(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification:
H04B 7/02 (2006.01)   H04L 1/02 (2006.01)

(21) International Application Number:
PCT/AU2006/000429

(22) International Filing Date: 31 March 2006 (31.03.2006)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant (for all designated States except US): COM-
MONWEALTH SCIENTIFIC AND INDUSTRIAL
RESEARCH ORGANISATION [AU/AU]; Limestone
Avenue, Campbell, ACT 2612 (AU).

(72) Inventors; and
(75) Inventors/Applicants (for US only): SUZUKI, Ha-
jime [JP/AU]; 149 Ray Road, Epping, 2121 (AU).
DANIELS, Graham, Ross [AU/AU]; 6 Bedford
Street, Willoughby, NSW 2068 (AU). HEDLEY, Mark
[AU/AU]; 1 Holdsworth Avenue, St Leonards, NSW 2065
(AU).

(74) Agent: SPRUSON & FERGUSON; GPO BOX 3898,
Sydney, New South Wales 2001 (AU).

(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,
AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN,
CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI,
GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE,
KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV,
LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI,
NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG,
SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US,
UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every
kind of regional protection available): ARIPO (BW, GH,
GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,
ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),
European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,
FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT,
RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA,
GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
—   with international search report

For two-letter codes and other abbreviations, refer to the "Guid-
ance Notes on Codes and Abbreviations" appearing at the begin-
ning of each regular issue of the PCT Gazette.

(54) Title: DECODING FREQUENCY CHANNELISED SIGNALS
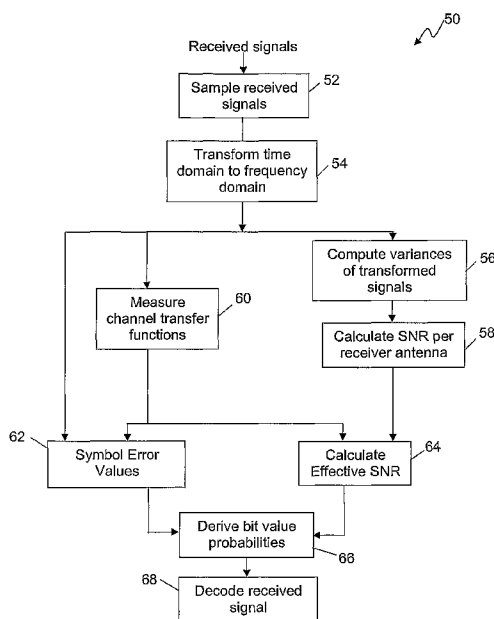
(57) Abstract: There is disclosed a method and receiver for decoding a data signal from analogue signals received at one or more
receiving antennas (24). The decoding is performed on the basis of bit value probabilities (66) derived from an effective signal to
noise ratio (ESNR) (64) and a respective symbol error value (SEV) (62) calculated for all of the one or more receiving antennas. The
ESNR is calculated utilising signal to noise ratios (SNRs) (58) per sub-channel and sub-channel transfer functions (60) measured for
each of the one or more receiving antennas (24). Also, the SEVs are calculated utilising the measured sub-channel transfer functions.

Decoding frequency channelised signals

Field of the invention

The present invention relates to digital communications and more particularly to decoding frequency channelised signals received by one or more antennas on the basis of bit value probabilities.

Background

Wireless communication systems are widely used for transmission of digital signals, for example in cellular phone communication and the transmission of digital television. One method of implementing a wireless communication system of this type is to use a singular antenna at each of the transmitter and receiver ends. Such systems generally operate satisfactorily in free space environments where there is a direct (line-of-sight) path from the transmitter to the receiver. However, in some environments, such as urban space where infrastructure creates obstructions to the line of sight between transmitter and receiver, the direct path may be partially or completely blocked, and the transmitted signal may undergo scattering and diffraction from such obstructions (obstacles) before it is received. Importantly, the effects of scattering and diffraction are often not homogeneous across the spectral bandwidth of the signal because the signal travels between the transmitter and receiver via multiple paths. As the distance of each path may vary, so too does the delay experienced by each of the signal's multiple path components. For the present purposes these effects are referred to as *multipath scattering*. If the transmitter, receiver and obstacles are in relative motion, then the results of such multipath scattering become time varying.

The delays introduced by multipath scattering may result in a transmitted symbol being received over a period longer than the transmitted symbol period. Thus, part of the symbol energy will interfere with the received energy of adjacent received symbols, known as *intersymbol interference*. If the effects of intersymbol interference are significant enough, they may introduce errors into the interpretation of adjacent symbols. As each symbol transmitted is susceptible to multipath scatter, so too is every symbol affected by intersymbol interference. A related problem occurs where signal components following different paths destructively interfere, resulting in *signal fading*.

An approach to alleviating this problem is the *channelisation* of data at the transmitter into multiple concurrent data streams (data sub-channels). One implementation providing multipath tolerance is orthogonal frequency division multiplex (OFDM) where a single channel is separated into frequency

2

sub-bands (sub-channels). Another approach is used in code division multiplex (CDM) systems that achieve code diversity by using multiple spreading codes.

Multiple-input multiple-output (MIMO) wireless communication systems are another recent approach of not only reducing the degrading effects of multipath scattering but also potentially increasing the spectral efficiency of the system. MIMO systems utilise the effects of spatially separating a plurality of antennas at both the transmitting and receiving ends of a communication system. There are a number of different methods for implementing a MIMO system which maximise different benefits (see A. J. Paulraj, D. A. Gore, R. U. Nabar, and H. Bolcskei, "*An overview of MIMO communications - A key to gigabit wireless*," Proceedings of the IEEE, vol. 92, no. 2, pp. 198-218, February 2004). For example, a diversity gain may be realised by transmitting differently encoded versions of the same data from each of the spatially separated transmitters in unison. Ideally, each of the transmitter-receiver combinations creates a propagation sub-channel which fades independently of the other combinations. By suitably combining and decoding the signals received at each of the receiver antennas, the resulting data stream exhibits less impairment due to fading than the equivalent single-input single-output (SISO) implementation without increasing transmission time or bandwidth.

In another implementation, a MIMO system may exploit the benefit of spatial multiplexing to increase capacity without increasing transmission power or bandwidth requirements. Such a gain is realised by splitting a serial signal into a plurality of independent signals which are then transmitted in parallel from each of the independent transmitters. The data rate of each of the parallel transmission signals is less than the original serial signal data rate, which has the additional benefit of reducing the effects of intersymbol interference within the parallel channel. In this implementation, the receiver system combines the information from the parallel received signals to recover the transmitted data at the original (higher) data rate.

In any channelised communications system, it is preferable that some form of error resistant coding is employed to allow errors in the received signal to be detected and corrected, or the data content of the signal to be estimated more reliably, particularly since some sub-channels typically suffer greater impairment than others. A Low Density Parity Check (LDPC) code is one such type of coding (see, for example, R. G. Gallager, "*Low-density parity-check codes*", IRE Transaction on

3

Information Theory, vol. 8, no. 1, pp. 21-28, January 1962). Parity check codes, as a general rule, work by combining a block of binary information bits with a block of check bits. Each check bit represents the modulo 2 sum of a prescribed selection of the binary bits which constitute the information digits. For example, if the data bits $[x_1, x_2, ..., x_5]$ have parity check bits $[x_6, x_7, x_8]$

5   appended to make a 8 digit coded block, x_6 may be defined as

$$x_6 = x_1 \oplus x_4 \oplus x_5$$

If the parity check equation defining each parity check bit is chosen correctly, then errors can be detected and corrected. The above parity check equation may be represented by the matrix:

10            $[\ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ |\ 1\ \ 0\ \ 0\ ]$

             $x_1\ \ x_2\ \ x_3\ \ x_4\ \ x_5\ |\ \ x_6\ \ x_7\ \ x_8$

A parity check code usually is derived from a number of such parity check equations, which are normally represented as the individual rows of a parity check matrix. The code may be described in

15   terms of a generator matrix which is used to generate code word vectors from data word vectors by modulo 2 matrix multiplication (also known as *mapping*).

A LDPC code comprises a number of parity check equations, which normally are represented as the individual rows of a parity check matrix. The LDPC matrix is dominated by 0's, with only a small

20   number of 1's. A regular LDPC code may be described as an (*a, b, c*) code whereby the matrix of the code has a block length of *a*, each column of the matrix contains a small fixed number of *b* 1's, and each row a small fixed number of *c* 1's. A regular LDPC code has the same number of 1's in each column or row, whereas for an irregular LDPC code, some variation is permitted. In a systematic code, the data bits are explicitly identifiable in the code words, but this is not always the

25   case. The modulo-2 dot product of a valid code word with any of the rows in the parity check matrix should equal zero.

The outline given above, and the following detailed description, refer to a systematic LDPC code in which each parity check bit is a function of the data bits only. It should be noted that the more

30   general cases of systematic codes in which parity check equations contain more than one check bit, or of non-systematic codes may be treated similarly, and are equally applicable coding approaches.

4

In the most basic (error detection) implementations, a data packet consisting of a stream of bits forming one or more data words is encoded by multiplying the data words by the generator matrix. At the receiver end, the message is checked for accuracy by verifying the modulo 2 product of each code word and the parity check matrix to be the zero vector. If there are no errors, the code words

5   may be multiplied by an inverse of the generator matrix to extract the original data words. An error correcting decoder uses the parity check results to find the valid code words which are "nearest" to the received code words.

Soft decision LDPC decoders accept a sequence of bit value (0,1) estimates and associated

10   correctness probability estimates, the pairs normally being combined as single values representing the estimated prior probabilities that the associated bit is a 1. These estimates correspond to the information and check (redundancy) bits of the received message, and they are employed by the decoder to generate a more reliable estimate of the message bits, normally via an iterative process.

15   It is therefore possible and desirable to increase the reliability of decoders (such as LDPC decoders) by providing more accurate probability estimates.

Summary

In a broad aspect, there is provided a method and receiver for decoding a data signal from analogue

20   signals received at one or more receiving antennas, said decoding being performed computationally on the basis of bit value probabilities derived from an effective signal to noise ratios (ESNR) and a respective symbol error value (SEV) for all said one or more receiving antennas, said ESNR being calculated utilising signal to noise ratios (SNRs) per sub-channel and measured sub-channel transfer functions for each of said one or more receiving antennas, and said SEVs being calculated utilising

25   said transfer functions.

There is further disclosed a method for decoding a data signal comprising the steps of: receiving one or more transmitted signals at each of one or more receiving antennas, each said transmitted signal having multiple frequency sub-channels containing data symbols; calculating a signal to noise ratio

30   (SNR) per sub-channel for all said one or more receiving antennas' respective received signal; measuring channel transfer functions for each of said one or more receiving antennas; calculating an effective signal to noise ratio (ESNR) for all said one or more receiving antennas utilising a respective said SNR and a respective said channel transfer function; calculating symbol error values

5

(SEVs) for all said one or more receiving antennas utilising a respective said channel transfer function and a respective estimated value of said data symbols; deriving bit value probabilities utilising said ESNR and a respective said SEV; and decoding said data signal utilising said derived bit value probabilities.

5

There is yet further disclosed a receiver for decoding a data signal comprising: one or more receiving antennas receiving one or more transmitted signals, each said transmitted signal having multiple frequency sub-channels containing data symbols; a circuit calculating a signal to noise ratio (SNR) per sub-channel for each said one or more receiving antennas' respective received signal; a

10 circuit measuring channel transfer functions for each of said one or more receiving antennas; a circuit calculating an effective signal to noise ratio (ESNR) for all said one or more receiving antennas utilising a respective said SNR per sub-channel and a respective said channel transfer function; a circuit calculating symbol error values (SEVs) for all said one or more receiving antennas utilising a respective said channel transfer function and a respective estimated value of said

15 data symbols; a circuit deriving bit value probabilities utilising said ESNR and a respective said SEV; and a decoder decoding said data signal utilising said derived bit value probabilities.

Preferably, calculating the SEVs includes determining the distance of the estimated symbol values from a predetermined ideal constellation point.

20

Further preferably, calculating the SNRs per sub-channel includes sampling each receiving antenna's received signal at each of a first period when there are no data symbols present and a second period when there is at least one data symbol present, determining the variance of the received signals over the duration of the first period and the variance of the received signals over the

25 duration of the second period for each receiving antenna, and calculating the SNR per sub-channel for each receiving antenna utilising said variances. Advantageously, the SNR per sub-channel is determined by determining the difference between the first period variance and the second period variance, and dividing said difference by the first period variance.

30 The decoding can utilise low density parity check decoding. Alternatively, the decoding can be Viterbi or turbo decoding.

The received signals can be converted from the time domain to the frequency domain before calculating the SNRs. The conversion, in one form, is performed by an inverse Fast Fourier transformation (IFFT) process.

5   Preferably, the sub-channels are encoded by orthogonal frequency division multiplex modulation.

Description of the drawings

Figure 1 is a block schematic of a MIMO communication system incorporating an LDPC encoder and corresponding decoder.

10

Figure 2 is a flow diagram of the steps of decoding a signal received in a MIMO system.

Figure 3 is a QPSK Gray labelled mapping constellation of symbols which may be transmitted by MIMO-OFDM.

15

Figure 4 is a screenshot of a system implementing the method of LDPC decoding using the SNR calculated according to the steps of **Figure 2**.

Detailed description

20   Where reference is made in any one or more of the accompanying drawings to steps and/or features, which have the same reference numerals, those steps and/or features have for the purposes of this description the same functions and/or operations, unless the contrary intention appears.

Introduction

25   In the following description, an implementation using orthogonal frequency division multiplexing (OFDM) in a MIMO communication system using LDPC encoding/decoding is given. However, it is to be understood that the invention applies generally to other frequency channelised communications systems that rely on decoding a received signal using estimates of bit probabilities.

30   OFDM is a communication technique often utilised in wireless communication systems. OFDM may be combined with arrays of antennas at both the transmitting and receiving ends to enhance the system capacity on frequency selective channels, and in such case is referred to as a MIMO-OFDM system. The use of MIMO-OFDM is presently being considered for use in the IEEE 802.11

Wireless LAN standards. In particular, MIMO-OFDM is proposed in an amendment to the standard to be known as 802.11n for higher throughput improvements to wireless communications. The 802.11n standard will build upon previously accepted standards such as 802.11a and 802.11g which utilise OFDM.

5

In a MIMO-OFDM system, the bandwidth available for communication is divided into frequency domain sub-channels that are orthogonal to one another at the chosen symbol rate. This is beneficial as it converts the frequency selective MIMO channel into a set of parallel, MIMO sub-channels, each of which is essentially frequency-flat. In OFDM the sub-channel carrier frequencies are spaced

10   in the frequency domain to ensure that the corresponding time domain signals are orthogonal on a symbol-by-symbol basis, whilst allowing the spectrum of each of the sub-carrier signals to overlap the spectra of signals in adjacent sub-channels.

Transmitter and receiver circuits

15   Referring to **Figure 1** there is shown a block schematic of the transmitter end 10 and the receiver end 11 of a MIMO-OFDM communication system. In the transmitter end 10, the data signal consisting of a binary data sequence 13 to be transmitted is output by a source 12. An LDPC encoder 14 is used to encode the data signal 13 to be transmitted. The encoded data signal 15, now containing additional check bits, is fed into a serial to parallel (S/P) converter 16. The encoded data

20   signal 15 is split (de-multiplexed) by the S/P converter 16 prior to transmission so that each de-multiplexed component data stream 17 is transmitted concurrently from a different one of the transmission antennas 22. Each of the component data streams 17 is mapped by a respective mapping device 18 into sets of $k\ 2^m$ = M-ary QAM modulation symbols 19, corresponding to the OFDM carriers, using Gray-labelling. The symbols 19 are converted into time domain OFDM

25   symbols 21 by respective F/T transformation (eg. inverse Fast Fourier Transform (IFFT)) circuits 20.

The OFDM symbol streams 21 are fed into time domain reconstruction circuits 23 to generate corresponding analogue radio frequency (RF) signals 21A. The reconstruction circuits 23 may

30   comprise any or all of a digital to analogue converter (DAC, not shown), a frequency translator (such as a mixer, not shown) and at least one frequency domain filter (not shown). In a basic implementation, the digital signal sequence corresponding to OFDM symbols 21 is firstly converted to a complex analogue signal by the DAC. A low pass filter (not shown) may be applied to the

DAC output to smooth the signal. The smoothed signal is then frequency translated, for example, by an image reject mixer coupled to a local oscillator (not shown). The frequency translation process produces a signal at the desired higher (e.g. radio) frequency range. In another implementation, the analogue complexity may be reduced by placing the DAC after the mixer, such that the frequency-translation occurs in the digital domain, although this somewhat increases the computational complexity of the signal reconstruction. It will be apparent to those skilled in the art that various different reconstruction circuits 23 may be utilized to generate the reconstructed RF signal 21A. The reconstructed RF signals 21A then are transmitted via the plurality of antennas 22.

At the receiver end 11, the transmitted analogue RF signals 21A corresponding to the concurrent OFDM symbols 21 are received via the plurality of antennas 24 and passed through respective time domain sampling circuits 25 whose function will be described below. The sampling circuits essentially perform the inverse task of the reconstruction circuits 23. The sampling circuit 25 may comprise any or all of a frequency translator (such as a mixer, not shown), filters (not shown) and an analogue-to-digital converter (ADC, not shown). The received analogue RF signals are frequency-translated to obtain lower, intermediate or baseband frequencies by the frequency translator, which may comprise a mixer coupled to a local oscillator. The frequency translated signal is passed through a filter to attenuate unwanted signal components. The translated, filtered signal is converted into the digital, sampled signal by the ADC. It will be apparent to those skilled in the art, that other configurations of sampling circuits may be substituted for the embodiment described.

The receiver end 11 utilises the following method to recover the data signal 13 within the signals transmitted from the transmitter antennas 22. However, not all the following steps need to be practised in accordance with the invention in its broadest form. The analogue signal received by each of the receiver antennas 24 and is sampled by the respective sampling circuits 25 during periods when there are no received symbols (i.e. 'noise only') and when there are received data symbols (i.e. 'data + noise'). The resultant sampled signals 25A are converted into the frequency domain by respective time/frequency (T/F) transformation circuits 26. The conversion to the frequency domain can be performed by applying a FFT process to the received time domain sampled signal. The sets of frequency domain samples 27 corresponding to OFDM sub-channel composite values due to *all* transmitters are sent to a MIMO detector 28. The MIMO detector 28 can be implemented as a discrete circuit, a programmed microprocessor circuit or as an application specific integrated circuit.

A number of processes occur during MIMO detection (as is described in detail with reference to **Figure 2**). In particular, the sampled frequency domain samples 27 are used to compute the variances for each frequency component of the received signal, and subsequently utilized to

5   calculate the corresponding Signal-to-Noise ratios (SNRs), Symbol Error Values (SEVs) and effective signal to noise ratios (ESNRs). Also, the samples 27 are used to estimate the symbol error values corresponding to each of the one or more possible transmitted symbol values. The MIMO detector 28 thus provides a soft decision for each component bit of each of the concurrently received OFDM symbols. The soft decision results are cyclically multiplexed by a multiplexer 30 (ie. a

10   parallel to serial (P/S) conversion) into a single composite stream 31 and passed into a LDPC decoder 32.

The LDPC decoder 32 takes the output of the MIMO detector 28 to form estimates of the transmitted information bits, which it outputs as a binary stream 33 of the received data bits to a sink

15   34. These estimates are more reliable than known arrangements.

In the example of **Figure 1**, a four transmitter antenna x four receiver antenna arrangement (10,11) is shown. It is also to be understood that other numbers of transmitters and receivers can be supported. For example, in the mathematical treatment (discussed later) the number of receiver

20   antenna 24 is required to be equal to or greater than the number of transmitter antenna 22.

Processing to derive bit value probabilities

Referring now to **Figure 2**, there is illustrated a flow diagram 50 of the steps taken in decoding the received signals according to an embodiment of the invention. To accurately decode the transmitted

25   binary data sequence 15 based upon the received and demodulated signal 27, it is important that the decoder 32 receives an accurate estimate of the reliability (and hence probability of correctness) of the estimate of each data or check bit.

As noted above with reference to **Figure 1**, the transmitted OFDM signals (21A) are received by a

30   plurality of antennas 24, and time sampled, via each of the plurality of sampling circuits 25 (step 52). The received signals 25A are sampled in the time domain during periods when there are no received symbols (i.e. 'noise only') and also when there are received symbols (i.e. 'data + noise'). Furthermore, the period of sampling of noise only may occur before or after a received packet. Each

of the parallel received sampled signals is then transformed into the frequency domain by the respective T/F circuits 26 (step 54).

The result of step 54 forms three subsequent process branches in **Figure 2**. In one branch, variances
5   per receiver per OFDM sub-carrier of the received signals are computed (step 56) from the transformed frequency domain signals 27. The variances of each signal received per receiver per OFDM sub-carrier are calculated during the 'noise only' period as well as the 'data + noise' period. The 'noise only' variance per receiver per OFDM sub-carrier may be expressed as $N(j, k)$, where j is an index of the receiver and k is an index of the OFDM sub-carrier frequency. Similarly, the 'data +
10   noise' signal variance may be expressed as $S(j, k)$. The computed variances are used to calculate the SNR per OFDM sub-channel per receiver antenna (step 58).

In a second branch of **Figure 2**, the frequency domain signals 27 are utilised to measure the propagation sub-channel transfer functions corresponding to each transmitter-receiver antenna pair
15   (step 60).

In the third branch, the frequency domain signals 27 resulting from step 54 are used – together with the measured propagation sub-channel transfer functions (S-CTFs) resulting from step 60 – to estimate the SEVs corresponding to the LDPC-encoded data symbols (step 62). The SEVs are the
20   distances of the estimated (I,Q) symbol values from the (predetermined/known) ideal constellation points.

The S-CTFs resulting from step 60 also are used with the calculated SNR per receiver per OFDM sub-carrier to calculate an estimated signal to noise ratio (ESNR) (step 64). The ESNR is a measure
25   of the signal to noise per transmitter antenna.

The SEVs resulting from step 62 and calculated ESNRs resulting from step 64 are utilised to derive a sequence of bit value probabilities (step 66) corresponding to the bit value sequence of the received OFDM signals per receiver. The bit value probabilities are passed to the LDPC decoder
30   32, and the received signal is decoded (step 68) to produce a best estimate of the transmitted binary data sequence 13.

11

With reference to the foregoing description of method 50, it will be apparent to those skilled in the art that other embodiments are achievable. For example, the step of measuring the propagation sub-channel transfer functions (step 60) cay be done at any time *after* the signals are received by the receiver 24 and before estimating the SEVs (step 62) or calculating the ESNR (step 64).

5   Furthermore, it should be evident that the sampling of the 'noise only' and 'data + noise' signals may be considered as independent of one another, and hence they may occur in any order. Also, transforming the sampled signals into the frequency domain (step 54) and computing the variances of the 'noise only' and 'data + noise' signals may occur in series or in parallel, and in any order, without affecting the calculated SNR. The sampling periods (steps 52) are each chosen to be long

10   enough that the power averages (variances) have sufficiently low uncertainty to give SNR estimates of acceptable accuracy.

Mathematical analysis

The following is a mathematical analysis of how the frequency domain signals 27 (step 54) and

15   measured transfer functions (step 60) are utilized in decoding the received signals 25A. Although the analysis is not strictly presented in the same order as the method steps of Figure 2, it will be apparent to those skilled in the art that the mathematical analysis can be implemented to perform the preferred arrangement illustrated in **Figure 2**.

20   Suppose n_t transmitters and n_r receivers are used in a MIMO-OFDM system, where $n_r \geq n_t$. Also assume that n_f OFDM data sub-carriers are used to transmit n_s OFDM symbols at each transmitter. The transmitted signal at the ith transmitter, kth OFDM sub-carrier, and lth OFDM symbol is $x(i, k, l)$, where $i = 1 \ldots n_t$, $k = 1 \ldots n_f$, and $l = 1 \ldots n_s$. The received data signal and noise signal at the jth receiver, kth OFDM sub-carrier, and lth OFDM symbol are $r(j, k, l)$ and $n(j, k, l)$, respectively,

25   where $j = 1 \ldots n_r$. The propagation sub-channel transfer function, $H(i, j, k)$, describes the transmission characteristics between the ith transmitter and jth receiver at the kth OFDM sub-carrier, where the channel is assumed to be stationary for the duration of n_s OFDM symbols. The received signal may thus be expressed as

30

$$r(j,k,l) = \sum_{i=1}^{n_t} H(i,j,k)x(i,k,l) + n(j,k,l) \qquad - (1)$$

The above equation can be written in a vector-matrix form as

12

$$\mathbf{r}(k,l) = \mathbf{H}(k)\mathbf{x}(k,l) + \mathbf{n}(k,l) \qquad —(2)$$

Recalling that n_r is the number of receivers and n_t is the number of transmitters, $\mathbf{r}(k, l)$ is an $n_r \times 1$ column vector whose jth element is $r(j, k, l)$, $\mathbf{H}(k)$ is an $n_r \times n_t$ matrix whose jth row and ith column element is $H(i, j, k)$, $\mathbf{x}(k, l)$ is an $n_t \times 1$ column vector whose ith element is $x(i, k, l)$, and $\mathbf{n}(k, l)$ is an $n_r \times 1$ column vector whose jth element is $n(j, k, l)$.

$\mathbf{H}(k)$ is normally estimated by measurements of the received signals produced by the transmission of known and unique reference signals from each of the transmitters. In this example, as is often convenient, $\mathbf{H}(k)$ will be assumed to be normalised such that var$[x(i,k,l)]$, averaged over l is considered to be unity.

If a zero-forcing (ZF) approach is used for the MIMO detection, the reconstructed estimate of the transmitted signals, $\mathbf{z}(k, l)$, is given by

$$\mathbf{z}(k,l) = \mathbf{W}(k)\mathbf{r}(k,l) = \mathbf{x}(k,l) + \mathbf{W}(k)\mathbf{n}(k,l) \qquad —(3)$$

where $\mathbf{W}(k)$ is an $n_t \times n_r$ matrix whose ith row and jth column element is $W(i, j, k)$ and $z(k,l)$ is an $n_t \times 1$ vector whose ith element is $z(i, k, l)$. $\mathbf{W}(k)$ is a left inverse of $\mathbf{H}(k)$, meaning that

$$\mathbf{W}(k)\mathbf{H}(k) = \mathbf{1} \qquad —(4)$$

A possible set of matrices, $\mathbf{W}(k)$ is given by

$$\mathbf{W}(k) = \left(\mathbf{H}(k)^H \mathbf{H}(k)\right)^{-1} \mathbf{H}(k)^H \qquad —(5)$$

The above set is the unique set of inverses if $n_r = n_t$. If $n_r > n_s$, there are an infinite number of left-inverses. In the latter case, this set (of generalised inverses) is optimal for the case of equal noise power at each receiver.

Assume that, after LDPC coding of the binary data stream to be transmitted, M-QAM sub-channel modulation is used, and at the receiver system 11, a zero-forcing MIMO orthogonalisation

13

(detection) process is employed (i.e. the MIMO detector 28), followed by a soft-decision LDPC decoder 32.

Bit value probabilities

5   The bit value probability (resulting from step 66) $P(b, i, k, l)$ that the bth bit within the M-QAM symbol sent from the ith transmitter, on the kth OFDM sub-carrier, and in the lth OFDM symbol is a 1, is given by

$$P(b,i,k,l) = \sum_{q \in A(b)} p(i,k,l,q) \qquad\qquad — (6)$$

10

where $A(b)$ is the set of indices in the range $q = 1 \dots 2^m$ for which the bth bit in the corresponding element of the M-QAM alphabet is 1. An example of $A(b)$ is given in **Figure 3** where $A(b)$ is given for Gray-labelled mapping of QPSK. In the above equation, $p(i, k, l, q)$ is the probability that the qth element is sent from the ith transmitter, at the kth OFDM sub-carrier, and at the lth OFDM symbol,

15   and is given by

$$p(i,k,l,q) = \frac{p'(i,k,l,q)}{\sum_{q=1}^{2^m} p'(i,k,l,q)} \qquad\qquad — (7)$$

where $p'(i,k,l,q)$ represents the probability density of the SEVs.

20

The probability density of the SEV is known, for the case of additive white Gaussian noise (AWGN), to be given by

$$p'(i,k,l,q) = \alpha(\eta'(i,k)) \cdot \exp\left(-\frac{d(i,k,l,q)^2}{2\eta'(i,k)} \right) \qquad — (8)$$

25   In the above equation, $\alpha(\eta'(i,k))$ is a scale factor. Therefore, to derive the bit value probabilities, $P(b, i, k, l)$ using the above equation, it is necessary to determine the SEVs $d(i, k, l, q)$ and the ESNR, $1 / \eta'(i,k)$.

Symbol Error Values

14

The SEVs, $d(i,k,l,q)$, of the M-QAM symbol transmitted from the ith transmitter, on the kth OFDM sub-carrier, in the lth OFDM symbol, corresponding to the qth element of the M-QAM alphabet is given by

$$d(i,k,l,q) = |z(i,k,l) - a(q)| \qquad\qquad — (9)$$

In the above equation, $z(i,k,l)$ is the sub-channel symbol value estimate using the zero-forcing process, and $a(q)$ is the value of the qth element (symbol) in the chosen M-QAM alphabet, where $q = 1 \dots 2^m$. Under the assumption of a normalized $\mathbf{H}(k)$, the $a(q)$ are scaled such that var[a(q)] is unity.

Effective Signal to Noise Ratio

The ESNR is the reciprocal of the effective noise factor, which is the noise variance per dimension of signal space (for a complex scalar QAM signal, the dimensionality is 2 corresponding to the independent real and imaginary components), consistent with the scaling of the SEVs. Since the zero forcing process forms weighted sums of the receiver outputs, the effective noise factor is given by:

$$\eta'(i,k) = \sum_{j=1}^{n_r} \left[|W(j,i,k)|^2 \cdot \eta(j,k) \right] \qquad\qquad — (10)$$

where $\eta(j,k)$ is the noise variance per dimension of signal space at the jth receiver and kth OFDM sub-carrier, and is given by

$$\eta(j,k) = \frac{\sum_{i=1}^{n_t} |H(i,j,k)|^2}{\gamma \cdot snr(j,k)} \qquad\qquad — (11)$$

where γ (=2 for this case of complex scalars) is the dimensionality of the signal space. The sum in the above equation corresponds to the total signal power (signal variance) of the kth OFDM sub-carrier at the jth receiver, due to all transmitters. To determine the noise variance according to the above equation, it is also necessary that an estimate of the SNR, $snr(j,k)$ be calculated.

Signal to Noise Ratio

The SNR per OFDM sub-carrier per receiver is given by

$$snr(j,k) = \frac{S(j,k) - N(j,k)}{N(j,k)} \qquad\qquad -(12)$$

where $S(j,k)$ and $N(j,k)$ have been previously defined as the variance of 'data + noise' signals and 'noise' signal, respectively, and the variances are of complex signals averaged over time.

The bit value probabilities $P(b, i, k, l)$ of the data bits and LDPC check bits are fed into the LDPC decoder 32 which uses them to generate a better estimate of the sequence of data bits actually transmitted. This estimate makes use of the data redundancy introduced by the LDPC check bits.

Example

Referring now to **Figure 4** there is illustrated the received signal characteristics corresponding to a MIMO-OFDM packet transmission. Referring also to Appendix A, there is provided an embodiment of the present invention implemented in MATLAB™ code, which when practised gives results consistent with those shown in **Figure 4**.

In the example shown, an aggregate data rate of transmission of 486 Mbps is used. A configuration of four transmitters 22 and four receivers 24 utilise a 40 MHz bandwidth comprising 108 OFDM carriers. The carriers are modulated using 64 QAM to achieve 12 bps/Hz. The 16 graphs 120 shown on the left of **Figure 4** illustrate the frequency response of each of the links between transmitter-receiver pairs in a MIMO configuration of 4 transmitters and 4 receivers. The four graphs 130 shown on the right are the 64 QAM constellations reconstructed from the received signals by the zero forcing process. It can be seen that system and environmental noise corrupts the received symbols and deviates them from the true 64 QAM symbol constellation. The received signal was used in the embodiment described above to estimate the SNR per receiver per sub-carrier and from these estimates to derive the ESNR per data sub-channel per sub-carrier, and thence to provide more reliable estimation of bit error probabilities to an LDPC decoder. The data bit sequences produced by the LDPC decoder from this information contained no errors, based on a comparison against the transmitted data signal.

For the LDPC decoder tested, that utilised the standard belief propagation decoding algorithm, it was empirically found that a sampling period of 20 symbols was sufficient to accurately determine the transmitted signal. Here an irregular LDPC matrix of size 11664 × 23328 was implemented. The LDPC decoder tested utilised the standard belief propagation decoding algorithm.

5

Discussion

It should be noted that, although the description of the embodiment refers to Additive White Gaussian Noise (AWGN), other embodiments are equally applicable to cases where the received signal is contaminated by noise or interference having different statistics.

10

The foregoing describes only the use of LDPC, but the method can be equally applied to decoding other forms of forward error correction, such as using Viterbi or turbo decoding. For example, the LDPC encoder 14 can be replaced with a convolutional code encoder plus space-frequency interleaver, while the LDPC decoder 32 can be replaced with a corresponding space-frequency de-

15 interleaver plus a soft-decision Viterbi decoder (not shown). If the soft-decision Viterbi decoder requires log-likelihood ratio as an input, the bit value probability can be converted into the log-likelihood ratio $L(b, i, k, l)$ by

$$L(b,i,k,l) = \log \frac{P(b,i,k,l)}{1 - P(b,i,k,l)} \qquad - (13)$$

20

In another implementation, a parallel concatenated code encoder plus a space-frequency interleaver can replace the LDPC encoder 14, while a corresponding space-frequency de-interleaver plus a turbo decoder replaces the LDPC decoder 32 (not shown).

25 In the preceding description, the preferred embodiments describe measuring the 'noise only' and 'signal + noise' variances per sub-carrier at each receiver, and calculating the ESNRs per sub-carrier per transmitter using these variances in combination with the (measured) propagation sub-channel transfer functions. However, it will be appreciated by those persons skilled in the art that the order of the steps described for the preferred embodiment is not limiting, and similar approaches may

30 yield the same or similar outcome. For example, provided that the propagation sub-channel transfer functions gains are determined first, the noise and signal plus noise variances may be measured at the output of the zero-forcing process MIMO detector, and the effective SNRs per transmitter

evaluated directly. It is clear that this approach involves equivalent calculations and so will give the same estimates for the effective SNRs per sub-carrier per transmitter and hence will provide the decoder with an equally accurate estimate of bit value probabilities.

5   It is also to be understood that techniques which do *not* rely on the presence of a 'noise only' period may be used as an alternative to the approach described above for estimating the noise variances and still fall within the broad scope of the invention. As an example, the 'noise only' variances may be estimated with the aid of portions of the received signal which are know to be repetitive or predictable, by measuring the differences between those receiver sample values corresponding to

10   various distinct occurrences of the same received signal (values). This approach may be of value in situations involving streaming (e.g. digital television broadcasts), rather than burst transmission, since almost all such transmissions already carry known synchronization/training information suitable for this noise variance estimation process.

18

Appendix

The following MATLAB™ code may be used when implementing the method described
hereinbefore in a programmable microprocessor.

5

The material contained in the MATLAB™ code set out below is subject to copyright protection. The
copyright owner has no objection to anyone who requires a copy of the program disclosed therein
for purposes of understanding or analyzing the invention, but otherwise reserves all copyright rights
whatsoever. This includes making a copy for any other purposes including the loading of a
10   processing device with code in any form or language.

```
      % mimo_ofdm_ldpc_demo
      %
      % This Matlab code demonstrates the implementation of MIMO-OFDM-LDPC
15    % packet transmission at 486 Mbps data rate using 4 transmitters, 4
      % receivers, 40 MHz bandwidth, 108 OFDM data carriers, and 64QAM,
      % achieving 12 bps/Hz.
      %
      % An LDPC matrix with size 11664 x 23328 coving 6 OFDM symbols is
20    % used.
      %
      % A packet consists of long AGC settler, MIMO preamble, 10 MIMO channel
      % estimates, 30 OFDM data symbols, 5 dummy OFDM data symbols, and 30
      % zero signal for noise measurement.
25    %
      % init_agc
      %    initialises AGC control
      % ieee802_11n_fi_long_sequence
      %    returns channel estimation training sequence defined by TGn Sync draft
30    %    specification of IEEE 802.11n
      % ieee802_11a_pilot_scrambler
      %    OFDM pilot scrambler sequence defined by IEEE 802.11a
      % siso_mqam_parameters
      %    defines 64QAM signal constellation and A(b), set of indices for which
35    %    the bth bit in the corresponding element of the 64QAM alphabet is 1
      % ieee802_11n_interleaving_40MHz
      %    interleaving indices defined by TGn Sync draft specification of IEEE
      %    802.11n
      % LDPC_code_23328.mat
40    %    defines LDPC parity check matrix, encoding matrix, and associated
      %    parameters
      % LDPC_encodeMaxWt2
      %    LDPC encoder
      % MIMOopen
45    %    initialises ADC/DAC-Matlab interface
      % MIMOsetupDAC
      %    load generated MIMO-OFDM-LDPC packet signal to DAC for transmission.
      % MIMOreadADC
      %    capture signals at ADC
```

19

```
% LDPC_decode
%     perform LDPC decoding

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% initialisation
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
MIMOopen;                       % initialise MIMO ADC/DAC-Matlab interface
ntx=4;                          % number of Tx
nrx=4;                          % number of Rx
mde=8;                          % IEEE 802.11a transmission mode (1, 3, 5, 6,
or 8)
init_agc                        % initialise AGC control
mnm=64;                         % MQAM number
crt=3/4;                        % coding rate
bps=log2(mnm);                  % bits per MQAM symbol
spb=max(bps/2,1);               % spatial parsing bit
nof=30;                         % number of OFDM frames
nob=nof*crt*bps*108*ntx;        % number of information bit
ncd=nob/crt;                    % number of coded bits
nos=ncd/bps/ntx;                % number of symbols sent
% IEEE 802.11n long sequence, time domain, vector format
[lsq,lsf,lqx,lfx]=ieee802_11n_fi_long_sequence;
cts=lqx(33:192,:);              % channel training sequence
ifo=[1:4:125].';
ifi=[ifo ifo+1 ifo+2 ifo+3];    % interleaved frequency index for 128 fft
index
ifx{1}=[[3:4:55] [60:4:112]].'; % interleaved frequency index for 114 sub-
carriers
ifx{2}=[[4:4:56] [61:4:113]].';
ifx{3}=[[1:4:57] [58:4:114]].';
ifx{4}=[[2:4:54] [59:4:111]].';
itc=[1 2 3 4; 2 3 4 1; 3 4 1 2; 4 1 2 3].'; % rotation for channel estimate
lls=length(lsq);                % length of total long training sequence
lct=length(cts);                % length of channel training sequence
noc=128;                        % length of symbols repeated for frame
detection
pid=[6 34 48 67 81 109];        % index of pilot
pif=[6 34 48 [67 81 109]+3].';  % index of pilot out of 117 sub-carriers
psc=ieee802_11a_pilot_scrambler; % IEEE 802.11a pilot scrambler
pso=[1 1 -1 1 -1 1].';          % pilot subcarrier
% IEEE 802.11n data carrier index out of 114 carriers
dci=[1:5 7:33 35:47 49:66 68:80 82:108 110:114];
% SISO MQAM parameters
[cst,c2b,zi0,zi1]=siso_mqam_parameters(mnm);
% IEEE 802.11n TGnSync interleaving matrix
[fpm,spm,tpm]=ieee802_11n_interleaving_40MHz;
idi=fpm{mde};
idk=spm{mde};
cbo=length(idi);                % coded bits per OFDM symbol
idj=zeros(cbo,ntx);
for itx=1:ntx
    idj(:,itx)=tpm{mde,itx};
end
odb=zeros(cbo,1);               % original data bit
fpd=zeros(cbo,1);               % first permutation frame
```

20

```
       spd=zeros(cbo,1);              .       % second permutation frame
       tpd=zeros(cbo,1);                      % third permutation frame
       load LDPC_code_23328.mat              % 1/2 rate LDPC
       MaxDecodeLoops = 80;
   5   InfoBits = cols - rows;
       Decoded = zeros(InfoBits, 1);
       q1mn = zeros(MaxWeightCol,cols);       % for LDPC_decode
       TentDec = int32(zeros(1,cols));        % for LDPC_decode
       CheckSumErrs = int32(zeros(1,1));      % for LDPC_decode
  10   DecodedBits = int32(zeros(1,cols-rows));  % for LDPC_decode
       IMaxIter = int32(MaxDecodeLoops);      % for LDPC_decode
       Iters = int32(0);                      % for LDPC_decode
       CodedI = int8(zeros(1,cols));          % for LDPC_encode
       ldc=cols;                              % columns
  15   ldr=rows;                              % rows
       % 3/4 puncturing index
       pci=repmat([1 2 3 6 7 8 9 12 13 14 15 18].',1,nob*2/18)+repmat([0:18:(nob*2-
       18)],12,1);
       pci=pci(:);
  20   pch=repmat([4 5 10 11 16 17].',1,nob*2/18)+repmat([0:18:(nob*2-18)],6,1);
       pch=pch(:);
       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
       %
       % construction of MIMO-OFDM-LDPC packet
  25   %
       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
       % AGC settler
       ags=repmat(lsq,8,4);
       % preamble, baseband time domain
  30   pra=repmat(lsq,4,4);
       % MIMO channel estimates
       mce=repmat([...
           cts;
           cts(:,[2 3 4 1]);
  35       cts(:,[3 4 1 2]);
           cts(:,[4 1 2 3])]],10,1);
       % generation of data part
       tdt=zeros(nob,1);                      % information bits
       cdt=zeros(nob*2,1);                    % coded transmitted bits
  40   pdt=zeros(ncd,1);                      % punctured transmitted bits
       mdd=zeros(spb,ntx,ncd/spb/ntx);       %
       mdt=zeros(ncd/ntx,ntx);               % parallel data
       idt=zeros(ncd/ntx,ntx);               % interleaved transmitted data
       ddt=zeros(ntx,nos);                   % index for MQAM constellation
  45   mst=zeros(nos,ntx);                   % transmitted MQAM waveform
       msr=zeros(nos,ntx);                   % received MQAM waveform
       ost=zeros(nos*(160/108),ntx);        % transmitted baseband time domain signals
       tdt=round(rand(nob,1));              % random source data
       % LDPC encoding
  50   id2=[1:ldr];
       id3=[1:ldc];
       for idf=1:(nof/6)
         % encoding
         LDPC_encodeMaxWt2(int8(tdt(id2)),CodedI,IWeightRow,IIndexCol);
  55     cdt(id3)=double(CodedI)';
         id2=id2+ldr;
         id3=id3+ldc;
       end
```

```
% puncturing
pdt=cdt(pci);
% serial to parallel
idt=reshape(pdt,ntx,length(pdt)/ntx).';
% 64QAM modulation
for itx=1:ntx
    ddt=reshape(idt(:,itx),bps,nos);

mst(:,itx)=cst(ddt(1,:)*32+ddt(2,:)*16+ddt(3,:)*8+ddt(4,:)*4+ddt(5,:)*2+ddt(6,:)
+1);       % modulated signal with power 1
end
% for each OFDM frames
for idf=1:nof
    % index for pilot scrambler
    ips=mod(idf-1,127)+1;
    % scrambled pilot
    pcc=repmat(pso*psc(ips),1,ntx);
    % extract OFDM frame
    odt=mst((idf-1)*108+1:108*idf,:);
    % insert pilots (currently zeros) 108+6+3=117
    oft=[...
        odt( 1: 5,:);       %
        pcc(1,:);           % pilot 1
        odt( 6:32,:);       %
        pcc(2,:);           % pilot 2
        odt(33:45,:);       %
        pcc(3,:);           % pilot 3
        odt(46:54,:);
        zeros(3,ntx);       % DC
        odt(55:63,:);
        pcc(4,:);           % pilot 4
        odt(64:76,:);       %
        pcc(5,:);           % pilot 5
        odt(77:103,:);      %
        pcc(6,:);           % pilot 6
        odt(104:108,:)];    %
    % 128 point ifft as described in 802.11n Figure 67, Section
    % 11.2.1.5.9 * oft(59,:) should be zeros
    oet=[...
        zeros( 1,ntx);
        oft(60:end,:);
        zeros(11,ntx);
        oft( 1:58,:)];
    % ifft
    ott=ifft(oet,128);
    % cyclic extension (overlapping by one sample as described in 802.11a
    % ignored
    ett=[...
        ott(97:end,:);
        ott];
    % time domain baseband signal
    ost((idf-1)*160+1:160*idf,:)=ett;
end
% transmitted signal, baseband
tsp=[...
    ags;                  % AGC settler
    pra;                  % preamble
    mce;                  % MIMO channel estimates
```

```
        ost zeros(size(ost,1),4-ntx); % data
        ost(1:160*5,:) zeros(160*5,4-ntx); % for whatever reason, last 2nd sample
   gets corrupted, dummy data inserted
        zeros(160*30,4)];    %
   tsb=[...
        tsp;
        zeros(16,4)];
   % 112 Msps upsampled signal
   usp=upsample(tsb,3);
   % low pass filter
   bpr=fir1(100,1/3);
   fts=filter(bpr,1,usp);
   % shifting in frequency by 28 MHz
   spr=repmat(exp(j*2*pi*[0:size(fts,1)-1].'/4),1,4);
   sts=fts.*spr;
   % take only the real part to send to IF
   str=real(sts);
   % DAC is set up so that the rms of signal (std) is
   % 1/4 of maximum range of DAC
   stx=std(str(200+[1:320*3],1));              % std
   % create signed 12 bit version
   dtx=round((str/stx)*(2047/4));
   % clipping
   dtx=min(dtx,2047);
   dtx=max(dtx,-2047);
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   %
   % load the MIMO-OFDM-LDPC packet data into DAC
   %
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   flg=MIMOsetupDAC(dtx(:,4).',dtx(:,3).',dtx(:,2).',dtx(:,1)',0);
   if flg~=0
       error('DAC was not setup properly')
   end
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   %
   % capture MIMO-OFDM-LDPC packet data into ADC
   %
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   acq=71680;                              % number of samples to be captured
   [flg dt1 dt2 dt3 dt4]=MIMOreadADC(acq,0);
   if flg~=0
     MIMOclose
     error('Could not perform proper capture')
   end
   % captured IF signal
   srr=double([dt1(:) dt2(:) dt3(:) dt4(:)]);
   % shifting in frequency by -20 MHz
   spr=repmat(exp(-j*2*pi*[0:size(srr,1)-1].'*1/4),1,4);
   srs=srr.*spr;
   % low pass filter
   bpr=fir1(100,1/3);
   frs=filter(bpr,1,srs);
   % 112/3 Msps downsampled signal
   rsb=downsample(frs,3)*3;
   idp=3971;                               % index of the beginning of packet, when using
   trigger
   % extract complete packet
```

23

```
      rsp=rsb((idp-lls*4):(idp+160*4*10+nof*160-1),:)*10^(-4);
      % extract noise measurement part, 30 OFDM symbols
      rnp=rsb(idp+160*4*10+nof*160+160*5+[1:160*30],:)*10^(-4);
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 5    %
      % MIMO-OFDM channel estimation
      %
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      mce=rsp(lls*4+[1:160*4*10],:);      % channel estimation measured part
10    lsz=zeros(128,4);                   %
      for itx=1:4
         lsz(:,itx)=fft(lqx(65:192,itx))';
      end
      hce=zeros(128,1);
15    hx2=zeros(128,4,4,10);              %
      cep=zeros(128,4);
      for idf=1:10                        % 10 channel estimates
         for itp=1:4                      %  4 estimate parts
            cep=mce(640*(idf-1)+160*(itp-1)+[33:160],:);
20          for itx=1:4                   % for each part, combination of 16 MIMO
      channels
               for irx=1:4
                  hce=diag(lsz(:,itc(itp,itx)))*fft(cep(:,irx));
                  hx2(ifi(:,itc(itp,itx)),irx,itx,idf)=hce(ifi(:,itc(itp,itx)));
25             end
            end
         end
      end
      % measured 10 channel estimation
30    hxx=[...
         hx2(71:127,:,:,:);
         hx2( 3:59,:,:,:)];
      % measured 10 channel estimation, vector version
      hx3=zeros(114,16,10);
35    for itx=1:4
         hx3(:,(itx-1)*4+[1:4],:)=squeeze(hxx(:,:,itx,:));
      end
      % phase correction
      % for each MIMO sub-channel and for each OFDM sub-carrier, the
40    % phase should be linearly changing.  Derive phase slope.
      psp=zeros(114,16);      % phase slope
      pbs=zeros(114,16);      % phase bias
      tx1=[1:10].';           % time index
      for idf=1:114
45       for ich=1:16
            tp2=squeeze(unwrap(angle(hx3(idf,ich,:))));
            [pa1,pb1]=linear_fit(tx1,tp2);
            %pm1=polyfit(tx1,tp2,1);
            psp(idf,ich)=pa1;
50          pbs(idf,ich)=pb1;
         end
      end
      % average slope, remove four outliners out of 16
      ps4=zeros(114,1);
55    for idf=1:114
         [tp4,id4]=sort(psp(idf,:));
         ps4(idf)=mean(tp4(3:14));
      end
```

24

```
% determine frequency phase slope
fax=[1:58 62:117].';     % frequency index
[pa1,pb1]=linear_fit(fax,ps4);
% model phase slope for each frequency, per OFDM symbol
% ms1 is only used to correct phase offset during channel estimation
ms1=(pa1*fax+pb1)/4;
% estimated channel, with phase offset correction, at the first
% OFDM data symbol,
hxz=zeros(114,4,4,10);
for idf=1:10                      % 10 channel estimates
    for itp=1:4                   % 4 estimate parts
        for itx=1:4               % for each Tx
            for irx=1:4           % for each Rx
                ifq=ifx{itc(itp,itx)};
                hxz(ifq,irx,itx,idf)=hxx(ifq,irx,itx,idf).*exp(j*((5-itp)+(10-
idf)*4)*ms1(ifq));
            end
        end
    end
end
% simple averaging
h_x=mean(hxz,4)/4;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% conversion of data signal into frequency
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% extract data portion
osr=rsp(lls*4+160*4*10+[1:nof*160],1:nrx);
% for each OFDM frames
etr=zeros(160,nrx);                         % time domain 160 samples
otr=zeros(128,nrx);                         % time domain 128 samples
oer=zeros(128,nrx);                         % frequency domain 128 samples
ofr=zeros(114,nrx);                         % frequency domain 114 samples
oxr=zeros(114,4,nof);                       % frequency domain data
id1=[1:160];
% convert to frequency domain data
for idf=1:nof
    % extract current OFDM frame
    etr=osr(id1,:);
    id1=id1+160;
    % remove cyclic prefix
    otr=etr(33:160,:);
    % FFT
    oer=fft(otr);
    % remove zeros
    ofr=[...
        oer(71:127,:);
        oer(3:59,:)];
    % store data
    oxr(:,:,idf)=ofr;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% conversion of noise into frequency
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
oxn=zeros(114,4,30);                         % frequency domain noise
```

25

```
    id1=[1:160];
    % convert to frequency domain data
    for idf=1:30
        % extract current OFDM frame
5       etr=rnp(id1,:);
        id1=id1+160;
        % remove cyclic prefix
        otr=etr(33:160,:);
        % FFT
10      oer=fft(otr);
        % remove zeros
        ofr=[...
           oer(71:127,:);
           oer(3:59,:)];
15      % store data
        oxn(:,:,idf)=ofr;
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %
20  % estimation of signal to noise power per receiver and per sub-carrier
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    snx=zeros(114,4);
    nvx=zeros(114,4);
    for idf=1:114
25      spw=var(squeeze(oxr(idf,:,:)).');       % signal power
        npw=var(squeeze(oxn(idf,:,:)).');       % noise power
        snx(idf,:)=(spw-npw)./npw;              % SNR per Rx per OFDM carrier, Eqtn 12
        % noise variance per Rx per OFDM carrier
        nvx(idf,:)=sum(abs(squeeze(h_x(idf,:,:)).').^2)./(2*snx(idf,:));
30  end
    % extract pilot only, corrected for polarity
    psx=zeros(6,4,nof);
    for idf=1:nof
        % index for pilot scrambler
35      ips=mod(idf-1,127)+1;
        % scrambled pilot
        psx(:,:,idf)=oxr(pid,:,idf)*psc(ips);
    end
    % extract phase offset information, corrected for wrapping
40  pox=zeros(6,4,nof);
    for ipd=1:6
        for irx=1:4
           pox(ipd,irx,:)=unwrap(angle(psx(ipd,irx,:)));
        end
45  end
    % derive linear model curves (phase vs time)
    psl=zeros(6,4);          % phase slope
    pbs=zeros(6,4);          % phase bias
    tx2=[1:nof].';           % time axis
50  for ipd=1:6
        for irx=1:4
           [pa1,pb1]=linear_fit(tx2,squeeze(pox(ipd,irx,:)));
           psl(ipd,irx)=pa1;
           pbs(ipd,irx)=pb1;
55      end
    end
    % derive linear model curve (phase vs frequency)
    ps2=zeros(6,1);          % phase slope
```

26

```
      for ipd=1:6
        [tp3,id3]=sort(psl(ipd,:));
        ps2(ipd)=mean(tp3(2:3));
      end
 5    % determine frequency phase slope
      [pal,pbl]=linear_fit(pif,ps2);
      % model phase slope for each frequency, per OFDM symbol
      % ms2 is used during data transmission
      ms2=pal*fax+pbl;
10    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %
      % Demodulation of MIMO-OFDM-LDPC packet
      %
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15    ogr=zeros(114,ntx);                 % equalisation
      msr=zeros(108*nof,ntx);             % data symbols, total
      idr=zeros(ncd/ntx,ntx);             % interleaved received soft matrics
      idz=zeros(ncd/ntx,ntx);             % interleaved received hard decided data
      mdd=zeros(spb,ntx,ncd/spb/ntx);     %
20    mdr=zeros(ncd/ntx,ntx);             % parallel received data
      pdr=zeros(ncd,1);                   % punctured received data
      pdz=zeros(ncd,1);                   % punctured received data
      cdr=zeros(nob*2,1);                 % coded received bits
      cdz=zeros(nob*2,1);                 % coded received bits
25    tdr=zeros(nob,1);                   % received information bits
      hxc=zeros(114,nrx,ntx);             % estimated channel at particular OFDM symbol
      wmx=zeros(ntx,nrx);                 % inverse of H matrix
      wmz=zeros(114,ntx,nrx);             % inverse of H matrix, all sub-carrier
      hcc=zeros(nrx,ntx);                 % H matrix at one sub-carrier
30    ocr=zeros(nrx,1);                   % received signal at one sub-carrier
      emx=zeros(108,ntx,30);              % error distance
      % for each OFDM data symbol
      for idf=1:nof
        % current estimated channel, phase corrected for the idf th OFDM
35      % symbol
        hxc=h_x(:,1:nrx,1:ntx).*repmat(exp(j*(idf-1)*ms2),[1 nrx ntx]);
        % current received signals, frequency domain
        ofr=oxr(:,:,idf);
        % zero-forcing equalisation
40      for iff=1:114
          hcc=squeeze(hxc(iff,:,:));            % channel transfer function for current
      sub-carrier
          ocr=ofr(iff,:);                       % received signal for current sub-
      carrier
45        wmx=inv(hcc'*hcc)*hcc';               % w matrix, Eqtn 5
          wmz(iff,:,:)=wmx;                     % store w matrix
          ogr(iff,:)=(wmx*ocr(:)).';            % zero-forcing, Eqtn 3
        end
        % remove pilots and collect OFDM frames
50      msr(((idf-1)*108+1):108*idf,:)=ogr(dci,:);
        % detection / bit value probability estimate
        for idd=1:108
          % index for current received bits
          tid=(idf-1)*108*bps+(idd-1)*bps+1;
55        % data carrier index
          iff=dci(idd);
          % current recovered constellation
          rzs=ogr(iff,:);
```

```
         % current w matrix
         wmx=squeeze(wmz(iff,:,:));
         % for each tx
         for itx=1:ntx
            % eucredean distances, Eqtn 9
            dsz=abs(rzs(itx)-cst).^2;
            %ESNR, Eqtn 10
            vcn=sum(abs(wmx(itx,:)).^2.*nvx(iff,:));
            % probability density of SEV, Eqtn 8
            prt=exp(-dsz/(2*vcn));
            alp=sum(prt);
            prb=prt/alp;                    % probability, Eqtn 7
            % bit value probability
            for idb=1:bps
               %  Eqtn 6
               sox(idb,itx)=sum(prb(zil(idb,:)));
            end
         end
         idr(tid:tid+bps-1,:)=sox;
      end
   end
   % parallel to serial
   pdr=reshape(idr.',ncd,1);
   % depuncturing soft decision data
   cdr(pch)=0.5;
   cdr(pci)=pdr;
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   %
   % LDPC decoding
   %
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   id2=[1:ldr];
   id3=[1:ldc];
   for idf=1:(nof/6)
      LDPC_decode(...
         cdr(id3).',...
         qlmn,...
         IWeightCol,...      % defined in LDPC_Test_Code.mat
         IWeightRow,...      % defined in LDPC_Test_Code.mat
         RICoffset,...       % defined in LDPC_Test_Code.mat
         TentDec,...         %
         CheckSumErrs,...
         IIndexCol,...       % defined in LDPC_Test_Code.mat
         IInfoBitPosn,...    % defined in LDPC_Test_Code.mat
         DecodedBits,...     %
         IMaxIter,...
         Iters);
      tdr(id2)=DecodedBits.';
      id2=id2+ldr;
      id3=id3+ldc;
   end
   % count number of bit errors
   nbe=sum(xor(tdt,tdr))
   MIMOclose;
```

28

Claims:

1.   A method for decoding a data signal from analogue signals received at one or more receiving antennas, said decoding being performed on the basis of bit value probabilities derived from an effective signal to noise ratio (ESNR) and a respective symbol error value (SEV) for all said one or more receiving antennas, said ESNR being calculated utilising signal to noise ratios (SNRs) per sub-channel and measured sub-channel transfer functions for each of said one or more receiving antennas, and said SEVs being calculated utilising said transfer functions.

2.   A method for decoding a data signal comprising the steps of:
        receiving one or more transmitted signals at each of one or more receiving antennas, each said transmitted signal having multiple frequency sub-channels containing data symbols;
        calculating a signal to noise ratio (SNR) per sub-channel for each said one or more receiving antennas' respective received signal;
        measuring channel transfer functions for each of said one or more receiving antennas;
        calculating an effective signal to noise ratio (ESNR) for all said one or more receiving antennas utilising a respective said SNR per sub-channel and a respective said channel transfer function;
        calculating symbol error values (SEVs) for all said one or more receiving antennas utilising a respective said channel transfer function and a respective estimated value of said data symbols;
        deriving bit value probabilities utilising said ESNR and a respective said SEV; and
        decoding said data signal utilising said derived bit value probabilities.

3.   A method according to claim 2, wherein the step of calculating said SEVs includes determining the distance of said estimated symbol values from a predetermined ideal constellation point.

4.   A method according to claim 2 or claim 3, wherein the step of calculating said SNRs per sub-channel includes sampling each said receiving antenna's received signal at each of a first period when there are no data symbols present and a second period when there is at least one data symbol present, determining the variance of said received signals over the duration of the first period and the variance of said received signals over the duration of the second period for each receiving

antenna, and calculating said SNR per sub-channel for each receiving antenna utilising said variances.

5.   A method according to claim 4, further comprising determining the difference between the
5   first period variance and the second period variance, and dividing said difference by the first period variance.

6.   A method according to any one of claims 2 – 5, wherein said decoding step utilises low density parity check decoding.

10

7.   A method according to any one of claims 2 – 5, wherein said decoding step utilises Viterbi decoding.

8.   A method according to any one of claims 2 – 5, wherein said decoding step utilises turbo
15   decoding.

9.   A method according to any one of claims 2 – 8, further comprising the step of converting said received signals from the time domain to the frequency domain before calculating said SNRs.

20   10.   A method according to claim 9, wherein said converting step is performed by an Fast Fourier transformation (FFT) process.

11.   A method according to any one of claims 2 – 10, wherein said sub-channels are encoded by orthogonal frequency division multiplex modulation.

25

12.   A receiver for decoding a data signal from analogue signals received at one or more receiving antennas, said decoding being performed computationally on the basis of bit value probabilities derived from an effective signal to noise ratios (ESNR) and a respective symbol error value (SEV) for all said one or more receiving antennas, said ESNR being calculated utilising signal
30   to noise ratios (SNRs) per sub-channel and measured sub-channel transfer functions for each of said one or more receiving antennas, and said SEVs being calculated utilising said transfer functions.

13.   A receiver for decoding a data signal comprising:

30

one or more receiving antennas receiving one or more transmitted signals, each said transmitted signal having multiple frequency sub-channels containing data symbols;

a circuit calculating a signal to noise ratio (SNR) per sub-channel for each said one or more receiving antennas' respective received signal;

a circuit measuring channel transfer functions for each of said one or more receiving antennas;

a circuit calculating an effective signal to noise ratio (ESNR) for all said one or more receiving antennas utilising a respective said SNR per sub-channel and a respective said channel transfer function;

a circuit calculating symbol error values (SEVs) for all said one or more receiving antennas utilising a respective said channel transfer function and a respective estimated value of said data symbols;

a circuit deriving bit value probabilities utilising said ESNR and a respective said SEV; and

a decoder decoding said data signal utilising said derived bit value probabilities.

14.   A receiver according to claim 13, wherein said circuit calculating said SEVs determines the distance of said estimated symbol values from a predetermined ideal constellation point.

15.   A receiver according to claim 13 or claim 14, wherein the circuit calculating said SNRs per sub-channel samples each said receiving antenna's received signal at each of a first period when there are no data symbols present and a second period when there is at least one data symbol present, determines the variance of said received signals over the duration of the first period and the variance of said received signals over the duration of the second period for each receiving antenna, and calculates said SNR per sub-channel for each receiving antenna utilising said variances.

16.   A receiver according to claim 15, further comprising a circuit determining the difference between the first period variance and the second period variance, and dividing said difference by the first period variance.

17.   A receiver according to any one of claims 13 – 16, wherein said decoder utilises low density parity check decoding.

31

18.　A receiver according to any one of claims 13 – 16, wherein said decoder utilises Viterbi decoding.

19.　A receiver according to any one of claims 13 – 16, wherein said decoder utilises turbo decoding.

20.　A receiver according to any one of claims 13 – 19, further comprising a circuit converting said received signals from the time domain to the frequency domain before said SNRs per sub-channel are calculated.

21.　A receiver according to claim 20, wherein said converter circuit is an Fast Fourier transformation (FFT) circuit.

22.　A receiver according to any one of claims 13 – 21, wherein said sub-channels are encoded by orthogonal frequency division multiplex modulation.
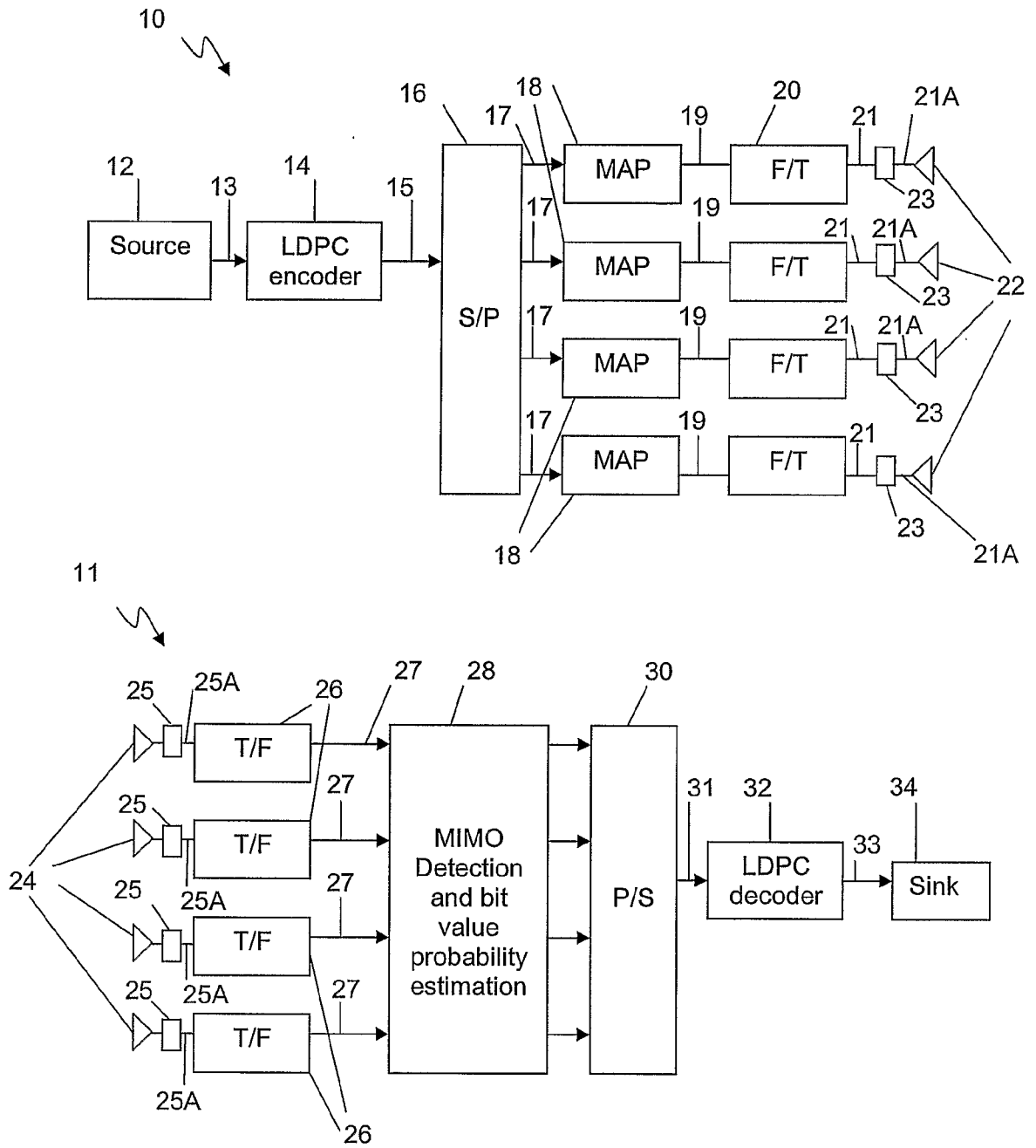
FIGURE 1
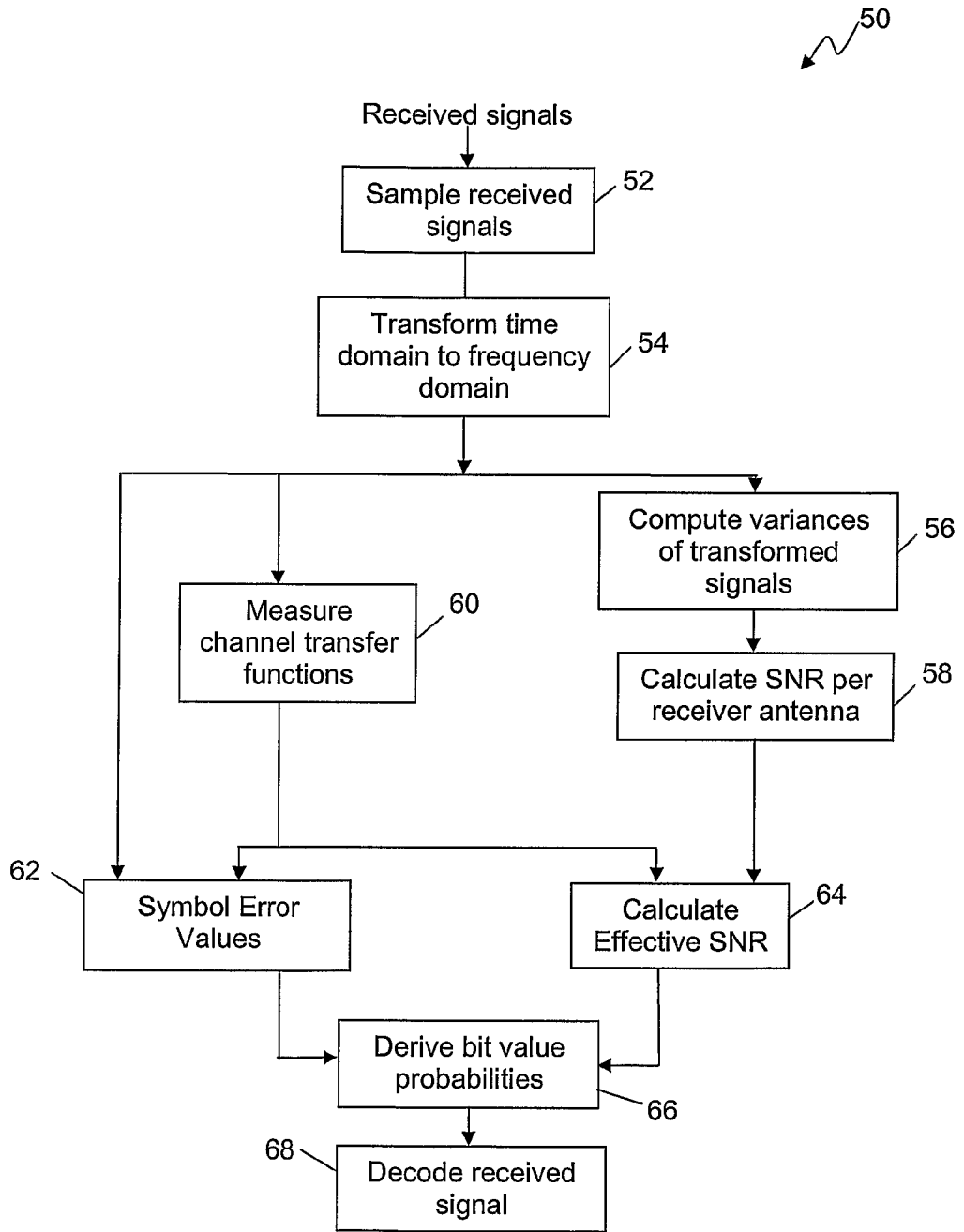
FIGURE 2
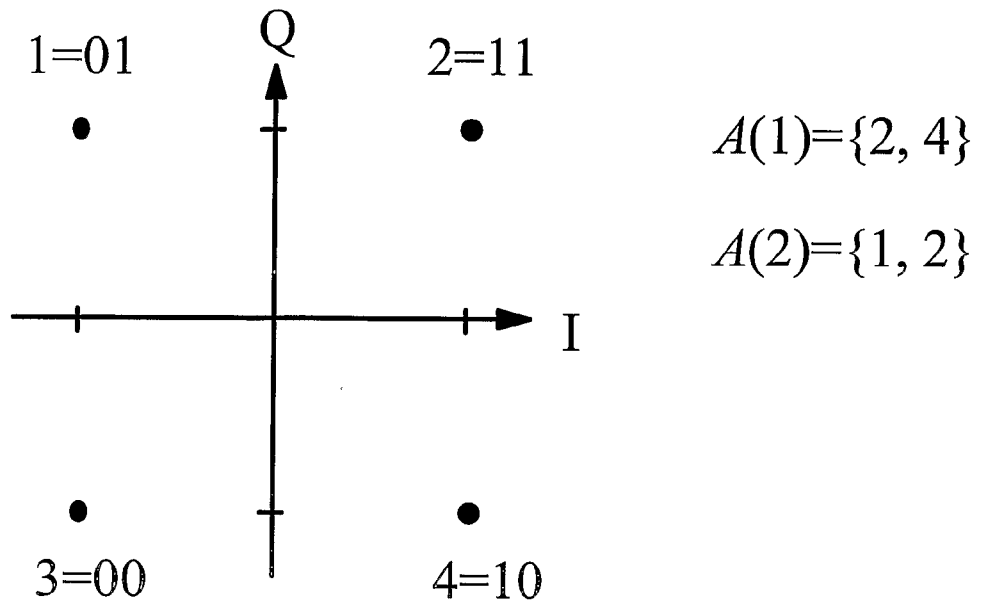
FIGURE 3
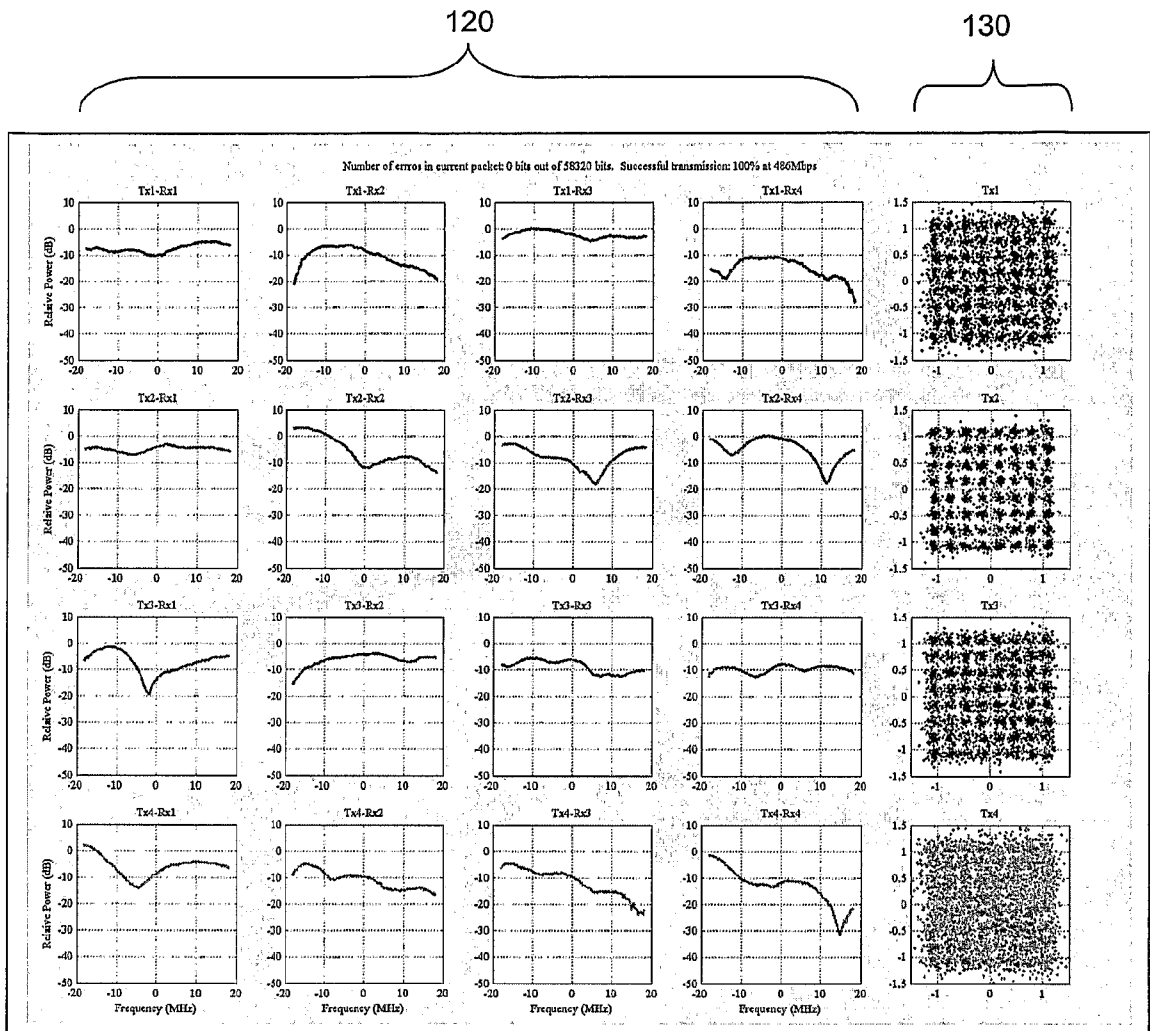
FIGURE 4

INTERNATIONAL SEARCH REPORT

A.	CLASSIFICATION OF SUBJECT MATTER

Int. Cl.

H04B 7/02 (2006.01)　　　*H04L 1/02* (2006.01)

According to International Patent Classification (IPC) or to both national classification and IPC

B.	FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
WPAT, USPTO, IEEE keywords : receiver, decoder, antenna, snr, signal noise ratio, symbol error, sub channel and like keywords

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 7006848 B2 (LING et al) 28 February 2006 Whole document	1-22
Y	US 6990059 B1 (ANIKHINDI et al) 24 January 2006 Col 2 lines 24-67 Col 4 lines 21-35 Col 5 lines 17-23; line 55-col 6 line 20 Fig 3	1-22
Y	US 6785341 B2 (WALTON et al) 31 August 2004 Whole document	1-22

[X] Further documents are listed in the continuation of Box C	[X] See patent family annex

Date of the actual completion of the international search	Date of mailing of the international search report
04 May 2006	- 9 MAY 2006

Name and mailing address of the ISA/AU	Authorized officer
AUSTRALIAN PATENT OFFICE PO BOX 200, WODEN ACT 2606, AUSTRALIA E-mail address: pct@ipaustralia.gov.au Facsimile No. (02) 6285 3929	**ATA MAQBOOL** Telephone No : (02) 6283 7905

INTERNATIONAL SEARCH REPORT

C (Continuation).	DOCUMENTS CONSIDERED TO BE RELEVANT	
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 2003/069816 A2 (WITCOM LTD) 21 August 2003 Whole document	1-22
A	US 2004/0125900 A1 (LIU et al) 1 July 2004 Whole document	1-22

INTERNATIONAL SEARCH REPORT

Information on patent family members

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars · which are merely given for the purpose of information.

Patent Document Cited in Search Report		Patent Family Member						
US	6990059	US	2006056283					
US	7006848	BR	I0208312	CN	1552132	EP	1371147	
		US	6771706	US	2002191703	US	2003003880	
		US	2004165558	US	2005002326	WO	02078211	
US·	6785341	BR	0209521	CA	2446512	CN	1531787	
		EP	1386424	MX	PA03010231	RU	2003135853	
		US	2003035491	US	2005002468	WO	02093784	
WO	03069816	AU	2003206115	EP	1483856			
US	2004125900							

Due to data integration issues this family listing may not include 10 digit Australian applications filed since May 2001.

END OF ANNEX