Algorithms for Wireless Communications II

Heinrich Meyr, Andreas Burg

Institute for Integrated Signal Processing Systems, RWTH Aachen

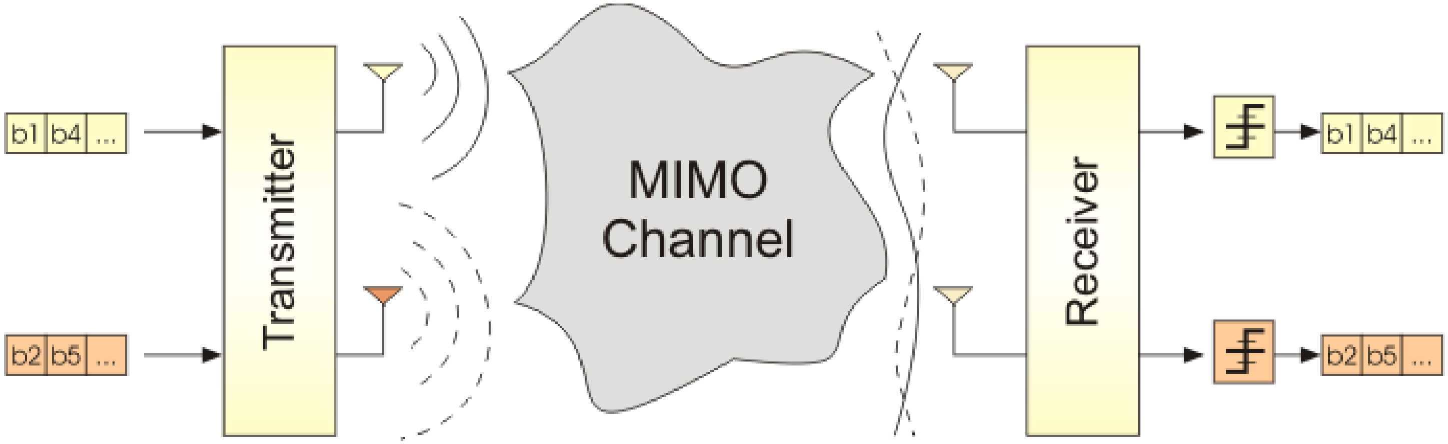
Telecommunications Circuits Laboratory, EPFL

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Basic MIMO Receivers

- System Model
- Maximum Likelihood Receiver
- Linear Receivers (MMSE & ZF)
- *Successive Cancellation Receivers*
- *Performance*

Reminder: MIMO Spatial Multiplexing

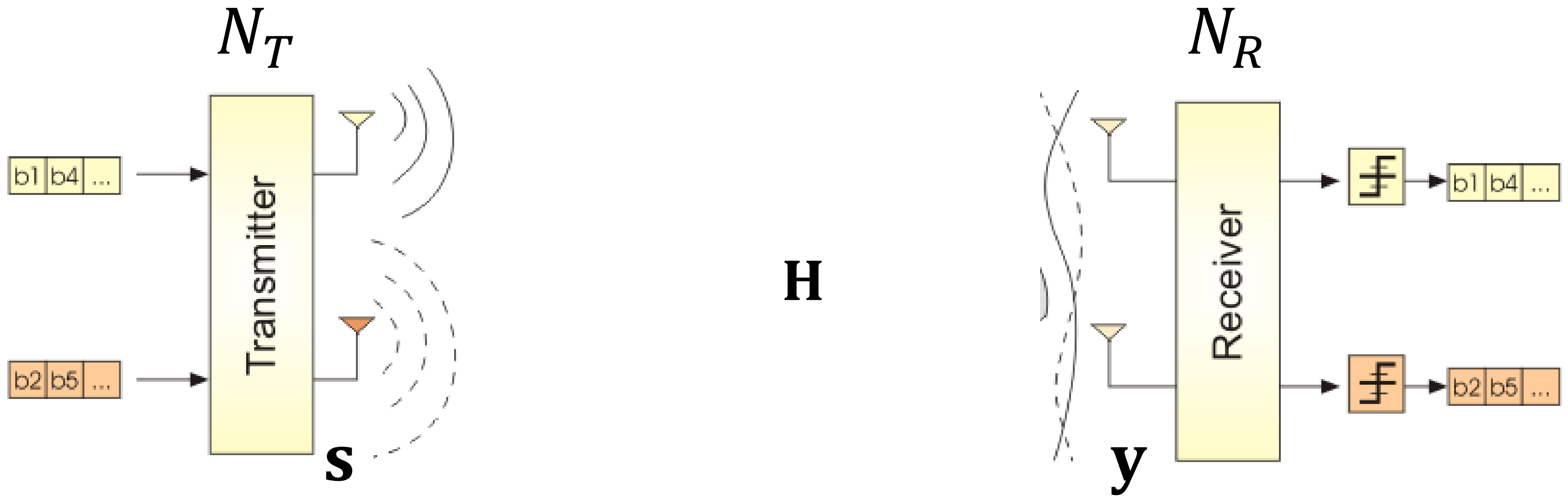


- Split the high rate data stream into N_T independent, lower rate streams
- Each stream is modulated independently using M-QAM constellations
- No transmit channel knowledge: each stream is transmitted from its own transmit antenna (at the same time in the same frequency band)

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

RWTH AACHEN UNIVERSITY

System Model for Spatial Multiplexing



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

- $\mathbf{s} = [s_1 \quad \cdots \quad s_{N_T}]^T$: Transmitted vector-symbol with $s_i \in \mathcal{O}$ ➔ $\mathbf{s} \in \mathcal{O}^{N_T}$
- $\mathcal{O}$: Set of constellation points
- $\mathbf{H} = [\mathbf{h}_1 \quad \cdots \quad \mathbf{h}_{N_T}]$: Channel matrix
- $\mathbf{y} = [y_1 \quad \cdots \quad y_{N_T}]^T$: Received vector
- $\mathbf{n} = [n_1 \quad \cdots \quad n_{N_T}]^T$: Gaussian noise with i.i.d. entries $\mathbf{n} \sim \mathcal{CN}(0, \sigma^2)$

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

RWTH AACHEN UNIVERSITY

Optimum MIMO Detection

Task of the MIMO detector: estimate the transmitted symbol vector $\mathbf{s}$ based on the received vector $\mathbf{y}$

- Assume we know the channel matrix $\mathbf{H}$

- Best possible estimate if obtained using Maximum Likelihood criterion

$$\hat{\boldsymbol{s}}_{\mathrm{ML}} = \arg\max_{\hat{\boldsymbol{s}} \in \mathcal{O}^{N_T}} Pr(\boldsymbol{y}|\boldsymbol{s} = \hat{\boldsymbol{s}})$$

- The noise is i.i.d. Gaussian, hence

$$Pr(\boldsymbol{y}|\boldsymbol{s} = \hat{\boldsymbol{s}}) \propto \boldsymbol{e}^{-\|\boldsymbol{y}-\boldsymbol{H}\hat{\boldsymbol{s}}\|^{2}}$$

  - Remember: taking the logarithm yields a sufficient statistics

- The ML detection rule for MIMO spatial multiplexing becomes

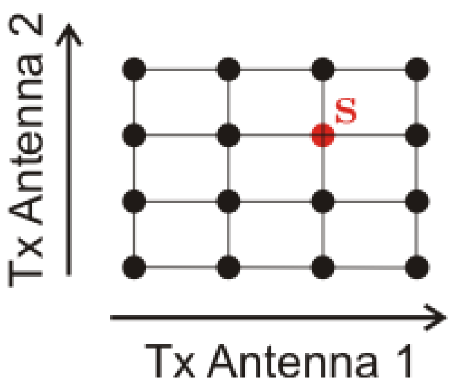
$$\hat{\boldsymbol{s}}_{\mathrm{ML}} = \arg\min_{\hat{\boldsymbol{s}} \in \mathcal{O}^{N_T}} \|\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{s}}\|^{2}$$

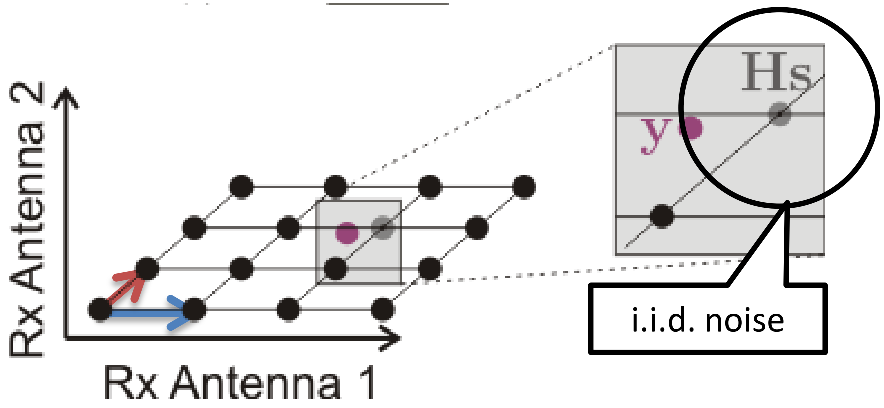
Graphical Interpretation

- Consider a system with $N_T = N_R = 2$ (2x2) with real-valued 4-PAM modulation on each antenna/stream

- Transmitter: streams lie along the orthogonal axis in the signal space
- Receiver: Signal components lie in the direction of the columns of $\mathbf{H}$
  ➔**Skewed constellation points**

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0.3 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$$





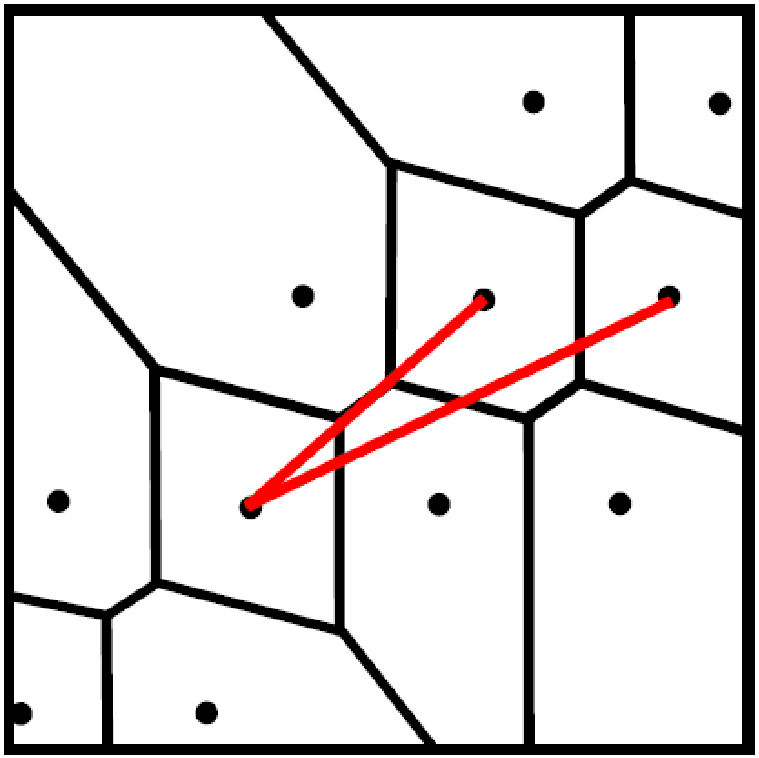
- The noise translates the received point $\mathbf{y}$ away from $\mathbf{Hs}$

ML: find the point among all $\mathbf{Hs}$ that lies closest to $\mathbf{y}$

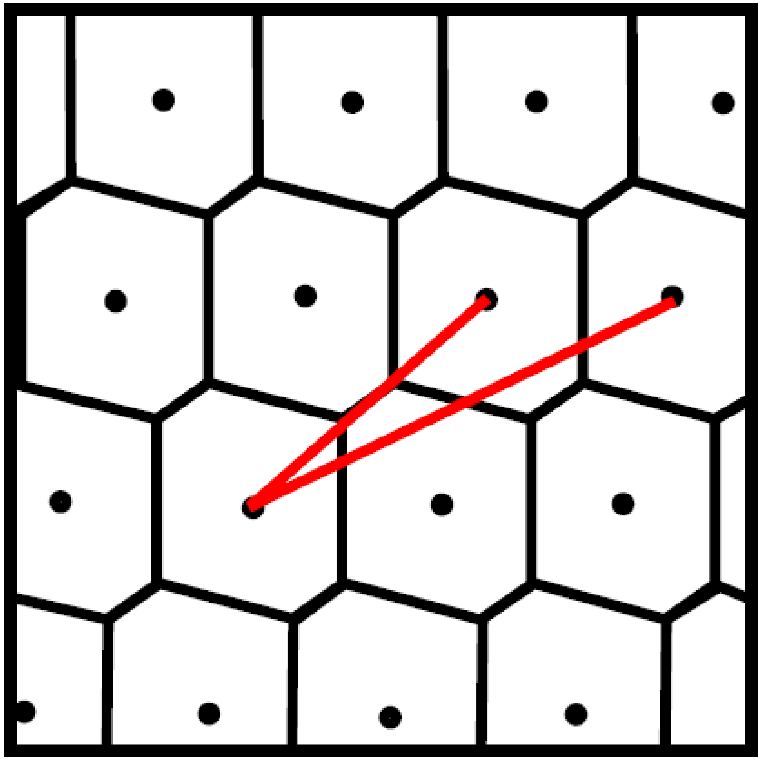
Voronoi Regions are Decision Regions of the ML Detector

Voronoi region: region around a reference constellation point in which all points lie closer to this point than to any other constellation point

Examples



Finite constellation alphabet



Infinite constellation alphabet

Complexity of ML Detection

- Straightforward approach to solving the ML detection problem: Evaluate $||\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{s}}||^2$ for all possible candidate symbols and find the minimum

- Number of possible candidates
  - For M-QAM: M possible candidates for each spatial stream
  - For N_T spatial streams: M^{N_T} candidates

- Assume $M = 2^q$: transmit q bits on each stream
- Number of symbols: 2^{qN_T}
- Spectral efficiency: qN_T bits per vector-symbol (bits/s/Hz)

Number of candidate symbols to check grows exponentially with the spectral efficiency

- Example: 4 streams, 64-QAM
  - Spectral efficiency is 24 bits/s/Hz
  - For each received symbol ML detection checks 16 million candidates

Linear MIMO Detection

- ML detection is complex since it considers all spatial streams jointly

A pragmatic solution

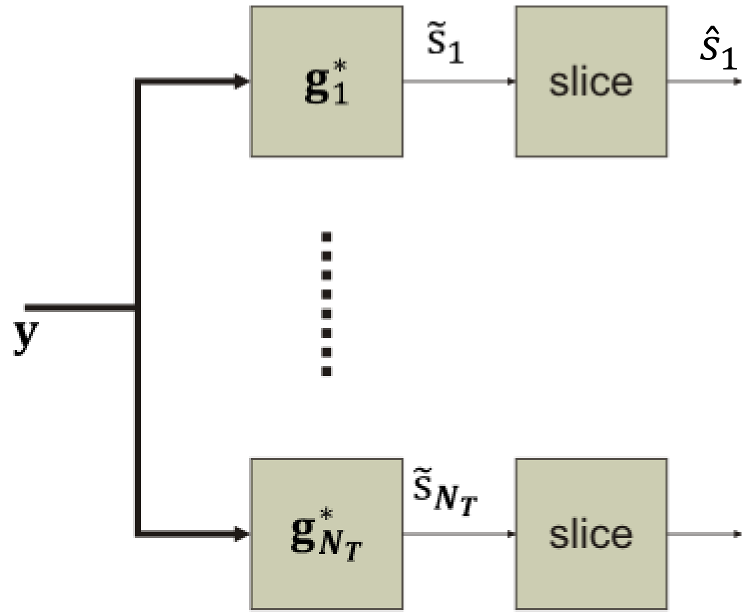
- Initially neglect the fact that constellation points are chosen from a limited set
- Use a linear estimator to recover the transmitted signal

$$\tilde{s}_k = \mathbf{g}_k^* \mathbf{y} = \mathbf{g}_k^* \mathbf{H}\mathbf{s} + \mathbf{g}_k^* \mathbf{n}$$

$$\tilde{\mathbf{s}} = \mathbf{G}\mathbf{y} = \mathbf{G}\mathbf{H}\mathbf{s} + \mathbf{G}\mathbf{n} \quad \text{with} \quad \mathbf{G}^{\mathrm{H}} = [\mathbf{g}_1 \quad \cdots \quad \mathbf{g}_{N_T}]$$

- Map each recovered stream to the nearest constellation point (slicing)

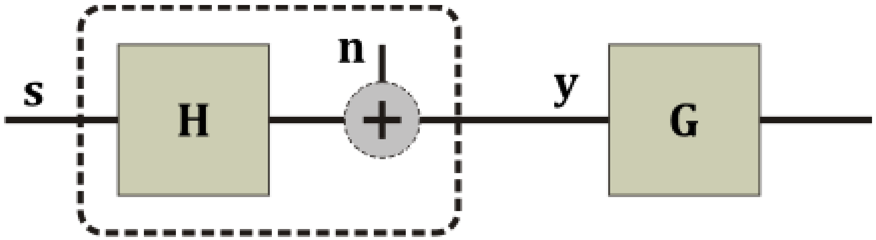
$$\hat{s}_k = Q(\tilde{s}_k)$$



How to choose G??

Linear Zero Forcing Detector

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



Choose the linear filter to undo the effect of the channel

$$\mathbf{G}\mathbf{H} = \mathbf{I} \quad \text{so that} \quad \mathbf{G}\mathbf{y} = \mathbf{s} + \mathbf{G}\mathbf{n}$$

- For $N_T = N_R$: $\mathbf{G} = \mathbf{H}^{-1}$
- For $N_T < N_R$: use the least-squares solution $\mathbf{G} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H = \mathbf{H}^\dagger$
  - Given by the pseudo inverse
  - For $N_T = N_R$, $\mathbf{H}^\dagger = \mathbf{H}^{-1}$
- For $N_T > N_R$: no solution exists (linear detection is not possible)

Performance of Zero Forcing with a Static Channel

- Consider $\mathbf{G} = \mathbf{H}^{\dagger}$

- The input of the slicers is given by
  - The transmitted symbol
  - The additive Gaussian noise multiplied with the zero-forcing equalizer

$$\boxed{\tilde{\mathbf{s}} = \mathbf{H}^{\dagger}\mathbf{y}} = \mathbf{s} + \underbrace{\mathbf{H}^{\dagger}\mathbf{n}}_{\tilde{\mathbf{n}}}$$

- The zero forcing equalizer completely eliminates interference between streams
- The noise is now correlated (colored)

$$\mathbb{E}\{\tilde{\mathbf{n}}\tilde{\mathbf{n}}^{H}\} = \sigma^{2}(\mathbf{H}^{H}\mathbf{H})^{-1}$$

- By decoding each stream independently, we ignore this noise correlation

Performance of Zero Forcing with a Static Channel

- Diagonal elements of the noise covariance matrix correspond to the noise variance on the individual streams
- Signal to noise ratio for each stream is given by

$$ZF - SNR_i = \frac{P_S/N_T}{\sigma^2[(\mathbf{H}^H\mathbf{H})^{-1}]_{ii}}$$

- The total signal to noise ratio is given by

$$ZF - SNR = \frac{P_S}{\sigma^2 \sum_{\boldsymbol{i=1}}^{N_T}[(\mathbf{H}^H\mathbf{H})^{-1}]_{ii}}$$

- With the SVD of $\mathbf{H} = \mathbf{U}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{H}}$ we find that $\sum_{\boldsymbol{i=1}}^{N_T}[(\mathbf{H}^H\mathbf{H})^{-1}]_{ii} = \sum_{\boldsymbol{i=1}}^{N_T}\frac{1}{\lambda_i^2}$
  - λ_i : Eigenvalues of $\mathbf{H}$

$$ZF - SNR = \frac{P_S}{\sigma^2 \sum_{\boldsymbol{i=1}}^{N_T}\frac{1}{\lambda_i^2}} < \frac{P_S\lambda_{N_T}^2}{\sigma^2}$$

- **If the channel has a small Eigenvalue the SNR degrades significantly**

Performance of Zero Forcing with a Static Channel

For comparison, we use a lower bound on the SNR without the zero forcing equalizer

- Transmitted signal has power P_S and $\lambda_{N_T}^2$ is the smallest Eigenvalue of the channel
- Total received signal power : $P_S \sum_{\boldsymbol{i=1}}^{N_T} \lambda_i^2 > P_S N_T \lambda_{N_T}^2$ (lower bound with smallest EV)
- Total noise power $\sigma^2 N_T$

$$SNR = \frac{P_S \sum_{\boldsymbol{i=1}}^{N_T} \lambda_i^2}{\sigma^2 N_T} > \frac{P_S \lambda_{N_T}^2}{\sigma^2}$$

Compare this to the SNR after zero forcing

$$ZF - SNR < \frac{P_S \lambda_{N_T}^2}{\sigma^2} < SNR$$

Noise enhancement: zero forcing degrades SNR

A Graphical Interpretation

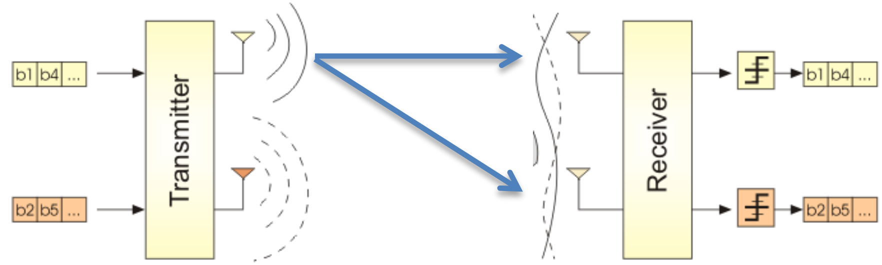
Assume we are interested in the 1st stream

$$\mathbf{y} = \mathbf{h}_1 s_1 + \sum_{i \neq k} \mathbf{h}_i s_i + \mathbf{n}$$

Desired stream

Interference (other streams)

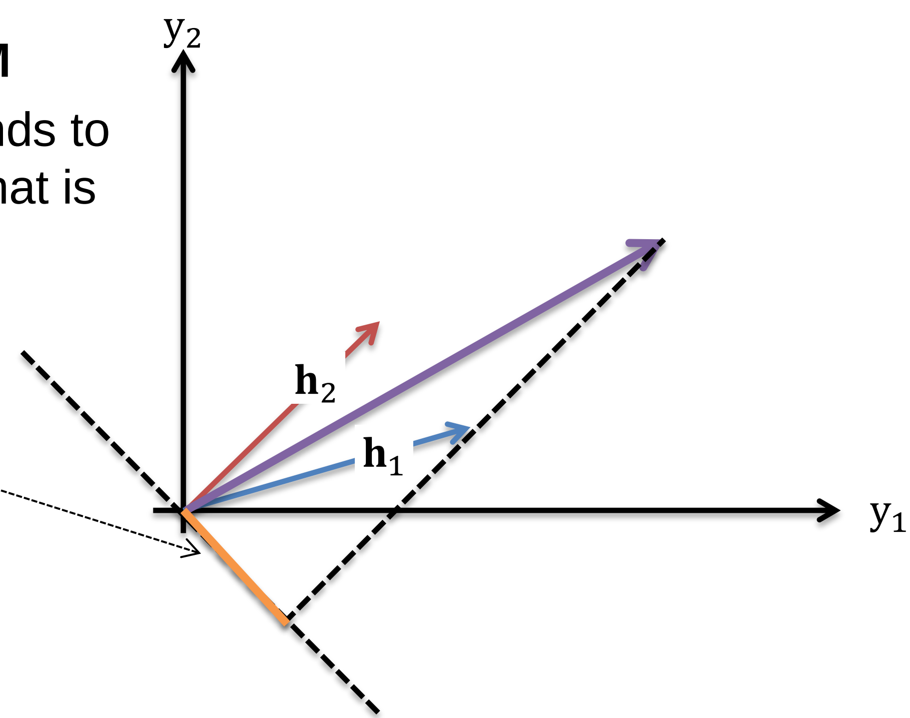
Noise



$$\mathbf{H} = [\mathbf{h}_1 \quad \cdots \quad \mathbf{h}_{N_T}]$$

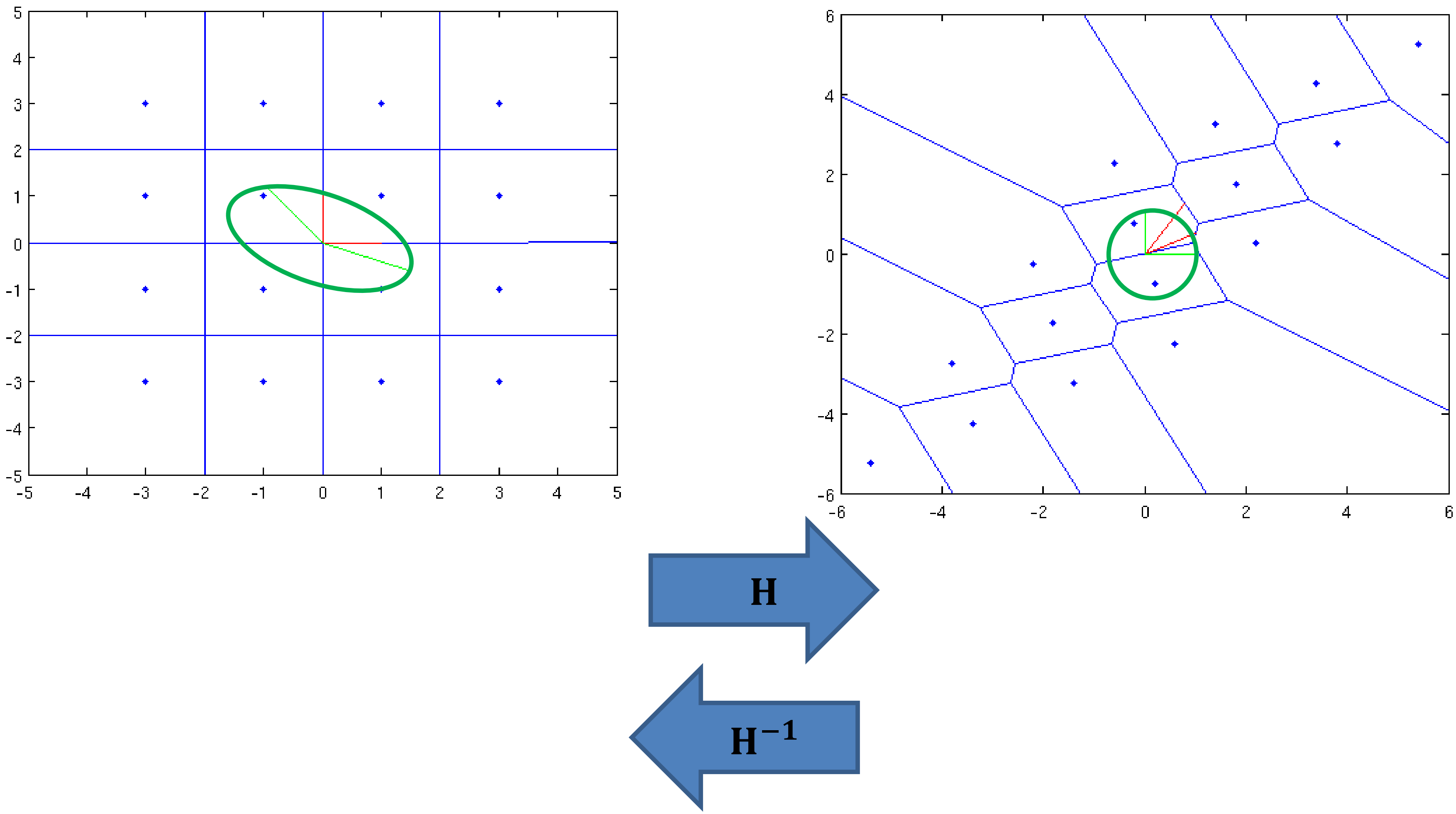
Consider again a 2x2 system with 4-PAM

- Nulling the interference of $\mathbf{h}_2 s_2$ corresponds to projecting the signal onto the subspace that is orthogonal to $\mathbf{h}_2$
- Only a small part of the energy of $\mathbf{h}_1 s_1$ lies in that subspace (unless $\mathbf{h}_1 \perp \mathbf{h}_2$)
- SNR is degraded since the noise power remain the same as along $\mathbf{h}_1$



Graphical Interpretation

Basis transformation and transformation of the noise



MMSE Receiver

- Zero interference comes at the cost of *noise enhancement*

Basic idea: balance the distortion due to noise and due to residual interference

- MMSE filter design criterion: determine the linear filter to minimize the mean squared error

$$\mathbf{W} = \arg\min_{\widetilde{\mathbf{W}}} \mathbb{E}\left\{\left\|\widetilde{\mathbf{W}}\mathbf{y} - \mathbf{s}\right\|^{\mathbf{2}}\right\}$$

- We obtain $\mathbf{W} = \boldsymbol{\Phi}_{\boldsymbol{ys}}\boldsymbol{\Phi}_{yy}^{-1}$ with $\boldsymbol{\Phi}_{\boldsymbol{ys}} = \frac{P_S}{N_T}\mathbf{H}^{\mathrm{H}}$ and $\boldsymbol{\Phi}_{\boldsymbol{yy}} = \left(\mathbf{H}\mathbf{H}^{\mathrm{H}} + \sigma^2\mathbf{I}\right)$

$$\mathbf{W} = \mathbf{H}^{\mathrm{H}}\left(\mathbf{H}\mathbf{H}^{\mathrm{H}} + N_T\frac{\sigma^2}{P_S}\mathbf{I}\right)^{-\mathbf{1}} = \left(\mathbf{H}^{\mathrm{H}}\mathbf{H} + N_T\frac{\sigma^2}{P_S}\mathbf{I}\right)^{-\mathbf{1}}\mathbf{H}^{\mathrm{H}}$$

A Note on Complexity

Since $N_R \geq N_T$ we know that $\mathbf{H}$ is either square of tall

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ h_{31} & h_{32} \end{bmatrix}$$

Option 1: $\mathbf{H}^{\mathrm{H}}\left(\mathbf{H}\mathbf{H}^{\mathrm{H}} + N_T \frac{\sigma^2}{P_S}\mathbf{I}\right)^{-\mathbf{1}}$ involves

- $\mathbf{Z} = \mathbf{H}\mathbf{H}^{\mathrm{H}} + N_T \frac{\sigma^2}{P_S}\mathbf{I}$: $N_R^2 N_T$ multiplications
- $inv(\mathbf{Z})$: Inversion of a $N_R \times N_R$ matrix (ill conditioned for $N_T \frac{\sigma^2}{P_S}$ small)
- $\mathbf{H}^{\mathrm{H}}\mathbf{Z}$: $N_T^2 N_R$ multiplications

Option 2: $\left(\mathbf{H}^{\mathrm{H}}\mathbf{H} + N_T \frac{\sigma^2}{P_S}\mathbf{I}\right)^{-\mathbf{1}}\mathbf{H}^{\mathrm{H}}$ involves

- $\mathbf{Z} = \mathbf{H}^{\mathrm{H}}\mathbf{H} + N_T \frac{\sigma^2}{P_S}\mathbf{I}$: $N_T^2 N_R$ multiplications
- $inv(\mathbf{Z})$: Inversion of a $N_T \times N_T$ matrix
- $\mathbf{Z}\mathbf{H}^{\mathrm{H}}$: $N_T^2 N_R$ multiplications

Computationally less complex and numerically more stable for high SNR

Relationship Between Estimators

$$\mathbf{W} = \left(\mathbf{H}^{\mathrm{H}}\mathbf{H} + N_T \frac{\sigma^2}{P_S}\mathbf{I}\right)^{-\mathbf{1}} \mathbf{H}^{\mathrm{H}}$$

For low-SNR ($\frac{\sigma^2}{P_S} \to \infty$)

- Additive noise is the dominant source of distortion

$$\mathbf{W} \to \left(N_T \frac{\sigma^2}{P_S}\mathbf{I}\right)^{-\mathbf{1}} \mathbf{H}^{\mathrm{H}}$$

- *MMSE converges to a matched filter*

For high-SNR ($\frac{\sigma^2}{P_S} \to 0$)

- Interference is the dominant source of distortion

$$\mathbf{W} \to \left(\mathbf{H}^{\mathrm{H}}\mathbf{H}\right)^{-\mathbf{1}} \mathbf{H}^{\mathrm{H}}$$

- *MMSE converges to the zero-forcing filter*

Unbiased MMSE Receiver

- Consider $\mathbf{W} = \left(\mathbf{H}^{\mathrm{H}}\mathbf{H} + N_T \frac{\sigma^2}{P_S}\mathbf{I}\right)^{-\mathbf{1}} \mathbf{H}^{\mathrm{H}}$ and apply to the received vector

$$\tilde{\mathbf{s}} = \mathbf{W}\mathbf{y} = \underbrace{\mathbf{W}\mathbf{H}}_{\boldsymbol{R}}\,\mathbf{s} + \underbrace{\mathbf{W}\mathbf{n}}_{\tilde{\mathbf{n}}}$$

$$\tilde{s}_k = r_{kk} s_{\boldsymbol{k}} + \sum_{\boldsymbol{i} \neq \boldsymbol{k}} r_{ki} s_{\boldsymbol{i}} + \tilde{s}_k$$

- Note that
  $\mathbf{R} = \mathbf{W}\mathbf{H} \neq \mathbf{I}$ ➔ Residual interference (off-diagonal elements)
  $r_{kk} < 1$ ➔ Desired stream is attenuated (MMSE is a biased estimator)

- Biased MMSE estimate provides the lowest MSE, but not necessarily the best BER
  - Example: In the low SNR regime, σ^2 is large and $r_{kk} \ll 1$. Hence, constellations shrink
- Multiplying $\tilde{s}_k$ with a scalar does not change the SNR

Bias can be removed by slicing a scaled version of $\tilde{s}_k$: $Q(\tilde{s}_k / r_{kk})$

Summary of Linear Receivers

Three step process:

1. Compute a linear filter from the channel estimate $\mathbf{H}$

$$\mathbf{G} = \left(\mathbf{H}^{\mathrm{H}}\mathbf{H}\right)^{-\mathbf{1}}\mathbf{H}^{\mathrm{H}} \qquad \textbf{(ZF)}$$

$$\mathbf{G} = \left(\mathbf{H}^{\mathrm{H}}\mathbf{H} + N_T\frac{\sigma^2}{P_S}\mathbf{I}\right)^{-\mathbf{1}}\mathbf{H}^{\mathrm{H}} \qquad \textbf{(MMSE - Biased)}$$

2. Apply linear filter to the received vector $\mathbf{y}$

$$\tilde{\mathbf{s}} = \mathbf{W}\mathbf{y}$$

3. Quantize to the nearest constellation point

$$\hat{s}_k = Q(\tilde{\mathrm{s}}_k)$$

Linear ZF Detection with QR Decomposition

QR-decomposition of the channel matrix

$$\mathbf{H} = \mathbf{QR}$$

- $\mathbf{Q} = [\boldsymbol{q}_1 \quad \cdots \quad \boldsymbol{q}_{N_T}] \in \mathcal{C}^{N_R \times N_R}$: unitary matrix, i.e., $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$
- $\mathbf{R} = \begin{bmatrix} r_{11} & \cdots & r_{1N_T} \\ \mathbf{0} & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & r_{N_T N_T} \end{bmatrix} \in \mathcal{C}^{N_T \times N_T}$: upper triangular
- MATLAB: `[Q,R]=qr(H)` yields the QR-decomposition of **H**

Some properties:

- The ith column of $\mathbf{H}$ is a linear combination of $[\boldsymbol{q}_1 \quad \cdots \quad \boldsymbol{q}_i]$

$$\mathbf{h}_i = \sum_{k=i}^{N_T} r_{ki}\boldsymbol{q}_i$$

- $\boldsymbol{q}_i$ contributes only to $[\mathbf{h}_i \quad \cdots \quad \mathbf{h}_{N_T}]$, but NOT to $[\mathbf{h}_1 \quad \cdots \quad \mathbf{h}_{i-1}]$ ➔ $\boldsymbol{q}_i \perp \mathbf{h}_j$, j<i

Linear ZF Detection with QR Decomposition

For example:

- $\boldsymbol{q}_{N_T}$ is orthogonal to $\mathbf{h}_1 \cdots \boldsymbol{h}_{N_T-1}$
- $\boldsymbol{q}_{N_T-1}$ is orthogonal to $\mathbf{h}_1 \cdots \boldsymbol{h}_{N_{T-2}}$
- and so on…

Basis transform: project the received vector onto the columns of $\mathbf{Q}$

$$\tilde{\mathbf{y}} = \mathbf{Q}^{\mathrm{H}}\mathbf{y} = \mathbf{Q}^{\mathrm{H}}\mathbf{Q}\mathbf{R}\boldsymbol{s} + \underbrace{\mathbf{Q}^{\mathrm{H}}\mathbf{n}}_{\tilde{\boldsymbol{n}}} = \mathbf{R}\boldsymbol{s} + \tilde{\boldsymbol{n}}$$

$$\begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \vdots \\ \tilde{y}_{N_T} \end{bmatrix} = \begin{bmatrix} r_{11} & \cdots & \cdots & r_{1N_T} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{N_T N_T} \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ \vdots \\ s_{N_T} \end{bmatrix} + \begin{bmatrix} \tilde{n}_1 \\ \vdots \\ \vdots \\ \tilde{n}_{N_T} \end{bmatrix}$$

- $\tilde{y}_i$: projection of the received signal onto $\boldsymbol{q}_{N_T}$ ➔ only contributions from $s_i \cdots s_{N_T}$
- $\tilde{\boldsymbol{n}}$: i.i.d. Gaussian with same properties as $\mathbf{n}$ ($\mathbf{Q}^{\mathrm{H}}$ is unitary)

Linear ZF Detection with QR Decomposition

$$\begin{bmatrix} r_{11} & \cdots & \cdots & r_{1N_T} \\ 0 & \ddots & & \vdots \\ & & \ddots & \vdots \\ 0 & & 0 & r_{N_T N_T} \end{bmatrix} \begin{bmatrix} \hat{s}_1 \\ \vdots \\ \vdots \\ \hat{s}_{N_T} \end{bmatrix} = \begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \vdots \\ \tilde{y}_{N_T} \end{bmatrix}$$

ZF detection: solve $\mathbf{R}\tilde{s} = \tilde{\mathbf{y}}$ with back substitution

- Start with the last stream and proceed backward

- For $i = N_T$ downto 1
  - Scaling : $\tilde{s}_i = \tilde{y}_i / r_{ii}$
  - Backsubstitution : $\tilde{\mathbf{y}} \leftarrow \tilde{\mathbf{y}} - \boldsymbol{r}_i \tilde{s}_i$
- End
- Quantize $\hat{s}_i = \mathrm{Q}(\tilde{s}_i)$

$$\tilde{\mathbf{s}} = \mathbf{R}^{-1}\tilde{\mathbf{y}}$$

MMSE Receiver Based on QR Decomposition

- We know that solving $\mathbf{R}\tilde{\boldsymbol{s}} = \mathbf{Q}^H\mathbf{y}$ in a LS sense with $\mathbf{H} = \mathbf{QR}$ through backsubstitution corresponds to the solution of $\tilde{\boldsymbol{s}} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\mathbf{y}$

We are interested in $\tilde{\boldsymbol{s}} = \left(\mathbf{H}^{\mathrm{H}}\mathbf{H} + N_T\frac{\sigma^2}{P_S}\mathbf{I}\right)^{-\mathbf{1}}\mathbf{H}^{\mathrm{H}}\boldsymbol{y}$

- Introduce the augmented channel matrix $\overline{\boldsymbol{H}} = \begin{bmatrix}\mathbf{H}\\ \sqrt{N_T\frac{\sigma^2}{P_S}}\mathbf{I}\end{bmatrix}$ and $\bar{\mathbf{y}} = \begin{bmatrix}\mathbf{y}\\ \mathbf{0}\end{bmatrix}$
- Note that $\mathbf{H}^{\mathrm{H}}\mathbf{H} + N_T\frac{\sigma^2}{P_S}\mathbf{I} = \overline{\boldsymbol{H}}^{\mathrm{H}}\overline{\boldsymbol{H}}$ and $\mathbf{H}^{\mathrm{H}}\boldsymbol{y} = \overline{\boldsymbol{H}}^{\mathrm{H}}\bar{\mathbf{y}}$
- We can therefore write $\tilde{\boldsymbol{s}} = \left(\overline{\boldsymbol{H}}^{\mathrm{H}}\overline{\boldsymbol{H}}\right)^{-\mathbf{1}}\overline{\boldsymbol{H}}^{\mathrm{H}}\bar{\mathbf{y}}$
- Introduce the QR decomposition of $\overline{\boldsymbol{H}} = \begin{bmatrix}\overline{\mathbf{Q}_1} & \overline{\mathbf{Q}_3}\\ \overline{\mathbf{Q}_2} & \overline{\mathbf{Q}_4}\end{bmatrix}\begin{bmatrix}\overline{\mathbf{R}}\\ \mathbf{0}\end{bmatrix}$ and solve

$$\begin{bmatrix}\overline{\mathbf{R}}\\ \mathbf{0}\end{bmatrix}\tilde{\boldsymbol{s}} = \begin{bmatrix}\overline{\mathbf{Q}_1}^{\mathrm{H}} & \overline{\mathbf{Q}_2}^{\mathrm{H}}\\ \overline{\mathbf{Q}_3}^{\mathrm{H}} & \overline{\mathbf{Q}_4}^{\mathrm{H}}\end{bmatrix}\begin{bmatrix}\mathbf{y}\\ \mathbf{0}\end{bmatrix} \overset{LS}{\rightarrow} \overline{\mathbf{R}}\tilde{\boldsymbol{s}} = \overline{\mathbf{Q}_1}^{\mathrm{H}}\mathbf{y}$$

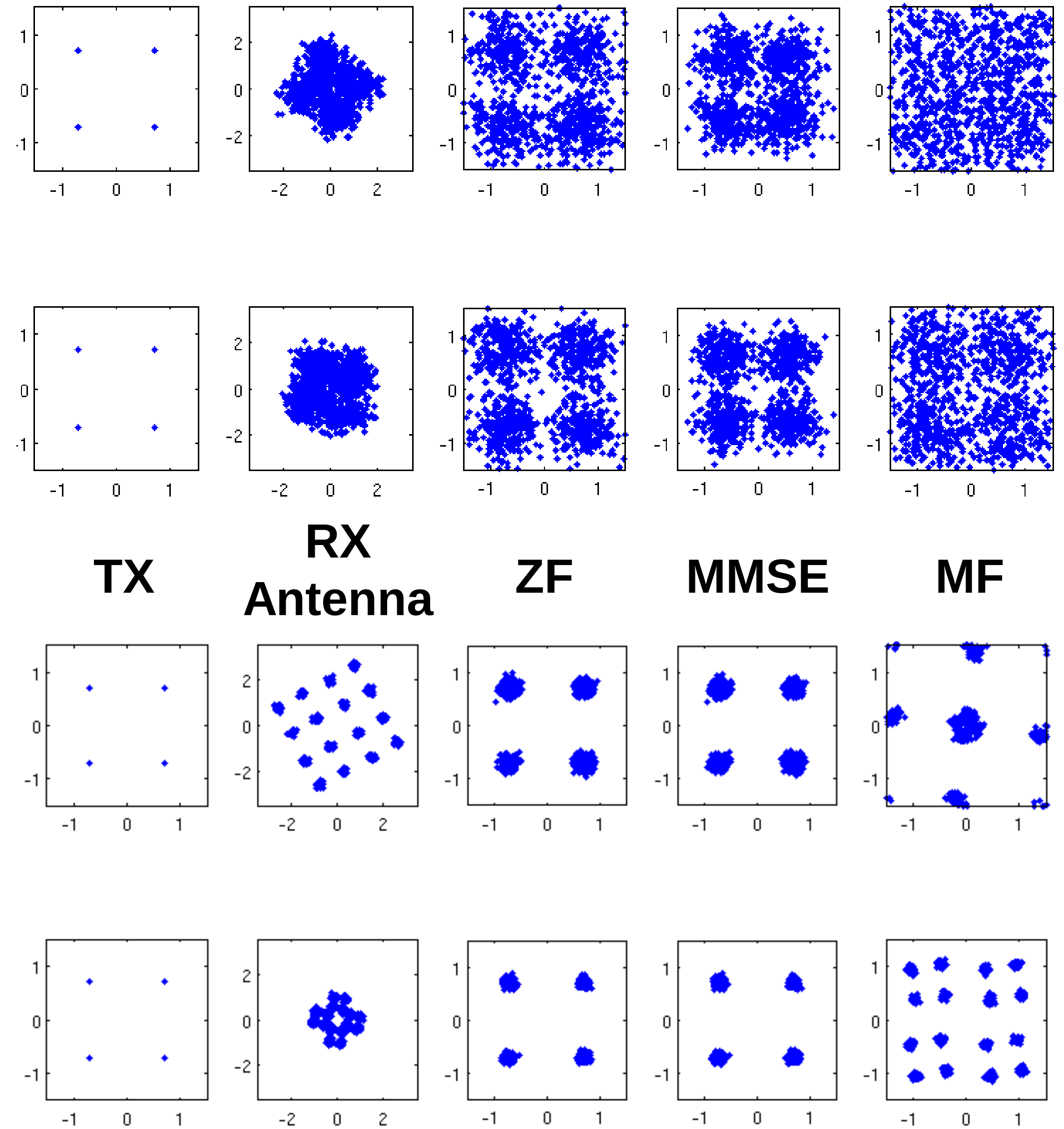
Illustrative Example

Low SNR: 6dB

- ZF: severe noise enhancement
- MMSE: good tradeoff between noise and interference, but signal is scaled (BIAS)
- MF: hard to distinguish between noise and interference

High SNR: 20dB

- ZF and MMSE perform almost equally well
- MF: Interference is clearly visible (no cancellation capability)



Successive Interference Cancellation (SIC)

Problem of linear detection:

- Detection of each stream requires nulling of the interference of $N_T - 1$ streams
- Since we always project into the null-space of $N_T - 1$ streams we loose all the energy in these dimensions (for all streams)

Interference cancellation

- Assume we already know $K - 1$ streams (signal components) s_k with $k = 1 \ldots K - 1$
- In this case, we can cancel the impact of s_k from the received vector $\mathbf{y}$

$$\mathbf{y}^{(K)} = \mathbf{y} - \sum_{k=1}^{K-1} \mathbf{h}_k s_k \qquad \boldsymbol{y} = \mathbf{Hs} + \mathbf{n} \rightarrow \mathbf{y}^{(K)} = \mathbf{H}^{(K)} \begin{bmatrix} s_K \\ \vdots \\ s_{N_T} \end{bmatrix} + \mathbf{n}$$

with $\mathbf{H}^{(K)} = [\mathbf{h}_K \quad \cdots \quad \mathbf{h}_{N_T}] \in \mathcal{C}^{N_R \times N_T - K + 1}$

After perfect interference cancellation of $K - 1$ streams we obtain a new system with $N_T - K + 1$ streams and N_R «receive antennas»

Successive Interference Cancellation (SIC)

Basic idea of SIC: perform the following two steps iteratively

- Linear detection on one stream only and assume the detected symbol is correct
- Perform interference cancellation with the detected symbol

SIC algorithm

Unit row vector

$$\boldsymbol{e}_1 = [1 \quad 0 \quad \cdots \quad 0]$$

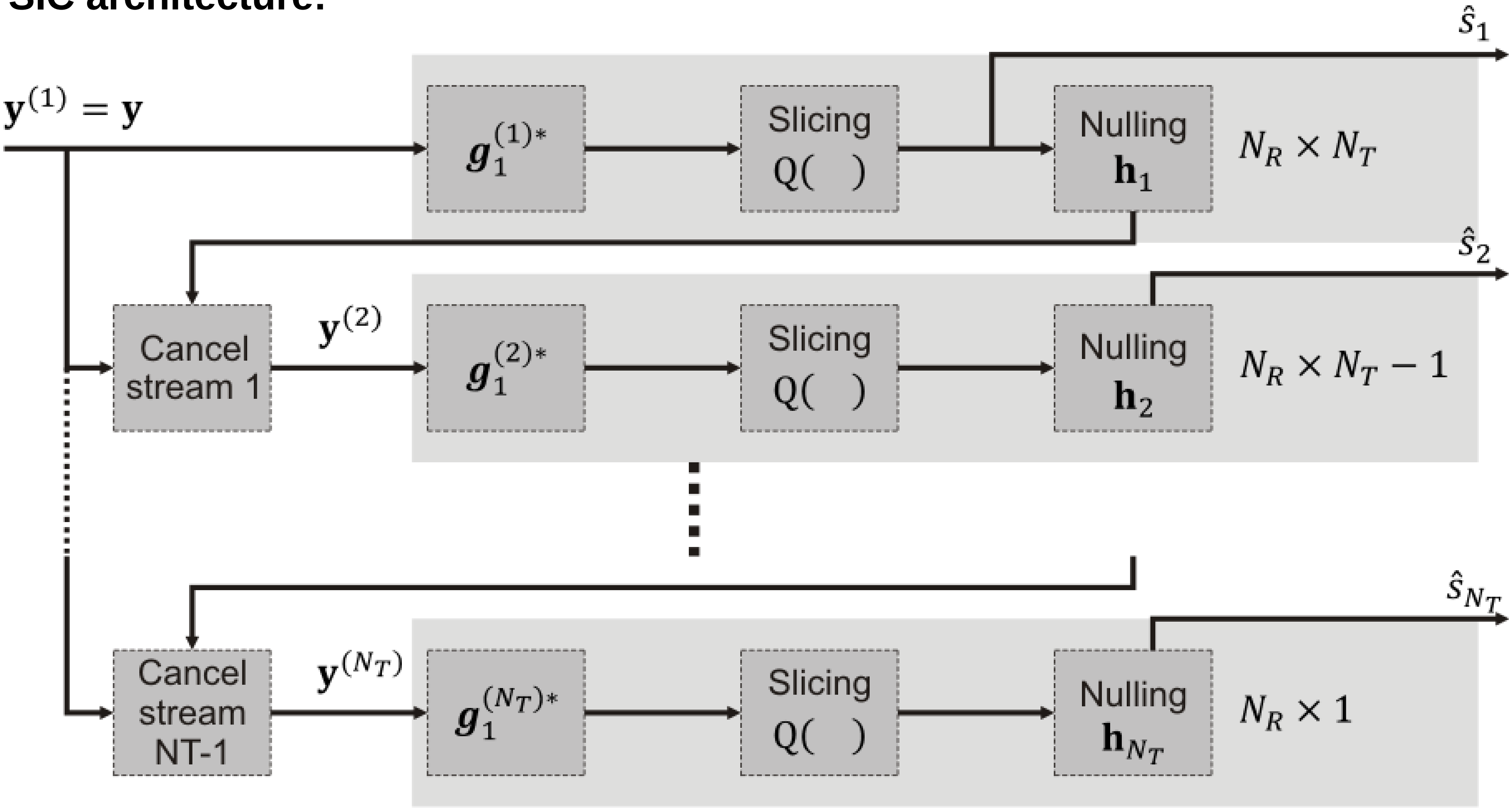
- Set $\mathbf{y}^{(1)} = \mathbf{y}$, $\mathbf{H}^{(1)} = \mathbf{H}$
- For $i = 1$ to N_T
  - Linear estimator for the first remaining stream: $\boldsymbol{g}_1^{(i)*} = \boldsymbol{e}_1 \mathbf{H}^{(i)\dagger}$ (first row of $\mathbf{H}^{(i)\dagger}$)
  - Symbol detection: $\hat{s}_i = \mathrm{Q}\left(\boldsymbol{g}_1^{(i)*} \mathbf{y}^{(i)}\right)$
  - ***Interference cancellation***: $\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} - \mathbf{h}_i \hat{s}_i$
  - Remove 1st column from $\mathbf{H}^{(i)} \to \mathbf{H}^{(i+1)}$ $\quad \mathbf{H}^{(i+1)} = \left[\mathbf{h}_2^{(i)} \quad \cdots \quad \mathbf{h}_{N_T}^{(i)}\right]$
- End

Note that

- $\mathbf{H}^{(i)} = [\mathbf{h}_i \quad \cdots \quad \mathbf{h}_{N_T}]$
- $\mathbf{y}^{(i)} = \mathbf{y} - \sum_{k=1}^{i-1} \mathbf{h}_k \hat{s}_k \approx \mathbf{y} - \sum_{k=1}^{i-1} \mathbf{h}_k s_k$

Successive Interference Cancellation (SIC)

SIC architecture:

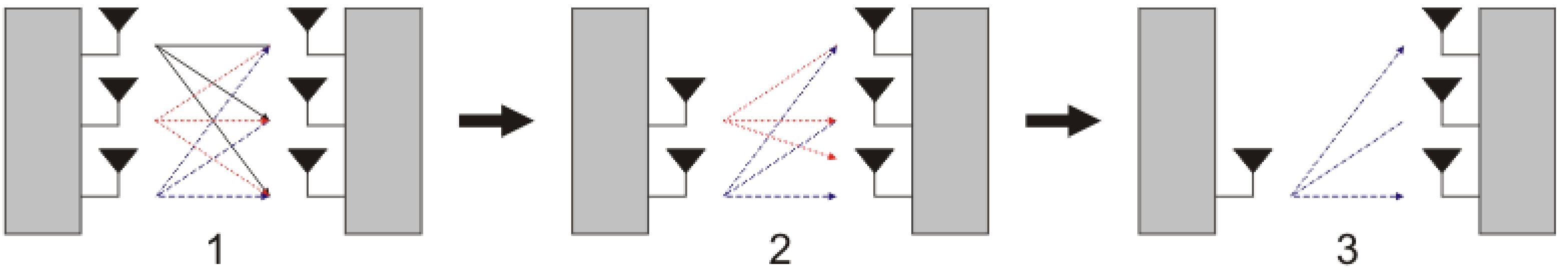


- Note: the number of interferers decreases with each stage

Performance of SIC

Observations:

- The numbers of interferers decreases with each stage, but the number of observations remains the same
- Intuitively, nulling becomes “easier” with each stream that we cancel
- Eventually, we arrive at a SIMO system with one stream and N_R observations



Unfortunately: performance gain is only marginal

- The error probability of the weakest stream determines the symbol-error probability
- Without “perfect” cancellation, error propagation limits the performance (damages/interfers with later detected streams)

Need to detect the most reliable streams first

Detection Order with SIC

- No particular reason forces us to perform detection and SIC in natural order
- In each stage of the SIC algorithm, we can choose which stream to detect
- In total there are $N_T!$ different orders for the SIC

- The detection order can be changed by
  - Permuting the columns of $\mathbf{H}$ and rearranging the order of the entries of the symbol-vector and accordingly

$$\mathbf{H} \leftarrow \mathbf{H}\boldsymbol{\Pi} \text{ and } \hat{\boldsymbol{s}} \leftarrow \boldsymbol{\Pi}^{\mathrm{H}}\hat{\boldsymbol{s}} \qquad \Rightarrow \qquad \mathbf{H}\boldsymbol{\Pi}\boldsymbol{\Pi}^{\mathrm{H}}\hat{\boldsymbol{s}} = \mathbf{H}\hat{\boldsymbol{s}}$$

  with the unitary permutation matrix $\boldsymbol{\Pi}$

  - Directly: detecting any stream and removing the corresponding column of $\mathbf{H}$

Ordered SIC

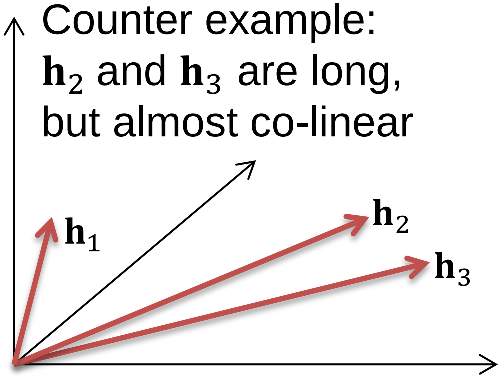
Objective: detect the more reliable streams as early as possible

Received Power based ordering

- Consider the contribution to the received power of each stream

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n} = \sum_{k=1}^{N_T} s_k\, \mathbf{h}_k + \mathbf{n}$$

Counter example:
$\mathbf{h}_2$ and $\mathbf{h}_3$ are long,
but almost co-linear



- Contribution of stream k to the received power $\|\mathbf{h}_k\|^2$

- **Column-norm ordering**: choose the detection order according to the norms of the columns of the channel $\|\mathbf{h}_k\|^2$

- **Unfortunately, the geometry of the channel is not accounted: If two or multiple streams are strong, but lie almost in the same subspace, separation is still difficult (noise enhancement)!**

V-BLAST Order

So, what is the right detection order?

- Ultimately, the quality of the decision depends on the SINR at the slicer

- Slicer input signal

$$\tilde{s}_k = \boldsymbol{g}_k^{(i)*}\mathbf{y}^{(i)} = s_k + \boldsymbol{g}_k^{(i)*}\mathbf{n}$$

$$\|\tilde{s}_k - s_k\|^2 = \left\|\boldsymbol{g}_k^{(i)*}\right\|^2 \|\mathbf{n}\|^2$$

- In each stage, the SINR is different for each stream

Easier to compute since $\boldsymbol{g}_k^{(i)*}$ is computed anyway, while $\left(\mathbf{H}^{(i)H}\mathbf{H}^{(i)}\right)^{-1}$ may not be explicitly available

$$SINR_k^{(i)} = \frac{P_S/N_T}{\sigma^2[(\mathbf{H}^{(i)H}\mathbf{H}^{(i)})^{-1}]_{kk}} \propto 1/\left\|\boldsymbol{g}_k^{(i)*}\right\|^2$$

- V-BLAST ordering: always select the stream with the best post-detection $SINR_k^{(i)}$, i.e., with the *smallest* equalizer gain $\left\|\boldsymbol{g}_k^{(i)*}\right\|^2$ (i.e., noise gain/enhancement)

V-BLAST Algorithm

Basic idea: in each stage, minimize the detection error probability by choosing the stream k with the best post-detection signal to interference plus noise ratio

$$k^{(i)} = \underset{\boldsymbol{k}}{\operatorname{argmin}} \left\{ SINR_k^{(i)} \right\}$$

V-BLAST algorithm

- Set $\mathbf{y}^{(1)} = \mathbf{y}$, $\mathbf{H}^{(1)} = \mathbf{H}$
- For $i = 1$ to N_T
  - Compute the linear estimator for all streams $\mathbf{G}^{(i)} = \mathbf{H}^{(i)\dagger}$ with $\boldsymbol{g}_k^{(i)*} = \boldsymbol{e}_k \mathbf{G}^{(i)}$: rows of $\mathbf{G}$
  - Find the row of $\mathbf{H}^{(i)\dagger}$ with the smallest norm $k^{(i)} = \underset{\boldsymbol{k}}{\operatorname{argmin}} \left\{ \left\| \boldsymbol{g}_k^{(i)*} \right\|^2 \right\}$
  - Symbol detection of the $k^{(i)}$th stream $\hat{s}_i = \mathrm{Q}\left(\boldsymbol{g}_{k^{(i)}}^{(i)*} \mathbf{y}^{(i)} \right)$
  - Interference cancellation: $\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} - \mathbf{h}_i \hat{s}_i$
  - Remove $k^{(i)}$th column from $\mathbf{H}^{(i)} \rightarrow \mathbf{H}^{(i+1)}$ $\mathbf{H}^{(i+1)} = \left[\mathbf{h}_1^{(i)} \cdots \mathbf{h}_{k^{(i)}-1}^{(i)} \mathbf{h}_{k^{(i)}+1}^{(i)} \cdots \mathbf{h}_{N_T}^{(i)} \right]$
- End loop
- Reorder $\hat{s}_i$ according to $k^{(i)}$

MMSE V-BLAST

MMSE V-BLAST follows the same strategy

- Linear detection in each stage based on MMSE filter

$$\mathbf{H}^{(i)\dagger} \rightarrow \mathbf{G}^{(i)} = \left(\mathbf{H}^{(\boldsymbol{i})H}\mathbf{H}^{(\boldsymbol{i})} + N_T \frac{\sigma^2}{P_S}\mathbf{I}\right)^{-\mathbf{1}} \mathbf{H}^{(\boldsymbol{i})H}$$

- Calculate the SINR for each stream with linear MMSE filter

$$MMSE - SNR_k^{(i)} = \frac{P_S/N_T}{\sigma^2[(\mathbf{H}^{(\boldsymbol{i})H}\mathbf{H}^{(\boldsymbol{i})} + \mathbf{I}\sigma^2 N_T/P_S)^{-1}]_{kk}}$$

  and select the best stream for detection

- Detect the symbol on the selected stream

- Remove the impact of the detected stream from the received vector, the corresponding channel vector from the channel matrix

- Start from the beginning with the reduced system

Efficient SIC with QR Decomposition

- Drawback of SIC: each of the N_T stages requires a matrix inversion to compute the linear estimator
- More efficient implementation exists based on QR decomposition

QR-decomposition based SIC

- Given $\mathbf{H} = \mathbf{QR}$ of the channel
- Since $\mathbf{Q}$ is unitary, we can always consider $\tilde{\mathbf{y}} = \mathbf{Q}^{\mathrm{H}}\mathbf{y} = \mathbf{Rs} + \tilde{\boldsymbol{n}}$
- Start from the last stream and work backward to the first performing backsubstitution using quantized symbols

$$\begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \tilde{y}_3 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} + \begin{bmatrix} \tilde{n}_1 \\ \tilde{n}_2 \\ \tilde{n}_3 \end{bmatrix}$$

- For $i = N_T$ downto 1
  - $\hat{s}_i = \mathrm{Q}(\tilde{y}_i / r_{ii})$
  - $\tilde{\mathbf{y}} \leftarrow \tilde{\mathbf{y}} - \mathbf{r}_i \hat{s}_i$
- End

$$\begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \tilde{y}_3 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} \tilde{n}_1 \\ \tilde{n}_2 \\ \tilde{n}_3 \end{bmatrix}$$

$$\begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \tilde{y}_3 \end{bmatrix} = \begin{bmatrix} r_{11} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} s_1 \end{bmatrix} + \begin{bmatrix} \tilde{n}_1 \\ \tilde{n}_2 \\ \tilde{n}_3 \end{bmatrix}$$

Ordered SIC based on QR Decomposition

Problem: Backsubstitution determines detection order

- Nevertheless, we can start by reordering the channel matrix prior to QR decomposition
- **Unfortunately**, at this stage, the post-detection SNR is unknown!

Goal: find an alternative ordering strategy/criterion

- Consider the SINR before the slicer when detection is performed with backsubstitution

$$\tilde{s}_i = \frac{\tilde{y}_i}{r_{ii}} = s_i + \tilde{n}_i / r_{ii} \quad \Rightarrow \quad |\tilde{s}_i - s_i|^2 = |n_i|^2 / |r_{ii}|^2$$

- Large diagonal elements of $\mathbf{R}$ result in better SNR after the slicer

- Ordering strategy: Reorder the streams such that $r_{ii} > r_{kk}$ for $i > k$

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

RWTH AACHEN UNIVERSITY

QR Decomposition

Objective

- Derive an orthonormal basis $\mathbf{Q}$ for $\mathbf{H}$ such that $\boldsymbol{q}_i \perp \mathbf{h}_j$, j<l

Gram Schmidt algorithm

- Set $\mathbf{Q} = \mathbf{H}$ will update $\boldsymbol{Q}$ later
- For $i = 1$ to N_T
  - $\boldsymbol{p} = \boldsymbol{q}_i$
  - For $k = i + 1$ to N_T
    - $r_{ik} = \boldsymbol{q}_k^H \boldsymbol{p}$
    - $\boldsymbol{q}_k = \boldsymbol{q}_k - r_{ik}\boldsymbol{q}_i$
  - End
  - $r_{ii} = ||\boldsymbol{p}||$
  - $\boldsymbol{q_i} = \boldsymbol{p}/r_{ii}$
- End

Orthogonalize $\mathbf{q}_{i+1} \quad \cdots \quad \mathbf{q}_{N_T}$

$$\boldsymbol{q}_k = \boldsymbol{q}_k - \sum_{k=1}^{i-1} \mathbf{q}_i \langle \mathbf{q}_k, \mathbf{q}_i \rangle$$

Normalize

Sorted QR Decomposition

- Unfortunately, r_{ii} are not known prior to performing QR decomposition

Solution: perform ordering within the QR decomposition (*Sorted QR decomposition*) [Wuebben, 2001]

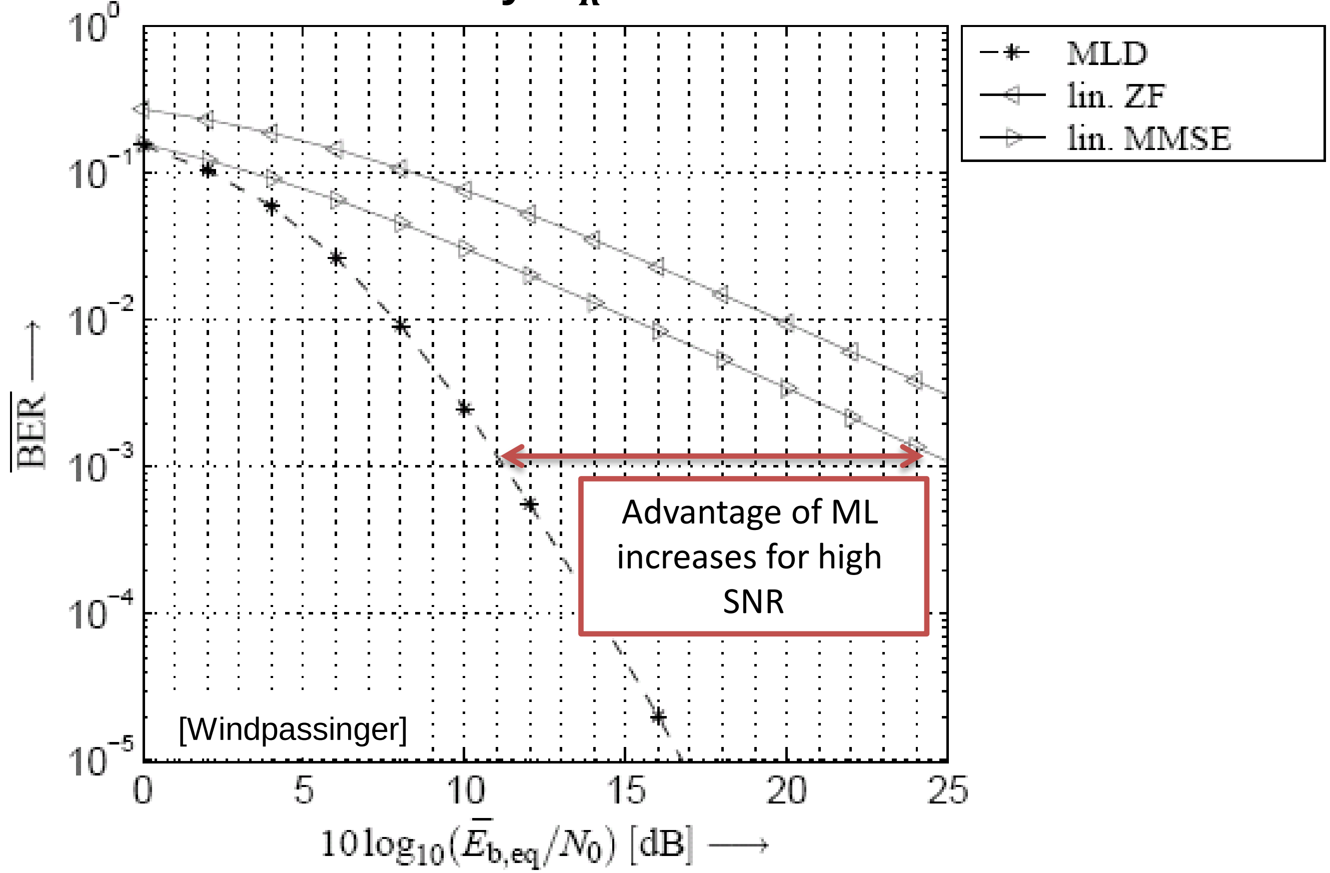
- Set $\mathbf{Q} = \mathbf{H}$
- For $i = 1$ to N_T
  - Among $\boldsymbol{q}_i \quad \cdots \quad \boldsymbol{q}_{N_T}$ find the one with the smallest norm ➔ m
  - Exchange $\boldsymbol{q}_i$ with $\boldsymbol{q}_m$
  - $\boldsymbol{p} = \boldsymbol{q}_i$
  - $r_{ii} = \|\boldsymbol{p}\|$
  - For $k = i + 1$ to N_T
    - $r_{ik} = \boldsymbol{q}_k^H \boldsymbol{p}$
    - $\boldsymbol{q}_k = \boldsymbol{q}_k - r_{ik}\boldsymbol{q}_i$
  - End
  - $\boldsymbol{q}_i = \boldsymbol{p}/r_{ii}$
- End

Greedy strategy: in each step choose among the remaining columns the one that yields the smallest r_{ii}
➔ r_{ii} in (approximately!) increasing order

Comparison of Basic MIMO Receivers with Fading for $N_T = N_R$

Observations

- Linear (ZF and MMSE) detection has diversity order of 1
- MMSE provides a gain in SNR (shift) over ZF
- **ML detection achieves full diversity N_R**



Comparison of Basic MIMO Receivers with Fading

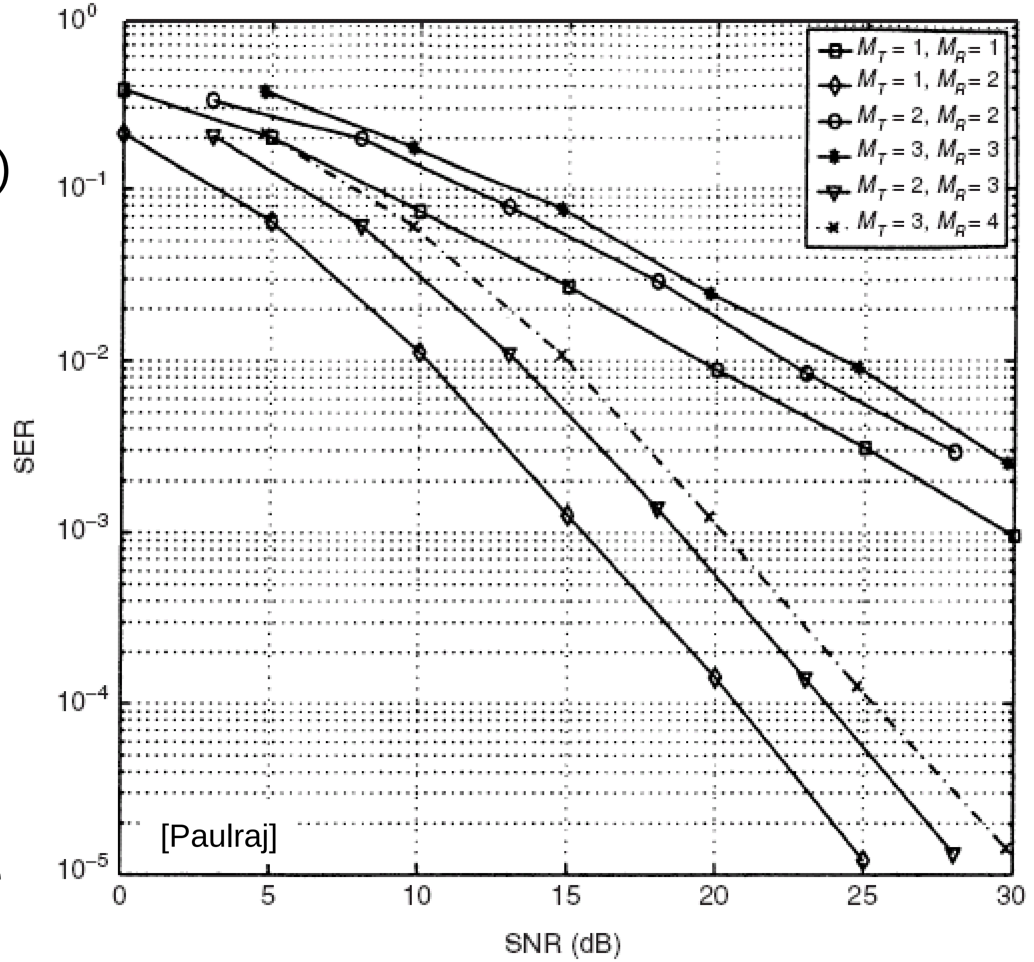
Observations: Increasing the number of receive antennas

- Improves diversity
- Provides a coding gain (SNR shift)

Diversity of linear detectors

$$N_R - N_T + 1$$

- The detailed proof is complicated



Intuition:

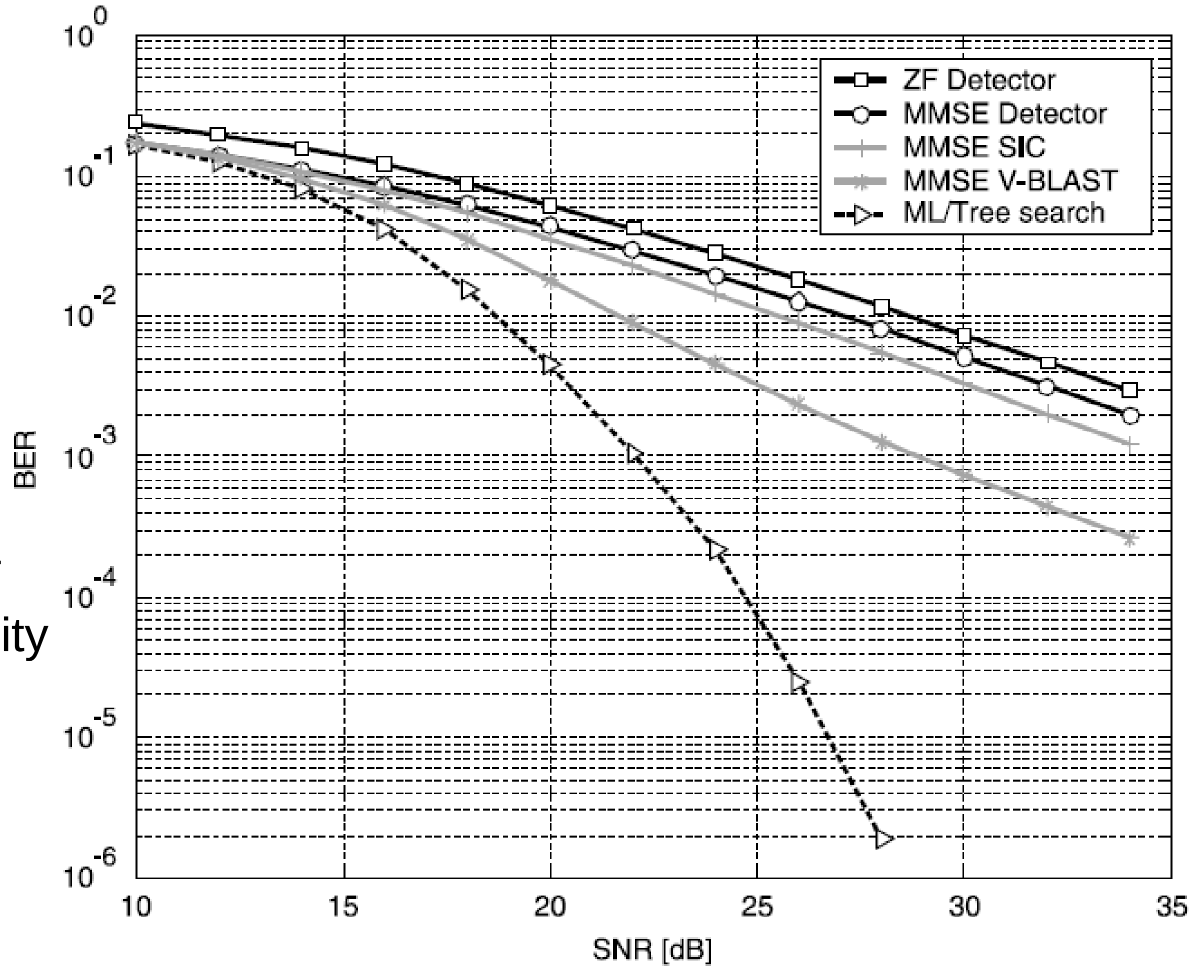
- N_R dimensional received vector $\mathbf{y}$
- Interference space: $N_T - 1$ dim.
  $\left[\mathbf{h}_1^{(i)} \cdots \mathbf{h}_{k^{(i)}-1}^{(i)} \mathbf{h}_{k^{(i)}+1}^{(i)} \cdots \mathbf{h}_{N_T}^{(i)}\right]$
- Project $\mathbf{y}$ into the null-space of the interference: $N_R + (N_T - 1)$ dim.
- Decision statistics has $N_R - N_T + 1$ degrees of freedom

Performance of SIC

- Symmetric 4x4 system

Observations:

- SIC yields an improvement over linear detection
- The real gain of SIC is only visible with ordering
- Unfortunetely, even V-BLAST achieves only 1st order diversity



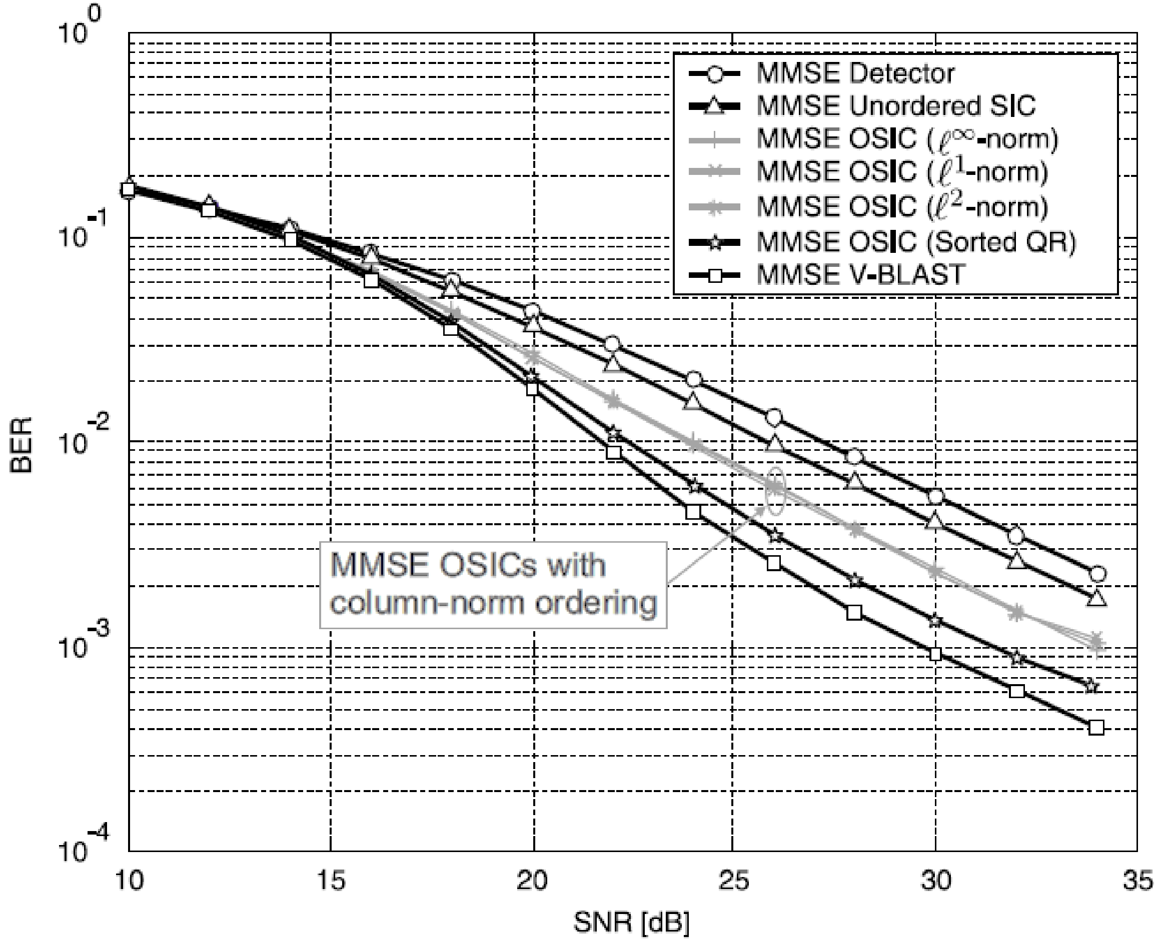
Performance of SIC with Sorted QR

Observation

- Sorted QR performs very close to V-BLAST, but is NOT the same

Note

- There is also a way to do optimum V-BLAST ordering with QR decomposition



Performance of SIC

The full performance analysis of SIC is difficult, though possible, but some intuitive arguments help to understand the behavior

- Given: N_R dimensional received vector $\mathbf{y}$

- Detecting the first stream requires projection onto the null-space of the remaining $N_R - 1$ interfering streams
- The corresponding projection is an $N_R - N_T + 1$ dimensional decision statistics

➢ **Diversity order of the first stream**: $N_R - N_T + 1$

- After cancellation of streams $1 - i$ the received vector $\mathbf{y}^{(i)}$ still has N_R dimensions
- But, (assuming no errors) only $N_T - i$ interferers are left ➔ null-space of the interference has only $N_T - i$ dimensions.

➢ **Diversity order of the ith stream**: $N_R - N_T + i$

➢ The subsequently detected streams have diversity order $N_R - N_T + i$

➢ Overall diversity is determined by the first stream. Hence still $N_R - N_T + 1$, BUT there is some performance improvement with appropriate ordering

Architecture for MIMO Detection

MIMO detection algorithms contain two types of operations

- Operations that depend only on the channel matrix (channel estimate)
- Operations that depend on the received vector

- Normally the channel (channel estimate) remain constant for many received vectors (keep training overhead low)

Receiver structure

- Preprocessing: pre-compute and store results that depend only on the channel
- Detection: processing for each individual received vector