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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 February 2011 (24.02.2011)

PCT

(10) International Publication Number
WO 2011/020770 Al

(51) International Patent Classification
H03M 13/41 (2006 01)

(21) International Application Number
PCT/EP20 10/06 1760

(22) International Filing Date
12 August 2010 (12 08 2010)

(25) Filing Language English

(26) Publication Language English

(30) Priority Data
61/234,701 18 August 2009 (18 08 2009) US
10159966 0 15 April 2010 (15 04 2010) EP

(71) Applicant *(for all designated States except US)* **TELEFONAKTIEBOLAGET L M ERICSSON (PUBL)** [SE/SE], S-164 83 Stockholm (SE)

(72) Inventors; and
(75) Inventors/Applicants *(for US only)* **KAMUF, Matthias** [DE/SE], Byggmastaregatan 8A, S-222 37 Lund (SE) **ANG, Lay Hong** [SG/SG], Block 461 Jurong West, St 41, #07-652, Singapore 640461 (SG) LIM, **Wee Guan** [SG/SG], 36 Man Novena Barat, Singapore 308618 (SG)

(74) Agent AKERMAN, **Marten,** Ericsson AB, Nya Vattentornet, S-221 83 Lund (SE)

(81) Designated States *(unless otherwise indicated, for every kind cf 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, RO, RS, RU, 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 cf regional protection available)* ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, 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, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)

[Continued on nextpage]

(54) Title SOFT OUTPUT VITERBI ALGORITHM METHOD AND DECODER

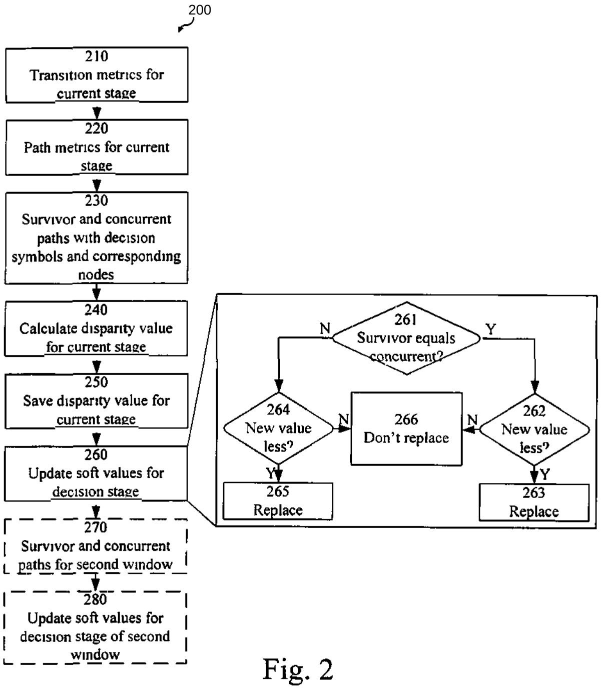
200
210
Transition metrics for current stage
220
Path metrics for current stage
230
Survivor and concurrent paths with decision symbols and corresponding nodes
240
Calculate disparity value for current stage
250
Save disparity value for current stage
260
Update soft values for decision stage
270
Survivor and concurrent paths for second window
280
Update soft values for decision stage of second window
261
Survivor equals concurrent?
N
Y
264
New value less?
N
Y
266
Don't replace
262
New value less?
N
Y
265
Replace
263
Replace

Fig. 2

(57) Abstract A method of decoding a block with a Soft Output Viterbi Algorithm (SOVA) using a trellis representation and a sliding window wherein each position of the sliding window has a path determination stage at one end of the sliding window and a symbol decision stage at another end of the sliding window is disclosed The method comprises determining, for each path determination stage and for each node of the path determination stage, a surviving path (including a surviving path input symbol and a surviving decision stage node) and a concurrent path (including a concurrent path input symbol and a concurrent decision stage node) based on path metrics A path metric disparity value is calculated and stored for each node Based on decision criteria, a soft output value of the surviving decision stage node is determined as either of the path metric disparity value of the node of the path determination stage, a function of the path metric disparity value of the node of the path determination stage and the stored path metric disparity value of the concurrent decision stage node, and the stored path metric disparity value of the surviving decision stage node Corresponding computer program product, decoder and communication apparatus are also disclosed

WO 2011/020770 A1

Published:

— *with international search report (Art. 21(3))*

SOFT OUTPUT VITERBI ALGORITHM METHOD AND DECODER

Technical Field

The present invention relates generally to the field of decoding of error correcting codes. More particularly, **it** relates to decoding of trellis codes.

Background

The principle of iterative decoders is described, for example, in J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes", IEEE Trans. Inf. Theory, vol. 42, no. 2, pp. 429-445, Mar. 1996. Typically, iterative decoding may be based on exchange of de-correlated extrinsic information between soft-input soft-output (SISO) constituent decoders.

Figure IA illustrates an example implementation of iterative decoding using such a setup of constituent SISO decoders 11 and 13 which may be suitable for decoding of e.g. a turbo code with rate 1/2 constituent codes.

In this example, received intrinsic soft values representing two code bits (code symbols in more general cases) are input to the first SISO decoder 11 (Decoder 1) as illustrated by reference numbers 4 1 and 42. Decoder 1 outputs extrinsic soft values for each information bit (or more generally each information symbol) as illustrated by reference number 22, and the extrinsic soft values are interleaved in an interleaver 12.

The interleaved extrinsic soft values are input to the second SISO decoder 13 (Decoder 2) as illustrated by reference number 23, together with the received intrinsic soft values as illustrated by reference numbers 43 and 44 (after interleaving as applicable depending on the particular code setup). Decoder 2 outputs extrinsic soft values for each information bit as illustrated by reference number 24. The extrinsic soft values are deinterleaved in a deinterleaver 14 and fed back to Decoder 1 to serve as input of a second iteration as illustrated by reference number 21, together with the received intrinsic soft values. In the first iteration the input 2 1 is fed with neutral soft values.

This iterative exchange of extrinsic information gradually improves the reliability of the extrinsic information. After an appropriate number of iterations, a hard decision 30 may be taken in relation to each information bit in a decision unit 15 based on extrinsic soft values output from Decoder 2.

The operation of each of the SISO modules 11, 13 of Figure IA may typically be illustrated using a trellis diagram. Figure IB illustrates part of an example trellis diagram 100 representing an example constituent code of a turbo code. In this particular example, each stage of the trellis has four nodes (101, 103, 105, 107 for the first stage, 111, 113, 115, 117 for the second stage, etc.), where each node represents a state of the corresponding encoder. Furthermore, this example trellis has two incoming paths to each node (e.g. I l ia, 11Ib for node 111). Trellis diagrams are very well known in the art and will therefore not be described in greater detail.

The SISO modules themselves may consist of (variants of) the BCJR (Bahl, Cocke, Jelinek, Raviv) algorithm (known from L.R. Bahl *et ah,* "Optimal decoding of linear codes for minimizing symbol error rate", IEEE Trans. Inf. Theory, vol. 20, no. 2, pp. 284-287, Mar. 1974 and from P. Robertson, P. Hóher, and E. Villebrun, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain", Proc. IEEE Intl. Conf. Comm., Seattle, June 1995, pp. 1009-1013 for the logarithmic domain).

However, the BCJR algorithm and its variants suffer from a considerable amount of complexity. To fulfill throughput requirements of some applications an implementation in dedicated hardware may often be required as well as advanced parallelization techniques.

Another possibility for the SISO modules is (variants of) an approach based on the traditional Viterbi algorithm, namely the soft-output Viterbi algorithm (SOVA) disclosed in J. Hagenauer and P. Hóher, "A Viterbi algorithm with soft-decision output and its applications", Proc. IEEE Globecom, Dallas, Nov. 1989, pp. 1680-1686.

Typically, SOVA provides soft output values at a lower complexity cost than the BCJR algorithm. Another advantage of SOVA is a reduction in latency compared to BCJR. Latency of SOVA basically depends on the depth of the decoding window (as known from traditional Viterbi decoding). SOVA may also have the advantage of less parallelization required. In some situations, however, the SOVA suffers from a worse error rate performance than the BCJR algorithm.

As known in the art, SOVA may initially assign a reliability value L to each node of the trellis, where the reliability value depends on the path metrics of the incoming paths of the node. For a trellis with two incoming paths to each node, the initial reliability value may be the absolute value of the difference between the metrics of the two incoming paths, $L \leftarrow \Delta = |M(s') - M(s'')|$, where s' and s'' are the two incoming paths and $M(s)$ is the path metric of path s. The path metric may for example, be a cumulative metric.

In some applications SOVA uses the following update rule, often termed the Hagenauer rule, for the reliability values:

Z-k-u(s) $\leftarrow$ min(Z,k(s),Z,$_k$-u(s)) if u_k-u(s)$\neq$ Wk-u(c) and

Z,k-u(s) unchanged otherwise,

where Zk-u(s) denotes the reliability value of the node of a surviving path at stage k-U . The notations used may be illustrated in a schematic trellis diagram as the one shown in Figure 1C.

In the example of Figure 1C, the SOVA applies a sliding window 120 of length U, which is currently in position k. A surviving path s for the uppermost node 124 at stage k is denoted with reference number 121 and has a corresponding decision bit (or more generally - decision symbol) Wk-u(s), denoted by reference number 126, at stage k-U. Its concurrent path c is denoted with reference number 122 and has a corresponding decision bit Wk-u(c), denoted by reference number 127, at stage k-U. At stage k-1 the node of the surviving path has a concurrent path denoted by reference number 123.

Thus, if the decision bits 126 and 127 differs and if the reliability value of the node 124 of the surviving path at stage k is lower than the reliability value of the node 125 of the surviving path at stage k-U, the Hagenauer rule suggests that the reliability value of node 125 should be replaced by the reliability value of node 124.

We denote a SOVA applying the Hagenauer update rule as HR-SOVA. Hardware architectures using the Hagenauer rule are know from E. Yeo et al., "A 500 Mbit/s soft output Viterbi decoder", IEEE J. Solid-State Circuits, vol. 38, no. 7, pp. 1234-1241, July 2003.

The update rule applied for the reliability values affects the quality (e.g. accuracy) of the extrinsic soft output values and thus the overall performance of the iterative decoder.

Another reliability value update rule that may be used with SOVA in the SISO modules is the Battail rule known from G. Battail, "Ponderation des symboles decodes par l'algorithme de Viterbi", Annales des Telecommuncations, No. 1-2, pp. 31-38, Jan.-Feb. 1987. We denote a SOVA applying the Battail update rule as BR-SOVA.

The Battail rule is equivalent in terms of performance to max-log-MAP (maximum aposteriori) decoding as illustrated in M. Fossorier *et ah,* "On the equivalence between SOVA and max-log-MAP decodings", IEEE Comm. Letters, vol. 2, no. 5, pp. 137-139, May 1998. However, the complexity increase of this update rule compared to the Hagenauer rule is troublesome, and prevents or at least severely obstructs any hardware implementations. The complexity of the Battail rule also counteracts the complexity reduction achieved by using SOVA in stead of BCJR.

For HR-SOVA only the metric differences along the survivor path are considered for updating the reliability values. For BR-SOVA the metric differences along the concurrent path are also used in the updating. This is the cause of the, potential many- fold, increase in complexity of BR-SOVA.

The Battail rule for updating of reliability values may be expressed as:

Z-k-u(s) $\leftarrow$ min(Z,k(s),Z,$_{k}$-u(s)) if Wk-u(s) $\neq$ Wk-u(c) and

Zk-u(s) $\leftarrow$ min(L_{k}(s)+.Lk-u(c),Lk-u(s)) if Wk-u(s) = Wk-u(c).

The Battail update rule may be illustrated in a schematic trellis diagram as the one shown in Figure ID.

In the example of Figure ID, the SOVA applies a sliding window 130 of length U, which is currently in position k. A surviving path s for the uppermost node 133 at stage k is denoted with reference number 131 and has a corresponding decision bit Mk-u(s), denoted by reference number 136, at stage k-U. Its concurrent path c is denoted with reference number 132 and has a corresponding decision bit Wk-u(c), denoted by reference number 137, at stage k-U.

Now, if the decision bits 136 and 137 are equal, the Battail rule suggests that the reliability value of node 138 may be replaced by a sum of the reliability value of node 133 and the reliability value of node 139. Thus, to evaluate of this replacement should take place and, if applicable, perform the replacement, the reliability value of node 139 must first be determined which involves evaluation of the secondary concurrent paths (134, 135, etc.) and results in a complexity increase and a latency increase.

The size and complexity of implementations of a turbo decoder may be an important implementation parameter as well as latency. At the same time, acceptable error rate performance should typically be achieved.

In some scenarios, parallelization is required to meet latency requirements. This may particularly be the case if BCJR-like algorithms are applied. Such parallelization further enhances the importance of small size modules.

On the other hand, low-latency SOVA-type algorithms require a complicated update procedure if MAP baseline performance is to be fully maintained as elaborated on above.

Thus, there is a need for improved SISO algorithms for decoding methods and decoders. Preferably, the improved algorithms should have low latency, low complexity and good error rate performance.

Summary

It should be emphasized that the term "comprises/comprising" when used in this specification is taken to specify the presence of stated features, integers, steps, or components, but does not preclude the presence or addition of one or more other features, integers, steps, components, or groups thereof.

It is an object of the invention to obviate at least some of the above disadvantages and to provide improved SISO algorithms for decoding methods and decoders.

According to a first aspect of the invention, this is achieved by a method of decoding a block with a Soft Output Viterbi Algorithm (SOVA), wherein the block comprises a plurality of symbols having a mutual association described by a trellis representation, the trellis representation having a node for each possible state of an encoder corresponding to the trellis representation and for each stage corresponding to a symbol input occasion of the encoder, each node having at least two incoming path transitions wherein each incoming path transition corresponds to a respective possible input symbol of the encoder, and wherein the SOVA uses a sliding window wherein each position of the sliding window has a path determination stage at one end of the sliding window and a symbol decision stage at another end of the sliding window.

In some embodiments, the block is encoded by an encoder of a code having a trellis representation (e.g. a convolutional code).

The method comprises determining soft output values of the SOVA. This is achieved by (for each path determination stage and for each node of the path determination stage) calculating a transition metric and a corresponding path metric for each incoming path transition, determining a surviving path and a concurrent path based on the path metrics, calculating a path metric disparity value based at least on the path metric of the surviving and concurrent paths, storing the path metric disparity value for the node of the path determination stage, and determining a surviving path input symbol corresponding to a path transition of the surviving path at the decision stage, a surviving decision stage node which has the path transition of the surviving path at the decision stage as an incoming path transition, a concurrent path input symbol corresponding to a path transition of the concurrent path at the decision stage, and a concurrent decision stage node which has the path transition of the concurrent path at the decision stage as an incoming path transition.

The soft output value of the surviving decision stage node is determined as the path metric disparity value of the node of the path determination stage, if the surviving path input symbol differs from the concurrent path input symbol and if the path metric disparity value of the node of the path determination stage is less than the stored path metric disparity value of the surviving decision stage node.

If the surviving path input symbol is equal to the concurrent path input symbol and if the function of the path metric disparity value of the node of the path determination stage and the stored path metric disparity value of the concurrent decision node is less than the stored path metric disparity value of the surviving decision stage node, the soft output value of the surviving decision stage node is determined as a function of the path metric disparity value of the node of the path determination stage and the stored path metric disparity value of the concurrent decision stage node.

Otherwise the soft output value of the surviving decision stage node is determined as the stored path metric disparity value of the surviving decision stage node.

In some embodiments, the transition and path metrics may be cumulative metrics.

The function may be a summation according to some embodiments.

In some embodiments, the step of calculating the path metric disparity value may comprise calculating an absolute value of the difference between the path metrics of the surviving and concurrent paths.

The at least two incoming path transitions of each node may consist of exactly two incoming path transitions according to some embodiments.

The sliding window may, according to some embodiments, be a first sliding window and the determined soft output values may be intermediate soft output values. In such embodiments, the method may further comprise applying a second sliding window, wherein the first and second sliding windows are non-overlapping. For each path determination stage of the second sliding window and for each node of the path determination stage, a soft output value of the surviving decision stage node of the second sliding window may be determined based on at least one of the intermediate soft output values using different criteria than in the first sliding window.

The different criteria may comprise determining the soft output value of the surviving decision stage node as the intermediate soft output value of the surviving decision stage node if the surviving path input symbol is equal to the concurrent path input symbol.

A second aspect of the invention is a computer program product comprising a computer readable medium, having thereon a computer program comprising program instructions. The computer program is loadable into a data-processing unit and adapted to cause the data-processing unit to execute the method steps of the first aspect of the invention when the computer program is run by the data-processing unit.

A third aspect of the invention is a decoder adapted to decode a block using SOVA.

The decoder comprises a metric calculation unit adapted to, for each path determination stage and for each node of the path determination stage, calculate a transition metric and a corresponding path metric for each incoming path transition, and calculate a path metric disparity value based at least on the path metric of a surviving path and a concurrent path.

The decoder also comprises at least one disparity value register unit adapted to store, for each stage and node of the sliding window, the corresponding path metric disparity value.

A survivor path unit of the decoder is adapted to, for each path determination stage and for each node of the path determination stage, determine the surviving path based on the path metrics, and determine a surviving path input symbol corresponding to a path transition of the surviving path at the decision stage, and a surviving decision stage node which has the path transition of the surviving path at the decision stage as an incoming path transition.

A path comparison unit of the decoder is adapted to, for each path determination stage and for each node of the path determination stage, determine the concurrent path based on the path metrics, determine a concurrent path input symbol corresponding to a path transition of the concurrent path at the decision stage, and a concurrent decision stage node which has the path transition of the concurrent path at the decision stage as an incoming path transition, and determine if the surviving path input symbol differs from the concurrent path input symbol.

The decoder further comprises a soft output value unit adapted to, for each path determination stage and for each node of the path determination stage, determine a soft output value of the surviving decision stage node.

The soft output value of the surviving decision stage node is determined as the path metric disparity value of the node of the path determination stage, if the surviving path input symbol differs from the concurrent path input symbol and if the path metric disparity value of the node of the path determination stage is less than the stored path metric disparity value of the surviving decision stage node.

If the surviving path input symbol is equal to the concurrent path input symbol and if the function of the path metric disparity value of the node of the path determination stage and the stored path metric disparity value of the concurrent decision node is less than the stored path metric disparity value of the surviving decision stage node, the soft output value of the surviving decision stage node is determined as a function of the path metric disparity value of the node of the path determination stage and the stored path metric disparity value of the concurrent decision stage node.

Otherwise the soft output value of the surviving decision stage node is determined as the stored path metric disparity value of the surviving decision stage node.

In some embodiments, the at least two incoming path transitions of each node may consist of two incoming path transitions and the decoder may further comprise a state register exchange unit wherein each register cell, corresponding to a node of the sliding window, and each register exchange connection, corresponding to a path transition between nodes, has a word length equal to one.

In such embodiments, the decoder may further comprise means to extract at least one of the concurrent decision stage node and the concurrent path input symbol from the state register exchange unit, wherein the means are adapted to extract the concurrent decision stage node by combining register cell content and/or to extract the concurrent path input symbol by application of an encoding polynomial of the encoder on register cell content.

A fourth aspect of the invention is a radio communication apparatus comprising the decoder of the third aspect of the invention.

In some embodiments, the third and fourth aspects of the invention may additionally have features identical with or corresponding to any of the various features as explained above for the first aspect of the invention.

It is to be noted that embodiments of the invention are applicable to all codes that may be represented by a trellis. Examples of such codes are the class of convolutional codes. Other examples include e.g. speech recognition applications.

An advantage of some embodiments of the invention is that the computational complexity and/or latency is improved (decreased) compared to BR-SOVA and/or BCJR.

Another advantage of some embodiments of the invention is that the complexity of implementations of a turbo decoder may be lowered. In some embodiments, the lower complexity is due to that a lower degree of parallelization is required to achieve a particular throughput.

Another advantage of some embodiments of the invention is that the error rate performance is improved (decreased error probability) compared to HR-SOVA.

Another advantage of some embodiments of the invention is that the error rate performance is comparable to that of BR-SOVA and/or BCJR.

Brief Description of the Drawings

Further objects, features and advantages of the invention will appear from the following detailed description of embodiments of the invention, with reference being made to the accompanying drawings, in which:

Fig. IA is a block diagram illustrating an iterative decoder structure according to some embodiments of the invention;

Fig. IB is a schematic drawing illustrating a trellis representation of an example code according to some embodiments of the invention;

Figs. IC-E are schematic drawings illustrating sliding windows according to some embodiments of the invention;

Fig. 2 is a flowchart illustrating example method steps according to some embodiments of the invention;

Figs. 3A-C are block diagrams illustrating example arrangements according to some embodiments of the invention;

Figs. 4A-B are block diagrams illustrating register unit arrangements according to some embodiments of the invention;

Fig. 5 is a schematic drawing illustrating a mobile terminal, wherein the mobile terminal may comprise an arrangement according to some embodiments of the invention; and

Fig. 6 is a schematic drawing illustrating a computer program product according to some embodiments of the invention.

Detailed Description

In the following, embodiments of the invention will be described where a modified update rule for **SOVA** is applied. **The** modified update rule results in a reduced complexity compared to **BR-SOVA** while achieving better error rate performance than **HR-SOVA.**

The modified update rule for updating of reliability values may, according to some embodiments, be expressed as:

Zk-u(s) $\leftarrow$ min(L_k(s),Lk-u(s)) if Wk-u(s) $\neq$ Wk-u(c) and

Zk-u(s) $\leftarrow$ mm(Z,k(s)+Δ_k-u(c),£k-u(s)) if Wk-u(s) = Wk-u(c),

where Δk-u(c) is the initial reliability value of Wk-u(c). Thus, this modified update rule uses initial reliability values (i.e. metric differences) of the concurrent path in stead of (possibly updated) reliability values as in the Battail rule. Therefore, there is no need to update the reliability values along the concurrent path, which radically improves latency and complexity of the algorithm.

In the Battail rule, Zk-u(c) is supposed to be updated with the minimum reliability value among its possible concurrent paths, i.e. min$_{C}^{'}$(Zk-u(c')) , where c' represents a concurrent path of the concurrent path (i.e. a secondary concurrent path). Therefore, it can be deduced that Δk-u(c) $\geq$ £k-u(c), which implies that Zk(s)+Δk-u(c) $\geq$ Z,k(s)+Z,k-u(c). This implies that, in the modified update rule, Z*-u(s) is updated with a value larger than in the Battail rule, which means that is less likely for the update value to be smaller than Zk-u(s). Therefore, Zk-u(s) is updated less often in the modified update rule than in **BR-SOVA.**

An implementation advantage of SOVA with the modified update rule is that the update of Zk-u(s) can be performed without a delay due to having to update Zk-u(c) first. This decreases complexity and latency. In some implementations SOVA with the modified update rule may also require less memory than **BR-SOVA.**

In some embodiments, the modified update rule is applied in a first sliding window and another update rule (e.g. the Hagenauer rule) is applied in a second sliding window. Such an approach may decrease complexity further without severely affecting the error rate performance.

In general, two non-overlapping sliding windows may be applied, where the sliding windows use different update rules (which may not necessarily comprise the modified update rule). For example, the first sliding window may use the Battail rule and the second may use the Hagenauer rule. This setup is illustrated in Figure IE, where two non-overlapping sliding windows 140 and 150 are illustrated.

Figure 2 illustrates an example method 200 that uses a sliding window approach and which may be used in a SISO decoding module. The method steps of the example method 200 may be performed for each node of each stage in the trellis representation.

Transition metrics for each incoming transition of the node are calculated in step 210, and path metrics for each incoming path are determined in step 220 based on the transition metrics and relevant node metrics of the previous stage. These operations are performed at the leading end of the sliding window (compare with node 124 of figure 1C).

In step 230, the surviving path and its concurrent path are determined based on the calculated path metrics. The determination may include performing trace-back or similar operations (e.g. retrieval of corresponding information from a memory unit, e.g. a register) to determine the decision bit of the surviving path and of the concurrent path (compare with 126 and 127 of Figure 1C) and either or both of their respective reliability values.

A path disparity value (which is to be used as an initial reliability value) for the node under consideration is calculated in step 240 and stored in step 250. The path disparity value may, for example, be an absolute value of the difference between the path metrics of the surviving and concurrent paths. Other disparity values may also be considered. For example, if there are more than two incoming paths all or a subset of the path metrics may be involved in the disparity value calculation. It is noted that steps 240 and 250 may alternatively be performed before or in parallel with step 230.

In step 260, the reliability value of the decision bit of the surviving path (compare with node value 125 in Figure 1C) may be updated if certain criteria are fulfilled as will be explained in the following.

The operations of step 260 may comprise comparing the decision bits Wk-u(s) and Wk-u(c) of the surviving and the concurrent paths (sub-step 261) to determine whether or not they are equal.

If they are not equal, the reliability value $L_k(s)$ for the current node and the reliability value Zk-u(s) f° r the decision bit of the surviving path are compared in sub-step 264. If Zk-u(s) is larger than Z,k(s), Z*-u(s) is replaced by Z,k(s) in sub-step 265. Otherwise Z*-u(s) is kept unchanged (sub-step 266).

If they are equal, a function of the reliability value Z,k(s) for the current node and the initial reliability value Δk-u(c) for the decision bit of the concurrent path is compared to the reliability value Zk-u(s) for the decision bit of the surviving path in sub-step 262. If Zk-u(s) is larger than the function value, Zk-u(s) is replaced by the function value in sub-step 263. Otherwise Zk-u(s) is kept unchanged (sub-step 266). As mentioned above, the function may be a summation, i.e. Zk(s)+Δk-u(c). However, other functions may be equally applicable, e.g. a weighted sum αLk(s)+(2- α)Δk-u(c).

Optional steps 270 and 280 represent embodiments where a second sliding window is applied after the first sliding window has passed and where a different reliability value update rule is used than in the first window. The operations of step 270 for the second window correspond to the operations of step 230 for the first window. Similarly, the operations of step 280 for the second window correspond to the operations of step 260 for the first window with at least one criteria or value formula being different from what has been described for step 260.

Figure 3A illustrates an example decoding arrangement that may be used for HR-SOVA implementations. The arrangement comprises a metric unit 310, a survivor path unit 311, a symbol register 312, a disparity register 313, a selection unit 314, a path comparison unit 315 and a soft value unit 316.

Received intrinsic soft values and extrinsic soft values (e.g. from a previous half-iteration of another SISO module, compare with Figure IA) are input to the metric unit 310 as illustrated at 321 and 322. The metric unit 310 determines the preferred incoming path and calculates the corresponding disparity value (i.e. the initial reliability value) for each node of each stage of the trellis. The operations of the metric unit 310 may be compared with steps 210, 220 and 240 of Figure 2.

The bit value 324 that corresponds to the preferred incoming path is stored in the symbol register 312 for each node, and the corresponding disparity value 325 is stored in the disparity register 313 (compare with step 250 of Figure 2).

The bit value 323 that corresponds to the preferred incoming path is also forwarded to the survivor path unit 311, which establishes relevant information associated with the surviving path (compare with step 230 - surviving path part - of Figure 2), such as the decision bit 326 and the decision stage survivor node 327 (compare with 125 and 126 of Figure 1C).

The information associated with the surviving path is forwarded to the path comparison unit 315 which relays the decision bit to an arrangement output 331. The path comparison unit also establishes relevant information associated with the concurrent path (compare with step 230 - concurrent path part - of Figure 2), such as the bit corresponding to the decision bit and the decision stage concurrent node. For this purpose, the path comparison unit 315 also has access to the symbol register 312 as shown at 328.

The path comparison unit 315 also compares the decision bit with the corresponding bit of the concurrent path (compare with sub-step 261 of Figure 2) and forwards the result 332 of the comparison to the soft value unit 316, which updates reliability values accordingly (e.g. according to the Hagenauer rule) and outputs the resulting reliability values as soft values as illustrated at 333. For this purpose, the soft value unit 316 also has access to adequate information of the disparity register 314 as shown at 330. The selection unit 314 selects the appropriate disparity value 330 from the values of the disparity register 239 based on the decision stage node 327.

The registers 312 and 313, along with the survivor path unit 311, typically cover a certain number of stages of the trellis at a time and may be seen as a sliding pre-window in which traditional Viterbi algorithm operations are performed and the disparity values are prepared for further operations. The path comparison unit 315 and the soft value unit 316 typically operate in a sliding window that follows the sliding pre-window. This arrangement of the sliding pre-window and the sliding window ensures that reliability updates need only be done for the surviving path of the node that corresponds to the bit decision value (i.e. at most one update per trellis stage). Other implementations may use only a single sliding window (and no sliding pre-window), which reduces latency but increases complexity since then the reliability updates need to be done for the surviving path of each node of the trellis.

Figure 3B illustrates another example decoding arrangement that may be used for BR-SOVA implementations or for implementations of SOVA with the modified update rule according to some embodiments of the invention. The arrangement comprises a metric unit 340, a survivor path unit 341, a disparity register 343, a selection and absolute value unit 344, a path comparison unit 345, a selection unit 347 and a soft value unit 346.

Received intrinsic soft values and extrinsic soft values (e.g. from a previous half-iteration of another SISO module, compare with Figure IA) are input to the metric unit 340 as illustrated at 35 1 and 352. The metric unit 340 determines the preferred incoming path and calculates the corresponding disparity value for each node of each stage of the trellis. The disparity values 355 are stored in the disparity register 343.

The bit value 353 that corresponds to the preferred incoming path is forwarded to the survivor path unit 341, which establishes relevant information associated with the surviving path, such as the decision bit 356 and the decision stage survivor node 357.The information associated with the surviving path is forwarded to the path comparison unit 345 which relays the decision bit to an arrangement output 361 . The path comparison unit also establishes relevant information associated with the concurrent path, such as the bit corresponding to the decision bit and the decision stage concurrent node. Such information may, for example, be extracted from the sign of the disparity values (as in the Hagenauer rule). Disparity values are commonly defined as absolute values in with case their sign contains no information. Therefore, embodiments of the invention describe disparity values as signed values since such an approach simplifies the implementation of the metric unit. The absolute value of the disparity values may be determined at a later stage of the processing according to such embodiments.

The path comparison unit 345 also compares the decision bit with the corresponding bit of the concurrent path and forwards the result 365 of the comparison to the selection unit 347.

Furthermore, the path comparison unit 345 provides information 364 relevant for a possible reliability update to the selection unit 347. For BR-SOVA, such information may include (updated) reliability values of the concurrent path, while for SOVA with the modified update rule such information may include initial reliability values of the concurrent path. For this purpose, the path comparison unit 345 also has access to the disparity register 343 as shown at 358.

The selection and absolute value unit 344 selects an appropriate disparity value (i.e. the one corresponding to the surviving path) from the values of the disparity register 259 based on the decision stage node 357, and provides its absolute value 360 to the selection unit 347 and the soft value unit 346.

The selection unit 347 determined whether or not the information 364 relevant for a possible reliability update is to be used based whether or not the decision bit was equal to the corresponding bit of the concurrent path (i.e. the information provided at 365) and outputs the result at 362. Thus, for SOVA with the modified update the result at 362 may equal $L_k(s)$ if u_{k}-u(s) $\neq$ Wk-u(c) and z,$_{\text{k}}$(s)+Δ_{k}-u(c) if u_{k}-u(s) = Wk-u(c).

Then, the soft value unit 346 updates reliability values accordingly and outputs the resulting reliability values as soft values as illustrated at 363. For example, if either of BR-SOVA or SOVA with the modified update rule are implemented, the soft output value may compare the result at 362 with the disparity value 360 (compare with sub-steps 262 and 264 of Figure 2) and output the least of the two (compare with sub-steps 263, 265 and 266 of Figure *X).*

Figure 3C illustrates yet another example decoding arrangement that may be used for a combination of two different SOVA implementations (e.g. BR-SOVA or SOVA with the modified update rule, followed by HR-SOVA) according to some embodiments of the invention.

The arrangement comprises a metric unit 370, a survivor path unit 371, a disparity register 373, a selection and absolute value unit 374, a path comparison unit 375, a selection unit 377 and a soft value unit 376. These units operate similarly or equivalently with the corresponding units 340, 341, 343, 344, 345, 347 and 346 of Figure 3B and will therefore not be described in detail.

The arrangement further comprises a second path comparison unit 379 and a second soft value unit 378.

The path comparison unit 375 forwards the decision bits 391 and the decision stage survivor nodes 396 to the second path comparison unit 379. The soft value unit 376 forwards its resulting reliability values 393 to the second soft value unit 378, which uses them as initial reliability values. The path comparison unit 379 and the soft value unit 378 operate similarly or equivalently with the corresponding units 315 and 316 of Figure 3A and will therefore not be described in detail. Values corresponding to the symbol register of Figure 3A may be provided to the path comparison unit 379 in Figure 3C e.g. by extracting the values from the sign of the disparity values as has been explained above.

The path comparison unit 375, the selection unit 377 and the soft value unit 376 typically cover a certain number of stages of the trellis at a time and may be seen as operating in a first sliding window. The path comparison unit 379 and the soft value unit 378 typically cover another certain number of stages of the trellis at a time and may be seen as operating in a second sliding window. The first and second sliding windows may or may not be overlapping, but are preferably non-overlapping for optimal performance.

It is noteworthy that the reliability update procedure used by the second soft value unit 378 differs for the reliability update procedure used by the soft value unit 376 and the selection unit 377. For example, the soft value unit 376 and the selection unit 377 may apply the modified update rule or the Battail rule, while the second soft value unit 378 may apply the Hagenauer rule.

The operations of units 375, 377 and 376 may be compared with the application of a first sliding window 140 in Figure IE and the operations of units 379 and 378 may be compared with the application of a second sliding window 150 in Figure IE and with steps 270 and 280 of Figure 2.

Thus, according to some embodiments a method of decoding a block with a Soft Output Viterbi Algorithm (SOVA) is provided wherein the block is encoded by an encoder of a code having a trellis representation, the trellis representation having a node for each possible state of the encoder and for each stage corresponding to a symbol input occasion of the encoder, each node having at least two incoming path transitions wherein each incoming path transition corresponds to a respective possible input symbol of the encoder. In these embodiments, the SOVA uses a first sliding window wherein each position of the sliding window has a path determination stage at one end of the sliding window and a symbol decision stage at another end of the sliding window. The SOVA also uses a second sliding window wherein each position of the sliding window has a path determination stage at one end of the sliding window and a symbol decision stage at another end of the sliding window. The first and second sliding windows may preferably be non-overlapping.

The method according to such embodiments may comprise determining intermediate soft output values of the SOVA by, for each path determination stage of the first sliding window and for each node of the path determination stage: calculating a transition metric and a corresponding path metric for each incoming path transition, determining a surviving path and a concurrent path based on the path metrics, calculating a path metric disparity value based at least on the path metric of the surviving and concurrent paths, storing the path metric disparity value for the node of the path determination stage, and determining a surviving path input symbol corresponding to a path transition of the surviving path at the decision stage, a surviving decision stage node which has the path transition of the surviving path at the decision stage as an incoming path transition, a concurrent path input symbol corresponding to a path transition of the concurrent path at the decision stage, and a concurrent decision stage node which has the path transition of the concurrent path at the decision stage as an incoming path transition.

The intermediate soft output value of the surviving decision stage node may be determined as the path metric disparity value of the node of the path determination stage, if the surviving path input symbol differs from the concurrent path input symbol and if the path metric disparity value of the node of the path determination stage is less than the stored path metric disparity value of the surviving decision stage node.

If the surviving path input symbol is equal to the concurrent path input symbol and if the function of the path metric disparity value of the node of the path determination stage and the stored path metric disparity value of the concurrent decision node is less than the stored path metric disparity value of the surviving decision stage node, intermediate soft output values of the surviving decision stage node may be determined as a function of the path metric disparity value of the node of the path determination stage and the stored path metric disparity value of the concurrent decision stage node. In some embodiments, intermediate soft output values of the surviving decision stage node may instead be determined as a function of the path metric disparity value of the node of the path determination stage and a reliability value of the concurrent decision stage node, if the surviving path input symbol is equal to the concurrent path input symbol and if the function of the path metric disparity value of the node of the path determination stage and the reliability value of the concurrent decision node is less than the stored path metric disparity value of the surviving decision stage node.

Otherwise, intermediate soft output value of the surviving decision stage node may be determined as the stored path metric disparity value of the surviving decision stage node.

The method according to such embodiments may further comprise determining soft output values of the SOVA by, for each path determination stage of the second sliding window and for each node of the path determination stage: determining a concurrent path input symbol corresponding to a path transition of the concurrent path at the decision stage, and a concurrent decision stage node which has the path transition of the concurrent path at the decision stage as an incoming path transition.

The soft output values of the surviving decision stage node may be determined using rules that are different in at least one aspect from those used to determine the intermediate soft output values of the first sliding window. The soft output values of the surviving decision stage node may be determined based on at least one of the intermediate soft output values and using different criteria than in the first sliding window.

The soft output values of the surviving decision stage node may, in some embodiments, be determined as the intermediate soft output value of the node of the path determination stage, if the surviving path input symbol differs from the concurrent path input symbol and if the intermediate soft output value of the node of the path determination stage is less than the intermediate soft output value of the surviving decision stage node.

Otherwise, the soft output value of the surviving decision stage node may be determined as the intermediate soft output value of the surviving decision stage node.

A corresponding decoder may also be envisioned comprising a metric calculation unit, at least one disparity value register unit, a survivor path unit, first and second path comparison units, and first and second soft output value units. The various parts being adapted to perform respective steps of the method as described above.

These embodiments form a class of hybrid reliability updating.

An advantage with such embodiments when the modified update rule is applied in the first sliding window is that the complexity (compared to BR-SOVA or BCJR) may be even further reduced. Another advantage is that the error rate performance is not (or merely slightly) worsened compared to using only the modified update rule with a sliding window length that is equal to the sum of the lengths of the first and second sliding window lengths of the hybrid setup.

Typically, there may be a trade-off between error rate performance and complexity in these embodiments. The trade-off may be illustrated by varying the lengths of the first and second sliding windows while keeping the total window length constant.

These and other advantages may be achieved by the combination of a SOVA with low-complexity properties (e.g. HR-SOVA), and a SOVA with performance-maintaining properties (e.g. BR-SOVA or SOVA with the modified update rule).

As illustrated in Figure 3C, one implementation achieves this by cascading the reliability update units (path comparison units 375, 379, and selection unit 377 and soft value units 376, 378) serially. By doing so, the likelihood that the reliability for the best concurrent path is eliminated within the first sliding window is reduced compared to if only HR-SOVA is used, and thus the overestimation of the extrinsic information in the

HR-SOVA is less pronounced. At the same time, the complexity increase from using a SOVA with performance-maintaining properties is kept low by applying the combination with HR-SOVA.

To keep track of and to be able to obtain the relevant metric differences (disparity values) for the survivor path and/or the concurrent path in a swift and convenient way, certain implementation considerations may be attended to.

In a decoder that utilizes the Viterbi algorithm, register exchange is a low-latency method for tracing the survivor path back to a merged state in order to put out a decision symbol. Register exchange may be used for survivor management to obtain a possible hard decision symbol, as well as to track the survivor and/or concurrent paths within the reliability value update units (e.g. soft value units if Figure 3A-C). Traditional register exchange units perform register exchange on hard decision bits, which is useful to obtain a merged final decision bit. However, such an approach is not suitable for SOVA reliability value updates, where the metric differences for the concurrent path need to be obtained (e.g. SOVA with the modified update rule as described above).

One way of obtaining the metric difference for the concurrent path is to perform register exchange on an array of metric differences. Depending on the bit width (word length) of the metric differences, this may result in a large number of multiplexers and undesirable wiring in the circuit. An alternative way involves storing the array of metric differences in an array of shift registers. However, in such cases the sequence of concurrent states of a path typically needs to be available in order to enable selection of the relevant metric difference values. Performing direct register exchange on the (generally multi-bit) state value is possible, but is not always an optimal solution.

In the cases where the encoder takes one input bit per input occasion (e.g. for rate 1/X convolutional encoders), each state is associated with two possible input transitions and two possible output transitions, one for decision bit '0' and the other for decision bit ' 1'. By controlling the initial inputs to the register exchange unit and by having it exchange the minimum required information for each state transition, register exchange of states can in these situations be performed by exchanging a single bit of data.

Figure 4A illustrates an example state register exchange unit according to some embodiments of the invention. Each stage of a corresponding trellis is represented by an appropriate number of memory elements 405, 406, 407, 408, each having a bit-width equal to one. A corresponding number of multiplexers 401, 402, 403, 404 inputs one of two possible values (either from initial input alternatives 0/1 in the first stage - 411, 412, 414, 415, 417, 418, 420, 421 - or from one of two previous memory element alternatives in later stages) based on the state transition having the best metric at the path determination stage 413, 416, 419, 422. This example implementation represents a minimal implementation since all of the memory elements and all wiring have a bit width equal to one.

Since only a single bit of the state value is exchanged, a transformation may be required to reconstruct the state and/or the output decision bit in case of multi-bit state values.

Figure 4B illustrates an example of such a transformation. In this example, the encoding polynomial is utilized to extract the relevant decision bit 430 by addition 531 of corresponding memory element content of a row of the register exchange unit, while the relevant state 440 may be extracted from corresponding memory element content of a row of the register exchange unit.

By performing such transformations for an entire row of the register exchange unit, the state sequence of a path (e.g. the concurrent path) may be efficiently obtained, and the relevant metric differences may thereafter be obtained from the disparity registers.

Figure 5 illustrates an example mobile terminal 500. The mobile terminal 500 may comprise a decoder and/or may perform methods according to embodiments of the invention. The mobile terminal 500 may, for example, comprise a decoder as described in connection to any of Figures 3B and 3C.

The example mobile terminal 500 may be adapted to connect to a mobile telecommunication network via the wireless link to a radio base station. To this end, the mobile terminal 500 and the base station may be compliant with at least one mobile telecommunication standard, for instance UMTS LTE (Universal Mobile Telecommunication Standard - Long Term Evolution).

The described embodiments of the invention and their equivalents may be realised in software or hardware or a combination thereof. They may be performed by general-purpose circuits associated with or integral to a communication device, such as digital signal processors (DSP), central processing units (CPU), co-processor units, field-programmable gate arrays (FPGA) or other programmable hardware, or by specialized circuits such as for example application-specific integrated circuits (ASIC). All such forms are contemplated to be within the scope of the invention.

The invention may be embodied within an electronic apparatus comprising circuitry/logic or performing methods according to any of the embodiments of the invention. The electronic apparatus may, for example, be radio communication apparatus, a portable or handheld mobile radio communication equipment, a mobile radio terminal, a mobile telephone, a base station, a pager, a communicator, an electronic organizer, a smartphone, a computer, a notebook, a modem, a plug-in card, or a mobile gaming device.

According to some embodiments of the invention, a computer program product comprises a computer readable medium such as, for example, a USB-stick, a diskette or a CD-ROM. Figure 6 is a schematic drawing illustrating a computer readable medium in the form of a CD-ROM 600 according to some embodiments of the invention. The computer readable medium may have stored thereon a computer program comprising program instructions. The computer program may be loadable (as shown by arrow 610) into an electronic device 620 comprising a processing unit 640 and possibly a separate memory unit 630. The electronic device 620 may, for example, be a mobile terminal. When loaded into the electronic device 620, the computer program may be stored in the memory unit 630. According to some embodiments, the computer program may, when loaded into the electronic device 620 and run by the processing unit 640, cause the electronic device 620 to execute method steps according to, for example, the method shown in Figure 2.

The invention has been described herein with reference to various embodiments. However, a person skilled in the art would recognize numerous variations to the described embodiments that would still fall within the scope of the invention. For example, the method embodiments described herein describes example methods through method steps being performed in a certain order. However, it is recognized that these sequences of events may take place in another order without departing from the scope of the invention. Furthermore, some method steps may be performed in parallel even though they have been described as being performed in sequence.

In the same manner, it should be noted that in the description of embodiments of the invention, the partition of functional blocks into particular units is by no means limiting to the invention. Contrarily, these partitions are merely examples. Functional blocks described herein as one unit may be split into two or more units. In the same manner, functional blocks that are described herein as being implemented as two or more units may be implemented as a single unit without departing from the scope of the invention.

Hence, it should be understood that the limitations of the described embodiments are merely for illustrative purpose and by no means limiting. Instead, the scope of the invention is defined by the appended claims rather than by the description, and all variations that fall within the range of the claims are intended to be embraced therein.

CLAIMS

1. A method of decoding a block with a Soft Output Viterbi Algorithm - SOVA, wherein the block comprises a plurality of symbols having a mutual association described by a trellis representation, the trellis representation having a node for each possible state of an encoder corresponding to the trellis representation and for each stage corresponding to a symbol input occasion of the encoder, each node having at least two incoming path transitions wherein each incoming path transition corresponds to a respective possible input symbol of the encoder, and

wherein the SOVA uses a sliding window (120, 130, 140) wherein each position of the sliding window has a path determination stage at one end of the sliding window and a symbol decision stage at another end of the sliding window,

the method comprising determining soft output values of the SOVA by, for each path determination stage and for each node of the path determination stage:

calculating (210, 220) a transition metric and a corresponding path metric for each incoming path transition;

determining (230) a surviving path and a concurrent path based on the path metrics;

calculating (240) a path metric disparity value based at least on the path metric of the surviving and concurrent paths;

storing (250) the path metric disparity value for the node of the path determination stage;

determining (230) a surviving path input symbol corresponding to a path transition of the surviving path at the decision stage, a surviving decision stage node which has the path transition of the surviving path at the decision stage as an incoming path transition, a concurrent path input symbol corresponding to a path transition of the concurrent path at the decision stage, and a concurrent decision stage node which has the path transition of the concurrent path at the decision stage as an incoming path transition;

determining (260, 263, 265, 266) a soft output value of the surviving decision stage node as:

the path metric disparity value of the node of the path determination stage, if (261) the surviving path input symbol differs from the concurrent path input symbol and if (264) the path metric disparity value of the node of the path determination stage is less than the stored path metric disparity value of the surviving decision stage node; a function of the path metric disparity value of the node of the path determination stage and the stored path metric disparity value of the concurrent decision stage node, if (261) the surviving path input symbol is equal to the concurrent path input symbol and if (262) the function of the path metric disparity value of the node of the path determination stage and the stored path metric disparity value of the concurrent decision node is less than the stored path metric disparity value of the surviving decision stage node; and

the stored path metric disparity value of the surviving decision stage node otherwise.

2. The method of claim 1, wherein the transition and path metrics are cumulative metrics.

3. The method of any of claims 1 through 2, wherein the function is a summation.

4. The method of any of claims 1 through 3, wherein the step of calculating the path metric disparity value comprises calculating an absolute value of the difference between the path metrics of the surviving and concurrent paths.

5. The method of any of claims 1 through 4, wherein the at least two incoming path transitions of each node consists of two incoming path transitions.

6. The method of any of claims 1 through 5, wherein the sliding window is a first sliding window (140) and wherein the determined soft output values are intermediate soft output values, the method further comprising:

applying a second sliding window (150), wherein the first and second sliding windows are non-overlapping; and

determining, for each path determination stage of the second sliding window and for each node of the path determination stage, a soft output value of the surviving decision stage node of the second sliding window based on at least one of the intermediate soft output values using different criteria than in the first sliding window.

7. The method of claim 6, wherein the different criteria comprises determining the soft output value of the surviving decision stage node as the intermediate soft output value of the surviving decision stage node if the surviving path input symbol is equal to the concurrent path input symbol.

8. A computer program product (600) comprising a computer readable medium, having thereon a computer program comprising program instructions, the computer program being loadable into a data-processing unit and adapted to cause the data-processing unit to execute the steps of any of claims 1 to 7 when the computer program is run by the data-processing unit.

9. A decoder adapted to decode a block with a Soft Output Viterbi Algorithm - SOVA,

wherein the block comprises a plurality of symbols having a mutual association described by a trellis representation, the trellis representation having a node for each possible state of an encoder corresponding to the trellis representation and for each stage corresponding to a symbol input occasion of the encoder, each node having at least two incoming path transitions wherein each incoming path transition corresponds to a respective possible input symbol of the encoder, and

wherein the SOVA uses a sliding window (120, 130, 140) wherein each position of the sliding window has a path determination stage at one end of the sliding window and a symbol decision stage at another end of the sliding window,

the decoder comprising:

a metric calculation unit (340, 370) adapted to, for each path determination stage and for each node of the path determination stage:

calculate a transition metric and a corresponding path metric for each incoming path transition; and

calculate a path metric disparity value based at least on the path metric of a surviving path and a concurrent path;

at least one disparity value register unit (343, 373) adapted to store, for each stage and node of the sliding window, the corresponding path metric disparity value;

a survivor path unit (341, 371) adapted to, for each path determination stage and for each node of the path determination stage:

determine the surviving path based on the path metrics; and

determine a surviving path input symbol corresponding to a path transition of the surviving path at the decision stage, and a surviving decision stage node which has the path transition of the surviving path at the decision stage as an incoming path transition;

a path comparison unit (345, 375) adapted to, for each path determination stage and for each node of the path determination stage:

determine the concurrent path based on the path metrics;

determine a concurrent path input symbol corresponding to a path transition of the concurrent path at the decision stage, and a concurrent decision stage node which has the path transition of the concurrent path at the decision stage as an incoming path transition; and

determine if the surviving path input symbol differs from the concurrent path input symbol;

characterized by:

a soft output value unit (346, 376) adapted to, for each path determination stage and for each node of the path determination stage, determine a soft output value of the surviving decision stage node as:

the path metric disparity value of the node of the path determination stage, if the surviving path input symbol differs from the concurrent path input symbol and if the path metric disparity value of the node of the path determination stage is less than the stored path metric disparity value of the surviving decision stage node;

a function of the path metric disparity value of the node of the path determination stage and the stored path metric disparity value of the concurrent decision stage node, if the surviving path input symbol is equal to the concurrent path input symbol and if the function of the path metric disparity value of the node of the path determination stage and the stored path metric disparity value of the concurrent decision stage node is less than the stored path metric disparity value of the surviving decision stage node; and

the stored path metric disparity value of the surviving decision stage node otherwise.

10. The decoder of claim 9, wherein the at least two incoming path transitions of each node consists of two incoming path transitions and further comprising a state register exchange unit wherein each register cell, corresponding to a node of the sliding window, and each register exchange connection, corresponding to a path transition between nodes, has a word length equal to one.

11. The decoder of claim 10, further comprising means to extract at least one of the concurrent decision stage node and the concurrent path input symbol from the state register exchange unit, wherein the means are adapted to extract the concurrent decision stage node by combining register cell content and/or to extract the concurrent path input symbol by application of an encoding polynomial of the encoder on register cell content.

12. The decoder of any of claims 9 through 11, wherein the sliding window is a first sliding window (140), wherein the determined soft output values are intermediate soft output values, and wherein the at least one disparity value register unit is also adapted to store the corresponding path metric disparity value, for each stage and node of a second sliding window (150) wherein the first and second sliding windows are non-overlapping, the decoder further comprising:

a second soft output value unit (378) adapted to, for each path determination stage of the second sliding window and for each node of the path determination stage, determine a soft output value of the surviving decision stage node based on at least one of the intermediate soft output values using different criteria than in the first sliding window.

13. The decoder of claim 12, wherein the second soft output value unit is adapted to determine the soft output value of the surviving decision stage node as the intermediate soft output value of the surviving decision stage node if the surviving path input symbol is equal to the concurrent path input symbol.

14. A radio communication apparatus (500) comprising the decoder of any of claims 9 through 13.

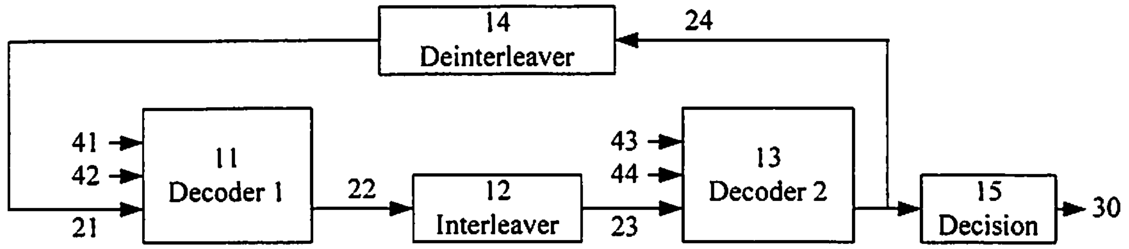

Fig. 1A

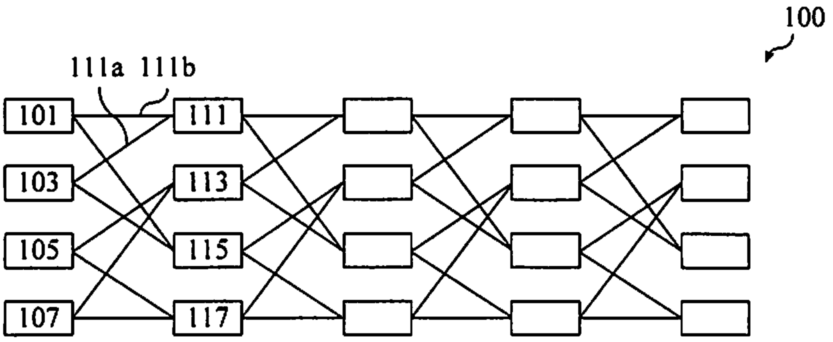

Fig. 1B

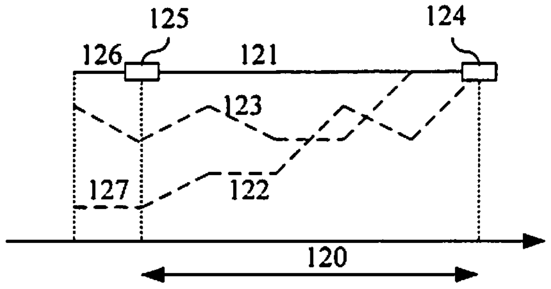

Fig. 1C

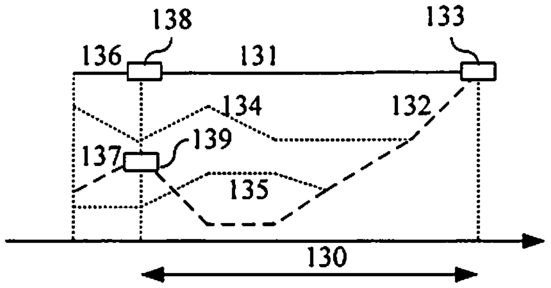

Fig. 1D

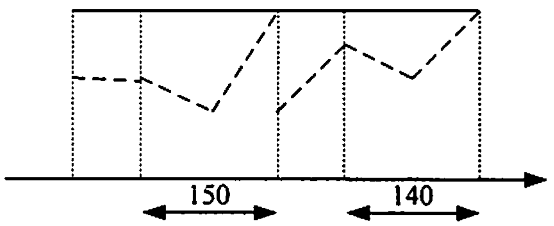

Fig. 1E

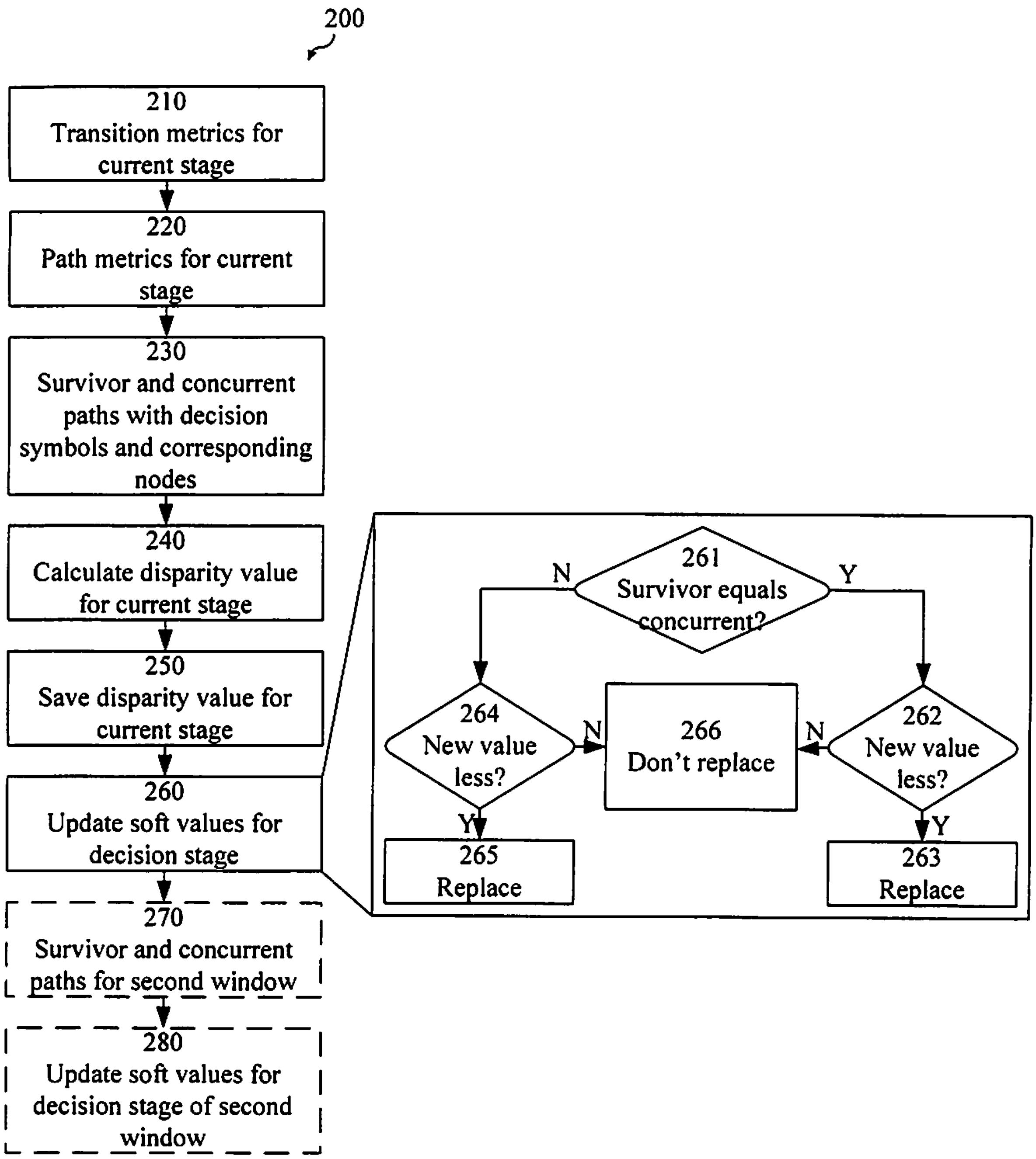
200
210
Transition metrics for current stage
220
Path metrics for current stage
230
Survivor and concurrent paths with decision symbols and corresponding nodes
240
Calculate disparity value for current stage
250
Save disparity value for current stage
260
Update soft values for decision stage
270
Survivor and concurrent paths for second window
280
Update soft values for decision stage of second window
261
Survivor equals concurrent?
N
Y
264
New value less?
N
Y
266
Don't replace
262
New value less?
N
Y
265
Replace
263
Replace

Fig. 2

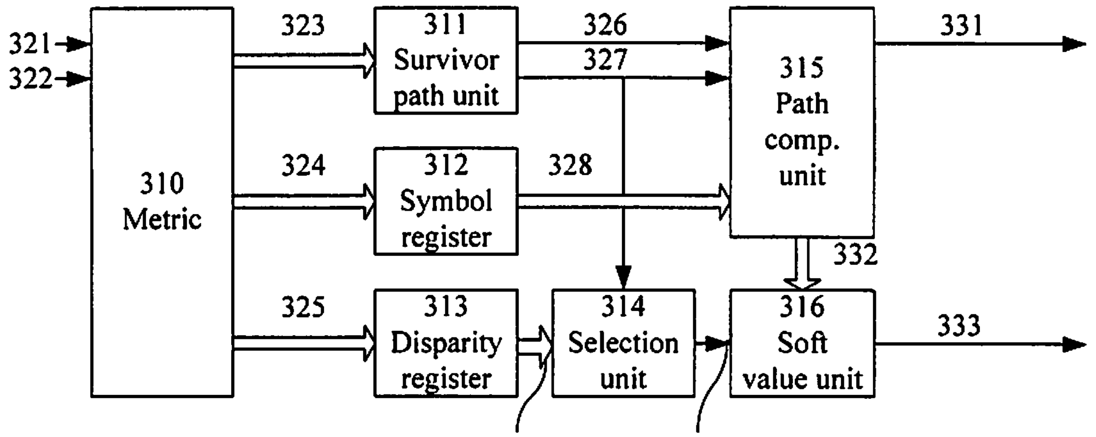

Fig. 3A

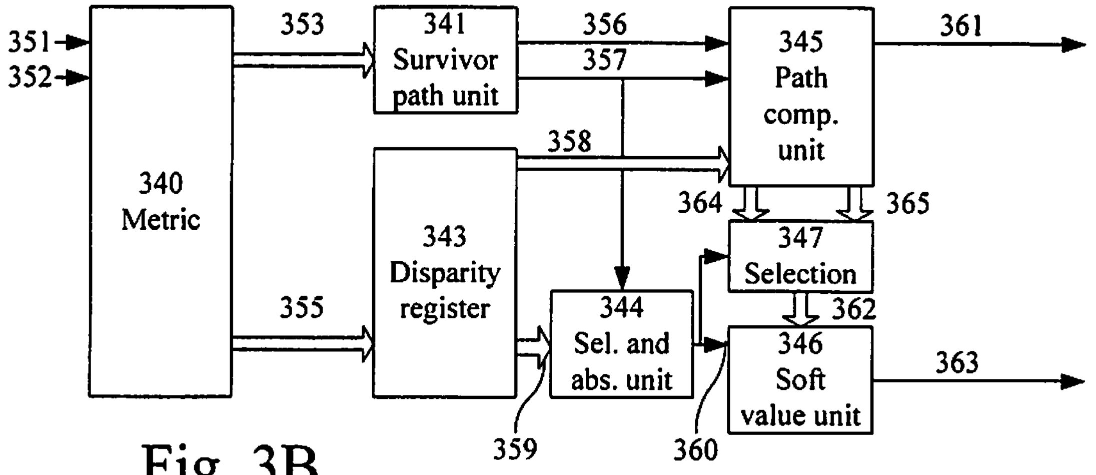

Fig. 3B

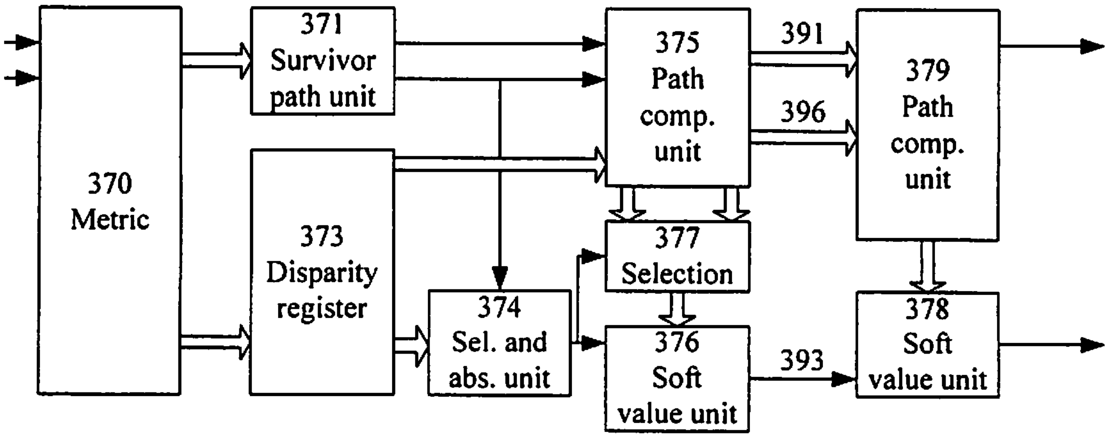

Fig. 3C

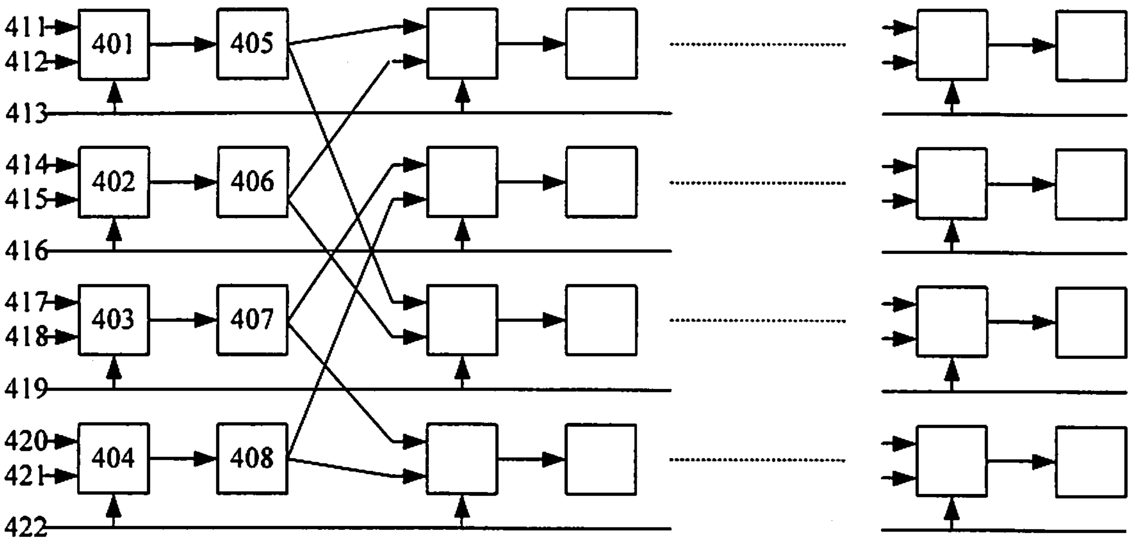

Fig. 4A

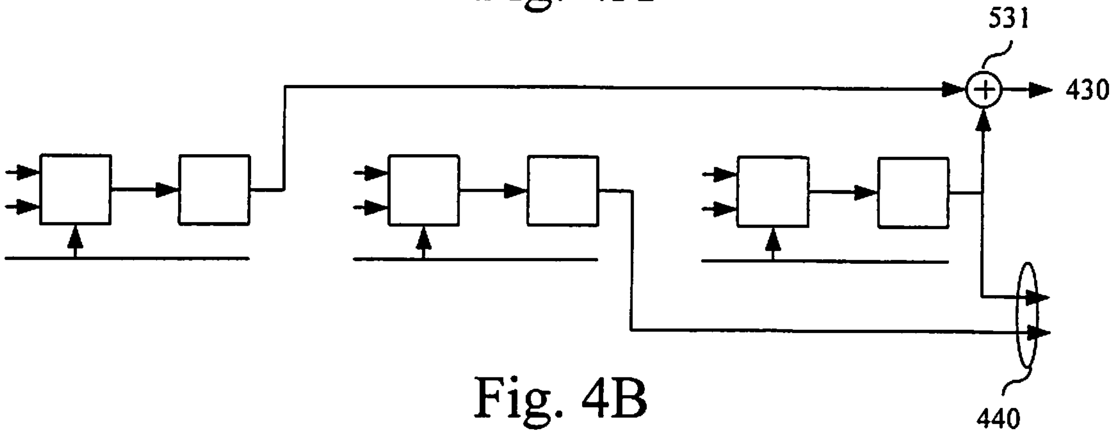

Fig. 4B

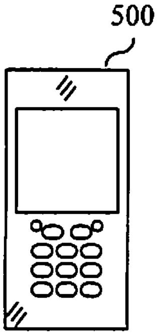

Fig. 5

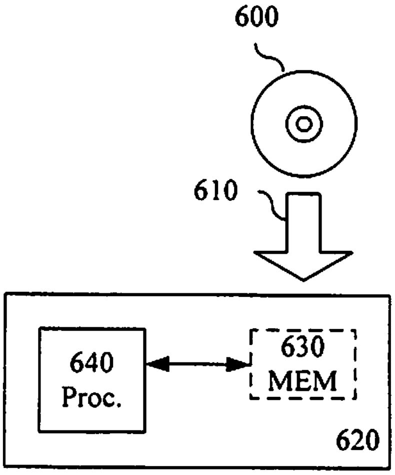

Fig. 6

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2010/061760

A. CLASSIFICATION OF SUBJECT MATTER
INV . H03M13/41
ADD.

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

B. FIEi. DS 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 , INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication where appropriate, of the relevant passages	Relevant to claim No
X	Hsin-Yih Li: "A Modified Soft-Output Viterbi Algorithm for Punctured Turbo-Codes" , Master Thesis, Dept. Electrical Engineering, National cheng Kung University, Tainan, Taiwan, July 2006, 9 September 2006 (2006-09-09), pages 1-78, XP002601407, Retrieved from the Internet: URL :http ://etdπcku .lib. ncku .edu .tw/ETD-db/ETD-search/view_etd?URN=etd-0909106-201156 [retrieved on 2010-09-16]	1-5,8,9, 14
Y	* abstract page 13 - page 41 ----- -/--	6,7, 10-13

☒ Further documents are listed in the continuation of Box C

D 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 claιm(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O' document referπng 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 pπonty date and not in conflict with the application but cited to understand the pnnciple 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 docu– ments, 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 26 November 2010	Date of mailing of the international search report 06/12/2010
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	Authonzed officer Offer, Elke

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	TAKESHITA 0 Y ET AL: "Asymmetric turbo-codes", INFORMPROC. , IEEE INTERNATIONAL SYMPOSIUM ON INFORMATION THEORY, ISIT'98, CAMBRIDGE, MA, USA, 16 August 1998 (1998-08-16), - 21 August 1998 (1998-08-21), page 179, XP010297107, ISBN: 978-0-7803-5000-7 the whole document -----	6,7,12, 13
Y	JOERESSEN 0 J ET AL: "HIGH-SPEED VLSI ARCHITECTURES FOR SOFT-OUTPUT VITERBI DECODING", JOURNAL OF VLSI SIGNAL PROCESSING SYSTEMS FOR SIGNAL, IMAGE, AND VIDEO TECHNOLOGY, SPRINGER, NEW YORK, NY, US, vol. 8 , no. 2 , 1 October 1994 (1994-10-01) , pages 169-181, XP000483302, ISSN: 0922-5773, DOI: DOI :10 .1007/BF02109383 page 174, right-hand column - page 175, right-hand column -----	10,11
X ,P	ANG L .-H. ET AL.: "Modification of SOVA-based Algorithms for Efficient Hardware Implementation", PROC, IEEE VEHICULAR TECHNOLOGY CONFERENCE, VTC 2010-SPRING, TAIPEI, TAIWAN, 16 May 2010 (2010-05-16), - 19 May 2010 (2010-05-19), pages 1-5, XP002602413, ISBN: 978-1-4244-2518-1	1-9, 12-14
Y ,P	the whole document -----	10,11
A	MARC P C FOSSORIER ET AL: "On the Equivalence Between SOVA and MAx-Log-MAP Decodings", IEEE COMMUNICATIONS LETTERS, IEEE SERVICE CENTER, PISCATAWAY , NJ, US, vol. 2 , no. 5 , 1 May 1998 (1998-05-01), XP011010575, ISSN: 1089-7798 cited in the application the whole document -----	1-14

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2010/061760

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons

1 □ Claims Nos
because they relate to subject matter not required to be searched by this Autho πty, namely

2 □ Claims Nos
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out specifically

3 □ Claims Nos
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6 4(a)

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application as follows

see additional sheet

1 ☒ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims

2 □ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees

3 □ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos

4 □ No required additional search fees were timely paid by the applicant Consequently, this international search report is restricted to the invention first mentioned in the claims, it is covered by claims Nos

Remark on Protest

□ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee

□ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation

☒ NO protest accompanied the payment of additional search fees

FURTHER INFORMATION CONTINUED FROM POT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-9, 12-14

Independent claims 1, 8 and 9 relate to a soft output value calculation method which is a modification of Battail's rule within a SOVA. Dependent claims 2-5 further specify details of said modification, dependent claim 14 specifies the implementation with a radio communication apparatus and dependent claims 6-7 and 12-13 specify that said modified Battail's rule is used within a first sliding window and that a different criteria for calculating the soft output is used within a second sliding window.

---

2. claims: 10, 11

Claims 10-11 depend on independent claim 9 and further specify an efficient register exchange unit for implementing the trace back operation in order to obtain the concurrent decision stage node and the concurrent path input symbol.
