ADMM-based Infinity-Norm Detection for Massive MIMO: Algorithm and VLSI Architecture

ECSE 6680 - Zhaolin Qiu

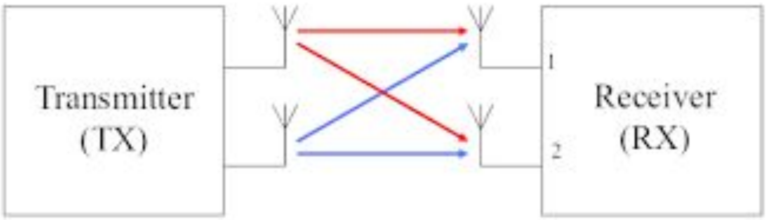
1. Problem
2. Traditional Algorithms
3. The ADMIN algorithm the paper proposes
   (1). ADMM and Infinity Norm
   (2). Why
   (3). How
4. VLSI structure
5. Performance

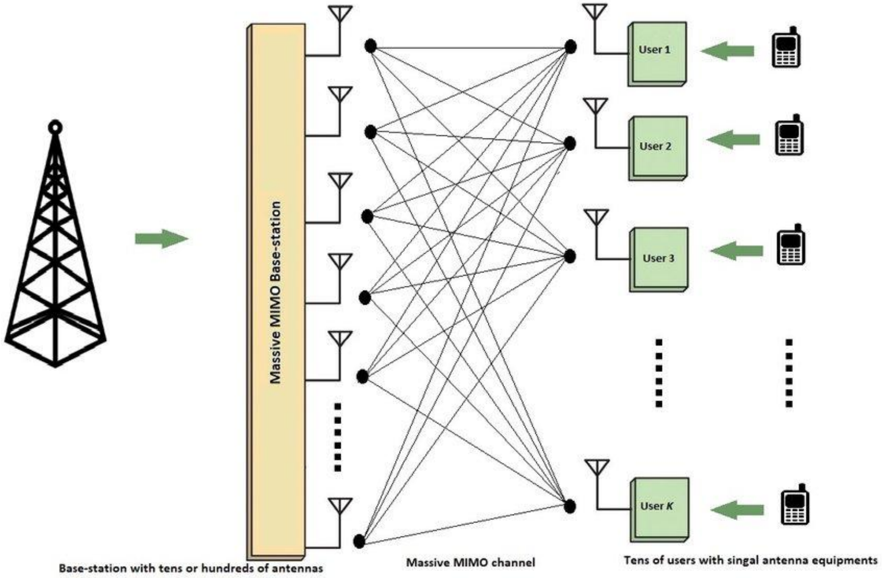
Massive MU-MIMO

MIMO: multiple antennas are used at both the transmitter and the receiver e.g. 2x2 MIMO

Multi-User(MU): The base station serves multiple users simultaneously

Massive MIMO: The base station has many antennas.





Model

$y \in \mathbb{C}^B$ **Y** (*Bx1* vector) represents the signals received by the base station's B antennas.

$x \in \mathbb{C}^U$ **x** (*Ux1* vector) represents symbols sent by all U users.

$H \in \mathbb{C}^{B \times U}$ **H** (*BxU* matrix) models the wireless channel

Each element h_b,u in **H** represents the channel gain

from user u to BS antenna b.

$n \in \mathbb{C}^B$: **n** is the noise vector applied to each antenna

B : The number of antennas in the BS

U: The number of users

Each BS antenna receives a superposition of signals from users

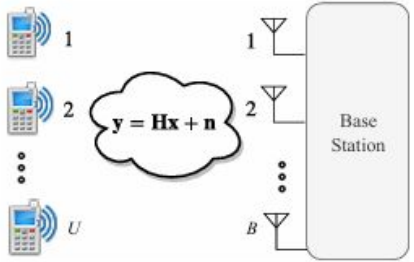


Fig. 1: A massive MU-MIMO system in which a large number of BS antennas are serving a large number of users. The channel between the BS and users is modeled as $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$.

Problem

A detection algorithm for the base station to recover the transmitted symbols **x** from the received signal **y**.

Goal: Separate and estimate the transmitted symbol from each user

$\hat{\mathbf{x}}$ is estimated transmit symbol vector (**x**)

O is the modulation alphabet(constellation set) (e.g., QPSK, 64-QAM).

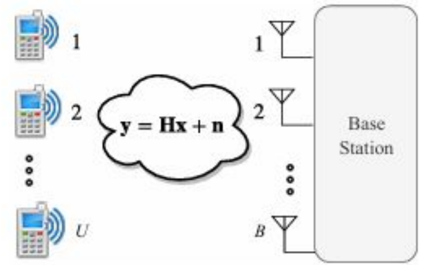


Fig. 1: A massive MU-MIMO system in which a large number of BS antennas are serving a large number of users. The channel between the BS and users is modeled as $\mathbf{y} = \mathbf{Hx} + \mathbf{n}$.

1. Maximum Likelihood (ML) Algorithm

O represent the constellation set of the transmitted signal x

||y - Hx||^2: Euclidean distance between the received signal and the possible transmitted signals

Evaluate every possible transmitted symbol in vector

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \arg\min_{\mathbf{x} \in \mathcal{O}^U} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2.$$

- Optimal (in terms of Error rate)

- O (***O*^U**) Exponential complexity: impractical for multi-user networks

(combinatorial problem: There are |***O***| possible values of x for each user)

2. Linear Methods

$$\hat{\mathbf{x}}_{\text{MMSE}} = \underset{\mathbf{x}\in\mathbb{C}^U}{\arg\min}\ \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + N_0 E_s^{-1}\|\mathbf{x}\|_2^2.$$

- Sub-Optimal
- Polynomial complexity: Efficient
- Traditional Method

ADMIN

ADMM-based **I**nfinity-**N**orm Detection

1. Infinity-Norm

2. ADMM-based

1. Infinity-Norm

Infinity-norm incorporates an additional constraint: $||x||_\infty \leq P$

||**x**||∞ is the largest absolute value of entries in **x** (infinity norm**)**

P is the power constraint ensuring realistic transmission levels. (like a bounding box)

(ignore possible x solutions that ||x|| > P as there are not realistic)

Purpose of Infinity-Norm

$$\hat{\mathbf{x}}_{\mathrm{BOX}} = \underset{\mathbf{x}\in\mathcal{C}_{\mathcal{O}}^{U}}{\arg\min} \, \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2.$$

Relax the original Maximum Likelihood problem to a **convex** polytope
(which needs a constraint on solutions)

Convert a **discrete**, combinatorial problem to a **continuous**, convex optimization problem

Brutal force -> Actively solve

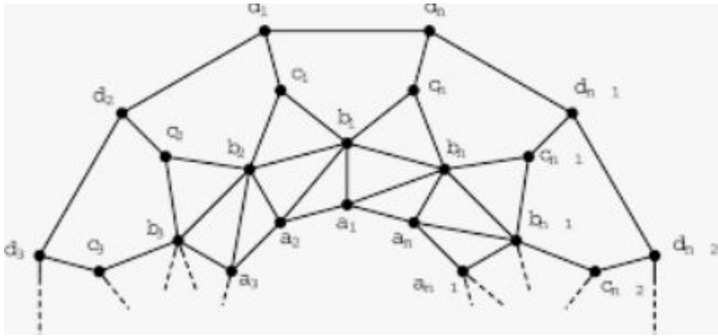
Lower performance than the original ML algorithm due to errors in relation and rounding

The constraint defines a box (hypercube), which is a convex polytope.

The original integer solutions become vertices of this polytope.

Convex optimization techniques can find an optimal solution within this relaxed space

Rounding methods can then retrieve discrete solutions.



Issues of Infinity-Norm

Such methods are not particularly hardware friendly

Special purpose solvers are required

2. ADMM - **A**lternating **D**irection **M**ethod of **M**ultipliers

A numerical method to solve convex optimization problems

(by breaking it into smaller sub-problems)

It is the algorithm that implements Infinity Norm Method on hardware for Massive MIMO Detection

LDL-decomposition is used to solve the linear system of equations

$$\hat{\mathbf{x}}_{\mathrm{BOX}} = \arg\min_{\mathbf{x}\in\mathcal{C}_{\mathcal{O}}^{U}} \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2. \quad \hat{\mathbf{x}}_{\mathrm{ML}} = \arg\min_{\mathbf{x}\in\mathcal{O}^{U}} \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2.$$

$$\underset{\mathbf{x},\mathbf{z}\in\mathbb{C}^{U}}{\text{minimize}}\ \frac{1}{2}\|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2 + g(\mathbf{z}) \quad \text{subject to } \mathbf{z}=\mathbf{x}$$

$$\Longrightarrow \hat{\mathbf{x}} = (\mathbf{H}^{H}\mathbf{H}+\beta\mathbf{I})^{-1}(\mathbf{H}^{H}\mathbf{y}+\beta(\mathbf{z}-\boldsymbol{\lambda})). \quad (9)$$

Algorithm 1 ADMIN

inputs: $\mathbf{y}$, $\mathbf{H}$, N_0 and E_s

1: ***preprocessing***

2: $\quad \beta = N_0 E_s^{-1} \epsilon$

3: $\quad \mathbf{G} = \mathbf{H}^H \mathbf{H} + \beta \mathbf{I}_U$

4: $\quad \mathbf{G} = \mathbf{L}\mathbf{D}\mathbf{L}^H$

5: $\quad \tilde{\mathbf{L}} = \mathbf{L}^{-1}$, $\tilde{\mathbf{D}} = \mathbf{D}^{-1}$

6: ***initialization***

7: $\quad \mathbf{z} = \mathbf{0}$

8: $\quad \boldsymbol{\lambda} = \mathbf{0}$

9: ***detection***

10: $\quad \mathbf{y}_{\mathrm{MF}} = \mathbf{H}^H \mathbf{y}$

11: $\quad$ **for** $i = 1, \ldots, K$

12: $\quad\quad \hat{\mathbf{x}} \leftarrow \tilde{\mathbf{L}}^H \tilde{\mathbf{D}} \tilde{\mathbf{L}} (\mathbf{y}_{\mathrm{MF}} + \beta(\mathbf{z} - \boldsymbol{\lambda}))$

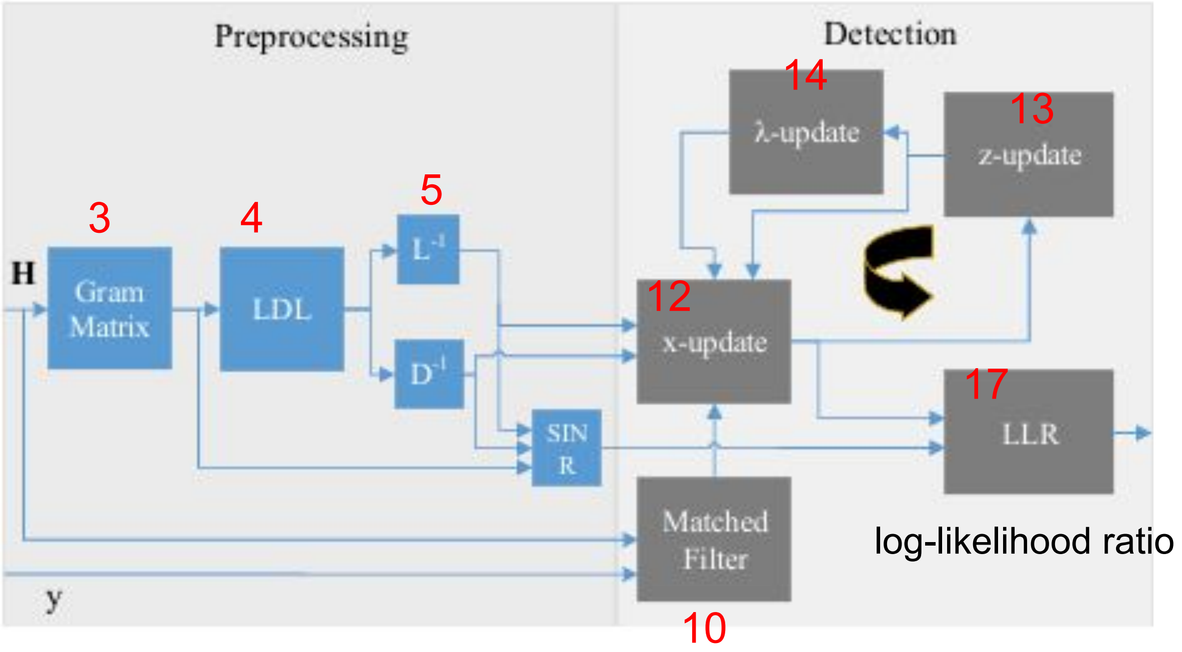
13: $\quad\quad \hat{\mathbf{z}} \leftarrow \mathrm{proj}_{\mathcal{C}_{\mathcal{O}}}(\hat{\mathbf{x}} + \boldsymbol{\lambda}, \alpha)$

14: $\quad\quad \boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \gamma(\hat{\mathbf{z}} - \hat{\mathbf{x}})$

15: $\quad\quad \mathbf{z} \leftarrow \hat{\mathbf{z}}$

16: $\quad$ **end**

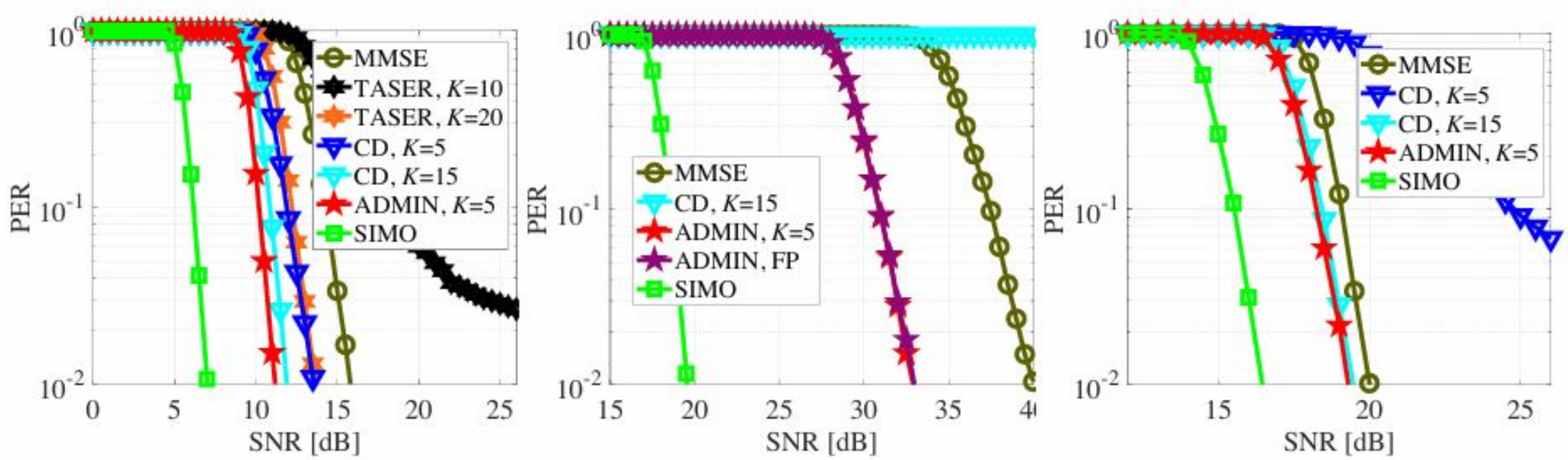
17: ***output***: $\hat{\mathbf{x}}$



Iterative Detection

Performance (coded) - Compared to other SOTA algorithms

SIMO Lower Bound: This is the performance as if no multi-user interference exist (only one active user) Theoretical Limit

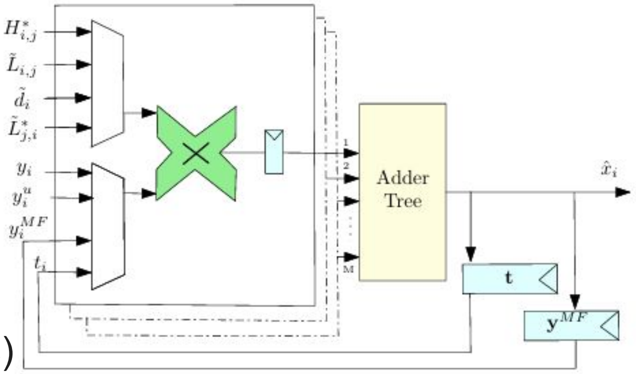


K is the number of iterations

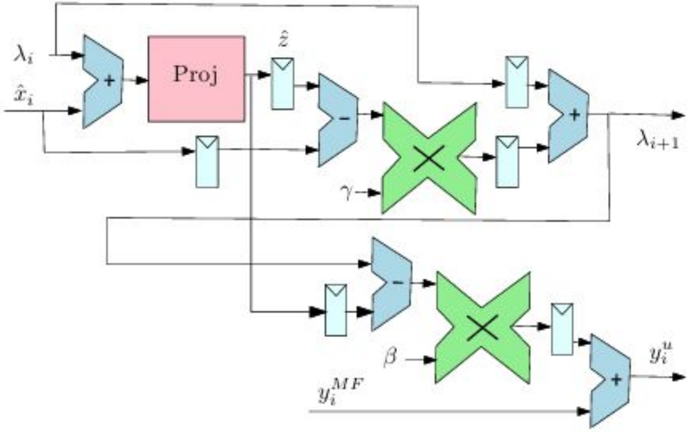
Techniques used in VLSI design

1. Fixed point arithmetic
2. Pipeline registers between each complex multiplier and adder
3. Matched filter update (MFU) (pipelined)

Vector Multiplier (M multipliers)



MFU



Latency and Throughput

K is # of iterations

M x U Network

		K=1	K=2	K=3	K=4	K=5
16×16	Cycles	70	109	148	187	226
	Throughput (Mbps)	979	629	463	366	303
	Latency (µs)	0.09	0.15	0.2	0.26	0.31
32×32	Cycles	134	205	276	347	418
	Throughput (Mbps)	895	585	434	345	287
	Latency (µs)	0.21	0.32	0.44	0.55	0.66
64×32	Cycles	198	269	340	411	482
	Throughput (Mbps)	588	432	342	283	241
	Latency (µs)	0.32	0.44	0.56	.67	.79