

(51) International Patent Classification:
H04B 7/08 (2006.01)(21) International Application Number:
PCT/US2016/066192(22) International Filing Date:
12 December 2016 (12.12.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/386,762 10 December 2015 (10.12.2015) US
62/316,336 31 March 2016 (31.03.2016) US(71) Applicant: **UNIVERSITY OF UTAH RESEARCH FOUNDATION** [US/US]; 615 Arapeen Drive, Suite 310, Salt Lake City, UT 84108 (US).(72) Inventors: **HEDSTROM, Jonathan**; 615 Arapeen Drive, Suite 310, Salt Lake City, UT 84108 (US). **YUEN, Chung Him**; 615 Arapeen Drive, Suite 310, Salt Lake City, UT 84108 (US). **FARHANG-BOROUJENY, Behrouz**; 615 Arapeen Drive, Suite 310, Salt Lake City, UT 84108 (US).(74) Agents: **ERICKSEN, Erik, S.** et al.; Thorpe North & Western, LLP, 8180 S. 700 E., Suite 350, Sandy, UT 84070 (US).

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

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, 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, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

(54) Title: MARKOV CHAIN MONTE CARLO MIMO DETECTOR METHOD WITH GIBBS SAMPLER EXCITATION

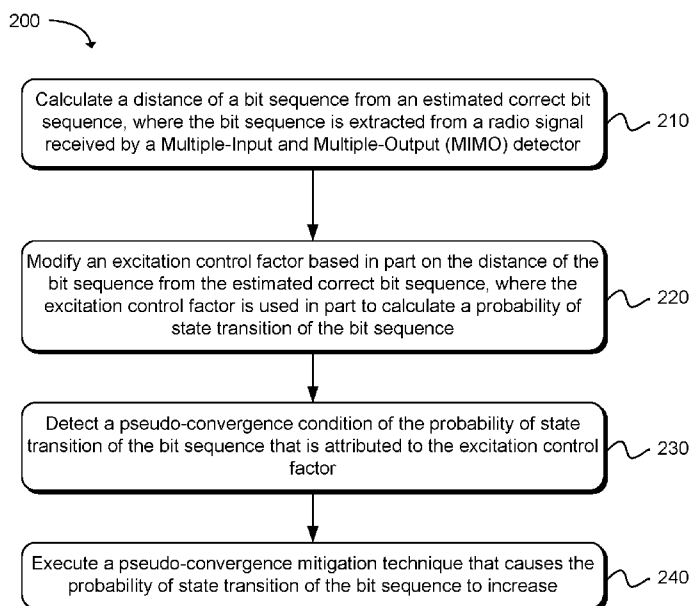


FIG. 2

(57) Abstract: A technology is disclosed and described for processing a radio signal using an excited Markov Chain Monte Carlo (MCMC) Gibbs sampler. An example method (200) may include calculating a distance (210) of a bit sequence from an estimated correct bit sequence, where the bit sequence may be extracted from a received radio signal received by a Multiple-Input and Multiple-Output (MIMO) detector. An excitation factor may be modified (220) based in part on the distance of the bit sequence from the estimated correct bit sequence, where the excitation factor may be used in part to calculate a probability of state transition of the bit sequence. A pseudo-convergence of the probability of state transition of the bit sequence may be detected (230) and a pseudo-convergence mitigation technique may be executed (240) that causes the probability of state transition of the bit sequence to increase.

MARKOV CHAIN MONTE CARLO MIMO DETECTOR METHOD WITH GIBBS SAMPLER EXCITATION

RELATED APPLICATION(S)

This application is related to U.S. Provisional Application No. 62/386,762 filed on December 10, 2015, and U.S. Provisional Application No. 62/316,336 filed on March 31, 2016, which are incorporated herein by reference.

5

GOVERNMENT INTEREST

This invention was made with government support under Grant Number 1449033 awarded by the National Science Foundation. The government has certain rights in the invention.

10

BACKGROUND

Wireless communication systems such as WiFi and LTE have a need to increase speed and capacity. Multiple-input-multiple-output (MIMO) and CDMA techniques can be used, but they may need high performance and cost effective ways to separate the independent streams, which have been mixed. One separation method is called the Markov Chain Monte Carlo (MCMC) detector. However, MCMC is known to have poor performance at high signal to noise ratio (SNR). Thus, improved MCMC methods are needed to improve bit error rate (BER) performance at high SNR.

15

SUMMARY

20

A technology is described for a Markov Chain Monte Carlo (MCMC) Multiple-Input Multiple-Output (MIMO) detector with Gibbs sampler excitation (X-MCMC). The technology can increase spectral efficiency of wireless communications using an excited MCMC Gibbs sampling technique configured to dynamically regulate excitation of the Gibbs sampler. In one example, the excited MCMC Gibbs sampling technique can be used to detect a stalling condition associated with extracting a bit sequence from a radio signal received at a MIMO detector and calculating a distance of the bit sequence from an estimated correct bit sequence. An excitation factor, used in part to calculate a probability

25

of state transition of the bit sequence can be modified, based in part, on a distance of the bit sequence from the estimated correct bit sequence. In doing so, a stalling condition of the probability of state transition of the bit sequence may be mitigated. In the event that a further pseudo-convergence condition is detected, a pseudo-convergence mitigation technique can be performed. The pseudo-convergence mitigation technique can cause the probability of state transition of the bit sequence to increase.

There has thus been outlined, rather broadly, the more important features of the invention so that the detailed description thereof that follows may be better understood, and so that the present contribution to the art may be better appreciated. Other features of the present invention will become clearer from the following detailed description of the invention, taken with the accompanying drawings and claims, or may be learned by the practice of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating an example system for processing a radio signal using an excited MCMC Gibbs sampler.

FIG. 2 is a flow diagram that illustrates an example method for an excited Markov Chain Monte Carlo (MCMC) Gibbs sampler.

FIG. 3 is block diagram illustrating an example of a computing device that may be used to execute a method for an excited Markov Chain Monte Carlo (MCMC) Gibbs sampler.

FIG. 4 illustrates a diagram of a wireless device (e.g., UE) and a base station (e.g., eNodeB) in accordance with an example.

FIG. 5 illustrates a transition diagram between possible states.

These drawings are provided to illustrate various aspects of the invention and are not intended to be limiting of the scope in terms of dimensions, materials, configurations, arrangements or proportions unless otherwise limited by the claims.

DETAILED DESCRIPTION

While these exemplary embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, it should be understood that other

embodiments may be realized and that various changes to the invention may be made without departing from the spirit and scope of the present invention. Thus, the following more detailed description of the embodiments of the present invention is not intended to limit the scope of the invention, as claimed, but is presented for purposes of illustration only and not limitation to describe the features and characteristics of the present invention, to set forth the best mode of operation of the invention, and to sufficiently enable one skilled in the art to practice the invention. Accordingly, the scope of the present invention is to be defined solely by the appended claims.

Definitions

In describing and claiming the present invention, the following terminology will be used.

The singular forms “a,” “an,” and “the” include plural referents unless the context clearly dictates otherwise. Thus, for example, reference to “a sampler” includes reference to one or more of such materials and reference to “modifying” refers to one or more such steps.

As used herein with respect to an identified property or circumstance, “substantially” refers to a degree of deviation that is sufficiently small so as to not measurably detract from the identified property or circumstance. The exact degree of deviation allowable may in some cases depend on the specific context.

As used herein, “adjacent” refers to the proximity of two structures or elements. Particularly, elements that are identified as being “adjacent” may be either abutting or connected. Such elements may also be near or close to each other without necessarily contacting each other. The exact degree of proximity may in some cases depend on the specific context.

As used herein, a plurality of items, structural elements, compositional elements, and/or materials may be presented in a common list for convenience. However, these lists should be construed as though each member of the list is individually identified as a separate and unique member. Thus, no individual member of such list should be construed as a *de facto* equivalent of any other member of the same list solely based on their presentation in a common group without indications to the contrary.

As used herein, the term “at least one of” is intended to be synonymous with “one or more of.” For example, “at least one of A, B and C” explicitly includes only A, only B, only C, and combinations of each (e.g. A+B, B+C, A+C, and A+B+C).

Concentrations, amounts, and other numerical data may be presented herein in a range format. It is to be understood that such range format is used merely for convenience and brevity and should be interpreted flexibly to include not only the numerical values explicitly recited as the limits of the range, but also to include all the individual numerical values or sub-ranges encompassed within that range as if each numerical value and sub-range is explicitly recited. For example, a numerical range of about 1 to about 4.5 should be interpreted to include not only the explicitly recited limits of 1 to about 4.5, but also to include individual numerals such as 2, 3, 4, and sub-ranges such as 1 to 3, 2 to 4, etc. The same principle applies to ranges reciting only one numerical value, such as “less than about 4.5,” which should be interpreted to include all of the above-recited values and ranges. Further, such an interpretation should apply regardless of the breadth of the range or the characteristic being described.

Any steps recited in any method or process claims may be executed in any order and are not limited to the order presented in the claims. Means-plus-function or step-plus-function limitations will only be employed where for a specific claim limitation all of the following conditions are present in that limitation: a) “means for” or “step for” is expressly recited; and b) a corresponding function is expressly recited. The structure, material or acts that support the means-plus function are expressly recited in the description herein. Accordingly, the scope of the invention should be determined solely by the appended claims and their legal equivalents, rather than by the descriptions and examples given herein.

As used herein, the term “processor” can include general purpose processors, specialized processors such as very-large-scale integration (VLSI), field-programmable gate arrays (FPGAs), or other types of specialized processors, as well as base band processors used in transceivers to send, receive, and process wireless communications. Some of the functional units described herein have been labeled as modules, in order to more particularly emphasize their implementation independence. For example, a module may be implemented as a hardware circuit comprising custom VLSI circuits or gate arrays,

off-the-shelf semiconductors such as logic chips, transistors, or other discrete components. A module may also be implemented in programmable hardware devices such as FPGAs, programmable array logic, programmable logic devices or the like.

As used herein, the term “memory” and “memory device” can include devices which store data including, but not limited to, RAM, ROM, cache, registers, flip-flops, latches, and specialized circuits and devices for storage of data or processor instructions.

As used herein, the term “memory device with instructions” can be devices which can be used to control a processor and can include a specialized circuit, module, or programmable device which is configured to directly execute the process that would be controlled by the memory device with instructions.

Multiple hardware circuits or multiple processors can be used to implement the functional units described in this specification. For example, a first hardware circuit or a first processor can be used to perform processing operations and a second hardware circuit or a second processor (e.g., a transceiver or a baseband processor) can be used to communicate with other entities. The first hardware circuit and the second hardware circuit can be integrated into a single hardware circuit, or alternatively, the first hardware circuit and the second hardware circuit can be separate hardware circuits.

Modules can also be implemented in software for execution by various types of processors. An identified module of executable code can, for instance, comprise one or more physical or logical blocks of computer instructions, which can, for instance, be organized as an object, procedure, or function. Nevertheless, the executables of an identified module need not be physically located together, but can comprise disparate instructions stored in different locations which, when joined logically together, comprise the module and achieve the stated purpose for the module.

The goal of most recent wireless technology advancements is to increase the spectral efficiency of wireless communications, thus allowing more data to be sent over the same amount of frequency spectrum. This is important because it directly affects the maximum capacity of a wireless network. For example, a cellular network with 200 MHz of licensed bandwidth and 1 bit/second/Hz spectral efficiency can serve a total of 200 Mbps (mega-bits per second) to its users. Assuming 100 users, each will be able to use 2 Mbps. Suppose a new technology allowed an increase of spectral efficiency to 4

bit/second/Hz, then the cell carrier could either increase to 400 users or increase each user's cellular speed to 8 Mbps, thereby increasing profits and customer satisfaction.

Developers have been consistently increasing spectral efficiency and capacity of wireless networks for over a hundred years, but it is becoming increasingly difficult to make continued, meaningful progress. One of the last frontiers is to use multiple antennas to simultaneously send data in parallel. This is called spatial-multiplexing MIMO (Multiple-Input Multiple-Output). It is included in the 802.11n WiFi, 802.11ac WiFi, LTE-Advanced, and 5G protocols. A common shorthand when describing MIMO systems is to state the number of receive and transmit antennas in the form $N_{transmit} \times N_{receive}$, so 8x8 means 8 transmit by 8 receive.

A primary reason why MIMO has only recently been used to increase spectral efficiency is because MIMO is computationally difficult. At the receiver the parallel signals are mixed together, requiring a large amount of processing to separate. The digital signal processing (DSP) to separate the mixed signals is called the "detector." Unfortunately, for historical reasons it is also sometimes called a "decoder" which causes confusion with another completely different step in the processing called "decoding" (e.g. LDPC decoder and convolutional decoder). Henceforth the method of separating signals will be called a "detector" and the method of recovering the codewords using the error correction code will be the "decoder," but this is not necessarily true in other documents where the precise meaning must be deciphered by context.

GENERAL MIMO DETECTOR OVERVIEW

MIMO MODEL

A typical single-input single-output (SISO) one stream wireless system can be modeled by the following, where y is the received signal, h is the attenuation and delay of the signal moving from transmitter to receiver through a flat fading channel, s is the original transmitted signal, and n is additive white Gaussian noise (AWGN) with variance σ_n^2 per complex element, Eq. 1.

$$y = |hs| + n \quad (1)$$

The simple one antenna SISO model can be expanded to MIMO by making all variables vectors except for the channel which becomes the matrix $\mathbf{H}$, Eq. 2. The channel matrix represents the attenuation and delay between each pair of transmit and receive antennas.

5

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n} \quad (2)$$

Generally it is assumed that one has a good estimate of the channel matrix $\mathbf{H}$. This is reasonable because real-world wireless protocols specify that a known training sequence of sufficient length be placed in the header. Detectors use this channel information in various ways.

10

The zero-forcing (ZF) and minimum-mean-square-error (MMSE) methods use the straightforward linear algebra to solve for $\hat{\mathbf{s}}$, an estimate of $\mathbf{s}$. They are simple but have low BER performance. These detectors are sometimes used to initialize other more complex detectors to assure starting locations somewhat close to the correct solution.

15

The theoretically best approaches to solving the signal separation problem is maximum likelihood (ML) and maximum a posteriori (MAP). They are brute force searches across all possible transmitted bit sequence permutations which make them too complex to use in practice, but they are useful for comparison purposes.

20

In practice the most common detectors are a large class often referred to as approximate-ML methods or approximate-MAP if using prior information. Rather than searching across all bit sequence permutations as in ML, they limit their search across a small subset. This class includes but is not limited to MCMC, sphere-decoding, k-best, and tree-search.

25

Also, there are many varieties of interference cancellation detectors. They are generally a hybrid method where one of the other detector methods is used to generate an estimate of interference, that estimate is subtracted from the original variables, and then the operation is repeated to improve the results.

MCMC DETECTOR BACKGROUND

The Markov Chain Monte Carlo (MCMC) detector can efficiently generate an accurate estimate $\hat{\mathbf{s}}$ in the MIMO model represented by Eq. 2. Its approach is to do an

intelligent random walk across a subset of important bit sequence permutations and use this subset to output soft-information.

This intelligent walk is designed by considering each permutation of the possible transmitted bit sequence as a state and each 1-bit change as a transition between possible states, as seen in FIG. 5. Using this idea one may derive Eqs. 3 and 5 which provide the probability of a specific state transition where b_k is the current k^{th} bit being examined, σ_n^2 is the noise variance, λ_k^a is the prior information for the k^{th} bit, $\mathbf{b}_{\setminus k}$ is the current state's bit sequence excluding the k^{th} bit, γ_k is the parameter from Eq. 5, $\mathbf{y}$ is the received radio signal, $\mathbf{H}$ is the attenuation and delay of the received radio signal, $\hat{\mathbf{s}}_{k-}$ is the current state with the k^{th} bit forced to a -1 mapped to constellation symbols, and $\hat{\mathbf{s}}_{k+}$ is the current state with the k^{th} bit forced to a one mapped to constellation symbols. Note that the bit sequence vector $\mathbf{b}$ uses -1 to represent a binary zero and +1 to represent a binary one.

$$\Pr(b_k = +1 | \lambda_k^a, \sigma_n^2, \mathbf{y}, \mathbf{H}, \mathbf{b}_{\setminus k}) = \frac{1}{1 + e^{-\gamma_k}} \quad (3)$$

$$\Pr(b_k = +1 | \lambda_k^a, \sigma_n^2, \mathbf{y}, \mathbf{H}, \mathbf{b}_{\setminus k}) \approx \begin{cases} \frac{\gamma_k}{8} + \frac{1}{2}, & -4 < \gamma_k < 4 \\ 1, & \gamma_k \geq 4 \\ 0, & \gamma_k \leq -4 \end{cases} \quad (4)$$

$$\gamma_k = \lambda_k^a + \frac{\|\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k-}\|^2 - \|\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k+}\|^2}{\sigma_n^2} \quad (5)$$

The algorithm that performs the intelligent random walk is the Gibbs sampler. The Gibbs sampler cycles through the bit sequence $\mathbf{b}$ one at a time, deciding if a bit should be changed and updating auxiliary variables. For example, the following describes the operation of one parallel Gibbs sampler where multiple parallel Gibbs samplers can be run at the same time to perform multiple intelligent random walks:

Initialize a bit sequence $\mathbf{b}$ either randomly or with an initial estimate from ZF, MMSE, λ^a , or another method with prior knowledge of the transmitted signal.

For $i = 0$ to $N_{\text{depth}} - 1$

For $k = 0$ to $N_t \log_2(N_{\text{QAM}}) - 1$

Accumulate the best $\eta_k^{+'}$ and $\eta_k^{-'}$.

Calculate the probability of the k^{th} bit transition.

Generate a random number r between 0 and 1 using the uniform distribution. If the random number r is less than the bit transition probability make $b_k = +1$, otherwise make $b_k = -1$.

where N_{depth} is the number of cycles over the transmitted bits $\mathbf{b}$, N_{QAM} is the number of points in the modulation constellation, and N_t is the number of transmit streams. Note that a set of additional variables were updated in the Gibbs sampler algorithm described above including η_k^+ and η_k^- .

Once the Gibbs sampler has completed sufficient iterations, the final output soft-information LLR is calculated as in Eq. 10. This LLR output is generally passed to a forward-error-correcting decoder such as LDPC which is capable of correcting a percentage of errors.

$$\eta_k^+ = \frac{1}{\sigma_n^2} ||\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k+}||^2 - \mathbf{b}_{\setminus k} \cdot \boldsymbol{\lambda}_{\setminus k}^a \quad (6)$$

$$\eta_k^- = \frac{1}{\sigma_n^2} ||\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k-}||^2 - \mathbf{b}_{\setminus k} \cdot \boldsymbol{\lambda}_{\setminus k}^a \quad (7)$$

$$\eta_k^{+'} = \min_{\mathcal{N}_k^+}(\eta_k^+) \quad (8)$$

$$\eta_k^{-'} = \min_{\mathcal{N}_k^-}(\eta_k^-) \quad (9)$$

$$\text{LLR}(k) = \eta_k^{-'} - \eta_k^{+'} \quad (10)$$

where $\mathcal{N}_k^+$ and $\mathcal{N}_k^-$ are the lists of all η_k^+ and η_k^- values computed across all parallel Gibbs sampler iterations. The $\mathbf{b}_{\setminus k} \cdot \boldsymbol{\lambda}_{\setminus k}^a$ term accumulates the prior information using a dot product, where $\mathbf{b}_{\setminus k}$ is the bit sequence vector without the k^{th} element and $\boldsymbol{\lambda}_{\setminus k}^a$ is the vector of prior information without the k^{th} element.

The conventional derivation and implementation of MCMC suffers from a well-known limitation that MCMC performs poorly at high Signal-to-Noise Ratios (SNRs). Past attempts have been made to resolve the high SNR problem. One of the most prominent attempts is to initialize a Gibbs sampler with an MMSE estimate instead of using a random

initialization. Other ad hoc methods include multiplying γ_k in a probability calculation by a constant temperature coefficient or by slowly changing a coefficient which starts small and grows to unity. These methods appear to have a positive effect, but upon closer inspection, the underlying problem still exists. At high SNRs the Gibbs sampler tends to not move or to move between only a few stable states, wasting many samples exploring the same small number of permutations.

Stalling at high SNR can be caused when the probability of a state transition reaches an extreme value. For example, with an input of $\gamma_k = 0$, the bit change probability is 50%, meaning that the Gibbs sampler is active. In practice $|\gamma_k| > 10$ is common at high SNR, whereas already at small values of $|\gamma_k| > 3$, there may be a meager 2% probability of a non-deterministic movement, as illustrated in Eq. 12. Thus the Gibbs sampler may become nearly deterministic and thus slow to be nearly stationary, dramatically lowering performance while increasing required number of samples and thus complexity, cost and power consumption.

$$\Pr(b_k = +1 | \gamma_k = 0) = 50\% \quad (11)$$

$$\Pr(b_k = +1 | \gamma_k < -3) < 2\% \quad (12)$$

FIG. 1 is a block diagram illustrating a high level example of a system on which the present technology may be executed. Illustrated is a MIMO configuration (providing a visual representation of channel matrix $\mathbf{H}$ with connecting arrows representing complex elements of signal strength and phase delay) that is communicatively connected to an apparatus configured to execute an excited MCMC Gibbs sampler.

The excited MCMC Gibbs sampler 130 can solve the high SNR stalling problem, improves performance, and decreases complexity at all SNR levels. The present technology does this by dynamically regulating the excitation, i.e. the probability of transition, of a Gibbs sampler so that the sampler is consistently moving. The excitation amount must be carefully selected to implement this algorithm effectively. If the probabilities are too extreme, then stalling will be encountered, wasting samples and lowering performance. If the probabilities are moderated too much, then there will be too much motion in the Gibbs sampler and the random walk will move past correct solutions, behaving more like a simple random walk rather than an intelligent random walk.

With this background in mind, the present technology can be configured to detect whether a bit sequence being processed by a Gibbs sampler is close or far away from a possible correct solution. Accordingly, a Gibbs sampler excitation factor can be increased or decreased, thereby adjusting the transition probability.

5 In one example configuration, the excited MCMC Gibbs sampler 130 can be configured to calculate a distance of an estimated bit sequence from correct bit sequence for a radio signal received by a MIMO detector. To detect how close an estimate is to a correct sequence, a distance metric can be used. As can be seen in Eq. 13, the expectation of square distances with correct bit sequences is $N_r \sigma_n^2$ where N_r is the number of receive
10 antennas. And therefore δ_p in Eq. 14, the ratio of the current bit sequence's distance to the expected correct solution can be used as a measure of the closeness to a reasonable solution and an excitation control metric.

$$E[||\mathbf{y} - \mathbf{H}\mathbf{s}||^2] = N_r \sigma_n^2 \quad (13)$$

$$\delta_p = \min\left(\frac{||\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k-}||^2}{N_r \sigma_n^2}, \frac{||\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k+}||^2}{N_r \sigma_n^2}\right) \quad (14)$$

15 This has been a heuristic explanation of δ_p , but it can also be derived directly by conditioning the probabilities on how much error is being caused by $\mathbf{b}_{\setminus k}$. If we assume that this error has a Gaussian distribution with variance σ_b^2 and is independent of noise with variance σ_n^2 , then we have the following.

$$\gamma_k = \lambda_k^a + \frac{||\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k-}||^2 - ||\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k+}||^2}{\sigma_n^2 + \sigma_b^2} \quad (15)$$

20 Next, an estimate of $(\sigma_n^2 + \sigma_b^2)$ is needed. There are many possible ways this can be done such as by using the minimum or mean of the current Gibbs sampler distances

$$\sigma_n^2 + \sigma_b^2 \approx \frac{\min(||\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k-}||^2, ||\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k+}||^2)}{N_r}$$

$$\sigma_n^2 + \sigma_b^2 \approx \frac{||\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k-}||^2 + ||\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k+}||^2}{2N_r}$$

The minimum is a good approximation when close to a good solution region and the errors from $\mathbf{b}_{\setminus k}$ are small. The mean is a good approximation when far away from a

good solution region and the errors from $\mathbf{b}_{\setminus k}$ are significant. Note that other approximations to $(\sigma_n^2 + \sigma_b^2)$ may also be used. For example, a good approximation is to take the minimum or mean across all k -bits. Another option is to correct for the approximation by multiplying by a scaling term or to transition between different approximations as one gets closer to a good solution region using the distance $\|\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k\pm}\|^2$ as a decision metric.

Also, note that the heuristic and statistical explanations can be shown to be equivalent in practice as in Eq. 16.

$$\sigma_n^2 + \sigma_b^2 = \delta_p \sigma_n^2 \quad (16)$$

In one example, a proportional excitation control factor can be modified based in part on the distance of a bit sequence from an estimated correct bit sequence. The excitation factor may be used, in part, to calculate a probability of state transition of the bit sequence, which may be used to detect a pseudo-convergence condition attributed to the excitation factor having a low influence on the probability of state transition of the bit sequence. The excitation control factor δ_p can be used directly to scale γ_k as in Eq. 17. The excitation control factor can also be adjusted in strength or behavior by simple manipulations such as limiting to > 1 or less than a constant, linearly scaling by a constant coefficient to increase or decrease effect, adding a constant, integrating over time, passing through a shaping function, or passing through a shaping look-up-table.

$$\gamma_k = \frac{\|\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k-}\|^2 - \|\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k+}\|^2}{\delta \sigma_n^2} \quad (17)$$

After introducing the excitation control factor δ_p , a performance increase can be observed and the high SNR problem can be solved. Then what may be observed is that the Gibbs sampler quickly finds a reasonable solution and encounters pseudo-convergence, a slowing or stopping of the Gibbs sampler, wasting further iterations. When the Gibbs sampler gets close to a bit sequence with distance close to the expected correct answer in Eq. 13, the excitation factor becomes near unity thus having little to no excitation effect.

Therefore, this additional pseudo-convergence condition can be detected, triggering a method to help move the Gibbs sampler's state away from the already sampled important region of bit sequences.

Detection of a pseudo-convergence condition can be performed in many ways. The rate of state transitions can be measured and an anti-pseudo-convergence condition can be triggered after exceeding a threshold. A running average or a last cycle's minimum probability of transition can be used. Another method is to detect pseudo-convergence when no state transition have been made in one full cycle of bits. Instead of using a lack of state transitions, a lack of improvement in $\|\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k+}\|^2$ or $\|\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k+}\|^2$ over a Gibbs sampler cycle of bits can also be used.

Once pseudo-convergence has been detected, a mitigation method can be triggered. Again this can be done in many ways. The bits in the bit sequence can be randomized, a subset of the bits can be randomized, or the excitation of the Gibbs sampler can be increased. The method shown in Eq. 18 creates an additional excitation term δ_I , which may be the minimum ratio taken across all neighboring Gibbs sampler states defined as being a 1-bit change in $\mathbf{b}$. The excitation can be adjusted in strength or behavior by manipulations such as limiting to > 0 or less than a constant, linearly scaling by a constant coefficient to increase or decrease effect, adding a constant, integrating over time, passing through a shaping function, or passing through a shaping look-up-table.

$$\delta_I = \min_{\mathcal{S}_{\text{neighbors}}} \left(\frac{\|\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}\|^2}{N_r \sigma_n^2} \right) \quad (18)$$

where the minimum is taken across $\mathcal{S}_{\text{neighbors}}$, the list of Gibbs sampler states $\hat{\mathbf{s}}$ which are only 1-bit different from the current pseudo-converged state.

The excitation metrics can be combined using straightforward manipulations including addition, linear scaling with addition as in Eq. 19, look-up-tables, or shaping functions.

$$\delta = a\delta_P + b\delta_I + c \quad (19)$$

With the combination of these methods to detect closeness to a reasonable solution, detect importance region pseudo-convergence, and excite the Gibbs sampler, performance of MCMC is improved. The present technology is compatible with using a priori information and can be implemented in field-programmable gate arrays (FPGA) and silicon designs.

In one example, the excited MCMC Gibbs sampler 130 may comprise a distance of a best estimate to a correct unknown signal, detection and measure of pseudo-convergence, and a technique for exciting the Gibbs sampler accordingly. The distance may be a magnitude squared difference between the received signal and the best estimate. The distance may be a magnitude difference between the received signal and the best estimate. The distance may be an estimate of the magnitude or magnitude squared for example, but without restriction, where $\text{magnitude} = \alpha \max(|I|, |Q|) + \beta \min(|I|, |Q|)$ where I and Q are real and imaginary components and (α, β) are constant coefficients adjusted for estimate accuracy. The distance may be a more convenient form of the difference between the received signal and the best estimate.

In one example, a distance δ_p may be a best estimate to the unknown true transmitted signal. The best estimate may be the best visited bit-sequence estimate. The best estimate may be the best visited bit-sequence estimate since the last bit flip. The best estimate may be the best bit-sequence estimate calculated whether visited or not. δ_p may be a ratio between the best distance and the expectation of the unknown true transmitted signal's distance.

In one example, pseudo-convergence may be detected and an additional excitation term δ_f may be calculated, where pseudo-convergence may be the slow down and possible stopping of the Gibbs sampler transition to new states. Pseudo-convergence may be experienced because the Gibbs sampler may be less excited due to being close to a reasonable solution. Pseudo-convergence may be detected by going through all bits once without flipping. Pseudo-convergence may be detected by going through all bits once without sampling a state with an improved smaller distance $\|\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k+}\|^2$ or $\|\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k+}\|^2$. Pseudo-convergence may be detected by measuring a slowdown in a rate of bit flips. Pseudo-convergence may be detected by exceeding a threshold of low probability of

bit flips. δ_I may be either 0 or the best distance differing from the current bit sequence by 1-bit.

In one example, after a pseudo-convergence condition is detected and measured, the Gibbs sampler may be excited, thereby forcing the Gibbs sampler to move away from a pseudo-converged bit sequence. Forcing may be performed by re-initializing the bit sequence to a new sequence. Forcing may be performed by randomly flipping several of the current bit sequences. Forcing may be performed by adding a possibly scaled version of δ_I to δ_P to excite the Gibbs sampler until a move away from the current state occurs.

In one example, the Gibbs sampler may be excited by scaling the Gibbs sampler's probability of flip calculation. The excitation may be made by scaling a precursor to the Gibbs sampler's probability of flip calculation. The excitation may be made by using $\frac{1}{\delta}$ to scale γ_k in Eq. 5. The δ may be used to scale γ_k in Eq. 5, except for prior information. The δ may be a linear combination of δ_P and δ_I .

In one example, the adjustment may be made by directly applying $\frac{1}{f(\delta)}$ to γ_k , where $f(\delta)$ may apply a transformation function used to optimize the improvement. For example, $f(\delta)$ may be any combination of changes to δ , may be a linear scaling function, may be a polynomial scaling function, may be a look-up-table, and/or $f(\delta)$ may limit the input to specific extreme values.

In one example, the Gibbs sampler's rate of movement may be increased. For example, a look-ahead unit may be used. The look-ahead unit may calculate the next Gibbs sample if the current bit is flipped and if the current bit is not flipped. Once the current sample is finished, the future bit may be ready and may only need to be selected using a multiplexer, resulting in a doubling of rate. Alternatively, the look-ahead unit may calculate the next Gibbs sample assuming the current sample does not flip. Both the current and future sample may complete simultaneously and may be both used regardless of whether the current bit is flipped or not. In one example, the Gibbs sampler may execute bits in parallel.

In one example, the complexity of the Gibbs sampler implementation may be decreased by hard coding a starting bit sequence that provides good separation between parallel Gibbs samplers in order to remove the necessity for a random number generator

performing initialization. In another example, complexity may be decreased by double processing, which may be done to decrease the processing needed for a block comprising multiple independent bit sequences. Each bit sequence can be first estimated with less Gibbs sampler depth and/or width. Then the closeness measure can be used to identify a subset of the bit sequences in the block which do not appear to be close to a reasonable solution for additional processing.

In another example, a decrease in complexity can be done by using one random number generator for all parallel Gibbs samplers. In another example, a decrease in complexity can be done by using one random number generator for all parallel Gibbs samplers, but then XOR the random bit sequence with a hard coded bit sequence that is unique to each parallel Gibbs sampler.

In one example, performance of the Gibbs sampler may be improved by starting the Gibbs sampler in a location that is likely to be near the correct solution. As one example, a hard decision of the a priori information bit sequence may be used. In another example, a hard decision of a priori information bit sequence may be used in addition to parallel Gibbs samplers initialized to spatially neighboring points in a QAM constellation.

FIG. 2 is a flow diagram that illustrates an example method 200 for processing a received radio signal using an excited MCMC Gibbs sampler. As in block 210, the radio signal can be received at a MIMO detector and a bit sequence can be extracted from the radio signal. The method 200 can calculate a distance of the bit sequence from an expectation of correct bit sequences.

In one example, the distance of the bit sequence from the estimated correct bit sequence may be calculated by calculating a magnitude squared difference between the bit sequence and the estimated correct bit sequence. In another example, a magnitude difference between the bit sequence and the estimated correct bit sequence may be calculated. In yet another example, a magnitude estimate may be calculated, where the $\text{magnitude_estimate} = \alpha \max(|I|, |Q|) + \beta \min(|I|, |Q|)$ where I and Q are real and imaginary components and (α, β) are constant coefficients such as $(1, 1/2)$ adjusted for estimate accuracy. Also, calculating the distance of the bit sequence from the estimated correct bit sequence can include measuring δ_p distance from the estimated correct bit sequence to the bit sequence extracted from the received radio signal, wherein $\delta_p =$

$\min\left(\frac{\|\mathbf{y}-\mathbf{H}\hat{\mathbf{s}}_{k-}\|^2}{N_r\sigma_n^2}, \frac{\|\mathbf{y}-\mathbf{H}\hat{\mathbf{s}}_{k+}\|^2}{N_r\sigma_n^2}\right)$, where $\mathbf{y}$ is the received radio signal, $\mathbf{H}$ is attenuation and delay of the received radio signal, $\hat{\mathbf{s}}$ is the estimated correct bit sequence, N_r is a number of receive antennas, and σ_n^2 is a noise variance.

In one example configuration, the distance of the bit sequence from the estimated correct bit sequence may be measured using a ratio of the distance of the bit sequence to the estimated correct bit sequence. As one example, an estimated correct bit sequence distance may be $E[\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2] = N_r\sigma_n^2$ where N_r is a number of receive antennas, $\mathbf{y}$ is the received radio signal, $\mathbf{H}$ is attenuation and delay of the received radio signal, $\mathbf{s}$ is a transmitted signal, N_r is a number of receive antennas, and σ_n^2 is a noise variance. As another example, the ratio of the distance of the bit sequence to the estimated correct bit sequence may be $\delta_p = \min\left(\frac{\|\mathbf{y}-\mathbf{H}\hat{\mathbf{s}}_{k-}\|^2}{N_r\sigma_n^2}, \frac{\|\mathbf{y}-\mathbf{H}\hat{\mathbf{s}}_{k+}\|^2}{N_r\sigma_n^2}\right)$, where $\mathbf{y}$ is the received radio signal, $\mathbf{H}$ is attenuation and delay of the received radio signal, $\hat{\mathbf{s}}$ is the estimated correct bit sequence, N_r is a number of receive antennas, and σ_n^2 is a noise variance.

As in block 220, an excitation control factor can be modified based in part on the distance of the bit sequence from the expectation of correct bit sequence, where the excitation control factor can be used in part to calculate a probability of state transition of the bit sequence. In one example, the excitation control factor may be scaled as described earlier. For example, the excitation control factor can be linearly scaled using a constant coefficient, or the excitation control factor can be linearly scaled using $\frac{1}{\delta}$, where δ is the distance of the bit sequence from the estimated correct bit sequence. In some examples, scaling the excitation control factor may include monitoring the state of the bits in the bit sequence for a state transition in the bits. Illustratively, the excitation control factor may be limited to > 0 .

As in block 230, a pseudo-convergence condition (i.e., pseudo-convergence of the probability of state transition of the bit sequence) attributed to the excitation factor can be detected. In one example, the pseudo-convergence condition may be detected by measuring a state transition rate for the bit sequence to determine that a transition rate threshold has been exceeded. The transition rate threshold may represent a low rate of state transitions of the bit sequence. In another example, the pseudo-convergence condition may be detected

via an absence of state transitions of bits included in the bit sequence after a cycle of analyzing the bit sequence.

In detecting a pseudo-convergence condition, as in block 240, a pseudo-convergence mitigation technique that causes the probability of state transition of the bit sequence to increase can be executed. In one example, the pseudo-convergence mitigation technique may include reinitializing the bit sequence to a new bit sequence. In another example, the pseudo-convergence mitigation technique may include selecting a random subset of bits in the bit sequence and changing the state of the random subset of bits to form the new bit sequence. In yet another example, the pseudo-convergence mitigation technique may include scaling the excitation control factor and monitoring the state of the bits in the bit sequence for a state transition in the bits.

FIG. 3 illustrates a computing device 310 on which modules of this technology may execute. A computing device 310 is illustrated on which a high level example of the technology may be executed. The computing device 310 may include one or more processors 312 that are in communication with memory devices 320. In some examples, a processor 312 may be a VLSI or FPGA device and a memory device 320 may be an integrated cache memory device. The computing device 310 may include a local communication interface 318 for the components in the computing device. For example, the local communication interface 318 may be a local data bus and/or any related address or control busses as may be desired.

The memory device 320 may contain modules 324 that are executable by the processor(s) 312 and data for the modules 324. The modules 324 may execute the functions described earlier. A data store 322 may also be located in the memory device 320 for storing data related to the modules 324 and other applications along with an operating system that is executable by the processor(s) 312.

Other applications may also be stored in the memory device 320 and may be executable by the processor(s) 312. Components or modules discussed in this description that may be implemented in the form of software using high programming level languages that are compiled, interpreted or executed using a hybrid of the methods.

The computing device may also have access to I/O (input/output) devices 314 that are usable by the computing devices. Networking devices 316 and similar communication

devices may be included in the computing device. The networking devices 316 may be wired or wireless networking devices that connect to the internet, a LAN, WAN, cellular, or other computing network.

The components or modules that are shown as being stored in the memory device 320 may be executed by the processor(s) 312. The term “executable” may mean a program file that is in a form that may be executed by a processor 312. For example, a program in a higher level language may be compiled into machine code in a format that may be loaded into a random access portion of the memory device 320 and executed by the processor 312, or source code may be loaded by another executable program and interpreted to generate instructions in a random access portion of the memory to be executed by a processor. The executable program may be stored in any portion or component of the memory device 320. For example, the memory device 320 may be random access memory (RAM), read only memory (ROM), flash memory, a solid state drive, memory card, a hard drive, optical disk, floppy disk, magnetic tape, or any other memory components.

The processor 312 may represent multiple processors and the memory device 320 may represent multiple memory units that operate in parallel to the processing circuits. This may provide parallel processing channels for the processes and data in the system. The local interface 318 may be used as a network to facilitate communication between any of the multiple processors and multiple memories. The local interface 318 may use additional systems designed for coordinating communication such as load balancing, bulk data transfer, and similar systems.

FIG. 4 provides an example illustration of a user equipment (UE) device 400 and a node 420. The UE device 400 and node 420 illustrate devices on which the technology described herein may be used. The UE device 400 can include a wireless device, a mobile station (MS), a mobile wireless device, a mobile communication device, a tablet, a handset, or other type of wireless device. The UE device 400 can include antennas configured to communicate with the node 420 or transmission station, such as a base station (BS), an evolved Node B (eNB), a baseband unit (BBU), a remote radio head (RRH), a remote radio equipment (RRE), a relay station (RS), a radio equipment (RE), a remote radio unit (RRU), a central processing module (CPM), or other type of wireless wide area network (WWAN) access point. The node 420 can include one or more processors 422, memory 424 and a

transceiver 426. The UE device 400 can be configured to communicate using at least one wireless communication standard including 3GPP LTE, WiMAX, High Speed Packet Access (HSPA), Bluetooth, and WiFi. The UE device 400 can communicate using separate antennas for each wireless communication standard or shared antennas for multiple wireless communication standards. The UE device 400 can communicate in a wireless local area network (WLAN), a wireless personal area network (WPAN), and/or a WWAN.

In some embodiments, the UE device 400 may include application circuitry 402, baseband circuitry 404, Radio Frequency (RF) circuitry 406, front-end module (FEM) circuitry 408 and one or more antennas 410, coupled together at least as shown. In addition, the node 420 may include, similar to that described for the UE device 400, application circuitry, baseband circuitry, Radio Frequency (RF) circuitry, front-end module (FEM) circuitry and one or more antennas

The application circuitry 402 may include one or more application processors. For example, the application circuitry 402 may include circuitry such as, but not limited to, one or more single-core or multi-core processors. The processor(s) may include any combination of general-purpose processors and dedicated processors (e.g., graphics processors, application processors, etc.). The processors may be coupled with and/or may include a storage medium, and may be configured to execute instructions stored in the storage medium to enable various applications and/or operating systems to run on the system.

The baseband circuitry 404 may include circuitry such as, but not limited to, one or more single-core or multi-core processors. The baseband circuitry 404 may include one or more baseband processors and/or control logic to process baseband signals received from a receive signal path of the RF circuitry 406 and to generate baseband signals for a transmit signal path of the RF circuitry 406. Baseband processing circuitry 404 may interface with the application circuitry 402 for generation and processing of the baseband signals and for controlling operations of the RF circuitry 406. For example, in some embodiments, the baseband circuitry 404 may include a second generation (2G) baseband processor 404a, third generation (3G) baseband processor 404b, fourth generation (4G) baseband processor 404c, and/or other baseband processor(s) 404d for other existing generations, generations in development or to be developed in the future (e.g., fifth generation (5G), 6G, etc.). The

baseband circuitry 404 (e.g., one or more of baseband processors 404a-d) may handle various radio control functions that enable communication with one or more radio networks via the RF circuitry 406. The radio control functions may include, but are not limited to, signal modulation/demodulation, encoding/decoding, radio frequency shifting, etc. In some embodiments, modulation/demodulation circuitry of the baseband circuitry 404 may include Fast-Fourier Transform (FFT), precoding, and/or constellation mapping/demapping functionality. In some embodiments, encoding/decoding circuitry of the baseband circuitry 404 may include convolution, tail-biting convolution, turbo, Viterbi, and/or Low Density Parity Check (LDPC) encoder/decoder functionality. Embodiments of modulation/demodulation and encoder/decoder functionality are not limited to these examples and may include other suitable functionality in other embodiments.

In some embodiments, the baseband circuitry 404 may include elements of a protocol stack such as, for example, elements of an evolved universal terrestrial radio access network (EUTRAN) protocol including, for example, physical (PHY), media access control (MAC), radio link control (RLC), packet data convergence protocol (PDCP), and/or radio resource control (RRC) elements. A central processing unit (CPU) 404e of the baseband circuitry 404 may be configured to run elements of the protocol stack for signaling of the PHY, MAC, RLC, PDCP and/or RRC layers. In some embodiments, the baseband circuitry may include one or more audio digital signal processor(s) (DSP) 404f. The audio DSP(s) 104f may be include elements for compression/decompression and echo cancellation and may include other suitable processing elements in other embodiments. Components of the baseband circuitry may be suitably combined in a single chip, a single chipset, or disposed on a same circuit board in some embodiments. In some embodiments, some or all of the constituent components of the baseband circuitry 404 and the application circuitry 402 may be implemented together such as, for example, on a system on a chip (SOC).

In some embodiments, the baseband circuitry 404 may provide for communication compatible with one or more radio technologies. For example, in some embodiments, the baseband circuitry 404 may support communication with an evolved universal terrestrial radio access network (EUTRAN) and/or other wireless metropolitan area networks (WMAN), a wireless local area network (WLAN), a wireless personal area network

(WPAN). Embodiments in which the baseband circuitry 404 is configured to support radio communications of more than one wireless protocol may be referred to as multi-mode baseband circuitry.

The RF circuitry 406 may enable communication with wireless networks using modulated electromagnetic radiation through a non-solid medium. In various embodiments, the RF circuitry 406 may include switches, filters, amplifiers, etc. to facilitate the communication with the wireless network. RF circuitry 406 may include a receive signal path which may include circuitry to down-convert RF signals received from the FEM circuitry 408 and provide baseband signals to the baseband circuitry 404. RF circuitry 406 may also include a transmit signal path which may include circuitry to up-convert baseband signals provided by the baseband circuitry 404 and provide RF output signals to the FEM circuitry 408 for transmission.

In some embodiments, the RF circuitry 406 may include a receive signal path and a transmit signal path. The receive signal path of the RF circuitry 406 may include mixer circuitry 406a, amplifier circuitry 406b and filter circuitry 406c. The transmit signal path of the RF circuitry 406 may include filter circuitry 406c and mixer circuitry 406a. RF circuitry 406 may also include synthesizer circuitry 406d for synthesizing a frequency for use by the mixer circuitry 406a of the receive signal path and the transmit signal path. In some embodiments, the mixer circuitry 406a of the receive signal path may be configured to down-convert RF signals received from the FEM circuitry 408 based on the synthesized frequency provided by synthesizer circuitry 406d. The amplifier circuitry 406b may be configured to amplify the down-converted signals and the filter circuitry 406c may be a low-pass filter (LPF) or band-pass filter (BPF) configured to remove unwanted signals from the down-converted signals to generate output baseband signals. Output baseband signals may be provided to the baseband circuitry 404 for further processing. In some embodiments, the output baseband signals may be zero-frequency baseband signals, although this is not a necessity. In some embodiments, mixer circuitry 406a of the receive signal path may comprise passive mixers, although the scope of the embodiments is not limited in this respect.

In some embodiments, the mixer circuitry 406a of the transmit signal path may be configured to up-convert input baseband signals based on the synthesized frequency

provided by the synthesizer circuitry 406d to generate RF output signals for the FEM circuitry 408. The baseband signals may be provided by the baseband circuitry 404 and may be filtered by filter circuitry 406c. The filter circuitry 406c may include a low-pass filter (LPF), although the scope of the embodiments is not limited in this respect.

5 In some embodiments, the mixer circuitry 406a of the receive signal path and the mixer circuitry 406a of the transmit signal path may include two or more mixers and may be arranged for quadrature down-conversion and/or up-conversion respectively. In some embodiments, the mixer circuitry 406a of the receive signal path and the mixer circuitry 406a of the transmit signal path may include two or more mixers and may be arranged for
10 image rejection (e.g., Hartley image rejection). In some embodiments, the mixer circuitry 406a of the receive signal path and the mixer circuitry 406a may be arranged for direct down-conversion and/or direct up-conversion, respectively. In some embodiments, the mixer circuitry 406a of the receive signal path and the mixer circuitry 406a of the transmit signal path may be configured for super-heterodyne operation.

15 In some embodiments, the output baseband signals and the input baseband signals may be analog baseband signals, although the scope of the embodiments is not limited in this respect. In some alternate embodiments, the output baseband signals and the input baseband signals may be digital baseband signals. In these alternate embodiments, the RF circuitry 406 may include analog-to-digital converter (ADC) and digital-to-analog
20 converter (DAC) circuitry and the baseband circuitry 404 may include a digital baseband interface to communicate with the RF circuitry 406.

In some dual-mode embodiments, a separate radio IC circuitry may be provided for processing signals for each spectrum, although the scope of the embodiments is not limited in this respect.

25 In some embodiments, the synthesizer circuitry 406d may be a fractional-N synthesizer or a fractional $N/N+1$ synthesizer, although the scope of the embodiments is not limited in this respect as other types of frequency synthesizers may be suitable. For example, synthesizer circuitry 406d may be a delta-sigma synthesizer, a frequency multiplier, or a synthesizer comprising a phase-locked loop with a frequency divider.

30 The synthesizer circuitry 406d may be configured to synthesize an output frequency for use by the mixer circuitry 406a of the RF circuitry 406 based on a frequency input and a

divider control input. In some embodiments, the synthesizer circuitry 406d may be a fractional $N/N+1$ synthesizer.

In some embodiments, frequency input may be provided by a voltage controlled oscillator (VCO), although that is not a necessity. Divider control input may be provided by
5 either the baseband circuitry 404 or the applications processor 402 depending on the desired output frequency. In some embodiments, a divider control input (e.g., N) may be determined from a look-up table based on a channel indicated by the applications processor 402.

Synthesizer circuitry 406d of the RF circuitry 406 may include a divider, a delay-
10 locked loop (DLL), a multiplexer and a phase accumulator. In some embodiments, the divider may be a dual modulus divider (DMD) and the phase accumulator may be a digital phase accumulator (DPA). In some embodiments, the DMD may be configured to divide the input signal by either N or $N+1$ (e.g., based on a carry out) to provide a fractional division ratio. In some example embodiments, the DLL may include a set of cascaded,
15 tunable, delay elements, a phase detector, a charge pump and a D-type flip-flop. In these embodiments, the delay elements may be configured to break a VCO period up into N_d equal packets of phase, where N_d is the number of delay elements in the delay line. In this way, the DLL provides negative feedback to help ensure that the total delay through the delay line is one VCO cycle.

In some embodiments, synthesizer circuitry 406d may be configured to generate a
20 carrier frequency as the output frequency, while in other embodiments, the output frequency may be a multiple of the carrier frequency (e.g., twice the carrier frequency, four times the carrier frequency) and used in conjunction with quadrature generator and divider circuitry to generate multiple signals at the carrier frequency with multiple different phases
25 with respect to each other. In some embodiments, the output frequency may be a LO frequency (fLO). In some embodiments, the RF circuitry 406 may include an IQ/polar converter.

FEM circuitry 408 may include a receive signal path which may include circuitry configured to operate on RF signals received from one or more antennas 410, amplify the
30 received signals and provide the amplified versions of the received signals to the RF circuitry 406 for further processing. FEM circuitry 408 may also include a transmit signal

path which may include circuitry configured to amplify signals for transmission provided by the RF circuitry 406 for transmission by one or more of the one or more antennas 410.

In some embodiments, the FEM circuitry 408 may include a TX/RX switch to switch between transmit mode and receive mode operation. The FEM circuitry may include a receive signal path and a transmit signal path. The receive signal path of the FEM circuitry may include a low-noise amplifier (LNA) to amplify received RF signals and provide the amplified received RF signals as an output (e.g., to the RF circuitry 406). The transmit signal path of the FEM circuitry 408 may include a power amplifier (PA) to amplify input RF signals (e.g., provided by RF circuitry 406), and one or more filters to generate RF signals for subsequent transmission (e.g., by one or more of the one or more antennas 410).

Various techniques, or certain aspects or portions thereof, may take the form of program code (i.e., instructions) embodied in tangible media, such as floppy diskettes, compact disc-read-only memory (CD-ROMs), hard drives, non-transitory computer readable storage medium, or any other machine-readable storage medium wherein, when the program code is loaded into and executed by a machine, such as a computer, the machine becomes an apparatus for practicing the various techniques. In the case of program code execution on programmable computers, the computing device may include a processor, a storage medium readable by the processor (including volatile and non-volatile memory and/or storage elements), at least one input device, and at least one output device. The volatile and non-volatile memory and/or storage elements may be a random-access memory (RAM), erasable programmable read only memory (EPROM), flash drive, optical drive, magnetic hard drive, solid state drive, or other medium for storing electronic data. The node and wireless device may also include a transceiver module (i.e., transceiver), a counter module (i.e., counter), a processing module (i.e., processor), and/or a clock module (i.e., clock) or timer module (i.e., timer). In one example, selected components of the transceiver module can be located in a cloud radio access network (C-RAN). One or more programs that may implement or utilize the various techniques described herein may use an application programming interface (API), reusable controls, and the like. Such programs may be implemented in a high level procedural or object oriented programming language to communicate with a computer system. However, the program(s) may be implemented in

assembly or machine language, if desired. In any case, the language may be a compiled or interpreted language, and combined with hardware implementations.

As used herein, the term "circuitry" may refer to, be part of, or include an Application Specific Integrated Circuit (ASIC), an electronic circuit, a processor (shared, dedicated, or group), and/or memory (shared, dedicated, or group) that execute one or more software or firmware programs, a combinational logic circuit, and/or other suitable hardware components that provide the described functionality. In some embodiments, the circuitry may be implemented in, or functions associated with the circuitry may be implemented by, one or more software or firmware modules. In some embodiments, circuitry may include logic, at least partially operable in hardware.

It should be understood that many of the functional units described in this specification have been labeled as modules, in order to more particularly emphasize their implementation independence. For example, a module may be implemented as a hardware circuit comprising custom very-large-scale integration (VLSI) circuits or gate arrays, off-the-shelf semiconductors such as logic chips, transistors, or other discrete components. A module may also be implemented in programmable hardware devices such as field programmable gate arrays, programmable array logic, programmable logic devices or the like.

Modules may also be implemented in software for execution by various types of processors. An identified module of executable code may, for instance, comprise one or more physical or logical blocks of computer instructions, which may, for instance, be organized as an object, procedure, or function. Nevertheless, the executables of an identified module may not be physically located together, but may comprise disparate instructions stored in different locations which, when joined logically together, comprise the module and achieve the stated purpose for the module.

Indeed, a module of executable code may be a single instruction, or many instructions, and may even be distributed over several different code segments, among different programs, and across several memory devices. Similarly, operational data may be identified and illustrated herein within modules, and may be embodied in any suitable form and organized within any suitable type of data structure. The operational data may be collected as a single data set, or may be distributed over different locations including over

different storage devices, and may exist, at least partially, merely as electronic signals on a system or network. The modules may be passive or active, including agents operable to perform desired functions.

Furthermore, the described features, structures, or characteristics may be combined
5 in any suitable manner in one or more embodiments. In the following description, numerous specific details are provided, such as examples of layouts, distances, network examples, etc., to provide a thorough understanding of embodiments of the technology. One skilled in the relevant art will recognize, however, that the technology can be practiced without one or more of the specific details, or with other methods, components, layouts,
10 etc. In other instances, well-known structures, materials, or operations are not shown or described in detail to avoid obscuring aspects of the technology.

While the forgoing examples are illustrative of the principles of the present technology in one or more particular applications, it will be apparent to those of ordinary skill in the art that numerous modifications in form, usage and details of implementation
15 can be made without the exercise of inventive faculty, and without departing from the principles and concepts of the technology. Accordingly, it is not intended that the technology be limited, except as by the claims set forth below.

The foregoing detailed description describes the invention with reference to specific exemplary embodiments. However, it will be appreciated that various modifications and
20 changes can be made without departing from the scope of the present invention as set forth in the appended claims. The detailed description and accompanying drawings are to be regarded as merely illustrative, rather than as restrictive, and all such modifications or changes, if any, are intended to fall within the scope of the present invention as described and set forth herein.

25

CLAIMS

What is claimed is:

1. A method for exciting a Markov Chain Monte Carlo (MCMC) Gibbs sampler, comprising:

5 calculating, using a processor, a distance of a bit sequence from an expectation of correct bit sequences, wherein the bit sequence is obtained from a received radio signal;

modifying, using the processor, an excitation control factor based in part on the distance of the bit sequence from the expectation of correct bit sequences, wherein the excitation control factor is used in part to calculate a probability of state transition of the
10 bit sequence;

detecting, using the processor, a pseudo-convergence condition of the probability of state transition of the bit sequence that is attributed to the excitation control factor; and

15 executing, using the processor, a pseudo-convergence mitigation technique that causes the probability of state transition of the bit sequence to increase.

2. A method as in claim 1, wherein calculating the distance of the bit sequence from the expectation of correct bit sequences further comprises calculating a magnitude squared difference between the bit sequence and an estimated correct bit sequence.

3. A method as in claim 1, wherein calculating the distance of the bit sequence from the expectation of correct bit sequences further comprises calculating a magnitude difference between the bit sequence and the estimated correct bit sequence.

4. A method as in claim 1, wherein calculating the distance of the bit sequence from the expectation of correct bit sequences further comprises calculating a
25 magnitude_estimate = $\alpha \max(|I|, |Q|) + \beta \min(|I|, |Q|)$ where I and Q are real and imaginary components and (α, β) are constant coefficients such as $(1, 1/2)$ adjusted for estimate accuracy.

5. A method as in claim 1, wherein calculating the distance of the bit sequence from the expectation of correct bit sequences further comprises measuring δ_p distance from the estimated correct bit sequence to the bit sequence extracted from the received radio signal, wherein $\delta_p = \min \left(\frac{\|y - H\hat{s}_{k-}\|^2}{N_r \sigma_n^2}, \frac{\|y - H\hat{s}_{k+}\|^2}{N_r \sigma_n^2} \right)$, where y is the received radio signal, H is attenuation and delay of the received radio signal, $\hat{s}$ is the estimated bit sequence, N_r is a number of receive antennas, and σ_n^2 is a noise variance.

6. A method as in claim 1, wherein modifying the excitation control factor further comprises scaling the excitation control factor.

7. A method as in claim 1, wherein detecting the pseudo-convergence condition further comprises measuring a state transition rate for the bit sequence to determine that a transition rate threshold has been exceeded, wherein the transition rate threshold represents a low rate of state transitions of the bit sequence.

8. A method as in claim 1, wherein detecting the pseudo-convergence condition further comprises detecting an absence of state transitions of bits included in the bit sequence after a cycle of analyzing the bit sequence.

9. A method as in claim 1, wherein executing the pseudo-convergence mitigation technique further comprises reinitializing the bit sequence to a new bit sequence.

10. A method as in claim 9, wherein executing the pseudo-convergence mitigation technique further comprises:

selecting a random subset of bits in the bit sequence; and
changing the state of the random subset of bits to form the new bit sequence.

11. A method as in claim 1, wherein executing the pseudo-convergence mitigation technique further comprises scaling the excitation control factor and monitoring the state of the bits in the bit sequence for a state transition in the bits.

12. A method as in claim 1, wherein detecting the pseudo-convergence condition further comprises going through all bits once without sampling a state with an improved smaller distance $\|\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k+}\|^2$ or $\|\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k+}\|^2$, where $\mathbf{y}$ is the received radio signal, $\mathbf{H}$ is attenuation and delay of the received radio signal, and $\hat{\mathbf{s}}$ is the estimated bit sequence.

5

13. A system for an excited Markov Chain Monte Carlo (MCMC) Gibbs sampler comprising:

a processor;

a memory device including instructions that, when executed by the processor,

10 cause the system to:

calculate a distance of a bit sequence from an estimated correct bit sequence for a radio signal received by a Multiple-Input and Multiple-Output (MIMO) detector;

modify a first excitation control factor based in part on the distance of the bit sequence from the estimated correct bit sequence, wherein the excitation control factor is used in part to calculate a probability of state transition of the bit sequence;

15

detect a pseudo-convergence condition attributed to the first excitation control factor having a low influence on the probability of state transition of the bit sequence;

set a second excitation control factor to the distance of the bit sequence from the estimated correct bit sequence combined with the first excitation control factor; and

20

apply the second excitation control factor to the Gibbs sampler, thereby increasing the probability of state transition of the bit sequence.

14. A system as in claim 13, wherein the memory device includes instructions that, when executed by the processor, causes the system to further linearly scale the second excitation control factor by a constant coefficient.

25

15. A system as in claim 14, wherein the memory device includes instructions that, when executed by the processor, causes the system to linearly scale the second excitation control factor using $\frac{1}{\delta}$, where δ is the distance of the bit sequence from the estimated correct bit sequence.

30

16. A system as in claim 13, wherein the memory device includes instructions that, when executed by the processor, causes the system to further limit the second excitation control factor to > 0 .

17. A system as in claim 13, wherein the memory device includes instructions that, when executed by the processor, causes the system to further pass the second excitation control factor through a shaping function.

18. A system as in claim 13, wherein the processor is a VLSI or FPGA device and the memory device is an integrated cache memory.

19. An apparatus comprising:
 a memory controller having circuitry configured to:
 calculate a distance of a bit sequence from an estimated correct bit sequence for
 a received radio signal received by a Multiple-Input and Multiple-Output (MIMO) detector;
 modify an excitation control factor based in part on the distance of the bit
 sequence from the estimated correct bit sequence, wherein the excitation control factor is
 used in part to calculate a probability of state transition of the bit sequence;
 detect a pseudo-convergence condition attributed to the excitation control factor
 having a low influence on the probability of state transition of the bit sequence; and
 execute a pseudo-convergence mitigation technique that causes the probability
 of state transition of the bit sequence to increase.

20. An apparatus of claim 19, wherein the memory controller further has circuitry configured to measure the distance of the bit sequence from the estimated correct bit sequence using a ratio of the distance of the bit sequence to the estimated correct bit sequence.

21. An apparatus of claim 20, wherein the estimated correct bit sequence distance is $E[\|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2] = N_r \sigma_n^2$ where N_r is a number of receive antennas, wherein $\mathbf{y}$ is the

received radio signal, $\mathbf{H}$ is attenuation and delay of the received radio signal, $\mathbf{s}$ is a transmitted signal, N_r is a number of receive antennas, and σ_n^2 is a noise variance.

22. An apparatus of claim 20, wherein the ratio of the distance of the bit sequence
 5 to the estimated correct bit sequence is $\delta_p = \min\left(\frac{\|\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k-}\|^2}{N_r \sigma_n^2}, \frac{\|\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{k+}\|^2}{N_r \sigma_n^2}\right)$, wherein $\mathbf{y}$ is the received radio signal, $\mathbf{H}$ is attenuation and delay of the received radio signal, $\hat{\mathbf{s}}$ is the estimated correct bit sequence, N_r is a number of receive antennas, and σ_n^2 is a noise variance.

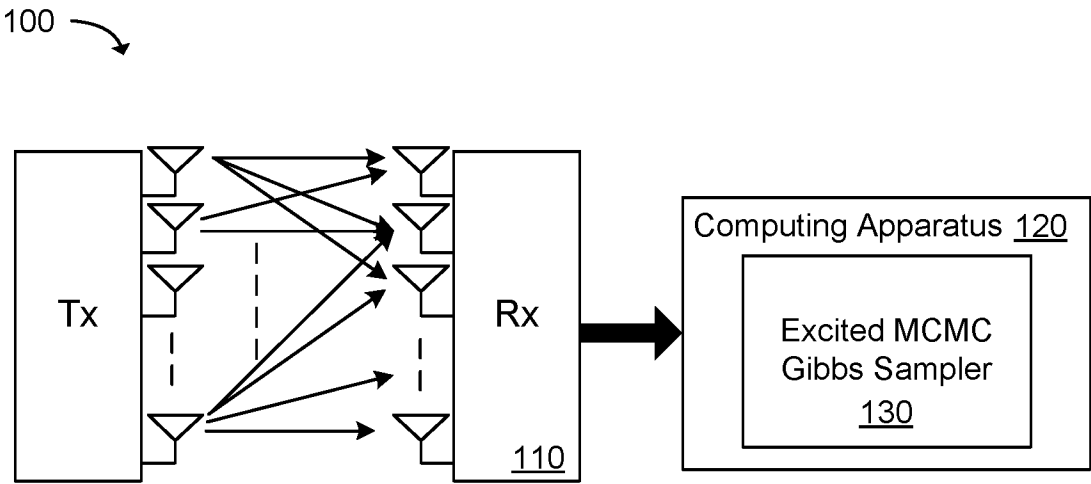


FIG. 1

2/5

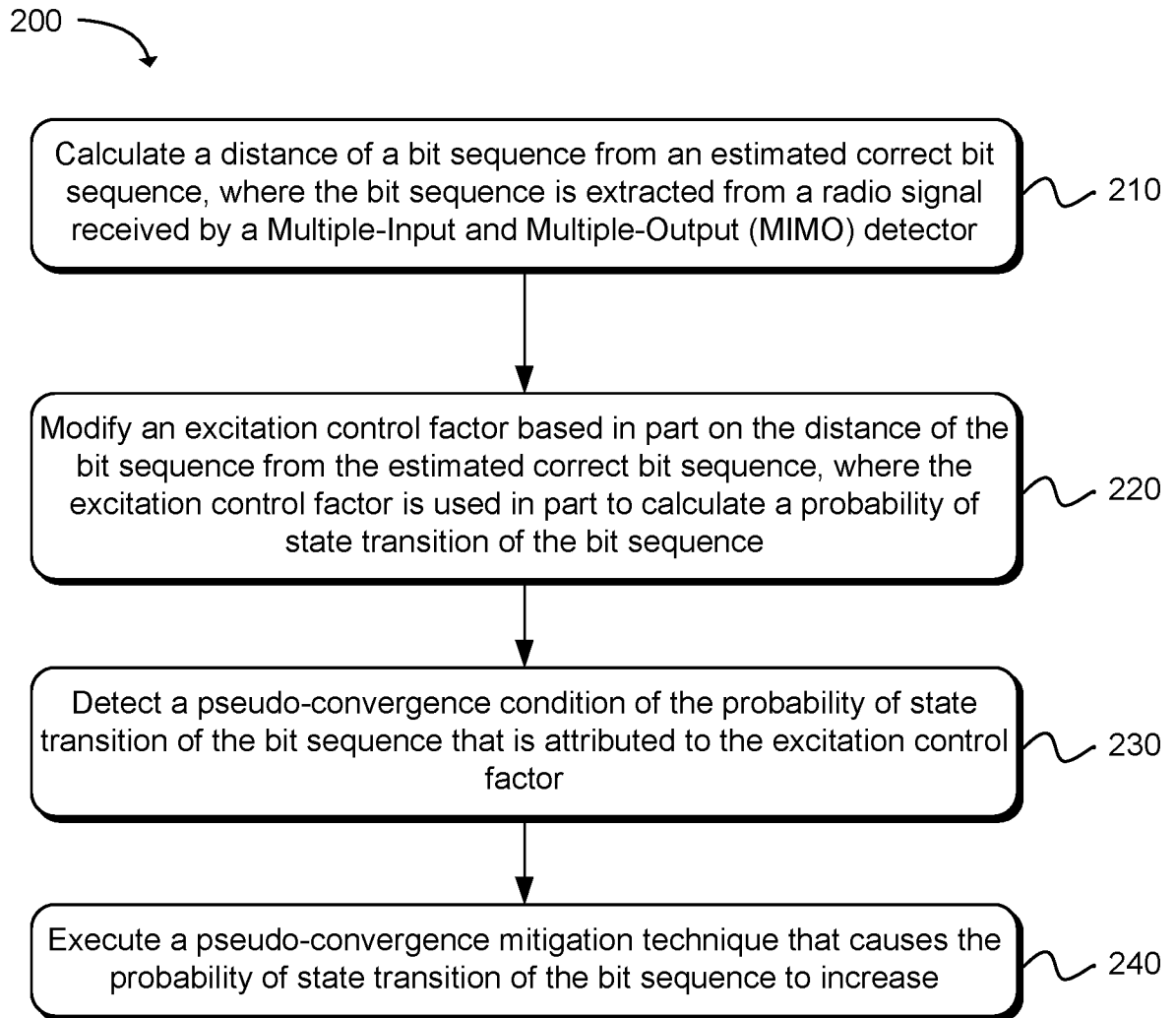


FIG. 2

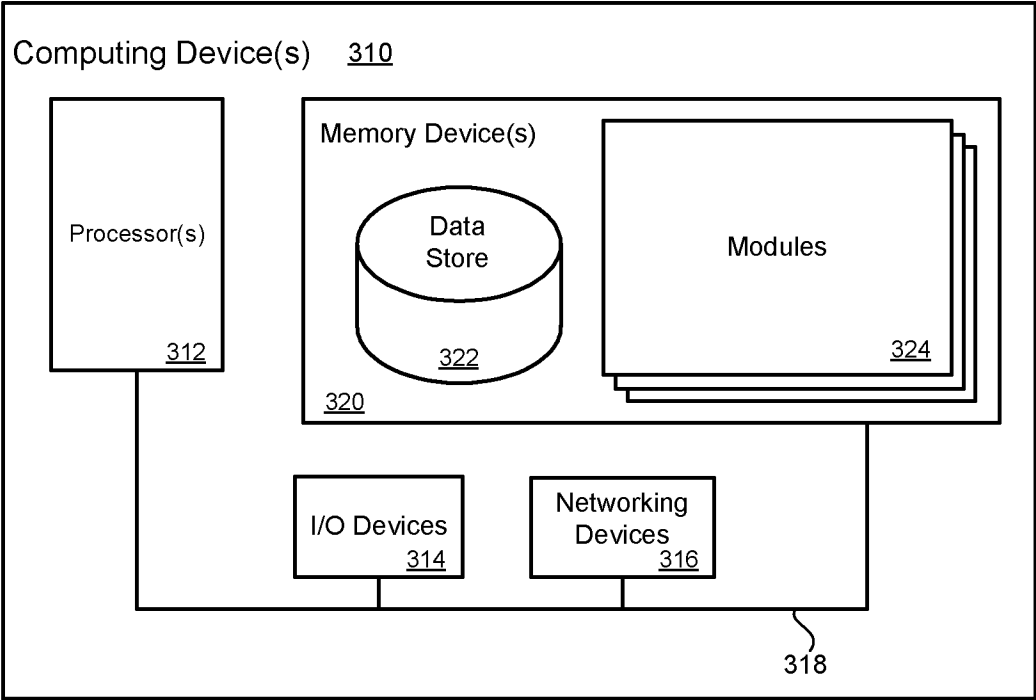


FIG. 3

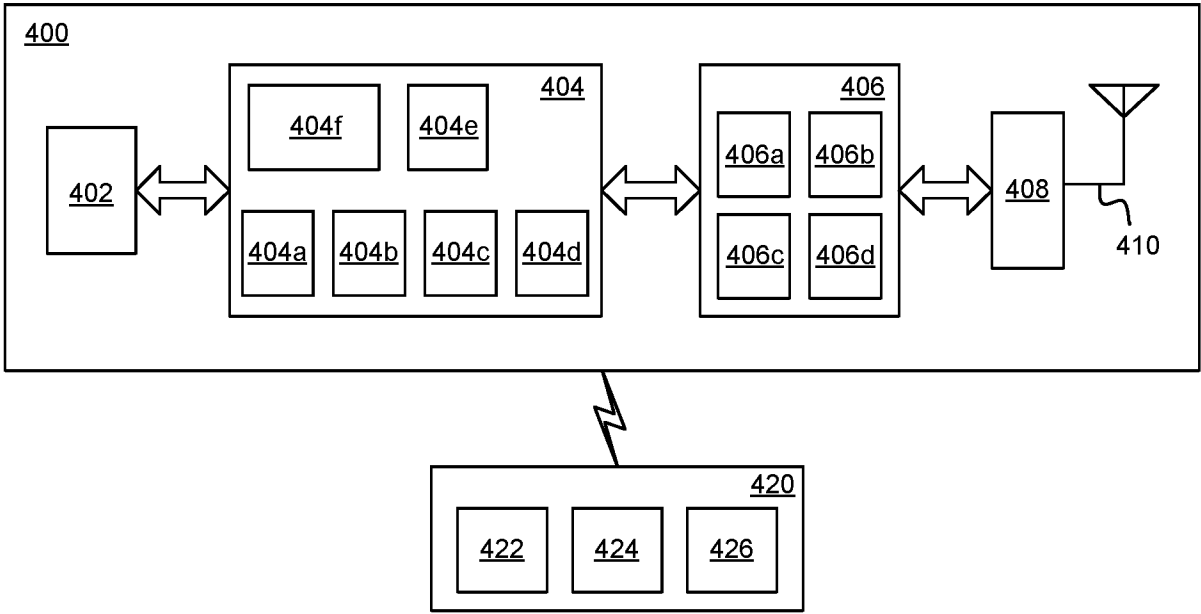


FIG. 4

5/5

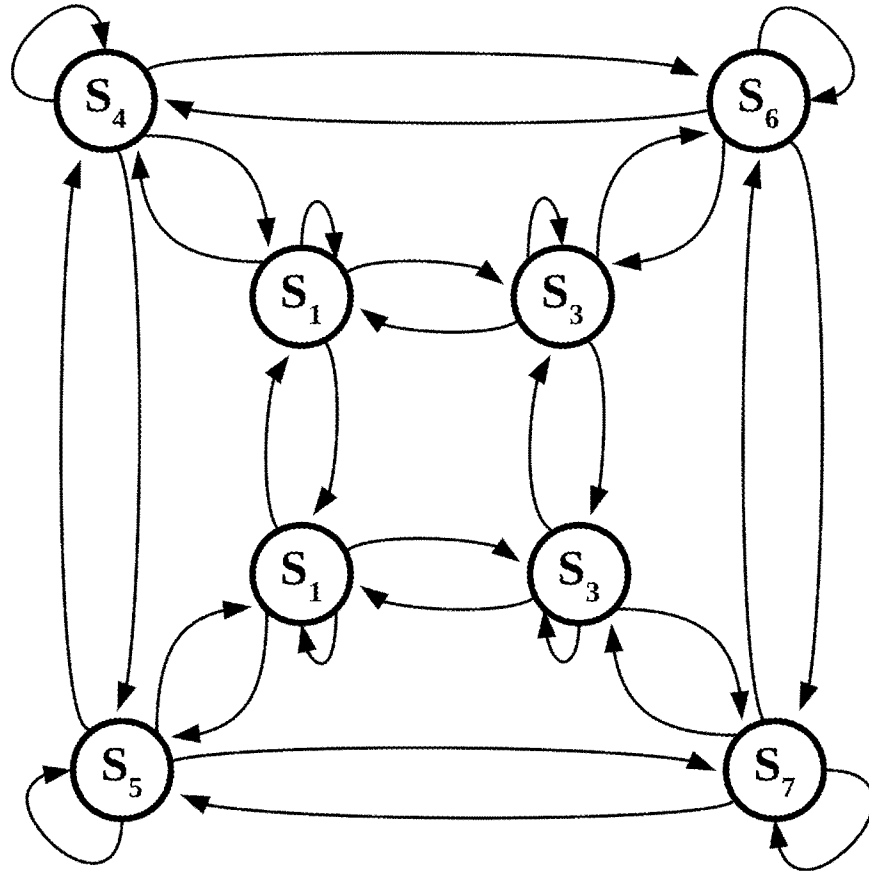


FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 16/66192

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - H04B 7/08 (2017.01)

CPC - H04B 7/0452; H04B 7/0848; H04B 7/0413; H04L 1/0631; H03D 1/00

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
CPC: H04B 7/0452; H04B 7/0848; H04B 7/0413; H04L 1/0631; H03D 1/00Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
IPC(8): H04B 7/08 (2017.01); USPC: 370/252; 375/260,267; CPC: H04B 7/0452; H04B 7/0848; H04B 7/0413; H04L 1/0631; H03D 1/00
(keyword limited; terms below)Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PatBase (PatBase); PubWEST(USPT,PGPB,EPAB,JPAB); Google Scholar
Search Terms: Markov Chain Monte Carlo, mcmc, Gibbs, sampler, distance, pseudo-convergence, bit sequence, excitation control, correct, magnitude, estimate, state, transition, noise, variance, bit.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Peng et al. Markov Chain Monte Carlo Detectors for Channels With Intersymbol Interference. 13 March 2010 (13.03.2010). Retrieved from the Internet at: https://pdfs.semanticscholar.org/2272/a9419bc36068c6b54f2e05736151f0fda241.pdf . Pp 2206-2217.	1-22
A	US 2011/0110449 A1 (RAMPRASHAD et al.) 12 May 2011 (12.05.2011), entire document, especially; para [0056], [0063], [0067], [0070], [0109].	1-22
A	US 2009/0220016 A1 (DANGL et al.) 03 September 2009 (03.09.2009), entire document, especially; para [0007], [0027], [0085], [0091].	1-22
A	US 2014/0223439 A1 (VIGODA) 07 August 2014 (07.08.2014), entire document.	1-22
A	US 2004/0010399 A1 (KADANE et al.) 15 January 2004 (15.01.2004), entire document.	1-22
A	US 2009/0213966 A1 (TEO et al.) 27 August 2009 (27.08.2009), entire document.	1-22
A	US 2012/0307924 A1 (CHOCKALINGAM et al.) 06 December 2012 (06.12.2012), entire document.	1-22

☐ Further documents are listed in the continuation of Box C.

* 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 application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

04 February 2017

Date of mailing of the international search report

27 MAR 2017

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, Virginia 22313-1450
Facsimile No. 571-273-8300

Authorized officer:

Lee W. Young

PCT Helpdesk: 571-272-4300
PCT OSP: 571-272-7774