ECE 650 – Lecture #7
Random Vectors: 2\textsuperscript{nd} Moment Theory, continued

[Ref: R. Scholtz Lecture Notes, USC]

D. van Alphen
Lecture Overview

- Review of 2\textsuperscript{nd}-Moment Theory for Real R Vectors
- Causal Linear Transformations (for Coloring Noise)
- Cholesky Decomposition
- Expanding the Covariance Matrix: Spectral Resolution
- Mean-squared Length of Random (Column) Vectors
- Directional Preference of Random Vectors
  - Scholtz Peanut
  - Directional Preference of White Noise
- Whitening Process for Colored Noise
2\textsuperscript{nd}-Moment Theory & Transforms: Review

- Recall: default random vectors are real column vectors
- Correlation matrix for R Vector \( \mathbf{X} \): \( R_{\mathbf{X}} = E[\mathbf{X}\mathbf{X}^T] \)
- Covariance Matrix for R Vector \( \mathbf{X} \): \( C_{\mathbf{X}} = E[(\mathbf{X} - \eta_{\mathbf{X}})(\mathbf{X} - \eta_{\mathbf{X}})^T] = R_{\mathbf{X}} - \eta_{\mathbf{X}}\eta_{\mathbf{X}}^T \)
- If \( \mathbf{Y} = \mathbf{H}\mathbf{X} \), then: \( E[\mathbf{Y}] = \mathbf{H} E[\mathbf{X}] \)
  and \( R_{\mathbf{Y}} = \mathbf{H} R_{\mathbf{X}} \mathbf{H}^T, \quad C_{\mathbf{Y}} = \mathbf{H} C_{\mathbf{X}} \mathbf{H}^T \)
- \( C_{\mathbf{Y}} = \mathbf{E} \Lambda \mathbf{E}^T \) (\( \mathbf{E} \): e-vector columns, \( \Lambda \): e-values on diag.)
- To generate colored noise vector \( \mathbf{Z} \) with mean vector \( \eta_{\mathbf{Z}} \), and covariance \( C_{\mathbf{Z}} \), starting with elementary white noise vector \( \mathbf{W} \):
  - Perform linear transform: \( \mathbf{Z} = \mathbf{H}\mathbf{W} + \mathbf{C} \)
    - where \( \mathbf{C} = \eta_{\mathbf{Z}} \), and where \( \mathbf{H} = \mathbf{E} \Lambda^{1/2} \);
Linear Transformations & Causality

- Consider the vector $Y = HX$; i.e.,

\[
Y = \begin{bmatrix}
Y_1 \\
Y_2 \\
\vdots \\
Y_m
\end{bmatrix} \begin{bmatrix}
h_{11} & h_{12} & \cdots & h_{1n} \\
h_{21} & h_{22} & \cdots & h_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
h_{m1} & h_{m2} & \cdots & h_{mn}
\end{bmatrix} \begin{bmatrix}
X_1 \\
X_2 \\
\vdots \\
X_n
\end{bmatrix}
\]

where the subscripts imply temporal order; i.e.,

input $X_1$ \{ comes before \}

output $Y_1$ \{ input $X_2$ \}

output $Y_2$

- Recall for a **causal** system, output $Y_i$ can only depend on current and previous input values ($X_j, j \leq i$).
Linear Transformations & Causality

\[ Y_1 = h_{11} X_1 + h_{12} X_2 + \ldots + h_{1n} X_n \]

0 for a causal system

\[ Y_2 = h_{21} X_1 + h_{22} X_2 + h_{23} X_3 \ldots + h_{1n} X_n \]

0 for a causal system

\[ \vdots \]

\[ Y_n = h_{n1} X_1 + h_{n2} X_2 + \ldots + h_{nn} X_n \]

No 0’s necessary for a causal system

Assuming m = n, so H is square
Claim: For a causal system, \( H \) needs to be lower triangular (all 0’s above the diagonal).

Repeating the example from Lecture 6: say we want to simulate a 0-mean vector \( \mathbf{Y} \) with covariance matrix:

\[
\mathbf{C}_\mathbf{Y} = \begin{bmatrix}
1 & -0.5 & -0.5 \\
-0.5 & 1 & -0.5 \\
-0.5 & -0.5 & 1
\end{bmatrix}
\]

Assume that we have access to elementary white vector \( \mathbf{W} \); find causal \( H \) such that \( \mathbf{Y} = \mathbf{HW} \).

D. van Alphen
Linear Transformations & Causality, continued

- We still need covariance matrix factorization: \( C_Y = HH^T \)
- But now \( H \) must be lower triangular.
- One approach: brute force factorization, with algebra:

\[
C_Y = \begin{bmatrix}
1 & -0.5 & -0.5 \\
-0.5 & 1 & -0.5 \\
-0.5 & -0.5 & 1 \\
\end{bmatrix}
\begin{bmatrix}
h_{11} & 0 & 0 \\
h_{21} & h_{22} & 0 \\
h_{31} & h_{32} & h_{33} \\
\end{bmatrix}
\begin{bmatrix}
h_{11} & h_{21} & h_{31} \\
h_{22} & h_{32} \\
h_{33} \\
\end{bmatrix}
\]

Using the row-1 equations from \( C_Y \):

\[\Rightarrow h_{11}^2 = 1 \quad \Rightarrow h_{11} = 1 \quad \text{(using the positive solution)}\]

and \( h_{11} h_{21} = -0.5 \quad \Rightarrow h_{21} = -0.5 \)

and \( h_{11} h_{31} = -0.5 \quad \Rightarrow h_{31} = -0.5 \)
Using the row-2 equations from $C_Y$:

\[
\Rightarrow h_{11} h_{21} = -0.5 \quad \text{(redundant)}
\]

and $h_{21}^2 + h_{22}^2 = 1 \Rightarrow h_{22} = \sqrt{3}/2 \quad \text{(using pos. solution)}$

and $h_{21} h_{31} + h_{22} h_{32} = -0.5 \Rightarrow h_{32} = -\sqrt{3}/2$

Using row-3 equations from $C_Y$ (to obtain $h_{33}$):

\[
h_{31}^2 + h_{32}^2 + h_{33}^2 = 1 \quad \Rightarrow h_{33} = 0
\]
Linear Transformations & Causality, continued

• Repeating:

\[ C_Y = \begin{bmatrix}
1 & -0.5 & -0.5 \\
-0.5 & 1 & -0.5 \\
-0.5 & -0.5 & 1 
\end{bmatrix} = \begin{bmatrix}
h_{11} & 0 & 0 \\
h_{21} & h_{22} & 0 \\
h_{31} & h_{32} & h_{33} 
\end{bmatrix} \begin{bmatrix}
h_{11} & h_{21} & h_{31} \\
h_{21} & h_{22} & h_{32} \\
h_{31} & h_{32} & h_{33} 
\end{bmatrix} \]

Thus,

\[ H = \begin{bmatrix}
1 & 0 & 0 \\
-1/2 & -\sqrt{3}/2 & 0 \\
-1/2 & -\sqrt{3}/2 & 0 
\end{bmatrix} \]

lower triangular

⇒ causal

Using MATLAB to verify \( HH^T = C_Y \):

\[
\begin{align*}
>> H &= \begin{bmatrix}
1 & 0 & 0 \\
-0.5 & \sqrt{3}/2 & 0 \\
-0.5 & \sqrt{3}/6 & \sqrt{6}/3 
\end{bmatrix}; \\
>> H^*H^T &= \text{(answer came back as } C_Y) 
\end{align*}
\]
Cholesky Decomposition  
(Another Way to Factor Covariance Matrices)

• This technique (Cholesky Decomposition) only works for positive definite* covariance matrices
  – Pos. Def. ⇔ _____________________________

• MATLAB command to factor covariance matrix C using Cholesky Decomposition: chol(C)

• chol(X) uses only the diagonal and upper triangular part of X.
  – The lower triangular is assumed to be the (complex conjugate) transpose of the upper.
    • O.K. for covariance matrices, since they are symmetric (for real R. Vectors)

* Recall that covariance matrices must be non-negative definite (NND), but not necessarily positive definite. (In the previous example, $C_Y$ had a zero-eigenvalue, and was hence not positive definite.)
Cholesky Decomposition, continued

- If $C$ is positive definite, then $R = \text{chol}(C)$ produces an upper triangular $R$ so that $R^\prime \cdot R = C$. ($R^\prime$ will be lower triangular)

- Example: Let $C = \begin{bmatrix} 1 & 1/2 & 0 \\ 1/2 & 1 & 1/2 \\ 0 & 1/2 & 1 \end{bmatrix}$

- MATLAB Code:

  ```matlab
  >> C = [1 .5 0; .5 1 .5; 0 .5 1];
  >> R = chol(C)
  R =
  1.0000    0.5000         0
  1.0000    0.8660    0.5774
  0        0    0.8165
  >> H = R';
  >> H*H'
  ans =
  1.0000    0.5000         0
  0.5000    1.0000    0.5000
  0        0    1.0000
  = C, ✔
  ```
Spectral Resolution of the Covariance Matrix

- Let $C$ be a covariance matrix for some real $\mathbb{R}$ vector, with eigenvectors $e_i$ and corresponding eigenvalues $\lambda_i$.

- Then $C = (E \Lambda)E^T = \begin{bmatrix} \lambda_1 e_1 & \lambda_2 e_2 & \cdots & \lambda_n e_n \end{bmatrix}$

\[ C = \sum_{i=1}^{n} \lambda_i e_i e_i^T \quad \text{(the spectral resolution of } C) \]

**Spectrum of $C$:** set of distinct eigenvalues, $\{\lambda_i\}$

- Each component matrix $e_i e_i^T$ is a projection matrix*.

---

*A projection matrix $P$ has 2 properties: $P^2 = P$, and $P = P^T$ (Symmetric). (The product $PX$ “projects” $X$ onto the column space of $P$).
Spectral Resolution (or Decomposition)

Example

- Consider covariance matrix $C = \begin{bmatrix} 7 & 2 \\ 2 & 4 \end{bmatrix}$.

- Eigenvalues: $\lambda_1 = 8$, $\lambda_2 = 3$; eigenvalues $e_1 = \begin{bmatrix} 2 \\ \sqrt{5} \end{bmatrix}$, $e_2 = \begin{bmatrix} 1 \\ \sqrt{5} \end{bmatrix}$

\[
\lambda_1 e_1 e_1^+ = 8 \begin{bmatrix} 2 \\ \sqrt{5} \\ 1 \end{bmatrix} \begin{bmatrix} 2 \\ \sqrt{5} \\ 1 \end{bmatrix} = 8 \begin{bmatrix} 4/5 & 2/5 \\ 2/5 & 1/5 \end{bmatrix}
\]

\[
\lambda_2 e_2 e_2^+ = 3 \begin{bmatrix} 1 \\ \sqrt{5} \\ -2 \end{bmatrix} \begin{bmatrix} 1 \\ \sqrt{5} \\ -2 \end{bmatrix} = 3 \begin{bmatrix} 1/5 & -2/5 \\ -2/5 & 4/5 \end{bmatrix}
\]

$\Rightarrow C = \lambda_1 e_1 e_1^+ + \lambda_2 e_2 e_2^+ = 8 \begin{bmatrix} 4/5 & 2/5 \\ 2/5 & 1/5 \end{bmatrix} + 3 \begin{bmatrix} 1/5 & -2/5 \\ -2/5 & 4/5 \end{bmatrix}$
Mean-squared Length of Random Vectors

- Recall for deterministic vectors $\mathbf{Y}$: $|\mathbf{Y}|^2 = \mathbf{Y}^T \mathbf{Y}$

  e.g., for $\mathbf{Y} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$, $\mathbf{Y}^T \mathbf{Y} = [3 \ 4] \begin{bmatrix} 3 \\ 4 \end{bmatrix} = 25 = |\mathbf{Y}|^2$

  $\Rightarrow E[|\mathbf{Y}|^2] = E\{\mathbf{Y}^T \mathbf{Y}\} = E\{ [Y_1 \ Y_2 \ \cdots \ Y_n] \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} \}$

  $= E[Y_1^2 + Y_2^2 + \cdots + Y_n^2] = \sum_{i=1}^{n} R_{ii} = \text{tr}(\mathbf{R}_Y) = \sum_{i=1}^{n} \lambda_i$

**Conclusion:** Mean-squared length of $\mathbf{Y}$ is: $E\{|\mathbf{Y}|^2\} = \text{tr}(\mathbf{R}_Y) = \sum_{i=1}^{n} \lambda_i$

---

Linear Algebra Note: The **trace** of a matrix is the sum of its diagonal elements, which also equals the sum of its eigenvalues.
Directional Preference of R. Vectors

• Recall if \( \mathbf{X} \) is a 3-D real vector, and \( \mathbf{U}_x, \mathbf{U}_y, \) and \( \mathbf{U}_z \) are orthonormal basis functions for \( \mathbb{R}^3 \), then

\[
(\mathbf{X} \cdot \mathbf{U}_x) \mathbf{U}_x
\]

is the projection of vector \( \mathbf{X} \) onto basis function \( \mathbf{U}_x \), or the projection of \( \mathbf{X} \) in the direction \( \mathbf{U}_x \).
Directional Preference, continued

- Now let $Y_0$ be the centered version of real random vector $Y$; 
  - let $b$ be any real unit-length vector: $|b| = 1$; then

$$ (Y_0 \cdot b) b = b^T Y_0 b $$

is the **projection** of $Y_0$ in the direction of $b$.

- Thus, the **mean-squared length** of the projection of $Y_0$ onto $b$ is:

$$ E[ |b^T Y_0 b|^2 ] = E[ |b^T Y_0|^2 |b|^2 ] = E[ |b^T Y_0 Y_0^T b| ] $$

$$ = b^T R_{Y_0} b = b^T C_Y b $$

Use to find the MS length of a R Vector in any direction (specified by $b$)

**LA Factoid:** $a^T b = b^T a = \text{dot}(a, b)$
Directional Preference, continued

- Repeating: For arbitrary real (column) random vector \( \mathbf{Y} \), the **MS length of \( \mathbf{Y} \)** in the direction of unit-length vector \( \mathbf{b} \) is:

\[
\mathbf{b}^\mathsf{T} C_\mathbf{Y} \mathbf{b}
\]

- **Question:** What choice of \( \mathbf{b} \) gives the largest mean-squared projection?

  - **Claim:** If \( \mathbf{e}_{\max} \) is the eigenvector corresponding to the largest eigenvalue \( \lambda_{\max} \) (of \( C_\mathbf{Y} \)), and \( \mathbf{e}_{\min} \) is the eigenvector corresponding to the smallest eigenvalue \( \lambda_{\min} \) (of \( C_\mathbf{Y} \)), then

\[
\lambda_{\min} = \mathbb{E}\{ |\mathbf{e}_{\min}^\mathsf{T} \mathbf{Y}_0|^2 \} \leq \mathbb{E}\{ |\mathbf{b}^\mathsf{T} \mathbf{Y}_0|^2 \} \leq \mathbb{E}\{ |\mathbf{e}_{\max}^\mathsf{T} \mathbf{Y}_0|^2 \} = \lambda_{\max}
\]

**Preferred direction** of \( \mathbf{R} \). Vector \( \mathbf{Y} \), given by the vector with the largest mean-squared projection, is \( \mathbf{e}_{\max} \).
Directional Preference: Example

- Let $\mathbf{Z}$ be a zero-mean vector with covariance matrix:

$$\mathbf{C}_Z = \begin{bmatrix} 3 & \sqrt{2} \\ \sqrt{2} & 4 \end{bmatrix}$$

- Using MATLAB to find e-values, e-vectors:

```matlab
>> CZ = [3 sqrt(2); sqrt(2) 4];
>> [V Lambda] = eig(CZ)
V =
    -0.8165    0.5774
    0.5774    0.8165
Lambda =
    2.0000         0
    0    5.0000
```

\[ e_{\text{max}} = \begin{bmatrix} .5774 \\ .8165 \end{bmatrix}, \quad \lambda_{\text{max}} = 5 \]
\[ e_{\text{min}} = \begin{bmatrix} -.8165 \\ .5774 \end{bmatrix}, \quad \lambda_{\text{min}} = 2 \]
Directional Preference: Example

- Repeating: $Z$ is zero-mean, with $C_Z = \begin{bmatrix} 3 & \sqrt{2} \\ \sqrt{2} & 4 \end{bmatrix}$

- Now generate some unit-length vectors $b$, at various angles $\theta$; then find the RMS length of $Z$ in each of these directions.

  - Unit-length vector, $b$, at angle $\theta$, can be found: $b = \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}$.

MATLAB function m-file:

```matlab
function B = unit_length_bees(theta)
% This function generates unit-length column vectors, b, at
% angles specified (in degrees) by input column vector theta.
% The column vectors b are stored in matrix B.
B = zeros(2, length(theta));
for icount = 1: length(theta)
    B(:,icount) = [cosd(theta(icount)); sind(theta(icount))];
end
B, scatterplot(B') % scatterplot requires 2-column input
```
Directional Preference Example, continued

Resulting Scatterplot for unit-length \( \mathbf{b} \) Vectors:

Finding the **MS length of \( \mathbf{Z} \)** in the direction of 0° \( \Leftrightarrow \mathbf{b} = [1 \ 0]' \):

\[
\mathbf{b}' \mathbf{C}_Z \mathbf{b} = [1 \ 0] \begin{bmatrix} 3 & \sqrt{2} \\ \sqrt{2} & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 & \sqrt{2} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 3
\]

RMS length: \( \text{sqrt}(3) \)
MATLAB Code:

```matlab
function RMS = rms_length(theta, cov)
% This function finds the RMS length
% of 2-dimensional column vectors with
% covariance matrix = cov, in various
% directions specified (in degrees) in
% input row vector theta.

B = unit_length_bees(theta)
scaled_B = zeros(2, length(B))
RMS = zeros(1,length(theta))
for icount = 1:length(theta)
    RMS(icount) = sqrt(B(:,icount)'*cov * B(:,icount));
scaled_B(:,icount) = RMS(icount)*B(:,icount);
end
scaled_B, scatterplot(scaled_B')
```

RMS-length in Directions, \( b \)

Scatter plot
**White Noise: No Directional Preference**

- **Claim:** *White noise has no directional preference.*
- **Verification:** Let $\mathbf{W}$ be a white column vector, so the covariance matrix is:
  
  $$C_{\mathbf{W}} = \sigma^2 \mathbf{I}$$

  - Directional preference of $\mathbf{W}$ is given by the eigenvector $\mathbf{e}_{\text{max}}$ corresponding to the largest eigenvalue of $C_{\mathbf{W}}$.

  - To find the eigenvalues* of $C_{\mathbf{W}} = \sigma^2 \mathbf{I}$:

    $$\det(\sigma^2 \mathbf{I} - \lambda \mathbf{I}) = 0 \Rightarrow \lambda = \sigma^2 \text{ (for all } \mathbf{i})$$

    $$\Rightarrow \lambda_{\text{max}} = \lambda_{\text{min}} \Rightarrow \text{No directional preference}$$

* Also note: diagonal matrices always have their e-values on the diagonal.
Scatterplot for White Noise Example (Uniform R. Variables)

- Generating 1000 2-dimensional R Vectors:
- MATLAB Code:

```matlab
>> X1 = unifrnd(-1,1,1000,1);
>> X2 = unifrnd(-1,1,1000,1)
>> X = [X1 X2];
>> scatterplot(X)
```

Random variables $X_1$ and $X_2$ are each uniform on (-1,1);

For scatterplot, $X$ must be a row vector; here we generate 1000 realizations of $X$.
Scatterplot for White Noise Example (Gaussian R. Variables)

- Generating 1000 2-dimensional R Vectors:

- MATLAB Code:
  - X1 = normrnd(0,1,1000,1);
  - X2 = normrnd(0,1,1000,1);
  - X = [X1 X2]
  - scatterplot(X)

Random variables $X_1$ and $X_2$ are each $N(0,1)$;

For scatterplot, $X$ must be a row vector; here we generate 1000 realizations of $X$.

No directional preference
Whitening Colored Noise (Going the Other Way)

• Question: How would we “whiten” a zero-mean noise vector $\mathbf{N}$ with covariance matrix $\mathbf{C}_N = \mathbf{H} \mathbf{H}^T$ (where $\mathbf{H} = \mathbf{E} \sqrt{\Lambda}$) ?

\[
\mathbf{N} \xrightarrow{\mathbf{G}} \mathbf{W} = \mathbf{G} \mathbf{N}
\]

Cov. $\mathbf{C}_N$, $\eta_N = 0$

Elt. White: Cov. $\mathbf{C}_N$, $\eta_N = 0$

Problem: Find the required transformation, $\mathbf{G}$.

• We know: $\mathbf{C}_W = \mathbf{G} \mathbf{C}_N \mathbf{G}^T = \mathbf{G}(\mathbf{H} \mathbf{H}^T)\mathbf{G}^T = (\mathbf{G}\mathbf{H})(\mathbf{H}^T\mathbf{G}^T) = (\mathbf{G}\mathbf{H})(\mathbf{G}\mathbf{H})^T$

• We want $\mathbf{G}$ such that: $\mathbf{C}_W = (\mathbf{G}\mathbf{H})(\mathbf{G}\mathbf{H})^T = \mathbf{I}$

$\Rightarrow$ Choose $\mathbf{G} = \mathbf{H}^{-1} = (\mathbf{E}\sqrt{\Lambda})^{-1} = \sqrt{\Lambda}^{-1} \mathbf{E}^T$

(Whitening Filter)
Property Summary for Symmetric NND Matrices (e.g., Covariance Matrices for Real R. Vectors)

Let A be Symmetric, NND; then

1. The eigenvalues of A are non-negative real numbers, which can be ordered: \( \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \)

2. The eigenvectors \( \mathbf{e}_i \) of A can be chosen to be orthonormal, so:
\[
\mathbf{e}_i^T \mathbf{e}_i = 1, \quad \mathbf{e}_i^T \mathbf{e}_j = 0 \quad (i \neq j)
\]

3. (Spectral Resolution) The original matrix A can be written as a weighted sum (with eigenvalue weights):
\[
A = \lambda_1 \mathbf{e}_1^T \mathbf{e}_1 + \lambda_2 \mathbf{e}_2^T \mathbf{e}_2 + \ldots + \lambda_n \mathbf{e}_n^T \mathbf{e}_n
\]
\[
\approx \lambda_1 \mathbf{e}_1^T \mathbf{e}_1 + \lambda_2 \mathbf{e}_2^T \mathbf{e}_2 + \ldots + \lambda_k \mathbf{e}_k^T \mathbf{e}_k, \quad k < n
\]

(used for data compression)