Linear Algebra Issues in Wireless Communications

Vladimir Lyashev

Moscow Research Center
[vladimir.lyashev@huawei.com]

Praesentationis et contentus

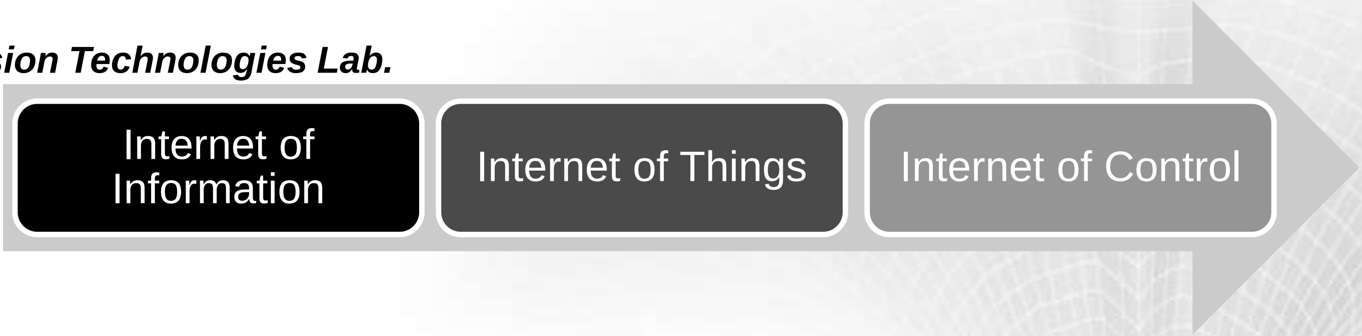
Moscow:

- Introduction & Definitions (#1)
- Matrix Algebra in Wireless (#2)
- Home task (#2)

Rome:

- Parse your homework (#3)
- MMSE criteria in wireless (#3)
- Math. & resource allocation (#4)

Head of *Radio Transmission Technologies Lab.*
at Huawei Technologies,
Moscow Research Center
Senior IEEE Member
https://lyashev.weebly.com



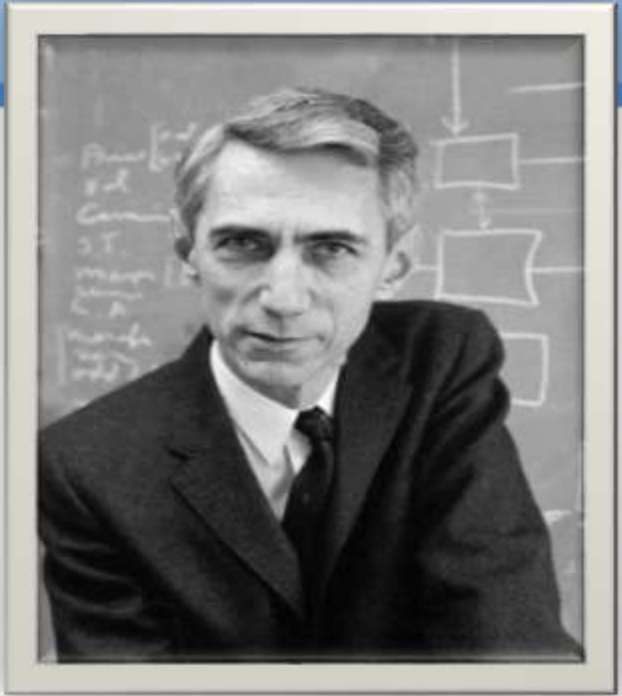
A History

[1948] Groundbreaking article by Claude E. Shannon

“**A mathematical theory of communication**”

introducing the definition of capacity
in communication systems:

The channel capacity is a measurement of the maximum amount of information that can be transmitted over a channel and received with a negligible probability of error at the receiver.

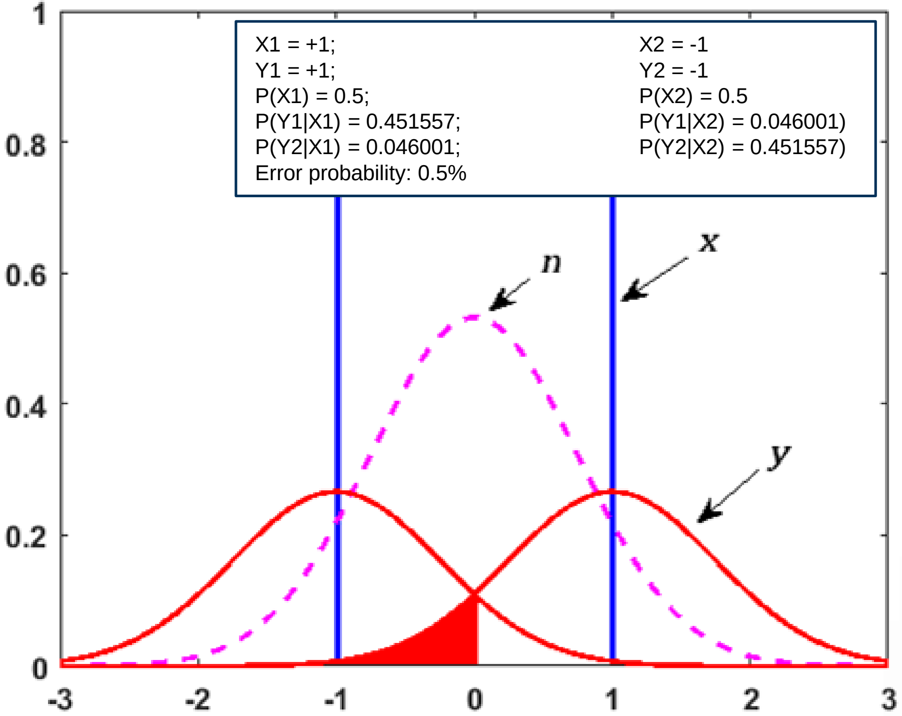


“Information is the resolution of uncertainty.”
- Claude Elwood Shannon
Father of Information Theory

$$C = B_f \log_2 (1 + \mathrm{SNR})$$

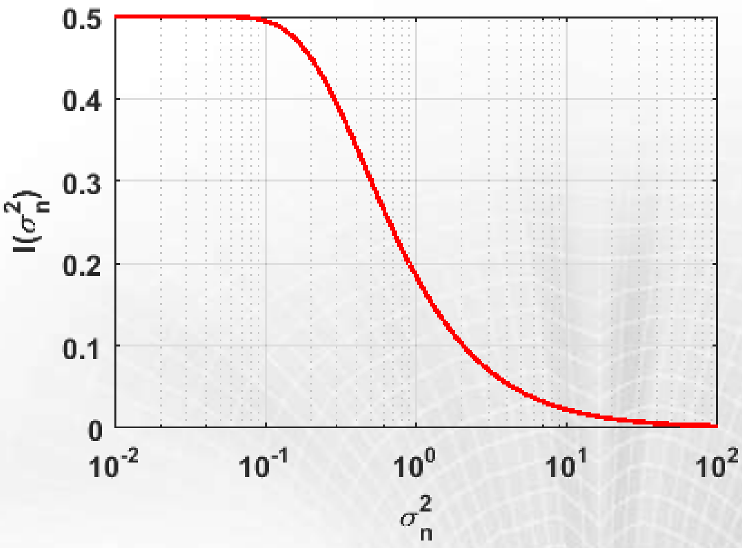
PDF: additive noise

$$y = x + n$$



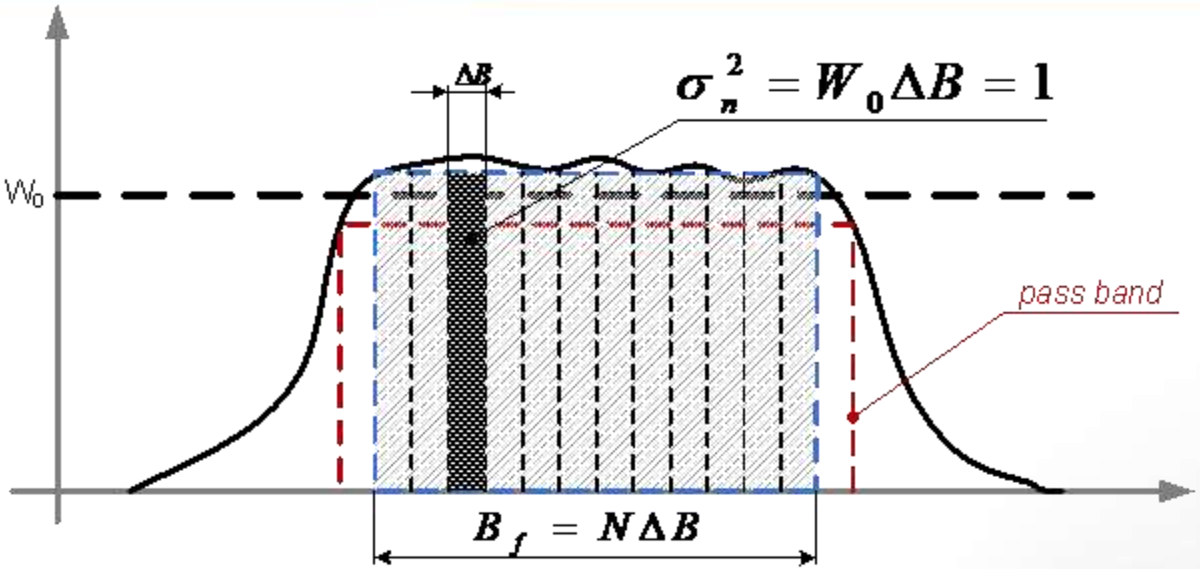
$$H(Y) = -\sum_n P(Y_n)\log_2 P(Y_n)$$

$$H(Y \mid X) = -\sum_k P(X_k)\sum_n P(Y_n \mid X_k)\log_2 P(Y_n \mid X_k)$$



Vladimir Lyashev

Signal-to-Noise Ratio



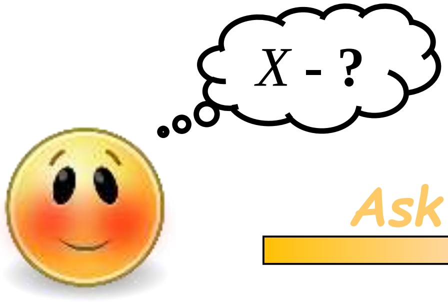
$$y = x + n$$

$$P_y = yy^H = (x+n)(x+n)^H = xx^H + \underbrace{xn^H}_{0} + \underbrace{nx^H}_{0} + nn^H = P_x + \sigma_n^2;$$

$$P_y = \sigma_n^2\left(1 + \frac{P_x}{\sigma_n^2}\right) \Rightarrow N\Delta B \log_2\left(P_y\right) = N\Delta B \log_2\left(\sigma_n^2\left(1 + \frac{P_x}{\sigma_n^2}\right)\right) = N\Delta B \log_2\left(1 + \frac{P_x}{\sigma_n^2}\right)$$

Detection problem

$$X \in [1, 2, \ldots, 32]$$



Search constraint: answer can be YES/NO

Search strategy: bisection method or interval halving or binary search or dichotomy method.

Binary channel

5 answers YES/NO ⟺ 5 bits of information

$$I = \log_2 M = \log_2\left(1 + \frac{P}{\sigma^2}\right)$$

Hartley's law: quantity of information *M*, which is necessary for detection the specific value, is the base-2-logarithm of the number of distinct messages *M* that could be sent.

$$C_{SISO} = B_f \log_2\left(1 + \frac{P_{TX}\|\mathbf{H}\|}{\sigma_n^2}\right) = B_f \log_2\left(1 + \frac{P_{TX}\lambda_1^2}{\sigma_n^2}\right)$$

Complex channel model

Ideal complex channel model

Consider two antennas A and B. Tx-signal is x; Rx-signal is y

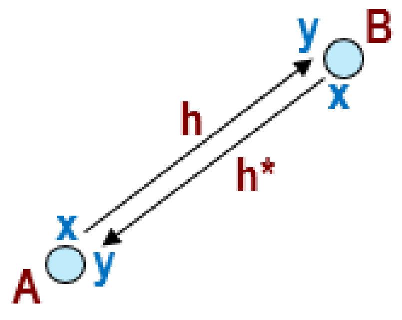
Channel $(Tx = A) \to (Rx = B)$: $\quad y = hx\,;\; h = \sqrt{g}e^{i\varphi}$

Channel $(Tx = B) \to (Rx = A)$: $\quad y = h^{*}x\,;\; h^{*} = \sqrt{g}e^{-i\varphi}$

Transmitted and received power are connected as follows

$$p(y) = y^{*}y = h^{*}h \cdot x^{*}x = h^{*}h \cdot p(x) = g \cdot p(x)$$

g is power loss on the path Tx → Rx
φ is phase shift on the path Tx → Rx



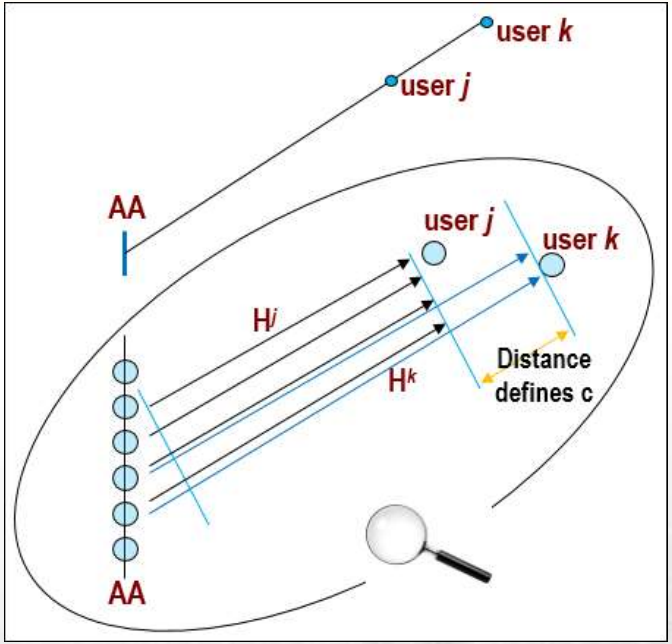
$$I = \log_2\left(1 + \frac{g \cdot P}{\sigma^2}\right)$$

Basic LOS MIMO channel

MIMO channel model

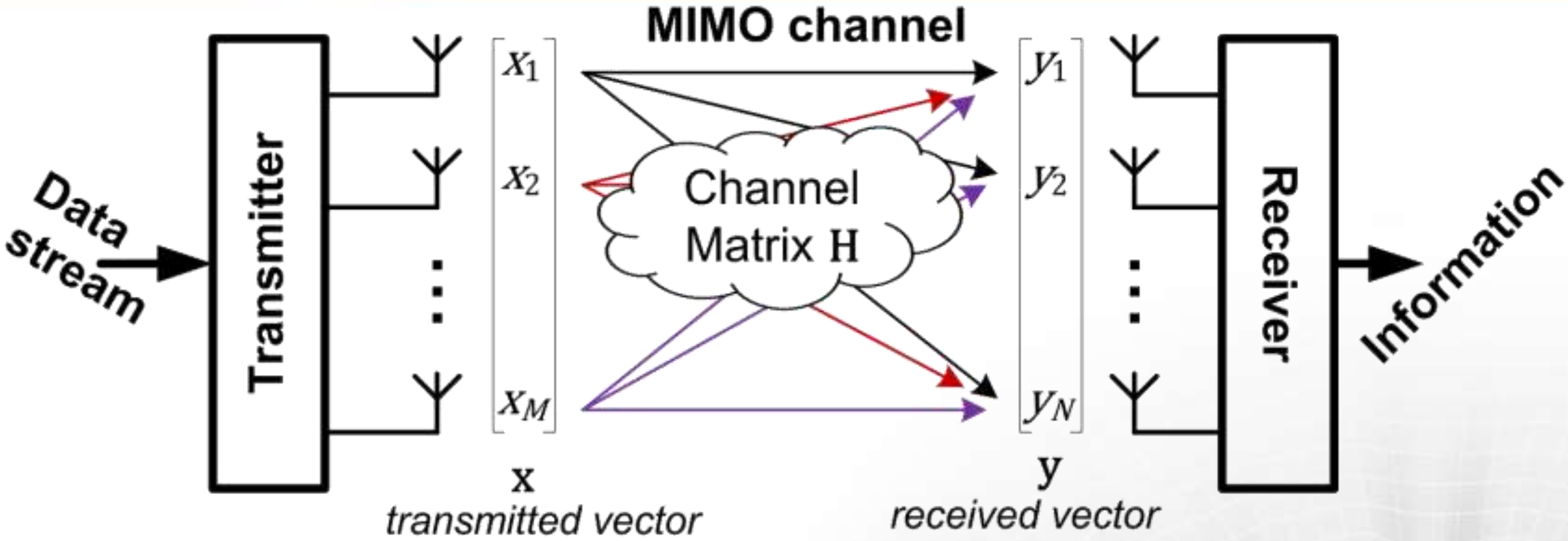
N_X Tx-antennas (antenna array AA on BS side) → one Rx-antenna (on user side)

	General model	Simplified model (AA constructive size << distance AA→user)
Channel vector AA→user j	$H^j = (h_1^j, h_2^j \dots h_{N_X}^j)^T$ $h_q^j = \lvert h_q^j \rvert e^{i\varphi_q^j}$; $q = 1,2 \dots N_X$ $\lvert H \rvert^j = (\lvert h_1^j \rvert, \lvert h_2^j \rvert \dots \lvert h_{N_X}^j \rvert)^T$ $\Phi^j = (\varphi_1^j, \varphi_2^j \dots \varphi_{N_X}^j)^T$	$\lvert h_q^j \rvert = \lvert h^j \rvert$; $q = 1,2 \dots N_X$ $\lvert H \rvert^j = \lvert h^j \rvert \cdot (1,1 \dots 1)^T$
Collinearity of	channel vectors AA→user j and	AA→user k
Definition	$H^k = c \cdot H^j$; $c = \lvert c \rvert e^{i\psi}$	
Redefinition	$\begin{cases} \lvert H \rvert^k = \lvert c \rvert \cdot \lvert H \rvert^j \\ \Phi^k - \Phi^j = \psi \cdot (1,1 \dots 1)^T \end{cases}$	$\Phi^k - \Phi^j = \psi \cdot (1,1 \dots 1)^T$
Test	$\frac{((H^k)^* H^j)^2}{(H^k)^* H^k \cdot (H^j)^* H^j} \to 1$	$\frac{V(\psi)}{V^0(\psi)} \ll 1$ $V(\psi) = \frac{1}{N_X - 1} \sum_{q=1}^{N_X} (\psi_q - \bar{\psi})^2$ $\bar{\psi} = \frac{1}{N_X} \sum_{q=1}^{N_X} \psi_q$; $\psi_q = \varphi_q^k - \varphi_q^j$ $V^0(\psi) = (2/3) \cdot \pi^2$ is variance of phase difference ψ if both phases are independent and uniformly distributed in $[-\pi \dots \pi)$



If two users ***j*** and ***k*** are located on the same beam then their channel vectors are close to collinear

Multiple Input – Multiple Output Communications



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

$$\mathbf{x} \in \mathbb{C}^{M\times 1}, \mathbf{y} \in \mathbb{C}^{N\times 1}, \mathbf{H} = \mathbf{R}_{TX}^{1/2}\mathcal{N}(0,\mathbf{I}) \in \mathbb{C}^{N\times M}$$

MIMO model

MIMO model

There are transmission on the same time-frequency resource

N_X Tx-antennas

N_Y Rx-antennas

N Data streams $N \leq \operatorname{rank}(H)$

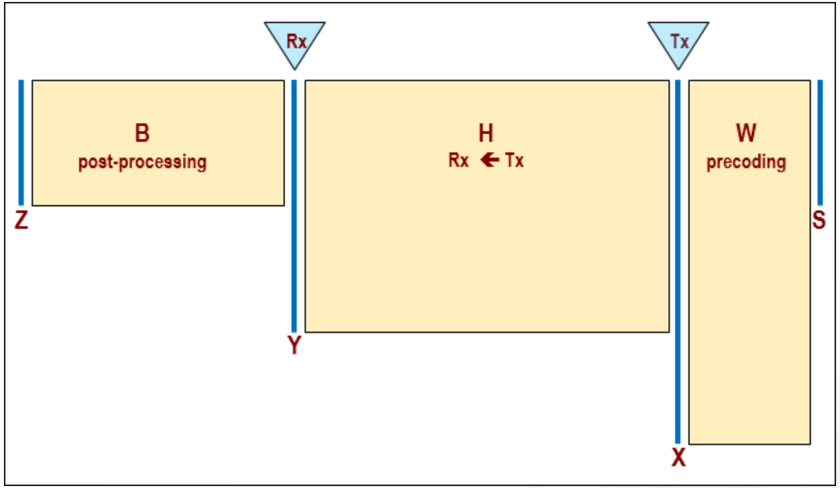
Vector/matrix	Rows	Columns	Description
S	N	1	Vector of input signals in data streams
X	N_X	1	Vector of signals generated by N_X Tx-antennas
Y	N_Y	1	Vector of signals received by N_Y Rx-antennas
Z	N	1	Vector of output signals in data streams
W	N_X	N	Precoding matrix; $X = WS$
H	N_Y	N_X	Channel matrix; $Y = HX$
B	N	N_Y	Post-processing matrix; $Z = BY$
T	N	N	MIMO map matrix $T = BHW$

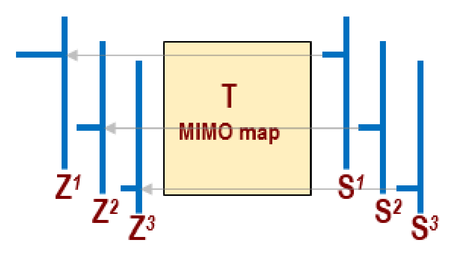
MIMO maps $S \to (X = WS) \to (Y = HWS) \to Z = BHWS = TS$

We target to minimize the interference between data streams. Zero interference means that vectors of Cartesian basis in S-space are eigenvectors of T, eigenvalues are real positive

$$Z^j = TS^j = \lambda_j S^j \; ; \; \lambda_j > 0 \; ; \; S^j = (0 \ldots 0{,}1{,}0 \ldots 0)^T : \text{component } \# \, j \text{ is } 1$$

Thus desired form of T is diagonal matrix.



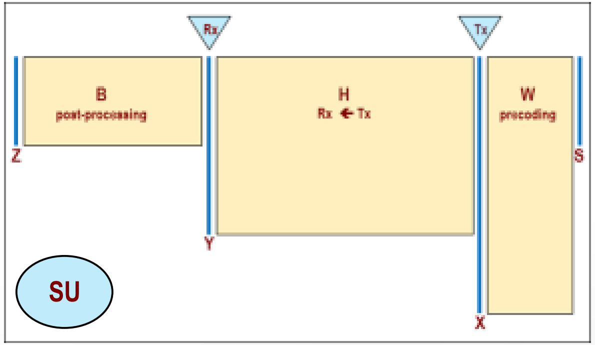


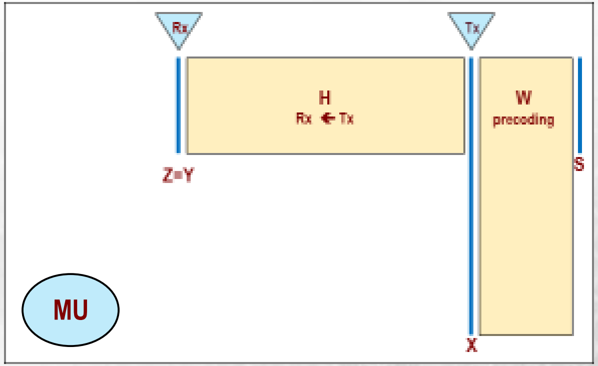
Vladimir Lyashev

MIMO model: SU/MU

Matrix H is beyond our control. We can control matrixes $W; B$ under the assumption that H is known. There are two following DL cases

MIMO case	Comments
SU (Single User)	All N_Y Rx-antennas belong to one UE. All N data streams are for this one UE. Due to this post-processing is possible (B is related to this one UE). Due to this total channel $S \to Z$ can be optimized
MU (Multi User)	There are N_Y users with one Rx-antenna each. One data stream corresponds to one user. $N = N_Y \le N_X;\ Z = Y\ ;\ B = I$ Due to this post-processing is impossible.





SU MIMO

SU MIMO

We use Singular Value Decomposition (SVD) $H = UDV^*$

Vector/matrix	Rows	Columns	Description
V	N_X	N_X	Unitary matrix $V^*V = VV^* = I$
D	N_Y	N_X	Real diagonal matrix $d_{11} \geq d_{22} \geq \cdots > 0$
U	N_Y	N_Y	Unitary matrix $U^*U = UU^* = I$
$\tilde{X}$	N_X	1	$\tilde{X} = V^*X$; $\tilde{X}^*\tilde{X} = X^*VV^*X = X^*X$
$\tilde{Y}$	N_Y	1	$\tilde{Y} = D\tilde{X}$; $Y = U\tilde{Y}$; $Y^*Y = \tilde{Y}^*\tilde{Y}$

Select $B^* = U ; W = V$ (only first N columns of $U; V$). MIMO maps vectors as follows

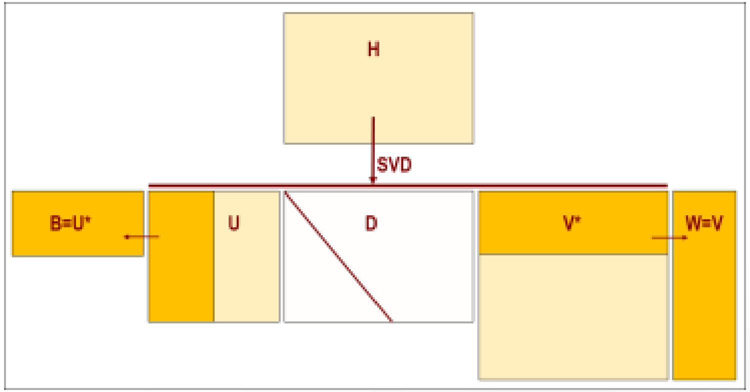
$$S^j \to (X^j = WS^j) \to (\tilde{X}^j = V^*WS^j = S^j) \to (\tilde{Y}^j = D\tilde{X}^j = d_{jj}S^j)$$

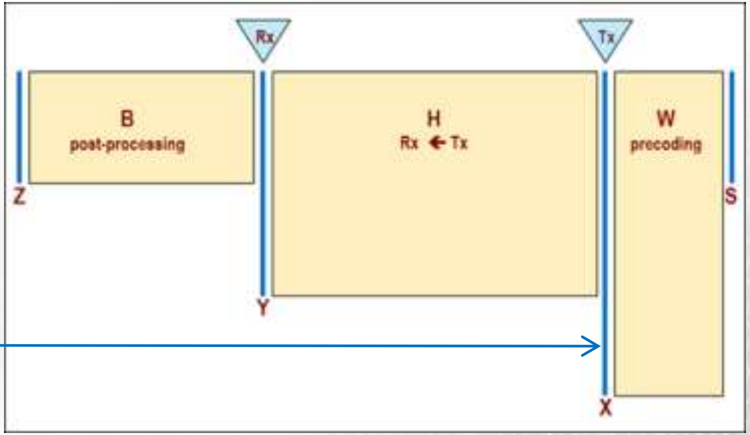
$$\to (Y^j = U\tilde{Y}^j = d_{jj}US^j) \to Z^j = BY^j = d_{jj}BUS^j = d_{jj}S^j$$

Data streams with low serial numbers have better channel ($d_{11} \geq d_{22} \geq \cdots > 0$). Tx-power handling is trivial

$$S = \sum_{j=1}^{N} s_j^j S^j \to X = \sum_{j=1}^{N} s_j^j X^j \; ; \; X^j = VS^j$$

$$p_{Tx}^j = (s_j^j X^j)^* s_j^j X^j = (s_j^j VS^j)^* s_j^j VS^j = (s_j^j)^2 \cdot (S^j)^* V^*VS^j = (s_j^j)^2$$





Vladimir Lyashev

MU MIMO

Matrix T has to be diagonal with real positive diagonal elements

$$T = HW = DA \quad (\mathcal{T} = H\mathcal{W} = D\,;\; T = \mathcal{T}A\,;\; W = \mathcal{W}A\,)\,;\;\; A = \mathrm{diag}(\alpha_1, \alpha_2 \ldots \alpha_N)$$

Case 1 $HH^* = D$

$H\mathcal{W} = D \;\Rightarrow\; \mathcal{W} = H^*\,;\; \mathcal{T} = HH^*\,;\; W = H^*A$: Maximum Ratio Transmission precoding

Case 2 General H of full-rank; $N \le N_X$. Equation $H\mathcal{W} = D\,;\; D = I$ has a lot of solutions. We choose pseudo-inverse one with minimum $\|\mathcal{W}\|$ (Frobenius norm)

$H\mathcal{W} = I \;\Rightarrow\; \mathcal{W} = H^+\,;\; W = H^+A$: Zero-Forcing precoding

There are two ways for H^+ computation

1. Tikhonov regularization approach: $H^+ = \lim_{\alpha \to +0} (H^*H + \alpha I)^{-1}H^*$
2. SVD approach ($H = UDV^*$): $H^+ = VD^+U^*\,;\; D^+ = \widehat{D}^T$: $\widehat{D}$ is D with inverted d_{jj}

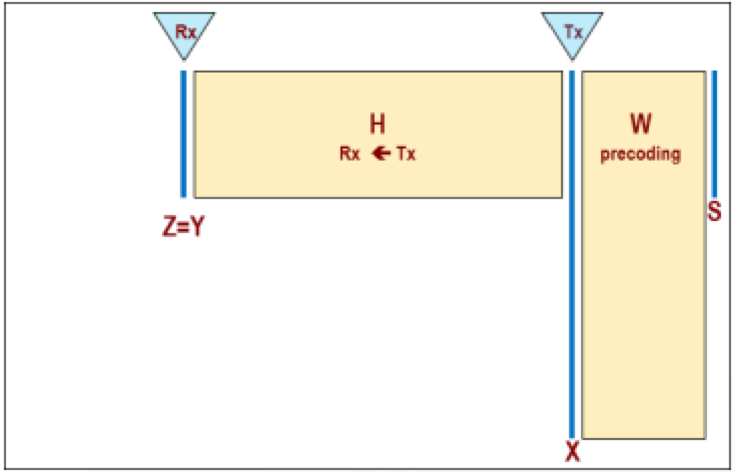
MIMO maps vectors as follows

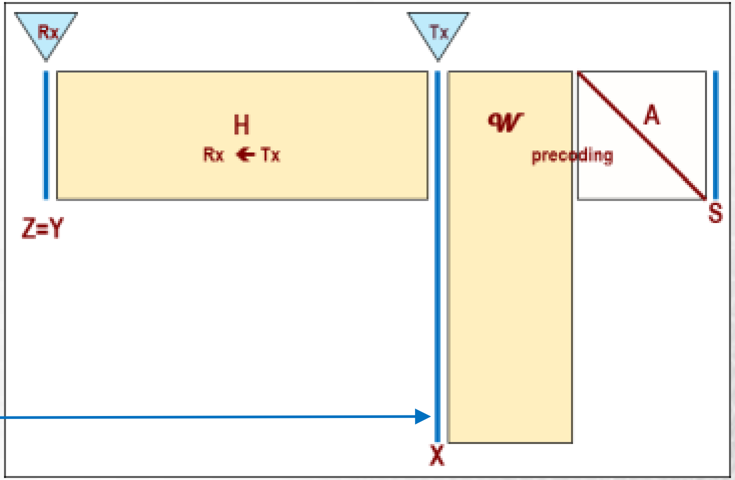
$$S^j \to (X^j = \alpha_j \mathcal{W} S^j) \to (Y^j = \alpha_j H\mathcal{W}S^j = \alpha_j S^j) \to Z^j = \alpha_j S^j$$

Parameters α_j are determined from Tx-power limitations

$$S = \sum_{j=1}^{N} s_j^j S^j \;\to\; X = \sum_{j=1}^{N} s_j^j X^j\,;\;\; X^j = \alpha_j \mathcal{W} S^j$$

$$p_{\mathrm{Tx}}^j = (s_j^j X^j)^* s_j^j X^j = (s_j^j \alpha_j)^2 (\mathcal{W}S^j)^* \mathcal{W}S^j \;\Rightarrow\; (s_j^j \alpha_j)^2 = p_{\mathrm{Tx}}^j / (\mathcal{W}S^j)^* \mathcal{W}S^j$$





Vladimir Lyashev

Interference between data streams in MIMO

Interference between data streams in MIMO

Theoretically, signals in output data streams are free of interference

$$S = \sum_{j=1}^{N} s_j^j S^j \rightarrow Z = \sum_{j=1}^{N} s_j^j Z^j \; ; \; Z^j = TS^j = \lambda_j S^j$$

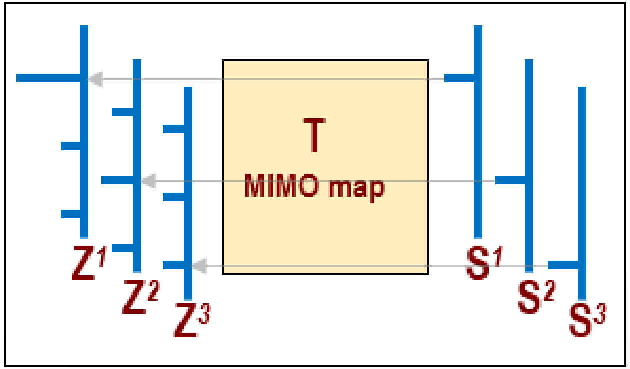
In practice, matrix T can be not strictly diagonal and interference can occur

$$Z^j = \lambda_j S^j \rightarrow Z^j = TS^j = \left(z_1^j, z_2^j \ldots\right)^T = \left(t_{1j}, t_{2j} \ldots\right)^T ; \; T = \{t_{kj}\} ; \; \mathcal{T} = \{\mathcal{t}_{kj}\}$$

In this case, SIR for data stream # j can be computed as follows

$$\mathrm{SIR}^j = \left(s_j^j z_j^j\right)^* s_j^j z_j^j \Bigg/ \sum_{\substack{k=1\ldots n \\ k\neq j}} \left(s_k^k z_j^k\right)^* s_k^k z_j^k = \left(s_j^j\right)^2 t_{jj}^* t_{jj} \Bigg/ \sum_{\substack{k=1\ldots n \\ k\neq j}} \left(s_k^k\right)^2 t_{jk}^* t_{jk} =$$

$$= \left(s_j^j \alpha_j\right)^2 \mathcal{t}_{jj}^* \mathcal{t}_{jj} \Bigg/ \sum_{\substack{k=1\ldots n \\ k\neq j}} \left(s_k^k \alpha_k\right)^2 \mathcal{t}_{jk}^* \mathcal{t}_{jk} = \frac{p_{\mathrm{Tx}}^j \mathcal{t}_{jj}^* \mathcal{t}_{jj}}{(\mathcal{W}S^j)^* \mathcal{W}S^j} \Bigg/ \sum_{\substack{k=1\ldots n \\ k\neq j}} \frac{p_{\mathrm{Tx}}^k \mathcal{t}_{jk}^* \mathcal{t}_{jk}}{(\mathcal{W}S^k)^* \mathcal{W}S^k}$$



$$T = B \times H \times W$$

One of the key goal in wireless communication algorithm design is minimize interference:

- Way #1: $T \rightarrow diag\ matrix$
- Way #2: ???

Vector Precoding

The set of equations that describe the MMSE solution for vector precoding are:

$$\mathbf{y} = \mathbf{Hx} + \mathbf{n}$$

Set $\mathbf{x} = \mathbf{p}s$ that

$$\mathbf{p} = \arg\min_{\hat{p}} \left\| \mathbf{H}^H \left(\mathbf{HH}^H + \mathbf{R}_{uu} \right)^{-1} \mathbf{y} - \hat{\mathbf{p}}s \right\|_2^2$$

Using SVD, the matrix **H** can be diagonalized by orthogonal matrices **U** and **V**

$$\mathbf{H} = \mathbf{U} \times \mathbf{\Lambda} \times \mathbf{V}^H$$

Thus, $\mathbf{y} = \left[\mathbf{U} \times \mathbf{\Lambda} \times \mathbf{V} \right] x + \mathbf{n}$ by transmitting $\dot{\mathbf{x}} = x\mathbf{V}$

Instead of **x** and pre-multiply the receive signal by vector $\mathbf{U}^H$, the transformed received signal vector becomes:

$$\tilde{\mathbf{y}} = \mathbf{U}^H \mathbf{y} = \mathbf{U}^H \left[\mathbf{U} \times \mathbf{\Lambda} \times \mathbf{V}^H \right] \underbrace{\dot{\mathbf{x}}}_{x\mathbf{V}} + \mathbf{U}^H \mathbf{n} = \mathbf{\Lambda} x + \tilde{\mathbf{n}}$$

In TDD systems, the channel is the same on transmitter and receiver, but it is changing in time and robust precoding of the channel is challengeable problem in wireless comm.

Open Problems in Wireless Communication

- MULTI-USERS SYSTEMS

  (CDMA, SCMA, MU-MIMO, etc.)

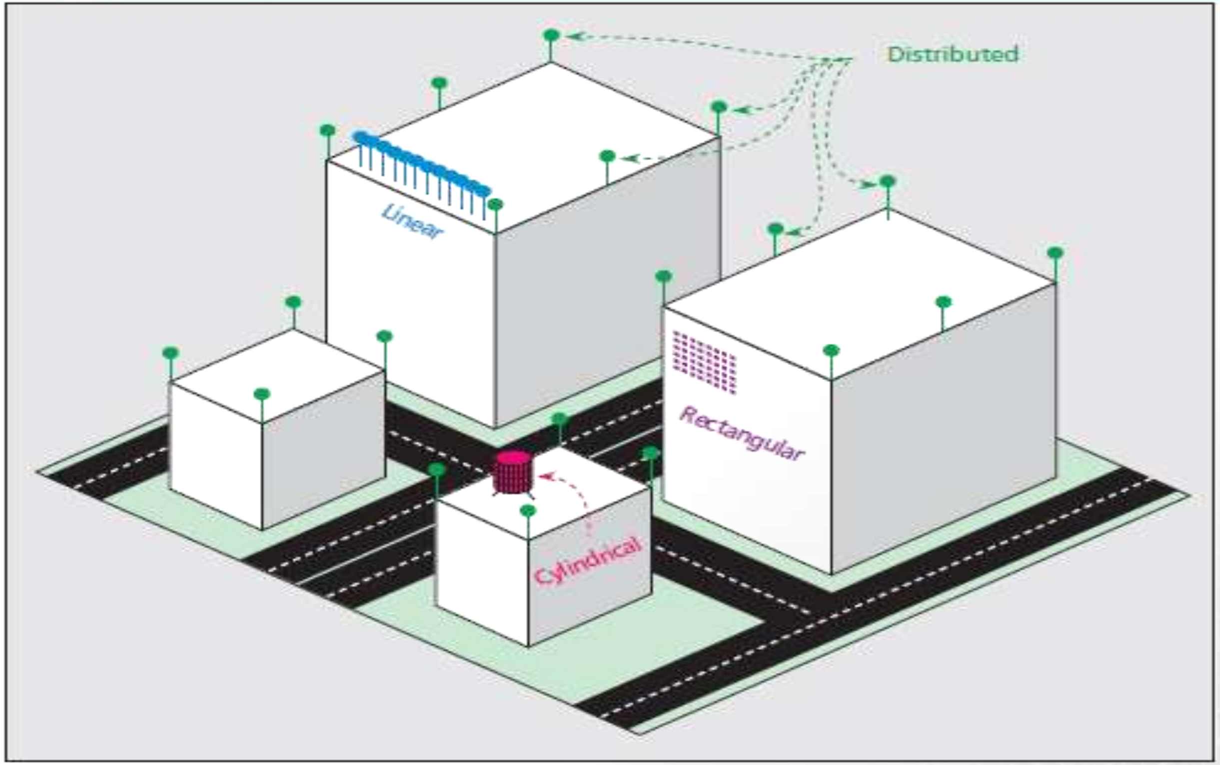
- MULTI-ANTENNA SYSTEMS

  (massive-MIMO,

  cell splitting/antenna selection,

  precoding/beamforming)

MIMO System Evolution



MORE spectrum efficiency → MORE antenna elements

Capacity of the system and linear space extension

$$C_{SISO} = B_f \log_2\left(1 + \frac{P_{TX}\|\mathbf{H}\|}{\sigma_n^2}\right) = B_f \log_2\left(1 + \frac{P_{TX}\lambda_1^2}{\sigma_n^2}\right)$$

$$C_{MIMO} = B_f \log_2 \det(\mathbf{I} + \mathbf{HQH}^H)$$

$$\mathbf{Q} \geq 0, \mathrm{trace}\,\mathbf{Q} \leq \frac{P_{TX}}{N_{TX}\sigma_n^2}$$

P_{TX} – power of single transmitter;
N_{TX} – number of transmission antennas;
Q – covariance matrix of transmitted signal.

In current product it is required matrix operations for
N_{RB} (6…110) matrices with size 4x4 … 128x128 per 1ms or less
In future N_{RB} can be extended to 500, and time scale can be reduced 10 times!!!

Vladimir Lyashev

What is channel matrix ?

$$\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^H \Rightarrow \mathbf{H}\mathbf{V} = \mathbf{U}\mathbf{\Lambda} \quad \Rightarrow \quad \mathbf{u}_1 = \frac{1}{\lambda_1}\mathbf{H}\mathbf{v}_1 = \frac{1}{\lambda_1}[\mathbf{h}_1 \ \mathbf{h}_2 \ldots \mathbf{h}_M]\mathbf{v}_1$$

$$\begin{aligned}
\mathbf{u}_i &= \tfrac{1}{\lambda_i}\mathbf{R}_{TX}^{1/2}\left[\mathcal{N}^{N\times 1}(0,\mathbf{I})\ \mathcal{N}^{N\times 1}(0,\mathbf{I})\ \ldots\ \mathcal{N}^{N\times 1}(0,\mathbf{I})\right]\mathbf{v}_i = \\
&= \tfrac{1}{\lambda_i}\mathbf{R}_{TX}^{1/2}\left[v_{i1}\mathcal{N}^{N\times 1}(0,\mathbf{I}) + v_{i2}\mathcal{N}^{N\times 1}(0,\mathbf{I}) + \cdots + v_{iM}\mathcal{N}^{N\times 1}(0,\mathbf{I})\right] = \\
&= \tfrac{1}{\lambda_i}\mathbf{R}_{TX}^{1/2}\mathcal{N}^{N\times 1}(0,\mathbf{I}) = \tfrac{1}{\lambda_i}\mathcal{N}^{N\times 1}(0,\mathbf{R}_{TX}).
\end{aligned}$$

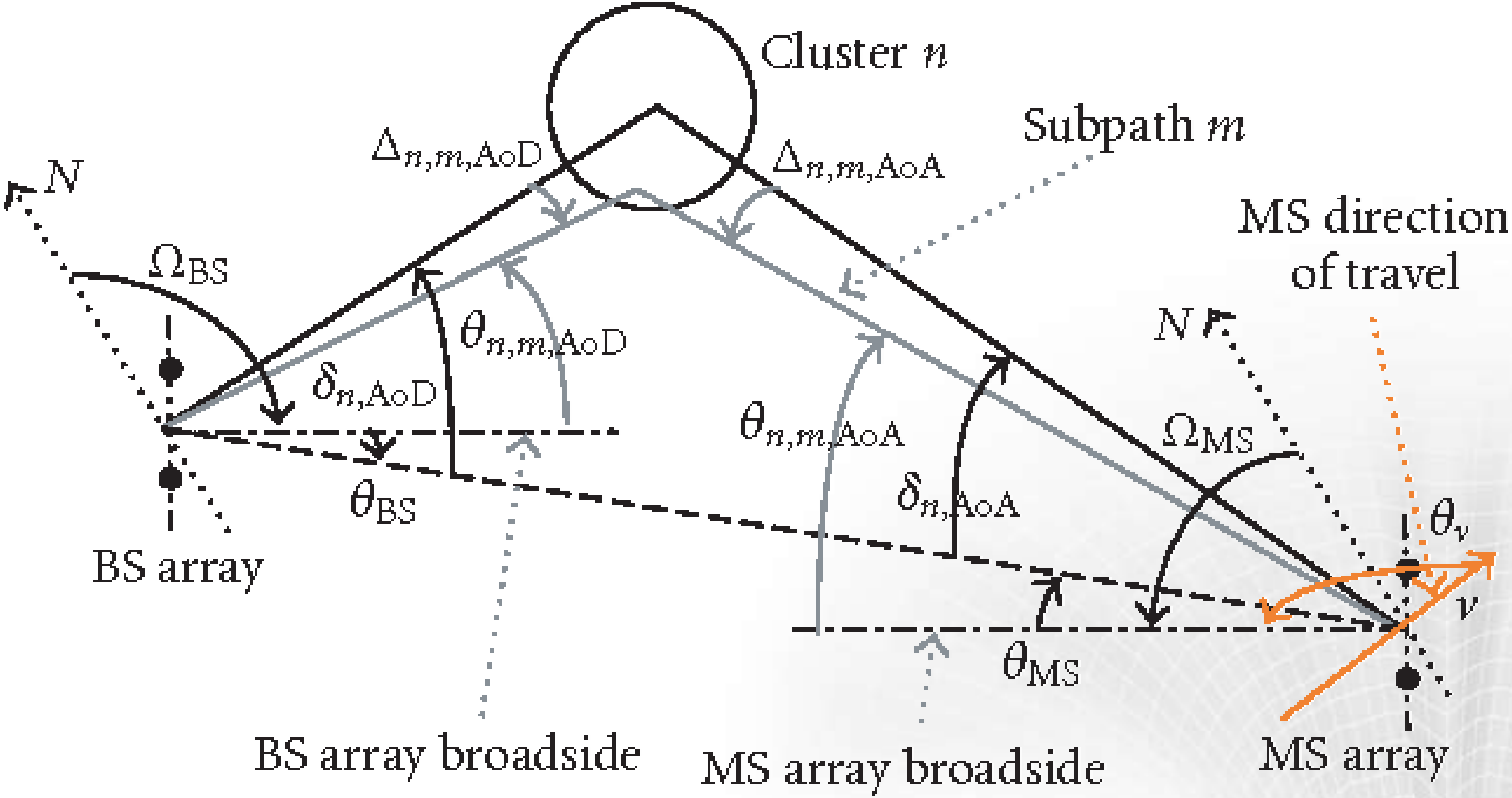
Actually, the spatial correlation matrix $\mathbf{R}_{TX}$ is depends of angle of destination (AoD). Thus we can build egenspace matrix $\hat{\mathbf{U}}$ from defined vectors $\mathbf{u}_i$ and correlation matrix $\hat{\mathbf{U}}\hat{\mathbf{U}}^H$, which are similar to Wishart matrices with some constrain on eigenvalues λ_i distribution.

In our assumption, egenvalues can be defined as

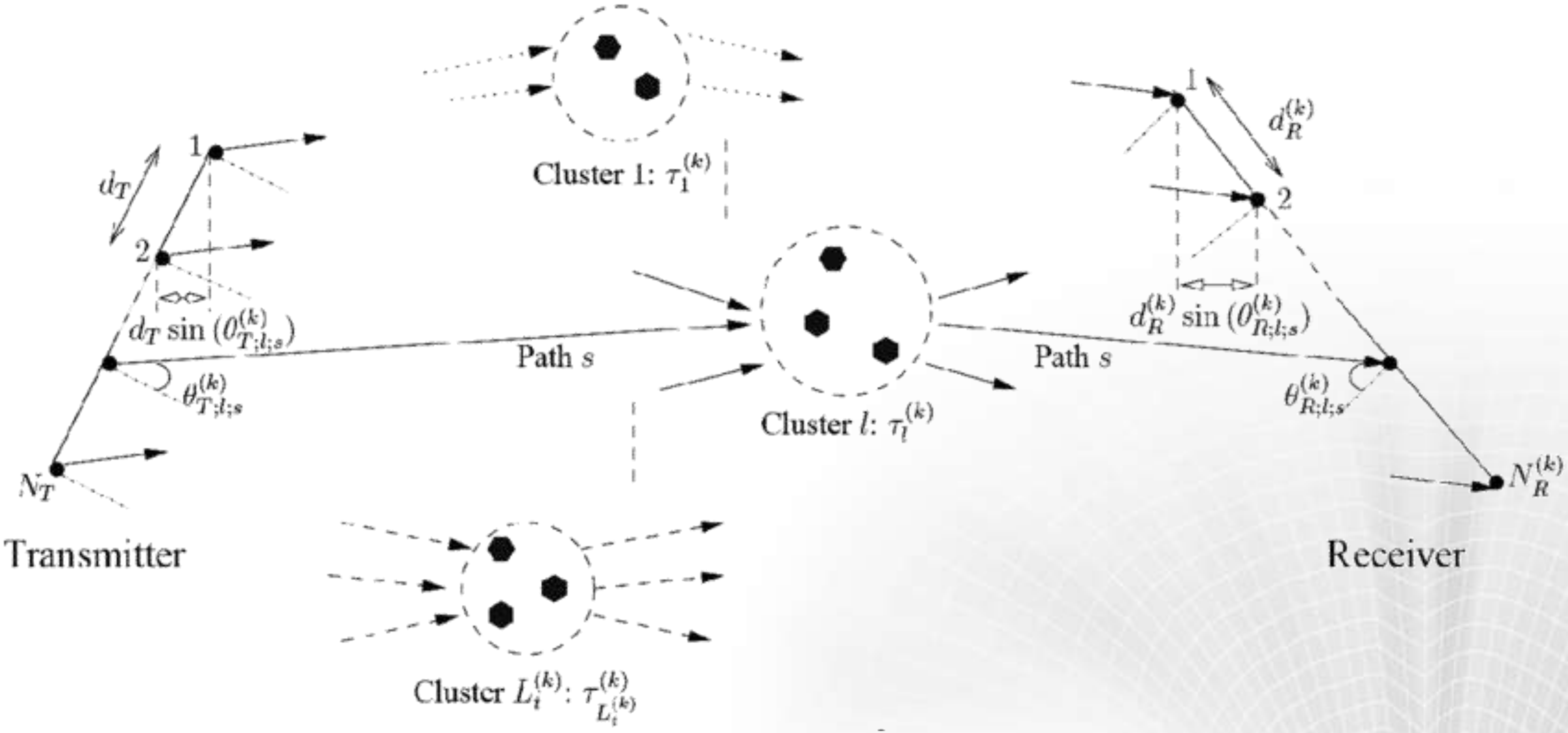
$$\frac{\lambda_2}{\lambda_1} \in [0...0.9] \text{ and } \left.\frac{\lambda_i}{\lambda_{i-1}}\right|_{i>2} < \varepsilon.$$

Vladimir Lyashev

Spatial Channel Model

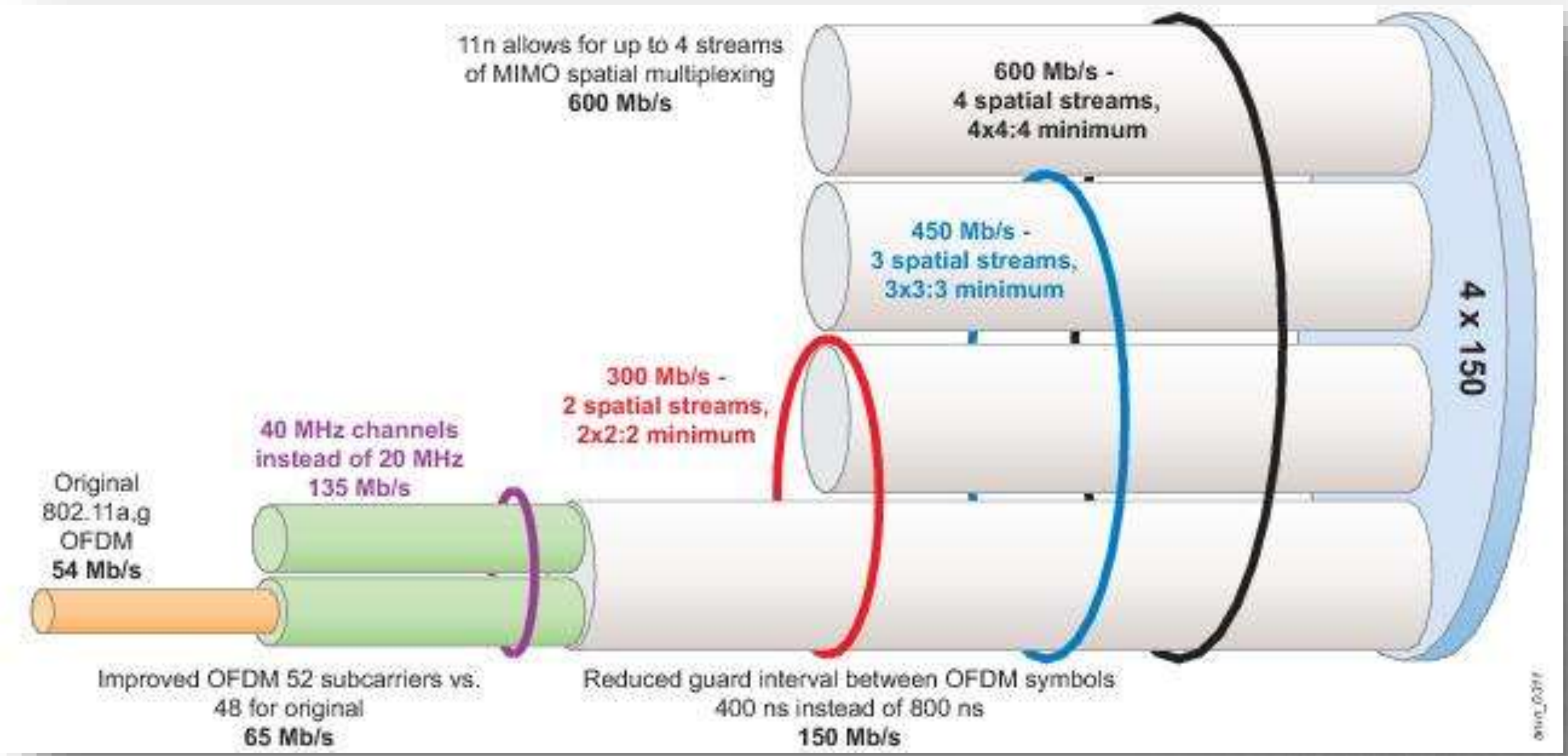


Spatial Channel Model



Vladimir Lyashev

Spatial Channel Model



State-of-the-Art Problem

- To solve the matrix equation as detection problem

$$\mathbf{y} = \mathbf{Hx} + \mathbf{n}$$

- Maximize Signal-to-Noise Ratio for better performance

$$\frac{\|\mathbf{wHx}\|_2}{\|\mathbf{n}\|_2} \rightarrow \max$$

- User grouping for better throughput

$$\gamma \log_2 \det(\mathbf{I} + \mathbf{HQH}^H) \rightarrow \max$$

Detection in Gaussian Noise

We can obtain a scalar sufficient statistic y (for x on the basis of the observation of $\mathbf{r}$), by projecting $\mathbf{r}$ on the direction of $\mathbf{h}$. Hence, the sufficient statistic y is given by

$$y = \mathbf{h}^T \mathbf{r} = \mathbf{h}^T \left(\underbrace{\mathbf{v}\sqrt{E_S}\,x}_{\mathbf{s}} + \mathbf{w} \right) = \|\mathbf{h}\|^2 \sqrt{E_S}\,x + \mathbf{h}^T \mathbf{w} = \sqrt{E_S}\,x + n$$

where $\quad n = \mathbf{h}^T \mathbf{w} \sim \mathcal{N}\left(0, N_0/2\right).$

As the probability density function of y given x is equal to

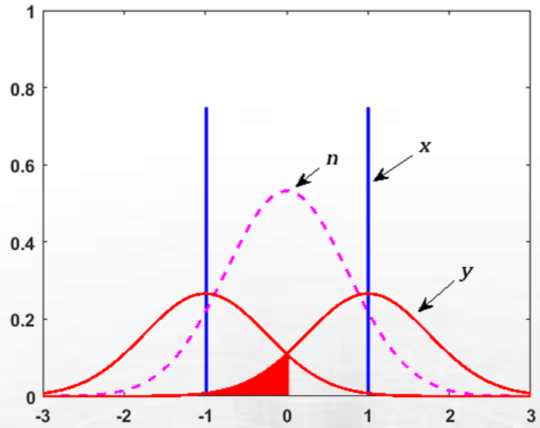
$$f_{Y|X}(y\,|\,x) = \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{\left(y - \sqrt{E_S}\,x\right)^2}{N_0} \right),$$

the log-likelihood ratio corresponding to y is given by

$$\mathrm{LLR}(y) = \log \frac{f_{Y|X}(y\,|\,x=1)}{f_{Y|X}(y\,|\,x=-1)} = \frac{4y\sqrt{E_S}}{N_0}.$$

Probability error

Furthermore, the threshold $\eta = P(x = -1)/P(x = 1)$ equals 1 and log η = 0. Hence, the MAP rule can be expressed as follows:



Choose $x = 1$ $if\ LLR(y) > 0$. Otherwise, choose $x = -1$.

Using the isotropic property of Gaussian noise, we readily find the probability of error:

$$P(e) = P(e\,|\,x=1) = P(\mathrm{LLR}(y) \le 0\,|\,x=1) =$$

$$= P(y \le 0\,|\,x=1) = P(z + \sqrt{E_S} \le 0) = P(z \ge \sqrt{E_S}) = Q\left(\sqrt{\frac{2E_S}{N_0}}\right).$$

SNR

Remark

MAP detector provides the best performance, but…
…complexity is not suitable for real-time processing!

MMSE solution

$$x = \arg\min\left(y - \mathbf{h}^T \mathbf{r}(x)\right)$$

channel estimation is necessary

$$y = \mathbf{h}^T \mathbf{r} = \mathbf{h}^T \left(\underbrace{\mathbf{v}\sqrt{E_S}\,x}_{\mathbf{s}} + \mathbf{w} \right) = \mathbf{h}^T (\mathbf{s} + \mathbf{w}), \qquad \mathbf{h}, \mathbf{r} \in \mathbb{C}^{N \times 1}$$

Least square channel estimation

$$h_{LS}^{(i)} = s_P^{(i)} y = h^{(i)} + s_P^{(i)} n,$$

$$\text{here } \left\langle s_P^{(i)}, s_P^{(j)} \right\rangle = \begin{cases} 1, & \text{for } i = j; \\ 0, & \text{for } i \neq j. \end{cases}$$

LS error: $\mathbf{E}\left[\left\| h_{LS}^{(i)} - h^{(i)} \right\|^2\right] = \mathbf{E}\left[\left| s_P^{(i)} n \right|^2\right] = \sigma_n^2$

That means if we use LS channel estimation, variance error of estimation equal to noise variance.

MMSE solution

We want to improve LS estimation using linear filtration $H_{mmse} = W \cdot H_{ls}$ and minimalizing expectation of square error

$$W = \arg\min_{W} \mathrm{E}\left[\left\| h - W h_{LS} \right\|^2 \right]$$

Reasonable question is : why linear filtration is good here?

To answers we need to recall channel model representation in time domain

$$h(t) = \sum_{i=1}^{L} \delta(t - \tau_i) \cdot a_i \rightarrow FFT \rightarrow H(f) = \sum_{i=1}^{L} a_i \cdot \exp(-j \cdot 2\pi \cdot \tau_i \cdot f)$$

L –maximal number of channel delays (taps). We see that frequency representation consists of sum of complex exponents (harmonics). In theory multiple harmonic estimation is non-linear problem, however if parameters of system was chosen smartly, correlation between neighbor subcarriers (measure of linear dependency) would quite large.

Coherence bandwidth estimation is : $BW_{coh} \approx \frac{1}{\tau_{rms}}$ where τ_{rms}-root mean spread of channel taps. That is why inside BW_{coh} we could use linear filtration.

Linear MMSE for ChEst: simple derivation

Assuming that pilot slice that we use inside coherence bandwidth let us solve

$$\mathbf{W} = \arg\min_{\mathbf{W}} \mathbf{E}\left[\left\|\mathbf{h} - \mathbf{W}\mathbf{h}_{LS}\right\|^2\right], \quad \mathbf{h}, \mathbf{h}_{LS} \in \mathbb{C}^{N\times 1}, \quad \mathbf{W} \in \mathbb{C}^{N\times N}$$

Principle that we use called orthogonalization principle and imply that error of estimation is not correlated we observed data – meaning that all linear information is absorbed by filtration

$$\mathbf{E}[(\mathbf{h} - \mathbf{W}\cdot\mathbf{h}_{ls})\cdot\mathbf{h}_{ls}] = \mathbf{0}$$

$$\mathbf{h}_{ls} = \mathbf{h} + \boldsymbol{n}$$

So

$$\mathbf{E}\left[\mathbf{h}\cdot\mathbf{h}_{ls}^{H}\right] - \mathbf{W}\cdot\mathbf{E}\left[\mathbf{h}_{ls}\cdot\mathbf{h}_{ls}^{H}\right] = \mathbf{0}$$

$$\mathbf{W} = \mathbf{R}_{h\cdot h_{ls}}\cdot\mathbf{R}_{h_{ls}\cdot h_{ls}}^{-1} = \mathbf{R}_{hh}\cdot(\mathbf{R}_{hh} + \sigma_n^2\cdot\mathbf{I})^{-1}$$

We assume here that noise is uncorrelated with channel and equal for all bandwidth which is true for thermal noise.

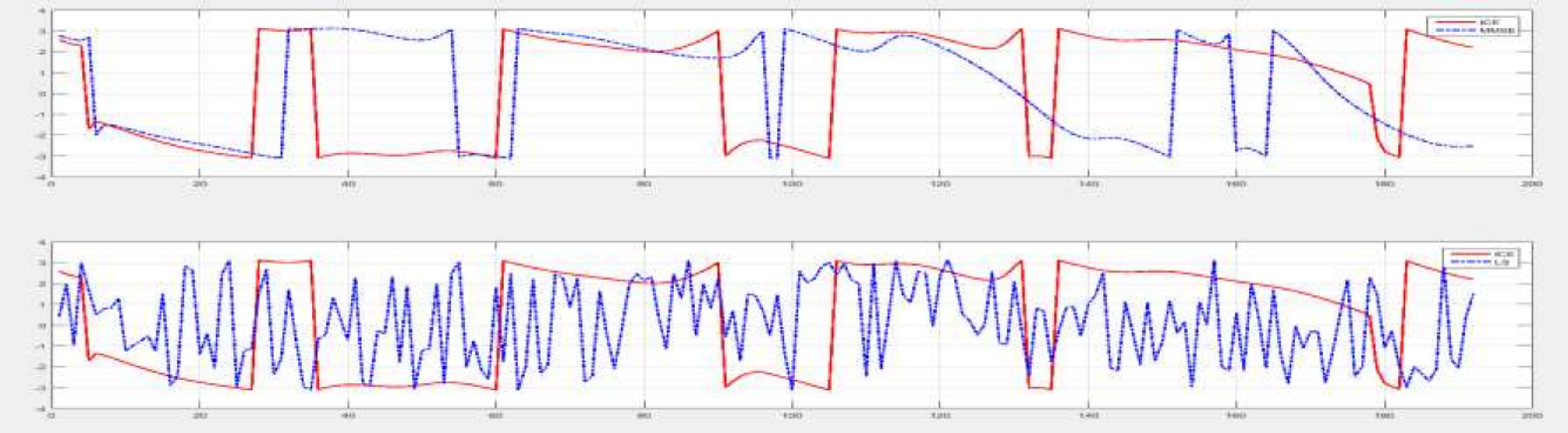
Estimation is ready but require some knowledge of $\mathbf{R}_{hh}$ -covariance matrix of the channel, and also noise variance

Where can we obtain matrix $\mathbf{R}_{hh}$?

$$\mathbf{R}_{hh}$$

measurements

modeling



Tikhonov Regularization in Inverse Problem

Each least squares problem has to be regularized. In the linear case,

$$\mathbf{Hx} = \mathbf{y} + \mathbf{n}$$

we want to solve minimization problem

$$\|\mathbf{Hx} - \mathbf{y}\|_2 \le \|\mathbf{n}\|_2$$

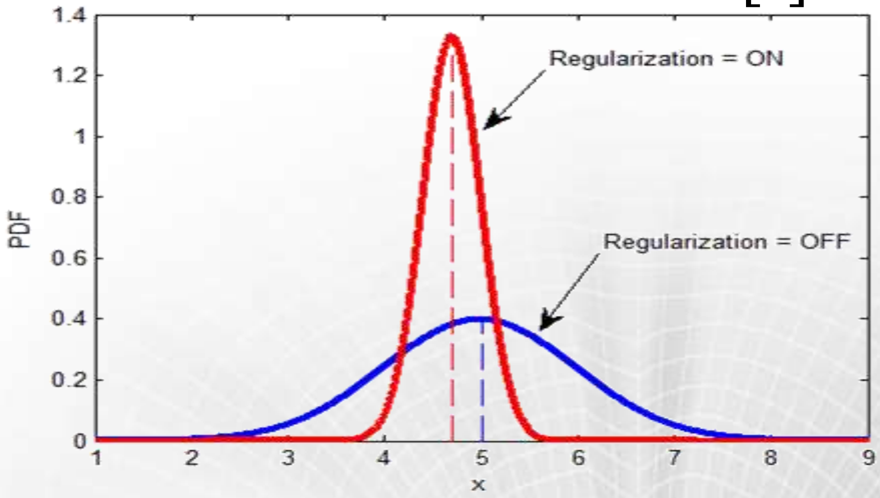
after regularization

$$\|\mathbf{Hx} - \mathbf{y}\|_2 + \|\mathbf{\Gamma x}\|_2 \to \min$$

the solution is

$$\mathbf{x} = \left(\mathbf{H}^H\mathbf{H} + \mathbf{\Gamma}^H\mathbf{\Gamma}\right)^{-1}\mathbf{H}^H\mathbf{y} = \left(\underbrace{\mathbf{H}^H\mathbf{H}}_{\mathbf{R}_{hh}} + \gamma\mathbf{I}\right)^{-1}\mathbf{H}^H\mathbf{y}$$

$$\mathbf{x} = [5]$$



How to define Tikhonov matrix Γ ?

QR ML Estimator (example)

QR decomposition often used in MIMO systems

Assuming **H** has a rank of *r*, we have: **H = QR**,
Where **Q** is an *N*×*r* orthonormal matrix, **R** is an *r*×*r* upper triangular matrix.

Solution:

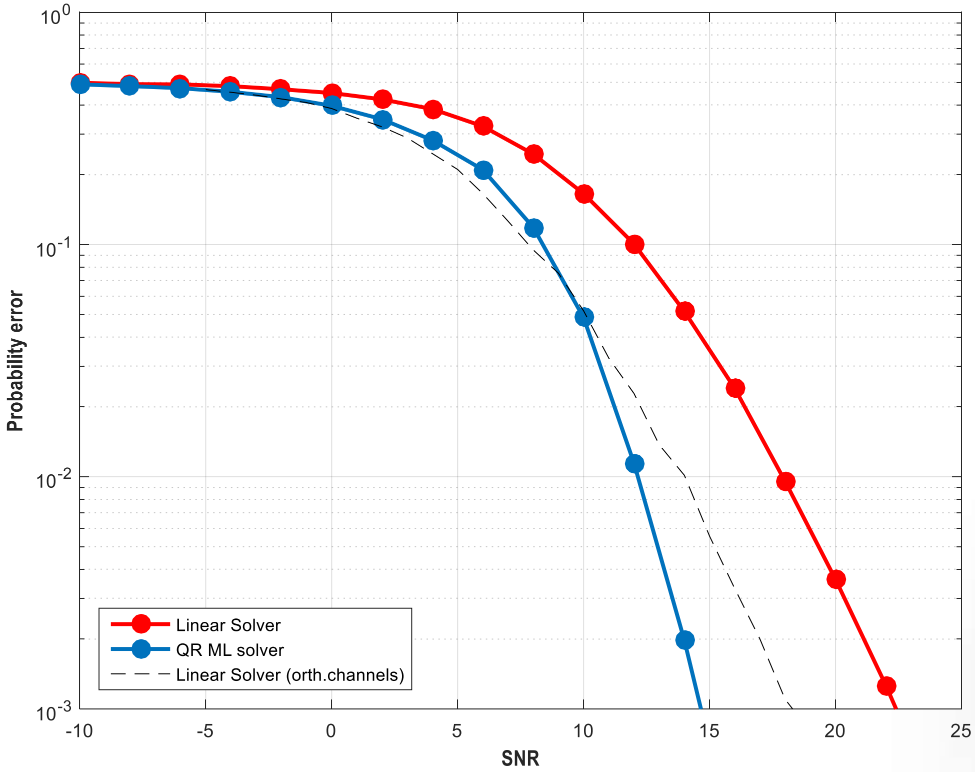
Since **Q** is orthonormal, we have:

$$\left\|\mathbf{y}-\mathbf{H}\hat{\mathbf{x}}_{ML}\right\|^2=\left\|\mathbf{y}-\mathbf{Q}\mathbf{R}\hat{\mathbf{x}}_{ML}\right\|^2=\left\|\mathbf{Q}\left(\mathbf{Q}^H\mathbf{y}-\mathbf{R}\hat{\mathbf{x}}_{ML}\right)\right\|^2=\left\|\mathbf{Q}^H\mathbf{y}-\mathbf{R}\hat{\mathbf{x}}_{ML}\right\|^2\cong$$

$$\cong\left\|\tilde{\mathbf{y}}-\mathbf{R}\hat{\mathbf{x}}_{ML}\right\|^2=\left\|\begin{pmatrix}\tilde{y}_0\\ \tilde{y}_1\\ \vdots\\ \tilde{y}_{r-1}\end{pmatrix}-\begin{pmatrix}R_{00} & R_{01} & \cdots & R_{0(r-1)}\\ 0 & R_{11} & & R_{1(r-1)}\\ \vdots & & \ddots & \vdots\\ 0 & 0 & 0 & R_{(r-1)(r-1)}\end{pmatrix}\begin{pmatrix}\hat{s}_0\\ \hat{s}_1\\ \vdots\\ \hat{s}_{r-1}\end{pmatrix}\right\|^2$$

Can be viewed as an *r* layer system.

QR ML vs. Linear Detection



Two-state system {-1; +1}

Linear receiver required:

matrix inversion and matrix product (depends of antenna number)

CPLX: (N^3+N^2) MUL

ML receiver required:

matrix product times 2^N

CPLX: $2^N N^2$ MUL

CPLX*: $2^N \Sigma n$ MUL

For system 4x4

- Linear: 80 MUL
- ML: 256 MUL
- ML-QR: 160 MUL (-37.5%)

Parallel

Vladimir Lyashev

SVD

is very important operation !!!

Principle of Biorthogonality

$$\mathbf{H} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}$$

$$\mathbf{U} = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_N \end{pmatrix} \qquad \mathbf{V} = \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_M \end{pmatrix}$$

$$\mathbf{u}_i^H \mathbf{H} = \mu_i \mathbf{u}_i^H \qquad \mu_i \neq \lambda_i \qquad \mathbf{H}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

to satisfy biorthogonality principle, we require $\langle \mathbf{x}, \mathbf{y} \rangle = 0$

$$\left.\begin{aligned} \mathbf{u}_i^H \mathbf{H} \mathbf{v}_j &= \mathbf{u}_i^H \lambda_j \mathbf{v}_j = \lambda_j \mathbf{u}_i^H \mathbf{v}_j = \lambda_j \left\langle \mathbf{u}_i, \mathbf{v}_j \right\rangle \\ \mathbf{u}_i^H \mathbf{H} \mathbf{v}_j &= \mu_i \mathbf{u}_i^H \mathbf{v}_j = \mu_i \left\langle \mathbf{u}_i, \mathbf{v}_j \right\rangle \end{aligned}\right\} \quad \lambda_j \left\langle \mathbf{u}_i, \mathbf{v}_j \right\rangle = \mu_i \left\langle \mathbf{u}_i, \mathbf{v}_j \right\rangle \quad \Rightarrow \quad \left\langle \mathbf{u}_i, \mathbf{v}_j \right\rangle = 0.$$

Vladimir Lyashev

Resume

Typical size of the matrix is less 64×64 elements. For such matrix we need fast algorithms for

- eigenvector decomposition;
- matrix inversion (current baseline is classical Cholesky decomposition algorithm)

The questions are:

1. Can we define some less complexity algorithm for matrix inversion and eigenvector calculation than provided baseline algorithms?
2. Do some fast algorithms (approaches) exist in modern linear algebra to compute such specific matrices?

Homework

1. Generate Wishart-matrix $\mathbf{A}_k$ with following parameters of distribution:

Sigma =

1.0000	0.1000	0.1000	0.1000
0.1000	1.0000	0.1000	0.1000
0.1000	0.1000	1.0000	0.1000
0.1000	0.1000	0.1000	1.0000

df = 8

```
df = 8;
Sigma = 0.1*ones(4) + 0.9*eye(4);
A = wishrnd(Sigma,df)/df
```

2. Set up 2000 samples of equation:

$$\mathbf{A}_k \in \mathbb{R}^{4\times 4}$$

$$\mathbf{A}_k \mathbf{x}_k = \mathbf{b}_k + \boldsymbol{\varepsilon}_k, \text{ where } \mathbf{x}_k = \mathbf{p}_k s_k, \ s_k \in [-1,+1];$$

$$\boldsymbol{\varepsilon}_k \in N(0,\sigma^2), \ \mathbf{p}_k \text{ is subject to } \left\|\mathbf{A}_k \hat{\mathbf{x}}_k - \mathbf{b}_k\right\|_2 < \sigma^2.$$

Homework

3. Solve noisy equation sample by sample and define probability of right solution averaged over all samples.
// default vector $\mathbf{p}_k = \frac{1}{\sqrt{4}}(1,1,1,1)^T$

$\|\mathbf{p}_k\|_2 = 1 \quad (!)$

4. Repeat item #3 for $\mathbf{p}_k = \mathbf{u}_k^{(1)}$- eigenvector of matrix $\mathbf{A}_k$, corresponded to the largest singular value.

5. How stochastic information about additive noise $\boldsymbol{\varepsilon}_k$ can be utilized for minimization of error probability?..

Homework

http://lyashev.weebly.com/notes/linear-algebra-issues-in-wireless-communications

I. Generate 2000 samples of matrix **A** and keep in memory for all numerical experiments.
II. Set mapping vector **p** (two ways).
III. Map one-bit symbol (-1/+1) from 1x1 to 4x1: **x** = **p***s*.
IV. Compute **b** = **Ax.**
V. Add gaussian noise (sigma is variation parameter for analysis):
 b’ = **b** + **n**.
VI. Solve noise equation: **Ax’** = **b**’, that define **x**’ as estimation value.
VII. Find a way to define *s’* by known **p** and estimated **x’**.
VIII. Check: how many *s’* = *s* ?...
 P_{error} = 1 - <right **s’**> / <number of samples>
IX. P_{error} can be defined as function of deviation of noise (sigma).