



(51) International Patent Classification:

H04B 7/04 (2017.01) *H04B 1/10* (2006.01)
H04L 25/02 (2006.01)

(21) International Application Number:

PCT/CN2018/082713

(22) International Filing Date:

11 April 2018 (11.04.2018)

(25) Filing Language:

English

(26) Publication Language:

English

(71) Applicants (*for CN only*): **NOKIA SHANGHAI BELL CO., LTD.** [CN/CN]; No. 388, Ningqiao Road, Pudong Jin-qiao, Shanghai 201206 (CN). **NOKIA SOLUTIONS AND NETWORKS OY** [FI/FI]; Karaportti 3, 02610 Espoo (FI).

(71) Applicant (*for all designated States except CN*): **NOKIA TECHNOLOGIES OY** [FI/FI]; Karaportti 3, 02610 Espoo (FI).

(72) Inventors: **MAO, Weiqiang**; Room 501, Unit 3, Build 2, Qiantangjinyuan, No. 399 Weiye Road, Binjiang District, Hangzhou, Zhejiang 310053 (CN). **HOYDIS, Jakob**; 47 rue Sarrette, 75014 Paris (FR). **CHEN, Yong**; Room 503, Build 34, Qianjiangwan Huayuan, No. 99 Dongxin Road, Binjiang District, Hangzhou, Zhejiang 310053 (CN).

(74) Agent: **KING & WOOD MALLESONS**; 20th Floor, East Tower, World Financial Centre, No. 1 Dongsanhuan Zhonglu, Chaoyang District, Beijing 100020 (CN).

(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,

(54) Title: METHOD AND APPARATUS FOR SIGNAL DETECTION IN WIRELESS COMMUNICATION SYSTEM

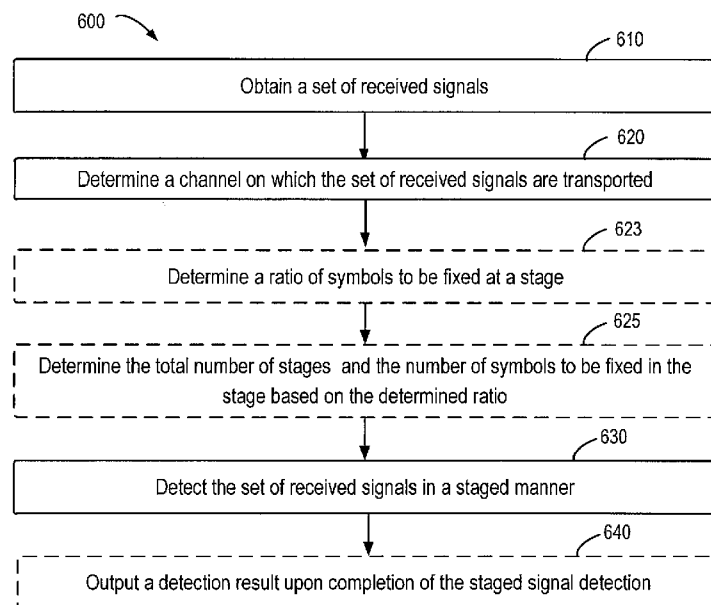


FIG. 6

(57) Abstract: Embodiments of the present disclosure relate to methods, apparatuses and computer program products for signal detection in a wireless communication system. A method implemented at a receiver device comprises: obtaining a set of received signals; determining a channel matrix via which the set of received signals have been transported; and detecting the set of received signals in a staged manner, wherein in a stage, the method comprises: detecting the set of received signals, based on the channel matrix and a detection algorithm for the stage; fixing one or more of detected symbols output from the detecting algorithm for the stage; and updating the channel matrix and the set of received signals for usage by a next stage, based on the one or more of the detected symbols that are fixed in the stage.

HR, HU, ID, IL, IN, IR, IS, JO, 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))

METHOD AND APPARATUS FOR SIGNAL DETECTION IN WIRELESS COMMUNICATION SYSTEM

FIELD

[0001] Non-limiting and example embodiments of the present disclosure generally relate to a technical field of wireless communication, and specifically to methods, apparatuses and computer program products for signal detection in a wireless communication system.

BACKGROUND

[0002] This section introduces aspects that may facilitate better understanding of the disclosure. Accordingly, the statements of this section are to be read in this light and are not to be understood as admissions about what is in the prior art or what is not in the prior art.

[0003] In wireless communication systems, there is always a requirement for high spectrum efficiency (SE) and capacity. Multiple input multiple output (MIMO) techniques have been known as an effective way for improving SE and capacity in modern wireless communication systems. For example, MIMO has been adopted as a key feature in Long Term Evolution (LTE)/LTE-Advanced (LTE-A) system developed by the third generation project partnership (3GPP). To further enhance SE and throughput, a concept of massive MIMO (mMIMO) has been proposed.

[0004] mMIMO means that a large antenna array is deployed in a wireless communication system. mMIMO has been considered as a key technology in, for example, the coming fifth generation (5G) system. The number of antennas at a 5G node B (also referred to as a gNodeB) may be more than 128. The large number of antennas promises substantial performance gains, but at the same time brings a challenge in signal detection.

SUMMARY

[0005] Various embodiments of the present disclosure mainly aim at providing methods, apparatuses and computer program products for signal detection in a wireless communication system.

[0006] In a first aspect of the disclosure, there is provided a method implemented at a receiver device. The method comprises obtaining a set of received signals; determining a channel on which the set of received signals are transported; and detecting a set of

transmitted signals from the set of received signals in an iterative manner based on the determined channel, a modulation mode for the set of transmitted signals, and the set of received signals, by using a gradient descent (GD) algorithm.

[0007] In an embodiment of the present disclosure, detecting the set of transmitted signals in an iterative manner comprises: in each iteration round, for each of the set of received signals: determining a gradient direction for updating a detected value for each of the set of transmitted signals, based on a detection error caused by the detected value and an associated channel element in the determined channel; determining a learning rate for updating a detected value for each of the set of transmitted signals; and updating the detected value for each of the set of transmitted signals by adding a product of the determined learning rate and the determined gradient direction to the detected value.

[0008] In a further embodiment, the gradient direction may be determined as: $(y_i - \sum_k h_{ik} * x_k) h_{ij}^*$, where y_i represents a signal received from the i th receiving antenna, x_k represents a signal transmitted from the k th transmitting antenna, $\sum$ represents a sum function, h_{ik} represents a channel element from the k th transmitting antenna to the i th receiving antenna, h_{ij} represents a channel element from the j th transmitting antenna to the i th receiving antenna, and h_{ij}^* represents a conjugate of h_{ij} , $i=1, \dots, N_r$, j and $k=1, \dots, N_t$, and N_r and N_t represent the number of receiving antennas and transmitting antennas respectively.

[0009] In another embodiment, determining the learning rate may comprise: determining the learning rate based on a product of an initial learning rate and an increasing factor, the initial learning rate and the increasing factor each being less than 1; or determining the learning rate based on a look-up table and at least one of: quality of the channel, the number of the transmitting antennas, and the number of the receiving antennas.

[0010] In some embodiments, determining the learning rate may further comprise: normalizing the learning rate by multiplying with $\frac{HRowLenSquare_i}{maxlenSquare}$ or $\frac{1}{maxlenSquare}$, where $HRowLenSquare_i$ represents a sum of squares of all channel elements in the i th row of the channel, and $maxlenSquare$ represents a maximum value of $HRowLenSquare_i$, where $i=1, \dots, N_r$ and N_r is the number of receiving antennas.

[0011] In an embodiment, detecting the set of transmitted signals in an iterative manner may comprise: in each iteration, if difference between a detected value of one of the set of transmitted signals and a modulation symbol associated with the modulation mode is less than a proximity threshold, setting the detected value of the one of the set of transmitted signals to the modulation symbol.

[0012] In another embodiment, the proximity threshold may depend on the modulation mode.

[0013] In some embodiments, detecting the set of transmitted signals in an iterative manner may comprise: determining a set of modulation symbols associated with the modulation mode; determining a maximum real part of the set of modulation symbols; determining a maximum imaginary part of the set of modulation symbols; and in each iteration: if a real part of a detected value for one of the set of transmitted signals exceeds the determined maximum real part, setting the real part to the determined maximum real part; and if an imaginary part of a detected value for one of the set of transmitted signals exceeds the determined maximum imaginary part, setting the imaginary part to the determined maximum imaginary part.

[0014] In another embodiment, detecting the set of transmitted signals in an iterative manner may comprise: ceasing the detection in response to: a predetermined number of iteration being met; or difference between detected values for the set of transmitted signals obtained in current iteration round and last iteration round being less than a convergence threshold.

[0015] In a second aspect of the disclosure, there is provided a receiver device. The receiver device comprises a processing circuitry and a memory, said memory containing instructions executable by said processing circuitry whereby said receiver device is operative to carry out a method according to the first aspect of the present disclosure.

[0016] In a third aspect of the disclosure, there is provided a computer program. The computer program comprises instructions which, when executed by at least one processing circuitry of a device, causes the device to carry out a method according to the first aspect of the present disclosure.

[0017] In a fourth aspect of the disclosure, there is provided a computer readable medium with computer program stored thereon which, when executed by at least one processor of a device, causes the device to carry out the method according to the first aspect of the present disclosure.

[0018] In a fifth aspect of the present disclosure, there is provided a method. The method comprises: obtaining a set of received signals; determining a channel matrix via which the set of received signals have been transported; and detecting the set of received signals in a staged manner, wherein in a stage, the method comprises: detecting the set of received signals, based on the channel matrix and a detection algorithm for the stage; fixing one or more of detected symbols output from the detecting algorithm for the stage; and

updating the channel matrix and the set of received signals for usage by a next stage, based on the one or more of the detected symbols that are fixed in the stage.

[0019] In a sixth aspect of the present disclosure, there is provided a receiver device. The device comprises at least one processor and at least one memory including computer program code. The at least one memory and the computer program code are configured to, with the at least one processor, cause the receiver device at least to perform a method according to the fifth aspect of the present disclosure.

[0020] In a seventh aspect of the disclosure, there is provided a computer program. The computer program comprises instructions which, when executed by an apparatus, causes the apparatus to carry out the method according to the fifth aspect of the present disclosure.

[0021] In an eighth aspect of the disclosure, there is provided a computer readable medium with a computer program stored thereon which, when executed by an apparatus, causes the apparatus to carry out the method of the sixth aspect of the present disclosure.

[0022] In a ninth aspect of the present disclosure, there is provided a receiver device. The receiver device comprises: means for obtaining a set of received signals; means for determining a channel matrix via which the set of received signals have been transported; and means for detecting the set of received signals in a staged manner, wherein in a stage, the means for detecting is configured to: detect the set of received signals, based on the channel matrix and a detection algorithm for the stage; fix one or more of detected symbols output from the detecting algorithm for the stage; and update the channel matrix and the set of received signals for usage by a next stage, based on the one or more of the detected symbols that are fixed in the stage.

BRIEF DESCRIPTION OF THE DRAWINGS

[0023] The above and other aspects, features, and benefits of various embodiments of the present disclosure will become more fully apparent from the following detailed description with reference to the accompanying drawings, in which like reference signs are used to designate like or equivalent elements. The drawings are illustrated for facilitating better understanding of the embodiments of the disclosure and are not necessarily drawn to scale, in which:

[0024] FIG. 1 illustrates an example wireless communication network in which embodiments of the present disclosure may be implemented;

[0025] FIG. 2 shows a model of a two layer artificial neuron network (ANN);

[0026] FIG. 3 shows a flowchart of a method at a receiver device for signal detection in a wireless communication system according to an embodiment of the present disclosure;

[0027] FIG. 4 shows performance comparison of signal detection method according to an embodiment of the present disclosure and a conventional method;

[0028] FIG. 5 illustrates a simplified block diagram of an apparatus that may be embodied as/in a network device or a terminal device.

[0029] FIG. 6 shows a flowchart of a method in a receiver device for staged signal detection according to an embodiment of the present disclosure;

[0030] FIG. 7 shows example operations to be performed in a stage during staged signal detection according to an embodiment of the present disclosure;

[0031] FIG. 8 shows example operations fixing one or more symbols in one stage during staged signal detection according to an embodiment of the present disclosure;

[0032] FIG. 9 shows a flowchart of another method in a receiver device for staged signal detection according to an embodiment of the present disclosure;

[0033] FIGs. 10-15 show simulation results for staged signal detection solutions according to embodiments of the present disclosure; and

[0034] FIG. 16 illustrates a simplified block diagram of another apparatus that may be embodied as/in a network device or a terminal device.

DETAILED DESCRIPTION

[0035] Hereinafter, the principle and spirit of the present disclosure will be described with reference to illustrative embodiments. It should be understood that all these embodiments are given merely for one skilled in the art to better understand and further practice the present disclosure, but not for limiting the scope of the present disclosure. For example, features illustrated or described as part of one embodiment may be used with another embodiment to yield still a further embodiment. In the interest of clarity, not all features of an actual implementation are described in this specification.

[0036] References in the specification to “one embodiment,” “an embodiment,” “an example embodiment,” and the like indicate that the embodiment described may include a particular feature, structure, or characteristic, but it is not necessary that every embodiment includes the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an embodiment, it is submitted that it is

within the knowledge of one skilled in the art to affect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described.

[0037] It shall be understood that although the terms “first” and “second” etc. may be used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first element could be termed a second element, and similarly, a second element could be termed a first element, without departing from the scope of example embodiments. As used herein, the term “and/or” includes any and all combinations of one or more of the listed terms.

[0038] The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of example embodiments. As used herein, the singular forms “a”, “an” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms “comprises”, “comprising”, “has”, “having”, “includes” and/or “including”, when used herein, specify the presence of stated features, elements, and/or components etc., but do not preclude the presence or addition of one or more other features, elements, components and/or combinations thereof.

[0039] As used in this application, the term “circuitry” may refer to one or more or all of the following:

- (a) hardware-only circuit implementations (such as implementations in only analog and/or digital circuitry) and
- (b) combinations of hardware circuits and software, such as (as applicable):
 - (i) a combination of analog and/or digital hardware circuit(s) with software/firmware and
 - (ii) any portions of hardware processor(s) with software (including digital signal processor(s)), software, and memory(ies) that work together to cause an apparatus, such as a mobile phone or server, to perform various functions) and
- (c) hardware circuit(s) and or processor(s), such as a microprocessor(s) or a portion of a microprocessor(s), that requires software (e.g., firmware) for operation, but the software may not be present when it is not needed for operation.

[0040] This definition of circuitry applies to all uses of this term in this application, including in any claims. As a further example, as used in this application, the term circuitry also covers an implementation of merely a hardware circuit or processor (or multiple processors) or portion of a hardware circuit or processor and its (or their) accompanying software and/or firmware. The term circuitry also covers, for example and if applicable to

the particular claim element, a baseband integrated circuit or processor integrated circuit for a mobile device or a similar integrated circuit in server, a cellular network device, or other computing or network device.

[0041] As used herein, the term “wireless communication network” refers to a network following any suitable wireless communication standards, such as New Radio (NR), Long Term Evolution (LTE), LTE-Advanced (LTE-A), Wideband Code Division Multiple Access (WCDMA), High-Speed Packet Access (HSPA), and so on. The “wireless communication network” may also be referred to as a “wireless communication system.” Furthermore, communications between network devices, between a network device and a terminal device, or between terminal devices in the wireless communication network may be performed according to any suitable communication protocol, including, but not limited to, Global System for Mobile Communications (GSM), Universal Mobile Telecommunications System (UMTS), Long Term Evolution (LTE), New Radio (NR), wireless local area network (WLAN) standards, such as the IEEE 802.11 standards, and/or any other appropriate wireless communication standard either currently known or to be developed in the future.

[0042] As used herein, the term “network device” refers to a node in a wireless communication network via which a terminal device accesses the network and receives services therefrom. The network device may refer to a base station (BS) or an access point (AP), for example, a node B (NodeB or NB), an evolved NodeB (eNodeB or eNB), a NR NB (also referred to as a gNB), a Remote Radio Unit (RRU), a radio header (RH), a remote radio head (RRH), a relay, a low power node such as a femto, a pico, and so forth, depending on the applied terminology and technology.

[0043] The term “terminal device” refers to any end device that may be capable of wireless communications. By way of example rather than limitation, a terminal device may also be referred to as a communication device, user equipment (UE), a Subscriber Station (SS), a Portable Subscriber Station, a Mobile Station (MS), or an Access Terminal (AT). The terminal device may include, but not limited to, a mobile phone, a cellular phone, a smart phone, voice over IP (VoIP) phones, wireless local loop phones, a tablet, a wearable terminal device, a personal digital assistant (PDA), portable computers, desktop computer, image capture terminal devices such as digital cameras, gaming terminal devices, music storage and playback appliances, vehicle-mounted wireless terminal devices, wireless endpoints, mobile stations, laptop-embedded equipment (LEE), laptop-mounted equipment (LME), USB dongles, smart devices, wireless customer-premises equipment (CPE) and the like. In the

following description, the terms “terminal device”, “communication device”, “terminal”, “user equipment” and “UE” may be used interchangeably.

[0044] As yet another example, in an Internet of Things (IOT) scenario, a terminal device may represent a machine or other device that performs monitoring and/or measurements, and transmits the results of such monitoring and/or measurements to another terminal device and/or network equipment. The terminal device may in this case be a machine-to-machine (M2M) device, which may in a 3GPP context be referred to as a machine-type communication (MTC) device. As one particular example, the terminal device may be a UE implementing the 3GPP narrow band internet of things (NB-IoT) standard. Examples of such machines or devices are sensors, metering devices such as power meters, industrial machinery, or home or personal appliances, for example refrigerators, televisions, personal wearables such as watches etc. In other scenarios, a terminal device may represent a vehicle or other equipment that is capable of monitoring and/or reporting on its operational status or other functions associated with its operation.

[0045] As used herein, a downlink (DL) transmission refers to a transmission from a network device to UE, and an uplink (UL) transmission refers to a transmission in an opposite direction. That is, in DL, the network device is a transmitter and the UE is a receiver device; while in UL, the UE is a transmitter and the network device is a receiver device.

[0046] FIG. 1 illustrates an example wireless communication network 100 in which embodiments of the present disclosure may be implemented. As shown, the wireless communication network 100 may include one or more network devices, for example, network device 101. The network device 101 may be in a form of a base station (BS), a Node B (NB), an evolved NB (eNB), a gNB, a virtual BS, a Base Transceiver Station (BTS), or a Base Station Subsystem (BSS), AP and the like.

[0047] In this example, network device 101 provides radio connectivity to a set of UEs 102-1, 102-2, and 102-3, which is collectively referred to as “UE(s) 102”, within its coverage. It should be appreciated that in some embodiments, the network device may provide service to less or more UEs and the number of UEs in this shown example does not suggest any limitations as to the scope of the present disclosure.

[0048] In some embodiments, a network device, e.g., the network device 101 in FIG. 1, may serve UEs, e.g., UE 102 in FIG. 1, within its coverage with a plurality of antennas. For example, the network device 101 may be equipped with a plurality of transmitting antennas, and/or a plurality of receiving antennas. Likewise, in some embodiments, some or all of the

UEs 102 may be equipped with a plurality of antennas for transmitting and/or receiving. In addition, a plurality of UEs 102 may be scheduled to transmit or receive in a same or overlapped time-frequency resource, thereby forming a multi-user MIMO (MU-MIMO) scheme. Therefore, in some scenarios, a receiver device (which may be a network device or a UE) may be required to detect signals transmitted from a plurality of antennas.

[0049] MIMO detection algorithms have been proposed, for example, in a book titled “Large MIMO systems” by A. Chockalingam and B. Sundar Rajan, and a book titled “Introduction to MIMO communications” by Jerry R. Hampton, both of which provide a detailed introduction about signal detection algorithms, such as zero forcing (ZF), ZF with interference cancellation (ZF-IC), Linear Minimum Mean Square Error (LMMSE), LMMSE with interference cancellation (LMMSE-IC), and the like. In addition, other signal detection algorithms like sphere decoding, probabilistic data association (PDA), approximate message passing (AMP), deep neuron network (DNN) based MIMO detection algorithm, MIMO detection based on Markov Chain Monte Carlo (MCMC) technology, etc. have been proposed recently. For example, a detailed introduction about the AMP algorithm for MIMO detection may be found, for example, through the link of <http://ieeexplore.ieee.org/document/7282651/>, while analysis of complexity for the AMP algorithm may be found in <https://emtiyaz.github.io/Writings/approxMP0.pdf>. It is stated that AMP is an iteration based method with a complexity of about $O(\max(mn^2, m^2n) \cdot \text{iteration})$, where m and n denote the number of transmitting and receiving antennas respectively. A semi-definite relaxation (SDR) algorithm which can be used for MIMO detection is introduced in a paper titled “Semidefinite relaxation of quadratic optimization problems” published at pages 20-34 of IEEE SIGNAL PROCESSING MAGAZINE May 2010, by Zhi-Quan Luo et. al., and according to which, the complexity of the SDR algorithm is at least $O(n^{3.5} \log(1/\epsilon))$, where ϵ represents solution precision. Details of the DNN based MIMO detection algorithm may be found in <https://arxiv.org/abs/1706.01151>.

[0050] It is not easy to devise a mMIMO detection algorithm that provides high detection accuracy with reasonable complexity, and inventors of the present disclosure have observed that all existing MIMO detection algorithms have a disadvantage of high computation complexity or functional limitation. For example, some of the algorithms above, like linear detection algorithms ZF and Minimum Mean Square Error (MMSE), etc., have low computing complexity but only achieve low accuracy. Some of the algorithms, like AMP and SDR etc., may achieve high accuracy but at a cost of high computing complexity.

Some detection algorithms require calculation of a covariance matrix of a MIMO channel matrix H , and/or an inverse of the channel matrix H . Furthermore, in most detection algorithms, computation complexity increases with the number of the receiving and transmitting antennas in an un-linear manner. This is undesirable, especially for a mMIMO system with a large number of antennas.

[0051] In addition, some signal detection algorithm does not work for high order quadrature amplitude modulation (QAM) cases. For example, though the DNN based MIMO detection algorithm achieves comparable performance to that of the AMP/SDR with significantly reduced computing complexity, it requires significant training resources and a Binary Phase Shift Keying (BPSK) modulation scheme for all of the transmitters. It is unclear currently whether it could be extended to higher order modulation schemes and how to support a possibly different modulation scheme for each transmitter.

[0052] To solve at least a part of the above problems, methods, apparatuses and computer program products have been proposed in the present disclosure.

[0053] In general, in a MIMO or mMIMO system with N_t transmitting antennas and N_r receiving antennas, a received signal may be expressed as Equation (1):

$$y = Hx + z \quad (1)$$

where y represents a complex vector with size N_r and represents the received signal (also referred to as a receiving signal) from N_r receiving antennas; H is a complex matrix with size N_r by N_t (i.e., N_r rows and N_t columns) and stands for a channel on which the signal y is transported, i.e., a channel passed/experienced by the signal y . H may be obtained via, for example, channel estimation. x is complex vector with size N_t and represents a transmitted signal from N_t transmitting antennas; z is a complex vector with size N_r and represents received noise on N_r receiving antennas.

[0054] The term x in equation (1) is the transmitted signal to be detected from the received signal y at the receiver device side, and in some embodiments of the present disclosure, this signal detection problem may be solved by using y , H and modulation information of x (e.g., a modulation scheme such as BPSK, QPSK, 16QAM, etc., for x) as inputs.

[0055] Generally speaking, according to some embodiments of the present disclosure, signal detection may be implemented based on a model of a simple two layer artificial neuron network (ANN) shown in FIG. 2. In particular, as shown in FIG. 2, with this ANN model, each row vector ($h_{i1}, h_{i2}, \dots, h_{iN_t}$) of the channel matrix H is considered as an input vector, the

received value y_i associated with a corresponding row of H is considered as an output, while vector $x=[x_1, x_2, \dots, x_{N_t}]$ is modelled as weights to be learned.

[0056] With this model, an activation function in the output layer may be represented as $f(x) = x$, and there are N_r data samples in total. In a wireless communication system (e.g., 5G) with mMIMO, the number of receiving antennas at a receiver device (e.g., a gNodeB) may be more than 128, and therefore, the number of data samples is enough to learn the weights x which is the transmitted symbols to be detected.

[0057] In some embodiments, a Gradient descent (GD) algorithm, which is conventionally used in machine learning and other related areas, is proposed to be used for solving the signal detection problem.

[0058] As an example rather than limitation, a loss function expressed with equation (2) may be used in the GD algorithm.

$$E(x) = \frac{1}{2} \sum_i |y_i - \sum_k h_{ik} * x_k|^2 \quad (2)$$

where $| \cdot |$ represents modulus of a complex value. The loss function in this example is a sum of error squares, and is a convex function, which guarantees that the GD algorithm will converge to an optimal solution.

[0059] By using the GD algorithm for detecting the transmitted signal x , computation complexity required is only $O(N_r * N_t)$. Moreover, better symbol error rate (SER) can be achieved compared with legacy MMSE MIMO detection algorithms. Further, since GD algorithm is based on a simple ANN model, developed ANN technologies (for example, hardware parallel processing) can be used directly. Additionally, embodiments of the present disclosure may provide a framework for building more complex multi-layer type ANN mMIMO detection schemes. In particular, due to unfolding property of each GD iteration, it enables to provide a flexible mMIMO detection solution. That is, it is easy to add new functions into each iteration operation as needed, to obtain more complex and advanced detection solutions.

[0060] FIG. 3 shows a flowchart of a method 300 implemented at a receiver device for signal detection in a wireless communication system. The receiver device may be, for example, the network device 101 or UE 102 shown in FIG. 1. For ease of discussion, the method 300 will be described below with reference to network device 101 and the communication network 100 illustrated in FIG. 1. However, embodiments of the present disclosure are not limited thereto.

[0061] At block 310, the network device 101 obtains a set of received signals. For example, the set of received signals may be received from N_r receiving antennas, and may be denoted as a vector y with a size N_r . Note that the set of received signals may be received from one or more UEs, for example, one or more UEs 102 in FIG. 1.

[0062] At block 320, the network device 101 determines a channel on which the set of received signals are transported, or in other words, a channel passed/experienced by the set of received signals. In a scenario of MIMO communication, the channel may be represented by and called a channel matrix H . It is to be understood that embodiments are not limited to any specific way for obtaining the channel matrix H . Just for illustration purpose, the channel matrix H may be known in advance or obtained via channel estimation. Any blind channel estimation algorithm or channel estimation algorithm based on pilots, reference signals, or training sequences may be used for this purpose. For ease of discussion, assuming there are N_t transmitting antennas and N_r receiving antennas, then the determined channel matrix H has a size of N_r by N_t , i.e., it has N_r rows and N_t columns.

[0063] At block 330, the network device 101 detects a set of transmitted signals x from the set of received signals y in an iterative manner, based on the determined channel H , a modulation mode for the set of transmitted signals x , and the set of received signals y , by using the GD algorithm.

[0064] In general, GD algorithms include batch gradient descent (BGD) algorithms and stochastic gradient descent (SGD) algorithms. With BGD, all data samples or a subset of the data samples are processed in a batch, then errors caused by detection are summed and used for updating the detection of the set of transmitted signals x . Since x may be modelled as weights in an ANN model as shown in FIG. 2, the signal detection of x is also referred to as weight updating herein. With SGD, GD weights updating is performed for each sample data directly.

[0065] SGD or BGD can be used at block 330. In some embodiments, SGD may be used, so as to obtain channel space diversity gains, and/or, to make the detection converge quickly. However, it should be appreciated that embodiments are not limited to BGD or SGD algorithms.

[0066] In some embodiments, at block 330, in each iteration round, the network device 101 may update the detection for the set of transmitted signals x . For example, in each iteration round, for each of the set of received signals (y_j , corresponding to the j th row of the channel matrix H), the network device 101 determines a gradient direction d for updating a detected value for each of the set of transmitted signals x_i . The gradient direction d may be

determined based on a detection error caused by the detected value and an associated channel element h_{ij} in the determined channel matrix H . Additionally, the network device 100 determines a learning rate R for updating a detected value for each of the set of transmitted signals, and updates the detected value for each of the set of transmitted signals by adding a product of the determined learning rate R and the determined gradient direction d to the detected value. That is, the network device 101 updates the detection for the transmitted signal x as:

$$x = x + d * R \quad (3).$$

[0067] That is, in each iteration round, the network device 101 updates the detection for each of the transmitted signals x , based on each row of the channel matrix H , respectively, and then proceeds to next iteration round. In the context of the present disclosure, each iteration round is also called an epoch, and each updating based on one row of the channel matrix is also called an iteration.

[0068] For instance, the network device 101 may determine the j th element d_j of the gradient direction d by equation (4):

$$d_j = (y_i - \sum_k h_{ik} * x_k) h_{ij}^* = \delta_i \cdot h_{ij}^* \quad (4)$$

where

$$\delta_i = y_i - \sum_k h_{ik} * x_k \quad (4a)$$

δ_i represents detection error, y_i represents a signal received from the i th receiving antenna, x_k represents a signal transmitted from the k th transmitting antenna, $\sum$ represents a sum function, h_{ik} represents a channel element from the k th transmitting antenna to the i th receiving antenna, h_{ij} represents a channel element from the j th transmitting antenna to the i th receiving antenna, and h_{ij}^* represents a conjugate of h_{ij} , $i=1, \dots, N_r$, j and $k=1, \dots, N_t$, and N_r and N_t represent the number of receiving antennas and transmitting antennas, respectively.

[0069] In an embodiment, the network device 101 may adopt a SGD algorithm at block 330, and use a loss function defined in equation (5):

$$E_i = \frac{1}{2} |y_i - \sum_k h_{ik} * x_k|^2 \quad (5)$$

where E_i represents a loss value corresponding to the i th row of the channel matrix H . Based on this loss function, the best gradient direction d may be obtained, for example, by calculating the differential of E_i for each x_j , as below

$$d_j = -\frac{\partial E_i}{\partial x_j} = (y_i - \sum_k h_{ik} * x_k) h_{ij}^* \quad (6)$$

However, it should be appreciated that embodiments are not limited to the specific way for obtaining the gradient direction d.

[0070] Alternatively, or in addition, in other embodiments, the network device 101 may determine the learning rate R based on a product of an initial learning rate η and an increasing factor γ , and both the initial learning rate η and the increasing factor γ is less than 1. For example rather than limitation, R may be determined as:

$$R = \eta * \gamma \quad (7).$$

[0071] In a further embodiment, the increasing factor γ may be specific to an iteration round (i.e., epoch), and in this case the increasing factor may be denoted as γ^{ep} . In other words, the increasing factor γ may be different for different iteration rounds. It enables to adjust the step size for updating the detection in each iteration round.

[0072] Alternatively, in another embodiment, the network device 101 may determine the learning rate based on a look-up table and at least one of: quality of the channel, the number of the transmitting antennas, and the number of the receiving antennas. For instance, the network device 101 may use a received signal to noise ratio (SNR), N_r , and N_t as inputs to acquire a suitable learning rate R from the lookup table.

[0073] In some embodiments, optimized learning rates may be obtained in advance, for example, by machine searching and stored in a loop-up table for later use. For illustration purpose, some learning rate obtained via computer searching for different modulation modes, SNR status and antenna configurations are listed in Table 1 to Table 4.

Table 1. Learning rate for 4QAM

SNR	8dB	9dB	10dB	11dB	12dB	13dB	14dB
$N_r=64$ $N_t=16$	0.3	0.4	0.4	0.6	0.7	0.7	0.7
$N_r=128$ $N_t=16$	0.3	0.4	0.4	0.6	0.7	0.7	0.7
$N_r=64$ $N_t=32$	0.6	0.7	0.7	0.7	0.7	0.7	0.7
$N_r=128$ $N_t=32$	0.3	0.4	0.4	0.6	0.7	0.7	0.7

Table 2. Learning rate for 16QAM

SNR	8dB	9dB	10dB	11dB	12dB	13dB	14dB
$N_r=64$ $N_t=16$	0.3	0.3	0.3	0.3	0.4	0.4	0.4
$N_r=128$ $N_t=16$	0.2	0.2	0.2	0.2	0.2	0.2	0.4
$N_r=64$ $N_t=32$	0.6	0.7	0.7	0.7	0.7	0.7	0.7
$N_r=128$ $N_t=32$	0.3	0.3	0.3	0.3	0.3	0.3	0.3

Table 3. Learning rate for 64QAM with low SNR

SNR	8dB	9dB	10dB	11dB	12dB	13dB	14dB
Nr= 64 Nt=16	0.3	0.3	0.3	0.3	0.3	0.4	0.4
Nr=128 Nt= 16	0.2	0.2	0.2	0.2	0.2	0.2	0.2
Nr= 64 Nt=32	0.6	0.6	0.6	0.7	0.7	0.7	0.7
Nr 128 Nt=32	0.3	0.3	0.3	0.3	0.3	0.4	0.4

Table 4. Learning rate for 64QAM with high SNR

SNR	16dB	20dB	24dB	28dB	32dB	36dB	40dB
Nr= 64 Nt=16	0.4	0.4	0.7	0.7	0.7	0.7	0.7
Nr=128 Nt= 16	0.2	0.4	0.7	0.7	0.7	0.7	0.7
Nr= 64 Nt=32	0.7	0.7	0.7	0.7	0.7	0.7	0.7
Nr 128 Nt=32	0.4	0.4	0.7	0.7	0.7	0.7	0.7

[0074] Parameters for computer simulation to search for the above learning rates are listed in Table 5 and Table 6. In addition, during the simulation, γ in equation (7) is set to 1, and each channel element in the channel matrix H is generated according to a normal distribution $N(0,1)$, then normalized by $H=H/\sqrt{N_t}$. Noise Z in equation (1) is generated according to a normal distribution $N(0,1)$, then normalized by $Z=Z/(10^{(SNR/10)})$. The transmitted signals x is normalized by $x=x/\sqrt{\text{var}(x)}$, where $\text{var}(x)$ means the variance of symbol x .

Table 5. 504 cases for low SNR

Parameters	Values
Learning rate	0.2 ~ 0.7
Number of transmitted symbols Nt	16; 32
Number of received antenna Nr	64; 128
QAM	4QAM; 16QAM; 64QAM
SNR	8dB ~ 14dB
Number of Epoch	5

Table 6. 168 cases for high SNR

Parameters	Values
Learning rate	0.2 ~ 0.7

number of transmitted symbols Nt	16; 32
number of received antenna Nr	64; 128
QAM	64QAM
SNR	16dB; 20dB; 24dB; 28dB; 32dB; 36dB; 40dB
Number of Epoch	5

[0075] In some embodiments, it is also possible to use lower learning rate value while setting bigger epoch round for GD updating.

[0076] Furthermore, to obtain a better convergence property, in some embodiments, the determined learning rate R may be further normalized by the network device 101. As an example rather than limitation, the network device 101 may normalize the learning rate R by multiplying it with a factor $\frac{HRowLenSquare_i}{maxlenSquare}$ or $\frac{1}{maxlenSquare}$, where $HRowLenSquare_i$ represents a sum of squares of all channel elements in the i th row of the channel matrix i.e., $HRowLenSquare_i = \sum_j |h_{ij}|^2$. $maxlenSquare$ represents a maximum value of $HRowLenSquare_i$, i.e., $maxlenSquare = \max_i \{HRowLenSquare_i\}$, where $i=1, \dots, Nr$ and Nr is the number of receiving antennas.

[0077] Accordingly, in some embodiments, detection for the j th transmitted signal x_j may be updated using equation (8) or (9):

$$x_j = x_j + \frac{HRowLenSquare_i * rate * \delta_i * h_{ij}^*}{maxlenSquare} \quad (8)$$

$$x_j = x_j + \frac{rate * \delta_i * h_{ij}^*}{maxlenSquare} \quad (9).$$

For instance, if $maxlenSquare \leq 1$, x_j may be update based on (8) or (9), and if $maxlenSquare > 1$, x_j may be updated based on (9).

[0078] The above factor $\frac{HRowLenSquare_i}{maxlenSquare}$ and $\frac{1}{maxlenSquare}$ provide a way to adjust the learning rate by normalization, thereby ensuring the convergence of the iteration.

[0079] Alternatively, or in addition, in some embodiments, in order to further improve the convergence property of the detection algorithm, at block 330, if a detected value for one transmitted signal x_j is already close enough to one modulation value associated with the modulation mode of the transmitted signal (e.g., 64QAM, or QPSK), the network device 101 may set the detection value for the transmitted signal x_j as that modulation value. This helps to achieve quick convergence and avoid learning overfit.

[0080] For example, at block 330, in each iteration, if difference between a detected value of one of the set of transmitted signals x_j and a modulation symbol (denoted as mv) associated with the modulation mode is less than a proximity threshold T_{prox} , the network device 101 may set/anchor the detected value of the one of the set of transmitted signals to the modulation symbol mv . That is:

if $|mv - x_j| < T_{prox}$, then set $x_j = mv$,

where x_j represents the j th transmitted signal. Such operation is also referred to as an anchoring operation thereafter.

[0081] In some embodiments, the proximity threshold T_{prox} may depend on the modulation mode. For example, the proximity threshold T_{prox} may be different for QPSK and 16QAM modulation modes. For illustration rather than limitation, in some embodiments, the proximity threshold T_{prox} may be determined as:

$$T_{prox} = anRate * stepValue \quad (10)$$

where $anRate$ represents an initial anchor rate, which may be set to be, for example but not limited to, 0.1, while the $stepValue$ represents an adaptive factor which may depend on the modulation mode for the transmitted signals. For instance, the stepvalue for a modulation mode may be set to the nearest distance between modulation symbols associated with the modulation mode. For QPSK, values of the modulation symbols may be expressed as:

$$SMV = \{x + yj\} \text{ for } x, y \in \{-1, 1\} \quad (11)$$

In this example, the nearest distance between these symbols is 2. As a result, $stepValue$ for QPSK may be set to 2. Likewise, for 16QAM, values for the modulation symbols may be expressed as:

$$SMV = \{x + yj\} \text{ for } x, y \in \left\{-1, -\frac{1}{3}, \frac{1}{3}, 1\right\} \quad (12)$$

In this case, the nearest distance between these symbols is $2/3$, so $stepValue$ for 16QAM may be set to $2/3$. Alternatively, for 16QAM, values for the modulation symbols may be expressed as:

$$SMV = \{x + yj\} \text{ for } x, y \in \{-3, -1, 1, 3\} \quad (12')$$

In this case, the nearest distance between these symbols is 2, so $stepValue$ for 16QAM may also be set to 2.

[0082] Alternatively, or in addition, in some embodiments, to achieve quick convergence, at block 330, the network device 101 may put a constraint on the detected value for the transmitted signal x . For example, the network device 101 may determine a set of modulation symbols associated with the modulation mode (for example, using equation (11)

or (12) for QPSK and 16QAM respectively). For the set of modulation symbols, the network device 101 determines a maximum real part and a maximum imaginary part. The maximum real part and the maximum imaginary part serve as a limit for the real part and imaginary part of the detected signal respectively. That is, in each iteration, if a real part of a detected value for one of the set of transmitted signals (e.g., the j th transmitted signal x_j) exceeds the determined maximum real part, the network device 101 sets the real part to the determined maximum real part. Likewise, if an imaginary part of a detected value for one of the set of transmitted signals exceeds the determined maximum imaginary part, the network device 101 sets the imaginary part to the determined maximum imaginary part.

[0083] As an example rather than limitation, computer instructions for an anchor function (denoted as anchor ()) for implementing the above optional anchoring operations for improving the convergence property may be constructed as the following:

Initialization:

set anchor rate: anRate. For example, anRate=0.1

calculate the stepvalue for SMV, which is the symbol neighbour's nearest distance. for example: for QPSK, stepvalue=2; For 16QAM, stepvalue=2.

anchor() main routine:

for j=0 to Nt:

#real(x) to get real value of complex x; imag(x) to get imag value of complex x.

#real(SMV) is all SMV element's real value composed set.

#imag(SMV) is all SMV element's imag value composed set.

#x value cannot be bigger or less than the biggest or lest real or imag value in SMV.

if real(x_j)>max(real(SMV)):

real(x_j)=max(real(SMV))

if real(x_j)<min(real(SMV)):

real(x_j)=min(real(SMV))

if imag(x_j)>max(imag(SMV)):

imag(x_j)=max(imag(SMV))

if imag(x_j)<min(imag(SMV)):

imag(x_j)=min(imag(SMV))

find the nearest modulation value mv for x_j : $mv = \operatorname{argmin}_{mv \in SMV} |mv - x_j|$

```

if  $|mv - x_j| < anRate * stepValue$ , set  $x_j = mv$ 
#end of loop
output the anchored weights
#anchor() function ended

```

[0084] The anchoring operation may bring several benefits, such as quick convergence, and ANN overfit avoidance. For instance, if all the weights (i.e., transmitted signals x) can be anchored, the GD based iteration operation can be stopped immediately. In addition, in some embodiments, the anchoring operation also helps to improve mMIMO detection accuracy, e.g., symbol error rate may be reduced, since noise caused by the transmitted signal being anchored is removed.

[0085] The detection at block 330 may be terminated in response to various conditions. For example, in some embodiments, the network device 101 may cease the detection at block 330 if a predetermined number of iteration rounds (epoch) has been reached. This way of ending the detection may be referred to as EC1 hereafter.

[0086] Alternatively, in some embodiments, the difference between detected values for the set of transmitted signals obtained in current iteration round and last iteration round may be determined. If the difference falls below a convergence threshold, the detection may be terminated. For example, the network device may cease the update if a change in the transmitted signal x caused by updating is less than a convergence threshold ϵ , i.e.,

$$|x_{cur} - x_{pre}|^2 < \epsilon \quad (13)$$

where ϵ represents a pre-defined convergent threshold; x_{cur} represents detected value for x in current iteration round, and x_{pre} represents detected value for x in last iteration round. This way for ending the iteration round may be referred to as EC2 hereafter.

In some embodiments, the iteration/updating may be ceased by using a hybrid method of EC1 and EC2. That is, the network device 101 may end the updating if predetermined number of iteration rounds has been finished, or if the change in x is already less than ϵ shown in Equation (12). The hybrid way for stopping the updating may be referred to as EC3.

[0087] For illustration purpose and without limitation, computer procedures for implementing GD based signal detection with method EC3 for ending the update may be constructed as below:

Input:

Channel estimation H , which is $Nr * Nt$ matrix.

Received signal y , which is $N_r \times 1$ matrix

Modulation mode of transmitted symbol x , here each symbol x can use different modulation mode.

Output:

Decoded symbol x , which is $N_t \times 1$ matrix

Initialization:

Set GD end condition: epoch value based; threshold value based; or both epoch value and threshold value based. Set epoch value and ϵ value.

Set initial weights value $x=0$.

Set learning rate method:

Method 1: based on initial learning rate value η and learning rate epoch level decreasing rate γ , both two value shall be less than 1.

Method 2: based on pre-defined lookup table, the pre-defined table can be built based on the experience or testing.

#anchor() function is not mandately, but it's helpful for avoiding overfit

Set anchor rate value: anRate, for exmaple 0.1. this parameter is used in anchor() function

#optional initialization part is not mandately, but it's helpful for quick convergent.

Optional initialization:

Calculate H matrix's row length square: HRowLenSquare, which is $N_r \times 1$ matrix

Calculate maximum H row length square: maxlenSquare= $\max(\text{HRowLenSquare})$

H row order can be adjusted, for example, based on HRowLenSquare descently

GD based mMIMO main routine:

For $ep=0$ to epoch:

#this step is to set learning rate, two ways are suggested:

Set learning rate for this epoch: $\text{rate}=\eta \cdot \gamma^{ep}$ or rate is selected based on pre-defined lookup table

#if anchor() function is enabled, call it one time in each epoch round:

Call anchor()

#if anchor() find the learned weights are already good enough, quit immediately.

If all weights can be anchored:

quit the epoch loop

#start doing GD updating for each row of H :

For i=0 to Nr:

For i-th H row, calculate the error based on (4a):

For j=0 to Nt:

If optional initialization used:

If maxlenSquare $\leq$ 1, weights update can be based on (8) or (9); If maxlenSquare $>$ 1, weights update can be based on (9);

If optional initialization is not used, do the weights updating based on (3)-(4a);

If all H's rows' GD updating is finished in this epoch round, check the end condition:

If equation (13) is true, GD updating is finished, quit the epoch loop

#end of the GD updating loop

#end of the epoch loop

Output the decoded weights:

If output is softbits, just output the learned weights

If output is hardbit, find the nearest symbol value from SMV for each weight as output

#GD based mMIMO detection algorithm ended

[0088] To verify advantage of the proposed detection solution, performance of some embodiments of GD based signal detection methods have been compared with a conventional MMSE method, and results of the comparison are shown in FIG. 4.

[0089] During the comparison, end condition EC1 is adopted, and the number of iteration rounds (epoch) is set to 10. In addition, the learning rate R is set to a constant value of 0.3 for all different SNR and all epoch round, and its normalization is based on $1/\text{maxlenSquare}$. Anchoring operation is used and in which the anRate in equation (10) is set to 0.1. For each SNR case, 30000 data samples are used to obtain the result.

[0090] The results show that GD based detection is better than MMSE based detection in 4QAM case in terms of SER performance. In some embodiments, the performance improvement is guaranteed by the loss function which is a convex function and ensures convergence of the GD based updating. Furthermore, the proposed GD based detection solution does not involve any channel matrix transformation (for example, no need for calculating covariance matrix or pseudo inversion of the channel matrix H), and as a result, all channel spatial diversity properties are kept. This feature also contributes the better performance.

[0091] As shown in FIG. 4, for 16QAM and 64QAM cases, GD based detection and MMSE based detection achieve almost the same performance. However, GD-based

detection requires less computation complexity. In particular, the computation complexity of GD based detection is only $O(N_r \cdot N_t)$. During the detection, iteration can be stopped if anchoring operation finds that weights x are already close enough to expected symbols, therefore average number of iteration round needed is small.

[0092] In addition, the proposed detection method is based on the design of ANN, which means that developed ANN technologies can be reused in the detection. For example, it can support parallel processing in ANN supported hardware.

[0093] The proposed detection solution is also flexible. It is easy to add more advanced functions into each iteration or iteration round. For example, in some embodiments, anchoring operation and/or normalization operation for the learning rate may be added. This makes it possible for the solution to benefit from further development.

[0094] It should also be noted that embodiments of the present disclosure can be applied widely in various scenarios. For example, some embodiments may be used for complex mMIMO signal detection, and support different modulations, different SNRs, and different antenna numbers, etc.

[0095] FIG. 5 illustrates a simplified block diagram of an apparatus 500 that may be embodied in/as a network device, for example, the network device 101 shown in FIG. 1, or embodied in/as a terminal device, for example, the terminal device 102 shown in FIG. 1.

[0096] As shown by the example of FIG. 5, apparatus 500 comprises a processor 510 which controls operations and functions of apparatus 500. For example, in some embodiments, the processor 510 may implement various operations by means of instructions 530 stored in a memory 520 coupled thereto. The memory 520 may be any suitable type adapted to local technical environment and may be implemented using any suitable data storage technology, such as semiconductor based memory terminal devices, magnetic memory terminal devices and systems, optical memory terminal devices and systems, fixed memory and removable memory, as non-limiting examples. Though only one memory unit is shown in FIG. 5, a plurality of physically different memory units may exist in apparatus 500.

[0097] The processor 510 may be any proper type adapted to local technical environment, and may include one or more of general purpose computers, special purpose computers, microprocessors, digital signal processors DSPs and processors based on multicore processor architecture, as non-limiting examples. The apparatus 500 may also comprise a plurality of processors 510.

[0098] The processors 510 may also be coupled with a transceiver 540 which enables reception and transmission of information by means of one or more antennae 550 and/or other components. For example, the processor 510 and the memory 520 can operate in cooperation to implement method 300 described with reference to FIG.3. It shall be appreciated that all the features described above with reference to FIG. 3 also apply to apparatus 500, and therefore will not be detailed here.

[0099] Various embodiments of the present disclosure may be implemented by a computer program or a computer program product executable by one or more of the processors (for example processor 510 in FIG. 5), software, firmware, hardware or in a combination thereof.

[00100] Although some of the above description is made in the context of a wireless communication system shown in FIG. 1, it should not be construed as limiting the spirit and scope of the present disclosure. The principle and concept of the present disclosure may be more generally applicable to other scenarios.

[00101] In another aspect of the present disclosure, a staged signal detection framework and related methods, apparatuses and computer program products are proposed. Compared with conventional solutions, for example a VBLAST algorithm (https://en.wikipedia.org/wiki/Bell_Laboratories_Layered_Space-Time) and a detection algorithm disclosed in a US patent application US20040242179A (<http://pdfstore.patentorder.com/pdf/us/179/us2004242179.pdf>) which removes matrix inversion operations from the detection algorithm, some embodiments of the present disclosure provide more flexibility to achieve a tradeoff between performance and complexity, and achieves better performance.

[00102] In some embodiments, by using the staged signal detection framework which may also be referred to as a staged symbol fix (SSF) framework, a low complexity and low accuracy (m)MIMO detection algorithm may be transformed into a low complexity and high accuracy (m)MIMO detection algorithm. That is, some embodiments improve detection accuracy while keeping computation complexity at a low level.

[00103] In some embodiments, the proposed staged signal detection solution provides flexibility to choose any type of detection algorithm/detector for each processing stage independently. Furthermore, embodiments of the present disclosure are not limited to any specific detection algorithm/detector used in each stage. That is, an existing mMIMO detection algorithm may be used in a stage of the SSF framework to improve detection accuracy at a cost of slightly increased complexity. In some embodiments, the only

requirement for a mMIMO detection algorithm chosen for a stage is that it provides soft symbols as outputs, i.e., the detection algorithm outputs a probability estimation rather than a hard decision for an individual symbol.

[00104] For illustration rather than limitation, a conventional linear ZF detector, a linear MMSE detector, or a GD based mMIMO detection algorithm using a developed ANN technology proposed herein may be used at a stage of the proposed SSF framework.

[00105] In some embodiments, the staged signal detection solution allows choosing the total number of stages for the staged signal detection solution flexibly, and the total number of stages may be significantly smaller than the number of transmitting antennas N_t .

[00106] FIG. 6 shows a flowchart of a method 600 implemented at a receiver device for staged signal detection in a wireless communication system according to an embodiment of the present disclosure. The receiver device may be, for example, the network device 101 or terminal device 102 shown in FIG. 1. For ease of discussion, the method 600 will be described below with reference to network device 101 and the communication network 100 illustrated in FIG. 1. However, embodiments of the present disclosure are not limited thereto.

[00107] As shown in FIG. 6, at block 610, network device 101 obtains a set of received signals. For example, the set of received signals may be received from N_r receiving antennas, and may be denoted as a vector y with a size N_r . Note that the set of received signals may be received from one or more terminal devices, for example, one or more terminal devices 102 in FIG. 1.

[00108] At block 620, network device 101 determines a channel via which the set of received signals are transported, or in other words, a channel passed/experienced by the set of received signals. In a scenario of MIMO communication, the channel may be represented by and called a channel matrix H . It is to be understood that embodiments are not limited to any specific way for obtaining the channel matrix H . Just for illustration purpose, the channel matrix H may be known in advance or obtained via channel estimation. Any blind channel estimation algorithm or channel estimation algorithm based on pilots, reference signals, or training sequences may be used for this purpose. For ease of discussion, assuming there are N_t transmitting antennas and N_r receiving antennas, then the determined channel matrix H has a size of N_r by N_t , i.e., it has N_r rows and N_t columns.

[00109] At block 630, network device 101 detects the set of received signals y in a staged manner. The staged signal detection at block 630 comprises one or more processing

stages, and the total number of stages may be denoted as N_{tot} herein, where N_{tot} is an integer equal to or larger than 1.

[00110] In some embodiments, in a stage of the totally N_{tot} stages, the network device 101 may perform operations shown in FIG. 7 which illustrate an example for operations that may be performed in one processing stage.

[00111] In the example shown in FIG. 7, at block 631, the network device 101 detects the set of received signals y , based on the channel matrix H and a detection algorithm for the stage. Note that detecting the set of received signals y means estimating a set of transmitted signals x from the set of received signals y . The staged signal detection method proposed herein allows selecting a detection algorithm for each stage independently. That is to say, same or different detection algorithms may be used in different stages.

[00112] For illustration rather than limitation, the detection algorithm for one stage may be, but not limited to, one of: a ZF based algorithm, a MMSE based algorithm and a GD based algorithm (e.g., a SGD based algorithm) proposed in the present disclosure. Details for the ZF or MMSE based detection algorithm may be found, for example, in a book titled "Large MIMO systems" by A. Chockalingam and B. Sundar Rajan, or a book titled "Introduction to MIMO communications" by Jerry R. Hampton.

[00113] At block 632, the network device 101 fixes one or more of detected symbols output from the detecting algorithm for the stage. In other words, values for the one or more of the detected symbols are fixed/determined. In some embodiments, the network device 101 may fix the one or more detected symbols based on modulation information (e.g., a modulation scheme/mode such as BPSK or QPSK, or 16QAM) of the received signals. An example implementation for the fixing operation performed in block 632 is illustrated in FIG. 8.

[00114] As shown in FIG.8, at block 810, the network device 101 may select, for each detected symbol output from the detection algorithm for the current stage, a candidate symbol from a set of modulation constellation symbols determined by a modulation scheme (e.g., QPSK) for the set of received signals, for example based on a maximum likelihood (ML) criterion. For instance, a set of modulation constellation symbols A_{QPSK} for a QPSK modulation scheme includes 4 constellation symbols A_1 to A_4 , i.e., $A_{QPSK}=\{A_1, A_2, A_3, A_4\}$, and if a detected symbol has a minimum Euclidean distance with the constellation symbol A_1 , the receiver device 102 may select A_1 as the candidate symbol for the detected symbol.

Candidate symbols for other detected symbols output from the detection algorithm may be selected likewise.

[00115] At block 820, the network device 101 determines a distance between each detected symbol and its corresponding candidate symbol. For example, if there are M detected symbols S_i , $i=1, 2, \dots, M$, output from the detection algorithm at a stage, and a candidate symbol C_i with the minimum distance with a detected symbol S_i is selected for the detected symbol S_i , then at block 820, the network device 101 may determine a distance d_i between each S_i and C_i , where $i=1, 2, \dots, M$, and C_i is selected from a set of constellation symbols, for example $A_{QPSK}=\{A1, A2, A3, A4\}$.

[00116] At block 830, the network device 101 fixes one or more detected symbols associated with the minimum distance to a corresponding candidate symbol respectively. As an example, the receiver device may sort the distance d_i in an ascending order, and fix the first L detected symbols associated with the first L minimum distance to their corresponding candidate symbols.

[00117] It should be appreciated FIG. 8 just provide an example for the fixing operation performed at block 632 of FIG. 7, and the fixing operation is not limited to the exact example shown.

[00118] In some embodiments, the fixing operation performed at block 632 of FIG. 7 may be implemented via a function which may be denoted as $\text{Fix}()$ herein. Note that if the detected symbols output from the detection algorithm are complex values, the $\text{Fix}()$ function may process each detected symbol with a complex value directly, or process real and imaginary parts of the complex value separately. For the former case, a two dimensional (2D) Euclidean distance may be used for selecting the nearest candidate symbol for a detected symbol, while a one dimensional (1D) Euclidean distance may be used for the latter case.

[00119] For illustration rather than limitation, an example for constructing the $\text{Fix}()$ function is described below, where separate processing of the real and imaginary parts of the detected symbols is assumed, and correspondingly values of real part of constellation symbols are used for calculating a distance between a detected symbol and a candidate symbol.

Fix () function description:

Input:

ds: a list of detected symbols, each item of which includes an index and a value for a detected symbol. For example, the i th item is represented as $(index_i, dv_{index_i})$. In the *Fix()* function, these symbols' values will be checked, and values of some symbols will be fixed in a stage.

SMV: a set of values for constellation symbols.

Output:

fs: a list of fixed symbols and their corresponding values

Initialization:

(i) Determine a *SMV* to use. For example,

For QPSK,

$$SMV = \{-1, 1\} \quad (14)$$

For 16QAM

$$SMV = \{-3, -1, 1, 3\} \quad (15)$$

(ii) Set a value for *fNum* which represents the number of symbols to be fixed in a stage. As an example rather than limitation, the value for *fNum* may be determined/set based on a ratio of symbols to be fixed in a stage. Or, the value for *fNum* may decrease exponentially with the stage. For instance, the value for *fNum* may be determined/set as:

$$fNum = \max(1, \lceil \text{len}(ds) * fRatio \rceil) \quad (16)$$

or,

$$fNum = \lceil \text{len}(ds) * fRatio \rceil \quad (17)$$

where $\text{len}(ds)$ means the total number of detected symbols output from the detection algorithm for a current stage, *fRatio* represents a ratio of symbols to be fixed in the current stage, which may take a value between 0 and 1, and $\lceil \cdot \rceil$ denotes an operation of getting an integer part of a real number.

Fix() main routine:

(i) For each detected symbol, find a candidate symbol which is the nearest modulation constellation symbol value from the *SMV*. For example, for each item (e.g., the i th item $(index_i, dv_{index_i})$) in *ds*, find its nearest modulation value by:

$$mv_{index_i} = \underset{x \in SMV}{\operatorname{argmin}} |x - dv_{index_i}| \quad (18).$$

(ii) Build a minimum distance list (denoted as *minDis* herein), in which the i th item is obtained by:

$$\text{minDis}_i = |mv_{index_i} - dv_{index_i}| \quad (19)$$

(iii) *Select detected symbols to be fixed. For instance, order items in inDis in an ascending order, and select detected symbols associated with the first fNum elements in the sorted minDis as symbols to be fixed. Then the detected symbols fixed in a stage may be represented by a list fs, the ith item of which may be represented as:*

$$fs_i = (index_i, mv_{index_i}), i=1, \dots, fNum \quad (20)$$

(iv) *Output fs.*

(v) End of Fix() function.

[00120] In the example above, the Fix() function fixes/determines a value for one or more detected symbols based on a Euclidean distance based criterion, and the number of the one or more detected symbols is determined by fNum which is configurable and may be stage specific. It should be appreciated, however, embodiments are not limited to the specific way described above for fixing the detected symbols. In another embodiment, a different criterion may be used, and/or, the value of fNum may be constant for each stage.

[00121] Once a detected symbol is fixed, its detection is finished, and the detected symbol will not participate in later processing stages anymore. In some embodiments of the present disclosure, the detected symbols that are fixed in one stage may be excluded from participating in later processing stages by updating the channel matrix and the set of received signals, as shown in block 633 of FIG. 7.

[00122] At block 633, the network device 101 updates the channel matrix H and the set of received signals y for usage by a next stage, based on the one or more of the detected symbols that are fixed in the current stage. With the updating operation, impacts of the fixed symbols may be removed from the channel matrix and the set of received signals. For illustration rather than limitation, at block 633, the network device 101 may update the channel matrix H by removing, from the channel matrix H, one or more columns corresponding to the one or more of the detected symbols that are fixed in the current stage. In addition, the network device 101 may update the set of received signals y by subtracting from y a value contributed by the one or more of the detected symbols that are fixed in the current stage.

[00123] In some embodiments, the updating operation may be performed/controlled by a function which may be denoted as update () for simplicity. For illustration purpose only, an example for constructing the update () function is provided below in a form similar to computer instructions:

Update() function description:

Input:

fs: the list of fixed symbols output from the fixing operation, for example the *Fix()* function.

H: the channel matrix.

y: a vector of the received signals.

Output:

H: updated *H*, by removing columns of corresponding fixed symbols;

y: updated vector *y*, by subtracting fixed symbols' value;

Update() main routine:

#subtract fixed symbols' value from *y*:

for $i=0$ to $\text{len}(fs)$:

$k = \text{index}_i$

for $j=0$ to N_r :

$$y_j = y_j - h_{jk} * mv_k \quad (21)$$

#remove corresponding column in *H* matrix:

For each index_i of *fs*, remove its corresponding column (i.e., the index_i -th column) from *H* matrix.

Output *H* and *y*.

end of Update() function.

[00124] It should be appreciated that FIG. 7 and FIG. 8 only illustrate example operations that may be performed in a single processing stage. The method 600 shown in FIG. 6 may comprise a plurality of processing stages at block 630. In such a case, each stage may be performed in a way similar to that described with reference to FIGs. 7 and 8.

[00125] In some embodiments, the total number of stages N_{tot} involved in block 630 of method 600 may be configurable. As an example rather than limitation, the network device 101 may firstly determine a ratio of symbols to be fixed at a stage (which may be denoted as *fRatio* herein) at block 623 in FIG. 6, and then determine the total number of stages N_{tot} and the number of symbols (denoted as *fNum* herein) to be fixed in the stage based on the determined ratio (i.e., *fRatio*) at block 625 in FIG. 6.

[00126] For instance, the total number of stages N_{tot} may be determined based on *fRatio* by:

$$N_{\text{tot}} = \lceil -\log_{(1-f\text{Ratio})} N_t \rceil + 1 \quad (22)$$

where N_t stands for the number of transmitting antennas. It should be appreciated that embodiments of the present disclosure are not limited to any specific way for determining

values for $fRatio$, $fNum$, and N_{tot} . In some embodiments, the ratio of symbols to be fixed in a stage may be determined based on a receiving signal to noise ratio (SNR). For example, if the receiving SNR is high, a large number of detected symbols may be fixed in one stage; otherwise, a small number of detected symbols may be fixed in one stage.

[00127] Normally, the total number of stages N_{tot} required for the staged signal detection is far less than the number of detected symbols, for example, when $fNum$ is determined based on equation (16) or (17) which results in an exponential decrease in the number of symbols to be fixed in one stage.

[00128] The staged signal detection ends when the number of stages reaches the configured total number of stages N_{tot} , or all symbols are fixed. In some embodiments, the network device 101 outputs a detection result upon completion of the staged signal detection at block 640 in FIG. 6. In some embodiments, the detection result may only include symbols which are fixed during the stages of detection, which means that all detected signals have been fixed during the staged detection. In some embodiments, the detection result may include both fixed symbols and un-fixed symbols, which means that only a portion of detected signals have been fixed during the staged detection. The un-fixed symbols may be recovered via following processing, for example channel decoding operation.

[00129] Alternatively, in some embodiments, the network device 101 may output, at the end of a stage, the one or more symbols that are fixed in the stage, at block 634 in FIG. 4. In addition, if there is still one or more symbols remain unfixed after completion of the staged signal detection, the network device 101 also outputs the remaining symbol at block 640 in FIG. 6.

[00130] For illustration purpose, another example of the staged signal detection operation is shown in FIG. 9. This example may be considered as a specific implementation of blocks 630 and 640 in FIG. 6. The method shown in FIG. 9 may be implemented, for example, by the network 101 or the terminal device 102 in FIG. 1.

[00131] In the example of FIG. 9, the staged signal detection is implemented via a function denotes as $ssfMDA()$, which comprises several sub-functions including $MDA()$, $Fix()$ and $Update()$. The sub-function $MDA()$ is a function for signal detection in a stage of the SSF framework using a selected MIMO detection algorithm. It should be appreciated that the detection algorithm may be selected for each stage independently. Sub-function $Fix()$ is for fixing one or more (e.g., $fNum$) detected symbols output from $MDA()$ to corresponding modulated constellation symbol values with the least distance from the one or

more detected symbols. The fixed symbols will not participate in later processing stages. For example, the fixed symbols will not be further detected using the MDA() function in next stage. Since some example implementations of the Fix() function have been described above, details will not be repeated here. The Update() function is for updating the channel matrix H and the set of received signals y by removing impacts of the fixed symbols, as already described above.

[00132] As shown in FIG. 9, the channel matrix H and the set of received signals x are used as inputs. At block 910, MDA() function is performed for signal detection in a stage. The detection outputs soft estimated symbols at block 920, which means that a probability estimation rather than a hard decision for a symbol is output. At block 930, one or more of the detected symbols are fixed via the Fix () function. The Fix () function may use modulation information, e.g., modulation mode as input, as shown in block 931 of FIG. 9. Symbols fixed at block 930 is output at block 940. At block 950, the index of the stage is increased by 1, and the receiver device checks whether total number of stages is reached. If the total number of stages is not reached, the channel matrix H and the set of received signals y are updated via the Update () function at block 960, and the processing enters the next stage. If the total number of stages is reached, the staged signal detection stops, and the receiver device outputs the remaining symbols which are not fixed at block 970 if any. The fixed symbols and the unfixed symbols form a set of final estimated symbols for further processing, for example, for channel decoding.

[00133] As can be observed from the example shown in FIG. 9, the SSF framework proposed in the present disclosure has at least the following features:

(i) It may provide a type of ML method, by fixing estimated symbols using a Euclidean distance based criterion, stage by stage, to remove inter-symbol interference.

(ii) It has a complexity same as that of the used MDA() function, because neither Fix() nor Update() function introduces much complexity and the required stages are normally far less than N_t .

(iii) It may be used in a network device, and also a terminal device if the MDA() function can be supported by the terminal device.

(iv) It may achieve better detection accuracy even if a detection algorithm with low detection accuracy and low complexity is used in MDA() function in each stage. That is, the proposed framework enables to build low complexity and high accuracy mMIMO detection algorithms. If the detection algorithm used in MDA() is already an optimal

algorithm with high computing complexity, the staged processing may not achieve significant gains.

(v) It inherits the properties of the detection algorithm used in MDA() in each stage. For example, if a MDA() function supports hybrid modulation modes, the proposed SSF solution supports the hybrid modulation modes as well.

(vi) In each stage of the proposed staged detection frame work, a different MDA() function, i.e., a different detection algorithm, may be used.

(vii) The MDA() function in each stage outputs soft estimated symbols.

[00134] Among other advantages, some embodiments of the staged signal detection framework proposed in the present disclosure provide a more flexible signal detection solution. For example, the staged signal detection may be configured via one or more of the parameters shown in Table 7 below.

Table 7. Examples of configuration parameters

Name	Description
N_{tot}	the total number of stage.
fNum	used for example in Fix() to indicate number of symbols required to be fixed in a stage
fRatio	a value between 0 and 1, stands for a ratio of symbols to be fixed.

[00135] In some embodiments, fRatio may be used for determining fNum and/or N_{tot} . For example, fNum may be obtained using equation (16) or (17) based on fRatio, and N_{tot} may be obtained using (22) based on fRatio. If fRatio=1/2, equation (22) may be simplified to equation (23) below.

$$N_{tot} = [\log_2 N_t] + 1 \quad (23)$$

[00136] It can be observed from equation (23) that the total number of stages for detection may be much less than the number of transmitting antennas, which means low complexity of the proposed scheme. This is advantageous, since a target for mMIMO detection is to fix as much symbols as possible to get an optimal detection accuracy and at the same time keep the computing complexity as low as possible.

[00137] Note that, the staged signal detection solutions proposed in the present disclosure do not require fixing all the detected symbols. In practice, if detection accuracy already meets a predefined requirement by fixing only a portion of the detected symbols,

there is no need to fix all the detected symbols, and in such a case, the total number of stages may be reduced.

[00138] In some embodiments, the configuration parameters shown in Table 7 for the staged signal detection solution may be determined based on a receiving SNR. Normally, fRatio may take a value in a range of [0.5, 0.7], which allows fixing enough symbols in the first processing stage and at the same time avoids an increase in detection error probability.

[00139] Furthermore, as already described above, in some embodiments of the present disclosure, the detection algorithm used in a stage of the staged signal detection framework may be selected based on needs, and the selection may even be stage specific. Such a mechanism provides more flexibility to the proposed staged detection framework.

[00140] For illustration purpose, some examples for the detection algorithm may be described below, however, it should be appreciated that embodiments are not limited thereto.

[00141] In an example embodiment, a MMSE detection algorithm/detector is selected for each stage, and such a staged detection solution is also referred to as a ssfMMSE solution herein. MMSE is a popular low complexity detection algorithm, and a detected symbol output from a MMSE detector may be expressed as following equation (24):

$$\hat{x} = \rho(\rho H^H H + I)^{-1} H^H y \quad (24)$$

where ρ is related to SNR, and is obtained by $\rho = 10^{(\text{SNR}/10)}$, I is a unit matrix with size (N_t, N_t) , H stands for the channel matrix, and y stands for the received signals. In ssfMMSE, the $\hat{x}$ output from the MMSE detector is used as an input to a following fixing operation/function to obtain one or more fixed detected symbols.

[00142] In another example embodiment, a ZF detection algorithm is selected for each stage, and such a staged detection solution is also referred to as a ssfZF solution herein. ZF is another very popular detection algorithm, and a detected symbol output from a ZF detector may be expressed as following equation (25):

$$\hat{x} = (H^H H)^{-1} H^H y \quad (25)$$

where H stands for the channel matrix, and y stands for the received signals. In ssfZF, the $\hat{x}$ output from the ZF detector is used as an input to a following fixing operation/function to obtain one or more fixed detected symbols.

[00143] In still another example embodiment, a GD based detection algorithm, for example a SGD detection algorithm proposed in the present disclosure for MIMO detection, may be selected for each stage, and such a staged detection solution is also referred to as a ssfSGD solution herein.

[00144] The SGD detection algorithm transfers problem of equation (1) into an ANN problem, as illustrated schematically in FIG. 2. As shown in FIG. 2, with this ANN model, each row vector $(h_{i1}, h_{i2}, \dots, h_{iN_t})$ of the channel matrix H is considered as an input vector, the received value y_i associated with a corresponding row of H is considered as an output, while vector $x=[x_1, x_2, \dots, x_{N_t}]$ is modelled as weights to be learned.

[00145] The SGD detection method (which may also be referred to as SGD learning method) may be used to solve the ANN problem and converges quickly. In SGD learning method, a loss function for the i th row of H may be defined as:

$$\text{Loss}_i(x) = \frac{1}{2} |y_i - \sum_j h_{ij} * x_j|^2 \quad (26)$$

[00146] A differential for x_j may be obtained by:

$$\frac{\partial \text{Loss}_i}{\partial x_j} = -(y_i - \sum_j h_{ij} * x_j) h_{ij}^* \quad (27)$$

The obtained differential is used as a gradient descent direction for updating x_j . For example, x_j may be updated by:

$$x_j = x_j + R * (y_i - \sum_j h_{ij} * x_j) h_{ij}^* \quad (28)$$

where R stands for a learning rate, or called a learning step, which is used for controlling the update of the x_j . More details about the GD based mMIMO detection have been provided above in the present disclosure, and details will not be repeated.

[00147] In the ssfSGD solution, the x_j output from the SGD detector is used as an input to a following fixing operation/function to obtain one or more fixed detected symbols. Note that in the ssfSGD solution, a separate updating function as shown in block 960 of FIG. 9 may not be required, since the updating operation is already performed within the SGD detection algorithm, i.e., at block 910 of FIG. 9 when SGD detection algorithm is selected for signal detection in a stage.

[00148] One advantage of the ssfSGD solution is that it is easy to implement by reusing ANN implementations. For example, the fixing operation/function in the proposed ssfSGD solution may be implemented by using a dropout function in ANN. Originally, the dropout function in ANN is a regularization technique for reducing overfitting in neural networks by preventing complex co-adaptations on training data. Dropout refers to ignoring weights updating during the training phase, the dropout weights are selected randomly. In the ssfSGD solution, the dropout function shall be used in a different way. The dropout function in the ssfSGD solution is same as that in ANN in terms of ignoring weights (corresponding to detected symbols in the ssfSGD solution) updating during later learning

phase, but differs in that weights (i.e., detected symbols in the ssfSGD solution) to be fixed are determined by a $\text{Fix}(\cdot)$ function, while in ANN, the weight to be fixed is determined randomly. Same as that in ANN, once the weights (symbols) are fixed, they shall be always fixed/unchanged in later stages (later learning phases), and shall not participate in later detection. Therefore, an ANN method with a new dropout function may be easily implemented in an ANN enabled hardware, to provide a ssfSGD solution. That is, the ssfSGD solution for MIMO signal detection may be implemented by reusing ANN hardware. This is a big benefit for ssfSGD usage.

[00149] To evaluate performance of the staged signal detection solution proposed herein, tests for UL mMIMO signal detection have been conducted via computer simulation. Settings of signals and parameters for the tests are summarized in Table 8 and Table 9.

Table 8. Signal generation for the test

Parameters	Settings
H	element is generated by $N(0,1)$, then normalized by $H=H/\sqrt{N_t}$, $N(0,1)$ is normal distribution with a mean of 0 and a variance of 1
z	generated by $N(0,1)$, then normalized by $z=z/(10^{(SNR/10)})$
x	normalized by $x=x/\text{var}(x)$, where $\text{var}(x)$ means the variance of symbol x
y	calculated by equation (1)

Table 9. Uses cases for the test

Cases	Configuration of parameters
Case 1	$N_r=60$, $N_t=30$ Modulation: BPSK $SNR=8\text{dB}$ to 13dB , for each SNR, 20000 sampling points are generated for SER calculation; $N_{\text{tot}}=5$; $f\text{Ratio}=0.5$
Case 2	$N_r=64$, $N_t=16$

	Modulation: 16QAM SNR=8dB to 13dB, for each SNR, 20000 sampling datas are generated for SER calculation; $N_{\text{tot}}=5$; fRatio=0.5
--	--

[00150] Computer simulation results for the two testing cases and three different staged detection algorithms, including ssfZF, ssfMMSE and ssfSGD, are shown in FIGs. 10-15, from which improvement in detection accuracy brought by the staged signal detection solution proposed herein can be observed. For example, FIGs. 10-11 show that compared with a conventional ZF/MMSE detection algorithm, the staged ZF/MMSE detection solution (i.e., ssfZF/ssfMMSE) achieves lower symbol error rate (SER) at same SNR level for test case 1 where 60 receiving antennas, 30 transmitting antennas and BPSK modulation are used. Likewise, FIG. 12 shows that a staged SGD (i.e., ssfSGD) detection solution achieves better performance than a normal SGD detection solution. Results for test case 2 are shown in FIGs. 13-15, and from which it can be observed that performance gain from the staged detection solutions also exists in a MIMO system with 64 receiving antennas, 16 transmitting antennas and 16QAM modulation.

[00151] The staged signal detection solution proposed in the present disclosure for MIMO detection has some or all of the following advantage:

- (a) Computing complexity of the staged signal detection solution is almost in the same level as that of the detection algorithm used in a stage, i.e., no obvious extra computing complexity is introduced due to a staged operation. Furthermore, the number of stages may be selected properly to achieve a tradeoff between complexity and accuracy.
- (b) The staged signal detection solution may be used in a network device and also a terminal device, if the detection algorithm used in a stage is supported by the terminal device.
- (c) It improves detection accuracy considerably.
- (d) It provides a method to build a MIMO detection algorithm with low computing complexity and high detection accuracy.
- (e) It inherits properties of the detection algorithm used in each stage, i.e., if the detection algorithm is a flexible algorithm (e.g., SGD), the staged solution is also a flexible algorithm.

(f) a SGD based staged detection solution is very suitable to be implemented by using a developed ANN, especially in ANN enabled HW.

(g) The proposed staged detection solution allows using different detection algorithm in different stages, which provides extra flexibility for the detection.

[00152] Though computer simulation for UL mMIMO signal detection is conducted, it should be appreciated that the staged signal detection solution proposed in the present disclosure also applies to DL mMIMO signal detection. For DL mMIMO signal detection, the terminal device, for example, the terminal device 102 in FIG. 1 acts as a receiver device. The terminal device 102 may need to detect signals from a single base station or a plurality of base stations depending on its operating mode. In some embodiments, the terminal device 102 may receive signals from the network device 101, and in such a case, both the number of receiving antennas and the number of transmitting antennas may be reduced compared with UL mMIMO signal detection. That is, same staged signal detection solution as described with reference to UL MIMO detection also applies, but computation complexity may be further reduced due to reduced number of antennas.

[00153] FIG. 16 illustrates a simplified block diagram of an apparatus 1600 that may be embodied as or comprised in a receiver device, for example, a terminal device 102 or a network device 101 shown in FIG. 1.

[00154] The apparatus 1600 comprises at least one processor 1611, such as a data processor (DP) and at least one memory (MEM) 1612 coupled to the processor 1611. The apparatus 1610 may further include a transmitter TX and receiver RX 1613 coupled to the processor 1611, which may be operable to communicatively connect to other apparatuses. The MEM 1612 stores a program or computer program code 1614. The at least one memory 1612 and the computer program code 1614 are configured to, with the at least one processor 1611, cause the apparatus 1600 at least to perform in accordance with embodiments of the present disclosure, for example method 600 or 900.

[00155] A combination of the at least one processor 1611 and the at least one MEM 1612 may form processing means 1615 configured to implement various embodiments of the present disclosure.

[00156] Various embodiments of the present disclosure may be implemented by computer program executable by the processor 1611, software, firmware, hardware or in a combination thereof.

[00157] The MEM 1612 may be of any type suitable to the local technical environment and may be implemented using any suitable data storage technology, such as semiconductor

based memory devices, magnetic memory devices and systems, optical memory devices and systems, fixed memory and removable memory, as non-limiting examples.

[00158] The processor 1611 may be of any type suitable to the local technical environment, and may include one or more of general purpose computers, special purpose computers, microprocessors, digital signal processors (DSPs) and processors based on multicore processor architecture, as non-limiting examples.

[00159] Although some of the above descriptions on the GD based signal detection and staged signal detection are made in the context of a wireless communication system shown in FIG. 1, it should not be construed as limiting the spirit and scope of the present disclosure. The principle and concept of the present disclosure may be more generally applicable to other scenarios.

[00160] In addition, the present disclosure may also provide a carrier containing the computer program as mentioned above (e.g., computer instructions/program 530 in FIG. 5 or 1614 in FIG. 16). The carrier includes a computer readable storage medium and a transmission medium. The computer readable storage medium may include, for example, an optical compact disk or an electronic memory device like a RAM (random access memory), a ROM (read only memory), Flash memory, magnetic tape, CD-ROM, DVD, Blue-ray disc and the like. The transmission medium may include, for example, electrical, optical, radio, acoustical or other form of propagated signals, such as carrier waves, infrared signals, and the like.

[00161] The techniques described herein may be implemented by various means so that an apparatus implementing one or more functions of a corresponding apparatus described with an embodiment comprises not only prior art means, but also means for implementing the one or more functions of the corresponding apparatus and it may comprise separate means for each separate function, or means that may be configured to perform two or more functions. For example, these techniques may be implemented in hardware (e.g., circuit or a processor), firmware, software, or combinations thereof. For a firmware or software, implementation may be made through modules (e.g., procedures, functions, and so on) that perform the functions described herein.

[00162] Some example embodiments herein have been described above with reference to block diagrams and flowchart illustrations of methods and apparatuses. It will be understood that each block of the block diagrams and flowchart illustrations, and combinations of blocks in the block diagrams and flowchart illustrations, respectively, may be implemented by various means including computer program instructions. These

computer program instructions may be loaded onto a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions which execute on the computer or other programmable data processing apparatus create means for implementing the functions specified in the flowchart block or blocks.

[00163] While this specification contains many specific implementation details, these should not be construed as limitations on the scope of any implementation or of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular implementations. Certain features that are described in this specification in the context of separate embodiments may also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable sub-combination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a sub-combination or variation of a sub-combination.

[00164] It will be obvious to a person skilled in the art that, as the technology advances, the inventive concept may be implemented in various ways. The above described embodiments are given for describing rather than limiting the disclosure, and it is to be understood that modifications and variations may be resorted to without departing from the spirit and scope of the disclosure as those skilled in the art readily understand. Such modifications and variations are considered to be within the scope of the disclosure and the appended claims. The protection scope of the disclosure is defined by the accompanying claims.

[00165] Some abbreviations used in the present disclosure and their corresponding expressions are list below:

GD:	gradient descent
BGD:	batch gradient descent
SGD:	stochastic gradient descent
MIMO:	multiple input multiple output antenna system
mMIMO:	massive multiple input multiple output antenna system
SVD:	singular value decomposition
5G:	the fifth generation telecommunication system
gNodeB:	5G NodeB

AWGN:	additive white gaussian noise
ZF:	zero forcing
ZF-IC:	zero forcing with interference cancellation
LMMSE:	linear minimum mean square error
LMMSE-IC:	linear minimum mean square error with interference cancellation
SSF:	staged symbol fix
MDA:	MIMO detection algorithm
ssfMDA:	staged symbol fix based MIMO detection algorithm
AMP:	approximate message passing
PDA:	probabilistic data association
MCMC:	Markov Chain Monte Carlo
SDR:	semidefinite relaxation
QAM:	Quadrature Amplitude Modulation
BPSK:	Binary Phase Shift Keying
QPSK:	Quadrature Phase Shift Keying
ANN:	artificial neuron network
DNN:	deep neuron network
SER:	symbol error ratio
SNR:	signal noise ratio

[00166] References obtained from the following links are attached:

<http://ieeexplore.ieee.org/document/7282651/>;

<https://emtiyaz.github.io/Writings/approxMP0.pdf>; and

<https://arxiv.org/abs/1706.01151>.

Optimality of Large MIMO Detection via Approximate Message Passing

Charles Jeon, Ramina Ghods, Arian Maleki, and Christoph Studer

Abstract—Optimal data detection in multiple-input multiple-output (MIMO) communication systems with a large number of antennas at both ends of the wireless link entails prohibitive computational complexity. In order to reduce the computational complexity, a variety of sub-optimal detection algorithms have been proposed in the literature. In this paper, we analyze the optimality of a novel data-detection method for large MIMO systems that relies on approximate message passing (AMP). We show that our algorithm, referred to as individually-optimal (IO) large-MIMO AMP (short IO-LAMA), is able to perform IO data detection given certain conditions on the MIMO system and the constellation set (e.g., QAM or PSK) are met.

I. INTRODUCTION

We consider the problem of recovering the M_T -dimensional data vector $\mathbf{s}_0 \in \mathcal{O}^{M_T}$ from the noisy multiple-input multiple-output (MIMO) input-output relation $\mathbf{y} = \mathbf{H}\mathbf{s}_0 + \mathbf{n}$, by performing individually-optimal (IO) data detection [2], [3]

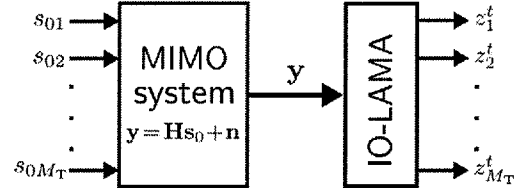
$$(\text{IO}) \quad s_\ell^{\text{IO}} = \arg \max_{\tilde{s}_\ell \in \mathcal{O}} p(\tilde{s}_\ell | \mathbf{y}, \mathbf{H}).$$

Here, s_ℓ^{IO} denotes the ℓ -th IO estimate, $\mathcal{O}$ is a finite constellation (e.g., QAM or PSK), $p(\tilde{s}_\ell | \mathbf{y}, \mathbf{H})$ is a probability density function assuming i.i.d. zero-mean complex Gaussian noise for the vector $\mathbf{n} \in \mathbb{C}^{M_R}$ with variance N_0 per complex dimension, M_T and M_R denotes the number of transmit and receive antennas, respectively, $\mathbf{y} \in \mathbb{C}^{M_R}$ is the receive vector, and $\mathbf{H} \in \mathbb{C}^{M_R \times M_T}$ is the (known) MIMO system matrix. In what follows, we assume that the entries of the MIMO system matrix $\mathbf{H}$ are i.i.d. zero-mean complex Gaussian with variance $1/M_R$, and we define the so-called *system ratio* as $\beta = M_T/M_R$.

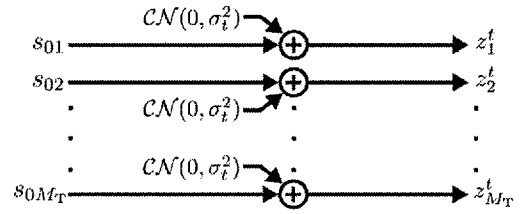
Although IO detection achieves the minimum symbol error rate [4], the combinatorial nature of the (IO) problem [2], [3] requires prohibitive computational complexity, especially in large (or massive) MIMO systems [4], [5]. In order to enable data detection in such high-dimensional systems, a large number of low-complexity but sub-optimal algorithms have been proposed in the literature (see, e.g., [6]–[8]).

A. Contributions

In this paper, we propose and analyze a novel, computationally efficient data-detection algorithm, referred to as IO-LAMA (short for IO large MIMO approximate message



(a) MIMO system with IO-LAMA as the data detector.



(b) Equivalent decoupled system with effective noise variance σ_t^2 .

Fig. 1. IO-LAMA decouples large MIMO systems (a) into a set of parallel and independent AWGN channels with equal noise variance; (b) equivalent system in the large-system limit, i.e., for $\beta = M_T/M_R$ with $M_T \rightarrow \infty$.

passing). We show that IO-LAMA decouples the noisy MIMO system into a set of independent additive white Gaussian noise (AWGN) channels with equal signal-to-noise ratio (SNR); see Fig. 1 for an illustration of this decoupling property. The state-evolution (SE) recursion of AMP enables us to track the effective noise variance σ_t^2 of each decoupled AWGN channel at every algorithm iteration t . Using these results, we provide precise conditions on the MIMO system matrix, the system ratio β , the noise variance N_0 , and the modulation scheme for which IO-LAMA *exactly* solves the (IO) problem.

B. Relevant Prior Art

Initial results for IO data detection in large MIMO systems reach back to [9] where Verdú and Shamai analyzed the achievable rates under optimal data detection in randomly-spread CDMA systems. Tanaka [10] derived expressions for the error-rate performance and the multi-user efficiency for IO detection using the replica method. While Tanaka's results were limited to BPSK constellations, Guo and Verdú extended his results to arbitrary discrete input distributions [3], [11]. All these results study the fundamental performance of IO data detection in the large-system limit, i.e., for $\beta = M_T/M_R$ with $M_T \rightarrow \infty$. Corresponding practical detection algorithms have been proposed for BPSK constellations [12], [13]—to the best of our knowledge, no computationally efficient algorithms for general constellation sets and complex-valued systems have been proposed in the open literature.

An extended version of this paper including all proofs is in preparation [11].
C. Jeon, R. Ghods, and C. Studer are with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY; e-mail: jeon@csl.cornell.edu, rghods@csl.cornell.edu, studer@cornell.edu.

A. Maleki is with Department of Statistics at Columbia University, New York City, NY; e-mail: arian@stat.columbia.edu.

This work was supported in part by Xilinx Inc. and by the US National Science Foundation (NSF) under grants ECCS-1408006 and CCF-1420328.

Our data-detection method, IO-LAMA, builds upon approximate message passing (AMP) [14]–[16], which was initially developed for the recovery of sparse signals. AMP has been generalized to arbitrary signal priors in [17]–[19] and enables a precise performance analysis via the SE recursion [14], [15]. Recently, AMP-related algorithms have been proposed for data detection [20]–[22]; these algorithms, however, lack of a theoretical performance analysis.

C. Notation

Lowercase and uppercase boldface letters designate vectors and matrices, respectively. For a matrix $\mathbf{H}$, we define its conjugate transpose to be $\mathbf{H}^H$. The ℓ -th column of $\mathbf{H}$ is denoted by $\mathbf{h}_\ell$. We use $\langle \cdot \rangle$ to write $\langle \mathbf{x} \rangle = \frac{1}{N} \sum_{k=1}^N x_k$. A multivariate complex-valued Gaussian probability density function (pdf) is denoted by $\mathcal{CN}(\mathbf{m}, \mathbf{K})$, where $\mathbf{m}$ is the mean vector and $\mathbf{K}$ the covariance matrix. $\mathbb{E}_X[\cdot]$ and $\text{Var}_X[\cdot]$ denotes the expectation and variance operator with respect to the pdf of the random variable X , respectively.

II. IO-LAMA: LARGE-MIMO DETECTION USING AMP

We now present IO-LAMA and the SE recursion, which is used in Section III for our optimality analysis.

A. The IO-LAMA Algorithm

We assume that the transmit symbols s_ℓ , $\ell = 1, \dots, M_T$, of the transmit data vector $\mathbf{s}$ are taken from a finite set $\mathcal{O} = \{a_j : j = 1, \dots, |\mathcal{O}|\}$ with constellation points a_j chosen, e.g., from a QAM or PSK alphabet. We assume an i.i.d. prior $p(\mathbf{s}) = \prod_{\ell=1}^{M_T} p(s_\ell)$, with the following distribution for each transmit symbol s_ℓ :

$$p(s_\ell) = \sum_{a \in \mathcal{O}} p_a \delta(s_\ell - a). \quad (1)$$

Here, p_a designates the (known) prior probability of each constellation point $a \in \mathcal{O}$ and $\delta(\cdot)$ is the Dirac delta function; for uniform priors, we have $p_a = |\mathcal{O}|^{-1}$.

The IO-LAMA algorithm summarized below is obtained by using the prior distribution in (1) within complex Bayesian AMP. A detailed derivation of the algorithm is given in [1].

Algorithm 1. Initialize $\hat{s}_\ell^1 = \mathbb{E}_S[S]$ for $\ell = 1, \dots, M_T$, $\mathbf{r}^1 = \mathbf{y}$, and $\tau^1 = \beta \text{Var}_S[S]/N_0$. Then, for every IO-LAMA iteration $t = 1, 2, \dots$, compute the following steps:

$$\begin{aligned} \mathbf{z}^t &= \hat{\mathbf{s}}^t + \mathbf{H}^H \mathbf{r}^t \\ \hat{\mathbf{s}}^{t+1} &= \mathbf{F}(\mathbf{z}^t, N_0(1 + \tau^t)) \\ \tau^{t+1} &= \frac{\beta}{N_0} \langle \mathbf{G}(\mathbf{z}^t, N_0(1 + \tau^t)) \rangle \\ \mathbf{r}^{t+1} &= \mathbf{y} - \mathbf{H} \hat{\mathbf{s}}^{t+1} + \frac{\tau^{t+1}}{1 + \tau^t} \mathbf{r}^t. \end{aligned}$$

The functions $\mathbf{F}(s_\ell, \tau)$ and $\mathbf{G}(s_\ell, \tau)$ correspond to the message mean and variance, and are computed as follows:

$$\begin{aligned} \mathbf{F}(\hat{s}_\ell, \tau) &= \int_{s_\ell} s_\ell f(s_\ell | \hat{s}_\ell, \tau) ds_\ell \\ \mathbf{G}(\hat{s}_\ell, \tau) &= \int_{s_\ell} |s_\ell|^2 f(s_\ell | \hat{s}_\ell, \tau) ds_\ell - |\mathbf{F}(\hat{s}_\ell, \tau)|^2. \end{aligned} \quad (2)$$

Here, $f(s_\ell | \hat{s}_\ell, \tau)$ is the posterior pdf defined by $f(s_\ell | \hat{s}_\ell, \tau) = \frac{1}{Z} p(\hat{s}_\ell | s_\ell, \tau) p(s_\ell)$ with $p(\hat{s}_\ell | s_\ell, \tau) \sim \mathcal{CN}(s_\ell, \tau)$ and a normalization constant Z . Both functions $\mathbf{F}(\hat{s}_\ell, \tau)$ and $\mathbf{G}(\hat{s}_\ell, \tau)$ operate element-wise on vectors.

In order to analyze the performance of IO-LAMA in the large-system limit, we next summarize the SE recursion. The SE recursion in the following theorem enables us to track the effective noise variance σ_t^2 for the decoupled MIMO system for every iteration t (cf. Fig. 1), which is key for the optimality analysis in Section III. A detailed derivation is given in [1].

Theorem 1. Fix the system ratio $\beta = M_T/M_R$ and the constellation set $\mathcal{O}$, and let $M_T \rightarrow \infty$. Initialize $\sigma_1^2 = N_0 + \beta \text{Var}_S[S]$. Then, the effective noise variance σ_t^2 of IO-LAMA at iteration t is given by the following recursion:

$$\sigma_t^2 = N_0 + \beta \Psi(\sigma_{t-1}^2). \quad (3)$$

The so-called mean-squared error (MSE) function is defined by

$$\Psi(\sigma_{t-1}^2) = \mathbb{E}_{S,Z} \left[|F(S + \sigma_{t-1} Z, \sigma_{t-1}^2) - S|^2 \right],$$

where F is given in (2) and $Z \sim \mathcal{CN}(0, 1)$.

B. IO-LAMA Decouples Large MIMO Systems

In the large-system limit and for every iteration t , IO-LAMA computes the marginal distribution of s_ℓ , $\ell = 1, \dots, M_T$, which corresponds to a Gaussian distribution centered around the original signal $s_{0\ell}$ with variance σ_ℓ^2 . These properties follow from [16, Sec. 6], which shows that $\mathbf{z}^t = \hat{\mathbf{s}}^t + \mathbf{H}^H \mathbf{r}^t$ is distributed according to $\mathcal{CN}(\mathbf{s}_0, \sigma_t^2 \mathbf{I}_{M_T})$. Hence, the input-output relation for each transmit stream $z_\ell^t = \hat{s}_\ell^t + (\mathbf{h}_\ell^H \mathbf{r}^t)$ is equivalent to the following single-stream AWGN channel:

$$z_\ell^t = s_{0\ell} + n_\ell^t.$$

Here, $s_{0\ell}$ is the ℓ -th original transmitted signal and n_ℓ^t is AWGN with variance σ_t^2 per complex entry. As a consequence, IO-LAMA decouples the MIMO system into M_T parallel and independent AWGN channels with equal noise variance σ_t^2 in the large-MIMO limit; see Fig. 1(b) for an illustration.

III. OPTIMALITY OF IO-LAMA

We now provide conditions for which IO-LAMA *exactly* solves the (IO) problem.

A. Fixed points of IO-LAMA's State Evolution

For $t \rightarrow \infty$, the SE recursion in Theorem 1 converges to the following fixed-point equation [1], [15]:

$$\sigma_{\text{IO}}^2 = N_0 + \beta \Psi(\sigma_{\text{IO}}^2), \quad (4)$$

which coincides with the “fixed-point equation” developed for IO detection by Guo and Verdú using the replica method in [3, Eq. (34)]. We note that (4) may have multiple fixed-point solutions. In the case of such non-unique fixed points, Guo and Verdú choose the solution that minimizes the “free energy” [3, Sec. 2-D], whereas IO-LAMA converges, in general, to the fixed-point solution with the largest effective noise variance σ^2 . We note that if the fixed-point solution to (4) is unique, then IO-LAMA recovers the solution with minimal effective noise variance σ^2 and thus, performs IO detection. However, if there are multiple fixed-points solutions to (4), IO-LAMA is, in general, sub-optimal and does not necessarily converge to the fixed-point solution with the minimal “free energy.” We next

¹Convergence to another fixed-point solution is possible if IO-LAMA is initialized sufficiently close to such a fixed point; see [1], [23] for the details.

provide conditions for which there is exactly one (unique) fixed point with minimum effective noise variance σ^2 and—as a consequence—IO-LAMA solves the (IO) problem.

B. Exact Recovery Thresholds (ERTs)

We start by analyzing IO-LAMA in the noiseless setting. We provide conditions on the system ratio β and the constellation set $\mathcal{O}$, which guarantee exact recovery of an unknown transmit signal $\mathbf{s}_0 \in \mathcal{O}^{M_T}$ in the large-system limit, i.e., β is fixed and $M_T \rightarrow \infty$. In particular, we show that if $\beta < \beta_{\mathcal{O}}^{\max}$, where $\beta_{\mathcal{O}}^{\max}$ is the so-called *exact recovery threshold (ERT)*, then IO-LAMA perfectly recovers $\mathbf{s}_0$; for $\beta \geq \beta_{\mathcal{O}}^{\max}$, perfect recovery is not guaranteed, in general.² To make this behavior explicit, we need the following technical result; the proof is given in Appendix A.

Lemma 2. *Fix the constellation set $\mathcal{O}$. If $\text{Var}_S[S]$ is finite, then there exists a non-negative gap $\sigma^2 - \Psi(\sigma^2) \geq 0$ with equality if and only if $\sigma^2 = 0$. As $\sigma^2 \rightarrow 0$, the MSE $\Psi(\sigma^2) \rightarrow 0$ and as $\sigma^2 \rightarrow \infty$, $\text{MSE } \Psi(\sigma^2) \rightarrow \text{Var}_S[S]$.*

For all $\sigma^2 > 0$, Lemma 2 guarantees that $\Psi(\sigma^2) < \sigma^2$. Suppose that for some $\beta > 1$, $\beta\Psi(\sigma^2) < \sigma^2$ also holds for all $\sigma^2 > 0$. Then, as long as $\beta > 1$ is not too large to also ensure $\beta\Psi(\sigma^2) < \sigma^2$ for all $\sigma^2 > 0$, there will only be a *single* fixed point at $\sigma^2 = 0$. Therefore, LAMA can still perfectly recover the original signal $\mathbf{s}_0$ by Theorem 1 since $\Psi(\sigma^2) = 0$. Leveraging the gap between $\Psi(\sigma^2)$ and σ^2 will allow us to find the exact recovery threshold (ERT) of LAMA for values of $\beta > 1$. For the fixed (discrete) constellation set $\mathcal{O}$, the largest β that ensures $\beta\Psi(\sigma^2) < \sigma^2$ is precisely the ERT defined next.

Definition 1. *Fix $\mathcal{O}$ and let $N_0 = 0$. Then, the exact recovery threshold (ERT) that enables perfect recovery of the original signal $\mathbf{s}_0$ using IO-LAMA is given by*

$$\beta_{\mathcal{O}}^{\max} = \min_{\sigma^2 > 0} \left\{ \left(\frac{\Psi(\sigma^2)}{\sigma^2} \right)^{-1} \right\}. \quad (5)$$

With Definition 1, we state Theorem 3, which establishes optimality in the noiseless case; the proof is given in Appendix B.

Theorem 3. *Let $N_0 = 0$ and fix a discrete set $\mathcal{O}$. If $\beta < \beta_{\mathcal{O}}^{\max}$, then IO-LAMA perfectly recovers the original signal $\mathbf{s}_0$ from $\mathbf{y} = \mathbf{H}\mathbf{s}_0 + \mathbf{n}$ in the large system limit.*

Note that for a given constellation set $\mathcal{O}$, the ERT $\beta_{\mathcal{O}}^{\max}$ can be computed numerically using (5). Furthermore, the signal variance, $\text{Var}_S[S]$, has no impact on the ERT as the MSE function $\Psi(\sigma^2)$ and σ^2 scale linearly with $\text{Var}_S[S]$. Table I summarizes ERTs $\beta_{\mathcal{O}}^{\max}$ for common QAM and PSK constellation sets.

C. Optimality Conditions for IO-LAMA

We now study the optimality of IO-LAMA in the presence of noise, where *exact* recovery is no longer guaranteed. In particular, we provide conditions for which IO-LAMA converges to the fixed point with minimal effective noise variance σ^2 , which corresponds to solving the (IO) problem.

²We assume the initialization in Algorithm 1. IO-LAMA may recover the original signal for $\beta \geq \beta_{\mathcal{O}}^{\max}$ if initialized appropriately; see, e.g., [23].

TABLE I
ERTs $\beta_{\mathcal{O}}^{\max}$, MRTs $\beta_{\mathcal{O}}^{\min}$, AND CRITICAL NOISE LEVELS $N_0^{\min}(\beta_{\mathcal{O}}^{\min})$ AND $N_0^{\max}(\beta_{\mathcal{O}}^{\max})$ OF IO-LAMA FOR COMMON CONSTELLATION SETS

Constellation	$\beta_{\mathcal{O}}^{\min}$	$N_0^{\min}(\beta_{\mathcal{O}}^{\min})$	$\beta_{\mathcal{O}}^{\max}$	$N_0^{\max}(\beta_{\mathcal{O}}^{\max})$
BPSK	2.9505	$2.999 \cdot 10^{-1}$	4.1709	$2.432 \cdot 10^{-1}$
QPSK	1.4752	$1.499 \cdot 10^{-1}$	2.0855	$1.216 \cdot 10^{-1}$
16-QAM	0.9830	$3.000 \cdot 10^{-2}$	1.3629	$2.454 \cdot 10^{-2}$
64-QAM	0.8424	$7.144 \cdot 10^{-3}$	1.1573	$5.868 \cdot 10^{-3}$
8-PSK	1.4576	$4.440 \cdot 10^{-2}$	1.8038	$3.826 \cdot 10^{-2}$
16-PSK	1.4728	$1.143 \cdot 10^{-2}$	1.8005	$9.953 \cdot 10^{-3}$

TABLE II
SUMMARY OF (SUB-)OPTIMALITY REGIMES OF IO-LAMA

	$\beta \leq \beta_{\mathcal{O}}^{\min}$	$\beta_{\mathcal{O}}^{\min} < \beta < \beta_{\mathcal{O}}^{\max}$	$\beta_{\mathcal{O}}^{\max} \leq \beta$
$N_0 < N_0^{\min}(\beta)$	optimal	optimal	suboptimal
$N_0^{\min}(\beta) \leq N_0 \leq N_0^{\max}(\beta)$	optimal	(sub-)optimal ³	suboptimal
$N_0^{\max}(\beta) < N_0$	optimal	optimal	optimal

Note that such a minimum free-energy solution is also the fixed point for the IO detector in [3, Eq. (34)]. We call the fixed point with minimum effective noise variance *optimal fixed point*; other fixed points are called *suboptimal fixed points*.

We identify three different operation regimes for IO-LAMA depending on the system ratio β (see Table II). To make these three regimes explicit, we need the following definition.

Definition 2. *Fix the constellation set $\mathcal{O}$. Then, the minimum recovery threshold (MRT) $\beta_{\mathcal{O}}^{\min}$ is defined by*

$$\beta_{\mathcal{O}}^{\min} = \min_{\sigma^2 > 0} \left\{ \left(\frac{d\Psi(\sigma^2)}{d\sigma^2} \right)^{-1} \right\}. \quad (6)$$

The definition of MRT shows that for all system ratios $\beta \leq \beta_{\mathcal{O}}^{\min}$, the fixed point of (4) is unique. The following lemma establishes a fundamental relationship between MRT and ERT; the proof is given in Appendix C.

Lemma 4. *The MRT never exceeds the ERT.*

We next define the minimum critical and maximum guaranteed noise variance, $N_0^{\min}(\beta)$ and $N_0^{\max}(\beta)$, that determine boundaries for the optimality regimes when $\beta > \beta_{\mathcal{O}}^{\min}$.

Definition 3. *Fix $\beta \in (\beta_{\mathcal{O}}^{\min}, \beta_{\mathcal{O}}^{\max})$. Then, the minimum critical noise $N_0^{\min}(\beta)$ that ensures convergence to the optimal fixed point is defined by*

$$N_0^{\min}(\beta) = \min_{\sigma^2 > 0} \left\{ \sigma^2 - \beta\Psi(\sigma^2) : \beta \frac{d\Psi(\sigma^2)}{d\sigma^2} = 1 \right\}.$$

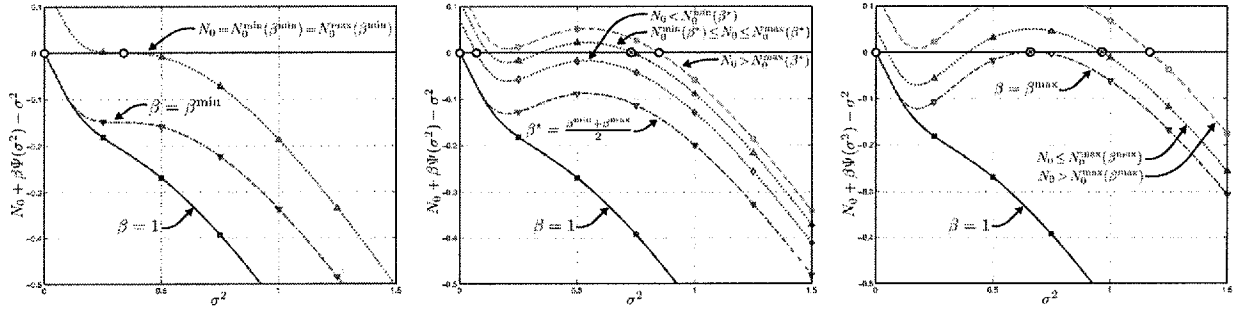
Definition 4. *Fix $\beta > \beta_{\mathcal{O}}^{\min}$. Then, the maximum guaranteed noise $N_0^{\max}(\beta)$ that ensures convergence to the optimal fixed point is defined by*

$$N_0^{\max}(\beta) = \max_{\sigma^2 > 0} \left\{ \sigma^2 - \beta\Psi(\sigma^2) : \beta \frac{d\Psi(\sigma^2)}{d\sigma^2} = 1 \right\}.$$

We recall that all the zero crossings of the function

$$g(\sigma^2, \beta, N_0)_{\mathcal{O}} = N_0 + \beta\Psi(\sigma^2) - \sigma^2 \quad (7)$$

³For certain constellation sets (e.g., 16-PSK), there exist sub-intervals in $[N_0^{\min}(\beta), N_0^{\max}(\beta)]$ where IO-LAMA is still optimal; see [1] for the details.



(a) $\beta \leq \beta_O^{\min}$: IO-LAMA always converges to the unique, optimal fixed point (FP) irrespective of N_0 . (b) $\beta \in (\beta_O^{\min}, \beta_O^{\max})$: IO-LAMA converges to the optimal FP if $N_0 < N_0^{\min}(\beta)$ or $N_0 > N_0^{\max}(\beta)$. (c) $\beta \geq \beta_O^{\max}$: IO-LAMA converges to the optimal fixed point if $N_0 > N_0^{\max}(\beta)$.

Fig. 2. Plot of the function (7) for three regimes (a) $\beta \leq \beta_O^{\min}$, (b) $\beta \in (\beta_O^{\min}, \beta_O^{\max})$, and (c) $\beta \geq \beta_O^{\max}$ for QPSK modulation, uniform priors, and $\text{Var}_S[S] = E_s = 1$. The optimal fixed points are designated by $\circ$; suboptimal fixed points are designated by $\otimes$.

correspond to all fixed points of the SE recursion of IO-LAMA; we use this function to study the algorithm's optimality.

Figure 2 illustrates our optimality analysis for a large-MIMO system with QPSK constellations. We show (7) depending on the effective noise variance σ^2 and for different system ratios β . The regimes $\beta \leq \beta_O^{\min}$, $\beta \in (\beta_O^{\min}, \beta_O^{\max})$, and $\beta \geq \beta_O^{\max}$ are shown in Fig. 2(a), Fig. 2(b), and Fig. 2(c), respectively. The special case for $\beta = 1$ with $N_0 = 0$ corresponds to the solid blue line, along with the corresponding (unique) fixed point at the origin. In the following three paragraphs, we discuss the three operation regimes of IO-LAMA in detail.

(i) $\beta \leq \beta_O^{\min}$: In this region, the SE recursion of IO-LAMA always converges to the unique, optimal fixed point. For $\beta < \beta_O^{\min}$, the slope of (7) for all σ^2 is strictly-negative. Hence, as (7) is always decreasing, there exists exactly one unique fixed point of the SE recursion regardless of the noise variance N_0 . Thus, IO-LAMA converges to the optimal fixed point and consequently, solves the (IO) problem.

We emphasize that we still obtain exactly one fixed point even when β is equal to the MRT. Since $\beta = \beta_O^{\min}$, there exists at least one σ_*^2 that satisfies $\beta_O^{\min} \frac{d}{d\sigma^2} \Psi(\sigma^2) \big|_{\sigma^2=\sigma_*^2} = 1$. By definition of $\beta_O^{\min}$, (7) at σ_*^2 implies that σ_*^2 is a saddle-point, so (7) has exactly one zero at σ_*^2 . We observe that if σ_*^2 is unique, then $N_0^{\min}(\beta_O^{\min}) = N_0^{\max}(\beta_O^{\min})$. For all other $\sigma^2 \neq \sigma_*^2$, the construction of σ_*^2 implies that $\beta_O^{\min} \frac{d}{d\sigma^2} \Psi(\sigma^2) < 1$, so the fixed point of (7) remains to be unique.

The green, dash-dotted and red, dotted line in Fig. 2(a) shows (7) for $\beta = \beta_O^{\min}$ with $N_0 = 0$ and $N_0 = N_0^{\min}(\beta_O^{\min}) = N_0^{\max}(\beta_O^{\min})$, respectively. In both cases, we see that the SE recursion of IO-LAMA converges to the unique fixed point.

(ii) $\beta_O^{\min} < \beta < \beta_O^{\max}$: In this region, the SE recursion of IO-LAMA converges to the unique, optimal fixed point if $N_0 < N_0^{\min}(\beta)$ or $N_0 > N_0^{\max}(\beta)$.

The green, dash-dotted line, cyan, dashed line, and magenta, dotted line in Fig. 2(b) shows (7) for $\beta^* = (\beta_O^{\min} + \beta_O^{\max})/2$ with $N_0 = 0$, $N_0 > N_0^{\max}(\beta^*)$ and $N_0 < N_0^{\min}(\beta^*)$, respectively. We note that for the three cases, the fixed point is unique, labeled in Fig. 2(b) by a circle. On the other hand, the red, dotted line in Fig. 2(b) shows (7) with β^* under noise

$N_0 \in [N_0^{\min}(\beta^*), N_0^{\max}(\beta^*)]$. In this case, however, we observe that SE recursion of IO-LAMA converges to the rightmost suboptimal fixed point labeled by the crossed circle $\otimes$. Hence, IO-LAMA does not, in general, solve the (IO) problem when $N_0^{\min}(\beta) \leq N_0 \leq N_0^{\max}(\beta)$.

(iii) $\beta \geq \beta_O^{\max}$: In this region, the SE recursion of IO-LAMA converges to the unique, optimal fixed point when $N_0 > N_0^{\max}(\beta)$. As $\beta \rightarrow \beta_O^{\max}$, the low noise $N_0 < N_0^{\min}(\beta)$ (or high SNR) region of optimality disappears because $N_0^{\min}(\beta) \rightarrow 0$ as $\beta \rightarrow \beta_O^{\max}$ from (5).

The green, dash-dotted line and red, dotted line in Fig. 2(c) shows (7) for $\beta = \beta_O^{\max}$ with $N_0 = 0$ and $0 < N_0 \leq N_0^{\max}(\beta)$, respectively. We observe that the SE recursion of IO-LAMA converges to the suboptimal fixed point when $\beta = \beta_O^{\max}$ even with $N_0 = 0$. On the other hand, the cyan, dashed line refers to (7) for $\beta = \beta_O^{\max}$ with $N_0 > N_0^{\max}(\beta)$. While the noiseless case resulted the SE recursion of IO-LAMA to converge to the suboptimal fixed point, we observe that for strong noise (or equivalently low SNR), the SE recursion of IO-LAMA actually recovers the IO solution. Therefore, when $\beta \geq \beta_O^{\max}$, IO-LAMA solves the (IO) problem when the noise is greater than the maximum guaranteed noise $N_0^{\max}(\beta)$.

As a final remark, we note that the ERT $\beta_O^{\max}$ and MRT $\beta_O^{\min}$ in Table I do not depend on $\text{Var}_S[S]$; the critical noise levels $N_0^{\min}(\beta)$ and $N_0^{\max}(\beta)$, however, depend on $\text{Var}_S[S]$.

D. ERT, MRT, and Critical Noise Levels

The ERT, MRT, as well as $N_0^{\min}(\beta)$ and $N_0^{\max}(\beta)$ for common constellations are summarized in Table I. We assume equally likely priors with the transmit signal normalized to $E_s = \text{Var}_S[S] = 1$.⁴ We note that the calculations of ERT and MRT for the simplest case of BPSK constellations involve computations of a logistic-normal integral for which no closed-form expression is known [24]. Consequently, the following results were obtained via numerical integration for computing the MSE function $\Psi(\sigma^2)$. As noted in Table I for a QPSK system under complex-valued noise, the ERT is $\beta_{\text{QPSK}}^{\max} \approx 2.0855$, and the MRT is given as $\beta_{\text{QPSK}}^{\min} \approx 1.4752$.

⁴The critical noise levels depend linearly on E_s . Hence, we assume that $E_s = 1$ without loss of generality.

The MRTs for 16-QAM and 64-QAM indicate that small system ratios $\beta < 1$ are required to always guarantee that IO-LAMA solves the (IO) problem in the presence of noise. For instance, we require $\beta \leq \beta_{64\text{-QAM}}^{\min} \approx 0.8424$, i.e. $M_T \leq 0.8424M_R$, to ensure that IO-LAMA solves the IO problem for 64-QAM in the large system limit. As $\beta \rightarrow \beta_{64\text{-QAM}}^{\max} \approx 1.1573$, IO-LAMA is only optimal for $N_0 > N_0^{\max}(\beta_{64\text{-QAM}}^{\max}) \approx 5.868 \cdot 10^{-3}$. From Table I, we see that IO-LAMA is a suitable candidate algorithm for the detection of higher-order QAM constellations in massive multi-user MIMO systems as one typically assumes $M_R \gg M_T$ [25].

IV. CONCLUSIONS

We have presented the IO-LAMA algorithm along with the state-evolution recursion. Using these results, we have established conditions on the MIMO system matrix, the noise variance N_0 , and the constellation set for which IO-LAMA exactly solves the (IO) problem. While the presented results are exclusively for the large-system limit, our own simulations indicate that IO-LAMA achieves near-optimal performance in realistic, finite-dimensional systems; see [1] for more details.

APPENDIX A PROOF OF LEMMA 2

Since the variance of S is finite, denote $\text{Var}_S[S] = \sigma_s^2$. By [26, Prop. 5], we have the following upper bound:

$$\Psi(\sigma^2) \leq \frac{\sigma_s^2}{\sigma_s^2 + \sigma^2} \sigma^2 = \frac{1}{1 + \sigma^2/\sigma_s^2} \sigma^2. \quad (8)$$

Here, equality holds for all σ^2 if and only if S is complex normal with variance σ_s^2 [26]. Note that if $\sigma^2 = 0$, then (8) is achieved for any σ_s^2 . If $\sigma^2 > 0$, then $\Psi(\sigma^2) < \sigma^2$ by (8).

The first part follows directly from (8) as $\Psi(\sigma^2)$ is non-negative. The second part requires one to realize that $\sigma^2 \rightarrow \infty$ also implies $F(\cdot, \sigma^2) \rightarrow \sum_{a \in \mathcal{O}} a p_a = \mathbb{E}_S[S]$, and hence,

$$\lim_{\sigma^2 \rightarrow \infty} \Psi(\sigma^2) \rightarrow \mathbb{E}_S[|S - \mathbb{E}_S[S]|^2] = \text{Var}_S[S].$$

APPENDIX B PROOF OF THEOREM 3

We assume the initialization in Algorithm 1. Since $N_0 = 0$, if LAMA perfectly recovers the original signal s_0 , then the fixed point in (4) is unique at $\sigma^2 = 0$. This happens if the system ratio is strictly less than the ERT $\beta_{\mathcal{O}}^{\max}$ because otherwise, i.e., $\beta \geq \beta_{\mathcal{O}}^{\max}$, there exists a non-unique fixed point to (4) for some $\sigma^2 > 0$ by Definition 1.

APPENDIX C PROOF OF LEMMA 4

We show that under a fixed constellation set $\mathcal{O}$, $\beta_{\mathcal{O}}^{\min} \leq \beta_{\mathcal{O}}^{\max}$. The proof is straightforward as,

$$\begin{aligned} \beta_{\mathcal{O}}^{\min} &\stackrel{(a)}{=} \min_{\sigma^2 > 0} \left\{ \left(\frac{d\Psi(\sigma^2)}{d\sigma^2} \right)^{-1} \right\} \leq \left(\frac{d\Psi(\sigma^2)}{d\sigma^2} \right)^{-1} \Big|_{\sigma^2 = \beta_{\mathcal{O}}^{\max} \Psi(\sigma^2)} \\ &\stackrel{(b)}{=} \left(\frac{1}{\beta_{\mathcal{O}}^{\max}} \right)^{-1} = \beta_{\mathcal{O}}^{\max}, \end{aligned}$$

where (a) and (b) follow from the MRT and ERT definitions.

REFERENCES

- [1] C. Jeon, R. Ghods, A. Maleki, and C. Studer, "Optimal data detection in large MIMO," *in preparation for a journal*.
- [2] D. Guo and S. Verdú, "Multiuser detection and statistical mechanics," in *Commun., Inform. and Netw. Security*. Springer, 2003, pp. 229–277.
- [3] —, "Randomly spread CDMA: Asymptotics via statistical physics," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1983–2010, Jun. 2005.
- [4] S. Verdú, *Multiuser Detection*, 1st ed. New York, NY, USA: Cambridge University Press, 1998.
- [5] D. Seethaler, J. Jaldén, C. Studer, and H. Bölcskei, "On the complexity distribution of sphere decoding," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 5754–5768, Sept. 2011.
- [6] T. Narasimhan and A. Chockalingam, "Channel hardening-exploiting message passing (CHEMP) receiver in large-scale MIMO systems," *IEEE J. Sel. Topics Sig. Proc. (J-STSP)*, vol. 8, no. 5, pp. 847–860, Oct. 2014.
- [7] M. Čirkić and E. Larsson, "SUMIS: Near-optimal soft-in soft-out MIMO detection with low and fixed complexity," *IEEE Trans. Sig. Proc.*, vol. 62, no. 12, pp. 3084–3097, Jun. 2014.
- [8] M. Wu, B. Yin, G. Wang, C. Dick, J. Cavallaro, and C. Studer, "Large-scale MIMO detection for 3GPP LTE: Algorithm and FPGA implementation," *IEEE J. Sel. Topics Sig. Proc. (J-STSP)*, vol. 8, no. 5, pp. 916–929, Oct. 2014.
- [9] S. Verdú and S. Shamai, "Spectral efficiency of CDMA with random spreading," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 622–640, Mar 1999.
- [10] T. Tanaka, "A statistical-mechanics approach to large-system analysis of CDMA multiuser detectors," *IEEE Trans. Inf. Theory*, vol. 48, no. 11, pp. 2888–2910, Nov. 2002.
- [11] D. Guo and C.-C. Wang, "Random sparse linear systems observed via arbitrary channels: A decoupling principle," in *Proc. IEEE Int'l. Symp. Inf. Theory (ISIT)*, June 2007, pp. 946–950.
- [12] Y. Kabashima, "A CDMA multiuser detection algorithm on the basis of belief propagation," *J. Phys. A: Math. Gen.*, vol. 36, no. 43, pp. 11 111–11 121, Oct. 2003.
- [13] G. Caire, R. Müller, and T. Tanaka, "Iterative multiuser joint decoding: optimal power allocation and low-complexity implementation," *IEEE Trans. Inf. Theory*, vol. 50, no. 9, pp. 1950–1973, Sept. 2004.
- [14] D. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Natl. Academy of Sciences (PNAS)*, vol. 106, no. 45, pp. 18 914–18 919, Sept. 2009.
- [15] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 764–785, Feb. 2011.
- [16] A. Montanari, *Graphical models concepts in compressed sensing, Compressed Sensing (Y.C. Eldar and G. Kutyniok, eds.)*. Cambridge University Press, 2012.
- [17] D. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. Motivation and construction," in *IEEE Inf. Theory Workshop (ITW)*, Jan. 2010, pp. 1–5.
- [18] —, "Message passing algorithms for compressed sensing: II. Analysis and validation," in *IEEE Inf. Theory Workshop (ITW)*, Jan. 2010, pp. 1–5.
- [19] A. Maleki, L. Anitori, Z. Yang, and R. Baraniuk, "Asymptotic analysis of complex LASSO via complex approximate message passing (CAMP)," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4290–4308, July 2013.
- [20] M. Nassar, P. Schniter, and B. Evans, "A factor graph approach to joint OFDM channel estimation and decoding in impulsive noise environments," *IEEE Trans. Sig. Proc.*, vol. 62, no. 6, pp. 1576–1589, Mar. 2014.
- [21] P. Schniter, "A message-passing receiver for BICM-OFDM over unknown clustered-sparse channels," *Proc. Intl. Workshop IEEE Sig. Proc. Adv. Wireless Commun. (SPAWC)*, pp. 246–250, Jun. 2011.
- [22] S. Wu, L. Kuang, Z. Ni, J. Lu, D. Huang, and Q. Guo, "Low-complexity iterative detection for large-scale multiuser MIMO-OFDM systems using approximate message passing," *IEEE J. Sel. Topics Sig. Proc. (J-STSP)*, vol. 8, no. 5, pp. 902–915, Oct. 2014.
- [23] L. Zheng, A. Maleki, X. Wang, and T. Long, "Does ℓ_p -minimization outperform ℓ_1 -minimization?" *arXiv preprint: 1501.03704*, Jan. 2015.
- [24] E. Demidenko, *Generalized Linear Mixed Models*. John Wiley & Sons, Inc., 2005, pp. 329–430.
- [25] E. Larsson, O. Edfors, F. Tufvesson, and T. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.
- [26] D. Guo, Y. Wu, S. Shamai, and S. Verdú, "Estimation in Gaussian noise: Properties of the minimum mean-square error," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2371–2385, Apr. 2011.

Approximate Message Passing

Mohammad Emtiyaz Khan
CS, UBC

February 8, 2012

Abstract

In this note, I summarize Sections 5.1 and 5.2 of Arian Maleki's PhD thesis.

1 Notation

We denote scalars by small letters e.g. $a, b, c, \dots$, vectors by boldface small letters e.g. $\mathbf{\lambda}, \mathbf{\alpha}, \mathbf{x}, \dots$, matrices by boldface capital letter e.g. $\mathbf{A}, \mathbf{B}, \mathbf{C}, \dots$, (subsets of) natural numbers by capital letters e.g. $N, M, \dots$. We denote i 'th element of a vector $\mathbf{a}$ by a_i and (i, j) 'th entry of a matrix $\mathbf{A}$ by A_{ij} . We denote the i 'th column (or row) of $\mathbf{A}$ by $\mathbf{A}_{:,i}$ (or $\mathbf{A}_{i,:}$). We use $\mathbf{A}_{a,-i}$ (or $\mathbf{A}-a,i$) to refer to the a 'th row (or i 'th column) without the element $A_{a,i}$. Also, $\mathbf{A}^T$ denote the transpose of a matrix $\mathbf{A}$.

2 Basis Pursuit Problem

Given measurements $\mathbf{y}$ of length n and matrix $\mathbf{A}$ of size $n \times N$, we wish to compute $\mathbf{s}$ which is the minimizer of Eq. 1. This is known as the basis pursuit problem. Here, $\|\cdot\|_1$ is the l_1 -norm. A version of this problem where we allow for errors in the measurements is called basis pursuit denoising problem (aka LASSO), shown in Eq. 2. Here, $\|\cdot\|_2$ is the l_2 -norm.

$$\text{BP: } \min_{\mathbf{s}} \|\mathbf{s}\|_1, \text{ s.t. } \mathbf{y} = \mathbf{A}\mathbf{s} \quad (1)$$

$$\text{BPDN: } \min_{\mathbf{s}} \lambda \|\mathbf{s}\|_1 + \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{s}\|_2^2 \quad (2)$$

3 Posterior Distribution

Consider the following posterior distributions in Eq. 3, where the prior distribution $p(s_i)$ is the Laplace distribution and the likelihood $p(y_a|\mathbf{s}, \mathbf{A}_{a,:})$ is the Dirac distribution.

$$p(\mathbf{s}|\mathbf{y}) \propto \prod_{i=1}^N p(s_i) \prod_{a=1}^n p(y_a|\mathbf{s}, \mathbf{A}_{a,:}) \quad (3)$$

$$= \prod_{i=1}^N \exp(-\beta|s_i|) \prod_{a=1}^n \delta(y_a - \mathbf{A}_{a,:}\mathbf{s}) \quad (4)$$

As $\beta \rightarrow \infty$, mass of this posterior distribution concentrates around the minimizer of BP. This implies that given the marginals of this posterior distribution, solution of BP is immediate. A formal proof is not given in [Mal11]. We give an intuitive explanation in Fig. 3.

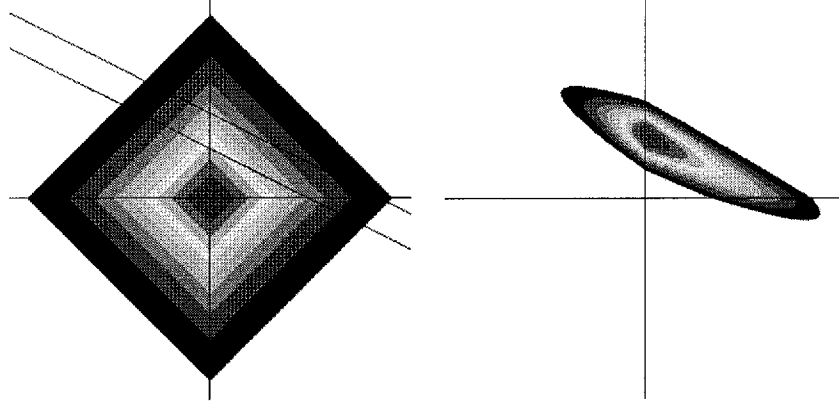


Figure 1: Visualization of the posterior distribution in Eq. 3 for two variables s_1 and s_2 . Left figure shows the negative-log of prior distribution which is $\beta(|s_1| + |s_2|)$ and the negative log-likelihood of a single measurement corresponding to a Gaussian likelihood (black lines). Right figure shows the negative-log of the posterior distribution. As $\beta \rightarrow \infty$, the posterior become more peaky around the sparse solution where s_1 is zero. We can also see that the marginal of s_1 concentrates around 0, while that of s_2 concentrates around a non-zero value. Figure from [See08].

4 Belief Propagation

Belief propagation can be used to compute the marginal distributions of a posterior distribution. We start by defining a *factor graph* which captures the statistical dependencies between the variables, and then do *message passing*. In this section, we will briefly describe belief propagation for the basis pursuit problem; interested reader should see [Bis06] for a general case. First consider the posterior distribution of Eq. 3 with the prior distribution $p(s_i)$ and likelihood $p(y_a|\mathbf{s}, \mathbf{A}_{a,:})$. We define a bipartite factor graph where $s_1, s_2, \dots, s_N$ are variables and $y_1, y_2, \dots, y_n$ are factors. We draw an edge between a variable and a factor if the corresponding measurement depends on the variable (in the BP problem, it will be a dense graph but if $\mathbf{A}$ was sparse then non-zero entries will correspond to an edge). Define, $\mathcal{N}(a)$ to be the neighborhood of a 'th factor i.e. the set of variables that are connected to factor a and define $\mathcal{N}(a) \setminus i$ to be the set without the variable i . The messages, defined below, are passed from variables to factors and then factors to variables.

$$m_{i \rightarrow a}(s_i) = p(s_i) \prod_{b \in \mathcal{N}(i) \setminus a} m_{b \rightarrow i}(s_i) \quad (5)$$

$$m_{a \rightarrow i}(s_i) = \int_{\mathbf{s}_{-i}} p(y_a|\mathbf{s}) \prod_{j \in \mathcal{N}(a) \setminus i} m_{j \rightarrow a}(s_j) d\mathbf{s}_{-i} \quad (6)$$

Intuitively, the message from a variable i to a factor a contains multiplication of prior belief $p(s_i)$ with all the messages received except the message that was sent by factor a . Similarly, the message from a factor a to a variable i contains multiplication of the likelihood $p(y_a|\mathbf{s})$ with all the messages received except the message that was sent by variable i . The variables other than i are then integrated out of the message. The marginal of a variable is then given by multiplication of all the messages that arrive at that variable along with the local belief as shown below.

$$p(s_i|\mathbf{y}) = p(s_i) \prod_{b \in \mathcal{N}(i)} m_{b \rightarrow i}(s_i) \quad (7)$$

We will now give a simple example to show that message passing results in the marginals at each node. Consider two variables s_1, s_2 and s_3 with two measurements y_a and y_b , following a joint distribution which

factorizes as shown below.

$$p(y_a, y_b, s_1, s_2, s_3) = p(y_a|s_1, s_2)p(y_b|s_2, s_3)p(s_1)p(s_2)p(s_3) \quad (8)$$

The statistical dependencies between variable and measurements can be expressed using the following factor graph: $s_1 - y_a - s_2 - y_b - s_3$. Here, y_a depends on s_1, s_2 and y_b depends on s_2, s_3 . Using Eq. 5 and 6, we can write down the messages explicitly as shown below.

Messages from factors to variables:	Messages from variables to factors:
$m_{a \rightarrow 1}(s_1) = \int_{s_2} p(y_a s_1, s_2)m_{2 \rightarrow a}(s_2)ds_2 \quad (9)$	$m_{1 \rightarrow a}(s_1) = p(s_1) \quad (13)$
$m_{a \rightarrow 2}(s_2) = \int_{s_1} p(y_a s_1, s_2)m_{1 \rightarrow a}(s_1)ds_1 \quad (10)$	$m_{2 \rightarrow a}(s_2) = p(s_2)m_{b \rightarrow 2}(s_2) \quad (14)$
$m_{b \rightarrow 2}(s_2) = \int_{s_3} p(y_b s_2, s_3)m_{3 \rightarrow b}(s_3)ds_3 \quad (11)$	$m_{2 \rightarrow b}(s_2) = p(s_2)m_{a \rightarrow 2}(s_2) \quad (15)$
$m_{b \rightarrow 3}(s_3) = \int_{s_2} p(y_b s_2, s_3)m_{2 \rightarrow b}(s_2)ds_2 \quad (12)$	$m_{3 \rightarrow b}(s_3) = p(s_3) \quad (16)$

Now, we establish that this message passing will result in the marginal of s_1, s_2 and s_3 . The marginal of s_1 is simplified below in Eq. 22.

$$p(s_1|y_a, y_b) \propto p(s_1, y_a, y_b) \quad (17)$$

$$= \int_{s_2} \int_{s_3} p(s_1, s_2, s_3, y_a, y_b)ds_3ds_2 \quad (18)$$

$$= \int_{s_2} \int_{s_3} p(y_a, y_b|s_1, s_2, s_3)p(s_1, s_2, s_3)ds_3ds_2 \quad (19)$$

$$= \int_{s_2} \int_{s_3} p(y_a|s_1, s_2)p(y_b|s_2, s_3)p(s_1)p(s_2)p(s_3)ds_3ds_2 \quad (20)$$

$$= p(s_1) \int_{s_2} \int_{s_3} p(y_a|s_1, s_2)p(y_b|s_2, s_3)p(s_2)p(s_3)ds_3ds_2 \quad (21)$$

$$= p(s_1) \int_{s_2} p(y_a|s_1, s_2)p(s_2) \int_{s_3} p(y_b|s_2, s_3)p(s_3)ds_3ds_2 \quad (22)$$

We see that after the following 4 message passes $3 \rightarrow b, b \rightarrow 2, 2 \rightarrow a, a \rightarrow 1$, we get the marginal of s_1 .

$$p(s_1|y_a, y_b) \propto p(s_1) \overbrace{\int_{s_2} p(y_a|s_1, s_2) \overbrace{\int_{s_3} \overbrace{p(y_b|s_2, s_3) p(s_3)}^{m_{3 \rightarrow b}(s_3)} ds_3}^{m_{2 \rightarrow a}(s_2)} ds_2}^{m_{a \rightarrow 1}(s_1)} \quad (23)$$

Similarly, marginal of s_2 can be written as follows,

$$p(s_2|y_a, y_b) \propto p(s_2) \int_{s_1} p(y_a|s_1, s_2)p(s_1)ds_1 \int_{s_3} p(y_b|s_2, s_3)p(s_3)ds_3ds_2 \quad (24)$$

and after the following 4 message passes $3 \rightarrow b, b \rightarrow 2, 1 \rightarrow a, a \rightarrow 2$, we get the marginal of s_2 .

$$p(s_2|y_a, y_b) \propto p(s_2) \overbrace{\int_{s_1} p(y_a|s_1, s_2) \underbrace{p(s_1)}_{m_{1 \rightarrow a}(s_1)} ds_1}^{m_{a \rightarrow 1}(s_1)} \overbrace{\int_{s_3} p(y_b|s_2, s_3) \underbrace{p(s_3)}_{m_{3 \rightarrow b}(s_3)} ds_3}^{m_{b \rightarrow 2}(s_2)} \quad (25)$$

5 Approximate Message Passing

Our goal is to compute the marginal distribution of the following posterior distribution,

$$p_1(\mathbf{s}|\mathbf{y}) \propto \prod_{i=1}^N \exp(-\beta|s_i|) \prod_{a=1}^n \delta(y_a = \mathbf{A}_{a,:}\mathbf{s}) \quad (26)$$

We define a factor graph with $\{s_i\}_{i=1}^N$ as variables and $\{y_a\}_{a=1}^n$ as factors. From the posterior distribution, it is easy to see that every y_a depends on all s_i 's. Therefore, in the factor graph each y_a is connected to all the s_i 's, i.e. the factor graph is a fully connected bipartite graph where each factor is connected to all variables. Using the belief propagation algorithm, we can compute marginal distributions of all variables s_i . A direct application of Eq. 5 and 6, however, is not possible because of the following reasons:

1. The marginal distributions $p(s_i|\mathbf{y})$ are not Gaussians since the likelihood $p(\mathbf{y}|\mathbf{s})$ is not conjugate to the prior distribution $p(\mathbf{s})$. Similarly, messages are also non-Gaussian and it is not clear how to parameterize them.
2. Number of messages that need to be propagated every iteration is in $O(nN)$ since every variable sends n messages to every factor (and vice-versa).

Problem (1) can be solved by approximating the messages by Gaussians using Lemma 5.1, 5.2 and 5.3. Problem (2) can be solved by using Lemma 5.4, which makes more approximations on messages to make them independent of the sink of the messages. We will now describe these lemmas briefly. We will leave the exact description of "approximations" in these lemmas and focus on intuitive explanations; please see [Mal11] for a detailed description.

For problem (1), it turns out that if the third moment of a message is bounded then a Gaussian approximation is a reasonable one. This is shown in next two lemmas. The following lemma assumes that if messages from variables to factor have their third moment bounded, then messages from factor to variables can be approximated by Gaussians. This lemma can be proved by using Eq. 6 and applying the Berry-Eseen central limit theorem.

Lemma 5.1. *Let us denote the mean and variance of the messages $m_{j \rightarrow a}(s_j)$ by X_{ja} and T_{ja}/β and assume that their third moment is bounded, then messages $m_{a \rightarrow i}(s_i)$ are "close" to the Gaussian distribution given in Eq. 27, defined through the mean parameter M_{ai} and variance parameter V_{ai} given in Eq. 28 and 29.*

$$m_{a \rightarrow i} \approx \mathcal{N}\left(\frac{M_{ai}}{A_{ai}}, \frac{V_{ai}}{\beta A_{ai}^2}\right) \quad (27)$$

$$M_{ai} := y_a - \mathbf{A}_{a,-i} \mathbf{X}_{-i,a} \quad (28)$$

$$V_{ai} := \mathbf{A}_{a,-i} \text{diag}(\mathbf{T}_{-i,a}) \mathbf{A}_{a,-i}^T \quad (29)$$

The following lemma shows that if messages from factors to variables are Gaussians, then message from variables to factors will follow a simple distribution. This lemma can be proved by a direct application of 5.

Algorithm 1 Message passing algorithm for the basis-pursuit problem

Require: Measurements $\mathbf{y}$ and matrix $\mathbf{A}$ **Ensure:** Marginals of the distribution Eq. 3

```

 $X_{ai} \leftarrow 0, \forall a, i$  and  $v = 1$ 
repeat
  for  $a = 1, 2, \dots, n$  do
    for  $i = 1, 2, \dots, N$  do
       $M_{ai} \leftarrow y_a - \mathbf{A}_{a,-i} \mathbf{X}_{-i,a}$ 
       $v \leftarrow \frac{v}{N} \sum_{i=1}^N \eta' \left( \mathbf{A}_{:,i}^T \mathbf{M}_{:,i}, v \right)$ 
    end for
  end for
  for  $a = 1, 2, \dots, n$  do
    for  $i = 1, 2, \dots, N$  do
       $X_{ia} \leftarrow \eta \left( \mathbf{A}_{-a,i}^T \mathbf{M}_{-a,i}, v \right)$ 
    end for
  end for
until convergence
  
```

Lemma 5.2. Assuming that each $m_{a \rightarrow i}(s_i)$ follows the Gaussian distribution defined in Eq. 27, the messages $m_{i \rightarrow a}(s_i)$ follow a distribution given in Eq. 30 which is defined through a distribution defined in Eq. 31.

$$m_{i \rightarrow a}(s_i) \approx p_\beta \left(s_i | \mathbf{A}_{-a,i}^T \mathbf{M}_{-a,i}, V_{ai} \right) \quad (30)$$

$$p_\beta(s | \mu, \sigma^2) \propto \exp \left[-\beta |s| - \frac{\beta}{2\sigma^2} (s - \mu)^2 \right] \quad (31)$$

A simple algorithm is to represent these messages by only first two moments. We can start the distribution from variables to factor $m_{j \rightarrow a}$ to a standard Gaussian, i.e. $X_{ja} = 0$ and $T_{ja} = 1$, then iterate as follows:

$$M_{ai} \leftarrow y_a - \mathbf{A}_{a,-i} \mathbf{X}_{-i,a} \quad (32)$$

$$V_{ai} \leftarrow \mathbf{A}_{a,-i} \text{diag}(\mathbf{T}_{-i,a}) \mathbf{A}_{a,-i}^T \quad (33)$$

$$X_{ia} \leftarrow \text{Mean} \left[p_\beta \left(s_i | \mathbf{A}_{-a,i}^T \mathbf{M}_{-a,i}, V_{ai} \right) \right] \quad (34)$$

$$T_{ia} \leftarrow \text{Variance} \left[p_\beta \left(s_i | \mathbf{A}_{-a,i}^T \mathbf{M}_{-a,i}, V_{ai} \right) \right] \quad (35)$$

This algorithm can be simplified further by assuming that V_{ai} is equal to a constant v for all a, i , then replacing $\mathbf{A}_{-a,i}^T \mathbf{M}_{-a,i}$ by $\mathbf{A}_{:,i}^T \mathbf{M}_{:,i}$ in Eq. 35 and then approximating Eq. 33 by a sample average.

$$M_{ai} \leftarrow y_a - \mathbf{A}_{a,-i} \mathbf{X}_{-i,a} \quad (36)$$

$$v \leftarrow \frac{1}{N} \sum_{i=1}^N \text{Variance} \left[p_\beta \left(s_i | \mathbf{A}_{:,i}^T \mathbf{M}_{:,i}, v \right) \right] \quad (37)$$

$$X_{ia} \leftarrow \text{Mean} \left[p_\beta \left(s_i | \mathbf{A}_{-a,i}^T \mathbf{M}_{-a,i}, v \right) \right] \quad (38)$$

Next lemma shows that in the limit as $\beta \rightarrow \infty$, computation of mean and variance can be done by a simple soft-thresholding function.

Lemma 5.3. For bounded μ and σ^2 ,

$$\lim_{\beta \rightarrow \infty} \text{Mean} [p_\beta(s | \mu, \sigma^2)] = \eta(\mu, \sigma^2) \quad (39)$$

$$\lim_{\beta \rightarrow \infty} \text{Variance} [p_\beta(s | \mu, \sigma^2)] = \sigma^2 \eta'(\mu, \sigma^2) \quad (40)$$

Algorithm 2 Approximate message passing algorithm for the basis-pursuit problem

Require: Measurements $\mathbf{y}$ and matrix $\mathbf{A}$ **Ensure:** Marginals of the distribution Eq. 3

```

 $\mathbf{x} \leftarrow 0, \mathbf{m} \leftarrow 0$  and  $v = 1$ 
repeat
   $\mathbf{x}^{old} \leftarrow \mathbf{x}$ 
   $v^{old} \leftarrow v$ 
   $\mathbf{t} \leftarrow \mathbf{A}^T \mathbf{m}$ 
   $\mathbf{x} \leftarrow \eta(\mathbf{t} + \mathbf{x}, v)$ 
   $v \leftarrow \frac{v}{\delta} \langle \eta'(\mathbf{t} + \mathbf{x}, v) \rangle$ 
   $\mathbf{m} \leftarrow \mathbf{y} - \mathbf{A}\mathbf{x} + \frac{1}{\delta} \mathbf{m} * \langle \eta'(\mathbf{t} + \mathbf{x}^{old}, v^{old}) \rangle$ 
until convergence
  
```

where $\eta(\mu, v)$ is the soft-threshold function where takes a value $\mu - v$ if $\mu > v$ or $\mu + v$ if $\mu < -v$ and zero elsewhere, $\eta'(\mu, v)$ is the derivative of $\eta(\mu, v)$.

Using this, we get the following message passing algorithm shown in Algorithm 1. Although this algorithm is simple, we still have too many messages. Each of these steps require matrix multiplication which needs to be done for all variables and factors. The following lemma shows that given a certain asymptotic behavior, a message can be approximated by another message that is independent of the sink, i.e. independent of the variable/factor that the message is sent to. This lemma can be derived by simply substituting the assumptions of Eq. 41 and 42 in the message passing iterations of Algorithm 1, and then simplifying by removing the term which are $O(1/N)$.

Lemma 5.4. Denote the messages at k 'th iteration with a subscript (k) . Let us assume that the messages at k 'th iteration follow the following asymptotic behavior:

$$X_{ia}^{(k)} = x_i^{(k)} + \delta X_{ia}^{(k)} + O(1/N) \quad (41)$$

$$M_{ai}^{(k)} = m_a^{(k)} + \delta M_{ai}^{(k)} + O(1/N) \quad (42)$$

with $\delta X_{ia}^{(k)}, \delta M_{ai}^{(k)} = O(1/N)$, then variable $x_i^{(k)}$ and $m_a^{(k)}$ satisfy the following,

$$x_i^{(k)} = \eta \left(\mathbf{A}_{:,i}^T \mathbf{m}^{(k-1)} + x_i^{(k-1)}, v^{(k-1)} \right) + o_N(1) \quad (43)$$

$$m_a^{(k)} = y_a - \mathbf{A}_{a,:} \mathbf{x}^{(k)} + \frac{1}{\delta} m_a^{(k-1)} \left\langle \eta' \left(\mathbf{A}^T \mathbf{m}^{(k-1)} + \mathbf{x}^{(k-1)}, v^{(k-1)} \right) \right\rangle + o_N(1) \quad (44)$$

$$v^{(k)} = \frac{v^{(k-1)}}{\delta} \left\langle \eta' \left(\mathbf{A}^T \mathbf{m}^{(k-1)} + \mathbf{x}^{(k)}, v^{(k-1)} \right) \right\rangle \quad (45)$$

where $o_N(1)$ terms vanish as $N, n \rightarrow \infty$.

Using this lemma, we can simplify Algorithm 1 to obtain Algorithm 2.

References

- [Bis06] C. Bishop. *Pattern recognition and machine learning*. Springer, 2006.
- [Mal11] A. Maleki. *Approximate message passing algorithms for compressed sensing*. PhD thesis, Stanford University, 2011.
- [See08] M. Seeger. Bayesian Inference and Optimal Design in the Sparse Linear Model. *J. of Machine Learning Research*, 9:759–813, 2008.

Deep MIMO Detection

Neev Samuel, Tzvi Diskin and Ami Wiesel

School of Computer Science and Engineering
The Hebrew University of Jerusalem
The Edmond J. Safra Campus
9190416 Jerusalem, Israel

Abstract—In this paper, we consider the use of deep neural networks in the context of Multiple-Input-Multiple-Output (MIMO) detection. We give a brief introduction to deep learning and propose a modern neural network architecture suitable for this detection task. First, we consider the case in which the MIMO channel is constant, and we learn a detector for a specific system. Next, we consider the harder case in which the parameters are known yet changing and a single detector must be learned for all multiple varying channels. We demonstrate the performance of our deep MIMO detector using numerical simulations in comparison to competing methods including approximate message passing and semidefinite relaxation. The results show that deep networks can achieve state of the art accuracy with significantly lower complexity while providing robustness against ill conditioned channels and mis-specified noise variance.

Keywords—MIMO Detection, Deep Learning, Neural Networks.

I. INTRODUCTION

Multiple input multiple output (MIMO) systems arise in most modern communication channels. The dimensions can account for time and frequency resources, multiple users, multiple antennas and other resources. These promise substantial performance gains, but present a challenging detection problem in terms of computational complexity. In recent years, the world is witnessing a revolution in deep machine learning. In many fields of engineering, e.g., computer vision, it was shown that computers can be fed with sample pairs of inputs and desired outputs, and “learn” the functions which relates them. These rules can then be used to classify (detect) the unknown outputs of future inputs. The goal of this paper is to apply deep machine learning in the classical MIMO detection problem and understand its advantages and disadvantages.

A. Background on MIMO detection

The binary MIMO detection setting is a classical problem in simple hypothesis testing [1]. The maximum likelihood (ML) detector is the optimal detector in the sense of minimum joint probability of error for detecting all the symbols simultaneously. It can be implemented via efficient search algorithms, e.g., the sphere decoder [2]. The difficulty is that its worst case computational complexity is impractical for many applications. Consequently, several modified search algorithms have been proposed, offering improved complexity performance [3][4]. There has been much interest in implementing suboptimal detection algorithms. The most common suboptimal detectors are the linear receivers, i.e., the matched filter (MF), the decorrelator or zero forcing (ZF) detector and the minimum mean

squared error (MMSE) detector. More advanced detectors are based on decision feedback equalization (DFE), approximate message passing (AMP) [5] and semidefinite relaxation (SDR) [6], [7]. Currently, both AMP and SDR provide near optimal accuracy under many practical scenarios. AMP is simple and cheap to implement in practice, but is an iterative method that may diverge in problematic settings. SDR is more robust and has polynomial complexity, but is much slower in practice.

B. Background on Machine Learning

In the last decade, there is an explosion of machine learning success stories in all fields of engineering. Supervised classification is similar to statistical detection theory. Both observe noisy data and output a decision on the discrete unknown it originated from. Typically, the two fields differ in that detection theory is based on a prior probabilistic model of the environment, whereas learning is data driven and is based on examples. In the context of MIMO detection, a model is known and allows us to generate as many synthetic examples as needed. Therefore we adapt an alternative notion. We interpret “learning” as the idea of choosing a best decoder from a prescribed class of algorithms. Classical detection theory tries to choose the best estimate of the unknowns, whereas machine learning tries to choose the best algorithm to be applied. Indeed, the hypotheses in detection are the unknown symbols, whereas the hypotheses in learning are the detection rules [8]. Practically, this means that the computationally involved part of detection is applied every time we get a new observation. In learning, the expensive stage is learning the algorithm which is typically performed off line. Once the optimal rule algorithm is found, we can cheaply implement it in real time.

Machine learning has a long history but was previously limited to simple and small problems. Fast forwarding to the last years, the field witnessed the deep revolution. The “deep” adjective is associated with the use of complicated and expressive classes of algorithms, also known as architectures. These are typically neural networks with many non-linear operations and layers. Deep architectures are more expressive than shallow ones [9], but were previously considered impossible to optimize. With the advances in big data, optimization algorithms and stronger computing resources, such networks are currently state of the art in different problems including speech processing and computer vision. In particular, one promising approach to designing deep architectures is by unfolding an existing iterative algorithm [10]. Each iteration is considered a layer and the algorithm is called a network. The learning begins with the existing algorithm as an initial starting point and uses

optimization methods to improve the algorithm. For example, this strategy has been shown successful in the context of sparse reconstruction. Leading algorithms as Iterative Shrinkage and Thresholding and a sparse version of AMP have both been improved by unfolding their iterations into a network and learning their optimal parameters [11], [12].

In recent years, deep learning methods have been purposed for improving the performance of a decoder for linear codes in fixed channels[13]. And in [14] several applications of deep learning for communication applications have been considered, including decoding signals over fading channels, but the architecture purposed there does not seem to be scalable for higher dimension signals.

C. Main contributions

The main contribution of this paper is the introduction of DetNET, a deep learning network for MIMO detection. DetNet is derived by unfolding a projected gradient descent method. Simulations show that it achieves near optimal detection performance while being a fast algorithm that can be implemented in real-time. Its accuracy is similar to SDR with running time that is more than 30 times faster. Compared to AMP, another detector with optimality guarantees, DetNet is more robust. It shows promising performance in handling ill conditioned channels, and does not require knowledge of the noise variance.

Another important contribution, in the general context of deep learning, is DetNet's ability to perform on multiple models with a single training. Recently, there were many works on learning to invert linear channels and reconstruct signals [11], [12], [15]. To the best of our knowledge, all of these were developed and trained to address a single fixed channel. In contrast, DetNet is designed for handling multiple channels simultaneously with a single training phase.

D. Notation

In this paper, we shall define the normal distribution where μ is the mean and σ^2 is the variance as $\mathcal{N}(\mu, \sigma^2)$. The uniform distribution with the minimum value a and the maximum value b will be $\mathcal{U}(a, b)$. Boldface uppercase letters denote matrices, Boldface lowercase letters denote vectors, the superscript $(\cdot)^T$ denotes the transpose. The i 'th element of the vector $\mathbf{x}$ will be denoted as x_i . Unless stated otherwise, the term independent and identically distributed (i.i.d.) Gaussian matrix, will refer to a matrix where each of its elements is i.i.d. sampled from the normal distribution $\mathcal{N}(0, 1)$. The rectified linear unit defined as $\rho(x) = \max\{0, x\}$ will be denoted as ρ .

II. LEARNING TO DETECT

In this section, we formulate the MIMO detection problem in a machine learning framework. We consider the standard linear MIMO model:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}, \quad (1)$$

where $\mathbf{y} \in \mathbb{R}^N$ is the received vector, $\mathbf{H} \in \mathbb{R}^{N \times K}$ is the channel matrix, $\mathbf{x} \in \{\pm 1\}^K$ is an unknown vector of independent and equal probability binary symbols, $\mathbf{w} \in \mathbb{R}^N$ is a noise vector with independent, zero mean Gaussian variables

of variance σ^2 . We do not assume knowledge of the variance as hypothesis testing theory guarantees that this is unnecessary for optimal detection. Indeed, the optimal ML rule does not require knowledge of σ^2 . This is contrast to the MMSE and AMP decoders that exploit this parameter and are therefore less robust.

We assume perfect channel state information (CSI) and that the channel $\mathbf{H}$ is exactly known. However, we differentiate between two possible cases:

- Fixed Channel (FC): In the FC scenario, $\mathbf{H}$ is deterministic and constant (or a realization of a degenerate distribution which only takes a single value).
- Varying Channel (VC): In the VC scenario, we assume $\mathbf{H}$ random with a known distribution.

Our goal is to detect $\mathbf{x}$, using an algorithm that receives $\mathbf{y}$ and $\mathbf{H}$ as inputs and estimates $\hat{\mathbf{x}}$.

The first step is choosing and fixing a detection architecture. An architecture is a function $\hat{\mathbf{x}}_\theta(\mathbf{H}, \mathbf{y})$ that detects the unknown symbols given $\mathbf{y}$ and $\mathbf{H}$. The architecture is parametrized by θ . Learning is the problem of finding the θ within the feasible set that will lead to strong detectors $\hat{\mathbf{x}}_\theta(\mathbf{H}, \mathbf{y})$. By choosing different functions and parameter sets, we characterize competing types of detectors which tradeoff accuracy with complexity.

To find the best detector, we fix a loss function $l(\mathbf{x}; \hat{\mathbf{x}}_\theta(\mathbf{H}, \mathbf{y}))$ that measures the distance between the true symbols and their estimates. Then, we find θ by minimizing the loss function we chose over the MIMO model distribution:

$$\min_{\theta} \mathbb{E} \{l(\mathbf{x}; \hat{\mathbf{x}}_\theta(\mathbf{H}, \mathbf{y}))\}, \quad (2)$$

where the expectation is with respect to all the random variables in (1), i.e., $\mathbf{x}$, $\mathbf{w}$, and $\mathbf{H}$. Learning to detect is defined as finding the best set of parameters θ of the architecture $\hat{\mathbf{x}}_\theta(\mathbf{y}, \mathbf{H})$ that minimize the expected loss $l(\cdot; \cdot)$ over the distribution in (1).

The next examples illustrate how the choice of architecture $\hat{\mathbf{x}}_\theta(\mathbf{y}, \mathbf{H})$ leads to different detectors that tradeoff accuracy for complexity.

Example 1: The goal in detection is to decrease the probability of error. Therefore, the best loss function in this problem

$$l(\mathbf{x}; \hat{\mathbf{x}}_\theta(\mathbf{H}, \mathbf{y})) = \begin{cases} 1 & \mathbf{x} \neq \hat{\mathbf{x}}_\theta(\mathbf{y}, \mathbf{H}) \\ 0 & \text{else.} \end{cases} \quad (3)$$

By choosing an unrealistically flexible architecture with unbounded parameterization and no restrictions such that

$$\{\hat{\mathbf{x}}_\theta(\mathbf{y}, \mathbf{H}) : \theta\} = \left\{ \begin{array}{l} \text{all the functions} \\ \mathbb{R}^N \times \mathbb{R}^{N \times K} \mapsto \{\pm 1\}^K \end{array} \right\}. \quad (4)$$

Then, the solution to (2) is the ML decoder:

$$\hat{\mathbf{x}}_\theta(\mathbf{y}, \mathbf{H}) = \arg \min_{\mathbf{x} \in \{\pm 1\}^K} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2. \quad (5)$$

This rule is optimal in terms of accuracy but requires a computationally intensive search of $O(2^K)$. Obviously, this example is theoretical since the architecture of all possible functions cannot be parametrized and (2) cannot be optimized.

Example 2: On the other extreme, consider the architecture of fixed linear detectors:

$$\hat{\mathbf{x}}_{\theta}(\mathbf{y}, \mathbf{H}) = \mathbf{A}\mathbf{y}, \quad (6)$$

where the parameter θ is a single fixed matrix to be optimized within $\mathbb{R}^{K \times N}$. In the FC model, choosing $\|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{y}, \mathbf{H})\|^2$ as the loss function and assuming $\sigma^2 \rightarrow 0$, the optimal decoder is the well known decorrelator:

$$\hat{\mathbf{x}}_{\theta}(\mathbf{y}, \mathbf{H}) = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y}. \quad (7)$$

The resulting detector involves a simple matrix multiplication that requires $O(NK)$ operations, but is not very accurate. On the other hand, if we consider the more challenging VC model, then the optimal linear transformation is simply $\mathbf{H} = \mathbf{0}$. A single linear decoder cannot decode arbitrary channels simultaneously, and the decoder is completely useless.

These two examples emphasize how fixing an architecture and a loss function determines what will be the optimal detector for the MIMO detection problem. The more expressive we choose $\hat{\mathbf{x}}$ to be, the more accurate the final detector can be, on the expense of the computational complexity.

We close this section with a technical note on the numerical implementation of the optimization in (2). In practice, it is intractable to optimize over an abstract class of functions as in (4). Numerical minimization is typically performed with respect to a finite parameter set as in (6). Thus, our deep architectures are based on multiple layers with multivariate linear operations and element-wise non-linear operators. These allow rich decoding functions while resorting to a finite and tractable parameterization. In addition, analytic computation of the expectation in the objective is usually impossible. Instead, we approximate it using an empirical mean of samples drawn from a data set of examples (thus the 'learning' notion). In our case, the data set is composed of synthetically generated samples satisfying (1). Both these technicalities, were considered unthinkable just a decade ago, but are now standard procedures in the deep learning community. Easy to use, open source tools, make it possible to create deep architectures and optimize them in a straight forward manner. Specifically, in this work, all the experiments were implemented on the TensorFlow framework [16].

III. DEEP MIMO DETECTOR

In this section, we propose a deep detector with an architecture which is specifically designed for MIMO detection that will be named from now on 'DetNet' (Detection Network). First, we note that an efficient detector should not work with $\mathbf{y}$ directly, but use the compressed sufficient statistic:

$$\mathbf{H}^T \mathbf{y} = \mathbf{H}^T \mathbf{H} \mathbf{x} + \mathbf{H}^T \mathbf{w}. \quad (8)$$

This hints that two main ingredients in the architecture should be $\mathbf{H}^T \mathbf{y}$ and $\mathbf{H}^T \mathbf{H} \mathbf{x}$. Second, our construction is based on mimicking a projected gradient descent like solution for the ML optimization in (5). Such an algorithm would lead to iterations of the form

$$\begin{aligned} \hat{\mathbf{x}}_{k+1} &= \Pi \left[\hat{\mathbf{x}}_k - \delta_k \frac{\partial \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}_k} \right] \\ &= \Pi \left[\hat{\mathbf{x}}_k - \delta_k \mathbf{H}^T \mathbf{y} + \delta_k \mathbf{H}^T \mathbf{H} \hat{\mathbf{x}}_k \right], \end{aligned} \quad (9)$$

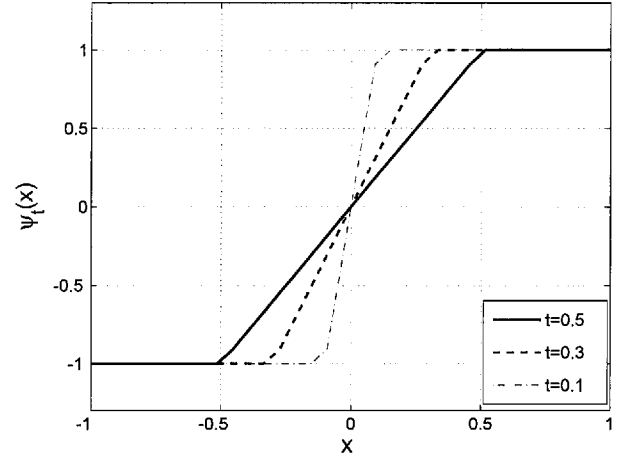


Fig. 1. A graph illustrating the linear soft sign function $\psi_t(\mathbf{x})$ for different values of the parameter t .

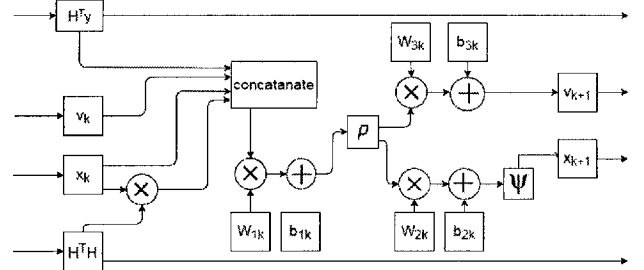


Fig. 2. A flowchart representing a single layer of DetNet.

where $\hat{\mathbf{x}}_k$ is the estimate in the k 'th iteration, $\Pi[\cdot]$ is a nonlinear projection operator, and δ_k is a step size. Intuitively, each iteration is a linear combination of the $\mathbf{x}_k$, $\mathbf{H}^T \mathbf{y}$, and $\mathbf{H}^T \mathbf{H} \mathbf{x}_k$ followed by a non-linear projection. We enrich these iterations by lifting the input to a higher dimension and applying standard non-linearities which are common in deep neural networks. This yields the following architecture:

$$\begin{aligned} \mathbf{z}_k &= \rho \left(\mathbf{W}_{1k} \begin{bmatrix} \mathbf{H}^T \mathbf{y} \\ \hat{\mathbf{x}}_k \\ \mathbf{H}^T \mathbf{H} \hat{\mathbf{x}}_k \\ \mathbf{v}_k \end{bmatrix} + \mathbf{b}_{1k} \right) \\ \hat{\mathbf{x}}_{k+1} &= \psi_{t_k}(\mathbf{W}_{2k} \mathbf{z}_k + \mathbf{b}_{2k}) \\ \hat{\mathbf{v}}_{k+1} &= \mathbf{W}_{3k} \mathbf{z}_k + \mathbf{b}_{3k} \\ \hat{\mathbf{x}}_1 &= \mathbf{0}, \end{aligned} \quad (10)$$

where $k = 1, \dots, L$ and $\psi_t(\cdot)$ is a piecewise linear soft sign operator defined as:

$$\psi_t(x) = -1 + \frac{\rho(x+t)}{|t|} - \frac{\rho(x-t)}{|t|}. \quad (11)$$

The operator is plotted in Fig. 1, and the structure of each DetNet layer is illustrated in Fig. 2. The final estimate is defined as $\hat{\mathbf{x}}_{\theta}(\mathbf{y}, \mathbf{H}) = \text{sign}(\hat{\mathbf{x}}_L)$.

The parameters of DetNet that are optimized during the

learning phase are:

$$\theta = \{W_{1k}, b_{1k}, W_{2k}, b_{2k}, W_{3k}, b_{1k}, t_k\}_{k=1}^L. \quad (12)$$

Training deep networks is a difficult task due to vanishing gradients, saturation of the activation functions, sensitivity to initializations and more [17]. To address these challenges, we adopted a loss function that takes into account the outputs of all of the layers. Moreover, since the errors depend on the channel's realization, we decided to normalize the errors with those of the decorrelator. Together, this led to the following loss function:

$$l(x; \hat{x}_\theta(H, y)) = \sum_{k=1}^L \log(k) \frac{\|x - \hat{x}_k\|^2}{\|x - \tilde{x}\|^2}, \quad (13)$$

where:

$$\tilde{x} = (H^T H)^{-1} H^T y. \quad (14)$$

is the standard decorrelator decoder.

In our final implementation, in order to further enhance the performance of DetNet, we added a residual feature from ResNet [18] where the output of each layer is a weighted average with the output of the previous layer. Note also that our loss function is motivated by the auxiliary classifiers feature in GoogLeNet [19].

We train the network using a variant of the stochastic gradient descent method [20],[21] for optimizing deep networks, named Adam Optimizer [22]. We used batch training with 5000 random data samples at each iteration, and trained the network for 50000 iterations. To give a rough idea of the complexity, learning the detectors in our numerical results took 2 days on a standard Intel i7-6700 processor. Each sample was independently generated from (1) according to the statistics of x , H (either in the FC or VC model) and w . With respect to the noise, its variance is unknown and therefore this too was randomly generated so that the SNR will be uniformly distributed on $\mathcal{U}(\text{SNR}_{\min}, \text{SNR}_{\max})$. This approach allows us to train the network to detect over a wide range of SNR values.

IV. NUMERICAL RESULTS

In this section, we demonstrate the advantages of our proposed detector using computer simulations.

All the experiments address a MIMO channel with an input of size $K = 30$ and output of size $N = 60$. It is well known that performance is highly dependent on the type of MIMO channel. Therefore, we tried two scenarios:

FC: In this model, we chose to test the algorithms on a deterministic and constant ill-conditioned matrix which is known to be challenging for detection [23]. The matrix was generated such that $H^T H$ would have a Toeplitz structure with $[H^T H]_{i,j} = 0.55^{|i-j|}$. We shall denote this matrix as the 0.55-Toeplitz matrix. This defines the singular values and right singular vectors of H . Its left singular vectors were randomly generated uniformly in the space of orthogonal matrices, and then fixed throughout the simulations.

VC: In this model, the matrices H were randomly generated with i.i.d. $\mathcal{N}(0, 1)$ elements. Each example was independently generated within the same experiment.

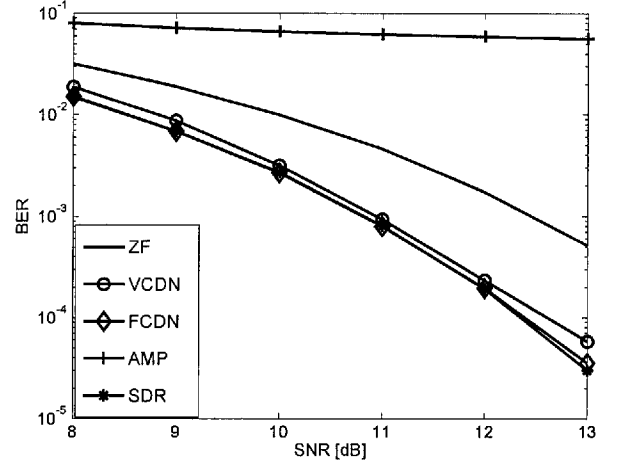


Fig. 3. Comparison of BER performance in the fixed channel case between the detection algorithms. all algorithms were tested on the 0.55-Toeplitz channel.

We have tested the performance of the following detection algorithms:

FCDN: DetNet algorithm described in (10) with $3K$ layers, z_k of size $8K$, and v_k of size $2K$. FCDN was trained using the FC model described above, and is specifically designed to handle a specific ill conditioned channel matrix.

VCDN: Same architecture as the FCDN but the training is on the VC model and is supposed to cope with arbitrary channel matrices.

ShVCDN : Same as the VCDN algorithm, but with a shallow network architecture using only K layers.

ZF: This is the classical decorrelator, also known as least squares or zero forcing (ZF) detector [1].

AMP: Approximate message passing algorithm from [5]. The algorithm was adjusted to the real-valued case and was implemented with $3K$ iterations.

AMP2: Same as the AMP algorithm but with a mis-specified SNR. The SNR in dB has an additional $\mathcal{N}(0, 2)$ bias.

SDR: A decoder based on semidefinite relaxation implemented using a specifically tailored and efficient interior point solver [6], [7].

In our first experiment, we focused on the FC model in which the channel is known and fixed, yet challenging due to its condition number. Figure 3 shows the results of all the algorithms in this setting. FCDN manages to reach the accuracy rates of the computationally expensive SDR algorithm which in our simulations took 30 times longer to detect. AMP does not manage to detect with reasonable accuracy in this challenging channel. It is interesting to notice that VCDN, which was not designed for this challenging channel, also manages to achieve good accuracy. This result indicates that VCDN generalizes itself during the training phase to detect over arbitrary random channels.

In our second experiment which results are presented in figure 4, we examine the performance in the VC model. SDR and AMP are theoretically known to be optimal in this setting, and VCDN manages to provide similar accuracy. Compared

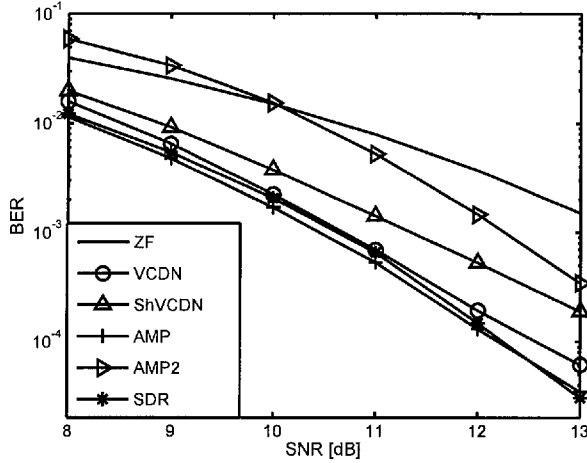


Fig. 4. Comparison of the detection algorithms BER performance in the varying channel case. All algorithms were tested on random i.i.d. Gaussian channels.

to SDR, VCDN runs 30 times faster. Compared to AMP in a scenario where the SNR values are not given accurately, we can notice a negative effect on the accuracy of the AMP, compared to VCDN that does not require any knowledge regarding the SNR.

Another important feature of DetNet is the ability to trade-off complexity and accuracy by adding or removing additional layers. In figure 4 we test the ShVCDN algorithm that is a shallow version on VCDN with only K layers, which is much faster, but less accurate. Since every layer in DetNet outputs a predicted signal $\hat{x}_k$, we can decide in real-time what layer will be the final output layer, and trade-off complexity for accuracy in real-time, without any further training.

V. CONCLUSION

In this paper we have presented deep neural networks as a general framework for MIMO detection. We have tested the performance in the fixed channel scenario over challenging channels, and in the more complicated VC scenario. The DetNet architecture we have suggested has proven to be computationally inexpensive and has near-optimal accuracy without any knowledge regarding the SNR level. The ability of DetNet to optimize over an entire distribution of channels, rather than a single or even a large-finite set of channels, makes it robust and enables implementation in systems where the channel is not fixed. Simulations show that DetNet succeeds to generalize and detect accurately over channels with different characteristics than those of the channels used in the training phase of DetNet. For more details, see [24], where further information is presented.

ACKNOWLEDGMENTS

We would like to thank Prof. Shai Shalev-Shwartz for his help during the research and his insights. This research was partly supported by the Heron Consortium and by ISF grant 1339/15.

REFERENCES

- [1] S. Verdú, *Multuser detection*, Cambridge university press, 1998.
- [2] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE transactions on information theory*, vol. 48, no. 8, pp. 2201–2214, 2002.
- [3] Z. Guo and P. Nilsson, "Algorithm and implementation of the k-best sphere decoding for mimo detection," *IEEE Journal on selected areas in communications*, vol. 24, no. 3, pp. 491–503, 2006.
- [4] S. Suh and J. R. Barry, "Reduced-complexity mimo detection via a slicing breadth-first tree search," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1782–1790, 2017.
- [5] C. Jeon, R. Ghods, A. Maleki, and C. Studer, "Optimality of large mimo detection via approximate message passing," in *IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2015, pp. 1227–1231.
- [6] Z. Q. Luo, W. K. Ma, A. M. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, 2010.
- [7] J. Jaldén and B. Ottersten, "The diversity order of the semidefinite relaxation detector," *IEEE Transactions on Information Theory*, vol. 54, no. 4, pp. 1406–1422, 2008.
- [8] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, New York, NY, USA, 2014.
- [9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [10] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv preprint arXiv:1409.2574*, 2014.
- [11] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 399–406.
- [12] M. Borgerding and P. Schniter, "Onsager-corrected deep learning for sparse linear inverse problems," *arXiv preprint arXiv:1607.05966*, 2016.
- [13] E. Nachmani, Y. Beery, and D. Burshtein, "Learning to decode linear codes using deep learning," *arXiv preprint arXiv:1607.04793*, 2016.
- [14] T. J. O'Shea and j. Hoydis, "An introduction to machine learning communications systems," *arXiv preprint arXiv:1702.00832*, 2017.
- [15] A. Mousavi and R. G. Baraniuk, "Learning to invert: Signal recovery via deep convolutional networks," *arXiv preprint arXiv:1701.03891*, 2017.
- [16] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.
- [17] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Aistats*, 2010, vol. 9, pp. 249–256.
- [18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.
- [20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, pp. 1, 1988.
- [21] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.
- [22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [23] S. Rangan, P. Schniter, and A. Fletcher, "On the convergence of approximate message passing with arbitrary matrices," in *IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2014, pp. 236–240.
- [24] N. Samuel and A. Wiesel, "Learning to detect," *In preparation for a journal*.

WHAT IS CLAIMED IS:

1. A method for staged signal detection, comprising:
 - obtaining a set of received signals;
 - determining a channel matrix via which the set of received signals have been transported; and
 - detecting the set of received signals in a staged manner,
 - wherein in a stage, the method comprises:
 - detecting the set of received signals, based on the channel matrix and a detection algorithm for the stage;
 - fixing one or more of detected symbols output from the detecting algorithm for the stage; and
 - updating the channel matrix and the set of received signals for usage by a next stage, based on the one or more of the detected symbols that are fixed in the stage.
2. The method of Claim 1, wherein the one or more of the detected symbols are determined based on a number that is configurable.
3. The method of Claim 2, further comprising:
 - determining a ratio of symbols to be fixed in a stage; and
 - determining the total number of stages for detecting and the number of symbols to be fixed in the stage, based on the determined ratio of symbols to be fixed.
4. The method of Claim 3, wherein determining the ratio of symbols to be fixed in a stage comprises:
 - determining the ratio of symbols to be fixed in the stage based on a receiving signal to noise ratio, SNR.
5. The method of Claim 3, wherein the total number of stages is determined by:

$$N_{tot} = \lceil -\log_{(1-fRatio)} N_t \rceil + 1$$
 where N_{tot} denotes the total number of stages, fRatio denotes the ratio of symbols to be fixed in a stage, and N_t denotes the number of transmitting antennas.

6. The method of Claim 3, wherein the number of symbols to be fixed in a stage is determined by:

$$\begin{aligned} fNum &= [\text{len}(ds) * fRatio], \text{ or} \\ fNum &= \max(1, [\text{len}(ds) * fRatio]); \end{aligned}$$

where $fNum$ denotes the number of symbols to be fixed in the stage, $\text{len}(ds)$ denotes the total number of input symbols for the stage, $fRatio$ denotes the ratio of symbols to be fixed in the stage, and $[.]$ denotes an operation of getting an integer part of a real number.

7. The method of Claim 1, wherein fixing one or more of the detected symbols comprises:

selecting, for each detected symbol output from the detection algorithm, a candidate symbol from a set of modulation constellation symbols determined by a modulation scheme for the set of received signals, based on a maximum likelihood criterion,

determining a distance between each detected symbol and a corresponding candidate symbol; and

fixing the one or more of the detected symbols associated with the minimum distance to a corresponding candidate symbol respectively.

8. The method of Claim 1, wherein the detection algorithm for a stage includes one of:

- a Zero forcing, ZF, based algorithm,
- a Minimum Mean Square Error, MMSE based algorithm, and
- a gradient descent, GD, based algorithm.

9. The method of Claim 1, wherein the detection algorithm is selected for each stage independently.

10. The method of Claim 1, wherein updating the channel matrix and the set of received signals comprises:

removing, from the channel matrix, one or more columns corresponding to the one or more of the detected symbols that are fixed in a current stage; and

subtracting, from the set of received signals, a value contributed by the one

or more of the detected symbols that are fixed in the current stage.

11. The method of Claim 1, further comprises:

outputting a detection result upon completion of the staged signal detection.

12. The method of Claim 1, further comprises:

outputting the one or more of the detected symbols that are fixed in each stage,
and

if there is a remaining symbol unfixed after completion of the staged signal detection, outputting the remaining symbol.

13. The method of Claim 1, wherein the detected symbols output from the detecting algorithm are soft estimated symbols.

14. A receiver device, comprising:

at least one processor; and

at least one memory including computer program code;

the at least one memory and the computer program code are configured to, with the at least one processor, cause the receiver device at least to:

obtain a set of received signals;

determine a channel matrix via which the set of received signals are transported;

detect the set of received signals in a staged manner, and

in a stage, perform:

detecting the set of received signals, based on the channel matrix and a detection algorithm for the stage;

fixing one or more of detected symbols output from the detecting algorithm for the stage; and

updating the channel matrix and the set of received signals for usage by a next stage, based on the one or more of the detected symbols that are fixed in the stage.

15. The receiver device of Claim 14, wherein the one or more of the detected symbols are determined based on a number that is configurable.

16. The receiver device of Claim 15, wherein the at least one memory and the computer program code are configured to, with the at least one processor, further cause said receiver device to:

determine a ratio of symbols to be fixed in a stage; and

determine the total number of stages and the number of symbols to be fixed in a stage based on the determined ratio of symbols to be fixed.

17 The receiver device of Claim 16, wherein said receiver device is further caused to determine the ratio of symbols to be fixed in a stage based on a receiving signal to noise ratio, SNR.

18. The receiver device of Claim 16, wherein the total number of stages for detecting is determined by:

$$N_{tot} = \lceil -\log_{(1-fRatio)} N_t \rceil + 1$$

where N_{tot} denotes the total number of stages, $fRatio$ denotes the ratio of symbols to be fixed in a stage, and N_t denotes the number of transmitting antennas.

19. The receiver device of Claim 16, wherein the number of symbols to be fixed in a stage is determined by:

$$fNum = \lfloor \text{len}(ds) * fRatio \rfloor, \text{ or}$$

$$fNum = \max(1, \lfloor \text{len}(ds) * fRatio \rfloor);$$

where $fNum$ denotes the number of symbols to be fixed in the stage, $\text{len}(ds)$ denotes the total number of input symbols for the stage, $fRatio$ denotes the ratio of symbols to be fixed in the stage, and $\lfloor . \rfloor$ denotes an operation of getting an interger part of a real number.

20. The receiver device of Claim 14, wherein the receiver device is further caused to fix one or more of the detected symbols by:

selecting, for each detected symbol output from the detection algorithm, a candidate symbol from a set of modulation constellation symbols determined by a modulation scheme for the set of received signals, based on a maximum likelihood criterion,

determining a distance between each detected symbol and a corresponding candidate symbol; and

fixing the one or more of the detected symbols associated with the minimum distance to a corresponding candidate symbol respectively.

21. The receiver device of Claim 14, wherein the detection algorithm for a stage includes one of:

a Zero forcing, ZF, based algorithm,

a Minimum Mean Square Error, MMSE based algorithm, and

a gradient descent, GD, based algorithm.

22. The receiver device of Claim 14, wherein the detection algorithm is selected for each stage independently.

23. The receiver device of Claim 14, wherein the at least one memory and the computer program code are configured to, with the at least one processor, further cause said receiver device to update the channel matrix and the set of received signals by:

removing, from the channel matrix, one or more columns corresponding to the one or more of the detected symbols that are fixed in a current stage; and

subtracting, from the set of received signals, a value contributed by the one or more of the detected symbols that are fixed in the current stage.

24. The receiver device of Claim 14, wherein the at least one memory and the computer program code are configured to, with the at least one processor, further cause said receiver device to:

output a detection result upon completion of the staged signal detection.

25. The receiver device of Claim 14, wherein the at least one memory and the computer program code are configured to, with the at least one processor, further cause said receiver device to:

output the one or more of the detected symbols that are fixed in each stage, and

if there is a remaining symbol unfixed after completion of the staged signal detection, output the remaining symbol.

26. The receiver device of Claim 14, wherein the detected symbols output from the detecting algorithm are soft estimated symbols.

27. A receiver device, comprising:
means for obtaining a set of received signals;
means for determining a channel matrix via which the set of received signals have been transported; and
means for detecting the set of received signals in a staged manner,
wherein in a stage, the means for detecting performs:
detecting the set of received signals, based on the channel matrix and a detection algorithm for the stage;
fixing one or more of detected symbols output from the detecting algorithm for the stage; and
updating the channel matrix and the set of received signals for usage by a next stage, based on the one or more of the detected symbols that are fixed in the stage.

28. A computer readable medium having a computer program stored thereon which, when executed by at least one processor of a device, causes the device to carry out the method of any of claims 1-13.

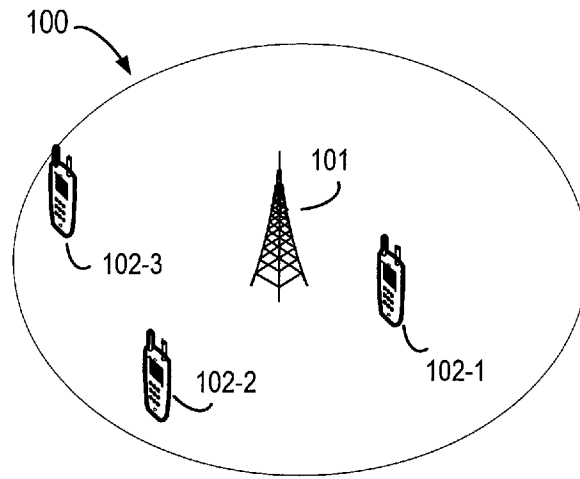


FIG. 1

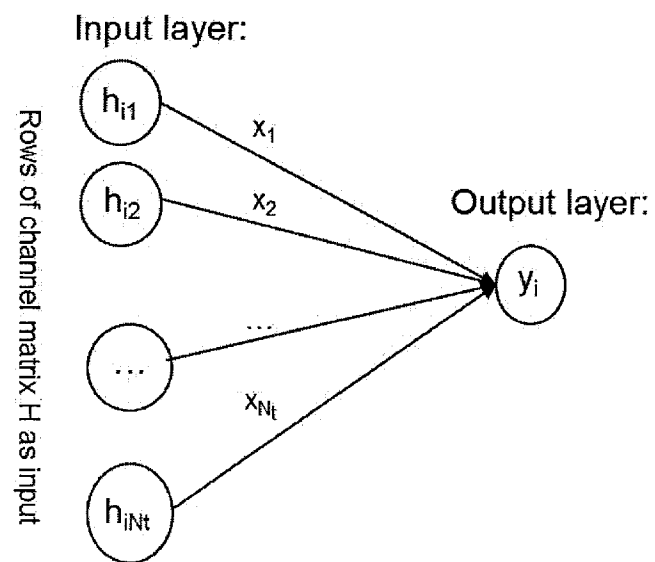


FIG. 2

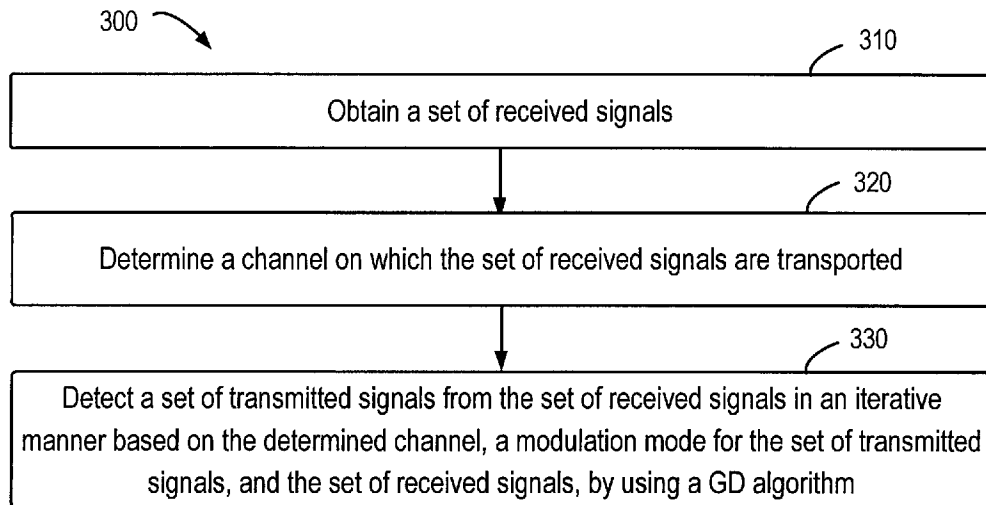


FIG. 3

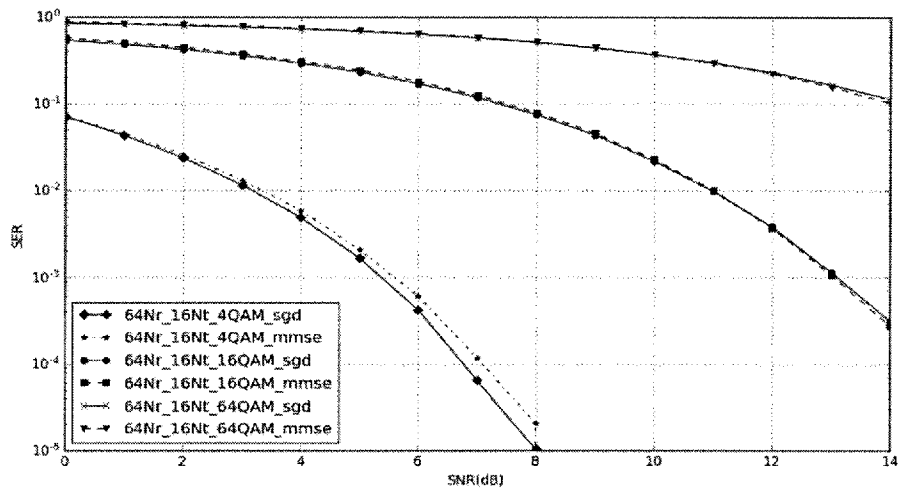


FIG. 4

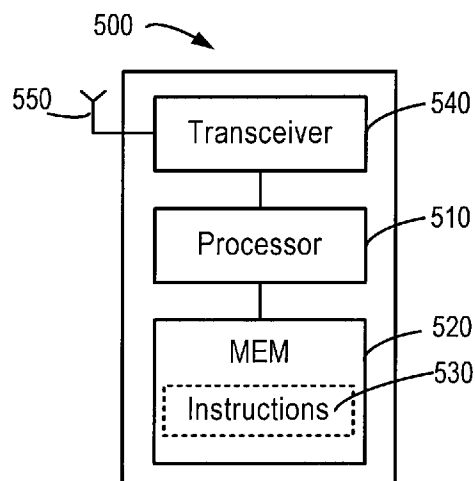


FIG. 5

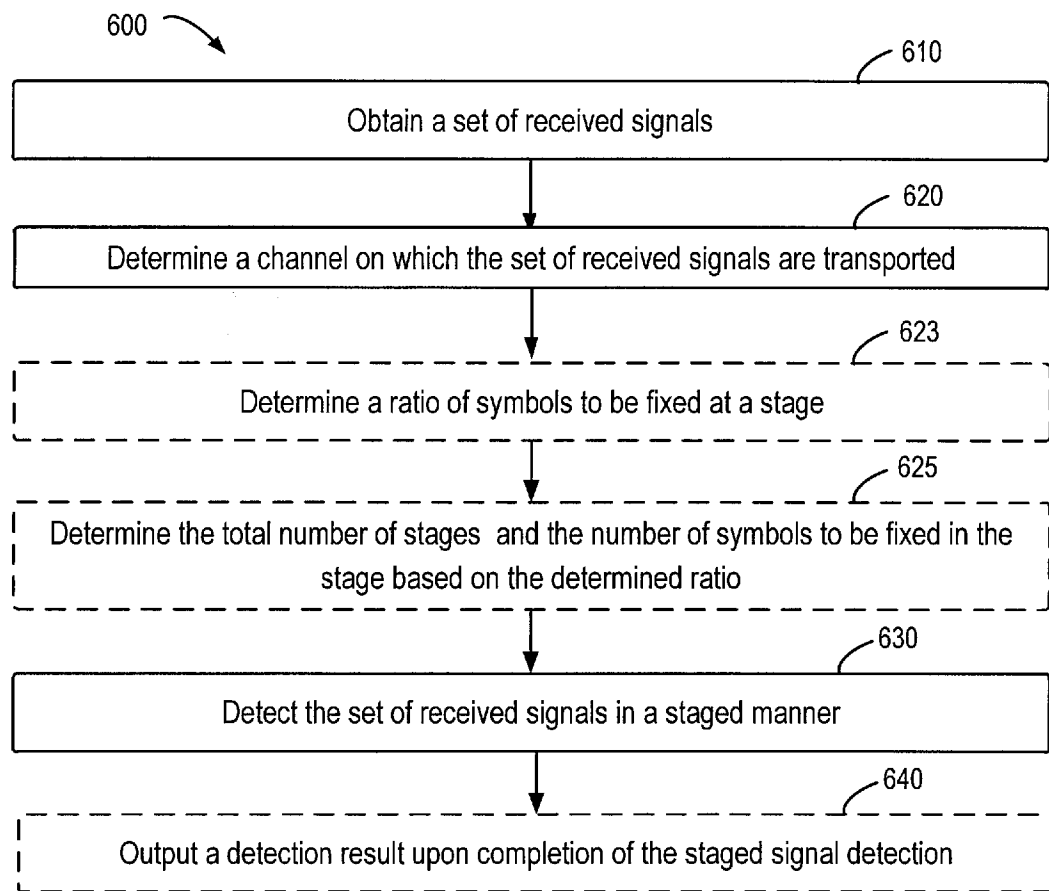


FIG. 6

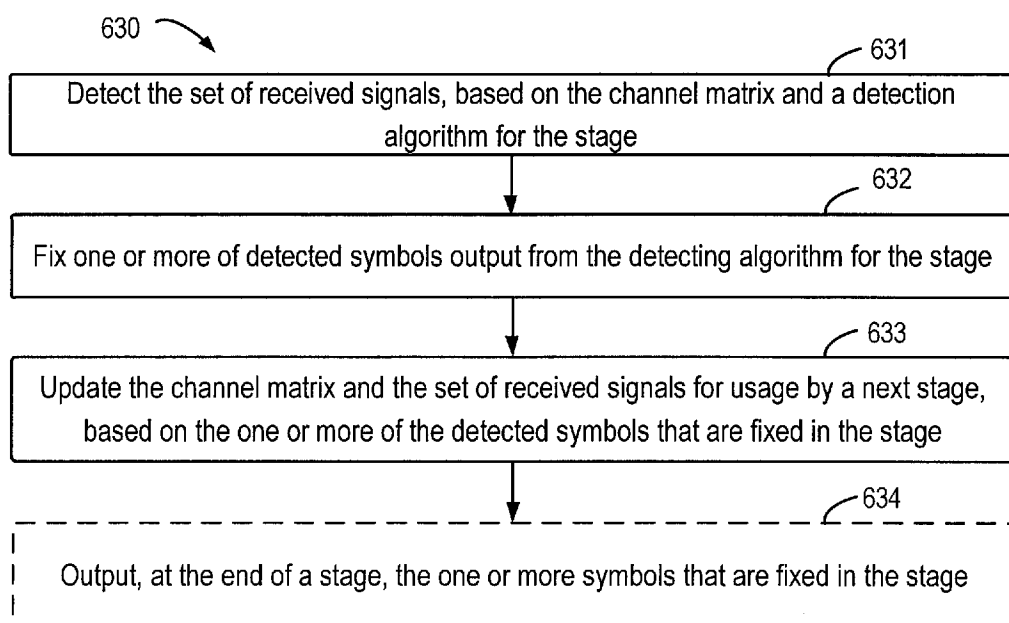


FIG. 7

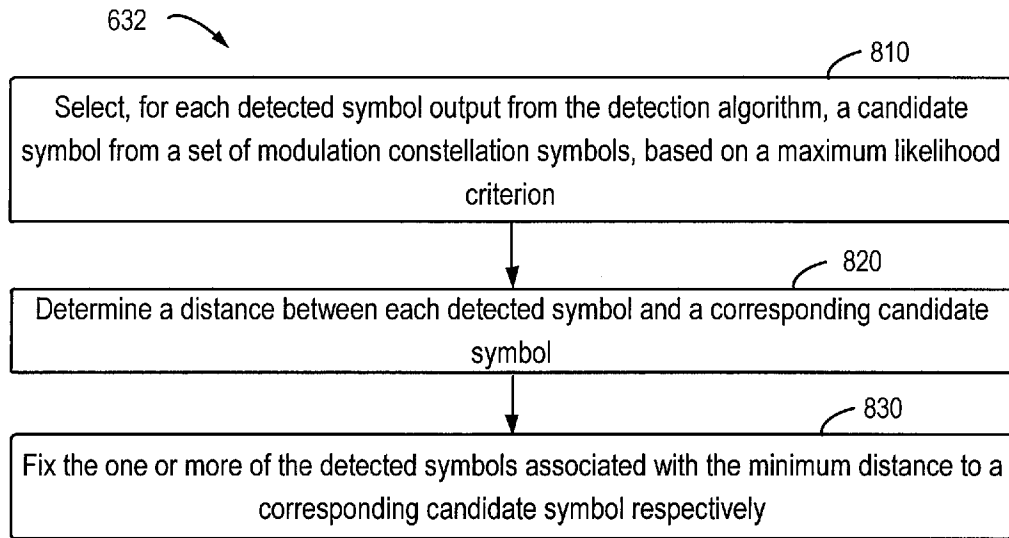


FIG. 8

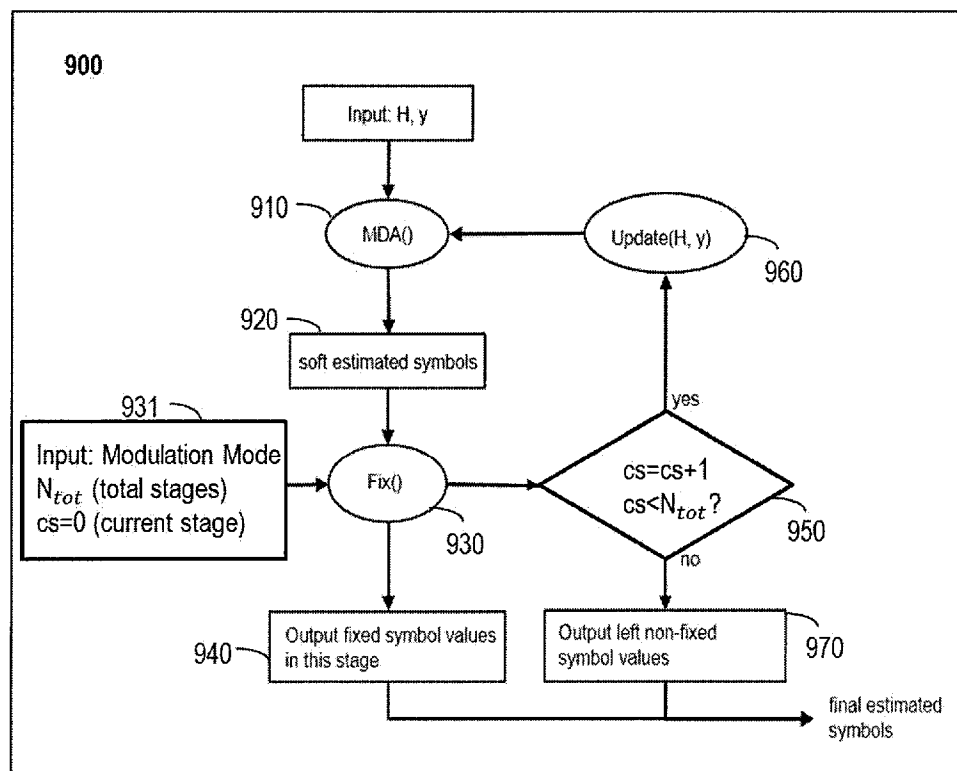


FIG. 9

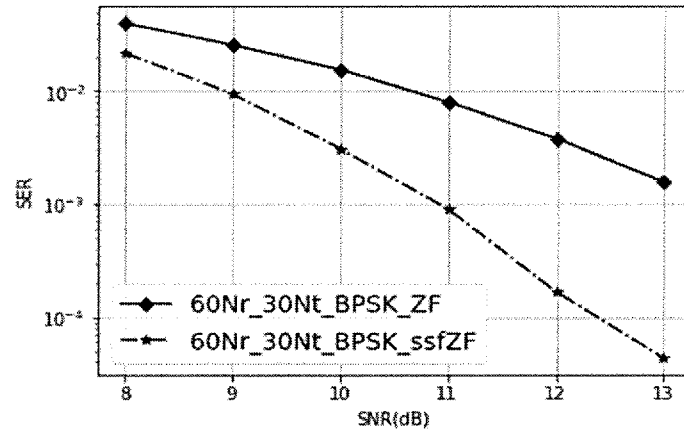


FIG. 10

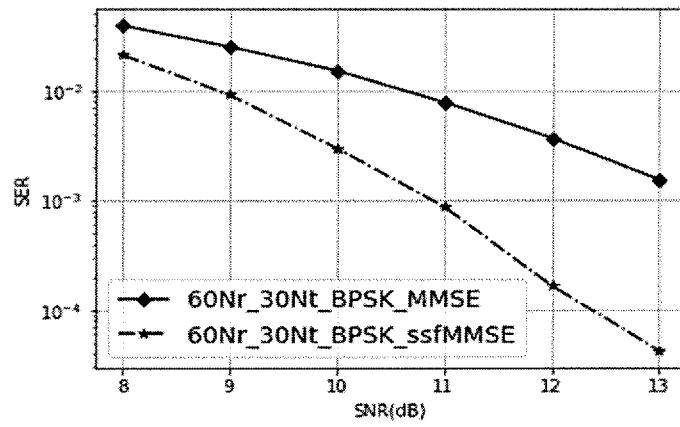


FIG. 11

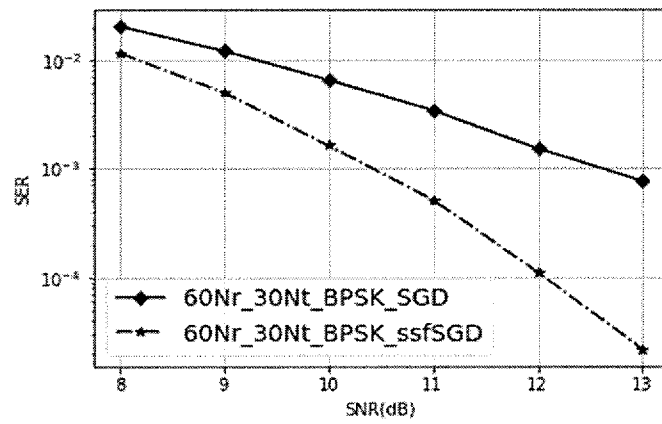


FIG. 12

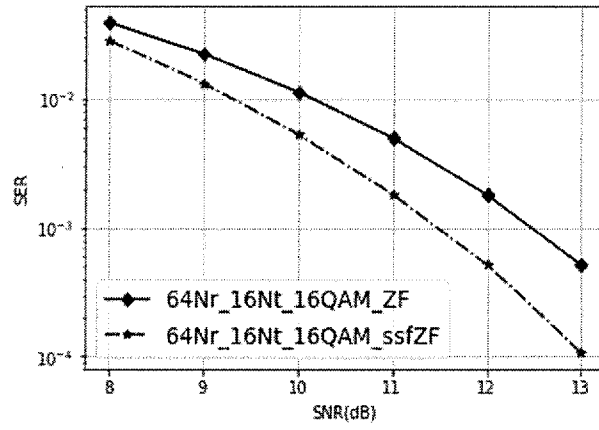


FIG. 13

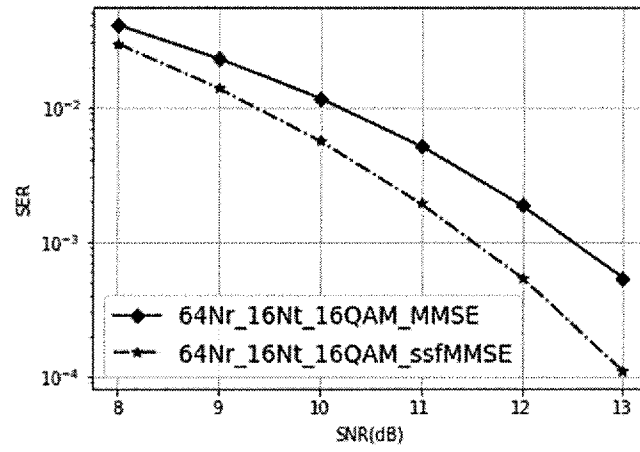


FIG. 14

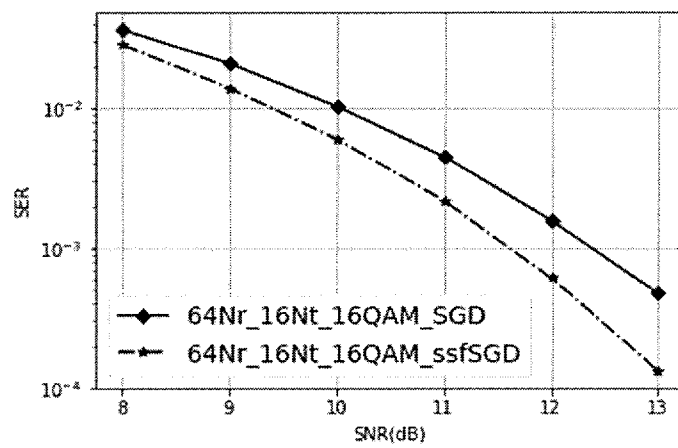


FIG. 15

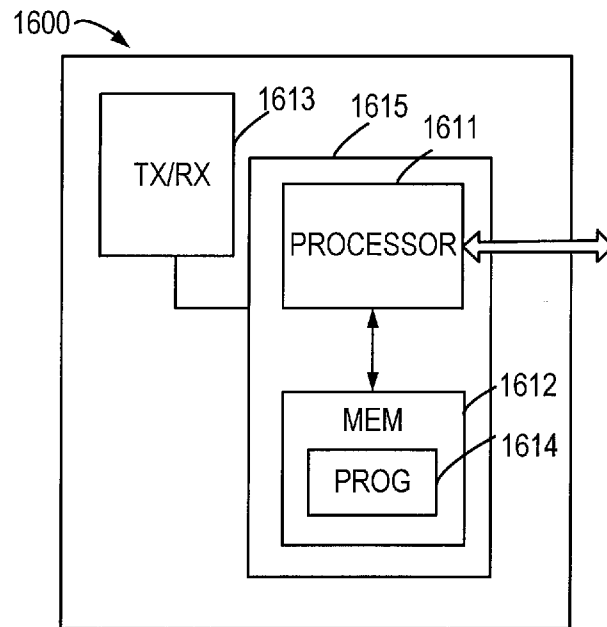


FIG. 16

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2018/082713

A. CLASSIFICATION OF SUBJECT MATTER

H04B 7/04(2017.01)i; H04L 25/02(2006.01)i; H04B 1/10(2006.01)i

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)

H04L; H04B

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT,EPODOC,WPI,CNKI,IEEE: signal+, data, symbol+, detect+, decision, MIMO, mMIMO, massive, large scale, stage, multi+, iterat+, update+, channel, remove, subtract, ZF, MMSE, GD

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	YANG, Lieliang. "Using Multi-Stage MMSE Detection to Approach Optimum Error Performance in Multiantenna MIMO Systems." <i>2009 IEEE 70th Vehicular Technology Conference Fall</i> , 12 January 2010 (2010-01-12), sections II and IV	1-28
A	US 2004242179 A1 (ONGGOSANUSI, EKO N. ET AL.) 02 December 2004 (2004-12-02) the whole document	1-28
A	US 2009080557 A1 (WILHELMSSON, LEIF ET AL.) 26 March 2009 (2009-03-26) the whole document	1-28
A	CN 102868434 A (BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS) 09 January 2013 (2013-01-09) the whole document	1-28
A	NGUYEN, Thanh Binh et al. "Efficiency Zero-Forcing Detectors based on Group Detection for Massive MIMO systems." <i>2017 International Conference on Advanced Technologies for Communications</i> , 07 December 2017 (2017-12-07), the whole document	1-28



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier 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

06 December 2018

Date of mailing of the international search report

29 December 2018

Name and mailing address of the ISA/CN

National Intellectual Property Administration, PRC
6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing
100088
China

Authorized officer

LIU,Shan

Facsimile No. (86-10)62019451

Telephone No. 86-10-53961626

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2018/082713

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2004242179	A1	02 December 2004	None			
US	2009080557	A1	26 March 2009	WO	2009037110	A1	26 March 2009
				EP	2191603	A1	02 June 2010
CN	102868434	A	09 January 2013	None			