition. If $q(\mathbf{x})$ is a Gaussian distribution $\mathcal{N}(\mathbf{x} : \boldsymbol{\mu}, \boldsymbol{\Sigma})$ then we minimize the Kullback-Leibler divergence by setting the mean $\boldsymbol{\mu}$ of $q(\mathbf{x})$ equal to the mean of $p(\mathbf{x})$ and the covariance $\boldsymbol{\Sigma}$ equal to the covariance of $p(\mathbf{x})$. However, the computation of the moments $\mathbb{E}_{p(\mathbf{x})}[\mathbf{\Phi}(\mathbf{x})]$ to construct $q(\mathbf{x})$ according to them is intractable because we can not infer over $p(\mathbf{x})$. To solve this problem, Minka proposed a sequential EP algorithm to iteratively obtain the solution in (11). The main idea behind the sequential EP algorithm is to do inference over a distribution of the form

$$\tilde{p}_i(\mathbf{x}) \propto \frac{q(\mathbf{x})}{\tilde{t}_i(\mathbf{x})} t_i(\mathbf{x}) \tag{14}$$

and optimize each factor $\tilde{t}_i(\mathbf{x})$ in turn independently in the context of all of the remaining factors. A detailed description of the EP algorithm is given in Algorithm 1 where $q^{(\ell)}(\mathbf{x})$ is the approximation to $q(\mathbf{x})$ in (12) at iteration ℓ.

[1]For simplicity, we omit the dependence on the observed data $\mathcal{D}$ to keep the notation uncluttered in the rest of the paper.

Algorithm 1 The EP algorithm

Initialiaze all the approximating factors $\tilde{t}_i(\mathbf{x})$ and then the approximation $q(\mathbf{x})$ in (12) by setting these factors $\tilde{t}_i(\mathbf{x})$.
repeat
  for $i = 1, ..., \mathcal{I}$ **do**
    1) Compute the $cavity$ distribution by removing $\tilde{t}_i(\mathbf{x})$ from the approximated distribution $q(\mathbf{x})$ by division, i.e., $q^{(\ell)\setminus i}(\mathbf{x}) = q^{(\ell)}(\mathbf{x})/\tilde{t}_i^{(\ell)}(\mathbf{x})$.
    2) Compute the distribution $\tilde{p}_i(\mathbf{x}) \propto t_i(\mathbf{x}) q^{(\ell)\setminus i}(\mathbf{x})$ and its moments

$$\mathbb{E}_{\tilde{p}_i(\mathbf{x})}[\mathbf{\Phi}(\mathbf{x})] \tag{15}$$

    3) Compute the new refined factor $\tilde{t}_i^{(\ell+1)}(\mathbf{x})$ by setting the moments of the distribution $\tilde{t}_i^{(\ell+1)}(\mathbf{x}) q^{(\ell)\setminus i}(\mathbf{x})$, denoted as $\mathbb{E}_{\tilde{t}_i^{(\ell+1)}(\mathbf{x}) q^{(\ell)\setminus i}(\mathbf{x})}(\mathbf{x})$, equal to (15).
  end for
until convergence (or stopped criterion)

4. BLOCK-EP EQUALIZER

In this section, we propose as a novel approach using the EP for channel equalization in a SISO system with ISI, naming it block-EP equalizer. We approximate the optimal solution in (5) by replacing each one of the non-Gaussian factors by an unnormalized Gaussian [4]

$$q(\mathbf{u}) \propto \mathcal{N}\left(\mathbf{y} : \mathbf{H}\mathbf{u}, \sigma_w^2 \mathbf{I}\right) \prod_{k=1}^{N} \exp\left(\gamma_k u_k - \frac{1}{2}\Lambda_k u_k^2\right) \tag{16}$$

where γ_k and $\Lambda_k > 0$ are real constants. For any value $\boldsymbol{\gamma} \in \mathbb{R}^{\mathrm{N}}$ and $\boldsymbol{\Lambda} \in \mathbb{R}_{+}^{\mathrm{N}}$, $q(\mathbf{u})$ is a Gaussian $\mathcal{N}(\mathbf{u} : \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$

$$\boldsymbol{\Sigma} = (\sigma_w^{-2}\mathbf{H}^{\top}\mathbf{H} + \mathrm{diag}(\boldsymbol{\Lambda}))^{-1} \tag{17}$$

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}(\sigma_w^{-2}\mathbf{H}^{\top}\mathbf{y} + \boldsymbol{\gamma}). \tag{18}$$

A detailed implementation of the *block-EP equalizer* (BEP equalizer) is included in Algorithm 2. At this point it is important to remark that in the approximation proposed we retain all the knowledge on the systems by including the first factor in (16) while approximating with EP the unknowns.

Algorithm 2 Block-EP equalizer

Initialize $\gamma_k^{(0)}=0$ and $\Lambda_k^{(0)}=E_s^{-1}$ for $k=1,...,N$.
The pair $(\gamma_k^{(\ell+1)},\Lambda_k^{(\ell+1)})$ is computed as follows:
for $\ell=0,...,P-1$ **do**
  for $k=1,...,N$ **do**
    1) Compute the k-th marginal of the distribution $q^{(\ell)}(\mathbf{u})$ in (16), namely $q_k^{(\ell)}(u_k)=\mathcal{N}\left(u_k:\mu_k^{(\ell)},\sigma_k^{2(\ell)}\right)$.
    2) Compute the $cavity$ marginal

$$q^{(\ell)\backslash k}(u_k)=\frac{q_k^{(\ell)}(u_k)}{\exp\left(\gamma_k^{(\ell)}u_k-\frac{1}{2}\Lambda_k^{(\ell)}u_k^2\right)}\sim \sim\mathcal{N}\left(u_k:t_k^{(\ell)},h_k^{2(\ell)}\right) \quad (19)$$

    where

$$h_k^{2(\ell)}=\frac{\sigma_k^{2(\ell)}}{1-\sigma_k^{2(\ell)}\Lambda_k^{(\ell)}},\quad t_k^{(\ell)}=h_k^{2(\ell)}\left(\frac{\mu_k^{(\ell)}}{\sigma_k^{2(\ell)}}-\gamma_k^{(\ell)}\right)$$

    3) Compute the mean $\mu_{p_k}^{(\ell)}$ and variance $\sigma_{p_k}^{2(\ell)}$ of the distribution $\hat{p}^{(\ell)}(u_k)\propto q^{(\ell)\backslash k}(u_k)\mathbb{I}_{u_k\in\mathcal{A}}$.
    4) Finally, the pair $(\gamma_k^{(\ell+1)},\Lambda_k^{(\ell+1)})$ is updated so that the following unnormalized Gaussian distribution

$$q^{(\ell)\backslash k}(u_k)\exp\left(\gamma_k^{(\ell+1)}u_k-\frac{1}{2}\Lambda_k^{(\ell+1)}u_k^2\right) \quad (20)$$

    has mean and variance equal to $\mu_{p_k}^{(\ell)}$ and $\sigma_{p_k}^{2(\ell)}$. The solution is given by

$$\Lambda_k^{(\ell+1)}=\beta\left(\frac{1}{\sigma_{p_k}^{2(\ell)}}-\frac{1}{h_k^{2(\ell)}}\right)+(1-\beta)\Lambda_k^{(\ell)} \quad (21)$$

$$\gamma_k^{(\ell+1)}=\beta\left(\frac{\mu_{p_k}^{(\ell)}}{\sigma_{p_k}^{2(\ell)}}-\frac{t_k^{(\ell)}}{h_k^{2(\ell)}}\right)+(1-\beta)\gamma_k^{(\ell)} \quad (22)$$

  end for
end for
Obtain the Gaussian approximation after EP algorithm, $q(\mathbf{u})\propto\mathcal{N}(\mathbf{u}:\boldsymbol{\mu},\boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are given by (18) and (17), respectively.
Compute the hard output and its symbol probability as

$$\hat{u}_k=\arg\min_{u_k\in\mathcal{A}}|u_k-\mu_k|^2 \quad (23)$$

$$q(u_k=\mathcal{A}_i)\propto\mathcal{N}(\mathcal{A}_i:\mu_k,\Sigma_{k,k}) \quad (24)$$

Eqn. (21) and (22) are proposed following the guidelines in [4, Eq. 35-36]. The parameter update in (21) may return a negative value $\Lambda_k^{(\ell+1)}$ for some k's which means that there is no pair $(\Lambda_k^{(\ell+1)},\gamma_k^{(\ell+1)})$ that sets the variance of the Gaussian in (20) at $\sigma_{p_k}^{2(\ell)}$. For that k's, we keep the previous values for these parameters. Note that all $(\gamma_k^{(\ell+1)},\Lambda_k^{(\ell+1)})$ pairs for $k=1,...,N$ can be updated in parallel and we only require the computation of a N-dimensional inverse matrix in (17) for each ℓ-iteration (typically around 10 [4]), so complexity of EP is dominated by the size of that inverse, i.e., $\mathcal{O}(N^3)$. We introduce a smoothing parameter $\beta\in[0,1]$ and a small constant ϵ that sets a minimum variance $\sigma_{p_k}^{2(\ell)}=\max(\epsilon,\sigma_{\hat{p}_k}^{2(\ell)})$ allowed per component to avoid numerical instabilities.

5. SIMULATION RESULTS

In this section, we illustrate the good performance of the BEP equalizer for channels with memory. We have set $\beta=0.3$, $\epsilon=1e^{-4}$ (for hard decisions), $\epsilon=0.5$ (for soft decisions) and $P=10$ iterations in the EP algorithm. In all the experiments presented in this section we consider block frames of 500 random bits encoded with a regular LDPC of rate 1/2 and we average the BER over 1000 different frames and 100 realizations of channels. Each tap is Gaussian distributed, and the whole channel response is normalized.

We first consider channels of 5 taps and 4-PAM modulation. In Figure 2, we depict the BER curves before (solid lines) and after (dashed lines) the LDPC decoder for BEP, LMMSE and BCJR equalization. Compared to the BCJR solution before the decoder, we are far about 3 dB for BER=10^{-3} and compared with the LMMSE method, BEP is able to improve the performance in 5 dB. After the decoder, we are less than 3 dB far from optimal solution for BER=10^{-3} and EP outperforms LMMSE in 3 dB for the same BER. A similar study is presented in Figure 3(a) for channels of 6 taps and 16-PAM modulation, excluding the BCJR solution, which we do not simulate due to its unaffordable computational complexity. For BER=10^{-3} BEP equalization outperforms LMMSE in 5 dB before the decoder and 4 dB after the decoder. In Figure 3(b) we illustrate the same constellation than in (a), but now increasing the number of taps to 15. Even with this high memory, EP exhibits an excellent performance. Specifically, we obtain a gain of 4 dB before the decoder for BER=10^{-3} and 2 dB after the decoder, with respect to LMMSE.

Finally, a computational complexity[2] analysis between BEP and BCJR is given in Table 1. LMMSE algorithm is not included because it only differs in a factor P compared with the BEP algorithm. When both L and M are not large, as in Figure 2, the complexity of the BCJR algorithm is not high and it can be computed. However, when L and M are increased, as in Figure 3, its complexity grows exponentially and becomes intractable while BEP remains unchanged with L or M.

[2]The complexity of the BCJR algorithm is $\mathcal{O}(M^LN)$ while BEP is $\mathcal{O}(PN^3)$.

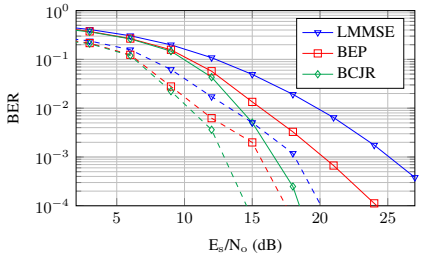
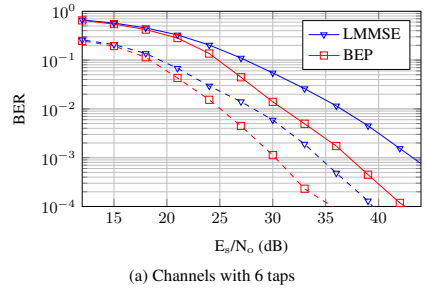
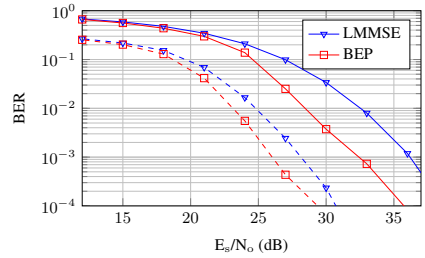


Fig. 2: BER for LMMSE, BEP and BCJR equalizers for channels with 5 taps and 4-PAM modulation.



(a) Channels with 6 taps



(b) Channels with 15 taps

Fig. 3: BER for LMMSE and BEP equalizers for 16-PAM.

Figure	M	L	N	Complexity BCJR	Complexity BEP	Reduct. factor
Fig. 2	4	5	500	$512e^3$	$125e^7$	$4e^{-4}$
Fig. 3(a)	16	6	250	$419e^7$	$156e^6$	27
Fig. 3(b)	16	15	250	$288e^{18}$	$156e^6$	$184e^{10}$

Table 1: Complexity comparison between algorithms.

6. CONCLUSION AND FUTURE WORK

The design of efficient equalizers is a challenging open problem. In this paper, we focus not only on symbol estimation but also the posterior probability estimation for each received symbol since the LDPC decoder needs a high quality APP to perform optimally. The optimal solution is intractable for the actual communications systems whenever we have large channel memory and/or large constellations. Classical methods such as linear MMSE can be used at the cost of a poorer performance. The novel BEP equalizer proposed in this paper is a soft-output algorithm that solves this problem, constructing tractable approximation to a given probability distribution. We have shown through simulations that the BEP equalizer quite outperforms the LMMSE, even with a high number of taps. Since it exhibits a similar structure, its computational burden and memory needs are similar to those of the LMMSE. Further improvements on the reduction of its computational complexity, i.e. of the covariance matrix inversion, remains as a future line of research.

REFERENCES

[1] John G. Proakis, *Digital Communications*, McGraw-Hill, New York, NY, 5 edition, 2008.

[2] Luis Salamanca, Juan José Murillo-Fuentes, and Fernando Pérez-Cruz, "Bayesian equalization for LDPC channel decoding," *IEEE Trans. on Signal Processing*, vol. 60, no. 5, pp. 2672–2676, May 2012.

[3] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *Information Theory, IEEE Trans. on*, vol. 20, no. 2, pp. 284–287, 1974.

[4] J. Céspedes, P.M. Olmos, M. Sánchez-Fernández, and F. Perez-Cruz, "Expectation propagation detection for high-order high-dimensional mimo systems," *IEEE Transactions on Communications*, vol. 62, no. 8, pp. 2840–2849, Aug 2014.

[5] P.M. Olmos, J.J. Murillo-Fuentes, and F. Perez-Cruz, "Tree-structure expectation propagation for LDPC decoding over the BEC," *IEEE Trans. on Information Theory*, vol. 59, no. 6, pp. 3354–3377, 2013.

[6] Thomas P. Minka, *A family of algorithms for approximate Bayesian Inference*, Ph.D. thesis, Massachusetts Institute of Technology, 2001.

[7] T. Minka, "Expectation propagation for approximate bayesian inference," in *UAI*, 2001, pp. 362–369.

[8] M.W. Seeger, "Expectation propagation for exponential families," *Univ. Calif., Berkeley, CA, USA, Tech. Rep.*, 2005.

[9] M. Tuchler and A.C. Singer, "Turbo equalization: An overview," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 920–952, Feb 2011.

[10] Christopher M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag, New York, Secaucus, NJ, USA, 2006.

E.2 Paper II

Authors: Irene Santos, and Juan José Murillo-Fuentes.
Title: Improved probabilistic EP-based receiver for MIMO systems and high-order modulations.
Conference: XXXIII Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2018).
Date: September 2018.
State: Accepted.
Abstract: Nowadays multiple-input multiple-output (MIMO) communications systems use efficient encoding schemes, high-order modulations and several number of antennas. In this scenario, the linear minimum-mean-square-error (LMMSE) is commonly used due to its low complexity. To improve its performance, some probabilistic detectors based on the expectation propagation (EP) algorithm has been proposed. In the design of an EP-based detector, the selection of its parameters is of great importance to control instabilities and avoid performance degradation, specially for high-order modulations. In this paper, we review the selection of these parameters in previous approaches and optimize them to achieve a robust performance. We focus specially on very high-order modulations, such as 64-QAM and 128-QAM, which has not been studied in previous EP-based proposals. We include some experiments where the set of parameters clearly shows a robust behavior and outperforms previous EP-based MIMO proposals in terms of bit error rate (BER).

Improved probabilistic EP-based receiver for MIMO systems and high-order modulations

Irene Santos, Juan José Murillo-Fuentes
irenesantos@us.es, murillo@us.es
Dept. Teoría de la Señal y Comunicaciones, Escuela T. Superior de Ingeniería, Universidad de Sevilla,
Camino de los Descubrimiento s/n, 41092 Sevilla, Spain.

***Abstract*—Nowadays multiple-input multiple-output (MIMO) communications systems use efficient encoding schemes, high-order modulations and several number of antennas. In this scenario, the linear minimum-mean-square-error (LMMSE) is commonly used due to its low complexity. To improve its performance, some probabilistic detectors based on the expectation propagation (EP) algorithm has been proposed. In the design of an EP-based detector, the selection of its parameters is of great importance to control instabilities and avoid performance degradation, specially for high-order modulations. In this paper, we review the selection of these parameters in previous approaches and optimize them to achieve a robust performance. We focus specially on very high-order modulations, such as 64-QAM and 128-QAM, which has not been studied in previous EP-based proposals. We include some experiments where the set of parameters clearly shows a robust behavior and outperforms previous EP-based MIMO proposals in terms of bit error rate (BER).**

I. Introduction

Current wireless communications are based on MIMO antenna systems, rather than single-input single-output (SISO) antennas, to increase the spectral efficiency and improve robustness against fading [1]. These systems transmit sequences of bits, which are modulated into symbols, $\mathbf{u}$, belonging to an alphabet $\mathcal{A}$ of size $\mathtt{M}$. They are then transmitted over communication channels, given by a matrix $\mathbf{H}$, assuming a narrowband channel, by N_t transmitted antennas. At the N_r receiver antennas, the transmitted symbols arrive mixed and corrupted by additive white Gaussian noise (AWGN), $\mathbf{w}$, during the propagation procedure. The described system can be expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{u} + \mathbf{w}, \tag{1}$$

where $\mathbf{w} \sim \mathcal{N}\left(\mathbf{w} : \mathbf{0}, \sigma_w^2\mathbf{I}\right)$. Based on the received signal and the knowledge of the channel, the detector estimates the transmitted symbols, $\hat{\mathbf{u}}$. These estimations can be probabilistic, resulting in a high benefit for modern channel decoders, such as low-density parity-check (LDPC) [2]. In addition, the output of the decoder can feed back the detector, improving even more the performance.

Optimal soft detection performed with the maximum a posteriori (MAP) algorithm maximizes the a posteriori probabilities (APP) for each transmitted symbol given the whole received signal,

$$\hat{\mathbf{u}} = \arg\max_{\mathbf{u}} p(\mathbf{u}|\mathbf{y}). \tag{2}$$

However, its complexity grows exponentially with the number of antennas and the constellation size, becoming intractable for just a few antennas and multilevel constellations. In this situation, low-complexity approximated approaches are used.

One extended solution is the well-known LMMSE, that assumes that the transmitted and received signal are jointly Gaussians. However, this algorithm does not consider the discrete nature of transmitted symbols, yielding results that are far from the optimal solution. To include the discrete feature of symbols and yet having low complexity, the EP detector can be used [3], [4]. This algorithm has been also successfully applied to turbo equalization [5], [6] and LDPC channel decoding [7], [8].

In the design of an EP-based detector, the selection of its parameters is of great importance to avoid performance degradation for high-order modulations and increasing signal-to-noise ratio (SNR). These parameters are: the minimum allowed variance, the smoothing parameter and the number of EP iterations. The latter determines the computational complexity of the algorithm and the other two control numerical instabilities in the EP updates. In this paper, we review the selection of these parameters and optimize them to achieve a robust performance, paying attention to very high-order modulations, such as 64-QAM and 128-QAM, which has not been studied in previous EP-based proposals.

The paper is organized as follows. We first describe in Section II the LMMSE and the EP solution. Section III is devoted to explain how the EP parameters can be optimized to achieve a robust performance even for high-order modulations. In Section IV, we include several experiments to compare previos EP-based detectors and our new proposal. We end with some conclusions in Section V.

II. Soft detection solutions

The MAP algorithm computes the APP for each transmitted symbol as

$$p(\mathbf{u}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{u})p(\mathbf{u})}{p(\mathbf{y})} \propto \mathcal{CN}\big(\mathbf{y} : \mathbf{H}\mathbf{u}, \sigma_w^2\mathbf{I}\big) \prod_{k=1}^{\mathtt{N}} \mathbb{I}_{u_k \in \mathcal{A}}, \tag{3}$$

where the indicator function is defined as

$$\mathbb{I}_{u_k \in \mathcal{A}} = \frac{1}{\mathtt{M}} \sum_{u \in \mathcal{A}} \delta(u_k - u). \tag{4}$$

This probability represents the true prior for the transmitted symbols. Once the posterior in (3) is obtained, the extrinsic log-likelihood ratios (LLRs) are computed accordingly and are handled to the channel decoder. The information provided by the decoder can then feed back the equalizer as an updated a priori information for the transmitted symbols.

However, the complexity of the MAP detector grows exponentially with the number of transmit antennas, N_t, and the size of the constellation used, $\mathtt{M}$. For large number of antennas

or high-order constellations, the MAP algorithm is computationally unfeasible and some low-complexity solutions, such as the LMMSE or EP algorithm are used. Both approximated solutions are explained below.

A. Turbo LMMSE

The posterior approximation provided by the LMMSE detector is obtained by replacing the discrete uniform prior $p(\mathbf{u})$ in (3) by a product of independent Gaussian distributions whose moments are set with the prior information of the transmitted symbols. In the case of zero mean constellations with energy E_s, we set these moments to $\mathbb{E}[u_k] = 0$ and $\mathbb{V}[u_k] = 1/E_s$. This yields another Gaussian for the posterior distribution given by

$$q(\mathbf{u}) = \mathcal{N}(\mathbf{u} : \boldsymbol{\mu}, \mathbf{S}) \tag{5}$$

where

$$\mathbf{S} = \left(\sigma_w^{-2}\mathbf{H}^{\mathrm{H}}\mathbf{H} + \boldsymbol{\Sigma}_{\mathbf{u}}^{-1}\right)^{-1}, \tag{6}$$

$$\boldsymbol{\mu} = \mathbf{S}\left(\sigma_w^{-2}\mathbf{H}^{\mathrm{H}}\mathbf{y} + \boldsymbol{\Sigma}_{\mathbf{u}}^{-1}\boldsymbol{\mu}_{\mathbf{u}}\right) \tag{7}$$

and $\boldsymbol{\mu}_{\mathbf{u}} = [\mathbb{E}[u_1], ..., \mathbb{E}[u_{\mathtt{N}}]]^\top$, $\boldsymbol{\Sigma}_{\mathbf{u}} = \mathrm{diag}\left(\mathbb{V}[u_1], ..., \mathbb{V}[u_{\mathtt{N}}]\right)$.

When the feedback from the decoder is available, the moments $\boldsymbol{\mu}_{\mathbf{u}}$ and $\boldsymbol{\Sigma}_{\mathbf{u}}$ will be updated with the statistics from the distribution at the output of the decoder. This process is repeated along T iterations, yielding a turbo LMMSE detection.

B. Turbo EP detector (EPD)

The EP algorithm is used to find a more accurate Gaussian approximation to the posterior in (3). To do this, it replaces the non Gaussian terms in (3), $\mathbb{I}_{u_k \in \mathcal{A}}$, by Gaussians given by

$$\tilde{p}^{[\ell]}(u_k) \sim \mathcal{CN}\Big(u_k : m_k^{[\ell]}, v_k^{[\ell]}\Big), \tag{8}$$

yielding a posterior distribution which is also Gaussian distribution,

$$q^{[\ell]}(\mathbf{u}) = \mathcal{CN}\big(\mathbf{y} : \mathbf{H}\mathbf{u}, \sigma_w^2\mathbf{I}\big) \prod_{k=1}^{\mathtt{N}} \tilde{p}^{[\ell]}(u_k) = \mathcal{CN}\Big(\mathbf{u} : \boldsymbol{\mu}^{[\ell]}, \mathbf{S}^{[\ell]}\Big) \tag{9}$$

where

$$\mathbf{S}^{[\ell]} = \left(\sigma_w^{-2}\mathbf{H}^{\mathrm{H}}\mathbf{H} + \mathbf{V}^{[\ell]^{-1}}\right)^{-1}, \tag{10}$$

$$\boldsymbol{\mu}^{[\ell]} = \mathbf{S}^{[\ell]}\left(\sigma_w^{-2}\mathbf{H}^{\mathrm{H}}\mathbf{y} + \mathbf{V}^{[\ell]^{-1}}\mathbf{m}^{[\ell]}\right) \tag{11}$$

and $\mathbf{m}^{[\ell]} = [m_1^{[\ell]}, ..., m_{\mathtt{N}}^{[\ell]}]^\top$ and $\mathbf{V}^{[\ell]} = \mathrm{diag}\left(v_1^{[\ell]}, ..., v_{\mathtt{N}}^{[\ell]}\right)$. This estimation computed by the detector exploits the discrete nature of the constellation by means of the EP algorithm, that proceeds iteratively as follows.

Each ℓ-th iteration, this approach computes a new Gaussian estimation for the prior,

$$\tilde{p}^{[\ell+1]}(u_k) \sim \mathcal{CN}\Big(u_k : m_k^{[\ell+1]}, v_k^{[\ell+1]}\Big), \tag{12}$$

by matching the moments of the true posterior and the estimated one,

$$\tilde{p}^{[\ell+1]}(u_k) = \frac{\mathrm{Proj}_G[q_E^{[\ell]}(u_k)\mathbb{I}_{u_k \in \mathcal{A}}]}{q_E^{[\ell]}(u_k)}, \tag{13}$$

where $\mathrm{Proj}_G[\cdot]$ is the projection of the distribution given as an argument into the family of Gaussians, $\mathbb{I}_{u_k \in \mathcal{A}}$ is given by (4) and $q_E^{[\ell]}(u_k)$ is the so called extrinsic function, that is defined as

$$q_E^{[\ell]}(u_k) = \frac{q^{[\ell]}(u_k)}{\tilde{p}^{[\ell]}(u_k)} \sim \mathcal{CN}\Big(u_k : z_k^{[\ell]}, \eta_k^{[\ell]}\Big) \tag{14}$$

where

$$z_k^{[\ell]} = \frac{\mu_k^{[\ell]} v_k^{[\ell]} - m_k^{[\ell]} s_k^{[\ell]}}{v_k^{[\ell]} - s_k^{[\ell]}}, \tag{15}$$

$$\eta_k^{[\ell]} = \frac{s_k^{[\ell]} v_k^{[\ell]}}{v_k^{[\ell]} - s_k^{[\ell]}} \tag{16}$$

and, in turn, μ_k is the k-th entry of (11) and s_k is the k-th element of the diagonal of (10). Note that $q^{[\ell]}(u_k)$ in the previous equation is computed by marginalizing (9). This EP procedure can also be seen as an LMMSE followed by the moment matching in (13), as shown in Fig. 1.

After S iterations of the EP algorithm, the extrinsic message given by (14), i.e., $q_E^{[S+1]}(u_k)$, is given to the decoder as a starting point which is more accurate than the one of the LMMSE solution. Then, the performance is improved with a computational complexity which is proportional to the one of the LMMSE.

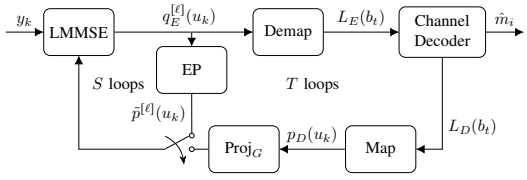


Fig. 1: Turbo receiver diagram used in EPD.

The whole EP turbo detection procedure is explained in Algorithm 1, where we have denoted the distributed computed by the channel decoder as $p_D(u_k)$. It also includes a control of negatives variances, a damping procedure (β) and a minimum allowed variance (ϵ). The values of these parameters will be explained in Section III.

III. Selection of the EP parameters

The update of the EP solution is a critical issue, since instabilities and performance degradation might appear. For high-order modulations, the influence of these problems will be even more pronounced. To control them, some EP parameters are introduced: the minimum allowed variance (ϵ), the smoothing parameter (β) and the number of EP iterations (S). Parameters ϵ and β control numerical instabilities in the EP updates and S determines the computational complexity of our algorithm. The parameter S is normally set to 10 [3], [9]. In the following subsections, we review the values of the parameters ϵ and β in previous approaches [3] and retune these values to control the instabilities that appears when using high-order modulations and increasing SNR. We denote the approach in [3] as EPD and our proposal as revised EPD (r-EPD). In Table I we include a comparison between the EP parameters used in [3] and the proposed ones.

Algorithm 1 EP Turbo Decoder

Initialization: Set $p_D(u_k) = \mathbb{I}_{u_k \in \mathcal{A}}$ for $k = 1, \ldots, N_t$

for $t = 1, ..., T$ **do**

1) Compute the mean $m_k^{[1]}$ and variance $v_k^{[1]}$ as

$$m_k^{[1]} = \sum_{u \in \mathcal{A}} u \cdot p_D(u_k = u), \tag{17}$$

$$v_k^{[1]} = \sum_{u \in \mathcal{A}} (u - m_k^{[1]})^*(u - m_k^{[1]}) \cdot p_D(u_k = u). \tag{18}$$

for $\ell = 1, ..., S$ **do**

for $k = 1, ..., N_t$ **do**

2) Compute the k-th extrinsic distribution, $q_E^{[\ell]}(u_k)$, as in (14).

3) Obtain the distribution $\hat{p}^{[\ell]}(u_k) \propto q_E^{[\ell]}(u_k)\mathbb{I}_{u_k \in \mathcal{A}}$ and estimate its mean $\mu_{p_k}^{[\ell]}$ and variance $\sigma_{p_k}^{2[\ell]}$. Set a minimum allowed variance as $\sigma_{p_k}^{2[\ell]} = \max(\epsilon, \sigma_{p_k}^{2[\ell]})$.

4) Run the moment matching procedure in (13) to obtain

$$v_{k,new}^{[\ell+1]} = \frac{\sigma_{p_k}^{2[\ell]} \eta_k^{[\ell]}}{\eta_k^{[\ell]} - \sigma_{p_k}^{2[\ell]}}, \tag{19}$$

$$m_{k,new}^{[\ell+1]} = v_{k,new}^{[\ell+1]} \left(\frac{\mu_{p_k}^{[\ell]}}{\sigma_{p_k}^{2[\ell]}} - \frac{z_k^{[\ell]}}{\eta_k^{[\ell]}} \right). \tag{20}$$

5) Run the damping procedure as

$$v_k^{[\ell+1]} = \left(\beta \frac{1}{v_{k,new}^{[\ell+1]}} + (1-\beta) \frac{1}{v_k^{[\ell]}} \right)^{-1}, \tag{21}$$

$$m_k^{[\ell+1]} = v_k^{[\ell+1]} \left(\beta \frac{m_{k,new}^{[\ell+1]}}{v_{k,new}^{[\ell+1]}} + (1-\beta) \frac{m_k^{[\ell]}}{v_k^{[\ell]}} \right). \tag{22}$$

6) Control of negative variances

if $\eta_k^{[\ell+1]} < 0$ **then**

$$v_k^{[\ell+1]} = v_k^{[\ell]}, \quad m_k^{[\ell+1]} = m_k^{[\ell]}. \tag{23}$$

end if

end for

end for

7) With the values $m_k^{[S+1]}, v_k^{[S+1]}$ obtained after the EP algorithm, calculate the extrinsic distribution $q_E^{[S+1]}(u_k)$ as in (14).

6) Demap the extrinsic distribution and compute the extrinsic log-likelihood ratio (LLR), $L_E(b_t)$, that will be given to the channel decoder.

7) Run the channel decoder to output $p_D(u_k)$

end for

Output: Deliver $L_E(b_t)$ to the channel decoder.

A. Damping

A good damping procedure is necessary to avoid performance degradation for high-order modulations and instabilities at large E_b/N_0. For this reason, the current update of (12) is averaged with its previous value by using damping as in (21)-(22). The damping parameter, β, will control the speed of the EP algorithm: a high value will allow the algorithm achieve the stationary point with less EP iterations than a low value. In [3], [4], this parameter is set to $\beta = 0.95$. However, after extensive simulations, we found performance degradation at large E_b/N_0 for high-order modulations (see Section IV). For this reason, we propose setting it to $\beta = 0.1$, letting the algorithm to reach slowly but confidently the stationary solution.

TABLE I: Comparison of EP parameters

Algorithm	ϵ	β	S
EPD [3]	$2^{-\max(\ell-4,1)}$	0.95	10
r-EPD	$1e^{-8}$	0.1	10

B. Minimum variance

Slight changes in the EP updates have a great impact in the overall performance. For this reason, the minimum variance allowed for the numerator in (13) is controlled by the value of ϵ. In [3], [4], this parameter decreases exponentially after the first 4 iterations as $\epsilon = 2^{-\max(\ell-4,1)}$. During the first iterations it is kept to a high value (0,5) to avoid over-fitting at low E_b/N_0. Since we propose to decrease the value of the damping parameter, the minimum variance is proposed to be $\epsilon = 1e^{-8}$.

IV. Simulation results

In this section, we compare the performance of EPD in [3] (○), our proposed r-EPD detector with optimized parameters (□) and the LMMSE (△). We used codewords of $V = 4096$ bits, 32×32 systems and modulations of different orders. The result is averaged over 10^4 random frames. The absolute value of LLRs given to the decoder is limited to 5 in order to avoid very confident probabilities. We use a (3,6)-regular LDPC of rate 1/2, and a maximum of 100 iterations in the decoding.

In Fig. 2, we depict the BER curves averaged over 100 random complex-valued channels and 64-QAM modulation. We represent the performance without feedback from the decoder with solid lines and after 5 loops between detector and decoder with dashed lines. Before the feedback from the decoder, it can be seen that EPD slightly improves our proposal for low E_b/N_0. However, in the large E_b/N_0-regime the EPD shows instabilities while our proposal has a quite robust behavior. Once the decoder starts feeding back the detector, the instabilities of EPD disappear and improves in about 0.5 dB its performance without feedback and in 2 dB the LMMSE performance. Our r-EPD detector keeps its robust behavior and have a remarkable improvement of 1 dB with respect to EPD.

In Fig. 3 we show the same scenario than in the previous figure, but increasing the order of the constellation to a 128-QAM. The EPD shows a more unstable behavior than in the previous scenario because it cannot avoid degradation for large E_b/N_0 even with the help of the decoder. On the other hand, our r-EPD detector exhibits a robust behavior and improves the EPD and LMMSE performance in 1 and 3.5 dB, respectively.

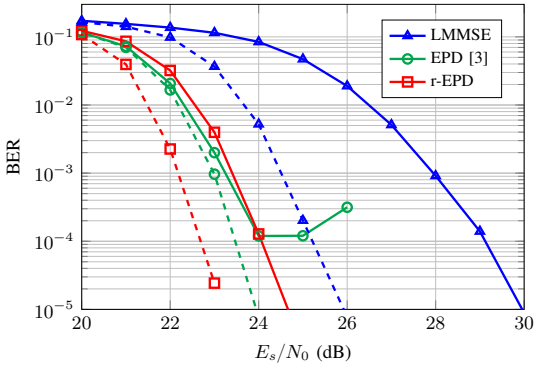


Fig. 2: BER along E_b/N_0 for r-EPD, EPD [3] and LMMSE detectors, 64-QAM and averaged over 100 randomly channels in a 32×32 system without feedback from the decoder (solid) and after $T = 5$ turbo loops (dashed).

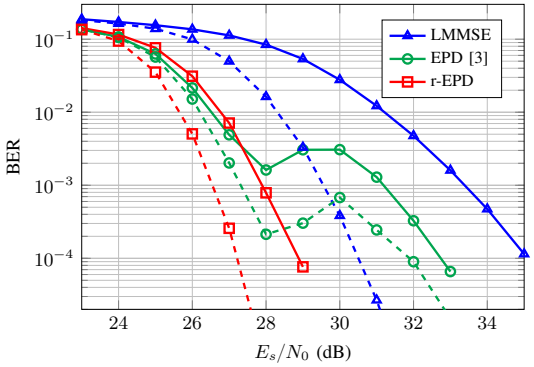


Fig. 3: BER along E_s/N_0 for r-EPD, EPD [3] and LMMSE detectors, 128-QAM and averaged over 100 randomly channels in a 32×32 system without feedback from the decoder (solid) and after $T = 5$ turbo loops (dashed).

V. Conclusions

The MAP optimal decoder becomes intractable for many communications systems, whenever we have large number of antennas and/or large constellations. Classical methods such as LMMSE can be used as a low-cost alternative with poorer performance. This performance can be greatly improved by means of the EP algorithm, that includes the restriction of symbols being discrete. When implementing an EP-based detector, the selection of its parameters is of crucial importance since they control the instabilities and avoid performance degradation.

In this paper, we review the parameters used by previous EP-based detectors and retuned them to avoid degradation with high-order modulations. After simulating previous approaches, we conclude that they are optimized for low E_s/N_0 regime and not higher order modulations because their performance degrades at large E_s/N_0. We compare these approaches with the new proposed parameters, showing quite a robust behavior for high-order modulations and a 3dB gain in terms of BER.

Acknowledgements

This work was partially funded by Spanish government (Ministerio de Economía y Competitividad TEC2012-38800-C03 and TEC2016-78434-C3) and by the European Union (FEDER).

References

[1] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up mimo: Opportunities and challenges with very large arrays," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 40–60, Jan 2013.

[2] L. Salamanca, J. J. Murillo-Fuentes, and F. Pérez-Cruz, "Bayesian equalization for LDPC channel decoding," *IEEE Trans. on Signal Processing*, vol. 60, no. 5, pp. 2672–2676, May 2012.

[3] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Pérez-Cruz, "Probabilistic MIMO symbol detection with expectation consistency approximate inference," *IEEE Trans. on Vehicular Technology*, Sep 2017. Accepted.

[4] J. Céspedes, "Approximate inference in massive mimo scenarios with moment matching techniques," Ph.D. dissertation, Universidad Carlos III de Madrid, Jan 2017.

[5] I. Santos, J. J. Murillo-Fuentes, R. Boloix-Tortosa, E. Arias-de-Reyna, and P. M. Olmos, "Expectation propagation as turbo equalizer in ISI channels," *IEEE Trans. on Communications*, vol. 65, no. 1, pp. 360–370, Jan 2017.

[6] I. Santos, J. J. Murillo-Fuentes, E. Arias-de-Reyna, and P. M. Olmos, "Probabilistic equalization with a smoothing expectation propagation approach," *IEEE Trans. on Wireless Communications*, vol. 16, no. 5, pp. 2950–2962, May 2017.

[7] P. M. Olmos, J. J. Murillo-Fuentes, and F. Pérez-Cruz, "Tree-structure expectation propagation for LDPC decoding over the BEC," *IEEE Trans. on Information Theory*, vol. 59, no. 6, pp. 3354–3377, 2013.

[8] L. Salamanca, P. M. Olmos, F. Pérez-Cruz, and J. J. Murillo-Fuentes, "Tree-structured expectation propagation for LDPC decoding over BMS channels," *IEEE Trans. on Communications*, vol. 61, no. 10, pp. 4086–4095, Oct 2013.

[9] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Pérez-Cruz, "Expectation propagation detection for high-order high-dimensional MIMO systems," *IEEE Trans. on Communications*, vol. 62, no. 8, pp. 2840–2849, Aug 2014.

List of Figures

List of Tables

Bibliography

[1] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, *Optimal decoding of linear codes for minimizing symbol error rate (corresp.)*, IEEE Trans. on Information Theory **20** (1974), no. 2, 284–287.

[2] C. Berrou, *Codes and turbo codes*, Collection IRIS, Springer Paris, 2010.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, *Near shannon limit error-correcting coding and decoding: Turbo-codes. 1*, Proc. IEEE International Conference on Communications (ICC), vol. 2, May 1993, pp. 1064–1070 vol.2.

[4] C. M. Bishop, *Pattern recognition and machine learning (information science and statistics)*, Springer-Verlag, New York, Secaucus, NJ, USA, 2006.

[5] J. Céspedes, *Approximate inference in massive MIMO scenarios with moment matching techniques*, Ph.D. thesis, Universidad Carlos III de Madrid, Jan 2017.

[6] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Pérez-Cruz, *Expectation propagation detection for high-order high-dimensional MIMO systems*, IEEE Trans. on Communications **62** (2014), no. 8, 2840–2849.

[7] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Pérez-Cruz, *Improved performance of LDPC-coded MIMO systems with EP-based soft-decisions*, Proc. IEEE International Symposium on Information Theory (ISIT), Jun 2014, pp. 1997–2001.

[8] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Pérez-Cruz, *Probabilistic MIMO symbol detection with expectation consistency approximate inference*, IEEE Trans. on Vehicular Technology **67** (2018), no. 4, 3481–3494.

[9] P. Chevillat and E. Eleftheriou, *Decoding of trellis-encoded signals in the presence of intersymbol interference and noise*, IEEE Trans. on Communications **37** (1989), no. 7, 669–676.

[10] G. Colavolpe, G. Ferrari, and R. Raheli, *Reduced-state BCJR-type algorithms*, IEEE Journal on Sel. Areas in Communications **19** (2001), no. 5, 848–859.

[11] G. Colavolpe, D. Fertonani, and A. Piemontese, *SISO detection over linear channels with linear complexity in the number of interferers*, IEEE Journal of Sel. Topics in Signal Processing **5** (2011), no. 8, 1475–1485.

[12] G. Colavolpe and G. Germi, *On the application of factor graphs and the sum-product algorithm to ISI channels*, IEEE Trans. on Communications **53** (2005), no. 5, 818–825.

[13] T. Datta, N. A. Kumar, A. Chockalingam, and B. S. Rajan, *A novel monte-carlo-sampling-based receiver for large-scale uplink multiuser MIMO systems*, IEEE Trans. on Vehicular Technology **62** (2013), no. 7, 3019–3038.

[14] C. Douillard, M. Jezequel, C. Berrou, P. Didier, and A. Picart, *Iterative correction of intersymbol interference: turbo-equalization*, European Trans. on Telecommunications **6** (1995), no. 5, 507–512.

[15] A. Duel-Hallen and C. Heegard, *Delayed decision-feedback sequence estimation*, IEEE Trans. on Communications **37** (1989), no. 5, 428–436.

[16] M.V. Eyuboglu and S.U.H. Qureshi, *Reduced-state sequence estimation with set partitioning and decision feedback*, IEEE Trans. on Communications **36** (1988), no. 1, 13–20.

[17] D. Falconer, S. L. Ariyavisitakul, A. Benyamin-Seeyar, and B. Eidson, *Frequency domain equalization for single-carrier broadband wireless systems*, IEEE Communications Magazine **40** (2002), no. 4, 58–66.

[18] D. Fertonani, A. Barbieri, and G. Colavolpe, *Reduced-complexity BCJR algorithm for turbo equalization*, IEEE Trans. on Communications **55** (2007), no. 12, 2279–2287.

[19] V. Franz and J.B. Anderson, *Concatenated decoding with a reduced-search BCJR algorithm*, IEEE Journal on Sel. Areas in Communications **16** (1998), no. 2, 186–195.

[20] J. Goldberger and A. Leshem, *MIMO detection for high-order QAM based on a gaussian tree approximation*, IEEE Trans. on Information Theory **57** (2011), no. 8, 4973–4982.

[21] Q. Guo and L. Ping, *LMMSE turbo equalization based on factor graphs*, IEEE Journal on Sel. Areas in Communications **26** (2008), no. 2, 311–319.

[22] S. Haykin, *Communication systems*, 5th ed., Wiley Publishing, 2009.

[23] C. Heegard and S. B. Wicker, *Turbo coding*, Kluwer Academic Publishers, Boston, MA, USA, 1999.

[24] J. Hu, H. A. Loeliger, J. Dauwels, and F. R. Kschischang, *A general computation rule for lossy summaries/messages with examples from equalization*, Proc. 44th Allerton Conf. Communication, Control, and Computing, Sep 2006, pp. 27–29.

[25] J. Karjalainen, N. Veselinovic, K. Kansanen, and T. Matsumoto, *Iterative frequency domain joint-over-antenna detection in multiuser MIMO*, IEEE Trans. on Wireless Communications **6** (2007), no. 10, 3620–3631.

[26] S. M. Kay, *Fundamentals of statistical signal processing, volume I: estimation theory*, Prentice Hall, 1993.

[27] N. Kim, J. Kim, S. C. Lim, and H. Park, *BER of MIMO-BICM system at current detection/decoding cycle*, IEEE Wireless Communications Letters **6** (2017), no. 1, 78–81.

[28] R. Koetter, A.C. Singer, and M. Tüchler, *Turbo equalization*, IEEE Signal Processing Magazine **21** (2004), no. 1, 67–80.

[29] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, *Factor graphs and the sum-product algorithm*, IEEE Trans. on Information Theory **47** (2001), no. 2, 498 –519.

[30] L. Liu and L. Ping, *An extending window MMSE turbo equalization algorithm*, IEEE Signal Processing Letters **11** (2004), no. 11, 891–894.

[31] D. J. C. MacKay, *Information theory, inference, and learning algorithms*, Cambridge University Press, 2003.

[32] T. P. Minka, *Expectation propagation for approximate Bayesian inference*, Proc. 17th Conference on Uncertainty in Artificial Intelligence (UAI), 2001, pp. 362–369.

[33] ———, *A family of algorithms for approximate Bayesian inference*, Ph.D. thesis, Massachusetts Institute of Technology, 2001.

[34] K. Muranov, *Survey of MMSE channel equalizers*, Tech. report, University of Illinois, Chicago, 2010.

[35] T. L. Narasimhan and A. Chockalingam, *Channel hardening-exploiting message passing (CHEMP) receiver in large-scale MIMO systems*, IEEE Journal of Sel. Topics in Signal Processing **8** (2014), no. 5, 847–860.

[36] B. Ning, R. Visoz, and A. O. Berthet, *Extrinsic versus a posteriori probability based iterative LMMSE-IC algorithms for coded MIMO communications: Performance and analysis*, International Symposium on Wireless Communication Systems (ISWCS), Aug 2012, pp. 386–390.

[37] M. Opper and O. Winther, *Expectation consistent approximate inference*, Journal of Machine Learning Research **6** (2005), 2177–2204.

[38] S. Park and S. Choi, *Iterative equalizer based on kalman filtering and smoothing for MIMO-ISI channels*, IEEE Transactions on Signal Processing **63** (2015), no. 19, 5111–5120.

[39] J. G. Proakis, *Digital communications*, 5 ed., McGraw-Hill, New York, NY, 2008.

[40] Y. Qi and T. Minka, *Expectation propagation for signal detection in flat-fading channels*, Proc. IEEE International Symposium on Information Theory (ISIT), June 2003, pp. 358–358.

[41] Y. Qi and T.P. Minka, *Window-based expectation propagation for adaptive signal detection in flat-fading channels*, IEEE Trans. on Wireless Communications **6** (2007), no. 1, 348–355.

[42] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, *Scaling up MIMO: Opportunities and challenges with very large arrays*, IEEE Signal Processing Magazine **30** (2013), no. 1, 40–60.

[43] F. Rusek and A. Prlja, *Optimal channel shortening for MIMO and ISI channels*, IEEE Trans. on Wireless Communications **11** (2012), no. 2, 810–818.

[44] C. Poulliat S. Sahin, A. M. Cipriano and M. L. Boucheret, *A framework for iterative frequency domain EP-based receiver design*, IEEE Trans. on Communications (2018. Submitted), [Online]. Available: *https://arxiv.org/abs/1804.01484*.

[45] L. Salamanca, J. J. Murillo-Fuentes, and F. Pérez-Cruz, *Bayesian equalization for LDPC channel decoding*, IEEE Trans. on Signal Processing **60** (2012), no. 5, 2672–2676.

[46] I. Santos and J. J. Murillo-Fuentes, *EP-based turbo detection for MIMO receivers and large-scale systems*, IEEE Trans. on Vehicular Technology (2018. Submitted), [Online]. Available: *https://arxiv.org/abs/1805.05065*.

[47] I. Santos, J. J. Murillo-Fuentes, and E. Arias-de-Reyna, *Equalization with expectation propagation at smoothing level*, (2018. To be submitted), [Online]. Available: *https://arxiv.org/abs/1809.00806*.

[48] I. Santos, J. J. Murillo-Fuentes, E. Arias-de-Reyna, and P. M. Olmos, *Probabilistic equalization with a smoothing expectation propagation approach*, IEEE Trans. on Wireless Communications **16** (2017), no. 5, 2950–2962.

[49] _____, *Turbo EP-based equalization: a filter-type implementation*, IEEE Trans. on Communications (2017. Accepted), [Online]. Available: *https://ieeexplore.ieee.org/document/8353388/*.

[50] I. Santos, J. J. Murillo-Fuentes, R. Boloix-Tortosa, E. Arias-de-Reyna, and P. M. Olmos, *Expectation propagation as turbo equalizer in ISI channels*, IEEE Trans. on Communications **65** (2017), no. 1, 360–370.

[51] I. Santos, J. J. Murillo-Fuentes, and P. M. Olmos, *Block expectation propagation equalization for ISI channels*, Proc. 23rd European Signal Processing Conference (EUSIPCO), Sep 2015, pp. 379–383.

[52] R. Schober, *Detection and estimation of signals in noise*, Tech. report, University of British Columbia, Vancouver, 2010, [Online]. Available: *https://pdfs.semanticscholar.org/67e1/05b1cc0c9addcc6bf4d398d04a786bba9b24.pdf*.

[53] M.W. Seeger, *Expectation propagation for exponential families*, Tech. report, Univ. Calif., Berkeley, CA, USA, Tech. Rep., 2005, [Online]. Available: *https://www-users.cs.umn.edu/~baner029/Teaching/Fall07/papers/epexpfam.pdf*.

[54] M. Senst and G. Ascheid, *How the framework of expectation propagation yields an iterative IC-LMMSE MIMO receiver*, Proc. IEEE Global Telecommunications Conference (GLOBECOM), Dec 2011, pp. 1–6.

[55] M. Sikora and D.J. Costello, *A new SISO algorithm with application to turbo equalization*, Proc. IEEE International Symposium on Information Theory (ISIT), Sep 2005, pp. 2031–2035.

[56] B. Steingrimsson, Zhi-Quan Luo, and Kon Max Wong, *Soft quasi-maximum-likelihood detection for multiple-antenna wireless channels*, IEEE Transactions on Signal Processing **51** (2003), no. 11, 2710–2719.

[57] C. Studer, A. Burg, and H. Bolcskei, *Soft-output sphere decoding: algorithms and VLSI implementation*, IEEE Journal on Sel. Areas in Communications **26** (2008), no. 2, 290–300.

[58] P. Sun, C. Zhang, Z. Wang, C.N. Manchón, and B.H. Fleury, *Iterative receiver design for ISI channels using combined belief- and expectation-propagation*, IEEE Signal Processing Letters **22** (2015), no. 10, 1733–1737.

[59] K. Takeuchi, *Rigorous dynamics of expectation-propagation-based signal recovery from unitarily invariant measurements*, Proc. IEEE International Symposium on Information Theory (ISIT), June 2017, pp. 501–505.

[60] M. Tüchler, R. Koetter, and A.C. Singer, *Turbo equalization: principles and new results*, IEEE Trans. on Communications **50** (2002), no. 5, 754–767.

[61] M. Tüchler and A.C. Singer., *Turbo equalization: An overview*, IEEE Trans. on Information Theory **57** (2011), no. 2, 920–952.

[62] M. Tüchler, A.C. Singer, and R. Koetter, *Minimum mean squared error equalization using a priori information*, IEEE Trans. on Signal Processing **50** (2002), no. 3, 673–683.

[63] C. M. Vithanage, C. Andrieu, and R. J. Piechocki, *Approximate inference in hidden Markov models using iterative active state selection*, IEEE Signal Processing Letters **13** (2006), no. 2, 65–68.

[64] R. Wang and G. B. Giannakis, *Approaching MIMO channel capacity with soft detection based on hard sphere decoding*, IEEE Trans. on Communications **54** (2006), no. 4, 587–590.

[65] M. Witzke, S. Baro, F. Schreckenbach, and J. Hagenauer, *Iterative detection of MIMO signals with linear detectors*, Proc. 36th Asilomar Conference on Signals, Systems and Computers, vol. 1, Nov 2002, pp. 289–293 vol.1.

[66] C. Zhang, Z. Wang, C. N. Manchón, P. Sun, Q. Guo, and B. H. Fleury, *Turbo equalization using partial Gaussian approximation*, IEEE Signal Processing Letters **23** (2016), no. 9, 1216–1220.

Glossary

APP a posteriori probabilities. 2, 7, 8, 10, 12, 22, 24, 34

AWGN additive white Gaussian noise. XIV, 1, 15, 16, 49, 50

BEP block expectation propagation. 23, 27, 28, 30–32, 39, 53, 62, 63

BER bit error rate. 1, 16, 46

BP belief propagation. 10, 12

BT backward-trellis. 8

CHEMP channel hardening-exploiting message passing. 12, 13

CS channel shortening. 8

CSI channel state information. XIV, 7, 8, 16, 65

D-BEP double block expectation propagation. 39, 45, 46, 48, 52, 53, 55, 56, 63, 64

D-FEP double filter expectation propagation. 40, 45, 46, 48, 63, 64

D-KSEP double Kalman smoothing expectation propagation. 42, 45, 46, 48, 63, 64

DT doubled-trellis. 8

ECC error correction code. 1

EP expectation propagation. III–V, 4, 5, 10–13, 15, 18, 20, 25, 29, 35, 37, 42, 51, 53–55, 61

EP-F expectation propagation filter. 40

EPD expectation propagation detector. 53, 54

FEC forward error correction. 1

FEP filter expectation propagation. 25, 30–32, 40, 62, 63

FT forward-trellis. 8

GMP Gaussian message passing. 10, 11, 15, 29, 43, 46, 47, 64

GTA Gaussian tree approximation. 12, 13

i.i.d. independent and identically distributed. 29, 42, 55

ISI intersymbol interference. 1, 15

KL Kullback-Leibler. III, V, 18

KSEP Kalman smoothing expectation propagation. 27, 28, 30, 32, 41, 62, 63

LDPC low-density parity-check. 29, 42, 55

LLR log-likelihood ratio. XV, 34, 51

LLRs log-likelihood ratios. 33

LMMSE linear minimum mean square error. III–V, 3–5, 9–13, 15, 17–19, 25, 27–30, 35, 42, 45, 46, 55, 56, 61, 62

LTI linear time invariant. 4

MAP maximum a posteriori. 2, 7, 12

MCMC Markov chain Monte Carlo. 12

MIMO multiple-input multiple-output. III, V, 4, 5, 11–13, 49–53, 55, 56, 58, 61, 62, 64, 65, 125

MMSE minimum mean square error. 2, 9

MSE mean square error. 2, 9, 12

nuBEP non-uniform block expectation propagation. 39, 52

P-BEP projected block expectation propagation. 39, 45, 46, 48, 52–55, 63, 64

P-FEP projected filter expectation propagation. 40, 45, 46, 48, 63

P-KSEP projected Kalman smoothing expectation propagation. 41, 42, 45, 46, 48, 63

pdf probability density function. III, XV, 3–5, 7, 12, 19, 30, 36, 37

RS-BCJR reduced-state BCJR. 8

RSSD reduced-state sequence detection. 8

S/P serial to paralell. 50

SD sphere decoding. 12

SEP smoothing expectation propagation. 28, 29, 32, 41

SER symbol error rate. 7

SIC successive interference cancellation. 65

SISO single-input single-output. III, V, 4, 16, 17, 33, 49, 51, 61, 125

SPA sum-product algorithm. 10