



- (51) International Patent Classification: **H03M 13/29** (2006.01)
- (21) International Application Number: **PCT/EP2012/054497**
- (22) International Filing Date: **14 March 2012 (14.03.2012)**
- (25) Filing Language: **English**
- (26) Publication Language: **English**
- (30) Priority Data: **11352003.5** 14 March 2011 (14.03.2011) **EP**
- (71) Applicant (for all designated States except US): **COMMISSARIAT À L'ÉNERGIE ATOMIQUE ET AUX ÉNERGIES ALTERNATIVES** [FR/FR]; 25, rue Leblanc, Bâtiment "Le Ponant D", F-75015 Paris (FR).
- (72) Inventor; and
- (75) Inventor/Applicant (for US only): **REDDY, Pallavi** [FR/FR]; 2 rue Pierre Ruybet, F-38000 Grenoble (FR).
- (74) Agents: **AUGARDE, Eric** et al.; Brevalex, 56 boulevard de l'Embouchure, B.P. 27519, F-31075 Toulouse Cedex 2 (FR).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU,

[Continued on next page]

(54) Title: STATE METRICS BASED STOPPING CRITERION FOR TURBO-DECODING

(57) Abstract: The present invention proposes a new stopping criterion for a turbo-encoding method. This criterion is based on a state metrics calculated by the forward backward recursion in the coding trellis of an elementary encoder. If, for at least one elementary decoding stage (610<sub>p</sub>), the forward state metrics of the last symbol of the block or the backward state metrics of the first symbol of the block exceeds a first threshold (612), the turbo-decoding iterations are stopped. If it is not the case, it is further checked (613) whether the state metrics exceeds a second threshold and if the absolute value of the difference between the current state metrics and the state metrics obtained at the previous iteration lies below a given margin. In the affirmative, the turbo-decoding iterations are stopped and a hard decision is taken on the extrinsic values (614).

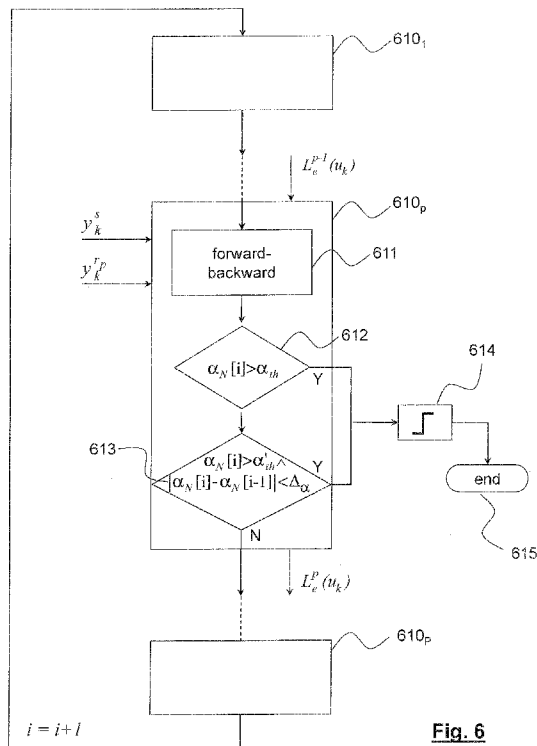


Fig. 6

WO 2012/123514 A1

LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, **Published:**  
SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, — *with international search report (Art. 21(3))*  
GW, ML, MR, NE, SN, TD, TG).

STATE METRICS BASED STOPPING CRITERION FOR TURBO-  
DECODING

TECHNICAL FIELD

5           The present application relates to the field of turbo codes and more particular to turbo-decoding.

PRIOR ART

10           The turbo-codes were introduced by C. Berrou et al. in the seminal article entitled "Near Shannon limit error-correcting coding and decoding: Turbo Codes", Proceedings of ICC' 93 pp. 1064-1070, have encountered since then widespread success. The remarkable performances of the turbo-codes in terms of bit error rate or BER (Bit Error Rate), close to those of random coding, are explained by the presence of one or more interleaver(s) between different elementary coders. They are used as error correcting codes in many digital communication standards.

20           In its most common form, a turbo-encoder can be regarded as a parallel or a serial concatenation of systematic elementary encoders separated by interleavers.

25           At the receiving side, turbo-decoding is achieved by an iterative process involving as many elementary decoders as the turbo-encoder has elementary encoders, each iteration improving the reliability of the decoded codeword.

30           Fig. 1 illustrates an example of the structure of a turbocoder comprising two parallel concatenated

elementary recursive systematic convolutional (RSC) encoders.

In this example, a block of information bits,  $I$ , are coded by two elementary encoders of the systematic convolutional type, arranged in parallel. Each elementary encoder is constituted by a (shared) systematic path, leaving by definition the information bits unchanged, and by a convolutional encoder. The first convolutional encoder 110 outputs parity bits  $P_1$  as a function of the information bits  $I$ . The block of parity bits is hereafter referred to as the first redundancy part. Similarly, the second convolutional encoder 120 outputs second parity bits  $P_2$  as a function of the information bits  $I$  previously interleaved by the interleaver 130. The block of parity bits  $P_2$  is referred to as the second redundancy part.

The multiplexer 150 multiplexes the information bits  $I$  (systematic part) with the parity bits  $P_1$  (first redundancy part) and  $P_2$  (second redundancy part) before transmission over the communication channel 160. More specifically, the multiplexed bits are converted into  $M$ -ary symbols (where  $M$  is the cardinal of the modulation alphabet) prior to carrier modulation.

Fig. 2 illustrates an example of turbo-decoder for decoding a block encoded by the turbo-encoder of Fig. 1.

The turbo-decoder comprises a first elementary decoder 210 corresponding to encoder 110, a second elementary decoder 220 corresponding to encoder 120.

The decoders are of the SISO type (*Soft In Soft Out*) and may carry out the MAP (*Maximum A posteriori Probability*) decoding algorithm or one of its derivatives e.g. the Log-MAP or Max-Log-MAP decoding  
5 algorithm as known in the art.

The turbo-decoder receives from the symbol demodulator a noisy version of the information bits, the first parity bits and the second parity bits,  
10 respectively denoted  $\tilde{I}$ ,  $\tilde{P}_1$  and  $\tilde{P}_2$ . The systematic bits  $\tilde{I}$  and first parity bits  $\tilde{P}_1$  are supplied to the first elementary decoder 110. The first elementary decoder also inputs *a priori* information (also called extrinsic information) from the second elementary decoder.

15 Similarly, the second elementary decoder 220 inputs the received information bits  $\tilde{I}$ , after having being interleaved in interleaver 230, along with the received parity bits  $\tilde{P}_2$ . The second elementary decoder additionally inputs extrinsic information from the  
20 first elementary decoder after it has been interleaved in interleaver 230'. The interleavers 230 and 230' exhibit the same interleaving pattern  $\Pi$  as interleaver 130. The extrinsic information provided by the second decoder is de-interleaved in de-interleaver 240 before  
25 being supplied to the first elementary decoder. The de-interleaver has the inverse interleaving pattern  $\Pi^{-1}$  of interleaver 130.

The encoded block is subjected to iterative decoding, each decoding iteration comprising a first  
30 half-iteration in decoder 210 and a subsequent second

half-iteration in decoder 220. The update and exchange of extrinsic information, *i.e.* of the extrinsic values relative to the different information bits, between the first and second elementary decoder, at each iteration, increases the reliability of the decoded block. It should be noted that the extrinsic information of the first decoder is initialized at values indicative of a lack of prior knowledge of bit reliability (0 if LLR values are used).

10

The extrinsic information is classically expressed in terms of soft values, *e.g.* LLRs (*Log Likelihood Ratios*) indicating the reliability of the decoding of each binary symbol. After completion of a given number of iterations or, more generally, when a predetermined stopping criterion is met, the decoding process is stopped and a hard decision is taken in 250 on the soft values to provide hard bit values. The hard decision is taken by performing a simple comparison with a given threshold (0 in case of LLR values).

The power consumption and the latency linearly depend upon the number of iterations of the decoder. It is therefore important to design an efficient stopping criterion, which does not lead to unnecessary iterations while simultaneously achieving a reasonable level of reliability of the decoded values.

Several stopping criteria have been proposed in the literature, based on soft and/or hard values.

US2007/0300139 discloses a stopping criterion based on the comparison of the LLR values output by an elementary decoder for two consecutive iterations. The

30

implementation of such a stopping criterion requires however large storage capacity since a soft value (e.g. coded with a byte) has to be stored for each binary symbol of the block.

5           EP-A-1455457 further proposes to compare the minimum absolute value of the LLR values of an block with a given threshold. Once this minimum absolute value exceeds the threshold, the iterative decoding process is stopped. In other words, the decoding  
10 process stops when the reliability level of the less reliable bit is high enough.

In practice, this stopping criterion is quite sensitive to the threshold value. A too high threshold may result in a frequent rejection of correctly decoded  
15 blocks.

An alternative stopping criterion is based on the cross entropy of the LLR related to two consecutive iterations. This criterion is efficient but requires extra computation and considerable memory capacity.

20           Further stopping criteria are based on monitoring the sign changes in the extrinsic information related to two consecutive iterations. Indeed if a sign of a soft value oscillates from on iteration to the text, this indicates that the corresponding bit is not yet  
25 reliable. This stopping criterion requires considerable memory capacity, as the signs of the LLR values for at least two blocks of extrinsic information need to be stored.

The aim of the present invention is therefore to  
30 propose a new stopping criterion for a turbodecoding

process, which is robust without needing computational overhead or large memory capacity.

#### DISCLOSURE OF THE INVENTION

5           The present invention is defined in the appended independent claims. Various advantageous embodiments are given in the dependent claims.

#### BRIEF DESCRIPTION OF THE DRAWINGS

10           The present invention will be better understood from the description of the following embodiments, by way of illustration and in no way limitative thereto:

          Fig. 1 schematically illustrates an example of turbo-encoder known from the prior art;

15           Fig. 2 schematically illustrates an example of turbo-decoder known from the prior art;

          Fig. 3 schematically illustrates a coding path in the trellis diagram of an elementary encoder of the turbo-encoder;

20           Fig. 4 schematically represents forward state metrics and backward state metrics used in the forward-backward decoding algorithm;

          Fig. 5A illustrates the evolution of the forward state metrics of the last block symbol for consecutive  
25 turbo-decoding iterations;

          Fig. 5B illustrates the evolution of the state metrics for the various symbols of the block and consecutive turbo-decoding iterations;

Fig. 6 schematically represents a flowchart of a turbo-decoding method using a stopping criterion according to an embodiment of the invention.

## 5 DETAILED DISCLOSURE OF PARTICULAR EMBODIMENTS

We will consider again in the following a turbo-decoder for decoding blocks of binary symbols which have been encoded by a turbo-encoder. The term block can be construed here as a frame or a codeword. The  
10 binary symbols processed by the turbo-decoder are expressed in terms of soft values (soft information), typically in terms of LLRs as known in the art.

The turbo-encoder may be constituted by a plurality  $P$  of parallel or serially concatenated  
15 elementary encoders or a combination of such serial and parallel concatenated elementary encoders (hybrid turbo-encoder). Without loss of generality we will assume that the elementary encoders are RSC (*Recursive Systematic Convolutional*) encoders. It will  
20 nevertheless be understood by the man skilled in the art that other types of encoders can be used while remaining in the scope of present invention.

The turbo-decoder is constituted by the same plurality  $P$  of elementary decoders and performs an  
25 iterative decoding process as set out above. It is recalled that each elementary decoder corresponds to an elementary encoder of the turbo-encoder and processes the same kind of information as the one output by the corresponding elementary encoder. More specifically,  
30 each elementary decoder inputs the systematic part of

the received block (e.g. after propagation over the communication channel), possibly interleaved, and a redundancy part corresponding to the one generated by said elementary encoder. In addition, the elementary decoder receives extrinsic information from at least another elementary decoder. Each elementary decoder generates its own extrinsic information from the input systematic part, redundancy part and extrinsic information. The extrinsic information thus generated is supplied in turn to at least another elementary decoder.

Before describing the operation of an elementary decoder, we first consider the corresponding elementary coder in the turbo-encoder. This elementary encoder, as any convolutional encoder, can be regarded as a finite state machine, the transitions of which are determined by the generator polynomial. If the memory of the encoder is of size  $v$  ( $v$  shift register stages), the finite state machine may take  $2^v$  states, each state being defined by the contents of the memory. It should be noted that the various elementary encoders of the turbo-encoder may have different memory sizes (i.e. shift registers having different lengths) and hence different numbers of states.

The coding operation may be regarded as a coding path in a trellis diagram as represented in Fig. 3.

Conventionally, the states of the encoder are represented vertically by circles whereas the horizontal direction is the time axis. Each column of states corresponds to a clock cycle. At each input bit, on progresses rightwards (or forward) in the trellis,

and the encoder goes from a state at a time  $k$  to a subsequent state at time  $k+1$ . The transition between the state at time  $k$  and the state at time  $k+1$  is therefore represented by an edge in the trellis (and by an arc in the state diagram). The coding of a block of bits results therefore in a path within the trellis. At each time  $k$ , the encoder outputs a parity bit (in addition to the systematic bit), which depends upon the state of the encoder and the input bit.

10

The corresponding elementary decoder implements a forward-backward algorithm as known in the art. By forward-backward algorithm we mean here the MAP decoding algorithm or one of its derivatives such as the Log-MAP or the Max-Log-MAP algorithm. A description of the MAP algorithm can be found in the original paper of Bahl *et al.* entitled "Optimal decoding of linear codes for minimizing symbol error rate" published in IEEE Trans. on Information Theory, pp. 284-287, March 1974. A description of the Max-Log-MAP algorithm can be found e.g. in the article of C. Hye-Mi *et al.* entitled "Low power hybrid turbo decoding based in reverse calculation, ISCAS 2006."

The forward-backward decoding algorithm is based on the same trellis as the coding trellis described above. More specifically, the forward-backward decoding algorithm mainly comprises three steps: a forward recursion, a backward recursion and an LLR calculation.

In the forward recursion, the algorithm recursively calculates the forward state probability of being in each state of the trellis at each time  $k$  given

30

the knowledge of all the previous received symbols. The forward state probability, also called forward state metrics, is recursively obtained by:

$$5 \quad \alpha_k(s) = \sum_{s'} \alpha_{k-1}(s') \gamma_k(s', s) \quad k=1, \dots, N-1 \quad (1)$$

where  $s$  is the current state and  $s'$  is the preceding state,  $\alpha_{k-1}(s')$  is the forward state probability of being at state  $s'$  at time  $k-1$ , and  $\gamma_k(s', s)$  is the branch  
 10 probability of the transition between state  $s'$  and state  $s$ , given the received symbol,  $N$  is the size of the block. The state  $s'$  at the beginning of the transition  $s' \rightarrow s$  will be referred to as the transition starting state whereas the state will be referred to as  
 15 the transition ending state. The branch probability is also called the branch metrics.

The forward recursion is initialized by forcing the starting state to state 0 and setting:

$$20 \quad \alpha_0(0) = 1 \text{ and } \alpha_0(s) = 0 \text{ for } s \neq 0 \quad (2)$$

Fig. 4 schematically illustrates how the forward state metrics  $\alpha_k(s)$  is calculated for a state  $s$  of the trellis. At time  $k$ , the forward state metrics of state  
 25  $s$  is calculated (by expression (1)) from the forward state metrics of the possible preceding states  $s'$  (in grey) at time  $k-1$  (i.e. those corresponding to an authorized transition  $s' \rightarrow s$ ),  $\alpha_{k-1}(s)$ , and the transition metrics between states  $s$  and  $s'$ ,  $\gamma_k(s', s)$ .

Similarly, in the backward recursion, the algorithm recursively calculates the backward state probability of being in each state of the trellis at each time  $k$  given the knowledge of all the following symbols at time  $k+1$ . The backward state probability, also called backward state metrics, is recursively obtained by:

$$\beta_{k-1}(s') = \sum_s \beta_k(s) \gamma_k(s', s) \quad k = N, \dots, 2 \quad (3)$$

10

As illustrated in Fig. 4, the backward state metrics at time  $k$  is calculated from the backward state metrics of the possible following states  $s$  (nodes shaded in grey) at time  $k+1$ ,  $\beta_{k+1}(s)$ , and the transition metrics between states  $s$  and  $s'$ ,  $\gamma_{k+1}(s', s)$ .

15

The backward recursion is initialized by forcing the ending state to state 0 and setting:

$$\beta_N(0) = 1 \text{ and } \beta_N(s) = 0 \text{ for } s \neq 0 \quad (4)$$

20

The output LLR for each bit  $u_k$  at time  $k$  is given by:

$$L(u_k) = \ln \frac{P(u_k = +1 | \mathbf{y}_k)}{P(u_k = -1 | \mathbf{y}_k)} = \ln \frac{\sum_{\substack{(s',s) \\ u_k = +1}} \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s)}{\sum_{\substack{(s',s) \\ u_k = -1}} \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s)} \quad (5)$$

25

where  $\mathbf{y}_k$  is the modulation symbol received on the communication channel at time  $k$  and where the bit

values have been noted  $\pm 1$ . The upper summation in (5) is carried out over all branches with input label +1 and the lower summation is carried out over all branches with input label -1.

5 For an AWGN (Additive White Gaussian Noise) channel, the branch metrics  $\gamma_k(s, s')$  can be obtained as:

$$\gamma_k(s', s) = \exp\left(-\frac{\|\mathbf{y}_k - \mathbf{x}_k\|^2}{2\sigma^2}\right) \quad (6)$$

10 where  $\sigma^2$  is the noise variance and  $\mathbf{x}_k$  is the expected symbol for the transition  $s'$  to  $s$ . Therefore, the  $\gamma_k(s', s)$  values can be calculated at each turbo-decoding iteration.

15 The expressions (1)-(6) refer to the MAP algorithm. They can be simplified by using logarithmic values (Log-MAP algorithm) and even further simplified by retaining only the most significant values under the summation signs (Max-Log-MAP algorithm). In the latter  
20 case, denoting  $\check{\rho}$  for  $\ln \rho$ :

$$\check{\alpha}_k(s) = \max_{s'} [\check{\alpha}_{k-1}(s') + \check{\gamma}_k(s', s)] \quad (1')$$

$$\check{\alpha}_0(0) = 0 \text{ and } \check{\alpha}_0(s) = -\infty \text{ for } s \neq 0 \quad (2')$$

25

$$\check{\beta}_{k-1}(s') = \max_{s'} [\check{\beta}_k(s) + \check{\gamma}_k(s', s)] \quad (3')$$

$$\tilde{\beta}_N(0) = 0 \text{ and } \tilde{\beta}_N(s) = -\infty \text{ for } s \neq 0 \tag{4'}$$

$$L(u_k) = \max_{\substack{(s',s) \\ u_k=+1}} [\tilde{\alpha}_{k-1}(s') + \tilde{\gamma}_k(s',s) + \tilde{\beta}_k(s)] - \max_{\substack{(s',s) \\ u_k=-1}} [\tilde{\alpha}_{k-1}(s') + \tilde{\gamma}_k(s',s) + \tilde{\beta}_k(s)] \tag{5'}$$

5

$$\tilde{\gamma}_k(s',s) = -\frac{1}{2\sigma^2} [\|\mathbf{y}_k\|^2 + \|\mathbf{x}_k\|^2] + \frac{1}{\sigma^2} (\mathbf{y}_k \cdot \mathbf{x}_k) \tag{6'}$$

As the first term between brackets is common to all branch metrics, it can be dropped for the purpose of calculating (1'), (3') and (5'), in other words:

$$\tilde{\gamma}_k(s',s) = \frac{1}{\sigma^2} (\mathbf{y}_k \cdot \mathbf{x}_k) \tag{6''}$$

In expressions (6) or (6''), the only changing parameter is  $\mathbf{x}_k$ .  $\mathbf{x}_k$  is a  $P+1$  dimension vector (assuming one parity bit per elementary coder), the elements of which are the systematic bit, denoted  $x_k^s$  and the parity bits denoted  $x_k^p$ ,  $p=1, \dots, P$ . The soft value of the systematic bit in (6) or (6'') is replaced by:

20

$$x_k^s = y_k^s + L_e(u_k) \tag{7}$$

where  $L_e(u_k)$  is the extrinsic information provided from another elementary decoder. Therefore, the  $\gamma_k(s',s)$  values can be calculated at each turbo-decoding iteration.

25

The forward state metrics of the last symbol of the block can be regarded as a cumulated metrics along the expected coding path in trellis, when travelled through in the forward direction. Similarly, the backward state metrics of the first symbol can be regarded as a cumulated metrics along the expected coding path in the trellis, when travelled through in the backward direction.

In general, the state metrics (either forward or backward) increases along the travelled path. For example, if the Max-Log-MAP algorithm is used, the forward state metrics increases between the first symbol and the last symbol from 0 to the cumulated value:

$$\tilde{\alpha}_N(s) = \max_{s'} \left( \sum_{k=1}^N \tilde{\gamma}_k(s', s) \right) \quad (8)$$

and, similarly, the backward state metrics increases between the last symbol and the first symbol from 0 to the same cumulated value:

$$\tilde{\beta}_0(s) = \max_{s'} \left( \sum_{k=1}^N \tilde{\gamma}_k(s', s) \right) \quad (9)$$

This cumulated value reflects the likelihood of the considered coding path. In the remaining part of the description, it should be understood that the coding path referred to is the one exhibiting the highest likelihood.

Fig. 5A represents the cumulated value  $\alpha_N[i]$  of the forward state metrics of the considered coding path, for a plurality of consecutive iterations  $i$ . The curves 510 correspond to different blocks of symbols and a SNR value of 1.1 dB.

It should be noticed that the forward state metrics saturates after 4 or 5 iterations. In other words, the cumulated value of the forward state metrics is a good indicator of the convergence of the turbo-decoding. Similarly, the backward state metrics saturates after a few iterations and is therefore an equivalent indicator of convergence.

Fig. 5B illustrates the variation of the forward state metrics  $\alpha_k[i]$  along the considered coding path,  $k=0,..,N$  for consecutive turbo-decoding iterations.

For each iteration, the forward state metrics ramps up from the first symbol to the last ( $N^{th}$ ) symbol of the block. As already noticed in Fig. 6, the cumulated value,  $\alpha_N$ , reached at the  $N^{th}$  symbol of the block, increases as the turbo-decoding proceeds and then saturates after a few iterations.

A first stopping criterion according to the present invention consists in comparing the cumulated value of the forward state metrics (resp. the backward state metrics) with a predetermined threshold  $\alpha_{th}$  (resp.  $\beta_{th}$ ). This threshold, hereinafter referred to as first threshold, can be tuned by simulation to the block length and the expected SNR (Signal to Noise Ratio). If the cumulated value at iteration  $i$  is greater than said threshold, i.e. if:

$$\alpha_N[i] > \alpha_{th} \quad (\text{resp. } \beta_N[i] > \beta_{th}) \quad (10)$$

the turbo-decoding iterations are stopped.

5

A second stopping criterion can be combined with the first in order to enhance its robustness. More specifically, if the cumulated value of the forward state metrics (resp. backward state metrics) does not meet the first stopping criterion, it is checked whether a second stopping criterion:

$$\alpha_N[i] > \alpha'_{th} \quad \text{and} \quad |\alpha_N[i] - \alpha_N[i-1]| < \Delta_\alpha \quad (11)$$

(resp.  $\beta_N[i] > \beta'_{th}$  and  $|\beta_N[i] - \beta_N[i-1]| < \Delta_\beta$ ) is met, where  $\alpha'_{th}$  (resp.  $\beta'_{th}$ ) is a second threshold and  $\Delta_\alpha$  (resp.  $\Delta_\beta$ ) is a margin. In the affirmative, the turbo-decoding iterations are stopped. The second criterion (11) expresses the saturation of the state metrics at a sufficient high level. Typically, the second threshold is lower than the first threshold.

Fig. 6 represents a flowchart of a turbo-decoding method using a stopping criterion according to an embodiment of the invention.

25 As already mentioned in the outset, the turbo-decoding method consists in a plurality of iterations, each iteration being subdivided into  $P$  decoding stages,  $610_1, \dots, 610_P$  sequentially performed by the  $P$  elementary decoders.

The stopping criterion is implemented here in the  $p^{\text{th}}$  decoder. More generally, it is implemented in at least one elementary decoder.

The  $p^{\text{th}}$  decoder receives the systematic soft bits  $y_k^s$  and the soft parity bits  $y_k^{r_p}$  from the demodulator. It also inputs the extrinsic values  $L_e^{p-1}(u_k)$  from the preceding decoder.

We consider here the  $i^{\text{th}}$  turbo-decoding iteration. The decoding stage 610<sub>p</sub> starts with a forward-backward recursion in 611. The forward-backward algorithm can be the MAP, Log-MAP or Max-Log-MAP decoding algorithm.

The state metrics of the last symbol of the block (forward state metrics) or the first symbol of the block (backward state metrics) is calculated at the end of the forward-backward recursion 611 and the first stopping criterion is checked in 612.

If the state metrics exceeds the first threshold (expression (10)), the turbo-encoding exits and a hard decision is taken in 614 on the extrinsic values just obtained from the forward-backward recursion.

In the negative, the second stopping criterion is checked in 613. If the state metrics exceeds the second threshold and if the absolute value of the difference of state metrics between the current iteration and the previous one lies below a predetermined margin, the turbo-encoding exits and a hard decision on the extrinsic values is taken in 614. The turbo-decoding then ends at 615.

Conversely, if neither the first nor the second stopping criterion is met, the iteration proceeds with the next decoding stage  $610_{p+1}$ .

According to a variant (not shown), the first  
5 stopping criterion can be implemented alone. However for increased robustness to SNR, the combination of the first and second criteria is preferred.

It will also be understood by the man skilled in the art that the implementation of the first stopping  
10 criterion does not require any storage in a memory (no state metrics is stored) and can be checked on the fly with very little additional overhead.

## CLAIMS

1. Turbo-decoding method for turbo-decoding a block of ( $N$ ) symbols having been coded by a turbo-encoder, said turbo-decoding comprising a plurality of iterations, each iteration comprising a plurality ( $P$ ) of decoding stages, each decoding stage being carried out by an elementary decoder corresponding to an elementary encoder of the turbo-encoder, each decoding stage performing a forward-backward recursion (611) on a coding trellis of the corresponding encoder, said trellis representing various ( $2^v$ ) states of the encoder for each symbol of said block, characterized in that:

- a forward state metrics ( $\alpha_N$ ) of the last symbol or a backward state metrics ( $\beta_0$ ) of the first symbol of said block is calculated (611);

- said state metrics is compared (612) with a first predetermined threshold ( $\alpha_{th}, \beta_{th}$ ); and

- if said state metrics exceeds said first predetermined threshold, the iterations are stopped.

2. Turbo-decoding method according to claim 1, characterized in that if said state metrics does not exceed said first predetermined threshold,

- it is checked:

(a) whether said state metrics exceeds a second predetermined threshold; and

(b) if the absolute value of the difference between said state metrics and the state metrics of the previous iteration lies below than a predetermined margin; and

- if both conditions (a) and (b) are met, the iterations are stopped.

3. Turbo-decoding method according to claim 2,  
5 characterized in that said second threshold is lower than said first threshold.

4. Turbo-decoding method according to any of  
claims 1 to 3, characterized in that that each decoding  
10 stage supplies extrinsic values of systematic bits to the next decoding stage and the last decoding stage supplies extrinsic values of the systematic bits to the first decoding stage.

15 5. Turbo-decoding method according to claim 4, characterized in that if said state metrics exceeds said first threshold, a hard decision is taken on the extrinsic values supplied by said at least one decoding stage to provide hard decided systematic bits.

20

6. Turbo-decoding method according to claims 2 and 4, characterized in that if said conditions (a) and (b) are met, a hard decision is taken on the extrinsic values supplied by said at least one decoding stage to  
25 provide hard decided systematic bits.

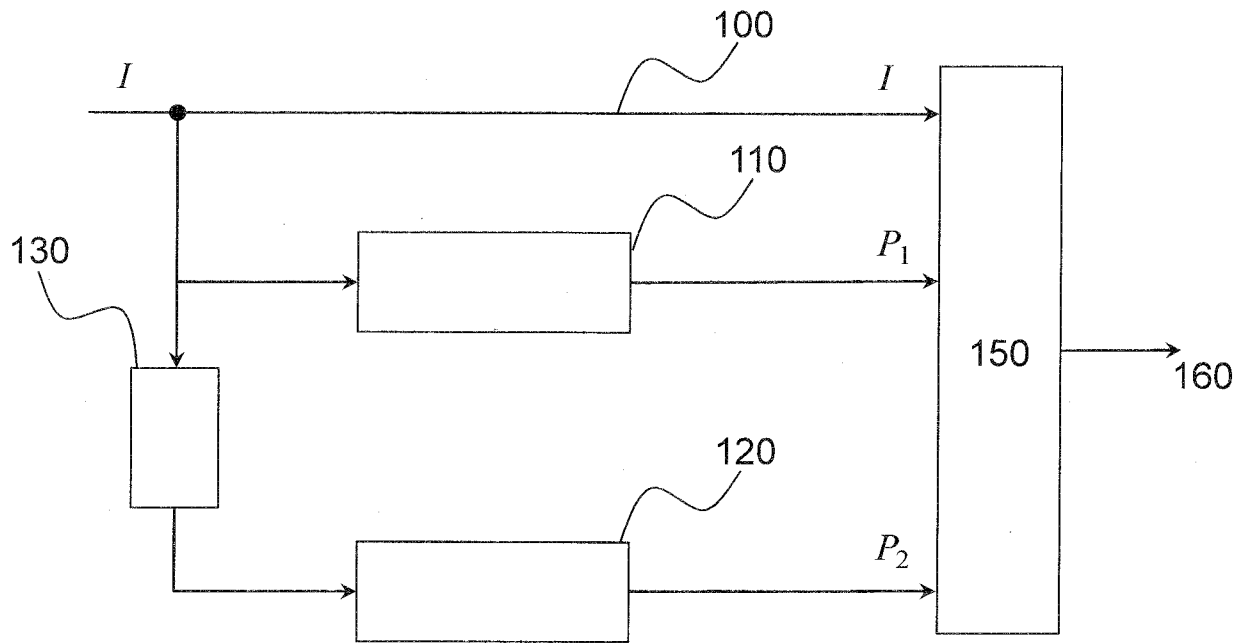
7. Turbo-decoding method according to any preceding claim, characterized in that the forward-backward recursion is performed according to the MAP  
30 decoding algorithm.

8. Turbo-decoding method according to any of claims 1 to 6, characterized in that the forward-backward recursion is performed according to the Log-MAP decoding algorithm.

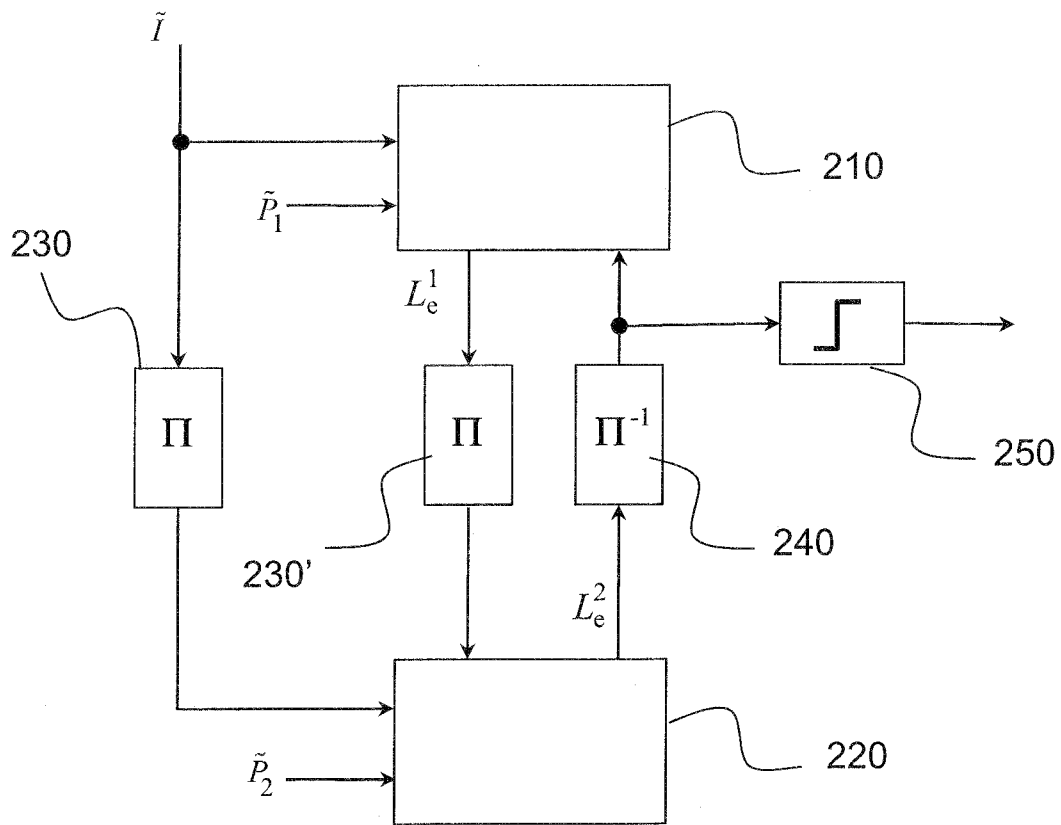
5

9. Turbo-decoding method according to any of claims 1 to 6, characterized in that the forward-backward recursion is performed according to the Max-Log-MAP decoding algorithm.

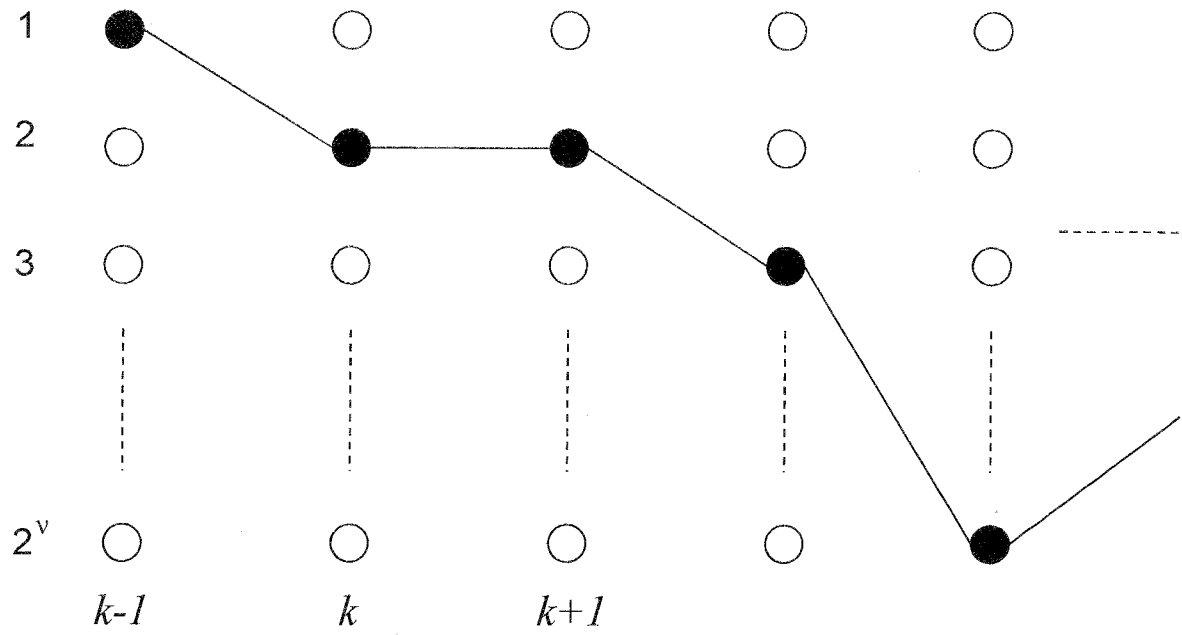
10



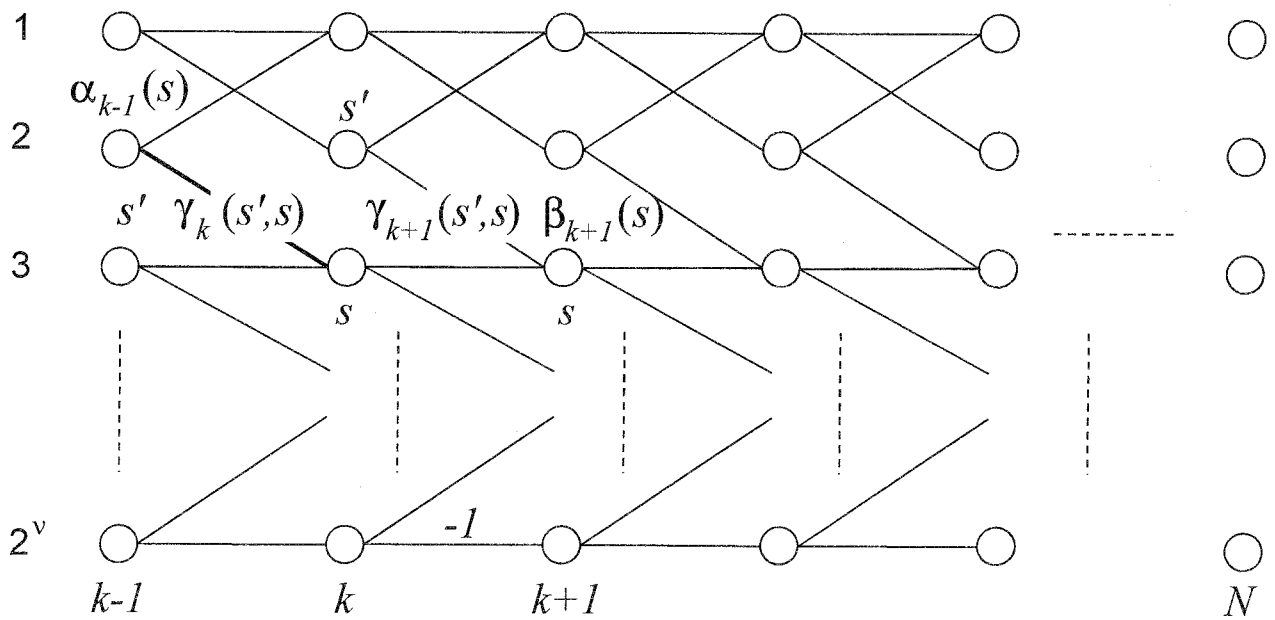
**Fig. 1**



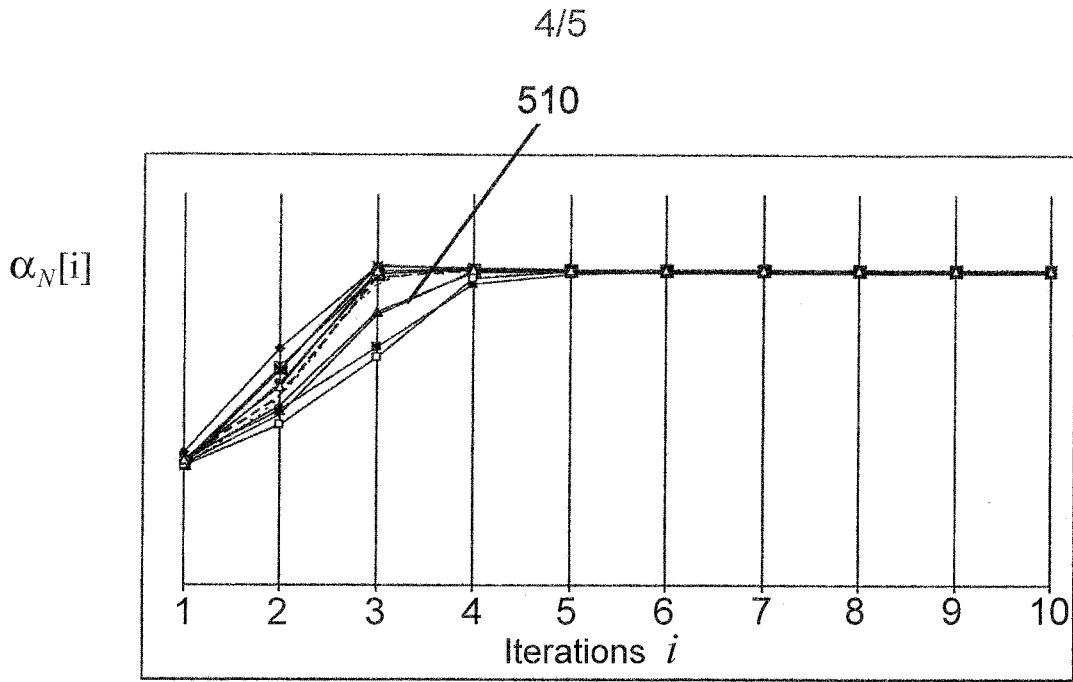
**Fig. 2**



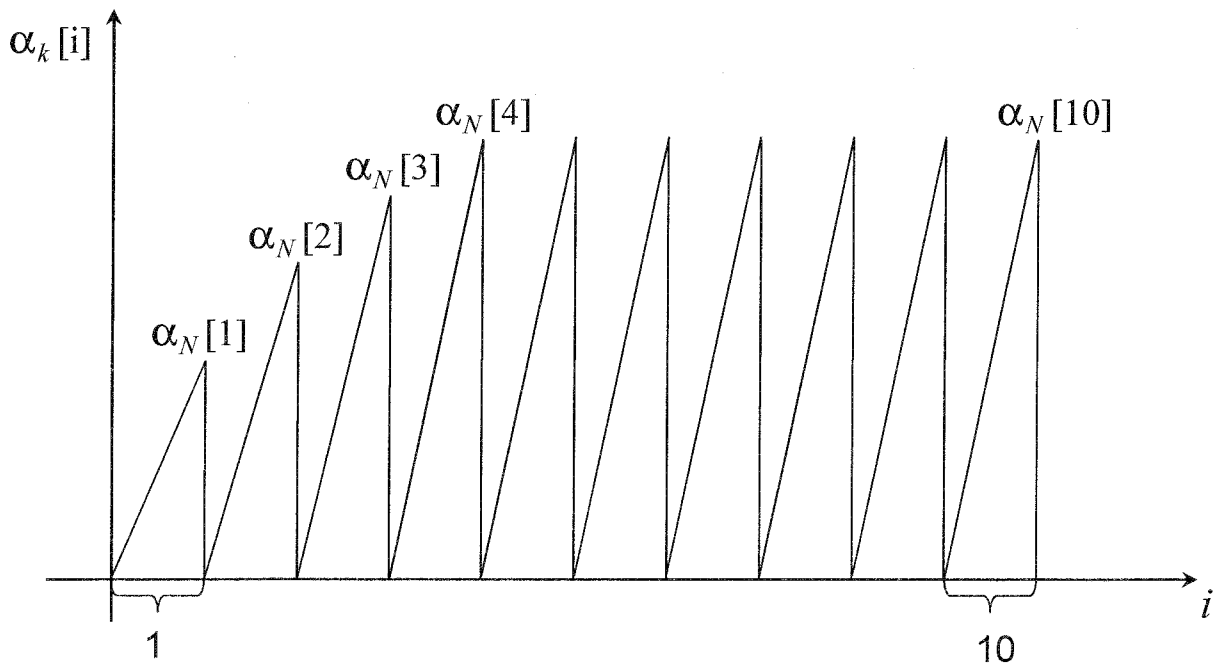
**Fig. 3**



**Fig. 4**

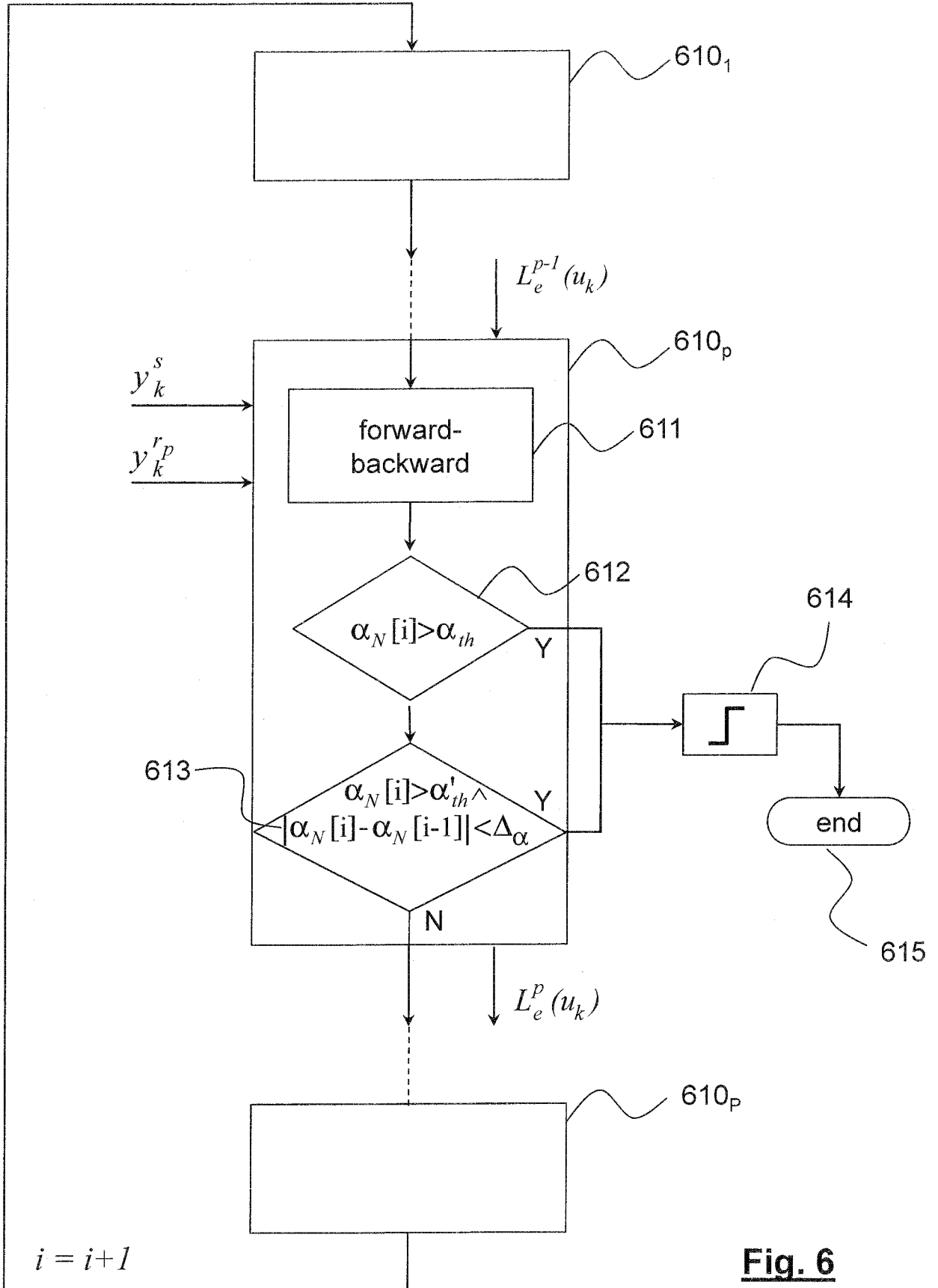


**Fig. 5A**



**Fig. 5B**

5/5



**Fig. 6**

INTERNATIONAL SEARCH REPORT

International application No  
PCT/EP2012/054497

A. CLASSIFICATION OF SUBJECT MATTER  
INV. H03M13/29  
ADD.  
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)  
H03M  
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 practical, search terms used)  
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	C.B. Schlegel, L.C. Perez: "Trellis and Turbo Coding", 2002, IEEE Press, XP002649770, Chapter 7.7, chapter 10.7 -----	1-9

Further documents are listed in the continuation of Box C.

See patent family annex.

\* Special categories of cited documents :

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- "&" document member of the same patent family

Date of the actual completion of the international search  
18 April 2012

Date of mailing of the international search report  
25/04/2012

Name and mailing address of the ISA/  
European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040,  
Fax: (+31-70) 340-3016

Authorized officer  
Rydyger, Kay