



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
15/564,248	10/04/2017	Mattias ANDERSSON	017997.1279	2262

5073 7590 05/09/2019

BAKER BOTTS L.L.P.
2001 ROSS AVENUE
SUITE 900
DALLAS, TX 75201-2980

EXAMINER

VALLECILLO, KYLE

ART UNIT	PAPER NUMBER
----------	--------------

2112

NOTIFICATION DATE	DELIVERY MODE
-------------------	---------------

05/09/2019

ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptomail1@bakerbotts.com

ptomail2@bakerbotts.com

DETAILED ACTION

Notice of Pre-AIA or AIA Status

1. The present application, filed on or after March 16, 2013, is being examined under the first inventor to file provisions of the AIA.

Drawings

2. Figure 1 should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g). Corrected drawings in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(a)(1) the claimed invention was patented, described in a printed publication, or in public use, on sale or otherwise available to the public before the effective filing date of the claimed invention.

4. Claims 1-5, 7-8, 10-18, 21, 23-31, 33-34, 36-44, 47 and 49-52 are rejected under 35 U.S.C. 102(a)(1) as being anticipated by Blankenship et al. (U.S. Patent Publication No. 2008/0320353 A1), hereinafter referred to as Blankenship.

Regarding claim 1, Blankenship discloses: **A method of producing a set of coded bits from a set of information bits for transmission between a first node and a second node in a wireless communications system** (Figures 1, 3; ¶¶ 1, 4), **the method comprising:**

generating a codeword vector by encoding the set of information bits with a low-density parity-check code, wherein the codeword vector is composed of systematic bits and parity bits (¶ 25: “During operation of transmitter 100, encoder 101 outputs a systematic bit stream and at least two parity streams. For this particular example, encoder 101 comprises a rate- $\frac{1}{3}$ turbo coder adopted in the 3GPP standardization. It is noted that the following concepts can be adapted to other types of error correction codes, such as turbo codes with other code rates, low-density parity check (LDPC) codes, convolutional codes, etc.”); **and**

performing circular buffer-based rate matching on the generated codeword vector to produce the coded bits for transmission (¶¶ 27–28: “Circular buffer rate matching selects parity bits approximately evenly distributed over the code trellis irrespective of desired code rate of operation if the sub-block interleavers are designed properly.”), **wherein the circular buffer-based rate matching comprises puncturing a first plurality of systematic bits** (¶¶ 78–82: “[T]he circular buffer can be re-configured to include systematic bit puncturing”).

Regarding claim 2, Blankenship discloses: **The method of claim 1, wherein the low-density parity-check code is specified through a parity check matrix** (¶ 25: low-density parity check (LDPC) codes. The Examiner finds the LDPC codes inherently include a parity-check matrix. See *e.g.*, Johnson, S.J., “Iterative Error Correction: Turbo, Low-Density Parity-Check and Repeat-Accumulate Codes”, Cambridge University Press, 2010, pages 37-39. (Year: 2010) (“As their name suggests, LDPC codes are block codes with parity-check matrices”).).

Regarding claim 3, Blankenship discloses: **The method of claim 1, wherein the first plurality of systematic bits that are punctured precedes** (¶ 109: “When both systematic

puncturing and 1st stage rate matching are used, the implementation may be illustrated as follows. First, the 1st stage RM is achieved by deleting one or more columns (preferably from the last column of the virtual circular buffer) from the circular buffer to form a shortened circular buffer. Thus systematic bits can be punctured by defining the first redundancy version to start at a position with an offset relative to the beginning of the shortened circular buffer. In other words, the systematic bit puncturing may be achieved by defining RV0 at a certain non-zero column number in the sub-block interleaver of the systematic sub-block interleaver.”), **in the codeword vector, a second plurality of systematic bits that are not punctured** (Fig. 10: Option 2; ¶ 82: “2. Option 2 in FIG. 10 contains the same bits as the normal circular buffer, but the circular buffer is modified; some of the systematic bits (SA) are taken from the systematic portion of the circular buffer and placed at the end of the buffer. Therefore, in the first transmission, the systematic bits in SA are punctured at many of the code rates. Option 2 may also be achieved by simply placing RV0 at the start at a position with an offset relative to the beginning of the circular buffer. This offset may be determined based on the amount of systematic bit puncturing required. For code rate $R \geq K_{info}/(3 \times L - SA)$, systematic bits in SA are always punctured.”).

Regarding claim 4, Blankenship discloses: **The method of claim 1, wherein a plurality of redundancy versions are defined over the circular buffer such that coded bits for a retransmission are read from the circular buffer according to a corresponding redundancy version defined for the retransmission** (¶¶ 70–77; Fig. 10: Option 2; ¶ 82: “Option 2 may also be achieved by simply placing RV0 at the start at a position with an offset relative to the beginning of the circular buffer.”).

Regarding claim 5, Blankenship discloses: **The method of claim 4, further comprising performing circular buffer-based rate matching on the generated codeword vector to produce coded bits for the retransmission, wherein the circular buffer-based rate**

matching for the retransmission comprises puncturing the first plurality of systematic

bits (§ 93: Easy to start each RV for HARQ retransmissions. For example, the RVs are defined to start at the top (i.e., first row) of the sub-block interleaver columns in the circular buffer of matrix format.).

Regarding claim 7, Blankenship discloses: **The method of claim 1, wherein puncturing the first plurality of systematic bits comprises omitting to write the first plurality of systematic bits into a circular buffer** (§ 109: “When both systematic puncturing and 1st stage rate matching are used, the implementation may be illustrated as follows. First, the 1st stage RM is achieved by deleting one or more columns (preferably from the last column of the virtual circular buffer) from the circular buffer to form a shortened circular buffer. Thus systematic bits can be punctured by defining the first redundancy version to start at a position with an offset relative to the beginning of the shortened circular buffer. In other words, the systematic bit puncturing may be achieved by defining RV0 at a certain non-zero column number in the sub-block interleaver of the systematic sub-block interleaver.” The Examiner finds the omission of writing the deleted one or more columns back into the circular buffer teaches the claimed limitation).

Regarding claim 8, Blankenship discloses: **The method of claim 1, wherein puncturing the first plurality of systematic bits comprises skipping the first plurality of systematic bits when reading the coded bits from a circular buffer** (§ 98: “The systematic bit puncturing can be switched on by setting σ to be a value greater than 0 (modulo $3 \times N_c$) for 3 streams (i.e., mother code rate $\frac{1}{3}$), thus skipping a small fraction of systematic bits when reading out bits for RV₀. In other words, RV₀ (the first redundancy version) is defined to start at the top of σ -th column, where σ is the number of offset columns. This is shown in FIG. 11. For

example, RV_0 may be defined to start from column #2 instead of column #0 by setting $\sigma=2$.

Other values of σ can also be considered.”).

Regarding claim 10, Blankenship discloses: **The method of claim 1, wherein performing circular buffer-based rate matching on the generated codeword vector comprises:**

reading the systematic bits and the parity bits out of a circular buffer to produce the transmission vector for transmission over a wireless channel (§ 28: For a desired code rate of operation, the number of coded bits N_{desired} to be selected for transmission is calculated and passed to logic circuitry 104 as an input. Logic circuitry 104 simply reads out a length- N_{desired} block of consecutive bits from the sequence the circular buffer (wrapped around to the beginning if last bit of the circular buffer is exceeded) from a certain starting point. Therefore, puncturing and repetition can be achieved using a single method. The circular-buffering technique has an advantage in flexibility (in code rates achieved) and granularity (in streams sizes). Circular buffer rate matching selects parity bits approximately evenly distributed over the code trellis irrespective of desired code rate of operation if the sub-block interleavers are designed properly.

¶ 32: This process of forming the circular buffer is illustrated in FIG. 4 and FIG. 5 with 4 dummy bits inserted in each stream . . . The two matrices S and P0-1 are combined as shown in FIG. 5 and stored in buffer 307. Buffer 307 is read out column-wise, starting at the top of the matrix. Although in the description above the dummy bits are inserted after a stream, in certain embodiments, they can also be inserted before the stream.).

Regarding claim 11, Blankenship discloses: **The method of claim 10, wherein reading the systematic bits and the parity bits out of the circular buffer to produce the transmission vector for transmission over the wireless channel comprises:**

if a number of bits of the transmission vector is larger than the total number of bits in the circular buffer, repeating one or more bits in the circular buffer by wrapping around (§ 33: Logic circuitry 305 simply reads out a length- N_{desired} block of consecutive bits from the output buffer 307 (wrapped around to the beginning if last bit of the output buffer is exceeded) from a certain starting point.).

Regarding claim 12, Blankenship discloses: **The method of claim 1, wherein generating the codeword vector by encoding the set of information bits comprises: attaching dummy bits to the set of information bits, the dummy bits comprising bits of known value, to produce an information vector; and**

encoding the information vector (§ 118: The logic flow begins as step 1201 where logic circuitry 305 receives systematic bits, a first block of parity bits, and a second block of parity bits. These bits are output from encoder 301. As discussed above, the systematic bits, the first block of parity bits, and the second block of parity bits all comprise dummy bits.).

Regarding claim 13, Blankenship discloses: **The method of claim 12, wherein the attached dummy bits comprise a second plurality of systematic bits that is separate from the punctured first plurality of systematic bits** (Fig. 10: Option 2; § 82: “2. Option 2 in FIG. 10 contains the same bits as the normal circular buffer, but the circular buffer is modified; some of the systematic bits (S_A) are taken from the systematic portion of the circular buffer and placed at the end of the buffer. Therefore, in the first transmission, the systematic bits in S_A are punctured at many of the code rates. Option 2 may also be achieved by simply placing RV_0 at the start at a position with an offset relative to the beginning of the circular buffer. This offset may be determined based on the amount of systematic bit puncturing required. For code rate $R \geq K_{\text{info}}/(3 \times L - S_A)$, systematic bits in S_A are always punctured.” The Examiner finds S_B systematic bits of Blankenship teaches the claimed “second plurality of systematic bits”).

Regarding claim 14, Blankenship discloses: **The method of claim 13, comprising puncturing the attached dummy bits** (§ 32: This process of forming the circular buffer is illustrated in FIG. 4 and FIG. 5 with 4 dummy bits inserted in each stream. With reference to FIG. 4, systematic bits 401, having dummy bits 407 added, are block interleaved to produce permuted systematic bits S 402. In a similar manner parity bits P0' and P1' 403 (having dummy bits added) are interleaved to produce parity bits P0 and P1 404. The dummy bits are appended to the streams so that the block interleaver is full for each stream. P0 and P1 are interlaced to produce interlaced portion interlaced P0 and P1 portion 405. The two matrices S and P0-1 are combined as shown in FIG. 5 and stored in buffer 307. Buffer 307 is read out column-wise, starting at the top of the matrix. Although in the description above the dummy bits are inserted after a stream, in certain embodiments, they can also be inserted before the stream.

Fig. 10: Option 2; § 82: "2. Option 2 in FIG. 10 contains the same bits as the normal circular buffer, but the circular buffer is modified; some of the systematic bits (S_A) are taken from the systematic portion of the circular buffer and placed at the end of the buffer. Therefore, in the first transmission, the systematic bits in S_A are punctured at many of the code rates. Option 2 may also be achieved by simply placing RV_0 at the start at a position with an offset relative to the beginning of the circular buffer. This offset may be determined based on the amount of systematic bit puncturing required. For code rate $R \geq K_{info}/(3 \times L - S_A)$, systematic bits in S_A are always punctured."

The Examiner finds the systematic bits S_A and S_B have dummy bits and thus, the systematic bits S_A and S_B teach the claim limitation.).

Regarding claim 15, Blankenship discloses: **The method of claim 14, wherein puncturing the attached dummy bits comprises omitting to write the attached dummy bits into the circular buffer** (§ 35: "If the dummy bits are removed, it is called a dummy-

removed circular buffer.” See *also* ¶ 44. The Examiner finds the omission of writing the deleted dummy bits back into the circular buffer teaches the claimed limitation).

Regarding claim 16, Blankenship discloses: **The method of claim 14, wherein puncturing the attached dummy bits comprises skipping the attached dummy bits when reading the coded bits from the circular buffer** (¶ 35: “If the dummy bits are removed, it is called a dummy-removed circular buffer.” See *also* ¶¶ 28, 33 and 44. The Examiner finds reading the dummy-removed circular buffer teaches the claimed limitation).

Regarding claim 17, Blankenship discloses: **The method of claim 1, wherein a subset of the systematic bits are omitted from being written into the circular buffer** (¶ 109: “When both systematic puncturing and 1st stage rate matching are used, the implementation may be illustrated as follows. First, the 1st stage RM is achieved by deleting one or more columns (preferably from the last column of the virtual circular buffer) from the circular buffer to form a shortened circular buffer. Thus systematic bits can be punctured by defining the first redundancy version to start at a position with an offset relative to the beginning of the shortened circular buffer. In other words, the systematic bit puncturing may be achieved by defining RV0 at a certain non-zero column number in the sub-block interleaver of the systematic sub-block interleaver.” The Examiner finds the omission of writing the deleted one or more columns back into the circular buffer teaches the claimed limitation).

Regarding claim 18, Blankenship discloses: **The method of claim 1, comprising: puncturing a subset of the systematic bits written into the circular buffer in a first transmission over the wireless channel; and including the subset of the systematic bits that were punctured in a retransmission over the wireless channel** (¶ 93: “Combining (a), the RV definition based on a

dummy-padded circular buffer and (b), systematic puncturing, the improved rate matching design has the following advantages, using $N_c=32$ and mother code rate $\frac{1}{3}$ (i.e., three streams out of turbo encoder 101) for illustration: Dummy bits included in the circular buffer to ease RV definition (i.e., dummy-padded circular buffer). Easy to start each RV for HARQ retransmissions.”).

Regarding claim 21, Blankenship discloses: **The method of claim 1, wherein the systematic bits and the parity bits are read out so that a channel interleaver is implemented as part of performing circular buffer-based rate matching on the generated codeword vector** (¶¶ 27–28: “Circular buffer rate matching selects parity bits approximately evenly distributed over the code trellis irrespective of desired code rate of operation if the sub-block interleavers are designed properly . . . Logic circuitry 104 simply reads out a length- $N_{desired}$ block of consecutive bits from the sequence the circular buffer (wrapped around to the beginning if last bit of the circular buffer is exceeded) from a certain starting point.”).

Regarding claim 23, Blankenship discloses: **The method of claim 1, wherein the systematic bits and the parity bits are read out such that no channel interleaving effect is implemented** (¶¶ 27–28: “Circular buffer rate matching selects parity bits approximately evenly distributed over the code trellis irrespective of desired code rate of operation if the sub-block interleavers are designed properly . . . Logic circuitry 104 simply reads out a length- $N_{desired}$ block of consecutive bits from the sequence the circular buffer (wrapped around to the beginning if last bit of the circular buffer is exceeded) from a certain starting point.” The Examiner finds Blankenship does not disclose a channel interleaving effect as a result of the read operation.).

Regarding claim 24, Blankenship discloses: **The method of claim 1, wherein a subset of the parity bits are omitted from being written into the circular buffer** (¶ 109: “When both

systematic puncturing and 1st stage rate matching are used, the implementation may be illustrated as follows. First, the 1st stage RM is achieved by deleting one or more columns (preferably from the last column of the virtual circular buffer) from the circular buffer to form a shortened circular buffer. Thus systematic bits can be punctured by defining the first redundancy version to start at a position with an offset relative to the beginning of the shortened circular buffer. In other words, the systematic bit puncturing may be achieved by defining RV_0 at a certain non-zero column number in the sub-block interleaver of the systematic sub-block interleaver.” The Examiner finds the omission of writing the deleted one or more columns back into the circular buffer teaches the claimed limitation).

Regarding claim 25, Blankenship discloses: **The method of claim 1, wherein performing circular buffer-based rate matching on the generated codeword vector comprises:**

writing at least a portion of the systematic bits and the parity bits as defined by a parity check matrix into a rectangular circular buffer, the rectangular circular buffer comprising one or more rows and one or more columns (Figs. 4–5; ¶¶ 44, 57).

Regarding claim 26, Blankenship discloses: **The method of claim 1, wherein the wireless communications system comprises a new radio system** (¶¶ 8, 20: “In order to address the above-mentioned need, a method and apparatus for assigning redundancy versions to a circular buffer within a communication system is provided herein.”).

Regarding claim 27, Blankenship discloses: **A first node for producing a set of coded bits from a set of information bits for transmission to a second node in a wireless communications system** (Figs. 1 or 3: First node (101 and 102/301 and 303) used for

transmission. ¶¶ 67, 71: Discloses a receiver. The Examiner finds the receiver disclosed in Blankenship) teaches the claimed “second node”), **the first node comprising:**

processing circuitry (¶ 29: Logic circuitry 305 is preferably a microprocessor, microcontroller, digital signal processors (DSPs), or such other devices known to those having ordinary skill in the art.), **the processing circuitry configured to:**

generate a codeword vector by encoding the set of information bits with a low-density parity-check code, wherein the codeword vector is composed of systematic bits and parity bits (¶ 25: “During operation of transmitter 100, encoder 101 outputs a systematic bit stream and at least two parity streams. For this particular example, encoder 101 comprises a rate- $\frac{1}{3}$ turbo coder adopted in the 3GPP standardization. It is noted that the following concepts can be adapted to other types of error correction codes, such as turbo codes with other code rates, low-density parity check (LDPC) codes, convolutional codes, etc.”); **and**

perform circular buffer-based rate matching on the generated codeword vector to produce the coded bits for transmission (¶¶ 27–28: “Circular buffer rate matching selects parity bits approximately evenly distributed over the code trellis irrespective of desired code rate of operation if the sub-block interleavers are designed properly.”), **wherein the circular buffer-based rate matching comprises puncturing a first plurality of systematic bits** (¶¶ 78–82: “[T]he circular buffer can be re-configured to include systematic bit puncturing”).

Regarding claim 28, Blankenship discloses: **The first node of claim 27, wherein the low-density parity-check code is specified through a parity check matrix** (¶ 25: low-density parity check (LDPC) codes. The Examiner finds the LDPC codes inherently include a parity-check matrix. See *e.g.*, Johnson, S.J., “Iterative Error Correction: Turbo, Low-Density Parity-Check and Repeat-Accumulate Codes”, Cambridge University Press, 2010, pages 37-39. (Year: 2010) (“As their name suggests, LDPC codes are block codes with parity-check matrices”).).

Regarding claim 29, Blankenship discloses: **The first node of claim 27, wherein the first plurality of systematic bits that are punctured precedes,** (§ 109: “When both systematic puncturing and 1st stage rate matching are used, the implementation may be illustrated as follows. First, the 1st stage RM is achieved by deleting one or more columns (preferably from the last column of the virtual circular buffer) from the circular buffer to form a shortened circular buffer. Thus systematic bits can be punctured by defining the first redundancy version to start at a position with an offset relative to the beginning of the shortened circular buffer. In other words, the systematic bit puncturing may be achieved by defining RV0 at a certain non-zero column number in the sub-block interleaver of the systematic sub-block interleaver.”), **in the codeword vector, a second plurality of systematic bits that are not punctured** (Fig. 10: Option 2; § 82: “2. Option 2 in FIG. 10 contains the same bits as the normal circular buffer, but the circular buffer is modified; some of the systematic bits (SA) are taken from the systematic portion of the circular buffer and placed at the end of the buffer. Therefore, in the first transmission, the systematic bits in SA are punctured at many of the code rates. Option 2 may also be achieved by simply placing RV0 at the start at a position with an offset relative to the beginning of the circular buffer. This offset may be determined based on the amount of systematic bit puncturing required. For code rate $R \geq K_{info}/(3 \times L - SA)$, systematic bits in SA are always punctured.”).

Regarding claim 30, Blankenship discloses: **The first node of claim 27, wherein a plurality of redundancy versions are defined over the circular buffer such that coded bits for a retransmission are read from the circular buffer according to a corresponding redundancy version defined for the retransmission** (§§ 70–77; Fig. 10: Option 2; § 82: “Option 2 may also be achieved by simply placing RV0 at the start at a position with an offset relative to the beginning of the circular buffer.”).

Regarding claim 31, Blankenship discloses: **The first node of claim 30, wherein the processing circuitry is further configured to perform circular buffer-based rate matching on the generated codeword vector to produce coded bits for the retransmission, wherein the circular buffer-based rate matching for the retransmission comprises puncturing the first plurality of systematic bits** (§ 93: Easy to start each RV for HARQ retransmissions. For example, the RVs are defined to start at the top (i.e., first row) of the sub-block interleaver columns in the circular buffer of matrix format.).

Regarding claim 33, Blankenship discloses: **The first node of claim 27, wherein the processing circuitry configured to puncture the first plurality of systematic bits comprises processing circuitry configured to omit writing the first plurality of systematic bits into a circular buffer** (§ 109: “When both systematic puncturing and 1st stage rate matching are used, the implementation may be illustrated as follows. First, the 1st stage RM is achieved by deleting one or more columns (preferably from the last column of the virtual circular buffer) from the circular buffer to form a shortened circular buffer. Thus systematic bits can be punctured by defining the first redundancy version to start at a position with an offset relative to the beginning of the shortened circular buffer. In other words, the systematic bit puncturing may be achieved by defining RV0 at a certain non-zero column number in the sub-block interleaver of the systematic sub-block interleaver.” The Examiner finds the omission of writing the deleted one or more columns back into the circular buffer teaches the claimed limitation).

Regarding claim 34, Blankenship discloses: **The first node of claim 27, wherein the processing circuitry configured to puncture the first plurality of systematic bits comprises processing circuitry configured to skip the first plurality of systematic bits when reading the coded bits from a circular buffer** (§ 98: “The systematic bit puncturing can be switched on by setting σ to be a value greater than 0 (modulo $3 \times N_c$) for 3 streams (i.e.,

mother code rate $\frac{1}{3}$), thus skipping a small fraction of systematic bits when reading out bits for RV_0 . In other words, RV_0 (the first redundancy version) is defined to start at the top of σ -th column, where σ is the number of offset columns. This is shown in FIG. 11. For example, RV_0 may be defined to start from column #2 instead of column #0 by setting $\sigma=2$. Other values of σ can also be considered.”).

Regarding claim 36, Blankenship discloses: **The first node of claim 27, wherein the processing circuitry configured to perform circular buffer-based rate matching on the generated codeword vector comprises processing circuitry configured to:**

read the systematic bits and the parity bits out of a circular buffer to produce the transmission vector for transmission over a wireless channel (§ 28: For a desired code rate of operation, the number of coded bits N_{desired} to be selected for transmission is calculated and passed to logic circuitry 104 as an input. Logic circuitry 104 simply reads out a length- N_{desired} block of consecutive bits from the sequence the circular buffer (wrapped around to the beginning if last bit of the circular buffer is exceeded) from a certain starting point. Therefore, puncturing and repetition can be achieved using a single method. The circular-buffering technique has an advantage in flexibility (in code rates achieved) and granularity (in streams sizes). Circular buffer rate matching selects parity bits approximately evenly distributed over the code trellis irrespective of desired code rate of operation if the sub-block interleavers are designed properly.

¶ 32: This process of forming the circular buffer is illustrated in FIG. 4 and FIG. 5 with 4 dummy bits inserted in each stream . . . The two matrices S and P0-1 are combined as shown in FIG. 5 and stored in buffer 307. Buffer 307 is read out column-wise, starting at the top of the matrix. Although in the description above the dummy bits are inserted after a stream, in certain embodiments, they can also be inserted before the stream.).

Regarding claim 37, Blankenship discloses: **The first node of claim 36, wherein the processing circuitry configured to read the systematic bits and the parity bits out of the circular buffer to produce the transmission vector for transmission over the wireless channel comprises processing circuitry configured to:**

if a number of bits of the transmission vector is larger than the total number of bits in the circular buffer, repeat one or more bits in the circular buffer by wrapping around (§ 33: Logic circuitry 305 simply reads out a length- $N_{desired}$ block of consecutive bits from the output buffer 307 (wrapped around to the beginning if last bit of the output buffer is exceeded) from a certain starting point.).

Regarding claim 38, Blankenship discloses: **The first node of claim 27, wherein the processing circuitry configured to generate the codeword vector by encoding the set of information bits comprises processing circuitry configured to:**

attach dummy bits to the set of information bits, the dummy bits comprising bits of known value, to produce an information vector; and

encode the information vector (§ 118: The logic flow begins as step 1201 where logic circuitry 305 receives systematic bits, a first block of parity bits, and a second block of parity bits. These bits are output from encoder 301. As discussed above, the systematic bits, the first block of parity bits, and the second block of parity bits all comprise dummy bits.).

Regarding claim 39, Blankenship discloses: **The first node of claim 38, wherein the attached dummy bits comprise a second plurality of systematic bits that is separate from the punctured first plurality of systematic bits** (Fig. 10: Option 2; § 82: “2. Option 2 in FIG. 10 contains the same bits as the normal circular buffer, but the circular buffer is modified; some of the systematic bits (S_A) are taken from the systematic portion of the circular buffer and placed at the end of the buffer. Therefore, in the first transmission, the systematic bits in S_A are

punctured at many of the code rates. Option 2 may also be achieved by simply placing RV_0 at the start at a position with an offset relative to the beginning of the circular buffer. This offset may be determined based on the amount of systematic bit puncturing required. For code rate $R \geq K_{info}/(3 \times L - S_A)$, systematic bits in S_A are always punctured.” The Examiner finds S_B systematic bits of Blankenship teaches the claimed “second plurality of systematic bits”).

Regarding claim 40, Blankenship discloses: **The first node of claim 39, wherein the processing circuitry is further configured to puncture the attached dummy bits** (§ 32: This process of forming the circular buffer is illustrated in FIG. 4 and FIG. 5 with 4 dummy bits inserted in each stream. With reference to FIG. 4, systematic bits 401, having dummy bits 407 added, are block interleaved to produce permuted systematic bits S 402. In a similar manner parity bits P0' and P1' 403 (having dummy bits added) are interleaved to produce parity bits P0 and P1 404. The dummy bits are appended to the streams so that the block interleaver is full for each stream. P0 and P1 are interlaced to produce interlaced portion interlaced P0 and P1 portion 405. The two matrices S and P0-1 are combined as shown in FIG. 5 and stored in buffer 307. Buffer 307 is read out column-wise, starting at the top of the matrix. Although in the description above the dummy bits are inserted after a stream, in certain embodiments, they can also be inserted before the stream.

Fig. 10: Option 2; § 82: “2. Option 2 in FIG. 10 contains the same bits as the normal circular buffer, but the circular buffer is modified; some of the systematic bits (S_A) are taken from the systematic portion of the circular buffer and placed at the end of the buffer. Therefore, in the first transmission, the systematic bits in S_A are punctured at many of the code rates. Option 2 may also be achieved by simply placing RV_0 at the start at a position with an offset relative to the beginning of the circular buffer. This offset may be determined based on the amount of systematic bit puncturing required. For code rate $R \geq K_{info}/(3 \times L - S_A)$, systematic bits in S_A are always punctured.”

The Examiner finds the systematic bits S_A and S_B have dummy bits and thus, the systematic bits S_A and S_B teach the claim limitation.).

Regarding claim 41, Blankenship discloses: **The first node of claim 40, wherein the processing circuitry configured to puncture the attached dummy bits comprises processing circuitry configured to omit writing the attached dummy bits into the circular buffer** (§ 35: “If the dummy bits are removed, it is called a dummy-removed circular buffer.” See *also* § 44. The Examiner finds the omission of writing the deleted dummy bits back into the circular buffer teaches the claimed limitation).

Regarding claim 42, Blankenship discloses: **The first node of claim 40, wherein the processing circuitry configured to puncture the attached dummy bits comprises processing circuitry configured to skip the attached dummy bits when reading the coded bits from the circular buffer** (§ 35: “If the dummy bits are removed, it is called a dummy-removed circular buffer.” See *also* §§ 28, 33 and 44. The Examiner finds reading the dummy-removed circular buffer teaches the claimed limitation).

Regarding claim 43, Blankenship discloses: **The first node of claim 27, wherein a subset of the systematic bits are omitted from being written into the circular buffer** (§ 109: “When both systematic puncturing and 1st stage rate matching are used, the implementation may be illustrated as follows. First, the 1st stage RM is achieved by deleting one or more columns (preferably from the last column of the virtual circular buffer) from the circular buffer to form a shortened circular buffer. Thus systematic bits can be punctured by defining the first redundancy version to start at a position with an offset relative to the beginning of the shortened circular buffer. In other words, the systematic bit puncturing may be achieved by defining RV0 at a certain non-zero column number in the sub-block interleaver of the systematic

sub-block interleaver.” The Examiner finds the omission of writing the deleted one or more columns back into the circular buffer teaches the claimed limitation).

Regarding claim 44, Blankenship discloses: **The first node of claim 27, wherein the processing circuitry is further configured to:**

puncture a subset of the systematic bits written into the circular buffer in a first transmission over the wireless channel; and

include the subset of the systematic bits that were punctured in a retransmission over the wireless channel (§ 93: “Combining (a), the RV definition based on a dummy-padded circular buffer and (b), systematic puncturing, the improved rate matching design has the following advantages, using $N_c=32$ and mother code rate $\frac{1}{3}$ (i.e., three streams out of turbo encoder 101) for illustration: Dummy bits included in the circular buffer to ease RV definition (i.e., dummy-padded circular buffer). Easy to start each RV for HARQ retransmissions.”).

Regarding claim 47, Blankenship discloses: **The first node of claim 27, wherein the processing circuitry is configured to read out the systematic bits and the parity bits so that a channel interleaver is implemented as part of performing circular buffer-based rate matching on the generated codeword vector** (§§ 27–28: “Circular buffer rate matching selects parity bits approximately evenly distributed over the code trellis irrespective of desired code rate of operation if the sub-block interleavers are designed properly . . . Logic circuitry 104 simply reads out a length- N_{desired} block of consecutive bits from the sequence the circular buffer (wrapped around to the beginning if last bit of the circular buffer is exceeded) from a certain starting point.”).

Regarding claim 49, Blankenship discloses: **The first node of claim 27, wherein the processing circuitry is configured to read out the systematic bits and the parity bits such**

that no channel interleaving effect is implemented (§§ 27–28: “Circular buffer rate matching selects parity bits approximately evenly distributed over the code trellis irrespective of desired code rate of operation if the sub-block interleavers are designed properly . . . Logic circuitry 104 simply reads out a length- N_{desired} block of consecutive bits from the sequence the circular buffer (wrapped around to the beginning if last bit of the circular buffer is exceeded) from a certain starting point.” The Examiner finds Blankenship does not disclose a channel interleaving effect as a result of the read operation.).

Regarding claim 50, Blankenship discloses: **The first node of claim 27, wherein a subset of the parity bits are omitted from being written into the circular buffer** (§ 109: “When both systematic puncturing and 1st stage rate matching are used, the implementation may be illustrated as follows. First, the 1st stage RM is achieved by deleting one or more columns (preferably from the last column of the virtual circular buffer) from the circular buffer to form a shortened circular buffer. Thus systematic bits can be punctured by defining the first redundancy version to start at a position with an offset relative to the beginning of the shortened circular buffer. In other words, the systematic bit puncturing may be achieved by defining RV_0 at a certain non-zero column number in the sub-block interleaver of the systematic sub-block interleaver.” The Examiner finds the omission of writing the deleted one or more columns back into the circular buffer teaches the claimed limitation).

Regarding claim 51, Blankenship discloses: **The first node of claim 27, wherein the processing circuitry configured to perform circular buffer-based rate matching on the generated codeword vector comprises processing circuitry configured to:**

write at least a portion of the systematic bits and the parity bits as defined by a parity check matrix into a rectangular circular buffer, the rectangular circular buffer comprising one or more rows and one or more columns (Figs. 4–5; §§ 44, 57).

Regarding claim 52, Blankenship discloses: **The first node of claim 27, wherein the wireless communications system comprises a new radio system** (¶¶ 8, 20: “In order to address the above-mentioned need, a method and apparatus for assigning redundancy versions to a circular buffer within a communication system is provided herein.”).

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103 which forms the basis for all obviousness rejections set forth in this Office action:

A patent for a claimed invention may not be obtained, notwithstanding that the claimed invention is not identically disclosed as set forth in section 102, if the differences between the claimed invention and the prior art are such that the claimed invention as a whole would have been obvious before the effective filing date of the claimed invention to a person having ordinary skill in the art to which the claimed invention pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 6, 9, 32 and 35 are rejected under 35 U.S.C. 103 as being unpatentable over Blankenship et al. (U.S. Patent Publication No. 2008/0320353 A1), hereinafter referred to as Blankenship, in view of Lomayev et al. (U.S. Patent Publication No. 2018/0019840 A1), hereinafter referred to as Lomayev.

Regarding claim 6, Blankenship discloses the limitations of claim 5, but does not teach: **The method of claim 5, wherein the first plurality of systematic bits punctured for the retransmission includes an even multiple of Z systematic bits, where Z is a lifting factor of the parity check matrix.**

Lomayev in an analogous art teaches: **The method of claim 5, wherein the first plurality of systematic bits punctured for the retransmission includes an even multiple of Z systematic bits, where Z is a lifting factor of the parity check matrix** (¶ 154: “In some demonstrative embodiments, encoder 129 may be configured, for example, to encode 1092

systematic data bits, for example, by applying an LDPC lifting matrix, e.g., the LDPC lifting matrix discussed above, to generate an LDPC of a length of 1344 bits.”).

The Examiner finds Blankenship discloses encoding systematic bits using LDPC codes, and Lomayev teaches encoding systematic data bits using an LDPC lifting matrix. The Examiner further finds it would have been obvious to one having ordinary skill in the art when the application was filed to substitute the LDPC code in Blankenship with the LDPC code of Lomayev to achieve the predictable result of applying an LDPC lifting matrix to systematic data bits to generate an encoded LDPC.

Regarding claim 9, Blankenship discloses the limitations of claim 1, but does not teach:
The method of claim 1, wherein the first plurality of punctured systematic bits includes an even multiple of Z systematic bits, where Z is a lifting factor of the parity check matrix.

Lomayev in an analogous art teaches: **The method of claim 1, wherein the first plurality of punctured systematic bits includes an even multiple of Z systematic bits, where Z is a lifting factor of the parity check matrix** (§ 154: “In some demonstrative embodiments, encoder 129 may be configured, for example, to encode 1092 systematic data bits, for example, by applying an LDPC lifting matrix, e.g., the LDPC lifting matrix discussed above, to generate an LDPC of a length of 1344 bits.”).

The Examiner finds Blankenship discloses encoding systematic bits using LDPC codes, and Lomayev teaches encoding systematic data bits using an LDPC lifting matrix that includes an even number of systematic bits. The Examiner further finds it would have been obvious to one having ordinary skill in the art when the application was filed to substitute the LDPC code in Blankenship with the LDPC code of Lomayev to achieve the predictable result of applying a LDPC lifting matrix to an even number of systematic data bits to generate an encoded LDPC.

Regarding claim 32, Blankenship discloses the limitations of claim 31, but does not teach: **The first node of claim 31, wherein the first plurality of systematic bits punctured for the retransmission includes an even multiple of Z systematic bits, where Z is a lifting factor of the parity check matrix.**

Lomayev in an analogous art teaches: **The first node of claim 31, wherein the first plurality of systematic bits punctured for the retransmission includes an even multiple of Z systematic bits, where Z is a lifting factor of the parity check matrix** (§ 154: "In some demonstrative embodiments, encoder 129 may be configured, for example, to encode 1092 systematic data bits, for example, by applying an LDPC lifting matrix, e.g., the LDPC lifting matrix discussed above, to generate an LDPC of a length of 1344 bits.").

The Examiner finds Blankenship discloses encoding systematic bits using LDPC codes, and Lomayev teaches encoding systematic data bits using an LDPC lifting matrix. The Examiner further finds it would have been obvious to one having ordinary skill in the art when the application was filed to substitute the LDPC code in Blankenship with the LDPC code of Lomayev to achieve the predictable result of applying an LDPC lifting matrix to systematic data bits to generate an encoded LDPC.

Regarding claim 35, Blankenship discloses the limitations of claim 27, but does not teach: **The first node of claim 27, wherein the first plurality of punctured systematic bits includes an even multiple of Z systematic bits, where Z is a lifting factor of the parity check matrix.**

Lomayev in an analogous art teaches: **The first node of claim 27, wherein the first plurality of punctured systematic bits includes an even multiple of Z systematic bits, where Z is a lifting factor of the parity check matrix** (§ 154: "In some demonstrative embodiments, encoder 129 may be configured, for example, to encode 1092 systematic data

bits, for example, by applying an LDPC lifting matrix, e.g., the LDPC lifting matrix discussed above, to generate an LDPC of a length of 1344 bits.”).

The Examiner finds Blankenship discloses encoding systematic bits using LDPC codes, and Lomayev teaches encoding systematic data bits using an LDPC lifting matrix that includes an even number of systematic bits. The Examiner further finds it would have been obvious to one having ordinary skill in the art when the application was filed to substitute the LDPC code in Blankenship with the LDPC code of Lomayev to achieve the predictable result of applying a LDPC lifting matrix to an even number of systematic data bits to generate an encoded LDPC.

Allowable Subject Matter

7. Claims 19-20, 22, 45-46 and 48 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Prior Art

The prior art of record, considered pertinent to the applicant's disclosure, is listed in the attached PTO-892 form.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to KYLE VALLECILLO whose telephone number is (571)272-7716. The examiner can normally be reached on 8:30 A.M. - 4:30 P.M..

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, ALBERT DECADY can be reached on (571)272-3819. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/KYLE VALLECILLO/
Primary Examiner, Art Unit 2112