MIMO Receivers: from Algorithms to VLSI Architectures

Andreas Burg
Christian Senning, Nicholas Preyss, Christoph Studer

Telecommunications Circuits Laboratory
EPFL

July 5, 2012

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Telecommunications
Circuits Laboratory

New applications drive the development of wireless systems

digital video (HDTV)

laptops, computer networking

digital cameras

access points

base station

audio devices

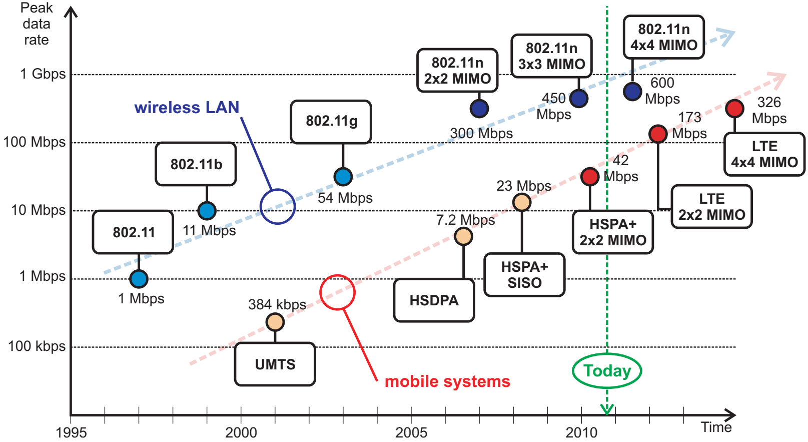
gaming consoles

mobile phones, PDAs

digital video recorders

- New applications require **higher throughput**
- Availability anywhere requires **improved coverage** and **range**

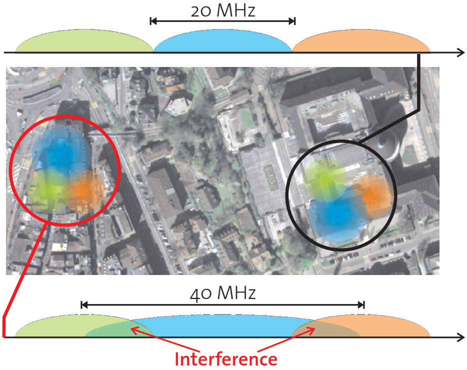
The rapid evolution of wireless communication



Edholm’s law [1]

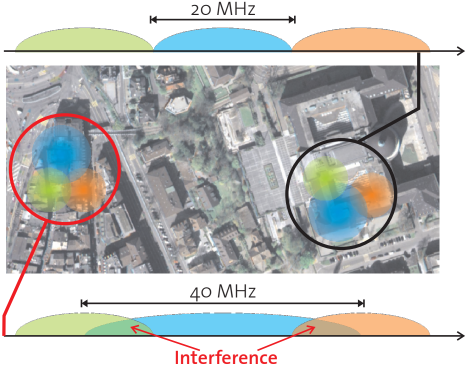
Wireless standards have followed the increase in data rate in wired networks at a pace that is close to doubling data rates every 18 months

Need for higher data rates cannot be met by simply increasing bandwidth or transmit power



- Licensed spectrum is **expensive**
- Massive **interference** in unlicensed spectrum
- Transmit power is **limited**

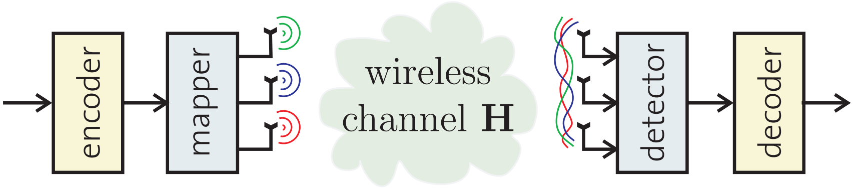
Need for higher data rates cannot be met by simply increasing bandwidth or transmit power



- Licensed spectrum is **expensive**
- Massive **interference** in unlicensed spectrum
- Transmit power is **limited**

Achieving higher throughput and supporting more users requires **better spectral efficiency**

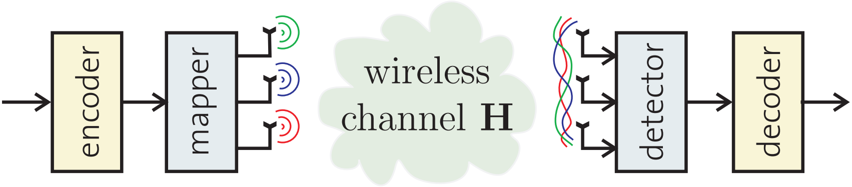
Solution: Wireless channels offer “spatial bandwidth”



Multi-antenna systems can leverage three types of gains [2]

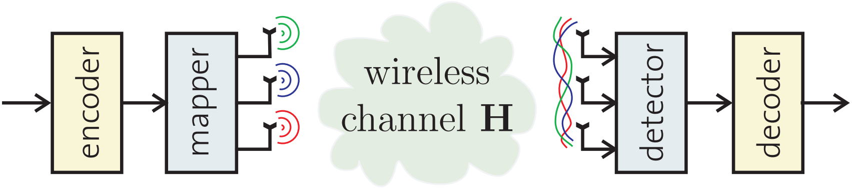
- **Diversity gain** mitigates the impact of fading, improving quality of service and reducing dead spots
- **Array gain** improves received signal strength, increasing range and coverage
- **Multiplexing gain** enables higher peak data rates and improves overall system capacity

Solution: Wireless channels offer “spatial bandwidth”



MIMO wireless technology combined coding improves **throughput, coverage, and range** at no expense in transmit power

Solution: Wireless channels offer “spatial bandwidth”

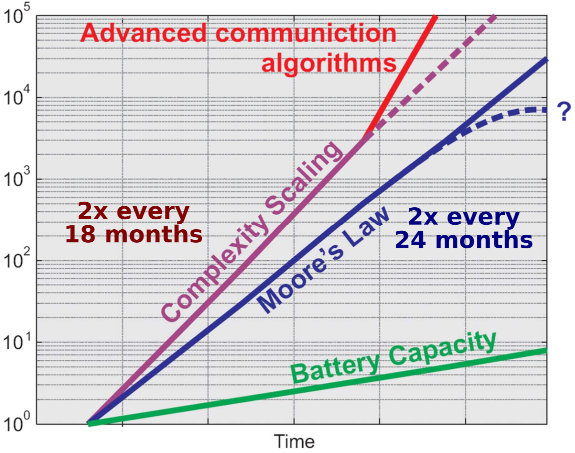


MIMO wireless technology combined coding improves **throughput, coverage, and range** at no expense in transmit power

Price to be paid: MIMO detection (i.e., separation of signal mixtures at receiver) incurs **significant computational burden**

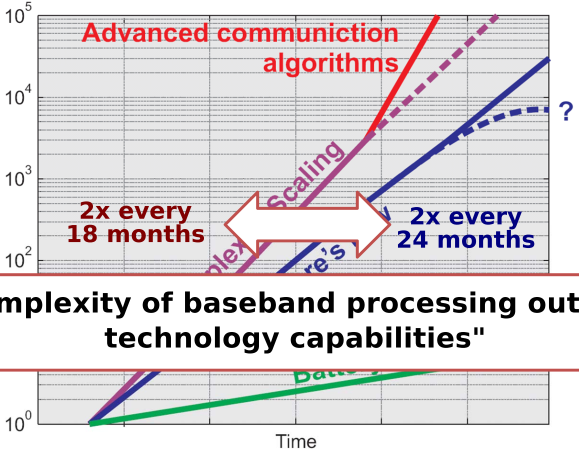
Imbalance between complexity and integration density

- Data rate doubles every 18 months
- Algorithm complexity grows (spectral efficiency)



Imbalance between complexity and integration density

- Data rate doubles every 18 months
- Algorithm complexity grows (spectral efficiency)

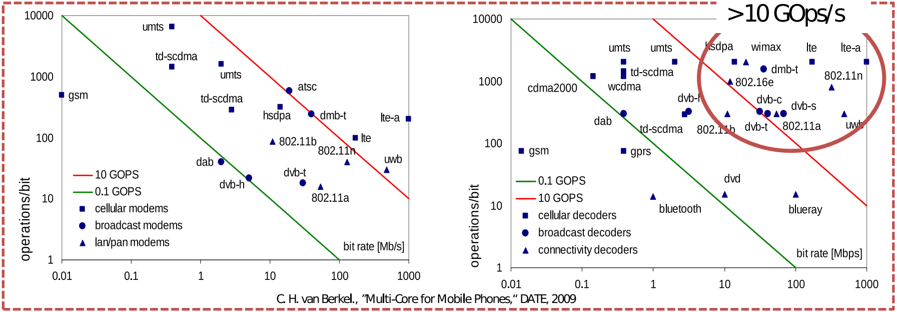


"Complexity of baseband processing outruns technology capabilities"

Evolution of complexity

Very rough complexity metric: GOps/s

- Definition of operation is very loose (add, mult, 4b, 8b, 16b,...)



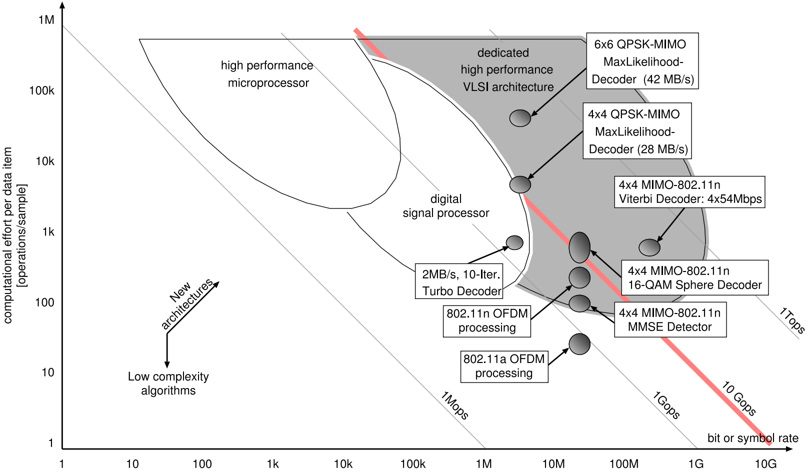
>10 GOps/s

C. H. van Berkel., "Multi-Core for Mobile Phones," DATE, 2009

802.11a	54Mbps	~1 GOps/s
UMTS/HSDPA	7.2Mbps	~5 GOps/s
LTE/LTE-A	>100Mbps	>10 GOps/s

Modern modems are in the **10 GOps/s** regime

Next generation wireless communication systems still require dedicated HW for critical blocks

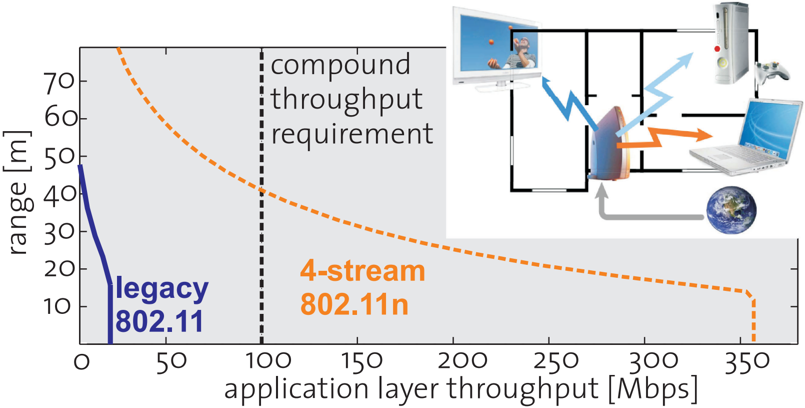


Outline

1. Introduction
2. MIMO-OFDM System Design Example: 802.11n
   - IEEE 802.11n System and Receiver Architecture
   - Linear MIMO-OFDM Receiver for IEEE 802.11n
3. Advanced MIMO Receivers
   - Lattice Reduction Aided Linear Detectors
   - Maximum Likelihood (ML) Detection with Sphere Decoding
   - Low Complexity Soft-Output Sphere Decoding
   - System Integration of Sphere Decoding
4. The Last Frontier: Iterative MIMO Detection and Decoding (IDD)
   - Soft-input Soft-output MIMO Detection
   - System Integration: Layered Detection and Decoding
5. MIMO with Transmit RF Impairments
   - Impact on Channel Capacity
   - Impact on MIMO Detection
   - MIMO Detection with Tx-RF Impairments
6. References

Case Study: Wireless LAN (IEEE 802.11)

- Home networking requires high throughput across large range
- Legacy WLAN standards can not meet these requirements



IEEE 802.11n is an enabling Technology for the Digital Home

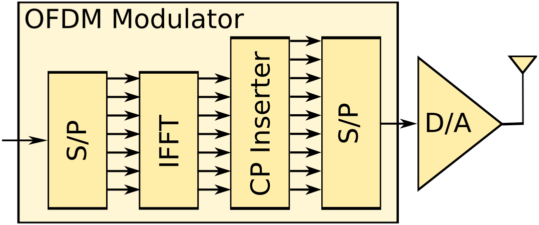
MIMO-OFDM System Design Example: 802.11n

IEEE 802.11n System and Receiver Architecture

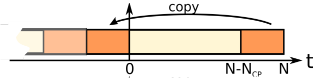
IEEE 802.11n is Based on IEEE 802.11g

OFDM

Signal is modulated on orthogonal sub-carriers to avoid inter symbol interfearence (ISI)

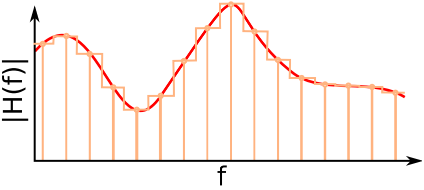


Cyclic prefix turn linear convolution into cyclic convolution



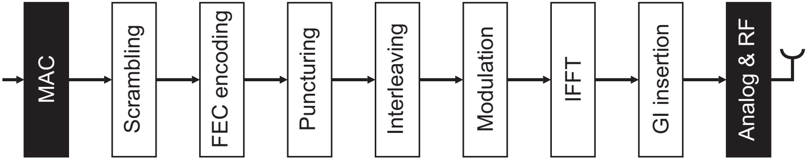
OFDM decomposes **wide-band frequency selective channel** into **narrow-band flat channels**

- No ISI
- Each tone can be treated **independently** (expect of coding)

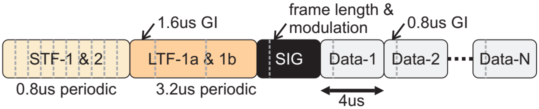


Packet Based TX-Scheme

- 20 MHz bandwidth with 64 tones (48 data + 4 pilots)
- Modulation: BPSK, QPSK, 16-QAM, 64-QAM
- Convolutional code: 1/2; with puncturing: 2/3, 3/4

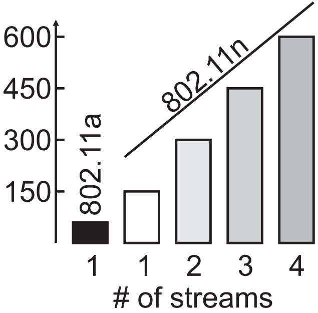


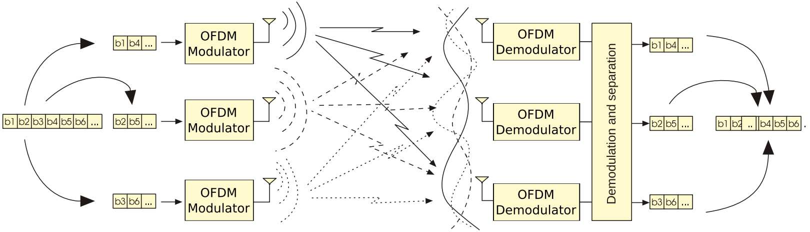
OFDM-mode frame format



IEEE 802.11n Key Extensions

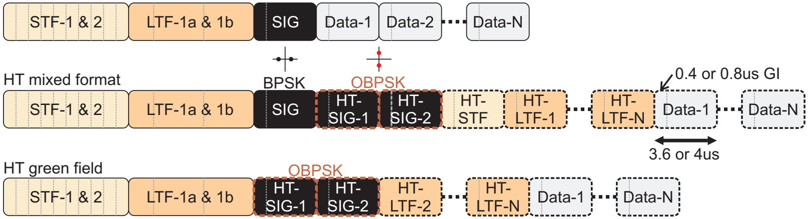
- Wider bandwidth: 20 MHz and **40 MHz**
- **Spatial multiplexing**:
  - 1,2 spatial streams mandatory
  - 3,4 spatial streams optional
- Optional support for **space-time block codes**
- Optional support for **beamforming**
- Additional code rate 5/6





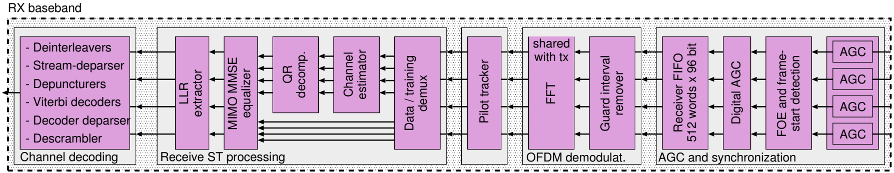
IEEE 802.11n Extensions

Two new frame formats to support MIMO with backward compatibility



- Initial training (STF and LTF) is identical with 11g frames
- Mixed-mode frame is compatible with 11g up to the SIG field
- Offset-BPSK (OBSK) modulation indicates HT-frame
- Additional modes require additional signal fields (HT-SIGx)
- MIMO modes require additional training (HT-LTF)

Receiver Architecture

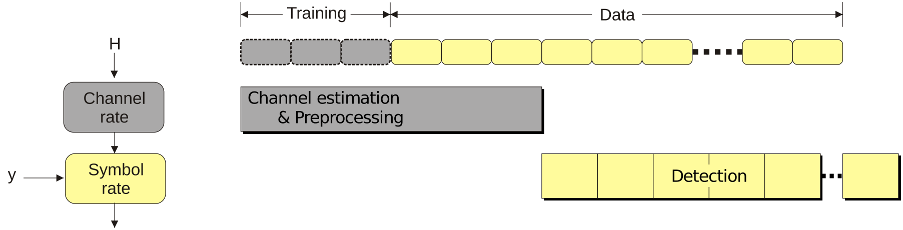


- AGC&Synchronisation
  Adjustment of the power level and frame synchronization.
- OFDM demodulation
  Removal of the cyclic prefix and conversion to frequency domain.
- ST Processing
  Separation of space-time streams.
- Channel decoding
  Error correction.

MIMO detection is a two-step process

We distinguish between **channel-rate** and **symbol-rate** processing:

- **Channel-rate preprocessing** comprises all those operations that are carried out only when the channel (estimate) changes
- **Symbol-rate detection** comprises all those operations that need to be carried out for each received vector
  - Detection can only start after preprocessing is complete

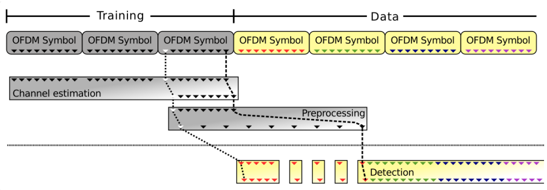


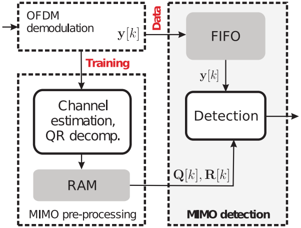
Computational effort for preprocessing introduces a latency

MIMO Detection in wideband systems

OFDM: preprocessing and detection are performed on a tone-by-tone basis

- OFDM demodulation (FFT) delivers received vectors tone-by-tone to the preprocessing and detection units





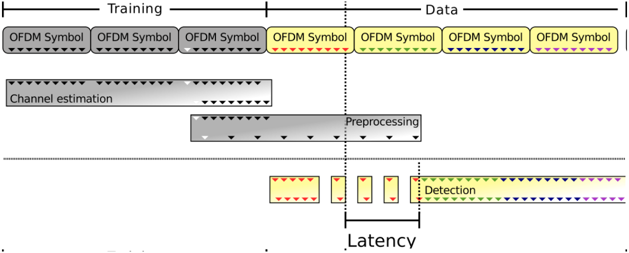
During detection the MIMO detector must keep up with the arrival rate of the vector symbols (tones) from the FFT

- Preprocessing results are stored to be used in the second step
- A FIFO stores received vectors until preprocessing is ready

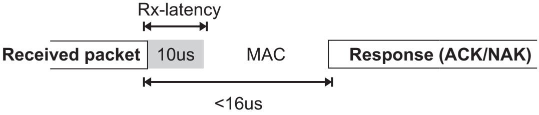
Preprocessing latency increases the size of the FIFO

Systems impose performance constraints on both preprocessing and detection

Slow preprocessing introduces additional overall processing latency.



CSMA in IEEE 802.11n requires a receiver latency below 10 μs



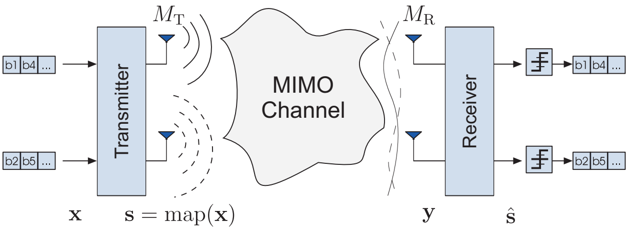
Latency is not tolerable. Fast preprocessing is required.

MIMO-OFDM System Design Example: 802.11n

Linear MIMO-OFDM Receiver for IEEE 802.11n

Uncoded MIMO system model

MIMO-OFDM: Consider a narrow band model for each tone



$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}$$

$\mathbf{x}$ ... binary-valued data vector

$\mathbf{s}$ ... transmitted symbol vector, $\mathbf{s} \in \mathcal{O}^{M_T}$

$\mathcal{O}$ ... set of constellation points (e.g., 2^Q-QAM)

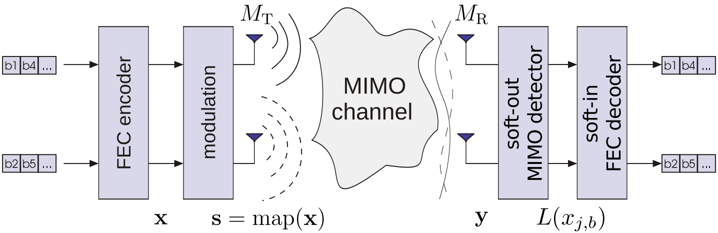
$\mathbf{H}$ ... $M_R \times M_T$ MIMO channel matrix, $M_R \geq M_T$

$\mathbf{n}$ ... additive Gaussian noise vector, $\mathbb{E}\left[\mathbf{n}\mathbf{n}^H\right] = N_o\mathbf{I}_{M_R}$

$\mathbf{y}$ ... received vector

Hard-decision MIMO detector delivers binary estimates of the transmitted bits

Soft-output MIMO detection obtains and forwards LLRs to the FEC-decoder



Log-likelihood ratios (LLRs)

$$L\left(x_{j,b}\right) = \log\left(\frac{P(x_{jb}=0|\mathbf{y})}{P(x_{jb}=1|\mathbf{y})}\right) = \log\left(\frac{P(\mathbf{y}|x_{jb}=0)}{P(\mathbf{y}|x_{jb}=1)}\right)$$

Hard decisions from LLRs

$$\hat{x}_{jb} = \mathrm{sign}\left(L\left(x_{j,b}\right)\right)$$

- $\left|L\left(x_{j,b}\right)\right|$ large: more reliable
- $\left|L\left(x_{j,b}\right)\right|$ small: less reliable

Straightforward soft-output MIMO detection requires an exhaustive search

Definition

$\mathcal{X}_{j,b}^{(0)}$ and $\mathcal{X}_{j,b}^{(1)}$ ... sets of vector symbols for which $x_{j,b} = 0, 1$

Optimum LLR metric

$$L\left(x_{j,b}\right) = \log \frac{\sum_{\mathbf{s}\in\mathcal{X}_{j,b}^{(0)}} e^{-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2}{2\sigma^2}}}{\sum_{\mathbf{s}\in\mathcal{X}_{j,b}^{(1)}} e^{-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2}{2\sigma^2}}}$$

Computational complexity

- Computation of optimum LLRs requires evaluation of $\|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2$ for all $|\mathcal{O}^{M_T}|$ candidate vector symbols
- The LLR expression itself has also a considerable complexity

The Log-sum approximation simplifies the LLR computation but complexity remains prohibitive

Application of the Log-sum approximation $\log_2 \sum_i e^{-|d_i|^2} \approx -\min |d_i|^2$ to $L(x_{j,b})$ yields

Max-log approximation for LLRs

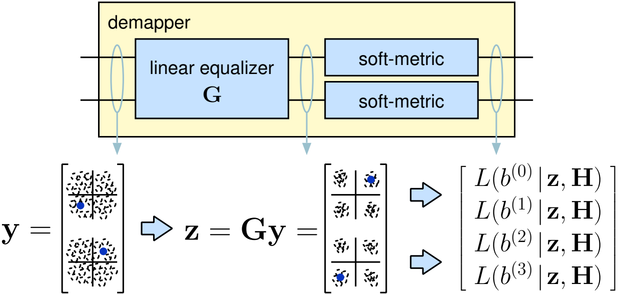
$$L(x_{j,b}) = \min_{\mathbf{s} \in \mathcal{X}_{j,b}^{(0)}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 - \min_{\mathbf{s} \in \mathcal{X}_{j,b}^{(1)}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2$$

- The max-log approximation simplifies the computation of the LLRs
- Unfortunately, even the max-log approximation requires searching two sets, each with cardinality $\frac{1}{2}|\mathcal{O}^{M_T}|$

Example

For 64-QAM and $M_T = 4$, computing max-log LLRs for 24 bits with an exhaustive search requires evaluation of 16.8 million Euclidean distances

Linear equalization decomposes the MIMO channel into M_T SISO channels


demapper
linear equalizer $\mathbf{G}$
soft-metric
soft-metric
$\mathbf{y} =$
$\mathbf{z} = \mathbf{G}\mathbf{y} =$
$L(b^{(0)} \,|\, \mathbf{z}, \mathbf{H})$
$L(b^{(1)} \,|\, \mathbf{z}, \mathbf{H})$
$L(b^{(2)} \,|\, \mathbf{z}, \mathbf{H})$
$L(b^{(3)} \,|\, \mathbf{z}, \mathbf{H})$

Soft-output linear detection algorithm [3, 4]

- A linear equalizer $\mathbf{G}$ can be used to spatially separate the transmitted signals from the received vector $\mathbf{y}$
- Based on the equalizer output $\mathbf{z} = \mathbf{G}\mathbf{y}$, the bit-metrics are computed on each spatial stream independently

Linear detection [2] requires solving a system of equations

Linear detection algorithm

- Estimate the received vector by solving $\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}$ for $\mathbf{s}$ through linear estimation techniques
- A straightforward approach is to obtain a **linear estimator** $\mathbf{G}$ which is then applied to the received vector $\mathbf{y}$
- Hard decisions can be obtained by quantizing to the nearest scalar constellation point

$$\hat{\mathbf{s}} = Q(\mathbf{G}\mathbf{y})$$

Zero-forcing detection

- Fully remove all interference from other streams
- Noise enhancement

$$\mathbf{G}_{\mathrm{ZF}} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H$$

Biased MMSE detection

- Take the noise into account
- Minimize distortion from noise and interference

$$\mathbf{G}_{\mathrm{MMSE}} = (\mathbf{H}^H\mathbf{H} + M_T\sigma^2\mathbf{I})^{-1}\mathbf{H}^H$$

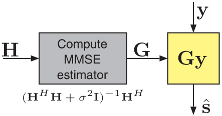
Linear receivers can be implemented based on matrix inversion or QR decomposition [5]

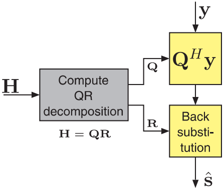
Direct matrix inversion (DMI):

- Preprocessing unit computes MMSE or ZF estimator
- Detection is based on matrix-vector multiplication

QR decomposition (QRD):

- Preprocessing computes QRD of $\mathbf{H} = \mathbf{QR}$
- Detection solves $\mathbf{Q}^H\mathbf{y} = \mathbf{Rs}$ through back substitution





Advantages of the QR-decomposition approach

- Good numerical properties (unitary transformations) and high regularity
- Compatible also with many other MIMO detection algorithms

QR based soft-output detection

1. **Spatial separation**
   Solve $\mathbf{Q}^H\mathbf{y} = \mathbf{R}\mathbf{z}$ with back substitution
2. **Max-log LLRs computation on scalar constellations $\mathcal{X}$ only**

$$L\left(x_{j,b}\right) = \rho_j \underbrace{\left(\min_{\mathbf{s}\in\mathcal{X}_b^{(0)}} \|z_j - s\|^2 - \min_{\mathbf{s}\in\mathcal{X}_b^{(1)}} \|z_j - s\|^2\right)}_{\Lambda_b(z_j)}$$

$\mathcal{X}_b^{(0)}, \mathcal{X}_b^{(1)}$	...	Scalar constellation points for which the bth bit is 0 or 1
ρ_j	...	Per-stream post-equalization SINR

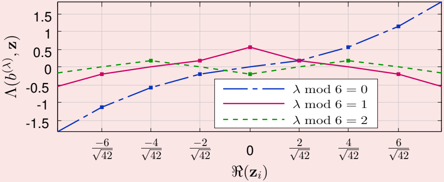
3. **Per-stream SINR computation**

$$\rho_i = \frac{1}{M_{\mathrm{T}}\sigma_n^2\left[\left(\mathbf{H}^H\mathbf{H} + M_{\mathrm{T}}\sigma_n^2\mathbf{I}\right)^{-1}\right]_{i,i}} - 1.$$

Complexity reduced QR soft-output

Max-log LLRs computation $L\left(x_{j,b}\right) = \rho_j \Lambda_b(z_j)$

- With the appropriate Gray-labeled QAM mapping, $\Lambda_b(z_j)$ reduces to a set of piecewise linear functions



ρ_i computation

$$\rho_i = \frac{1}{M_{\mathrm{T}}\sigma_n^2\left[(\mathbf{H}^H\mathbf{H} + M_{\mathrm{T}}\sigma_n^2\mathbf{I})^{-1}\right]_{i,i}} - 1 = \frac{1}{|\sqrt{M_{\mathrm{T}}\sigma_n^2}\mathbf{R}_{i,:}^{-1}|^2}$$

- **Note:** $\sqrt{M_T}\sigma\mathbf{R}^{-1} = \mathbf{Q}_c$

Two main strategies exist for QRD preprocessing

- QR-decomposition of an augmented channel matrix $\bar{\mathbf{H}} = \left[\mathbf{H}^T \; \sqrt{M_T}\sigma\mathbf{I}\right]^T$

$$\begin{bmatrix} \mathbf{H} \\ \sqrt{M_T}\sigma\mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_a, \mathbf{Q}_b \\ \mathbf{Q}_c, \mathbf{Q}_d \end{bmatrix} \mathbf{R} = \mathbf{QR}$$

$\mathbf{Q}$: Unitary $\quad$ $\mathbf{R}$: Upper triangular

$\mathbf{Q}_a, \mathbf{R}$	...	MIMO detection
$\mathbf{Q}_c$	...	$\sqrt{M_T}\sigma\mathbf{R}^{-1}$
$\mathbf{Q}_b, \mathbf{Q}_d$	...	no special properties

Givens rotation [6]

- Good numerical properties
- Implementation based on CORDICs
- Economy version provides $\mathbf{Q}_a, \mathbf{Q}_b$

Gram-Schmidt [SSB10]

- Numerical more challenging
- Implementation based on conventional arithmetic
- Economy version provides $\mathbf{Q}_a, \mathbf{Q}_c = \sqrt{M_T}\sigma\mathbf{R}^{-1}$, and r_{ii}^{-1}

Low complexity: Gram-Schmidt QRD provides $\mathbf{Q}_c$ as side-product

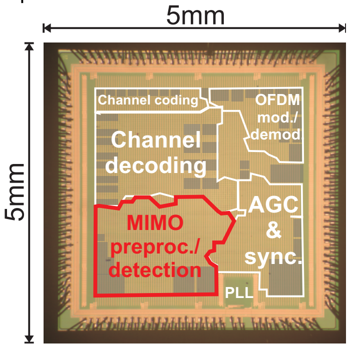
[SSB10] C. Senning, A. Staudacher, and A.Burg, "Systolic-array based regularized QR-decomposition for IEEE 802.11n compliant soft-MMSE detection," in *Proc. of Microelectronics (ICM), 2010 International Conference on*, Dec. 2009, pp. 391–394

Complexity of a linear detection is negligible

ASIC was manufactured in a 0.13 μm 1P8M process from UMC

ASIC key characteristics

- Die area: limited by 270 pads (256 signal pads / 14 power pads)
- Core area: 14.4 mm^2
- Logic complexity: 1.1M GEs
- Memory storage: 591 kBit in 49 single-port SRAM macros
- Reference clock: 80 MHz
- PLL derives 160 MHz and 320 MHz internal clocks



- MIMO preprocessing and detection consume together 25% of the active die area (250k GEs)
- Preprocessing and memories account for 80% of this area, while the linear soft-output detector requires only 50k GEs

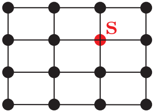
[AHB+09] A. Burg, S. Haene, M. Borgmann, D. Baum, T. Thaler, F. Carbognani, S. Zwicky, L. Barbero, C. Senning, P. Greisen, T. Peter, C. Foelmli, U. Schuster, P. Tejera, A. Staudacher, “A 4-Stream 802.11n Baseband Transceiver in 0.13um CMOS,” *Proc. of the Symposium on VLSI Circuits*, Jun. 2009.

Advanced MIMO Receivers

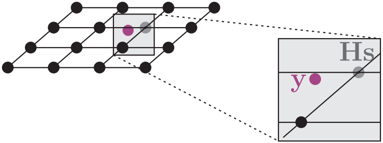
Lattice Reduction Aided Linear Detectors

MIMO detection

System with $M_{\mathrm{T}} = 2$ and real-valued 4-PAM constellations



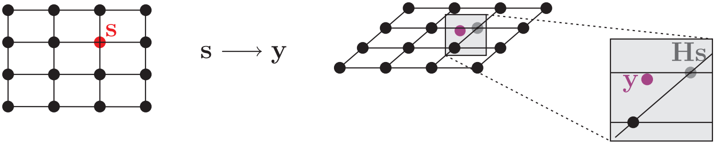
$$\mathbf{s} \longrightarrow \mathbf{y}$$



Task of the MIMO detector: recover $\mathbf{s}$ from $\mathbf{y}$ (assuming $\mathbf{H}$ is known)

MIMO detection

System with $M_{\mathrm{T}} = 2$ and real-valued 4-PAM constellations



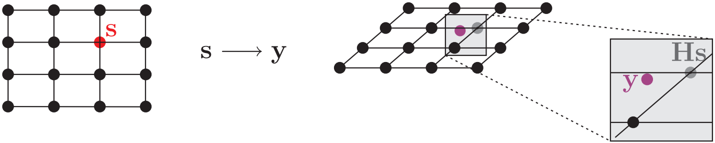
Task of the MIMO detector: recover $\mathbf{s}$ from $\mathbf{y}$ (assuming $\mathbf{H}$ is known)

- **Linear (ZF or MMSE) detection:** Solve the linear system for $\mathbf{y}$

  $\hat{\mathbf{s}} = Q(\mathbf{G}\mathbf{y})$ with $\mathbf{G}$ being a linear filter matrix, e.g., $\mathbf{G} = \mathbf{H}^{-1}$

MIMO detection

System with $M_\mathrm{T} = 2$ and real-valued 4-PAM constellations



Task of the MIMO detector: recover $\mathbf{s}$ from $\mathbf{y}$ (assuming $\mathbf{H}$ is known)

- **Linear (ZF or MMSE) detection:** Solve the linear system for $\mathbf{y}$

  $\hat{\mathbf{s}} = Q(\mathbf{G}\mathbf{y})$ with $\mathbf{G}$ being a linear filter matrix, e.g., $\mathbf{G} = \mathbf{H}^{-1}$

- **Maximum likelihood (ML) detection:**

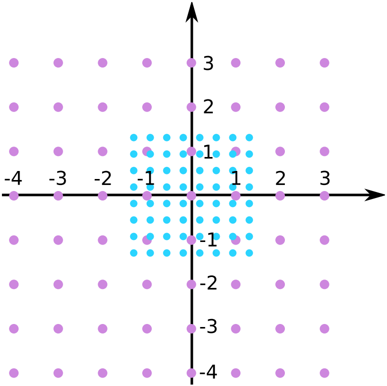
$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}\in\mathcal{O}^{M_T}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2$$

Constellation Point Remapping

- Constellation points are in most standards, not on an integer lattice
- Normally constellation points are normalized to power 1 at the receiver
- **The detection problem remains the same if the constellation points are trans-located.**
- Lattice point trans-location can be achieved by computing:

$$\tilde{\mathbf{y}} = \frac{1}{a}(\mathbf{y} - \mathbf{H}\mathbf{d})$$

where a is a modulation dependent scale factor and d is a constant vector with all elements equal $\frac{1}{2}$ for all QAM modulations.



Lattice Theory

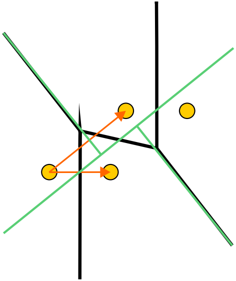
After trans-location on to an integer lattice, **lattice theory** can be applied

Lattice reduction: Perform MIMO detection on a “more orthogonal” basis

MIMO detection is mapping received vector to the closest lattice point

Linear detection (LD) and successive interference cancellation (SIC) struggle with **ill-conditioned channel matrices:**

- Lack of orthogonality between the columns of $\mathbf{H}$
- Decision regions deviate significantly from Voronoi regions



Lattice reduction (LR)-aided MIMO detection:

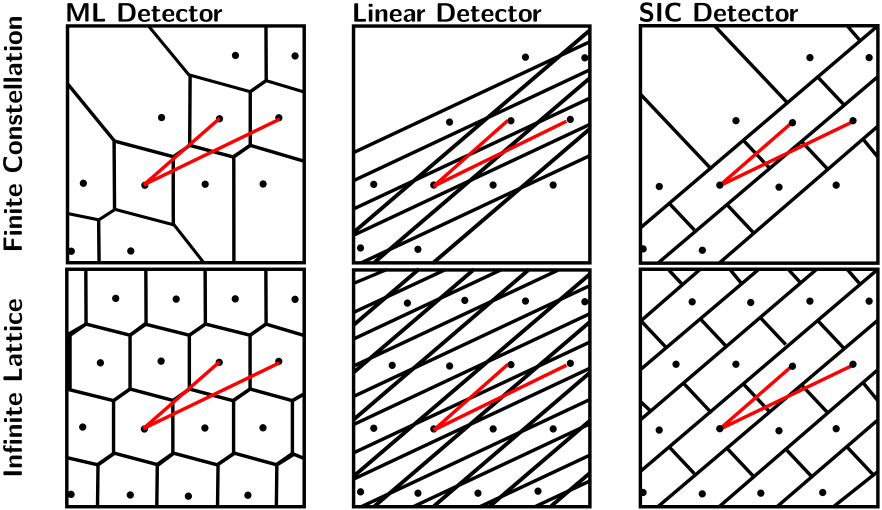
- Relax the (finite) constellation to an (infinite) lattice with basis $\mathbf{H}$

$$\left\{\mathbf{Hx} \mid \mathbf{x} \in \mathcal{X}^{M_T}\right\} \quad \rightarrow \quad \underline{\mathcal{L}}(\mathbf{H}) \triangleq \left\{\mathbf{H}\underline{\mathbf{x}} \mid \underline{\mathbf{x}} \in (\mathbb{CZ})^{M_T}\right\}$$

- For a **lattice** we can often **find a “more orthogonal” basis**

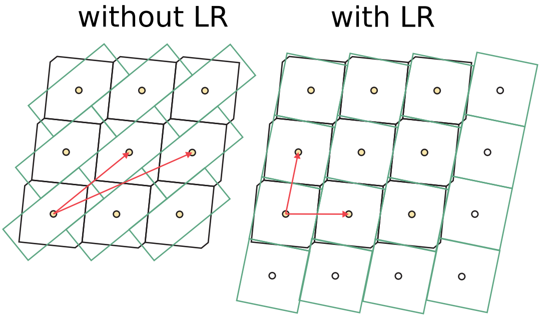
$$\mathbf{B} = \mathbf{HT} \quad \text{with} \quad |\det(\mathbf{T})| = 1\,(\text{unimodular})$$

Decision Regions of Different Detectors



Decision regions of LD and SIC are far from the Voronoi regions

Lattice reduction: Perform MIMO detection on a "more orthogonal" basis (cont'd)



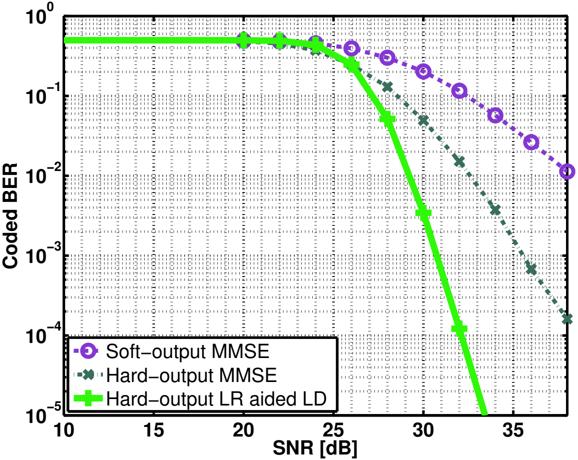
- Perform LD or SIC on the "more orthogonal" input-output relation

$$\mathbf{y} = \mathbf{B}\mathbf{z} + \mathbf{n} \quad \text{with} \quad \mathbf{z} \in (\mathbb{CZ})^{M_T}$$

- Perform slicing/ quantization within the "more orthogonal" basis
- Recover $\hat{\mathbf{s}} = \mathbf{T}^{-1}\hat{\mathbf{z}}$ followed by remapping to (finite) constellation $\mathcal{O}^{M_T}$

LR recovers decision regions that are close to the Voronoi region

LR preprocessing boosts performance of LD/SIC



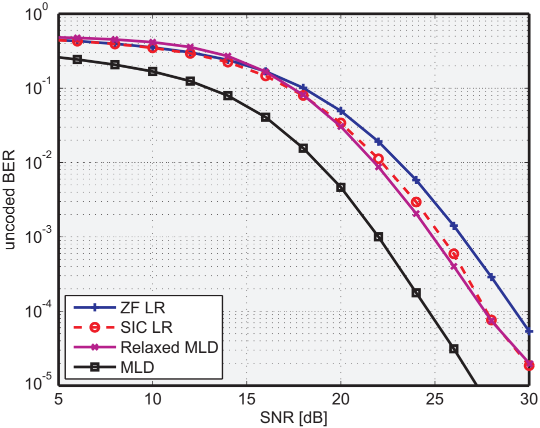
- Lattice reduction enables **full diversity** for LD and SIC

Remapping to infinite lattice points entails a BER performance loss

BER performance bound

LR-aided LD/SIC can not be better then a relaxed ML detector on an infinite lattice.

Combining SIC with LR improves the BER performance slightly more then combining LD with LR



Conclusion

Relaxing the (finite) constellation to an (infinite) lattice entails a BER performance loss

Many LR algorithms available

Brun's algorithm [7]

+ Low computational complexity
– Significant BER performance loss

Lenstra-Lenstra-Lovasz (LLL) algorithm [8, 9]

+ Near optimal performance with SIC detector
+ Based on QR decomposition
+ Many implementation available

Siegel LLL (S-LLL) algorithm [10]

+ Based on LLL with similar performance
+ Lower computational complexity and storage requirements

Gaussian reduction algorithm [11]

+ Optimal reduction
– Only possible for 2x2 MIMO

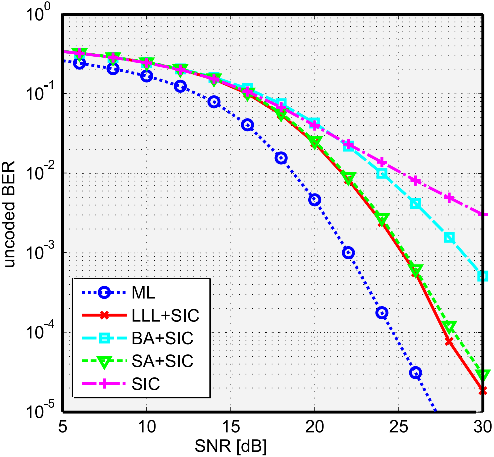
Korkone-Zolotareff algorithm [12]

+ Optimal reduction
– Exponential complexity

Seysen's algorithm [13, 14]

+ Better performance as LLL with linear detector
+ Based on direct matrix inversion
+ Computational complexity similar to S-LLL

Performance Comparison of different LR algorithms using SIC detection



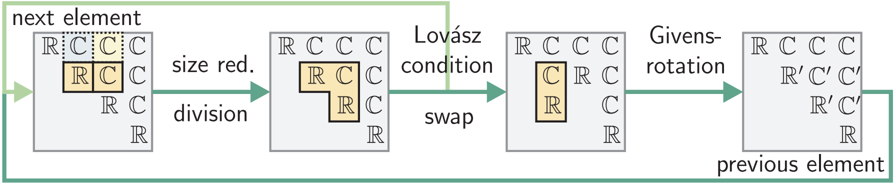
Observation:
For SIC detection Brun's algorithm has a very large BER performance loss.

Conclusion

- For SIC detection the LLL or S-LLL algorithm should be used.
- It can be shown that the S-LLL require lower computation complexity

LLL algorithm for MIMO detection

- Most prominent algorithm for LR-aided MIMO detection
- Starts from the QR-decomposition $\mathbf{H} = \mathbf{QR}$ of the channel



Basis is LLL reduced if:

1. Size is reduced

$$2|r_{i,j}| \leq r_{j,j}, 1 \leq j < i \leq n$$

2. Lovász condition holds

$$\delta|R_{k-1,k-1}|^2 \leq |R_{k,k}|^2 + |R_{k-1,k}|^2,$$
$$\forall k, \quad 0.25 < \delta < 1$$

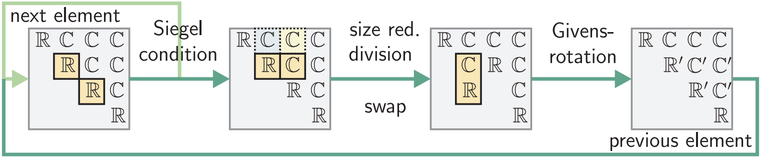
Size reduction requires per iteration:

- $2(k-1)$ real-valued divisions
- $2k\,(k-1)$ real-valued multiplications
- hard to parallelize

Replacing the Lovász condition with the Siegel criterion

Siegel criterion: $\epsilon|R_{k-1,k-1}| \leq |R_{k,k}|, \quad \forall k, \quad 0.25 < \epsilon \leq 0.5$

> Size reduction of $R_{l,k}$ ($\forall l, k$ and $l < k$) does not improve the performance for SIC, is not needed with the Siegel criterion, and hence, can be omitted

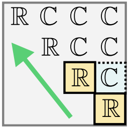


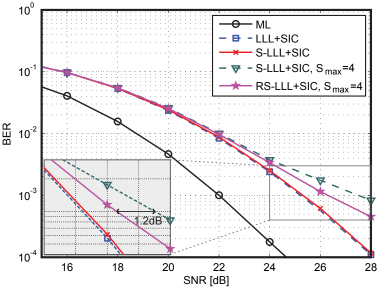
- **Computationally complexity per iteration significantly reduced**
- Worst-case iteration complexity of LLL is unbounded [Jaldén *et al.*, 2008]: Maximum number of iterations **must** be limited in practice
- Algorithms usually proceed from top left to bottom right, reverse procedure possible

Reverse processing order for delay - constrained LR

- Performance of SIC is dominated by weakest stream R_{M_T,M_T} of $\mathbf{R}$
- In delay-constrained systems S-LLL might never process the lower-most diagonal element

First iteration of reverse S-LLL:

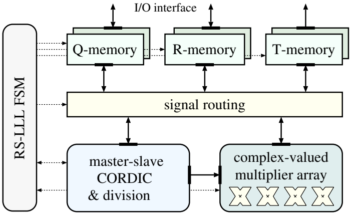


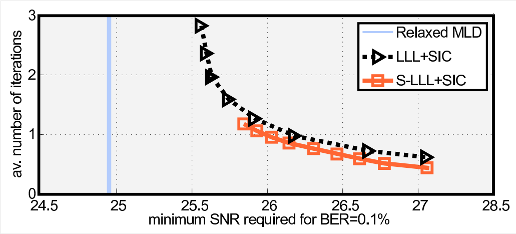


Reverse S-LLL (RS-LLL) guarantees that the weakest stream is processed first

RS-LLL yields substantial improvements in error-rate under tight runtime constraints (e.g., $\mathrm{S}_{\max} = 4$)

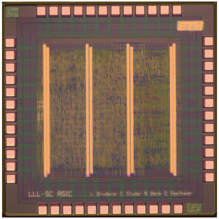
Architecture for RS-LLL





Processing elements:

- Master-slave CORDIC computes
  - the rotation vector for Givens rotation
  - the division
- 4 complex-valued multipliers
  - apply the rotation vectors
  - compute the Siegel criterion



LLL and S-LLL in 130 nm CMOS: 110k GE + QRD (250k GE) with avg. throughput 23.8 M LR/s [BS+10]

[BS+10] L. Bruderer, C. Studer, M. Wenk, D. Seethaler, and A. Burg, “VLSI Implementation of a Low-Complexity LLL Lattice Reduction Algorithm for MIMO Detection,” in *Proc. of IEEE ISCAS*, May. 2010

Seysen's algorithm for MIMO detection

Main characteristic:

- Similar performance as LLL for linear detection
- Seysen's algorithm minimizes the Seysen orthogonality metric: $S(\mathbf{H}) = \sum_{n=1}^{M_T} ||\mathbf{h}_n||^2 \, ||\mathbf{h}_n^{\#}||^2$

Basic Idea:

- Add or subtract integer multiple of columns from other columns until the Seysen metric can no longer be reduced.

$$\mathbf{h}_i = \mathbf{h}_i + \lambda_{i,j}\mathbf{h}_j$$

Iterates the following 4 steps until no further update step, improving the Seysen metric, is fund.

1. Compute all possible integer update factors $\lambda_{i,j}$
2. Computer for each $\lambda_{i,j}$ the impact $\Delta_{i,j}$ on Seysen metric
3. Select the index pair $\{s, t\}$ with largest $\Delta_{i,j}$
4. Update columns $\{s, t\}$ with the factor $\lambda_{s,t}$

Seysen's algorithm based on the Gramian matrices

Reminder

ZF detection estimation matrix: $\mathbf{W}_{\mathrm{ZF}} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H$

MMSE detection estimation matrix: $\mathbf{W}_{\mathrm{MMSE}} = (\mathbf{H}^H\mathbf{H} + M_T\sigma^2\mathbf{I})^{-1}\mathbf{H}^H$

Seysen's algorithm can be efficiently formulated based exclusively on the Gramian matrix $\mathbf{G} = \mathbf{H}^H\mathbf{H}$ and it its dual $\mathbf{G}^{\#} = \mathbf{G}^{-1}$.

Reuse of Gramian

ZF detection estimation matrix: $\mathbf{W}_{\mathrm{ZF}} = \mathbf{G}^{\#}\mathbf{H}^H$

MMSE detection estimation matrix: $\mathbf{W}_{\mathrm{MMSE}} = (\mathbf{G} + M_T\sigma^2\mathbf{I})^{-1}\mathbf{H}^H$

Hermitian property of Gramian and it's dual reduces storage requirements and computational complexity within Seysen's algorithm

λ and Δ Calculation

Step 1: λ calculation:
Compute the integer-rounded candidate update factors $\lambda_{i,j}$ for all possible index pairs i and j with $1 \leq i, j \leq M_T$ and $i \neq j$ as

$$\lambda_{i,j} = \left\lfloor \frac{1}{2} \left(\frac{G^{\#}_{j,i}}{G^{\#}_{i,i}} - \frac{G_{j,i}}{G_{j,j}} \right) \right\rceil$$

$\lambda_{i,j} = 0$ if columns are orthogonal

Step 2: Δ calculation:
For each candidate update factor $\lambda_{i,j}$, quantify the corresponding reduction of the Seysen metric $\Delta_{i,j} = S(\mathbf{H}) - S(\mathbf{H}^{'})$, where $S(\mathbf{H}^{'})$ is the Seysen metric after an update step with $\lambda_{i,j}$, according to

$$\begin{aligned}\Delta_{i,j} = & -2\big(G_{j,j} G^{\#}_{i,i} \, |\lambda_{i,j}|^2 \\ & - G_{j,j} \Re\{\lambda^*_{i,j} G^{\#}_{j,i}\} \\ & + G^{\#}_{i,i} \Re\{\lambda^*_{i,j} G^*_{i,j}\} \big)\end{aligned}$$

Seysen metric is never explicitly calculated

Index Selection and Matrix Update

Step 3: Index selection

- Selection of the indices used for the update step
- Many strategies possible:
    - **Exhaustive** search: Try all possible updates in each iteration results in best BER performance
    - **Greedy** search: Choose index pair $\{s, t\}$ from all possible pairs with largest $\Delta_{i,j}$ in each iteration
    - **K-Select** search: Greedy search over a reduced set of index pair with K elements.
    - **Lazy** search: Choose any index pair with $\Delta_{s,t} \neq 0$

Step 4: Matrix update

Update the matrix $\mathbf{G}$ according to

$$\begin{aligned} G'_{s,j} &= G_{s,j} + \lambda^*_{s,t} G_{t,j}, && j \neq s \\ G'_{j,s} &= G'^*_{s,j}, && j \neq s \\ G'_{s,s} &= G_{s,s} + \\ &\quad 2\Re\{\lambda^*_{s,t} G_{t,s}\} + \\ &\quad |\lambda_{s,t}|^2 \, G_{t,t} \end{aligned}$$

and similar for $\mathbf{G}^{\#}$ and update columns of $\mathbf{T}$ and rows of $\mathbf{T}^{\#}$ according to:

$$\mathbf{t}_s = \mathbf{t}_s + \lambda_{s,t} \mathbf{t}_t$$

$$(\mathbf{t}^{\#})_t = (\mathbf{t}^{\#})_t - \lambda_{s,t} (\mathbf{t}^{\#})_s.$$

Algorithmic Modifications for VLSI Implementation:

Restricting to unity-λ-factors

Nearly all multiplication in the algorithm can be omitted with $\left|\Re\{\lambda_{i,j}\}\right|$ and $\left|\Im\{\lambda_{i,j}\}\right| \leq 1$

Simplifies computation of $\lambda_{i,j}$ to:

$$\Re\{\lambda_{i,j}\} = 0 \text{ if}$$

$$\left|\Re\{G^{\#}_{j,i}G_{j,j} - G_{j,i}G^{\#}_{i,i}\}\right| \leq G^{\#}_{i,i}G_{j,j}$$

else $\Re\{\lambda_{i,j}\}$ is computed

$$\Re\{\lambda_{i,j}\} = \operatorname{sign}\left\{\Re\{G^{\#}_{j,i}G_{j,j} - G_{j,i}G^{\#}_{i,i}\}\right\} \times \operatorname{sign}\left\{G^{\#}_{i,i}G_{j,j}\right\}$$

Regularization

- Regularization of $\mathbf{H}$ results in MMSE detection and improves BER performance
- Reduces the total computational complexity (less iteration needed)
- Minimal additional processing before LR

Index selection scheme

- Evaluation of different index selection schemes required
- K=1 corresponds to lazy search
- For 4×4 MIMO K=12 corresponds to greedy search

Complexity Considerations: Runtime limit required

Guaranteed throughput:
Similar to the S-LLL algorithm there is no theoretical iteration limit of Seysen's algorithm

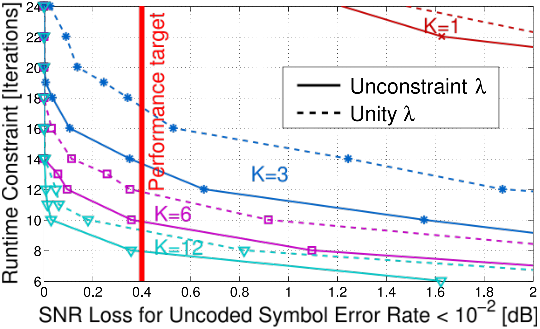
⇒ **Runtime limit for implementation is a must**

Performance considerations:
Every iteration increases the orthogonality of the basis. Hence the BER performance of LR-aided LD is also if LR is aborted, always better then LD without LR

BER-Performance / Iterations Trade-off

The iteration/ runtime limit can be used, to trade BER performance with number of iterations and thereby with **energy consumption per received frame** of the receiver.

Performance evaluation for different K-values, restricted, and unrestricted λ factors dependent on the iteration limit

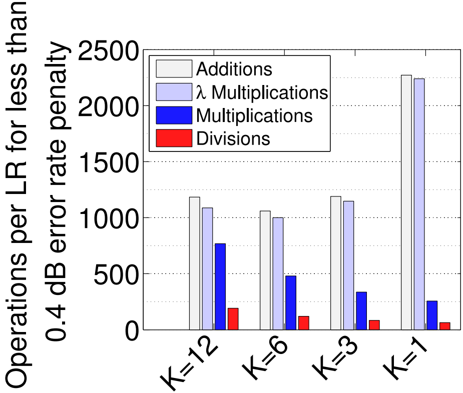


Observations:

- The larger K, the fewer iterations are required for the same BER performance
- Unity-λ updates increase the required iterations only insignificantly

The number of iterations does not directly correspond to the computational complexity of that configuration, as the complexity per iteration differ significantly

Impact of K on the arithmetic operations for a given performance target [BSB10]



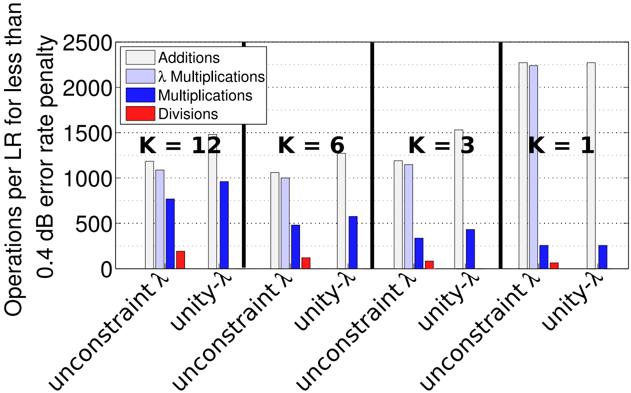
Observations

- The lower K the more additions and λ-multiplications are required
- The lower K the fewer computational complex full multiplications and divisions are required

A lower K reduces the number of computational intense operations per full LR. (But will increases storage requirement and latency of the implementation)

[BSB10] L. Bruderer, C. Senning, A. Burg, "Low-Complexity Seysen's Algorithm based Lattice Reduction-Aided MIMO Detection for Hardware Implementations," in *Proc. of IEEE Asilomar Conf. on Signals, Systems, and Computers*, Nov. 2010

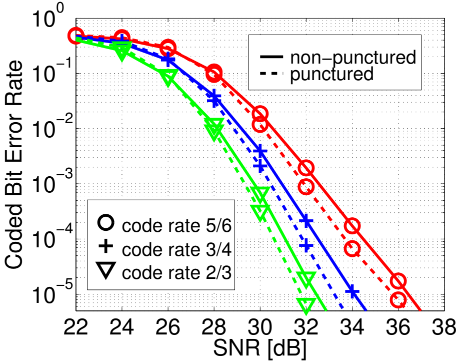
Computational Complexity Impact of Modifications



Observations: For a given performance target the unity-λ factor modification

- **reduces** the number of multiplications (including multiplications with λ) **drastically**
- **omits all divisions**
- increases the required number of additions minimal

Relaxed Lattice Effects [CBH+12]

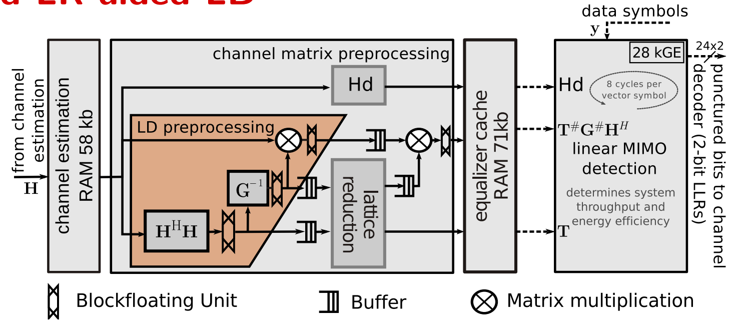


- Neighboring lattice points in the reduced basis do not have to be neighbors in the original lattice
- It is unknown in the reduced basis which lattice point is part of the constellation
- After remapping of detected lattice points into the original lattice, some points could be outside of the valid constellation
- Mapping lattice points outside of the constellation in the original basis to the nearest constellation point results in a reduced BER performance

Puncture known faulty symbols results in better BER performance (up to 0.75 dB) than, mapping to the nearest constellation point in the original basis

[CBH+12] C. Senning, L. Bruderer, J. Hunziker, A. Burg, “A Lattice Reduction-Aided MIMO Channel Equalizer in 90 nm CMOS achieving 720 Mb/s with 28 pJ/bit,” in *In preparation*

High-level Block diagram for Seysen's algorithm based LR-aided LD



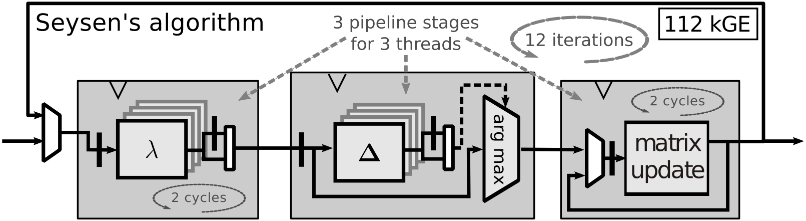
Direct matrix inversion preprocessing for LD is extended by

- a LR computation block
- a matrix multiplication block for the basis transformation $\mathbf{B} = \mathbf{HT}$
- a lattice trans-location unit

Equalizer cache is extended by

- shift vector size
- $\mathbf{T}$ matrix size

High-level Block diagram for SA-based LR



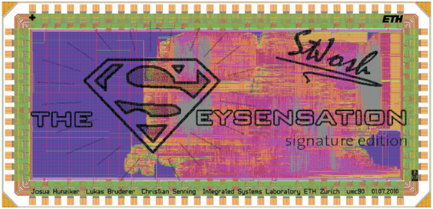
Explanations:

- Pipeline-interleaved architecture
- 2 clock-cycles per pipeline stage
- 3 independent channel matrices (tones) processed in parallel
- Index selection merged with Δ calculation

- Variable runtime with up to 12 iterations (rounds)
- Overtaking LRs have to be **rearranged** outside of the LR-block while writing into an alignment buffer

Implementation results

Unit	**Area [kGE]**
Channel-estimation Ram	64
Gram calculation	28
Gram inversion	99
Sum of block-floating units	31
multiplications	58
Equalizer cache	87
Detector	28
Total Area LD	**395**
LR computation	**112**
multiplications	29
Sum of FIFO	71
Total Area LR-aided LD	**622**

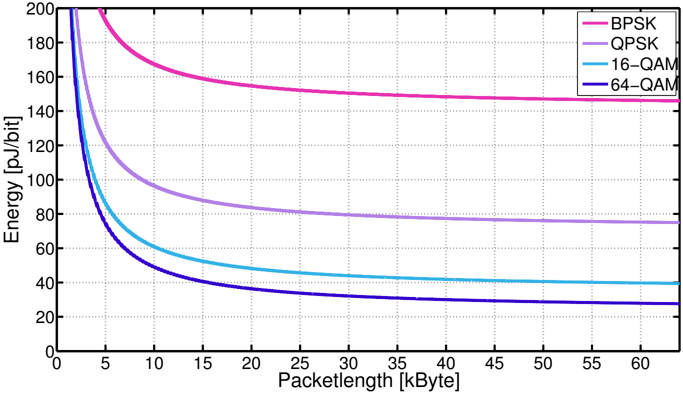

ETH
Stlosh
THE
EYSENSATION
signature edition
Josua Hunziker Lukas Bruderer Christian Senning Integrated Systems Laboratory ETH Zurich umc90 01.07.2010

Seysen's algorithm in 90 nm CMOS: throughput 15 M LR/s [CBH+12] including full preprocessing, buffers, and detection

- Only 18% of the area of the Seysen's algorithm based LR-aided LD is the LR-unit itself
- 82% of the total area is occupied by the channel matrix preprocessing units, running at channel rate

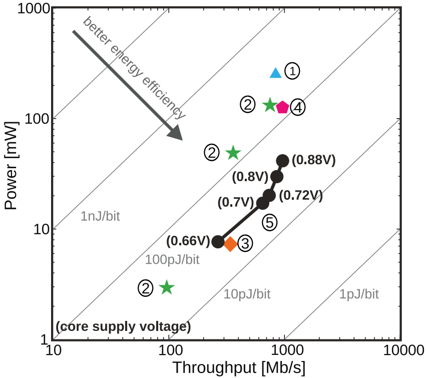
[CBH+12] C. Senning, L. Bruderer, J. Hunziker, A. Burg, "A Lattice Reduction-Aided MIMO Channel Equalizer in 90 nm CMOS achieving 720 Mb/s with 28 pJ/bit," in *In preparation*

Packet length vs energy consumption



The separation of the MIMO detection problem into preprocessing at channel rate and detection at symbol rate results in a strong dependency of the energy consumption on the packet length

Energy-Efficiency Comparison



1. MMSE PIC detector with 4 iterations [a]
2. Strongly constraint sphere 90 nm [b]
3. Strongly constraint sphere 65 nm with SIC performance only [b]
4. K-Best detector 130 nm [b]
5. SA-based LR-aided LD detector [a]

[a] including preprocessing
[b] detection only

Conclusion

The energy efficiency of the SA-based LR-aided LD benefits from the extremely simple detector implementation running at symbol rate and achieves near ML BER performance because of the advanced preprocessing

Exhaustive search ML detection suffers from an exponential increase in complexity with rate

The **optimum detector** solves the maximum likelihood (ML) criterion which is given by

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s} \in \mathcal{O}^{M_T}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2$$

- The ML detection problem can always be solved with an exhaustive search over all possible candidate vector symbols
- Unfortunately, the **complexity** of an exhaustive search **grows exponentially with the rate** R (measured in bits per channel use (bpcu))
- For example, a 4×4 system with 64-QAM modulation $(R = 24)$ requires the consideration of $16\,777\,216$ candidate vector symbols

Exhaustive search ML detection has also some advantages for VLSI implementation

As opposed to linear and SIC algorithms, an exhaustive search offers a number of advantages that greatly facilitate a VLSI implementation:

- Very high regularity, which reduces control overhead and allows for regular hardware architectures
- Few data dependencies allow for massive parallel processing
- No costly operations in terms of silicon area (e.g., divisions, square roots)
- Relaxed numerical requirements

An exhaustive search may yield good results for low rates because

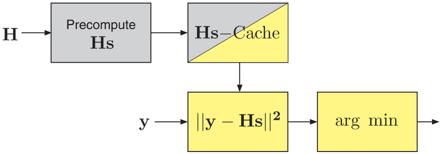
- the complexity order is less relevant in the low-rate regime $(R \leq 12)$
- the suitability for VLSI implementation is often the most important criterion

Optimizations must mitigate the impact of the exponential term in the complexity expression

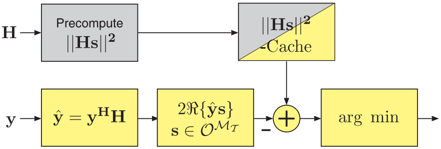
Algorithm transformations for complexity reduction focus on

- exposing common terms that can be precomputed once (Example: Based on the CSI)
- reducing the number of costly operations at the symbol-rate, possibly at the expense of additional, but less costly operations
- minimizing the memory required to store precomputed results

$$\underset{\mathbf{s}\in\mathcal{O}^{M_T}}{\arg\min}\|\mathbf{y}-\mathbf{Hs}\|^2 \quad\rightarrow\quad \underset{\mathbf{s}\in\mathcal{O}^{M_T}}{\arg\min}\left(\|\mathbf{Hs}\|^2-2\Re\{(\mathbf{y}^H\mathbf{H})\mathbf{s}\}\right)$$



[15]


$\mathbf{H}$
Precompute $||\mathbf{Hs}||^2$
$||\mathbf{Hs}||^2$-Cache
$\mathbf{y}$
$\hat{\mathbf{y}}=\mathbf{y}^{\mathbf{H}}\mathbf{H}$
$2\Re\{\hat{\mathbf{y}}\mathbf{s}\}$ $\mathbf{s}\in\mathcal{O}^{\mathcal{M}_\mathcal{T}}$
+
–
arg min

Proposed architecture [16]

An exhaustive search is feasible for $R \leq 12$, but becomes prohibitively complex for higher rates

The proposed optimizations lead to a considerable complexity reduction in the implementation of an exhaustive search:

		[15]		Proposed [16]	
$M_T \times M_R$	Rate R	Throughput	Area [GE]	Throughput	Area [GE]
2×2	4 bpcu	9.6 Mbps	93K	100 Mbps	13K
4×4	**8 bpcu**	19.2 Mbps	140K	**50 Mbps**	**42K**
6×6	12 bpcu	n.a.	n.a.	18.8 Mbps	160K

For $R >$12 bpcu the complexity of an exhaustive search becomes prohibitive

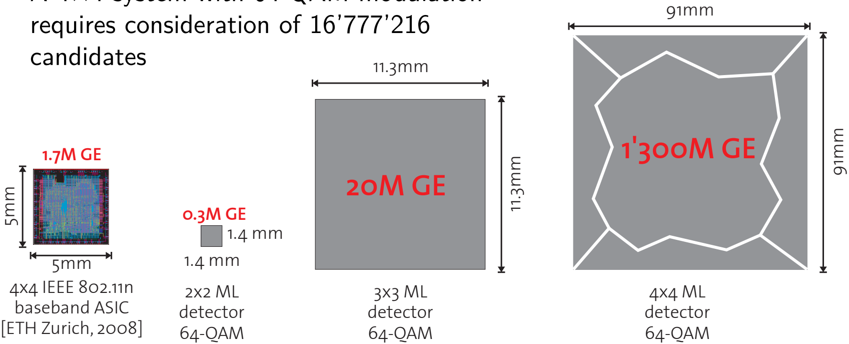
The key to complexity reduction is to

- constrain the search to a subset of the symbol alphabet ($\mathcal{O}^{M_T}$)
- Enumerate the candidates in a clever fashion.
- preferably without accidentally excluding the ML solution.

Maximum-likelihood (ML) detection

Exhaustive search: Enumerate all possible candidate vectors

- Number of candidate vectors grows exponentially in the number of transmit antennas M_T
- A 4×4 system with 64-QAM modulation requires consideration of 16’777’216 candidates



Exhaustive search is not economic for more than two spatial streams

QR-decomposition allows to compute partial Euclidean distances (PEDs)

With the QR-decomposition of $\mathbf{H}$ a modified input-output relation can be obtained according to

$$\hat{\mathbf{y}} = \mathbf{R}\mathbf{s} + \mathbf{Q}^H\mathbf{n} \quad \text{with} \quad \hat{\mathbf{y}} = \mathbf{Q}^H\mathbf{y}$$

The corresponding ML criterion is given by

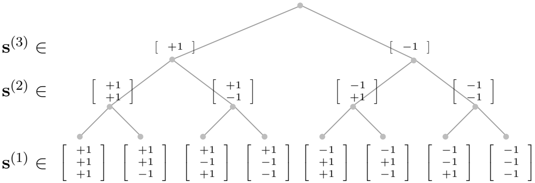
$$\hat{\mathbf{s}} = \underset{\mathbf{s}\in\mathcal{O}^{M_T}}{\arg\min}\, d(\mathbf{s}) \quad \text{with} \quad d(\mathbf{s}) = \|\hat{\mathbf{y}} - \mathbf{R}\mathbf{s}\|^2$$

Computing partial Euclidean distances (PED):

- Define $\mathbf{e} = \hat{\mathbf{y}} - \mathbf{R}\mathbf{s}$ and $\mathbf{e}^{(i)} = \left[\begin{array}{cccc} e_i & e_{i+1} & \dots & e_{M_T} \end{array}\right]$.
- The PED d_i is given by $d_i = \|\mathbf{e}^{(i)}\|^2$ which depends only on $\mathbf{s}^{(i)} = \left[\begin{array}{cccc} s_i & s_{i+1} & \dots & s_{M_T} \end{array}\right]$ because $\mathbf{R}$ is triangular.

The ML detection problem can be mapped to a tree-search problem

All possible incarnations of $\mathbf{s}^{(i)}$ and the corresponding PEDs d_i can be associated with the nodes of a tree, with the root in $i = M_T + 1$:


$\mathbf{s}^{(3)} \in$
$[+1]$
$[-1]$
$\mathbf{s}^{(2)} \in$
$\begin{bmatrix}+1\\+1\end{bmatrix}$
$\begin{bmatrix}+1\\-1\end{bmatrix}$
$\begin{bmatrix}-1\\+1\end{bmatrix}$
$\begin{bmatrix}-1\\-1\end{bmatrix}$
$\mathbf{s}^{(1)} \in$
$\begin{bmatrix}+1\\+1\\+1\end{bmatrix}$
$\begin{bmatrix}+1\\+1\\-1\end{bmatrix}$
$\begin{bmatrix}+1\\-1\\+1\end{bmatrix}$
$\begin{bmatrix}+1\\-1\\-1\end{bmatrix}$
$\begin{bmatrix}-1\\+1\\+1\end{bmatrix}$
$\begin{bmatrix}-1\\+1\\-1\end{bmatrix}$
$\begin{bmatrix}-1\\-1\\+1\end{bmatrix}$
$\begin{bmatrix}-1\\-1\\-1\end{bmatrix}$

- The leaf with the smallest d_1 corresponds to the ML solution
- The PEDs can be computed recursively while traversing the tree

$$d_i = d_{i+1} + |e_i|^2 \quad \text{with} \quad |e_i|^2 = \left|\hat{y}_i - \sum_{j=i+1}^{M_T} R_{ij} s_j - R_{ii} s_i\right|^2$$

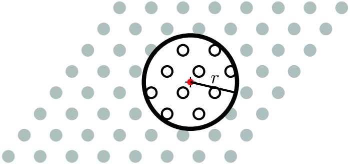
- Complexity reduction can be achieved by **pruning parts of the tree** which are at least unlikely to contain the ML solution

Sphere decoding (SD) [17, 18, 19] reduces the number of candidate vector symbols

SD considers only candidate vector symbols for which $\mathbf{R}\mathbf{s}$ lies within a sphere with radius r around $\hat{\mathbf{y}}$

- The corresponding inequality is given by the **sphere constraint** (SC):

$$d_1 < r^2$$



- Since $d_i \geq d_{i+1}$, the SC can be applied to the PEDs of all nodes in the tree: $d_i < r^2$

Rule for pruning the tree:

- Nodes, violating the partial SC $d_i < r^2$ can be pruned together with all their children.

- **Tree traversal for SD should be performed depth-first**

Radius reduction with Schnorr-Euchner [20] enumeration improves tree-pruning efficiency

The choice of the radius r is critical, because it governs the efficiency of the tree pruning.

Radius reduction (RR):

- The initial radius is set to $r = \infty$
- When a leaf is reached, the radius can be updated according to

$$r \leftarrow d_1$$

With RR, tree pruning is more efficient, when a good solution is found as early as possible

Schnorr-Euchner (SE) enumeration:

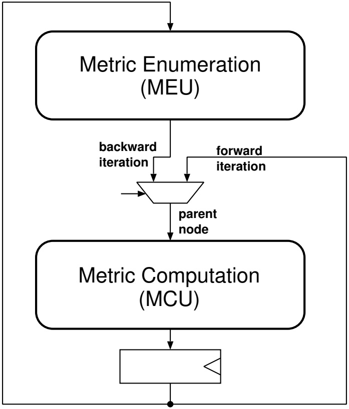
- With SE ordering the children of a node are visited in ascending order of their PEDs
- The result is a more rapid shrinkage of the sphere

One-node-per-cycle VLSI architecture [21]

With RR, parallel processing leads to poor hardware utilization.

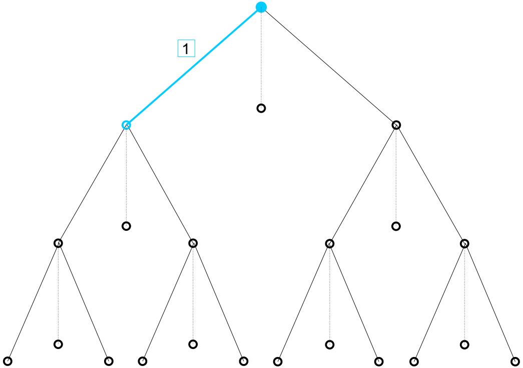
A **one-node-per-cycle architecture** ensures that **a new node is visited in each cycle** and no node is visited twice.

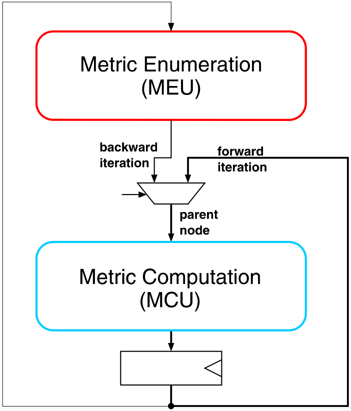
- **Metric computation unit** (MCU) handles the forward iteration
- **Metric enumeration unit** (MEU) handles the backward iteration



One-node-per-cycle VLSI architecture

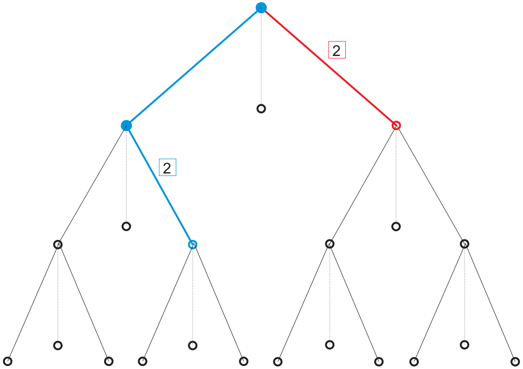
- MCU considers the children of the current node and finds the starting point for the SE enumeration
- The MEU is inactive in the first cycle

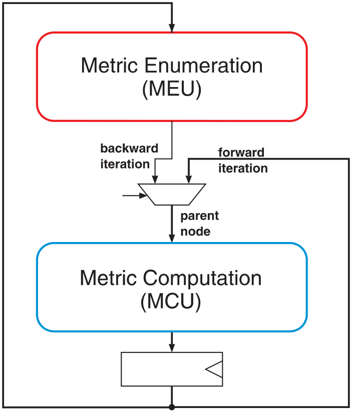




One-node-per-cycle VLSI architecture

- MCU advances to the next level
- The MEU follows the MCU on its path and considers the siblings of the current node to constructs a list of preferred children





Metric Enumeration (MEU)

backward iteration

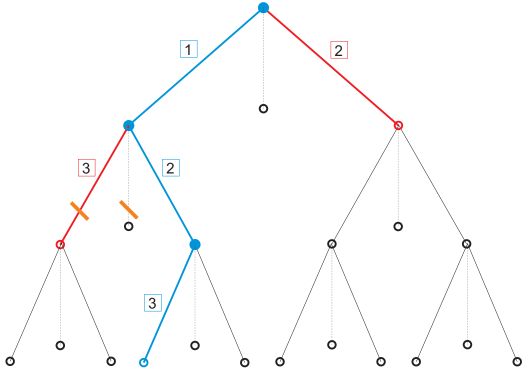
forward iteration

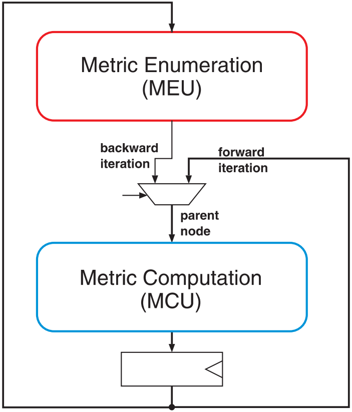
parent node

Metric Computation (MCU)

One-node-per-cycle VLSI architecture

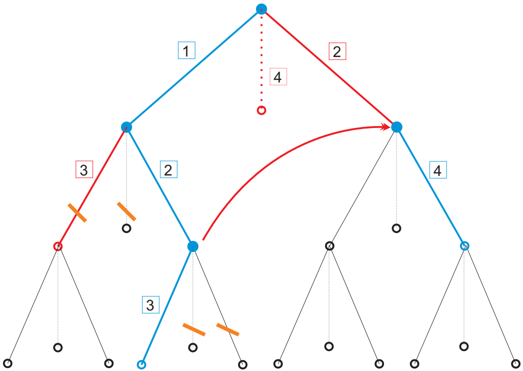
- When the MCU reaches a leaf, the radius is shrunk
- The list of preferred children in the MEU is ordered depth-first and membership is conditioned on compliance with the SC

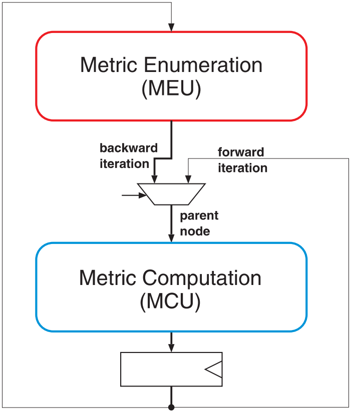




One-node-per-cycle VLSI architecture

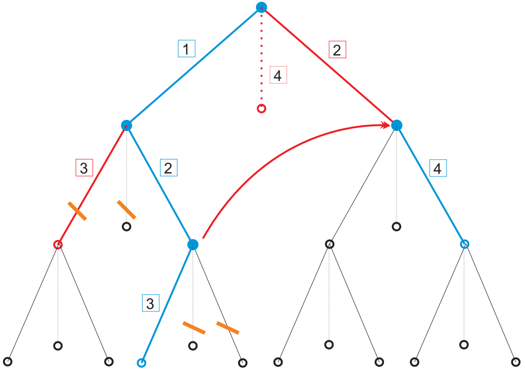
- When the forward iteration stalls, the MEU provides the PED of a new node to the MCU
- The MCU can immediately consider the children of a new node

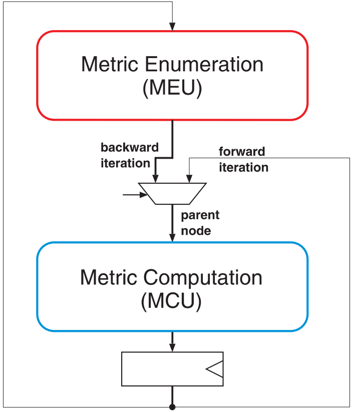




One-node-per-cycle VLSI architecture

- The average throughput Φ is determined by the average number of visited nodes $\mathcal{E}\{D\}$ and by the cycle time t_{pd}
- $\Phi = (M_T \log_2 \mathcal{O}) \,/\, (\mathcal{E}\{D\} t_{\text{pd}})$





The clock frequency is limited by the calculation of the ℓ^2-norm in the SC.

Simplified norm increases clock frequency and thereby, increase efficiency in terms of circuit area times processing time.

Modified norm algorithm [21]:

- Take the square root of the PED and substitute $x_i = \sqrt{d_i}$

$$x_i = \sqrt{x_{i+1}^2 + |e_i|^2} < r$$

- Replace the ℓ^2-norm with an approximation of the form $\sqrt{a^2 + b^2} \approx f(|a|, |b|)$ to obtain

$$x_i \approx f(x_{i+1}, |e_i|), \quad \text{where} \quad |e_i| \approx f(\Re\{e_i\}, \Im\{e_i\})$$

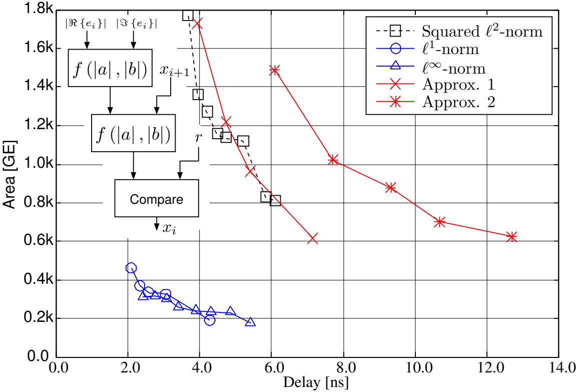
Low-complexity approximations of the ℓ^2-norm are given by

	$f(\|a\|, \|b\|)$
ℓ^1-norm	$\|a\| + \|b\|$
ℓ^∞-norm	$\max(\|a\|, \|b\|)$
Approx.1	$\frac{3}{8}(\|a\| + \|b\|) + \frac{5}{8}\max(\|a\|, \|b\|)$
Approx.2	$\max\left(\max(\|a\|, \|b\|), \frac{7}{8}\max(\|a\|, \|b\|) + \frac{1}{2}\min(\|a\|, \|b\|)\right)$

The ℓ^1- and the ℓ^∞-norm reduce the silicon area and the delay of the PED computation circuit

Explore the area/delay tradeoffs for the norm approximations:

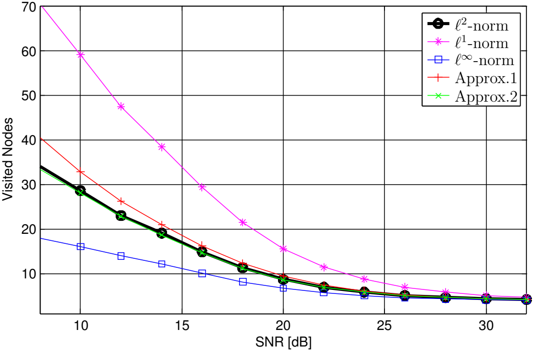
- Approx. 1 & 2 yield no advantages
- **The ℓ^1- and the ℓ^∞-norm reduce both silicon area and the critical path** compared to the squared ℓ^2-norm circuit



The choice of the norm approximation affects the efficiency of the tree-pruning

Consider the impact on the number of visited nodes:

- Using the ℓ^1-norm **increases** the number of visited nodes
- Using the ℓ^∞-norm **reduces** the number of visited nodes

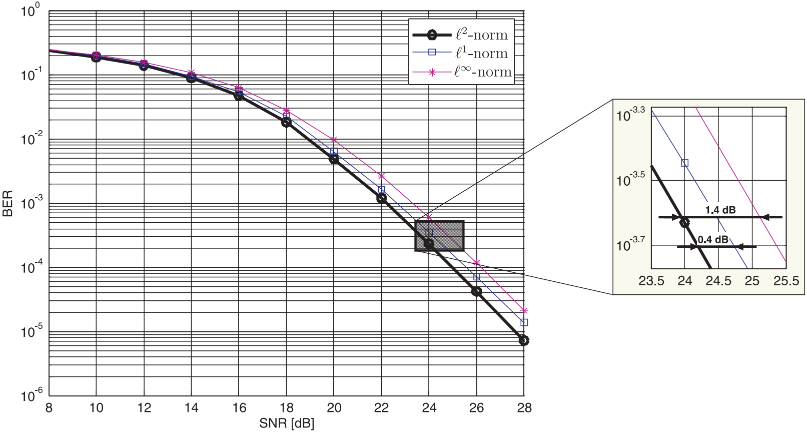


- The ℓ^∞-norm is best suited for VLSI implementation because it improves performance on algorithm and circuit level.

The choice of the norm has a small impact on BER performance, but preserves diversity

The modified-norm algorithm is no longer ML detection rule

- Using the ℓ^1-norm degrades the BER performance by 0.4 dB
- The ℓ^∞**-norm algorithm entails a 1.4 dB SNR penalty**



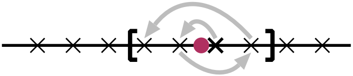
For QAM identifying admissible children and performing SE enumeration is difficult

For real-valued constellation points, the solution is simple:

- Solving $d_i < r^2$ for s_i yields an admissible interval

$$d_i = d_{i+1} + \left| \hat{y}_i - \sum_{j=i+1}^{M_T} R_{ij} s_j - R_{ii} s_i \right|^2$$

- SE ordering [20] proceeds from the center in a zig-zag fashion [19]



Real-valued decomposition (RVD) for QAM:

- The the M_T-dimensional complex-valued problem can be decomposed into a $2M_T$-dimensional real-valued problem:

$$\begin{bmatrix} \Re\{\mathbf{y}\} \\ \Im\{\mathbf{y}\} \end{bmatrix} = \begin{bmatrix} \Re\{\mathbf{H}\} & -\Im\{\mathbf{H}\} \\ \Im\{\mathbf{H}\} & \Re\{\mathbf{H}\} \end{bmatrix} \begin{bmatrix} \Re\{\mathbf{s}\} \\ \Im\{\mathbf{s}\} \end{bmatrix} + \begin{bmatrix} \Re\{\mathbf{n}\} \\ \Im\{\mathbf{n}\} \end{bmatrix}$$

- **Because the RVD increases the number of visited nodes, it is ill-suited for a one-node-per-cycle architecture.**

Exhaustive search enumeration [22] allows to operate directly on complex-valued constellations

Exhaustive search enumeration:

- At each node, compute the PEDs of all $|\mathcal{O}|$ children
- Identify the admissible children by checking the SC
- Perform SE enumeration by explicitly sorting the children in ascending order of their PEDs

- Complexity is reduced by exposing common terms in the computation of $|e_i|^2$ for all $s_i \in \mathcal{O}$:

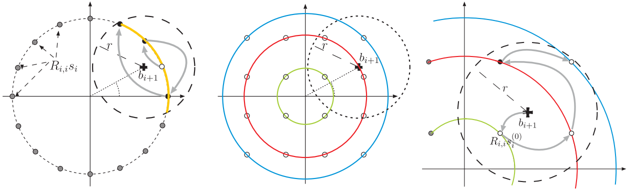
$$\left|R_{i,i}s_i - b_{i+1}\right|^2 = \left|R_{i,i}\right|^2 \left|s_i\right|^2 - 2\Re\left\{\left(R_{i,i}b_{i+1}\right)s_i^*\right\} + \left|b_{i+1}\right|^2,$$

  where $b_{i+1} = y_i - \sum_{j=i+1}^{M_T} R_{i,j}s_j$.
- **The drawbacks are the amount of memory required to store the PEDs and the need to find the minimum.**

The Hochwald/ten Brink scheme [23] allows for direct enumeration of QAM constellations

Direct-QAM enumeration:

- For PSK constellations, admissible intervals can be defined, based on the phase of the constellation points
- QAM constellation can be split into PSK subsets
- Within each 1-dimensional subset, SE enumeration follows a zig-zag pattern
- Enumeration across subsets is achieved by comparing the PEDs



Identifying the starting point within each subset and computing boundaries of admissible intervals requires costly trigonometric functions.

Introduction of decision boundaries reduces complexity of direct-PSK/QAM enumeration

The starting point for the SE enumeration in the kth PSK subset $\mathcal{O}^{(k)}$ can be identified based on the phase of b_{i+1} according to

$$s_i^{(k)} = \underset{s_i \in \mathcal{O}^{(k)}}{\arg\min} \left|\mathrm{arc}(b_{i+1}) - \mathrm{arc}(s_i)\right|$$

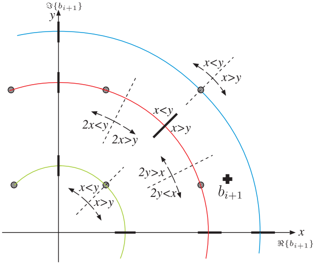
- The introduction of decision boundaries yields

$$s_i^{(k)} = \{\mathcal{O}^{(k)}\}_n \quad \text{if} \quad \alpha_n^k < \mathrm{arc}\, b_{i+1} \leq \alpha_{n+1}^k$$

- Operating on the tangent of those decision boundaries yields inequalities of the form

$$X\Re\{b_{i+1}\} \gtrless Y\Im\{b_{i+1}\} \text{ with } X, Y \in \mathbb{N}.$$

to find the *starting point* for the enumeration and the *initial direction*



Subset enumeration for higher-order constellations

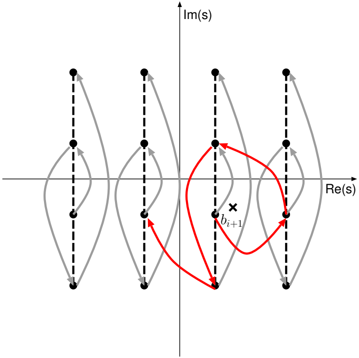
For 64-QAM or more, enumeration on PSK subsets becomes tedious

- Number of subsets grows rapidly
- Decision boundaries become tedious and more complex
- PSK enumeration is no longer strictly optimal (though performance penalty is hardly visible)

Subset enumeration based on PAM

- Subsets can either be based on $\Re\{b_{i+1}\}$ or $\Im\{b_{i+1}\}$
- 64-QAM requires 8 instead of 9 subsets compared to PSK
- Decision rules simplify to comparing to constants

$$\Re\{b_{i+1}\} \gtrless Y \text{ with } Y \in \mathbb{N}.$$



Approximate enumeration reduces complexity

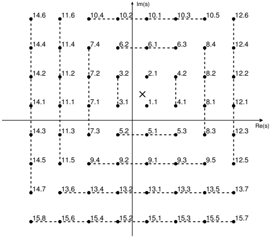
Subset enumeration has a few disadvantages

- Number of subsets is high for 64-QAM and above
- For each subset, the PED of one candidate must be computed and stored
- Enumeration across subsets must still be performed by explicit PED comparison (min-search)

ℓ^{∞}-norm enumeration enumerate constellation points based on

$$\max\{|\Re(b_{i+1} - R_{i,i}s_i)|, |\Im(b_{i+1} - R_{i,i}s_i)|\}$$

- Identify the closest point
- Use simple decision rules to localize b_{i+1} in one of 8 sectors around the closest point
- Enumeration follows a pattern that can be generated with only three small counters and a very small amount of combinational logic

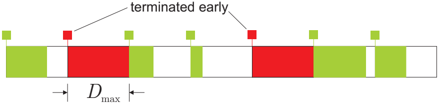


Runtime constraints can be used to enforce a guaranteed minimum throughput

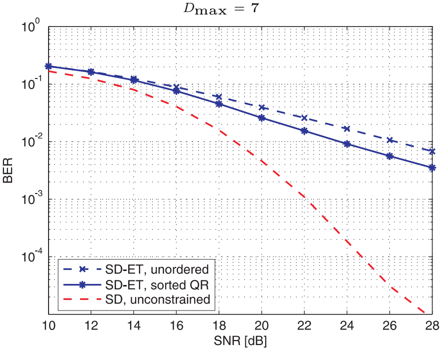
In real systems, the maximum decoding effort must be constrained:

Early termination:

The decoder **stops after $D_{\max}$ visited nodes** and **returns the so far best estimate**.



- **Sorted QR preprocessing improves BER performance**
- Resource utilization is poor
- **BER performance is poor**



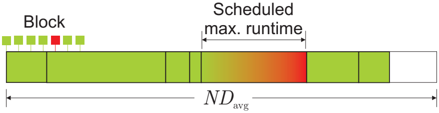
Block processing [24] reduces BER performance penalty from early termination

In real systems, the maximum decoding effort must be constrained:

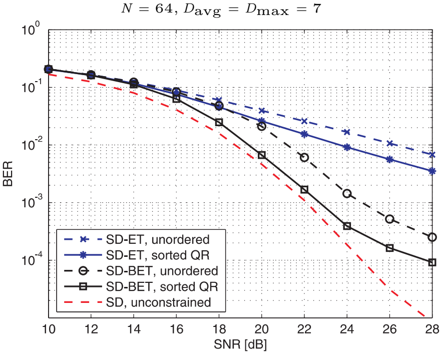
Early termination with block processing:

The **aggregate decoding effort** for a block of N symbols is **constrained to** ND_{avg} **visited nodes**.

A scheduler determines the max. runtime limit for each symbol:

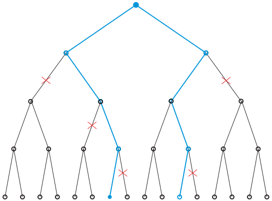


- **Early termination with block processing achieves close-to ML performance**



The K-best algorithm [25] traverses the tree with a constraint on the maximum decoding effort

- Tree traversal is performed breadth-first
- On each level of the tree, the decoder visits only (up to) K parent nodes and computes the PEDs of their children
- The K-best children of these nodes are selected as parent nodes for the next level



Rule for pruning the tree:

- The amount of pruning is governed by a runtime constraint
- The selection of the survivor-nodes is guided by the PED

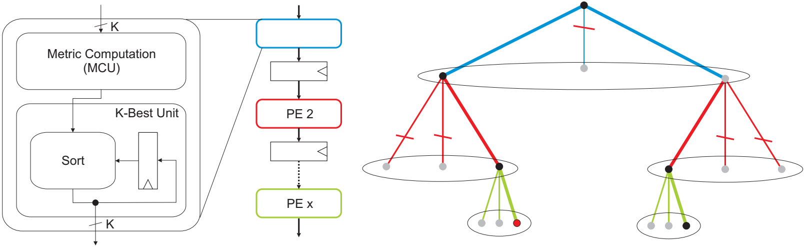
Impact on BER performance

- This pruning strategy cannot guarantee ML performance
- The BER will depend on the design parameter K

The fixed complexity of K-best decoding favors a pipelined VLSI architecture [25, 26]

The architecture is a pipelined array of processing elements (PEs)

- Each pipeline stage (PE) considers one level of the tree.
- The PEs receive K parent nodes from the proceeding PE
  - The MCU computes the PEDs of the corresponding children
  - The KBU chooses the K best children to be passed on to the next PE



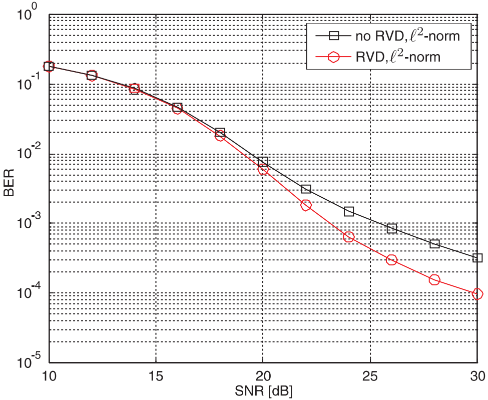
- Finding the K best children is costly. Hence, the **PEs distribute the computation over multiple cycles** (iterative decomposition).

The K-best algorithm favors the use of the real-valued decomposition [28]

The K-best decoder can operate directly on the complex-valued constellations or use the real-valued decomposition:

The RVD provides better BER:

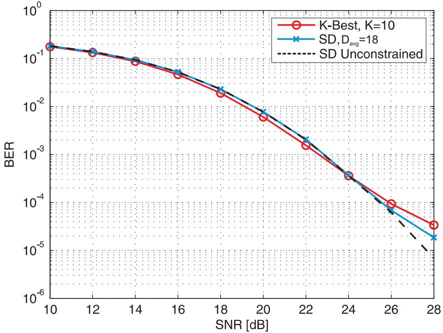
- Ordering is done for real and imaginary parts separately
- The 2nd quadrature component already sees 2nd order diversity [27]
- A similar result has been obtained for V-BLAST

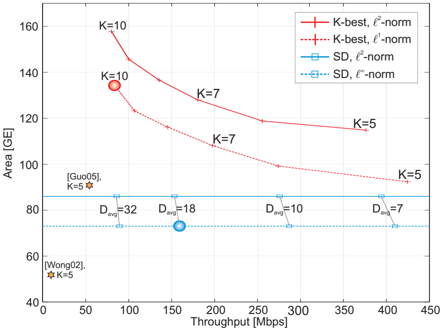


- The **RVD does not affect throughput** (because all levels of the tree are processed in parallel), **but improves BER performance**

Sphere decoding with block processing outperforms K-best decoding [24]

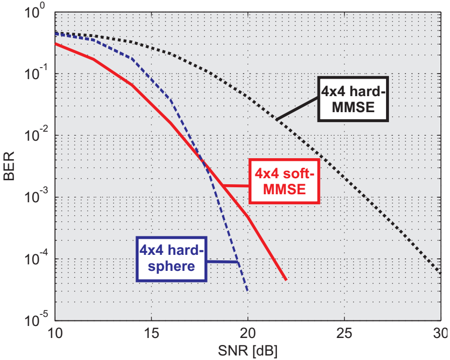
Compare sphere decoding and K-best decoding for a 4×4 system with 16-QAM modulation, implemented in a 0.25μm technology:





- For better BER, SD with $D_{\text{avg}} = 18$ provides almost twice the throughput at half the silicon area of a K-best decoder with $K = 10$

Hard-output ML detection competes with soft-output linear detection



4×4, 16-QAM, MIMO-OFDM, $R = 1/2$, TGn-C channel

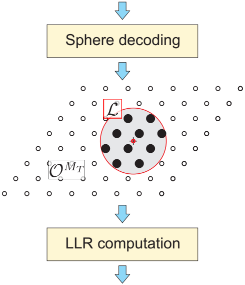
The lack of soft-information limits the advantage of hard-output ML

Approximate LLRs can be computed using the list-sphere decoding (LSD) algorithm [23]

List-sphere decoding algorithm

1. Obtain a reduced set $\mathcal{L}$ (list) of $|\mathcal{L}|$ candidate vector symbols for which $\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2$ is small
2. Compute LLRs based on $\mathcal{L}$ instead of $\mathcal{O}^{M_T}$

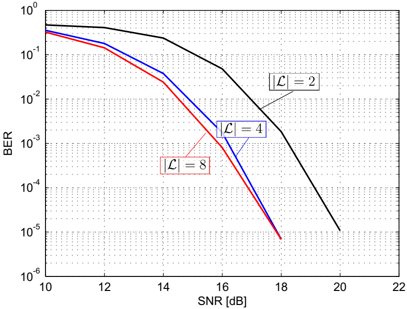
$$L\left(x_{j,b}\right)=\min_{\mathbf{s}\in\mathcal{L}_{j,b}^{(0)}}\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2-\min_{\mathbf{s}\in\mathcal{L}_{j,b}^{(1)}}\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2$$

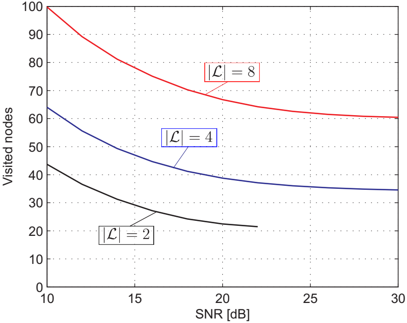


Construction of the list $\mathcal{L}$

- Initialize all entries of $\mathcal{L}$ and the radius r with infinity
- Perform depth-first tree traversal. When a leaf is reached
  - Insert the PED and its label $\mathbf{s}$ into $\mathcal{L}$
  - When the list is full, remove the entry with the largest PED
  - Update the radius: $r \leftarrow \max_{l_i\in\mathcal{L}} l_i$

For the LSD algorithm the list-size determines the complexity/performance trade-off





Complexity considerations

- A large list improves BER performance
- Complexity for identifying the entries of the list increases with $|\mathcal{L}|$
- Large lists increase the complexity of the LLR computation and the hardware overhead for the list administration during tree-traversal

Use sphere decoding for computation of LLRs

Max-log approximation for LLRs

$$L(x_{j,b}) = \min_{\mathbf{s}\in\mathcal{X}_{j,b}^{(0)}} ||\mathbf{y}-\mathbf{H}\mathbf{s}||^2 - \min_{\mathbf{s}\in\mathcal{X}_{j,b}^{(1)}} ||\mathbf{y}-\mathbf{H}\mathbf{s}||^2$$

$\mathcal{X}_{j,b}^{(0)}, \mathcal{X}_{j,b}^{(1)}$ ... sets of symbol vectors for which $x_{j,b} = 0, 1$

Use sphere decoding for computation of LLRs

Max-log approximation for LLRs

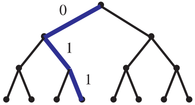
$$L(x_{j,b}) = \min_{\mathbf{s}\in\mathcal{X}_{j,b}^{(0)}} ||\mathbf{y}-\mathbf{H}\mathbf{s}||^2 - \min_{\mathbf{s}\in\mathcal{X}_{j,b}^{(1)}} ||\mathbf{y}-\mathbf{H}\mathbf{s}||^2$$

$\mathcal{X}_{j,b}^{(0)}, \mathcal{X}_{j,b}^{(1)}$ ... sets of symbol vectors for which $x_{j,b} = 0, 1$

Repeated tree-search algorithm (RTS) [WG04]

- For each bit, one of the two minima corresponds to the ML solution
- Each *counter-hypothesis* is found through a separate search

$x^{\mathrm{ML}} = [0\,1\,1]$



[WG04] R. Wang and G. B. Giannakis, "Approaching MIMO channel capacity with reduced-complexity soft sphere decoding," in *Proc. of IEEE Wireless Communications and Networking Conf. (WCNC)*, vol. 3, Mar. 2004, pp. 1620–1625

Use sphere decoding for computation of LLRs

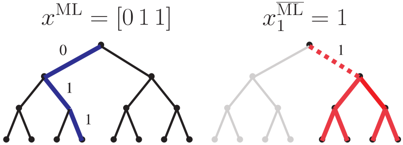
Max-log approximation for LLRs

$$L\left(x_{j,b}\right) = \min_{\mathbf{s}\in\mathcal{X}_{j,b}^{(0)}} \left\|\mathbf{y}-\mathbf{H}\mathbf{s}\right\|^2 - \min_{\mathbf{s}\in\mathcal{X}_{j,b}^{(1)}} \left\|\mathbf{y}-\mathbf{H}\mathbf{s}\right\|^2$$

$\mathcal{X}_{j,b}^{(0)}, \mathcal{X}_{j,b}^{(1)}$ ... sets of symbol vectors for which $x_{j,b} = 0, 1$

Repeated tree-search algorithm (RTS) [WG04]

- For each bit, one of the two minima corresponds to the ML solution
- Each *counter-hypothesis* is found through a separate search


$x^{\mathrm{ML}} = [0\,1\,1]$
0
1
1
$x_1^{\overline{\mathrm{ML}}} = 1$
1

[WG04] R. Wang and G. B. Giannakis, "Approaching MIMO channel capacity with reduced-complexity soft sphere decoding," in *Proc. of IEEE Wireless Communications and Networking Conf. (WCNC)*, vol. 3, Mar. 2004, pp. 1620–1625

Use sphere decoding for computation of LLRs

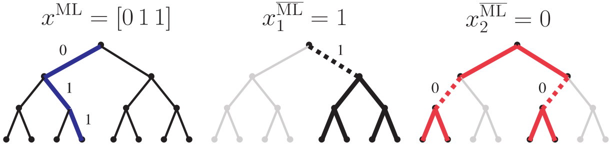
Max-log approximation for LLRs

$$L\left(x_{j,b}\right) = \min_{\mathbf{s}\in\mathcal{X}_{j,b}^{(0)}} ||\mathbf{y}-\mathbf{H}\mathbf{s}||^2 - \min_{\mathbf{s}\in\mathcal{X}_{j,b}^{(1)}} ||\mathbf{y}-\mathbf{H}\mathbf{s}||^2$$

$\mathcal{X}_{j,b}^{(0)}, \mathcal{X}_{j,b}^{(1)}$ … sets of symbol vectors for which $x_{j,b} = 0, 1$

Repeated tree-search algorithm (RTS) [WG04]

- For each bit, one of the two minima corresponds to the ML solution
- Each *counter-hypothesis* is found through a separate search


$x^{\mathrm{ML}} = [0\,1\,1]$ $\qquad x_1^{\overline{\mathrm{ML}}} = 1$ $\qquad x_2^{\overline{\mathrm{ML}}} = 0$

[WG04] R. Wang and G. B. Giannakis, "Approaching MIMO channel capacity with reduced-complexity soft sphere decoding," in *Proc. of IEEE Wireless Communications and Networking Conf. (WCNC)*, vol. 3, Mar. 2004, pp. 1620–1625

Use sphere decoding for computation of LLRs

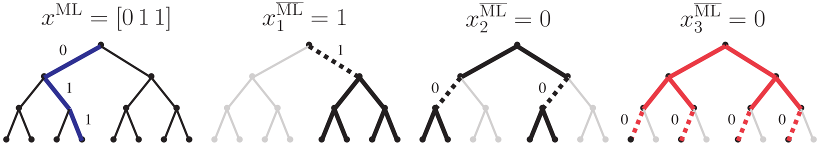
Max-log approximation for LLRs

$$L\left(x_{j,b}\right) = \min_{\mathbf{s}\in\mathcal{X}_{j,b}^{(0)}} \left|\left|\mathbf{y}-\mathbf{Hs}\right|\right|^2 - \min_{\mathbf{s}\in\mathcal{X}_{j,b}^{(1)}} \left|\left|\mathbf{y}-\mathbf{Hs}\right|\right|^2$$

$\mathcal{X}_{j,b}^{(0)}, \mathcal{X}_{j,b}^{(1)}$ … sets of symbol vectors for which $x_{j,b} = 0, 1$

Repeated tree-search algorithm (RTS) [WG04]

- For each bit, one of the two minima corresponds to the ML solution
- Each *counter-hypothesis* is found through a separate search


$x^{\mathrm{ML}} = [0\ 1\ 1]$
0
1
1
$x_1^{\overline{\mathrm{ML}}} = 1$
1
$x_2^{\overline{\mathrm{ML}}} = 0$
0
0
$x_3^{\overline{\mathrm{ML}}} = 0$
0
0
0
0

[WG04] R. Wang and G. B. Giannakis, "Approaching MIMO channel capacity with reduced-complexity soft sphere decoding," in *Proc. of IEEE Wireless Communications and Networking Conf. (WCNC)*, vol. 3, Mar. 2004, pp. 1620–1625

Single Tree Search (STS) Algorithm [23] [29]

- Ensure that every node in the tree is visited at most once

- Search for the ML solution and all counter-hypotheses concurrently

- Maintain a list containing
  - the ML hypothesis $\mathbf{x}^{\mathrm{ML}}$ and its metric λ^{ML}
  - the metrics of the counter-hypotheses $\lambda_{j,b}^{\overline{\mathrm{ML}}}$

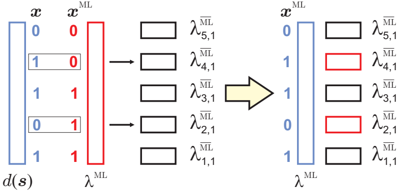
- Search a sub-tree only if the result can lead to an update of either λ^{ML} or of at least one of the $\lambda_{j,b}^{\overline{\mathrm{ML}}}$

STS List Administration

- Initialization of the list: Set $\lambda^{\mathrm{ML}} = \infty$ and $\lambda_{j,b}^{\overline{\mathrm{ML}}} = \infty$ for all j, b
- At a leaf (with label s), the decoder distinguishes between two cases:

$d(\mathrm{s}) < \lambda^{\mathrm{ML}}$: new ML hypothesis

- Update counter-hypotheses: $\lambda_{j,b}^{\overline{\mathrm{ML}}} \leftarrow \lambda^{\mathrm{ML}}$ for $x_{j,b} = x_{j,b}^{\overline{\mathrm{ML}}}$
- Update the ML hypothesis according to $\lambda^{\mathrm{ML}} \leftarrow d(\mathrm{s})$



$\boldsymbol{x}$
$\boldsymbol{x}^{\mathrm{ML}}$
0 0
1 0
1 1
0 1
1 1
$d(\boldsymbol{s})$
λ^{ML}
$\lambda^{\overline{\mathrm{ML}}}_{5,1}$
$\lambda^{\overline{\mathrm{ML}}}_{4,1}$
$\lambda^{\overline{\mathrm{ML}}}_{3,1}$
$\lambda^{\overline{\mathrm{ML}}}_{2,1}$
$\lambda^{\overline{\mathrm{ML}}}_{1,1}$
$\boldsymbol{x}^{\mathrm{ML}}$
0
1
1
0
1
λ^{ML}
$\lambda^{\overline{\mathrm{ML}}}_{5,1}$
$\lambda^{\overline{\mathrm{ML}}}_{4,1}$
$\lambda^{\overline{\mathrm{ML}}}_{3,1}$
$\lambda^{\overline{\mathrm{ML}}}_{2,1}$
$\lambda^{\overline{\mathrm{ML}}}_{1,1}$

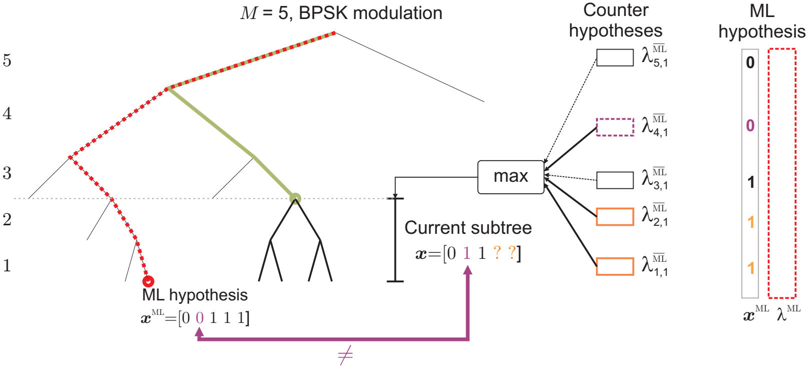
$d(\mathrm{s}) \geq \lambda^{\mathrm{ML}}$: update only the counter-hypotheses

- Update those counter-hypotheses for which $d(\mathrm{s}) < \lambda_{j,b}^{\overline{\mathrm{ML}}}$ according to $\lambda_{j,b}^{\overline{\mathrm{ML}}} \leftarrow d(\mathrm{s})$

STS Tree Pruning Criterion

$$\mathcal{A}_i = \{a_l\} = \left\{\lambda_{j,b}^{\overline{\mathrm{ML}}} \,\middle|\, x_{j,b} = \overline{x_{j,b}^{\mathrm{ML}}},\ j \geq i\right\} \cup \left\{\lambda_{j,b}^{\overline{\mathrm{ML}}} \,\middle|\, j < i\right\}$$

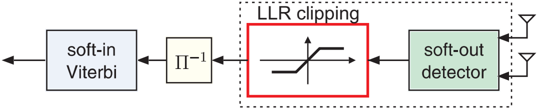
$$d\left(\mathbf{s}^{(i)}\right) > \max_{a_l \in \mathcal{A}_i} a_l$$



M = 5, BPSK modulation
5
4
3
2
1
ML hypothesis
$\boldsymbol{x}^{\mathrm{ML}}$=[0 0 1 1 1]
Current subtree
$\boldsymbol{x}$=[0 1 1 ? ?]
$\neq$
max
Counter hypotheses
$\lambda_{5,1}^{\overline{\mathrm{ML}}}$
$\lambda_{4,1}^{\overline{\mathrm{ML}}}$
$\lambda_{3,1}^{\overline{\mathrm{ML}}}$
$\lambda_{2,1}^{\overline{\mathrm{ML}}}$
$\lambda_{1,1}^{\overline{\mathrm{ML}}}$
ML hypothesis
0
0
1
1
1
$\boldsymbol{x}^{\mathrm{ML}}$ λ^{ML}

LLR Clipping [Yee, 2005; Studer et al., 2006]

Low-complexity implementations of channel decoding with small word lengths require **clipping of LLRs**:

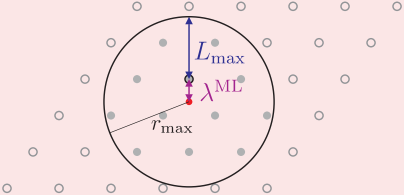


$$\left|\lambda_{j,b}^{\overline{\mathrm{ML}}} - \lambda^{\mathrm{ML}}\right| \leq L_{\max}$$

Clipping leads to a sphere constraint

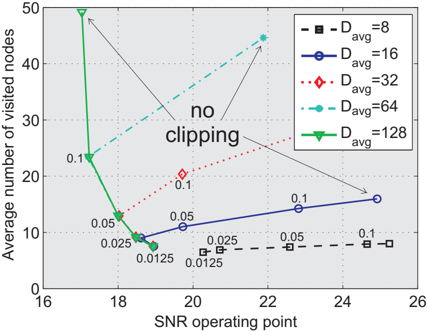
LLR clipping can be built into tree traversal by noting that it induces an upper bound on the search radius according to

$$r_{\max} = L_{\max} + \lambda^{\mathrm{ML}}$$



LLR clipping enables runtime constraints for STS-SD

Consider block ET ($N = 64$) with maximum-first scheduling



$L_{\max}$ too large (no clipping):

- Average runtime above D_{avg}
- Performance limited by ET

$L_{\max}$ too small:

- Performance limited by LLR clipping

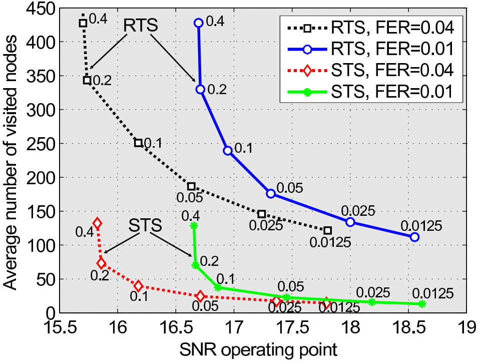
Straightforward application of a runtime constraint to STS-SD degrades performance severely

LLR clipping level $L_{\max}$ must be adjusted to runtime constraint D_{avg}

Comparison of RTS and STS

Adjust clipping level $L_{\max}$ to trade off complexity for performance

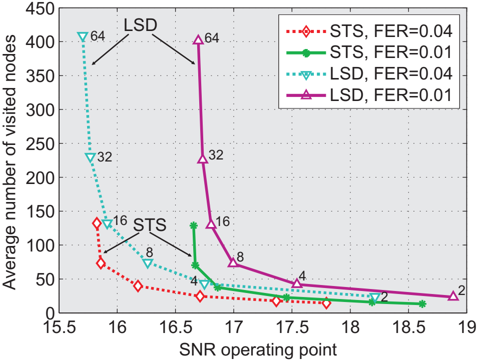
- Complexity: average number of visited nodes
- Performance: SNR required to achieve a given target FER



STS yields factor 3–10 complexity savings over RTS

Comparison of STS and LSD

List size is used to adjust the complexity of the LSD



For a given complexity constraint, STS SD outperforms LSD

Channel Matrix Preprocessing

Sorted QR decomposition (SQRD) [Wübben et al., 2003]

- Perform QR decomposition on reordered channel matrix: $\mathbf{HP} = \mathbf{Q}\,\mathbf{R}$
- Goal: Obtain $\mathbf{R}$ with $|r_{ii}| \geq |r_{jj}|$ for $i > j$
- SQRD preprocessing entails almost no additional complexity

MMSE sorted QR decomposition (MMSE-SQRD)

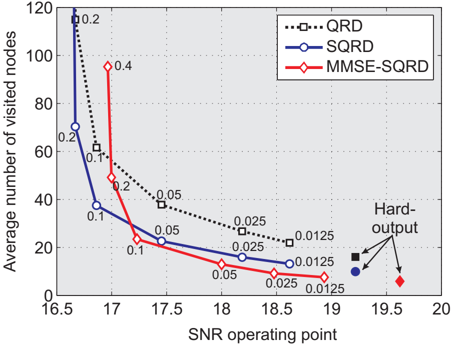
- Perform sorted QR decomposition on a regularized channel matrix

$$\begin{bmatrix} \mathbf{H} \\ \alpha\mathbf{I} \end{bmatrix} \mathbf{P} = \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \end{bmatrix} \mathbf{R}$$

- Compared to QRD/SQRD the complexity of MMSE-SQRD preprocessing increases by roughly 50%

Impact of Channel Matrix Preprocessing

Consider STS with QRD, SQRD, and MMSE-SQRD preprocessing

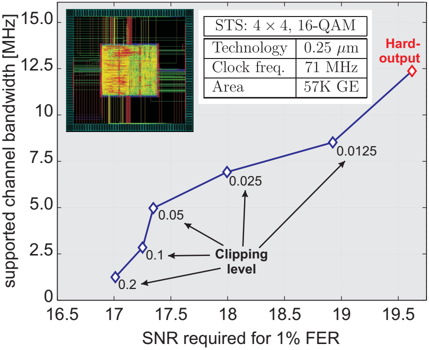


Performance improvement resulting from SQRD and MMSE preprocessing becomes significant under stringent complexity constraints

System bandwidth dictates the arrival rate of symbols at the detector

Decodig effort must be adjusted to keep up with the given arrival rate

- *LLR clipping* enables to adjust the average complexity (throughput)
- Run-time constraints enforce latency constraints and a guaranteed instantaneous throughput



STS: 4×4, 16-QAM	
Technology	$0.25\ \mu$m
Clock freq.	71 MHz
Area	57K GE

$$D_{\text{avrg}} = \frac{f_{\text{STS}}}{B}$$

B	...	Bandwidth
f_{STS}	...	STS frequency

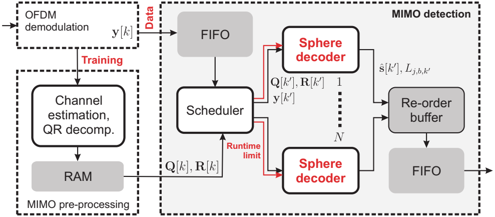
For a given detector implementation, the performance (SNR required for a given FER) depends on the bandwidth of the system

Meeting throughput requirements for any desired performance target

Instantiation of **multiple parallel STS-SD units** allows to **increase** the maximum **decoding effort** available per symbol vector:

$$D_{\text{avrg}} = \frac{N f_{\text{STS}}}{B}$$

B	...	Bandwidth
N	...	STS instances
f_{STS}	...	STS frequency



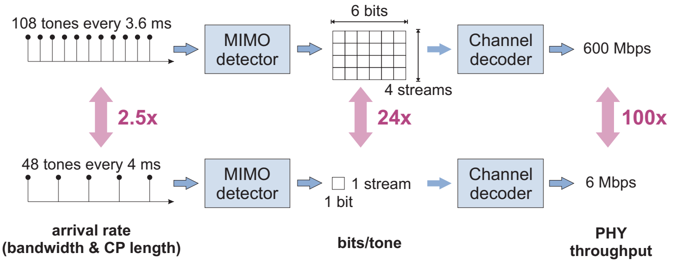
- A scheduler distributes symbols across detector instances
- Due to variable runtimes in the STS-SD instances, symbols must be reordered when they are collected at the output of the detectors

Better performance can easily be achieved at the cost of more silicon area and power

Sphere decoding suffers from variable complexity

Search effort varies as a function of noise and channel realization. The worst-case complexity corresponds to that of an exhaustive search

Complexity still grows exponentially with the spectral efficiency and linearly in the bandwidth

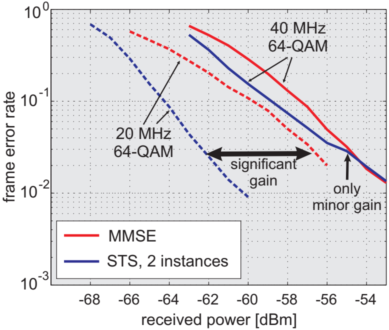


- Data rates specified in IEEE 802.11n vary by 2 orders of magnitude
- Truly optimum decoding remains prohibitive for highest-rate modes

STS-SD performance advantage depends on the transmission mode

Complexity of STS-SD algorithm can be adjusted at run-time

- LLR clipping ensures a graceful performance degradation when processing requirements increase

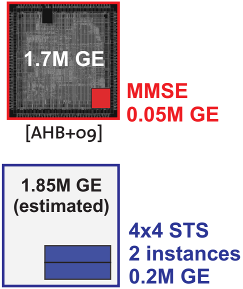


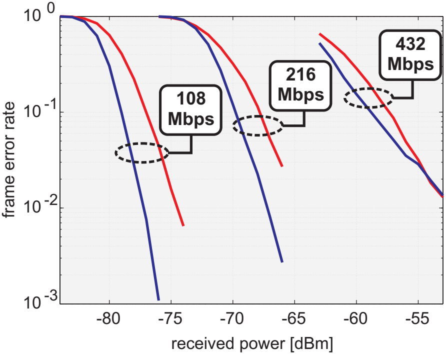
For a given implementation STS-SD leverages the maximum performance gain in each transmission mode

Application to 4-stream IEEE 802.11n

IEEE 802.11n uses 4-spatial streams with $B = 40\,\text{MHz}$ bandwidth:

- Single STS-SD unit [WBB+10]: 100k GE @ $f_{\text{STS}} = 320\,\text{MHz}$ in 0.13 μm
- Instantiation of **two STS-SD units**





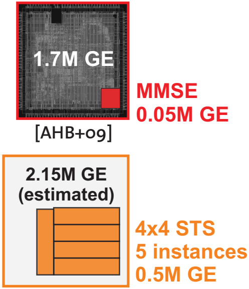
[AHB+09] A. Burg, S. Haene, M. Borgmann, D. Baum, T. Thaler, F. Carbognani, S. Zwicky, L. Barbero, C. Senning, P. Greisen, T. Peter, C. Foelmli, U. Schuster, P. Tejera, A. Staudacher, "A 4-Stream 802.11n Baseband Transceiver in 0.13um CMOS," *Proc. of the Symposium on VLSI Circuits*, Jun. 2009.

[WBB+10] M. Wenk, L. Bruderer, A. Burg, C. Studer, "Area- and Throughput-Optimized VLSI Architecture of Sphere Decoding," *Proc. of the VLSI-SoC Conf.*, Sep. 2010.

Application to 4-stream IEEE 802.11n

IEEE 802.11n uses 4-spatial streams with $B = 40\,\text{MHz}$ bandwidth:

- Single STS unit [WBB+10] : 100K GE @ $f_{\text{STS}} = 320\,\text{MHz}$ in 0.13 μm
- Instantiation of **5 STS units**



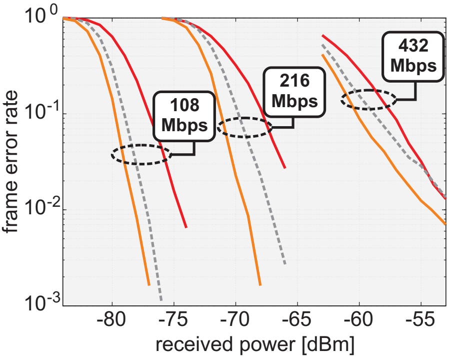
1.7M GE

MMSE
0.05M GE

[AHB+o9]

2.15M GE
(estimated)

4x4 STS
5 instances
0.5M GE

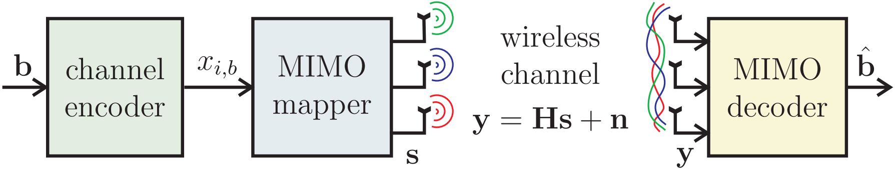


[AHB+09] A. Burg, S. Haene, M. Borgmann, D. Baum, T. Thaler, F. Carbognani, S. Zwicky, L. Barbero, C. Senning, P. Greisen, T. Peter, C. Foelmli, U. Schuster, P. Tejera, A. Staudacher, “A 4-Stream 802.11n Baseband Transceiver in 0.13um CMOS,” *Proc. of the Symposium on VLSI Circuits*, Jun. 2009.

[WBB+10] M. Wenk, L. Bruderer, A. Burg, C. Studer, “Area- and Throughput-Optimized VLSI Architecture of Sphere Decoding,” *Proc. of the VLSI-SoC Conf.*, Sep. 2010.

The Last Frontier: Iterative MIMO Detection and Decoding (IDD)

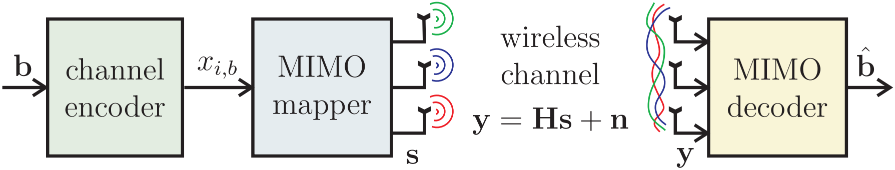
Approaching the performance limits



Optimum detection of MIMO-BICM requires joint ML estimation (joint detection and decoding) of an entire codeword

- Only large code blocks can get close to capacity
- Complexity of ML grows exponentially in the codeblock size

Approaching the performance limits

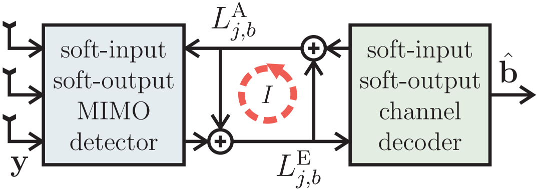


Optimum detection of MIMO-BICM requires joint ML estimation (joint detection and decoding) of an entire codeword

- Only large code blocks can get close to capacity
- Complexity of ML grows exponentially in the codeblock size

Joint MIMO detection and channel decoding is **infeasible** in practice

Iterative MIMO detection and decoding (IDD) [HB03] to the rescue



- Separate detection from channel decoding
- Exchange reliability (soft) information

- **Soft-in soft-out (SISO) MIMO detector:** Compute *extrinsic* LLRs

$$L^{\mathrm{E}}_{j,b} = \log\left(\frac{\mathrm{P}[x_{i,b} = 1 \mid \mathbf{y}, \mathbf{H}]}{\mathrm{P}[x_{j,b} = 0 \mid \mathbf{y}, \mathbf{H}]}\right) - L^{\mathrm{A}}_{j,b}$$

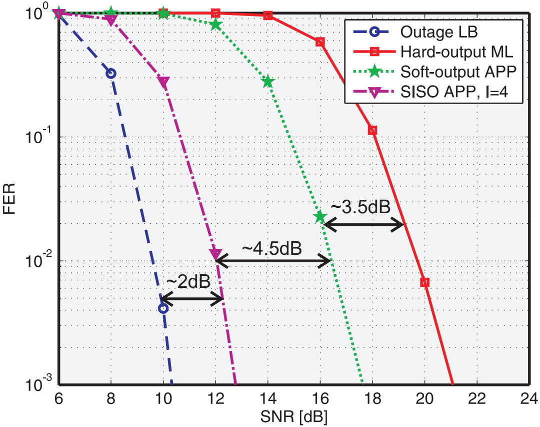
  based on $\mathbf{y}$, $\mathbf{H}$, and the a priori LLRs $L^{\mathrm{A}}_{j,b} = \log\left(\frac{\mathrm{P}[x_{j,b}=1]}{\mathrm{P}[x_{j,b}=0]}\right)$.

- **SISO channel decoder** (e.g., LDPC, BCJR): Computes new a-priori LLRs $L^{\mathrm{A}}_{j,b}$ based on the LLRs $L^{\mathrm{E}}_{j,b}$ and estimates of the transmitted bits $\hat{\mathbf{b}}$

[HB03] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, Mar. 2003

Near-optimum performance through MIMO IDD

Performance gains through IDD

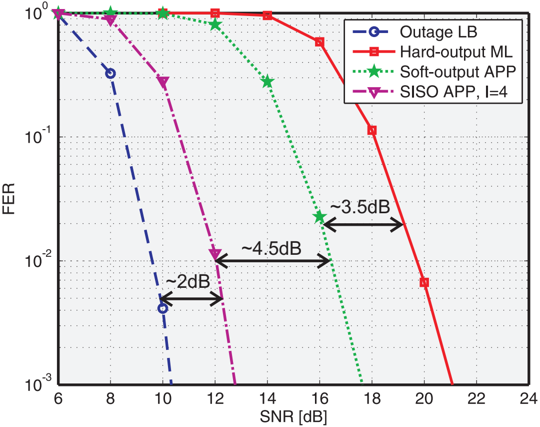


- MIMO-OFDM system, 16-QAM, 4×4, convolutional code, TGn C channel

- Fundamental performance limit: **outage lower bound**

Near-optimum performance through MIMO IDD

Performance gains through IDD



- MIMO-OFDM system, 16-QAM, 4×4, convolutional code, TGn C channel
- Fundamental performance limit: **outage lower bound**

Iterative MIMO detection and decoding is able to **approach the outage lower bound** on frame error rate

The Last Frontier: Iterative MIMO Detection and Decoding (IDD)

Soft-input Soft-output MIMO Detection

IDD receiver components under investigation

Candidate soft-input soft-output MIMO detection algorithms

- Soft-input soft-output MAP detection with max-log approximation
  - Low complexity algorithm based on sphere decoding [SB10]
  - Considerable, but manageable computational complexity
- SISO MMSE parallel interference cancellation (PIC) [WP99]
  - Algorithm is an extension of linear MMSE detection
  - Computational complexity is lower than that of SISO MAP

Channel code / decoder

- Quasi-cyclic low density parity check (LDPC) code
- Layered message passing with offset-min-sum approximation

[SB10] C. Studer and H. Bölcskei, "Soft-input soft-output single tree-search sphere decoding," *IEEE Trans. Inf. Theory*, Nov. 2010
[WP99] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, July 1999

SISO MAP detection can be performed with sphere decoding

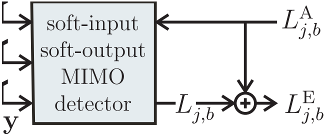
First iteration: conventional soft-output STS-SD metric $\mathcal{M}_{\mathrm{C}}$

Subsequent iterations $(I \geq 2)$:

Max-log-approximated LLRs:

$$L_{i,b} = \pm\left(\lambda^{\mathrm{MAP}} - \lambda_{i,b}^{\overline{\mathrm{MAP}}}\right),$$

where the sign depends on $x_{i,b}^{\mathrm{MAP}}$



- Metric of the maximum a posteriori (MAP) solution

$$\lambda^{\mathrm{MAP}} = \min_{\mathbf{s}\in\mathcal{O}^{M_T}} \left\{ \frac{1}{N_o}\|\mathbf{y}-\mathbf{Hs}\|^2 - \log \mathrm{P}[\mathbf{s}] \right\}$$

- Metric associated with counter-hypothesis

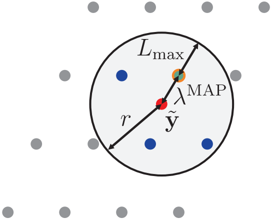
$$\lambda_{i,b}^{\overline{\mathrm{MAP}}} = \min_{\mathbf{s}\in\mathcal{X}_{i,b}^{\overline{\mathrm{MAP}}}} \left\{ \frac{1}{N_o}\|\mathbf{y}-\mathbf{Hs}\|^2 - \log \mathrm{P}[\mathbf{s}] \right\}$$

- $\mathrm{P}[\mathbf{s}] = \prod_{i=1}^{M_T} \mathrm{P}[s_i]$, where $\mathrm{P}[s_i]$ is derived from a-priori feedback $^{\mathrm{A}}_{i,b}$

SISO-STS SD with LLR clipping [SB10]

Reminder: Limiting the magnitude of LLRs can inprove tree-pruning efficiency

- Clipping $L_{i,b}$ to $L_{\max}$ can cause errors on $L^{\mathrm{E}}_{i,b} = [L_{i,b}]_{L_{\max}} - L^{\mathrm{A}}_{i,b}$, e.g., if $L^{\mathrm{A}}_{i,b} > L_{\max}$
- **LLR clipping needs to be performed on $L^{\mathrm{E}}_{i,b} < L_{\max}$ instead**
- Pruning criterion: $r^2 = \lambda^{\mathrm{MAP}} + L_{\max}$

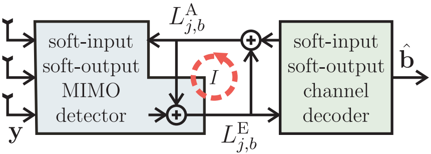


SISO STS-SD computes extrinsic information directly

$$L^{\mathrm{E}}_{i,b} = \pm\left(\lambda^{\mathrm{MAP}} - \underbrace{(\lambda^{\overline{\mathrm{MAP}}}_{i,b} \pm L^{\mathrm{A}}_{i,b})}_{=\Lambda^{\overline{\mathrm{MAP}}}_{i,b}}\right),$$

where sign depends on $x^{\mathrm{MAP}}_{i,b}$

- Tree search based on $\Lambda^{\overline{\mathrm{MAP}}}_{i,b}$



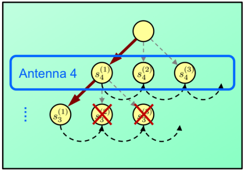
[SB10] C. Studer and H. Bölcskei, "Soft-input soft-output single tree-search sphere decoding," *IEEE Trans. Inf. Theory*, Nov. 2010

Implementation challenge lies in efficient enumeration

SISO STS-SD cost metric $\mathcal{M}_\mathrm{C} = \mathcal{M}_\mathrm{C} + \mathcal{M}_\mathrm{A}$

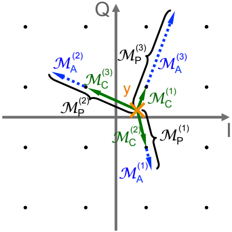
- **Euclidean distance metric** $\mathcal{M}_\mathrm{C}$
- **A-priori metric** $\mathcal{M}_\mathrm{A}$

- Conventional STS-SD: efficient enumeration based on geometrical properties of $\mathcal{M}_\mathrm{C}$
- SISO STS-SD: geometrical properties destroyed by a priori information $\mathcal{M}_\mathrm{A}$



Alternative solution: **Hybrid Enumeration** [WB+10]

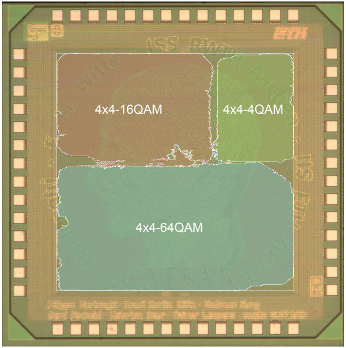
1. Enumerate concurrently based on $\mathcal{M}_\mathrm{C}$ and $\mathcal{M}_\mathrm{A}$
2. Select symbol with minimum $\mathcal{M}_\mathrm{P}$


Q
I
y
$\mathcal{M}_\mathrm{P}^{(3)}$
$\mathcal{M}_\mathrm{A}^{(3)}$
$\mathcal{M}_\mathrm{A}^{(2)}$
$\mathcal{M}_\mathrm{C}^{(3)}$
$\mathcal{M}_\mathrm{C}^{(1)}$
$\mathcal{M}_\mathrm{P}^{(2)}$
$\mathcal{M}_\mathrm{C}^{(2)}$
$\mathcal{M}_\mathrm{P}^{(1)}$
$\mathcal{M}_\mathrm{A}^{(1)}$

[WB+10] E. M. Witte, F. Borlenghi, G. Ascheid, R. Leupers, and H. Meyr, “A scalable VLSI architecture for soft-input soft-output single tree-search sphere decoding,” *IEEE Trans. Circuits and Systems II*, Nov. 2010

ASIC implementation of SISO STS-SD

SISO STS-SD by Witte, Borlenghi, et al., RWTH Aachen [BW+11]



Three configurable cores

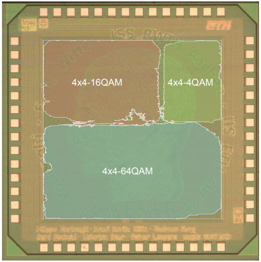
- Technology: 90 nm CMOS
- All support 4-streams
- 4-QAM: 0.18 mm^2, $330{\cdot}10^6$ visited nodes/s
- 16-QAM: 0.354 mm^2, $244{\cdot}10^6$ visited nodes/s
- 64-QAM: 0.665 mm^2, $193{\cdot}10^6$ visited nodes/s

[BW+11] F. Borlenghi, E. M. Witte, G. Ascheid, H. Meyr, and A. Burg, "A 772 Mbit/s 8.81 bit/nJ 90 nm CMOS Soft-Input Soft-Output Sphere Decoder," *IEEE Asian Solid State Circuits Conf. 2011*

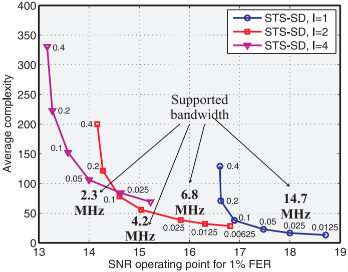
FER performance at the expense of supported bandwidth per instance

SISO STS-SD by Witte, Borlenghi, et al., RWTH Aachen [BW+11]

- Single instance ASIC (16-QAM): $f_{\text{node}} = 244{\cdot}10^6$ visited nodes/s



- IDD computational effort D_{avrg} per vector symbol for 4×4, 16-QAM

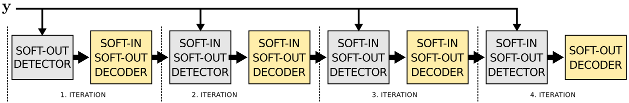


- Supported bandwidth: $BW_{\max} = f_{\text{node}} / D_{\text{avrg}}$

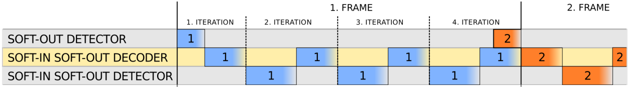
Computational effort increases dramatically for 64-QAM

IDD architecture: putting the pieces together

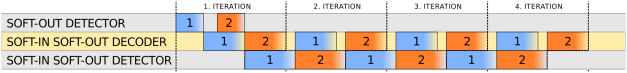
Conventional detection and decoding **schedule**



- **Serial** architecture
  MIMO detector and channel decoder take turns processing one block



- **Ping-pong** architecture
  MIMO detector and channel decoder process two blocks interleaved



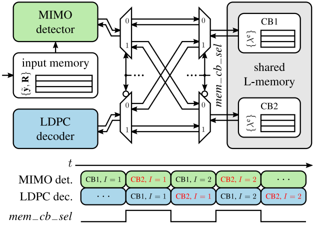
  - **Throughput** and hardware utilization **increase by 2x**
  - Processing **latency** remains **unaltered**

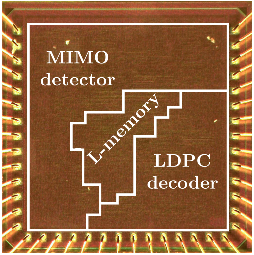
Complete IDD ASIC (RWTH/EPFL)[BWA+12]

- Ping-pong architecture
- 5 parallel SISO-STS cores served by a scheduler
- Layered OMS LDPC for 802.11n QC-LDPC

Performance characteristics:

- Technology: 65nm CMOS, 1.2V
- Area: 1.58M GE (5 SISO-STS: 872k GE, LDPC: 447k GE)
- Clock freq.: 135 MHz/299 MHz (SISO-STS/LDPC)
- Supports up to $\times 4$ with 64-QAM
- 4×4, 64-QAM $R = 1/2$ with 1% BLER
  - @ 18 dB: 33 Mbps
  - @ 28 dB: 1250 Mbps





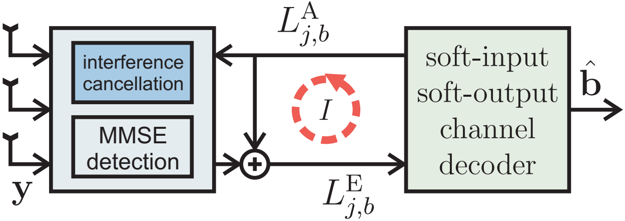
[BWA+12] F. Borlenghi, E.M.Witte, G. Ascheid, H. Meyr, and A. Burg, "A 2.78 mm2 65 nm CMOS Gigabit MIMO Iterative Detection and Decoding Receiver," ESSCIRC 2012, to appear

IDD with the MMSE-PIC algorithm

First iteration (initialization): conventional soft-output MMSE detection

Subsequent iterations ($I \geq 2$)

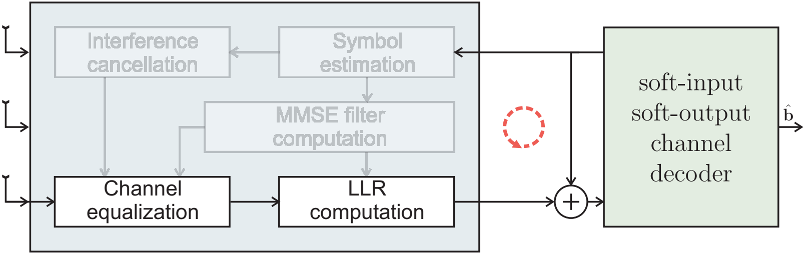
- Interference cancellation (IC) based on feedback from channel decoder
- Re-detection with modified (feedback dependent) MMSE filter



- IC step requires *intrinsic*, instead of *extrinsic* feedback [WBS+02]
- MMSE filter needs to be recomputed for each symbol and each iteration

[WBS+02] M. Witzke, S. Bäro, F. Schreckenbach, and J. Hagenauer, "Iterative detection of MIMO signals with linear detectors," *Proc. of the IEEE Asilomar Conf.*, Nov. 2002

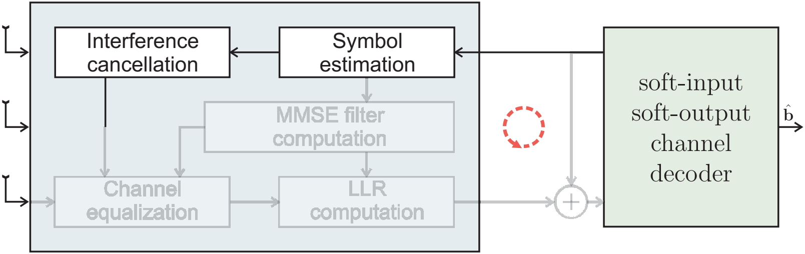
The MMSE-PIC algorithm



Step 1: Initialization (1st iteration, $I = 1$)

- Compute the linear MMSE filter vectors $\mathbf{w}_j^H = \mathbf{h}_j^H \left(\mathbf{H}\mathbf{H}^H + N_0\mathbf{I}\right)^{-1}$
- Unbiased MMSE estimation: $\hat{z}_j = \mu_j^{-1}\mathbf{w}_j^H\hat{\mathbf{y}}_j$ where $\mu_j = \mathbf{w}_j^H\mathbf{h}_j, \forall j$
- Compute LLRs $L_{j,b}^{\mathrm{E}}$ from $\hat{z}_j$ and from the per-stream SINRs ρ_j
- Soft-output channel decoding yields *intrinsic* LLRs $L_{j,b}^{\mathrm{A}}$

The MMSE-PIC algorithm



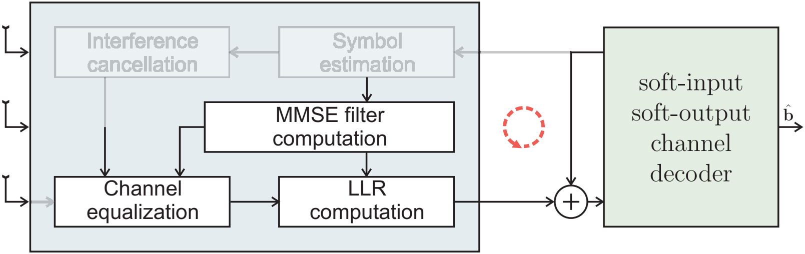
Step 2: Parallel interference cancellation (PIC)

- Compute soft-symbols $\hat{s}_j = \mathbb{E}[s_j]$, $\forall j$, based on a priori LLRs $L^{\mathrm{A}}_{j,b}$
- Perform interference cancellation

$$\hat{\mathbf{y}}_j = \mathbf{y} - \sum_{i \neq j} \mathbf{h}_i \hat{s}_i = \mathbf{h}_j s_j + \underbrace{\sum_{i \neq j} \mathbf{h}_i e_i + \mathbf{n}}_{\text{interference}-\text{plus}-\text{noise}} \ , \ j = 1, \ldots, M_T$$

- Variance of the residual interference terms e_j: $E_j = \mathbb{E}\left[\left| s_j - \hat{s}_j \right|^2 \right]$, $\forall j$

The MMSE-PIC algorithm



Step 3: Re-detection after IC

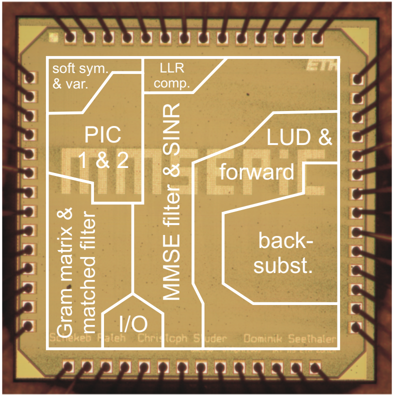
- Re-compute the linear MMSE filter vectors

$$\mathbf{w}_j^H = \mathbf{h}^H \left(\mathbf{H}\boldsymbol{\Lambda}_j\mathbf{H}^H + N_o\mathbf{I}_{M_R}\right)^{-1} \text{ with}$$
$$\boldsymbol{\Lambda}_j = \mathrm{diag}(E_1, \ldots, 1/M_T, \ldots, E_{M_T})$$

- Unbiased MMSE estimation: $\hat{z}_j = \mu_j^{-1}\mathbf{w}_j^H\hat{\mathbf{y}}_j$ where $\mu_j = \mathbf{w}_j^H\mathbf{h}_j, \forall j$
- Compute LLRs $L_{j,b}^{\mathrm{E}}$ from $\hat{z}_j$ and from the per-stream SINRs ρ_j
- Soft-output channel decoding yields *intrinsic* LLRs $L_{j,b}^{\mathrm{A}}$

ASIC implementation of SISO MMSE-PIC

MMSE-PIC ASIC implementation by Studer et al., ETH Zurich [SFB10]



- Support for 4-streams
- Modulation: BPSK to 64-QAM
- Technology: 90 nm CMOS
- Silicon area: 1.5 mm^2
- Supported bandwidth for 2 iterations: 16 MHz up to 64-QAM

[SFB10] C. Studer, S. Fateh, and D. Seethaler, “A 757Mb/s 1.5mm^2 90nm CMOS soft-input soft-output MIMO detector for IEEE 802.11n,” in Proc. of IEEE European Solid State Circuits Conf. (ESSCIRC), Sept. 2010

Scenarios for performance comparison

Consider two very different communication scenarios

Fast-fading: Channel changes rapidly from symbol to symbol

- Coding across independent channel realizations
- OFDM: highly frequency-selective channel

$$\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N] \quad \mathbf{y}_t = \mathbf{H}_t \mathbf{s}_t + \mathbf{n}_t \quad \mathbf{H}_l \neq \mathbf{H}_{k \neq l}$$

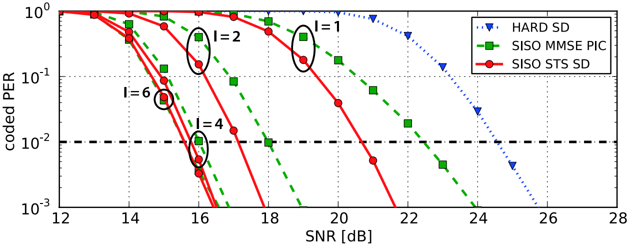
Block-fading: Channel remains constant for all symbols in a code block

- Coding across only one channel realization
- OFDM: highly frequency-flat channel

$$\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N] \quad \mathbf{y}_t = \mathbf{H} \mathbf{s}_t + \mathbf{n}_t$$

Performance comparison with fast-fading

Coding across independent channel realizations provides significant non-spatial diversity

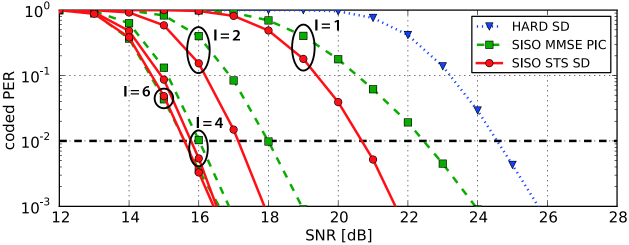


- Without iterations ($I = 1$), STS-SD clearly outperforms the MMSE
- For IDD ($I \geq 2$), SISO STS-SD has a small advantage over MMSE-PIC
- Diminishing returns from SISO STS-SD with increasing number of iteration

[BW+11] F. Borlenghi, E. M. Witte, G. Ascheid, H. Meyr, and A. Burg, "A 772 Mbit/s 8.81 bit/nJ 90 nm CMOS Soft-Input Soft-Output Sphere Decoder," *IEEE Asian Solid State Circuits Conf. 2011*, submitted to

Performance comparison with fast-fading

Coding across independent channel realizations provides significant non-spatial diversity

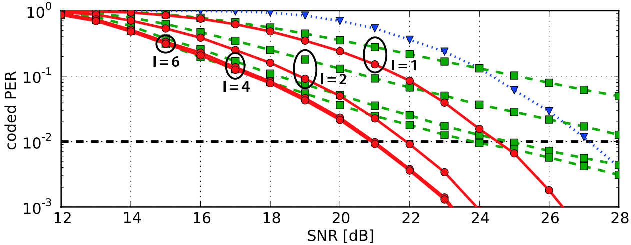


For fast-fading channels, MMSE-PIC is a good alternative for the more complex SISO STS-SD

[BW+11] F. Borlenghi, E. M. Witte, G. Ascheid, H. Meyr, and A. Burg, “A 772 Mbit/s 8.81 bit/nJ 90 nm CMOS Soft-Input Soft-Output Sphere Decoder,” *IEEE Asian Solid State Circuits Conf. 2011*, submitted to

Performance comparison with block-fading

Coding across only one channel realization provides no additional diversity (only spatial diversity)

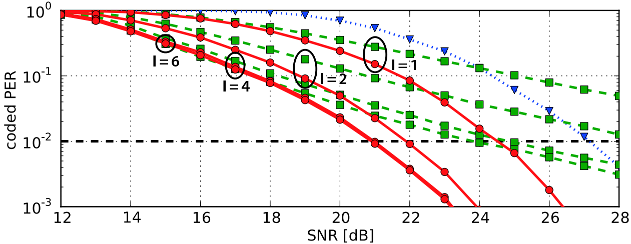


- MMSE-PIC suffers from a lack of diversity (for any number of iterations)
- SISO STS-SD outperforms MMSE-PIC even with much fewer iterations

[BW+11] F. Borlenghi, E. M. Witte, G. Ascheid, H. Meyr, and A. Burg, “A 772 Mbit/s 8.81 bit/nJ 90 nm CMOS Soft-Input Soft-Output Sphere Decoder,” *IEEE Asian Solid State Circuits Conf. 2011*, submitted to

Performance comparison with block-fading

Coding across only one channel realization provides no additional diversity (only spatial diversity)



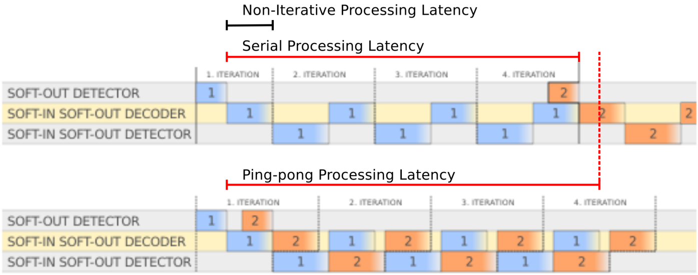
- MMSE-PIC fails if only spatial diversity is available
- A similar behavior is observed for code rates close to 1 (e.g., $R = 5/6$)

[BW+11] F. Borlenghi, E. M. Witte, G. Ascheid, H. Meyr, and A. Burg, "A 772 Mbit/s 8.81 bit/nJ 90 nm CMOS Soft-Input Soft-Output Sphere Decoder," *IEEE Asian Solid State Circuits Conf. 2011*, submitted to

System Integration: Layered Detection and Decoding

System integration of IDD is difficult

IDD entails low throughput and **its latency** grows with the number of iterations. But processing latency is crucial for system with tight round-trip time requirements.

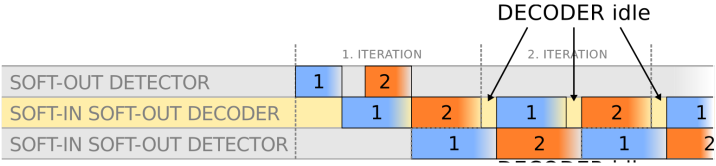


Techniques to increase the throughput (e.g. pipelining, interleaving, multiple instances) usually do not improve the latency.

IDD receivers can not achieve the full iterative gain in latency constrained environments.

Efficient system integration of IDD is even more difficult

100% hardware utilization requires matching the runtime of detection and decoding block. This poses already a significant constraint on the design.



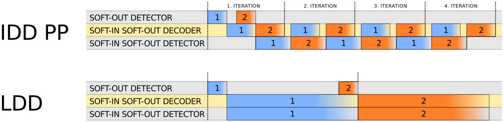
Modern communication standards usually define a wide variety of modes. Corresponding run-times of detector and decoder are highly variable and change with

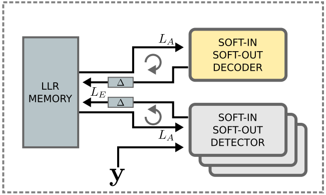
- MIMO detector: modulation, SNR, antenna-configuration
- Channel decoder: code rate, block size, SNR

Optimal hardware implementations of IDD across all modes is not feasible.

Layered detection and decoding (LDD) [PSB12] schedule

The **same block of data** is detected and decoded at the **same time**.





LDD gets rid of the run-time matching constraint by removing the block-wise data dependency between detection and decoding.
The decoder performs one continous decoding run, while updates from the detector are injected.

[PSB12] N. Preyss, C. Studer ,and A. Burg, “Layered Detection and Decoding in MIMO Wireless Systems” in *2012 Conference on Design and Architectures for Signal and Image Processing (DASIP)*, submitted to

Comparing the performance of different iterative receiver schedules

Performance metrics

- Hardware efficiency: (mm^2) per Mbps
- System performance: SNR required to achieve 1% packet-error rate (PER)

Silicon area and throughput estimnates are obtained based on

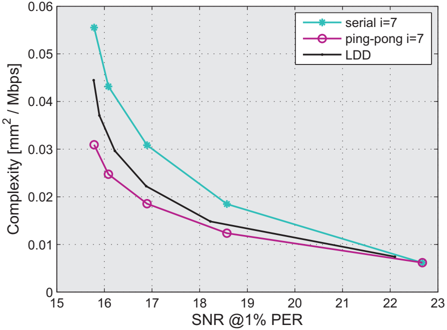
- the MMSE-PIC implementation in [SFB10]
- the LDPC decoder implementation in [RMS+10]

Complexity/performance tradeoff is adjusted by changing the number of detector instances

[SFB10] C. Studer, S. Fateh, and D. Seethaler, "A 757Mb/s 1.5mm^2 90nm CMOS soft-input soft-output MIMO detector for IEEE 802.11n," in Proc. of IEEE European Solid State Circuits Conf. (ESSCIRC), Sept. 2010
[RMS+10] C. Roth, P. Meinerzhagen, C. Studer, and A. Burg, "A 15.8 pJ/bit/iter quasi-cyclic LDPC decoder for IEEE 802.11n in 90 nm CMOS," in Proc. IEEE Asian Solid-State Circuits Conf., Nov. 2010, pp. 313–316.

LDD efficiency is comparable to the one of conventional IDD

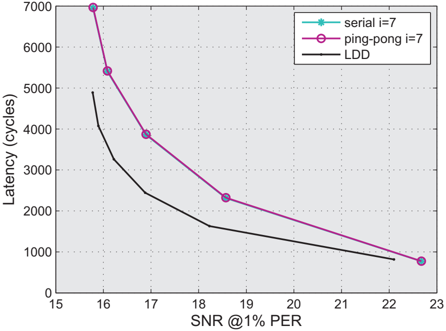
Do we achieve the same performance as with the conventional IDD schedule?



LDD is almost as efficient as the conventional schedule, but does not require run-time matching between detector and decoder

LDD significantly outperforms conventional IDD under latency constraint

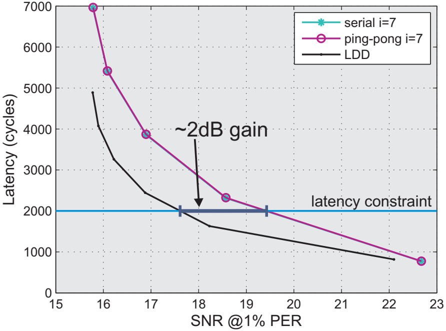
Parallelization of the detection and decoding operations cuts the overall processing latency almost into half.



For a given latency constraint, LDD provides a significant performance (SNR) advantage over IDD with ping-pong schedule

LDD significantly outperforms conventional IDD under latency constraint

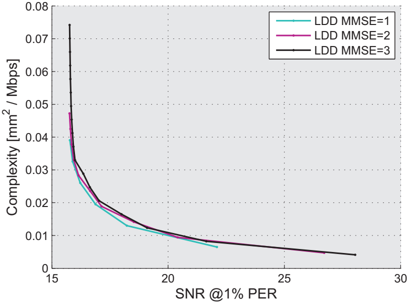
Parallelization of the detection and decoding operations cuts the overall processing latency almost into half.

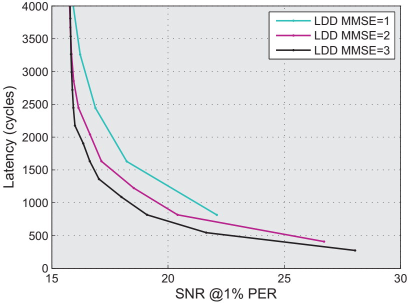


For a given latency constraint, LDD provides a significant performance (SNR) advantage over IDD with ping-pong schedule

LDD is highly scalable in all performance metrics

LDD allows to increase the MIMO detection effort by instantiating multiple detector units without sacrificing efficiency.

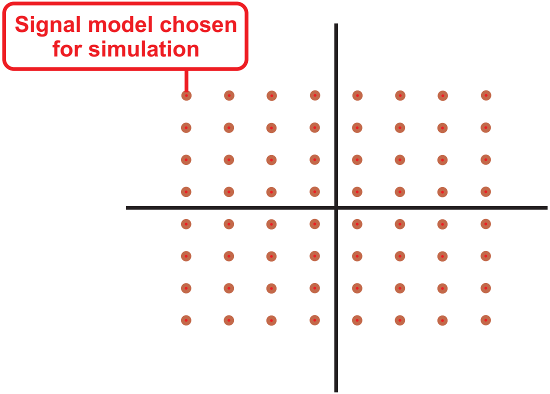




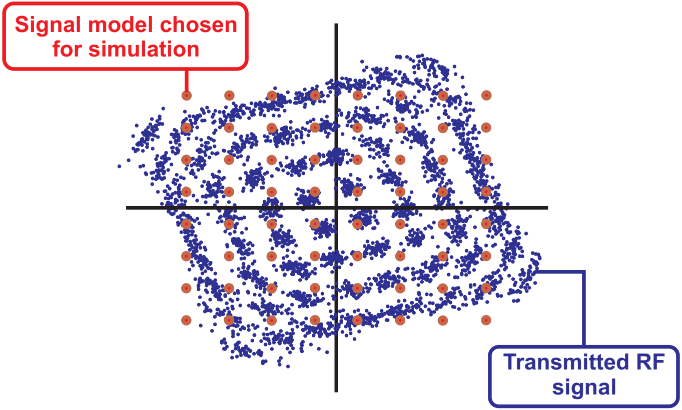
Additional complexity offers not only more throughput but also less latency.

MIMO with Transmit RF Impairments

System model used for simulation is based on highly idealistic assumptions

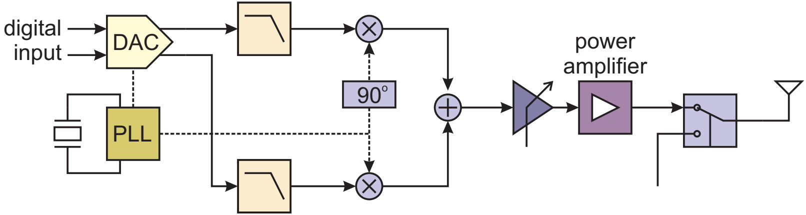


System model used for simulation is based on highly idealistic assumptions


Signal model chosen for simulation

Transmitted RF signal

Real-world RF transmitters suffer from a variety of impairments



- Quantization noise from baseband processing and from the DAC
- Sampling- and carrier-frequency offset
- Phase noise from the PLL
- I/Q imbalance in mixers and analog filters
- Non-linear behavior of the power amplifier
- MIMO: Crosstalk/coupling between transmit-RF chains

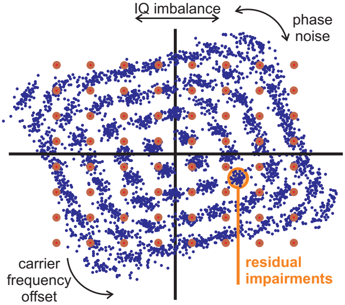
Distinguish between two types of impairments

Some impairments are well understood and their **impact can partially be mitigated** or avoided (e.g., through calibration)

Residual Tx-RF impairments

defy proper compensation since they ...

- ... are not well understood or difficult to model
- ... based on random processes with often unknown statistics
- ... require prohibitively complex compensation algorithms



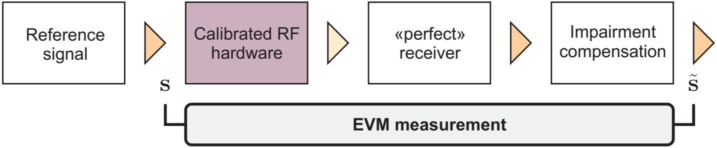
Characterizing residual Tx-RF impairments

The **error vector magnitude** (EVM) is a lump-sum **measure of the residual distortions** at the transmitter:

$$\text{EVM} = \frac{\mathbb{E}[\|\tilde{\mathbf{s}} - \mathbf{s}\|^2]}{\mathbb{E}[\|\mathbf{s}\|^2]}$$

$\mathbf{s}$ ... Reference transmitted signal

$\tilde{\mathbf{s}}$ ... Measured transmitted signal *after impairment compensation*



Typical EVM-values range from -22 dB to -32 dB

Model for transmitter with residual Tx-RF impairments

Assumptions:

- Gaussian distributed
- Independent from the transmitted signals
- I.i.d. among I- and Q-path
- I.i.d. across transmit antennas (MIMO)

Model for transmitter with residual Tx-RF impairments

Assumptions:

- Gaussian distributed
- Independent from the transmitted signals
- I.i.d. among I- and Q-path
- I.i.d. across transmit antennas (MIMO)

Model for a non-ideal transmitter

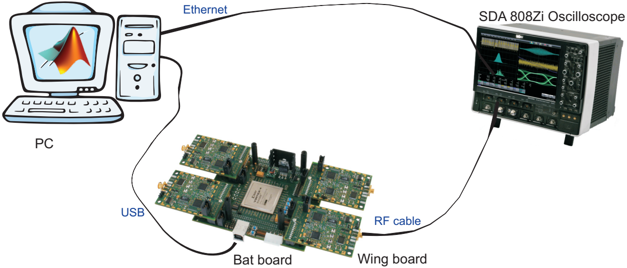
$$\tilde{\mathbf{s}} = \mathbf{s} + \mathbf{n}_\mathrm{t}$$

- $\mathbf{n}_\mathrm{t} \sim \mathcal{CN}(0, \sigma_\mathrm{t}^2 \mathbf{I}_{M_T})$ the **transmit-noise** (Tx-noise)
- Tx-noise variance: $\sigma_\mathrm{t}^2 = \mathrm{EVM}^{-1}$

Verification of the Tx-noise model for MIMO-OFDM transmission

Measurements are performed to verify the model assumptions:

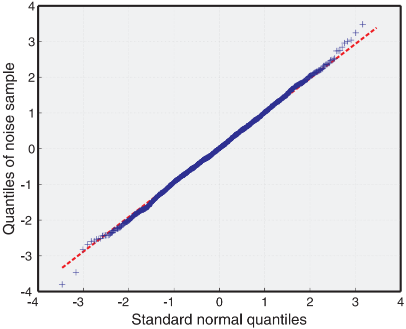
- 20 MHz reference signal uses OFDM modulation
- MIMO RF-chain [WLK+09] based on commercial single-stream RF chip-sets



[WLK+09] M. Wenk, P. Luethi, T. Koch, P. Maechler, M. Lerjen, N. Felber, and W. Fichtner, "Hardware platform and implementation of a real-time multi-user MIMO-OFDM testbed," in *Proc. of the Int. Symp. on Circuits and Systems (ISCAS)*, May 2009, pp. 789–792.

Measurements confirm model assumptions for MIMO-OFDM

- **Gaussian distribution:** Quantile-quantile plot confirms at least a Gaussian-like distribution



- **Circular symmetry:** Noise-correlation between I- and Q-path is negligible *after calibration*
- **Spatially white:** Tx-noise correlation across antennas is 19.5 dB below the Tx-noise level

MIMO system model with transmit noise

Baseband input-output relation with transmit noise:

$$\mathbf{y} = \mathbf{H}(\mathbf{s} + \mathbf{n}_t) + \mathbf{n} = \mathbf{H}\mathbf{s} + \underbrace{\mathbf{H}\mathbf{n}_t + \mathbf{n}}_{\mathbf{z}}$$

Noise covariance matrix:

$$\mathbf{K} = \mathbb{E}\left[\mathbf{z}\mathbf{z}^H\right] = \sigma_{\mathrm{t}}^2 \mathbf{H}\mathbf{H}^H + \sigma_{\mathrm{r}}^2 \mathbf{I}_{M_R}$$

Tx-noise appears spatially colored at the receiver

Equivalent received-signal model

Since $\mathbf{n}_t$ and $\mathbf{n}$ are independent and both i.i.d. Gaussian distributed we can write

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{K}^{\frac{1}{2}}\mathbf{w} \quad \text{with} \quad \mathbf{w} \sim \mathcal{CN}(0, \mathbf{I}_{M_R})$$

Impact on Channel Capacity

Impact of Tx-noise on channel capacity

MIMO channel capacity [T99] is well known

$$C(\mathbf{H}) = \log_2 \det\left(\mathbf{I}_{M_R} + \frac{1}{\sigma_{\mathrm{r}}^2}\mathbf{H}\mathbf{H}^H\right)$$

- Application of the Eigenvalue decomposition $\mathbf{H}\mathbf{H}^H = \mathbf{U}\tilde{}\mathbf{U}^H$ with $\mathbf{U}^H\mathbf{U} = \mathbf{I}_{M_T}$ and $\tilde{} = \mathrm{diag}(\lambda_1, \ldots, \lambda_{M_T})$

Channel capacity with Tx-RF impairments

$$C(\mathbf{H}) = \sum_{i=1}^{M_T} C_i, \quad C_i = \log_2\left(1 + \frac{\lambda_i}{\sigma_{\mathrm{r}}^2}\right).$$

[T99] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Trans. Telecomm.*, vol. 10, no. 6, pp. 585-596, Sep. 1999.

Impact of Tx-noise on channel capacity

MIMO channel capacity [T99] with spatially colored noise is well known

$$C(\mathbf{H}) = \log_2 \det\left(\mathbf{I}_{M_R} + \breve{}^{-1}\mathbf{H}\mathbf{H}^H\right)$$

$\breve{}$ ... noise covariance matrix

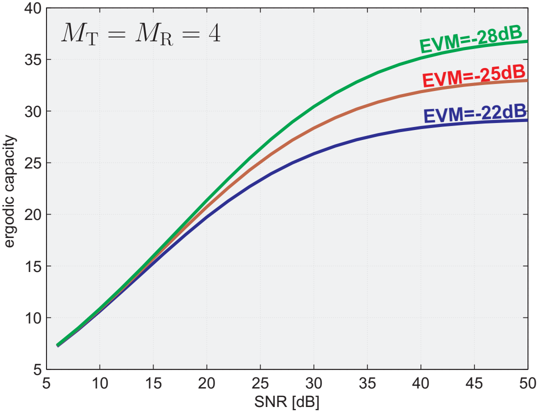
- In our specific case : $\breve{} = \mathbf{K} = \sigma_{\mathrm{t}}^2 \mathbf{H}\mathbf{H}^H + \sigma_{\mathrm{r}}^2 \mathbf{I}_{M_R}$
- Application of the Eigenvalue decomposition $\mathbf{H}\mathbf{H}^H = \mathbf{U}\tilde{}\mathbf{U}^H$ with $\mathbf{U}^H\mathbf{U} = \mathbf{I}_{M_T}$ and $\tilde{} = \mathrm{diag}(\lambda_1, \ldots, \lambda_{M_T})$

Channel capacity with Tx-RF impairments

$$C(\mathbf{H}) = \sum_{i=1}^{M_T} C_i, \quad C_i = \log_2\left(1 + \frac{\lambda_i}{\lambda_i \sigma_{\mathrm{t}}^2 + \sigma_{\mathrm{r}}^2}\right).$$

[T99] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Trans. Telecomm.*, vol. 10, no. 6, pp. 585-596, Sep. 1999.

Impact of Tx-noise on ergodic capacity



For $SNR \to \infty$ the **channel capacity is upper bounded** by

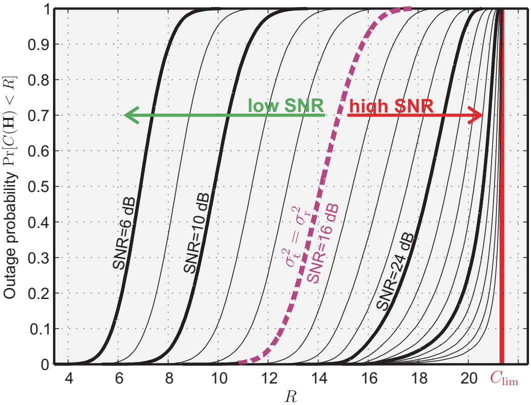
$$C_\mathrm{lim} = M_T \log_2 \left(1 + 1/\sigma_\mathrm{t}^2\right)$$

Impact of Tx-noise on outage probability

Outage probability: Cumulative distribution function (CDF) of the capacity

$$P_{\text{out}}\left(R, SNR\right) = \Pr\left(C\left(\mathbf{H}, SNR\right) < R\right)$$

R ... chosen transmission rate



$M_T = M_R = 4$, i.i.d. channel, EVM $= -15$ dB

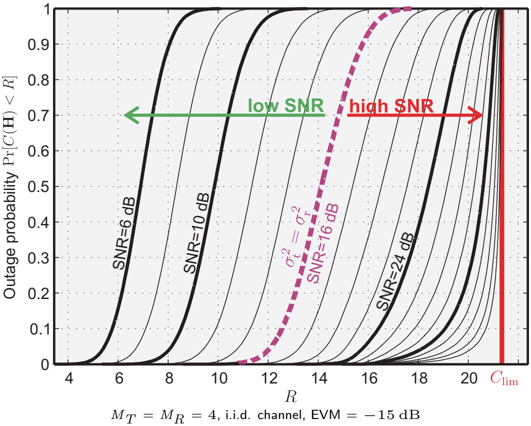
Without Tx-noise: SNR has no impact on CDF-shape

Impact of Tx-noise on outage probability

Outage probability: Cumulative distribution function (CDF) of the capacity

$$P_{\text{out}}(R, SNR) = \Pr\left(C\left(\mathbf{H}, SNR\right) < R\right)$$

R ... chosen transmission rate



$M_T = M_R = 4$, i.i.d. channel, EVM $= -15$ dB

Without Tx-noise: SNR has no impact on CDF-shape

With Tx-noise:

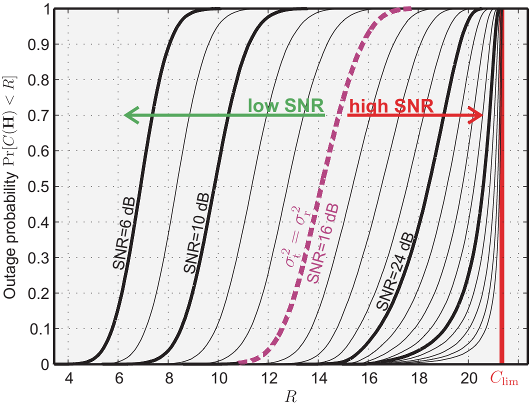
- Low SNR ($\sigma_t^2 \ll \sigma_r^2$): Transmit noise has no impact on CDF shape

Impact of Tx-noise on outage probability

Outage probability: Cumulative distribution function (CDF) of the capacity

$$P_{\text{out}}(R, SNR) = \Pr\left(C(\mathbf{H}, SNR) < R\right)$$

R ... chosen transmission rate



$M_T = M_R = 4$, i.i.d. channel, EVM $= -15$ dB

Without Tx-noise: SNR has no impact on CDF-shape

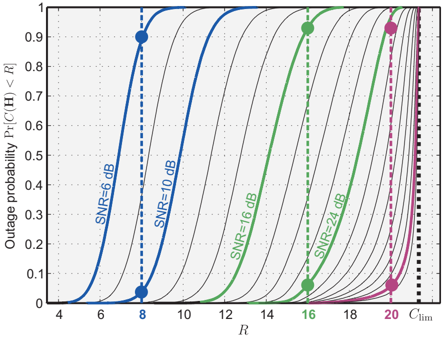
With Tx-noise:

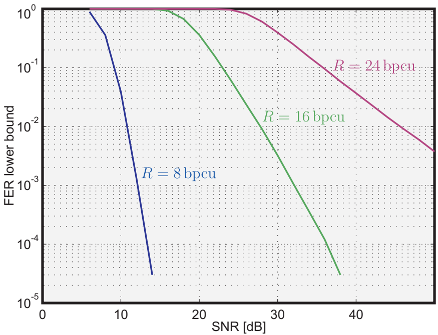
- Low SNR ($\sigma_t^2 \ll \sigma_r^2$): Transmit noise has no impact on CDF shape
- High SNR ($\sigma_t^2 \gg \sigma_r^2$): CDF becomes more and more skewed (steep) as SNR increases

Impact of Tx-noise on frame error rate

Frame error rate is lower-bounded by the outage probability [ZT03]:

$$P_{\text{out}}(R, SNR) < P_{\text{FER}}(R, SNR)$$



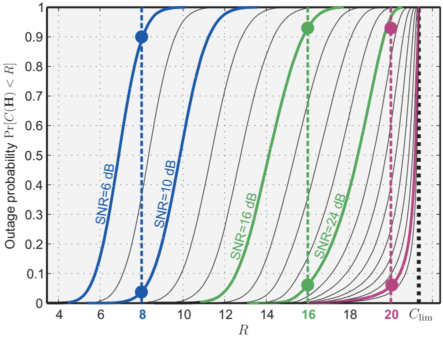


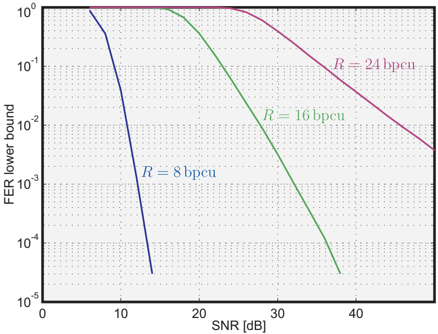
[ZT03] L. Zheng and D.N.C. Tse, D.N.C., "Diversity and multiplexing: a fundamental tradeoff in multiple-antenna channels," *IEEE Transactions on Information Theory*, vol.49, no.5, pp. 1073–1096, May 2003

Impact of Tx-noise on frame error rate

Frame error rate is lower-bounded by the outage probability [ZT03]:

$$P_{\text{out}}(R, SNR) < P_{\text{FER}}(R, SNR)$$



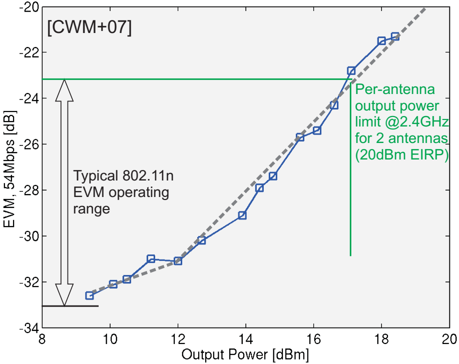


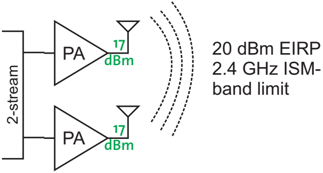
[ZT03] L. Zheng and D.N.C. Tse, D.N.C., "Diversity and multiplexing: a fundamental tradeoff in multiple-antenna channels," *IEEE Transactions on Information Theory*, vol.49, no.5, pp. 1073–1096, May 2003

In the presence of Tx-noise, achieving high rates R with **low FER** becomes increasingly difficult (requires very high SNR)

Typical EVM with real-world RF chains

EVM depends strongly on transmit power: limited by the power amplifier

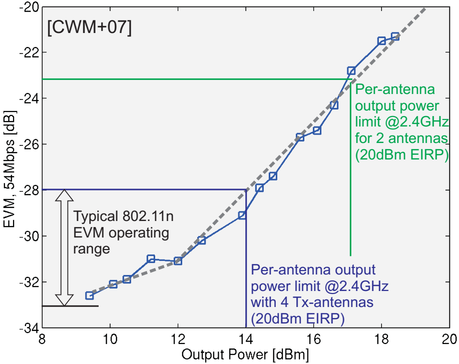


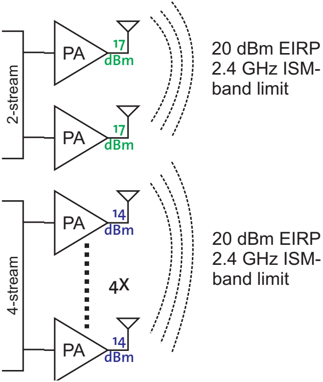


[CWM+07] R. Chang, D. Weber, L. MeeLan, D. Su, K. Vleugels, S. Wong, "A Fully Integrated RF Front-End with Independent RX/TX Matching and +20dBm Output Power for WLAN Applications," Proc. of the IEEE Solid-State Circuits Conference, Feb. 2007

Typical EVM with real-world RF chains

EVM depends strongly on transmit power: limited by the power amplifier



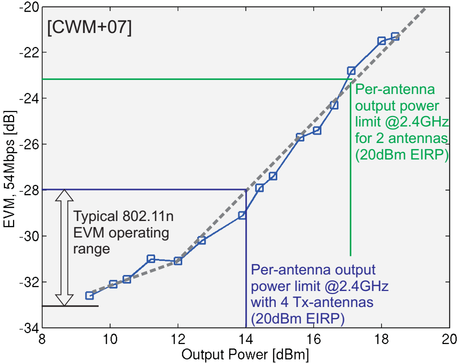

2-stream
PA
17 dBm
PA
17 dBm
20 dBm EIRP 2.4 GHz ISM-band limit
4-stream
PA
14 dBm
4X
PA
14 dBm
20 dBm EIRP 2.4 GHz ISM-band limit

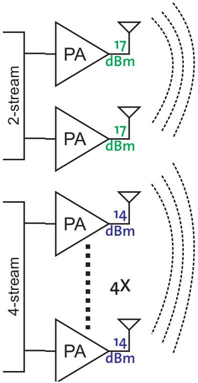
Total-power constraint: increasing the number of Tx-antennas reduces EVM

[CWM+07] R. Chang, D. Weber, L. MeeLan, D. Su, K. Vleugels, S. Wong, "A Fully Integrated RF Front-End with Independent RX/TX Matching and +20dBm Output Power for WLAN Applications," Proc. of the IEEE Solid-State Circuits Conference, Feb. 2007

Typical EVM with real-world RF chains

EVM depends strongly on transmit power: limited by the power amplifier





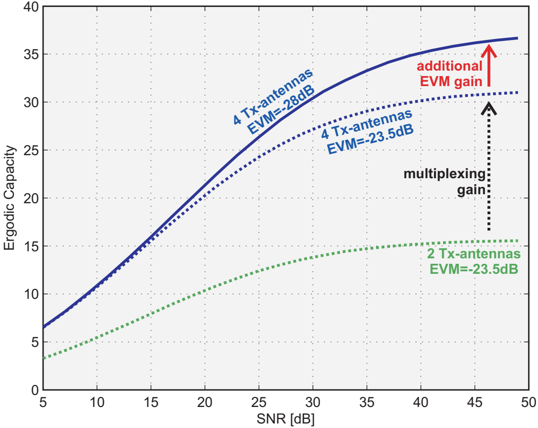
20 dBm EIRP
2.4 GHz ISM-band limit

20 dBm EIRP
2.4 GHz ISM-band limit

4-stream 802.11n system : EVM ranges from -28 dB to -33 dB

System-level implications of Tx-RF impairments

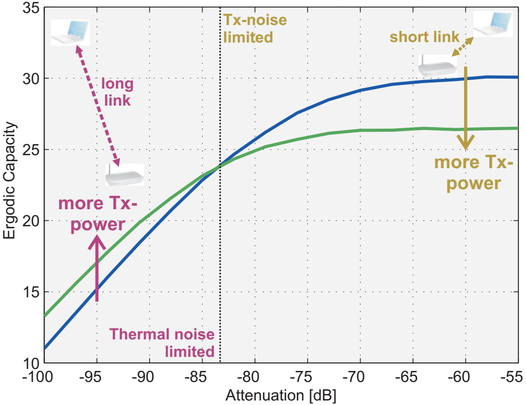
Compare ergodic capacity of systems with 2 and 4 transmit antennas under a *total-power constraint*



Additional transmit antennas reduce capacity-loss from Tx-noise

System-level implications

EVM can be improved by adjusting output power at the expense of SNR



Long distance: capacity limited by thermal noise

Short distance: capacity limited by Tx-noise

Reducing transmit power helps to increase channel capacity for short-distance links

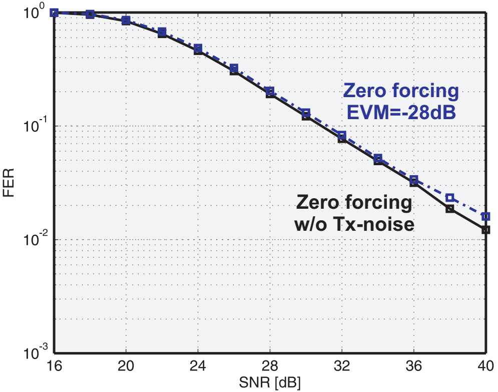
Impact on MIMO Detection

Impact of transmit-RF impairments on MIMO detector performance

Until today, **MIMO detectors** have been designed based on the **assumption of spatially white** Gaussian receive **noise**

- What is the performance impact of Tx-noise on these MIMO receivers?
- How can we mitigate the problem without too many changes or additional complexity in the receiver?

Impact of residual transmit-RF impairments on MIMO detector performance



$M_T = M_R = 4$, 64-QAM, block-fading, rate 1/2 coded, 1024 bits/frame, entries of $\mathbf{H}$ i.i.d. Gaussian

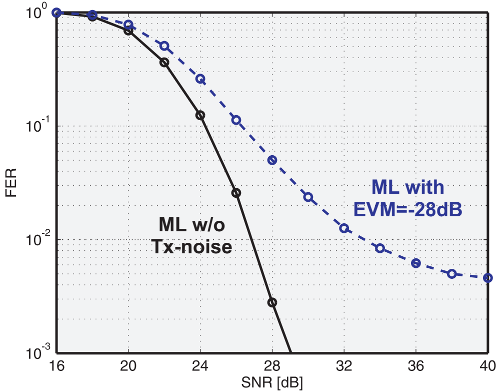
Linear (ZF or MMSE) receivers

$$\mathbf{H}^{-1}\mathbf{y} = \mathbf{s} + \mathbf{n}_t + \mathbf{H}^{-1}\mathbf{n}$$

- No assumption on noise or signal statistics
- Thermal noise $\mathbf{H}^{-1}\mathbf{n}$ dominates performance

Transmit-noise has only **minor impact** on linear receivers

Impact of residual transmit-RF impairments on MIMO detector performance



$M_T = M_R = 4$, 64-QAM, block-fading, rate 1/2 coded, 1024 bits/frame, entries of $\mathbf{H}$ i.i.d. Gaussian

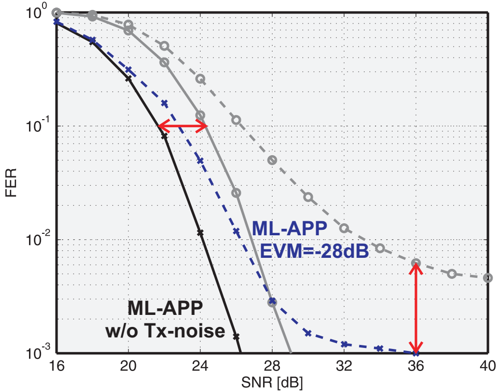
ML receiver

$$\hat{\mathbf{s}}^{\mathrm{ML}} = \arg\min_{\mathbf{s}\in\mathcal{O}^{M_T}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2$$

- Decision rule assumes noise is i.i.d. Gaussian
- Model mismatch!

ML detector is very sensitive to model mismatch

Impact of transmit-RF impairments on MIMO detector performance



$M_T = M_R = 4$, 64-QAM, block-fading, rate 1/2 coded, 1024 bits/frame, entries of $\mathbf{H}$ i.i.d. Gaussian

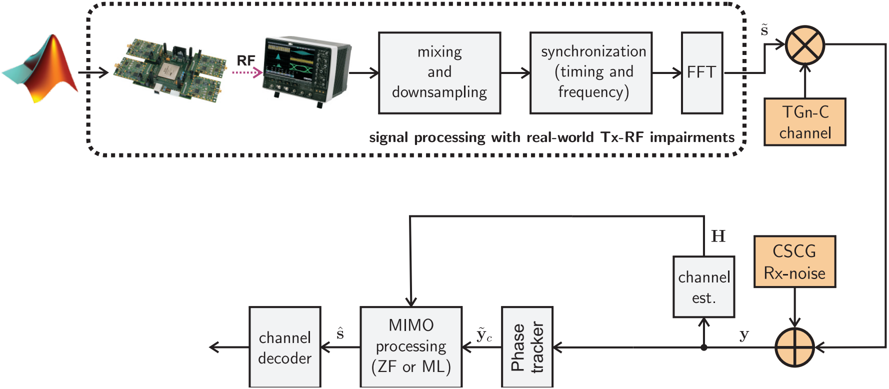
ML-APP receiver

- Assumes noise is i.i.d. Gaussian
- Model mismatch!
- SNR advantage over ML
- Impact of Tx-noise increases with SNR

SNR advantage of ML-APP over ML translates in **lower error floor**

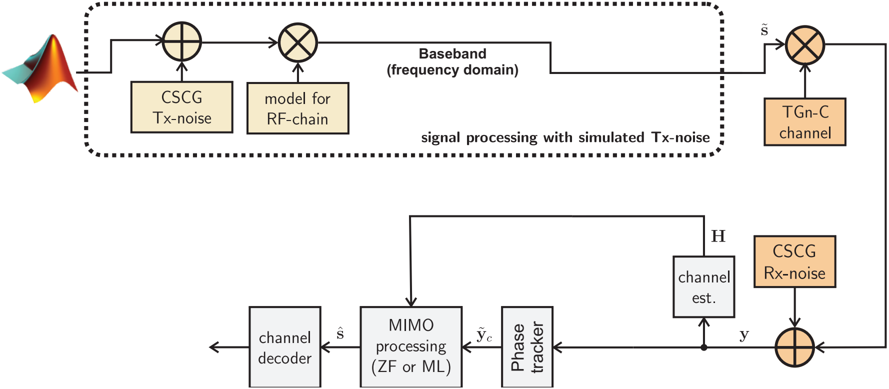
Tx-Noise - Measurement Results

Simulation and measurement results for OFDM data transmission including **synchronization**, **channel estimation**, and **pilot-tracking**



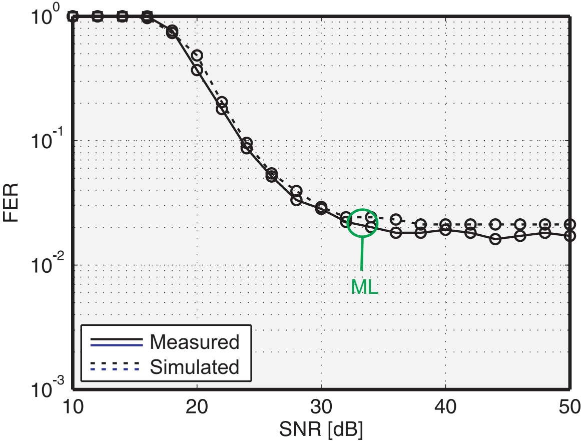
Tx-Noise - Measurement Results

Simulation and measurement results for OFDM data transmission including **synchronization**, **channel estimation**, and **pilot-tracking**



Tx-Noise - Measurement Results

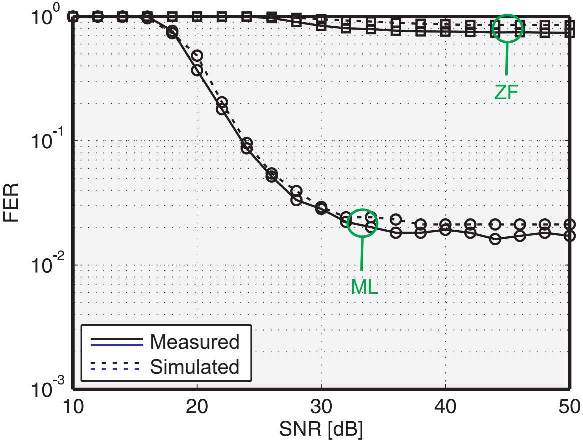
Simulation and measurement results for OFDM data transmission including **synchronization**, **channel estimation**, and **pilot-tracking**



TGn-C channel model, 16-QAM, rate 1/2, 32 OFDM data symbols

Tx-Noise - Measurement Results

Simulation and measurement results for OFDM data transmission including **synchronization**, **channel estimation**, and **pilot-tracking**

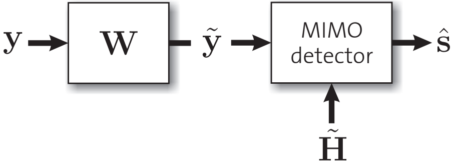


TGn-C channel model, 16-QAM, rate 1/2, 32 OFDM data symbols

MIMO Detection with Tx-RF Impairments

Noise whitening: Restore the original system model

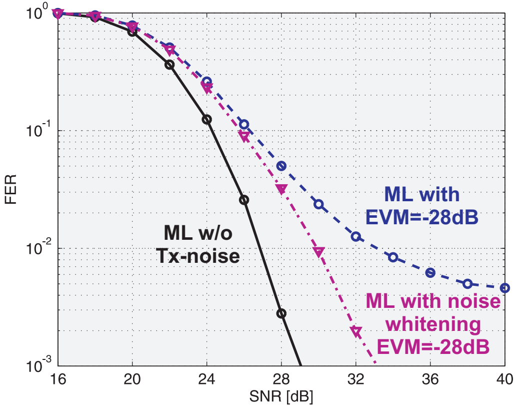
Employ spatial noise whitening at the receiver:



Whitening filter	...	$\mathbf{W} = \sqrt{\sigma_{\mathrm{r}}^2}\mathbf{K}^{-\frac{1}{2}}$
Whitened signal	...	$\tilde{\mathbf{y}} = \mathbf{W}\mathbf{y} = \tilde{\mathbf{H}}\mathbf{s} + \tilde{\mathbf{n}}$
Effective channel	...	$\tilde{\mathbf{H}} = \mathbf{W}\mathbf{H}$

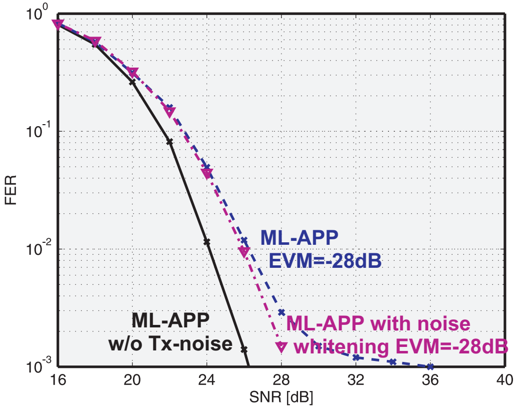
Noise seen by the detector $\tilde{\mathbf{n}} \sim \mathcal{CN}(0, \sigma_{\mathrm{r}}^2 \mathbf{I}_{M_R})$
is **spatially white** and has the **same statistics as** $\mathbf{n}$

Impact of noise whitening on error-rate performance



Noise whitening leads to a **significant performance improvement** for **ML detectors** in the presence of transmit-RF impairments

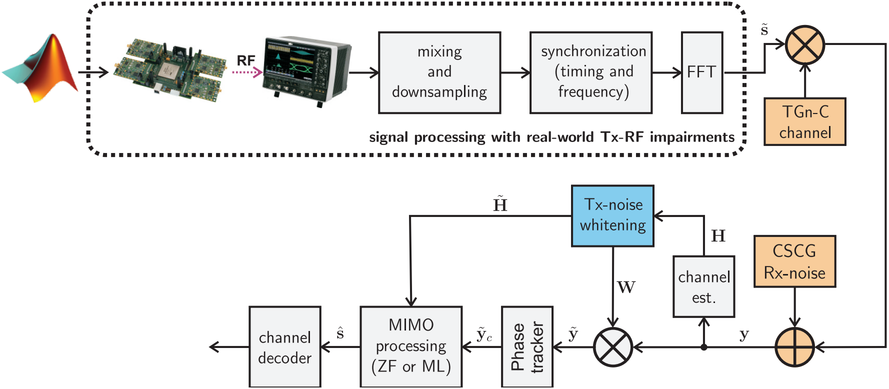
Impact of noise whitening on error performance



Noise whitening leads to a **significant performance improvement** for **ML-APP detectors** (STS-SD) in the presence of transmit-RF impairments

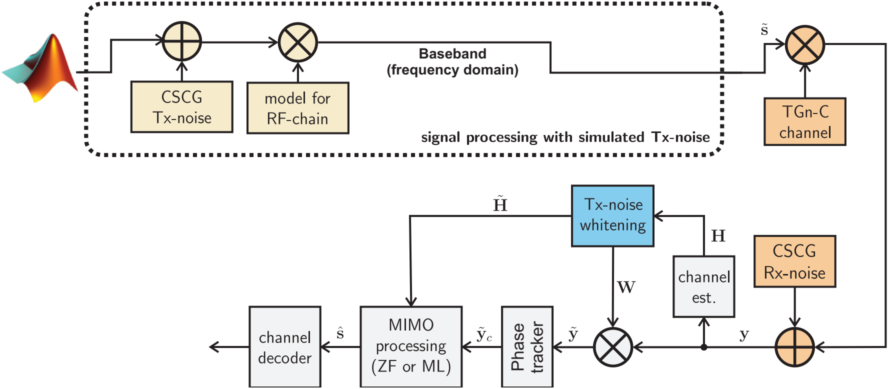
Tx-Noise - Measurement Results

Simulation and measurement results for OFDM data transmission including **synchronization**, **channel estimation**, and **pilot-tracking**



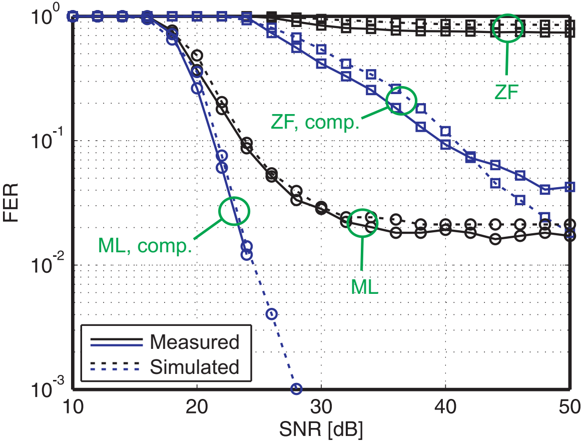
Tx-Noise - Measurement Results

Simulation and measurement results for OFDM data transmission including **synchronization**, **channel estimation**, and **pilot-tracking**



Tx-Noise - Measurement Results

Simulation and measurement results for OFDM data transmission including **synchronization**, **channel estimation**, and **pilot-tracking**



TGn-C channel model, 16-QAM, rate 1/2, 32 OFDM data symbols

Computation of the noise-whitening filter

Straightforward computation of the noise-whitening filter matrix:

$$\mathbf{W} = \sqrt{\sigma_{\mathrm{r}}^2}\mathbf{K}^{-\frac{1}{2}}$$

1. Computation of the covariance matrix: $\mathbf{K} = \sigma_{\mathrm{t}}^2\mathbf{H}\mathbf{H}^H + \sigma_{\mathrm{r}}^2\mathbf{I}_{M_R}$
2. Cholesky decomposition: $\mathbf{K} = \mathbf{L}\mathbf{L}^H$ with $\mathbf{L}$ lower-triangular
3. Matrix inversion and scaling: $\mathbf{W} = \sqrt{\sigma_{\mathrm{r}}^2}\mathbf{L}^{-1}$

Computation of the noise-whitening filter

Straightforward computation of the noise-whitening filter matrix:

$$\mathbf{W} = \sqrt{\sigma_{\mathrm{r}}^2}\mathbf{K}^{-\frac{1}{2}}$$

1. Computation of the covariance matrix: $\mathbf{K} = \sigma_{\mathrm{t}}^2 \mathbf{H}\mathbf{H}^H + \sigma_{\mathrm{r}}^2 \mathbf{I}_{M_R}$
2. Cholesky decomposition: $\mathbf{K} = \mathbf{L}\mathbf{L}^H$ with $\mathbf{L}$ lower-triangular
3. Matrix inversion and scaling: $\mathbf{W} = \sqrt{\sigma_{\mathrm{r}}^2}\mathbf{L}^{-1}$

This method suffers from high computational **complexity**
and requires considerable **arithmetic precision**

Efficient QR-decomposition-based solution

- Covariance matrix $\mathbf{K}$ can be written as

$$\mathbf{K} = \sigma_{\mathrm{t}}^2 \mathbf{H}\mathbf{H}^H + \sigma_{\mathrm{r}}^2 \mathbf{I}_{M_R} = \bar{\mathbf{H}}^H \bar{\mathbf{H}} \quad \text{with} \quad \bar{\mathbf{H}} = \begin{bmatrix} \sqrt{\sigma_{\mathrm{t}}^2}\mathbf{H}^H \\ \sqrt{\sigma_{\mathrm{r}}^2}\mathbf{I}_{M_R} \end{bmatrix}$$

Efficient QR-decomposition-based solution

- Covariance matrix $\mathbf{K}$ can be written as

$$\mathbf{K} = \sigma_{\mathrm{t}}^2 \mathbf{H}\mathbf{H}^H + \sigma_{\mathrm{r}}^2 \mathbf{I}_{M_R} = \bar{\mathbf{H}}^H \bar{\mathbf{H}} \quad \text{with} \quad \bar{\mathbf{H}} = \begin{bmatrix} \sqrt{\sigma_{\mathrm{t}}^2}\mathbf{H}^H \\ \sqrt{\sigma_{\mathrm{r}}^2}\mathbf{I}_{M_R} \end{bmatrix}$$

- Application of economy-size QR-decomposition (QRD) to $\bar{\mathbf{H}}$:

$$\begin{bmatrix} \sqrt{\sigma_{\mathrm{t}}^2}\mathbf{H}^H \\ \sqrt{\sigma_{\mathrm{r}}^2}\mathbf{I}_{M_R} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_a \\ \mathbf{Q}_c \end{bmatrix} \tilde{\mathbf{R}}$$

  - Since $\tilde{\mathbf{R}}^H \tilde{\mathbf{R}} = \mathbf{K}$, we have $\tilde{\mathbf{R}}^H = \mathbf{K}^{\frac{1}{2}}$
  - Since $\mathbf{Q}_c \tilde{\mathbf{R}} = \sqrt{\sigma_{\mathrm{r}}^2}\mathbf{I}_{M_R}$, we have $\mathbf{Q}_c = \sqrt{\sigma_{\mathrm{r}}^2}\tilde{\mathbf{R}}^{-1}$

Efficient QR-decomposition-based solution

- Covariance matrix $\mathbf{K}$ can be written as

$$\mathbf{K} = \sigma_{\mathrm{t}}^2 \mathbf{H}\mathbf{H}^H + \sigma_{\mathrm{r}}^2 \mathbf{I}_{M_R} = \bar{\mathbf{H}}^H \bar{\mathbf{H}} \quad \text{with} \quad \bar{\mathbf{H}} = \begin{bmatrix} \sqrt{\sigma_{\mathrm{t}}^2}\mathbf{H}^H \\ \sqrt{\sigma_{\mathrm{r}}^2}\mathbf{I}_{M_R} \end{bmatrix}$$

- Application of economy-size QR-decomposition (QRD) to $\bar{\mathbf{H}}$:

$$\begin{bmatrix} \sqrt{\sigma_{\mathrm{t}}^2}\mathbf{H}^H \\ \sqrt{\sigma_{\mathrm{r}}^2}\mathbf{I}_{M_R} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_a \\ \mathbf{Q}_c \end{bmatrix} \tilde{\mathbf{R}}$$

  - Since $\tilde{\mathbf{R}}^H \tilde{\mathbf{R}} = \mathbf{K}$, we have $\tilde{\mathbf{R}}^H = \mathbf{K}^{\frac{1}{2}}$
  - Since $\mathbf{Q}_c \tilde{\mathbf{R}} = \sqrt{\sigma_{\mathrm{r}}^2}\mathbf{I}_{M_R}$, we have $\mathbf{Q}_c = \sqrt{\sigma_{\mathrm{r}}^2}\tilde{\mathbf{R}}^{-1}$

Noise-whitening filter is immediately given by $\mathbf{W} = \sqrt{\sigma_{\mathrm{r}}^2}\mathbf{K}^{-\frac{1}{2}} = \mathbf{Q}_c^H$

Efficient QR-decomposition-based solution

- Covariance matrix $\mathbf{K}$ can be written as

$$\mathbf{K} = \sigma_{\mathrm{t}}^2 \mathbf{H}\mathbf{H}^H + \sigma_{\mathrm{r}}^2 \mathbf{I}_{M_R} = \bar{\mathbf{H}}^H \bar{\mathbf{H}} \quad \text{with} \quad \bar{\mathbf{H}} = \begin{bmatrix} \sqrt{\sigma_{\mathrm{t}}^2}\mathbf{H}^H \\ \sqrt{\sigma_{\mathrm{r}}^2}\mathbf{I}_{M_R} \end{bmatrix}$$

- Application of economy-size QR-decomposition (QRD) to $\bar{\mathbf{H}}$:

$$\begin{bmatrix} \sqrt{\sigma_{\mathrm{t}}^2}\mathbf{H}^H \\ \sqrt{\sigma_{\mathrm{r}}^2}\mathbf{I}_{M_R} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_a \\ \mathbf{Q}_c \end{bmatrix} \tilde{\mathbf{R}}$$

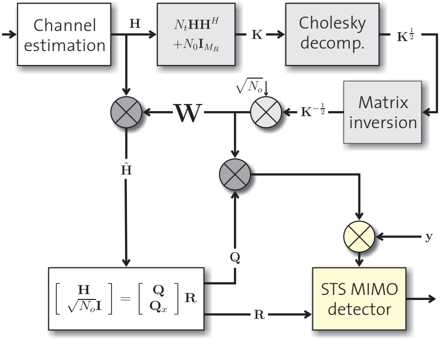
  - Since $\tilde{\mathbf{R}}^H \tilde{\mathbf{R}} = \mathbf{K}$, we have $\tilde{\mathbf{R}}^H = \mathbf{K}^{\frac{1}{2}}$
  - Since $\mathbf{Q}_c \tilde{\mathbf{R}} = \sqrt{\sigma_{\mathrm{r}}^2}\mathbf{I}_{M_R}$, we have $\mathbf{Q}_c = \sqrt{\sigma_{\mathrm{r}}^2}\tilde{\mathbf{R}}^{-1}$

Noise-whitening filter is immediately given by $\mathbf{W} = \sqrt{\sigma_{\mathrm{r}}^2}\mathbf{K}^{-\frac{1}{2}} = \mathbf{Q}_c^H$

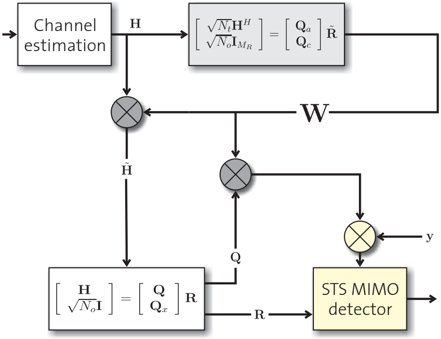
Requires **one** economy-size QRD at preprocessing rate

Algorithm comparison

Straightforward implementation

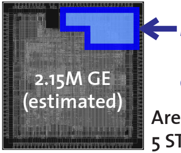

Channel estimation
$\mathbf{H}$
$N_t\mathbf{H}\mathbf{H}^H$ $+N_0\mathbf{I}_{M_R}$
$\mathbf{K}$
Cholesky decomp.
$\mathbf{K}^{\frac{1}{2}}$
$\sqrt{N_o}$
$\mathbf{W}$
$\mathbf{K}^{-\frac{1}{2}}$
Matrix inversion
$\tilde{\mathbf{H}}$
$\mathbf{y}$
$\mathbf{Q}$
$\left[\begin{array}{c}\mathbf{H}\\ \sqrt{N_o}\mathbf{I}\end{array}\right]=\left[\begin{array}{c}\mathbf{Q}\\ \mathbf{Q}_x\end{array}\right]\mathbf{R}$
$\mathbf{R}$
STS MIMO detector

QR-based implementation


Channel estimation
$\mathbf{H}$
$\left[\begin{array}{c}\sqrt{N_t}\mathbf{H}^H\\ \sqrt{N_o}\mathbf{I}_{M_R}\end{array}\right]=\left[\begin{array}{c}\mathbf{Q}_a\\ \mathbf{Q}_c\end{array}\right]\tilde{\mathbf{R}}$
$\mathbf{W}$
$\tilde{\mathbf{H}}$
$\mathbf{y}$
$\mathbf{Q}$
$\left[\begin{array}{c}\mathbf{H}\\ \sqrt{N_o}\mathbf{I}\end{array}\right]=\left[\begin{array}{c}\mathbf{Q}\\ \mathbf{Q}_x\end{array}\right]\mathbf{R}$
$\mathbf{R}$
STS MIMO detector

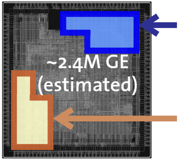
QR-based implementation of noise whitening is **numerically more stable** and has **lower computational complexity**

Impact of QRD-based noise whitening on silicon area (estimated)



MMSE-QRD STS preprocessing: 0.25M GE

Area reference: 5 STS units



MMSE-QRD STS preprocessing

MMSE-QRD for noise whitening

Assumptions for area estimation

- Area of noise whitening dominated by QR decomposition for computing $\mathbf{W}$
- Area estimates based on MMSE-QR preprocessing required for MMSE and STS MIMO detector
- Preprocessing latency must remain constant

Transmit-noise whitening has **only minor impact on overall silicon area**

References

References I

- S. Cherry, “Edholm’s law of bandwidth,” *IEEE Signal Proc. Magazine*, vol. 41, no. 7, pp. 58–60, Jul. 2004.
- A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*. Cambridge Univ. Press, 2003.
- I. B. Collings, M. R. G. Butler, and M. McKay, “Low complexity receiver design for MIMO bit-interleaved coded modulation,” in *IEEE Eighth International Symposium on Spread Spectrum Techniques and Applications*, 2004, pp. 12–16.
- S. Haene, A. Burg, D. Perels, P. Luethi, N. Felber, and W. Fichtner, “Silicon implementation of an MMSE-based soft demapper for MIMO-BICM,” in *Proc. of the IEEE Int. Symp. on Circuits and Systems*, May 2006.
- G. H. Golub and C. F. Van Loan, *Matrix Computations*. John Hopkins Univ. Press, 1996.

References II

- R.-H. Lai, C.-M. Chen, P.-A. Ting, and Y.-H. Huang, “A modified sorted-QR decomposition algorithm for parallel processing in MIMO detection,” in *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, may 2009, pp. 1405 –1408.

- A. Burg, D. Seethaler, and G. Matz, “Vlsi implementation of a lattice-reduction algorithm for multi-antenna broadcast precoding,” in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, may 2007, pp. 673 –676.

- C. Ling and N. Howgrave-Graham, “Effective LLL reduction for lattice decoding,” in *Proc. IEEE ISIT*, Jun. 2007, pp. 196–200.

- H. Vetter, V. Ponnampalam, M. Sandell, and P. A. Hoeher, “Fixed complexity LLL algorithm,” *IEEE Trans. Signal Process.*, vol. 57, pp. 1634–1637, Apr. 2009.

References III

- L. Bruderer, C. Studer, D. Seethaler, M. Wenk, and A. Burg, “VLSI implementation of a low-complexity LLL lattice reduction algorithm for MIMO detection,” in *Proc. IEEE ISCAS*, May 2010.
- B. Vallée, “The gaussàlgorithm revisited,” *Journal of Algorithms*, vol. 12, no. 4, pp. 556–572, Dec. 1991.
- J. Lagarias, H. Lenstra, and C. Schnorr, “Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice,” *Combinatorica*, vol. 10, no. 4, pp. 333–348, 1990.
- D. Seethaler, G. Matz, and F. Hlawatsch, “Low-complexity MIMO data detection using Seysen’s lattice reduction algorithm,” in *Proc. IEEE ICASSP*, vol. 3, Apr. 2007, pp. 53–56.
- L. Bruderer, C. Senning, and A. Burg, “Low-complexity Seysen’s algorithm based lattice reduction-aided MIMO detection for hardware implementations,” in *Proc. IEEE Asilomar*, Nov. 2010.

References IV

- D. Garrett, L. Davis, and G. Woodward, “19.2 mbit/s 4 /spl times/ 4 BLAST/MIMO detector with soft ML outputs,” *IEE Electronics Letters*, vol. 39, pp. 233–235, 2003.
- A. Burg, N. Felber, and W. Fichtner, “A 50 Mbps 4×4 maximum likelihood decoder for multiple-input multiple-output systems with QPSK modulation,” in *Proc. IEEE ICECS*, vol. 1, 2003, pp. 332–335.
- M. Pohst, “On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications,” *SIGSAM Bulletin*, vol. 15, no. 1, pp. 37–44, Feb. 1981.
- U. Fincke and M. Pohst, “Improved methods for calculating vectors of short length in a lattice, including a complexity analysis,” *Mathematics of Computation*, vol. 44, no. 170, pp. 463–471, Apr. 1985.

References V

- M. O. Damen, H. El Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.

- C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improving practical lattice basis reduction and solving subset sum problems," *Math. Programming*, vol. 66, pp. 181–191, 1994.

- A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Boelcskei, "Vlsi implementation of mimo detection using the sphere decoding algorithm," *IEEE Journal on Solid-State Circuits*, vol. 40, no. 3, pp. 1566–1577, Jul. 2005.

- A. Burg, M. Wenk, M. Zellweger, M. Wegmueller, N. Felber, and W. Fichtner, "VLSI implementation of the sphere decoder algorithm," in *Proc. IEEE ESSCIRC*, 2004.

References VI

B. M. Hochwald and S. ten Brink, “Achieving near-capacity on a multiple-antenna channel,” *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, 2003.

A. Burg, M. Borgmann, M. Wenk, C. Studer, and H. Bölcskei, “Advanced receiver algorithms for MIMO wireless communications,” in *Proc. of the Design Automation and Test Europe Conf.*, vol. 1, Mar. 2006, pp. 593–598, invited paper.

K. Wong, C. Tsui, R.-K. Cheng, and W. Mow, “A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels,” in *Proc. IEEE ISCAS*, vol. 3, 2002, pp. 273–276.

Z. Guo and P. Nilsson, “A 53.3 Mb/s 4×4 16-QAM MIMO decoder in 0.35μm CMOS,” in *Proc. IEEE ISCAS*, May 2005, pp. 4947–4950.

References VII

- R. F. H. Fischer and C. Windpassinger, "Real versus complex-valued equalization in V-BLAST systems," *IEE Electronics Letters*, vol. 39, no. 5, pp. 470–471, Mar. 2003.
- M. Wenk, M. Zellweger, A. Burg, N. Felber, and W. Fichtner, "K-best MIMO detection VLSI architectures achieving up to 424 Mbps," in *Proc. of the IEEE Int. Symp. on Circuits and Systems*, May 2006.
- S. C., W. M., and B. A., "Soft-output sphere decoding: Performance and implementation aspects," in *Proceedings Asilomar Conference on Signals, Systems and Computers*, Nov. 2006.