

Iterative MIMO Detection for Large Wireless MIMO Systems

Alexandre P. J. Dixneuf
Student, CentraleSupélec

Antoine Berthet
Professor, CentraleSupélec

Abstract—This report explores iterative detection and decoding techniques for large-scale MIMO systems, focusing on the application of Belief Propagation (BP) and its Gaussian variant (GaBP) within factor graphs. Starting with a detailed derivation of message-passing equations, the study evaluates baseline performance under massive MIMO settings. A group-wise approximation strategy is then introduced to reduce computational complexity by modeling subsets of variables as Gaussian-distributed. The work further integrates Low-Density Parity-Check Codes (LDPC) and Bit-Interleaved Coded Modulation (BICM) to enhance error correction and spectral efficiency. Through simulations and performance analysis, the proposed implementations demonstrate potential improvements in complexity-performance tradeoffs, paving the way for scalable receiver architectures suitable for next-generation wireless networks such as 5G and beyond.

Index terms—Massive MIMO, Belief Propagation, Gaussian Approximation, Factor Graphs, LDPC Codes, BICM, Iterative Detection

I. INTRODUCTION

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

II. BACKGROUND

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa. [1]

III. THEORY AND REASONING

A. Theory

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

B. Reasoning

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio. 1.

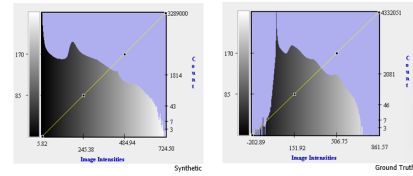


Fig. 1. A side-by-side comparison of the intensity histogram between the post-processing synthesized T2w compared to the ground truth.

IV. METHODS

To run the necessary simulations, several criteria had to be met. To begin, an overall network simulation needed to be created. This simulation must handle the creation of a random signal, the propagation of the signal through a channel, and then the easy manipulation and handling of the necessary data and problem parameters. Then, a basic decoder is necessary to allow for easy addition of decoders. Since the average user will

use completely different frameworks for any and all codes, the simulation uses a virtual class Decoder which handles most of the data. This allows new decoder classes to simply override the virtual decode function to properly run, and thus new decoders can be added in the future for proper comparison. Lastly, the individual decoders analyzed throughout this report were implemented.

A. General Structure

The simulation depends on the use of a storage class called ProblemParameters. The simulation of transmitted signals in a MIMO system depends on multiple variables, such as the number of transmitting antennas, receiving antennas, the degree of modulation, the number of time steps and/or the signal-to-noise ratio. The variables all have different constraints, including non-negativity and the modulation required to be a power of 4. To avoid constant checks to avoid errors and bugs, all of the data is stored in the storage class ProblemParameters. The class's task is to verify the parameters given are acceptable and to convert everything to constants such that they may no longer be altered. It also keeps a constant structure so that every future component knows the proper steps to handle the data. The next component is the class MySignal. This class is similar to the ProblemParameters in that it has the main purpose of being a robust storage class. Rather than trusting the user to know what format the signal should be, it forces a constant format, and verifies the accuracy and altering of the contained data. It also serves to generate the random signal, although it does mean it would require additional steps to run the simulation on given data rather than random data. While this will have no effect on the study's result, it does hurt any future application of this code. The next class is QAMConstellation. This class serves to independently handle the M-QAM modulation of signals, as well as receiving and drawing the proper gray code constellation. Lastly, there remains the Channel class. This class handles taking the data, generating the channel conditions, and propagating the given signal through the channel.

B. Virtual Decoder

One of the guiding principles behind the structure of the code in this project is to ensure that future decoders can be easily added and thus compared to the current methods. To this purpose, a virtual Decoder class was developed. This class handles taking in any data that a decoder could use, all the while keeping the rest of the data still accessible in case of non-standard decoders. Then, it handles the majority of the work necessary for a C++ class. It leaves only the function "decode" virtual such that overriding classes only need to override the single function to be fully compatible.

C. Belief Propagation Decoder

D. Gaussian Approximation Belief Propagation Decoder

E. Clustered (Group-Wise) Decoder

F. Low-Density Parity-Check Code Decoder

G. Bit Interleaved Constellation Modulation Decoder

V. EXPERIMENTS AND RESULTS

There are five independent variables being verified during this project: the number of transmitting antennas, the number of receiving antennas, the number of messages sent, the level of modulation (M-QAM), and the signal-to-noise ratio (SNR) of the transmitted signal. It would be impossible to test every combination of every variable to receive usable data, so a base framework was used. In "base" conditions, there are two transmitting and receiving antennas each, only two messages are sent in 16-QAM with an SNR of 20. Then, each variable independently will be selected, going from a given minimum and maximum value while the remaining variables are set to constant be the base conditions. To allow for direct comparison between each variable and the resulting accuracy of the decoders.

A. Transmitting Antennas

The transmitting antenna can be considered the most constraining variable, as most decoding methods were unable to handle anything past 5 transmitting antennas. Therefore, the range of tested values was from 2 antennas to 5 antennas, enough to get a general understanding of the relation between the number of antennas and the overall accuracy, without spending several days per simulation.

B. Receiving Antennas

The receiving antennas were much less constraining than the transmitting antennas, and thus a larger range of 2 to 10 antennas was able to be used.

C. Messages Sent

Compared to the previous sections, the number of messages sent has very little effect on the overall complexity of the decoders. Instead, it only lengthened the length of the simulations. A larger range of values was therefore possible, and thus the minimum of 2 and maximum of 20 was used.

D. M-Modulation

Testing values of M-QAM modulation was complicated, as only square constellations were possible to test, meaning values go from 4, to 16, then 64, 256, and then 1024. This means only 5 simulations change the complexity of the simulation by a factor of almost 1000. 1024 was too complicated and 4 gave no information as even a guess could have good decoding odds, so the values of 16-QAM, 64-QAM, and 256-QAM were used. While this gives minimal data, it should still give a rough idea of the relation.

E. Signal-to-Noise Ratio

The SNR has no effect on the complexity of the system, so a large range of 10 to 40 was able to be used.

VI. DISCUSSION

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

REFERENCES

- [1] Gruszecki M, Lancaster G, Stefanovska A, Neary JP, Dech RT, Guminski W, Frydrychowski AF, Kot J, Winklewski PJ. *Human Subarachnoid Space Width Oscillations in the Resting State*, 2018