Spectrum Estimation & Adaptive SP
Complex–Valued Adaptive Filters

Danilo Mandic

room 813, ext: 46271

Department of Electrical and Electronic Engineering
Imperial College London, UK

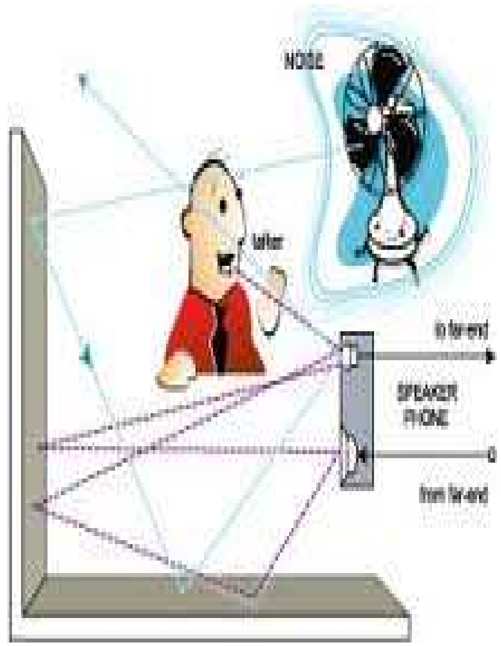
d.mandic@imperial.ac.uk, URL: www.commsp.ee.ic.ac.uk/∼mandic

Outline:

really, to de-mystify and make rigorous several concepts

- Recursive solution to the Wiener filter $\looparrowright$ Recursive Least Squares
- Multidimensional and multichannel sensors – a unified approach to adaptive filtering of such signals
- Circularity – a unique signature of bivariate signals $\looparrowright$ second order circularity (properness) $\looparrowright$ augmented complex statistics
- Duality between the processing in $\mathbb{R}^2$ and $\mathbb{C}$ (isomorphism)
- The issue of complex gradient $\looparrowright$ $\mathbb{CR}$ calculus
- Covariance, pseudocovariance, and widely linear models
- Complex least mean square (CLMS) and augmented CLMS (ACLMS)
- Multivariate adaptive filters (any number of data channels)
- Applications: communications, radar and sonar, target tracking, renewable energy, smart grid

Where can we apply multidimensional (multichannel) adaptive filters?



Brain Computer Interface

Decoding brain activity

to control computers

Spect. Est., ASP

Audio applications

Echo cancellation

Multiple mics and speakers

Spect. Est,

Radar and sonar

Trajectory tracking

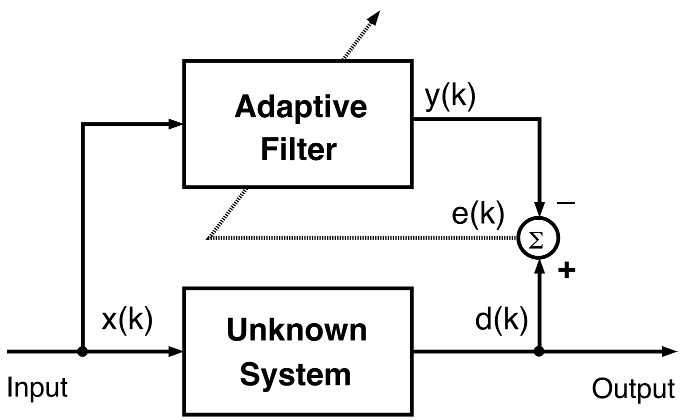
Radar, sonar: Manouver prediction

Spect. Est., ASP.

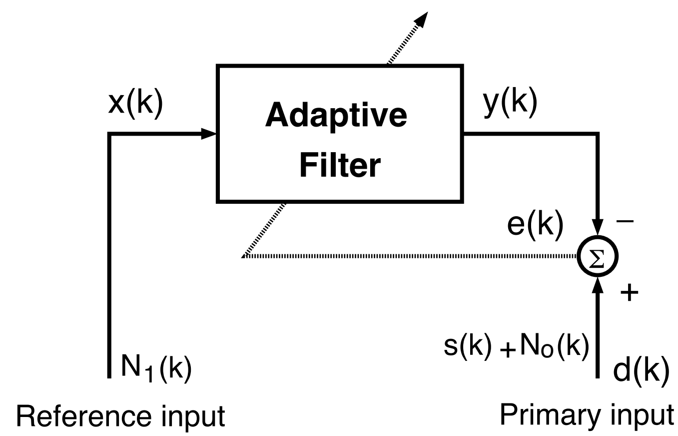
Also: seismics and oil industry, finance, social groups behaviour, economics

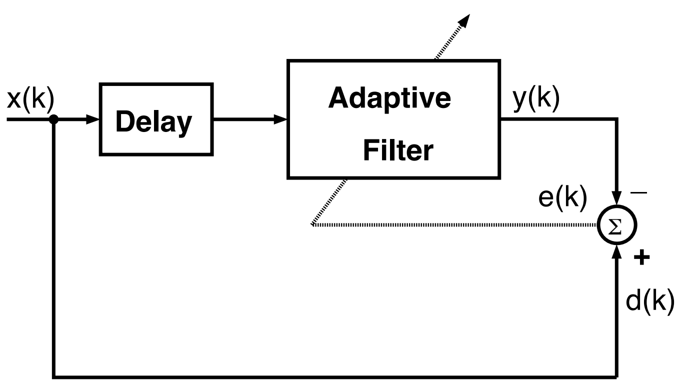
And we can use them within our usual adaptive filtering configurations!

System Identification

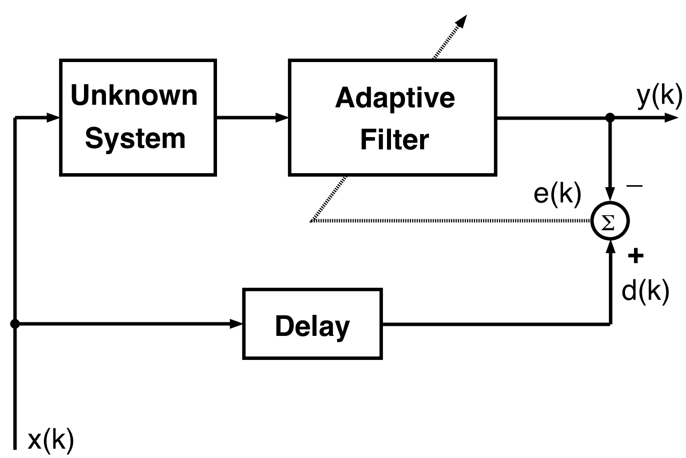


Noise Cancellation





Adaptive Prediction



Inverse System Modelling

Let us first look at the big picture: There are two main families of adapt. filt. algorithms

- **Gradient descent based methods** (e.g. the Least Mean Square (LMS)) provide a recursive form of an **approximate minimisation** of the mean square error (MSE), where the cost function

$$J \sim E\{e^2(n)\} \quad \text{for LMS} \quad J(n) \sim e^2(n) \qquad \textbf{stochastic}$$

Knowledge of the autocorrelation of the input process and cross-correlation between the input an teaching signal required.

Rapid convergence or sufficiently small excess MSE not guaranteed.

- **Least squares (LS) techniques** are based on the **exact minimisation** of a sum of instantaneous squared errors

$$J(n) = \sum_{i=0}^{n} e^2(i) = \sum_{i=0}^{n} \left| d(i) - \mathbf{x}^T(i)\mathbf{w}_n \right|^2 \qquad \textbf{deterministic}$$

Deterministic cost function $\looparrowright$ no statistical information about $\mathbf{x}$ and d!

☞ The LS error depends on particular $x(n)$ and $d(n)$ $\looparrowright$ for different signals we obtain different filters.

☞ The MSE does not depend on particular $x(n)$ and $d(n)$, just on their statistics $\looparrowright$ produces equal weights for signals with the same statistics.

The Recursive Least Squares (RLS) algorithm

(recursive solution to the "deterministic" Wiener filtering problem)

Aim: For a filter of order N, find the weight vector $\mathbf{w}_n = \mathbf{w}_{opt}(n)$ which minimizes the sum of squares of output errors up until the time instant n

$$\mathbf{w}_n = \mathbf{w}_{opt}^{LS} \text{ over n time instants} \quad \Rightarrow \quad \mathbf{w}_n = arg \; \min_{\mathbf{w}} \sum_{i=0}^{n} \left| d(i) - \mathbf{x}^T(i)\mathbf{w}_n \right|^2$$

(the summation over i is performed for the **latest** set of coefficients $\mathbf{w}_n$)

☞ Notice, the weights $\mathbf{w}_n$ are **held constant** over the whole observation $[0, n]$ – similar to the Wiener setting (but a deterministic cost function)

To find $\mathbf{w}_n$: set $\nabla_{\mathbf{w}} J(n) = \mathbf{0}$ to solve for $\mathbf{w}_{opt}(n) = \mathbf{w}_n = \mathbf{R}^{-1}(n)\mathbf{p}(n)$

◦ **Careful here, as:** $\mathbf{R}(n) = \sum_{i=0}^{n} \mathbf{x}(i)\mathbf{x}^T(i), \quad \mathbf{p}(n) = \mathbf{r}_{d\mathbf{x}} = \sum_{i=0}^{n} d(i)\mathbf{x}(i)$

that is, $\mathbf{R}(n)$, $\mathbf{p}(n)$ are purely ***deterministic*** (*cf.* $\mathbf{R} = E\{\mathbf{x}\mathbf{x}^T\}, \mathbf{p} = E\{d\mathbf{x}\}$ in Wiener filt.)

Recursive least squares (RLS). The aim is to recursively update:

(a) $J(n+1) = J(n) + |e(n+1)|^2$ (b) $\mathbf{p}(n+1) = \mathbf{p}(n) + d(n+1)\mathbf{x}(n+1)$

(c) $\mathbf{R}(n+1) = \mathbf{R}(n) + \mathbf{x}(n+1)\mathbf{x}^T(n+1)$

The weight update: $\mathbf{w}_{n+1} = \mathbf{w}_{opt}(n+1) = \mathbf{w}(n+1) = \mathbf{R}^{-1}(n+1)\mathbf{p}(n+1)$

Ways to compute a matrix inverse

while direct, this technique is computationally wasteful, $N^3 + 2N^2 + N$ multiplic.

There are many ways (Neumann series, geometric series). For $|a| < 1$

$$\sum_{i=0}^{\infty} a^i = \frac{1}{1-a}, \qquad \text{now swap} \quad \mathrm{b} = 1 - \mathrm{a} \quad \text{to get} \quad \sum_{\mathrm{i}=1}^{\infty} (1-\mathrm{b})^{\mathrm{i}} = \frac{1}{\mathrm{b}}$$

$\looparrowright$ **we have found the inverse of b via a series in $1-b$.**

Generalization to matrices: Consider a nonsingular matrix $\mathbf{A}$, to yield

$$\lim_{i = \rightarrow \infty} (\mathbf{I} - \mathbf{A})^i = \mathbf{0} \quad \Rightarrow \quad \mathbf{A}^{-1} = \sum_{i=0}^{\infty} (\mathbf{I} - \mathbf{A})^i$$

Then, for an invertible matrix $\mathbf{X}$, and its inverse $\mathbf{A}$, we can write

$$\lim_{i \rightarrow \infty} \left(\mathbf{I} - \mathbf{X}^{-1}\mathbf{A}\right)^i = \lim_{i \rightarrow \infty} \left(\mathbf{I} - \mathbf{A}\mathbf{X}^{-1}\right)^i = \mathbf{0} \quad \Rightarrow \quad \mathbf{A}^{-1} = \sum_{i=0}^{\infty} \Big(\mathbf{X}^{-1}(\mathbf{X} - \mathbf{A})\Big)^i \mathbf{X}^{-1}$$

If $(\mathbf{A} - \mathbf{X})$ has rank 1, we finally have: $\quad \mathbf{A}^{-1} = \mathbf{X}^{-1} + \dfrac{\mathbf{X}^{-1}\big(\mathbf{X} - \mathbf{A}\big)\mathbf{X}^{-1}}{1 - \mathrm{tr}\Big(\mathbf{X}^{-1}\big(\mathbf{X} - \mathbf{A}\big)\Big)}$

The RLS formulation $\looparrowright$ $\mathbf{R}^{-1}$ calculation is $\mathcal{O}(N^3)$

(we use the standard matrix inversion lemma, also known as Woodbury's identity)

Key to RLS: all we have to do is to employ a recursive update of $\mathbf{R}^{-1}$

Start from the matrix inversion lemma (also known as ABCD lemma)

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B\big(DA^{-1}B + C^{-1}\big)^{-1}DA^{-1}$$

and set $A = \mathbf{R}(n), B = \mathbf{x}(n+1), C = 1, D = \mathbf{x}^T(n+1)$, to give

$$\mathbf{R}(n+1) = \mathbf{R}(n) + \mathbf{x}(n+1)\mathbf{x}^T(n+1) = A + BCD$$

Then $\mathbf{R}^{-1}(n+1)$ is given by

$$\mathbf{R}^{-1}(k+1) = \mathbf{R}^{-1}(k) - \frac{\mathbf{R}^{-1}(k)\mathbf{x}(k+1)\mathbf{x}^T(k+1)\mathbf{R}^{-1}(k)}{\mathbf{x}^T(k+1)\mathbf{R}^{-1}(k)\mathbf{x}(k+1)+1} \quad \looparrowright \quad \mathcal{O}(N^2)$$

$\looparrowright$ **we never compute $\mathbf{R}(n+1)$ or $\mathbf{R}^{-1}(n)$ directly, but recursively**

The optimal RLS weight vector: $\mathbf{w}_{opt}(n+1) = \mathbf{w}(n+1) = \mathbf{R}^{-1}(n+1)\mathbf{p}(n+1)$

Minimum LSE: $J_{min}^{LS} = \| \mathbf{d}(n) \|_2^2 - \mathbf{r}_{dx}^T(n)\mathbf{w}_n$

where $\mathbf{r}_{dx}(n) = \mathbf{p}(n), \mathbf{d}(n) = [d(0), \ldots, d(n)]^T$.

RLS: Adaptive Noise Cancellation

rlsdemo in Matlab

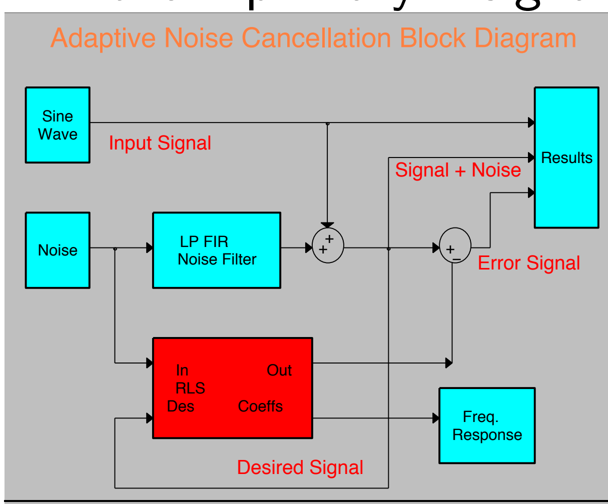
Similar scenario to noise-cancelling headphones.

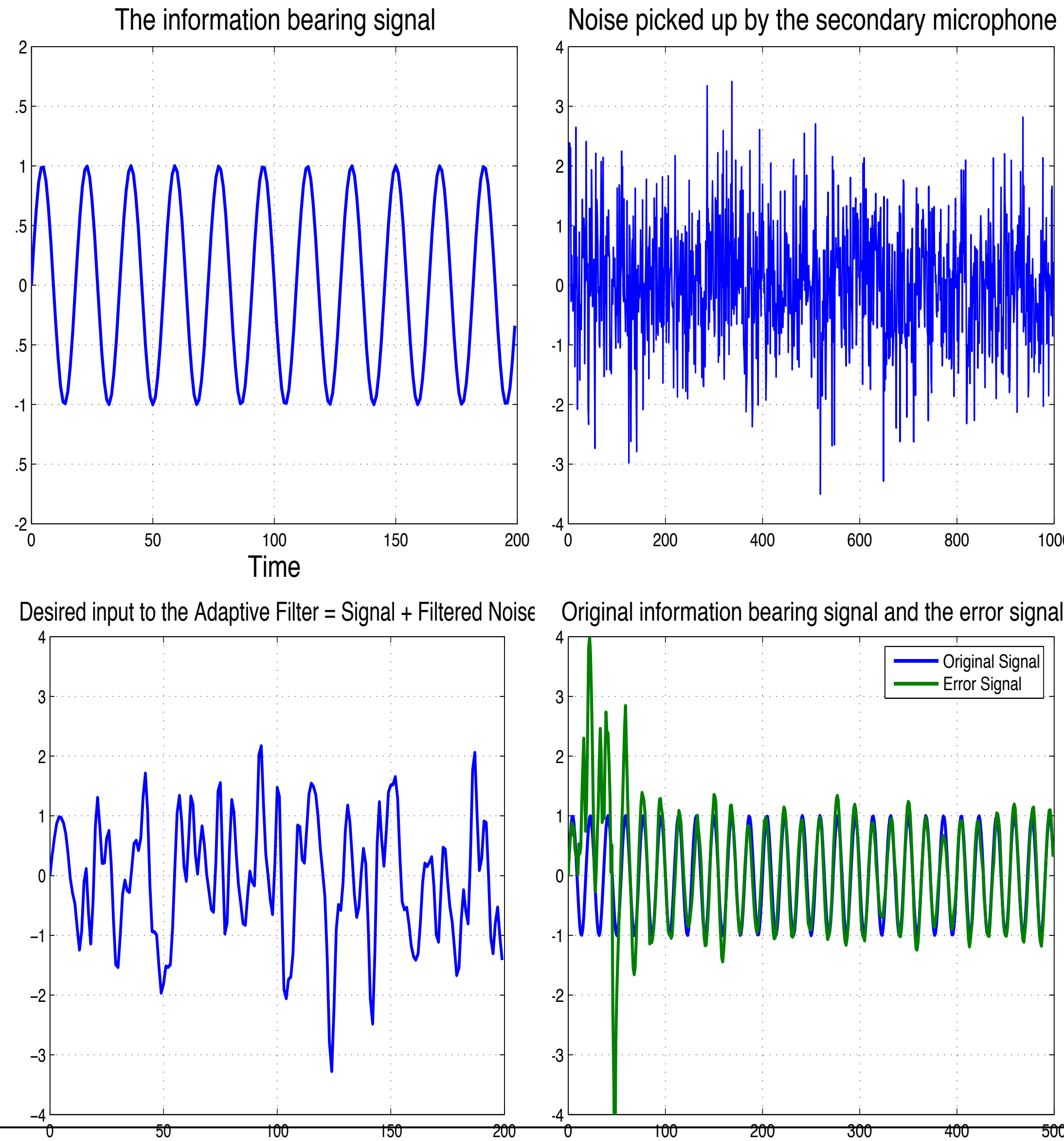
Primary signal:

$$\sin(n) + q(n)$$

Reference signal:

any noise correlated with the noise $q(n)$ in the primary signal





Some things to remember about RLS

- Results are exactly the same as for the normal least squares $\looparrowright$ no approximation involved

- RLS does not perform matrix inversion at any stage $\looparrowright$ it calculates is recursively

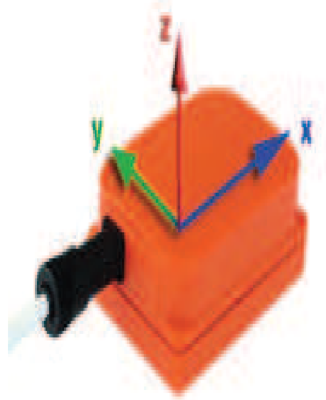
- There is no explicit notion of learning rate

- RLS guarantees convergence to the optimal weight vector

- Variants of RLS (sliding window, forgetting factor) deal with pragmatic issues (nonstationarity etc)

- The price to pay is increased computational complexity

Complex adaptive signal processing: Applications



Renewable Energy

2D and 3D anemometers

control of wind turbine

Body motion sensor

3D - position, gyroscope, speed

gait, biometrics

Wearable technologies

Biomechanics

virtual reality

Wind sensors - 2D and 3D anemometers

History of mathematical notation

⊛ $9th$ century Al Kwarizimi's *Algebra* - solutions descriptive rather than in form of equations

⊛ $16th$ century - G. Cardano *Ars Magna* - unknowns denoted by single roman letters

⊛ Descartes (1630-s) established general rules

– lowercase italic letters at the beginning of the alphabet for unknown constants a, b, c, d

– lowercase italic letters at the end of the alphabet for unknown variables x, y, z

⊛ $\sqrt{-1} = \imath$ – Gauss 1830s, boldface letters for vectors $\mathbf{x}, \mathbf{v}$ - Oliver Heaviside

⊛ Hence $\quad ax^2 + by + cz = 0$

More detail: F. Cajori, *History of Mathematical Notations*, 1929

Why modelling in $\mathbb{C}$?

- ☐ Complex signals by design (communications, analytic signals, equivalent baseband represenation to eliminate spectral redundancy)
- ☐ By convenience of representation (radar, sonar, wind field)
- ☐ Problem: Different algebra (no ordering - operator "$\leq$" makes no sense!), and the notion of pdf has to be induced
- ☐ Problem: Special form of nonlinearity (the only continuously differentiable function in $\mathbb{C}$ is a constant (Liouville theorem)
- ☐ Solution: Special statistics – augmented complex statistics (started in mathematics in 1992)
- ☐ We can differentiate between several kinds of noises (doubly white circular with various distributions $n_r \perp n_i$ & $\sigma^2_{n_r} = \sigma^2_{n_i}$, doubly white noncircular $n_r \perp n_i$ & $\sigma^2_{n_r} > \sigma^2_{n_i}$, noncircular noise)

What is the right basis for real world data?

Back to Dennis Gabor $\looparrowright$ but careful: Bedrossian and Nutall theorems

Consider

- Amplitude modulated signal $x(t) = m(t)\cos(\omega_0 t) \rightarrow m(t) \looparrowright$ envelope
- Phase modulated signal $x(t) = a\cos(\Phi(t)) \rightarrow \Phi(t) \looparrowright$ phase

Problem: there is an infinite number of pairs $[a(t), \Phi(t)]$ s.t. $m(t)\cos(\omega_0 t) = a\cos(\Phi(t))$

Solution: an analytic transform $z(t) = x(t) + \jmath\mathcal{H}\big(x(t)\big) = a(t)e^{\jmath\Phi(t)}$

Remark#1: $z(t)$ **cannot** be real, as $\mathcal{F}\big(z(t)\big) = 0 \text{ for } \omega < 0$

Remark#2: Hilbert transform (analytic signal) makes it possible to associate a **unique** pair $\big[a(t), \Phi(t)\big]$ to any real $x(t) = \Re\big\{a(t)e^{\jmath\Phi(t)}\big\}$

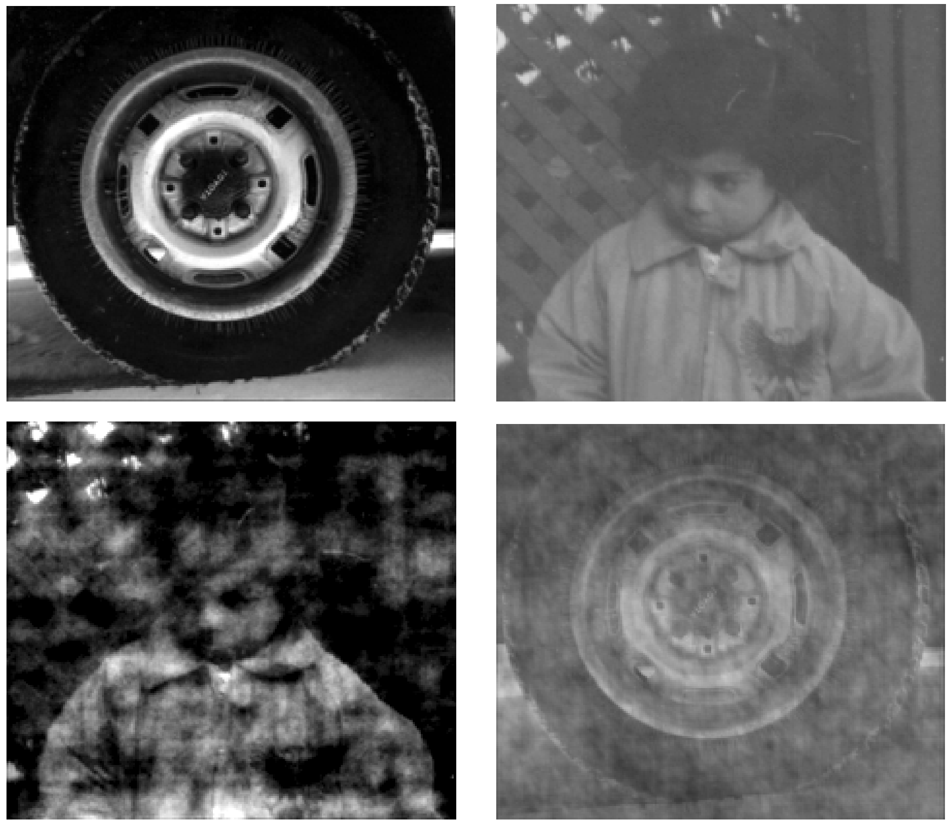
Remark#3: For $x(t) = a(t)\cos\Phi(t) \quad \Rightarrow \quad \mathcal{H}\big\{x(t)\big\} = a(t)\sin\Phi(t)$

Remark#4: From instantaneous phase $\Phi(t) \quad \rightarrow \quad$ **instantaneous frequency**

$$f(t) = d\Phi(t)/dt$$

so we have an excellent resolution and do not depend on stationarity

Human Visual System – Importance of Phase Information



Surrogate images. *Top:* Original images I_1 and I_2; *Bottom:* Images $\hat{I}_1$ and $\hat{I}_2$ generated by exchanging the amplitude and phase spectra of the original images.

Standard adaptive filtering algorithms in $\mathbb{C}$

Considered straightforward extensions of the corresponding algorithms in $\mathbb{R}$ – replace the vector transpose by the Hermitian transpose:

- Covariance

$$\mathcal{C} = E\{\mathbf{x}\mathbf{x}^T\} \qquad \rightsquigarrow \qquad \mathcal{C} = E\{\mathbf{z}\mathbf{z}^H\}$$

- Autoregressive model and Wiener solution

$$\mathbf{w} = \mathbf{R}^{-1}\mathbf{p} \qquad \rightsquigarrow \qquad \mathbf{w}^* = \mathbf{R}^{-1}\mathbf{p}$$

- Least mean square (LMS) $\rightsquigarrow$ complex LMS [Widrow *at al.* 1975]

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\mathbf{x}(k) \qquad \rightsquigarrow \qquad \mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\mathbf{z}^*(k)$$

- Real time recurrent learning (RTRL) [Williams & Zipser 1989] $\rightsquigarrow$ complex RTRL [Goh & Mandic 2004]

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\mathbf{\Pi}(k) \qquad \rightsquigarrow \qquad \mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\mathbf{\Pi}^*(k)$$

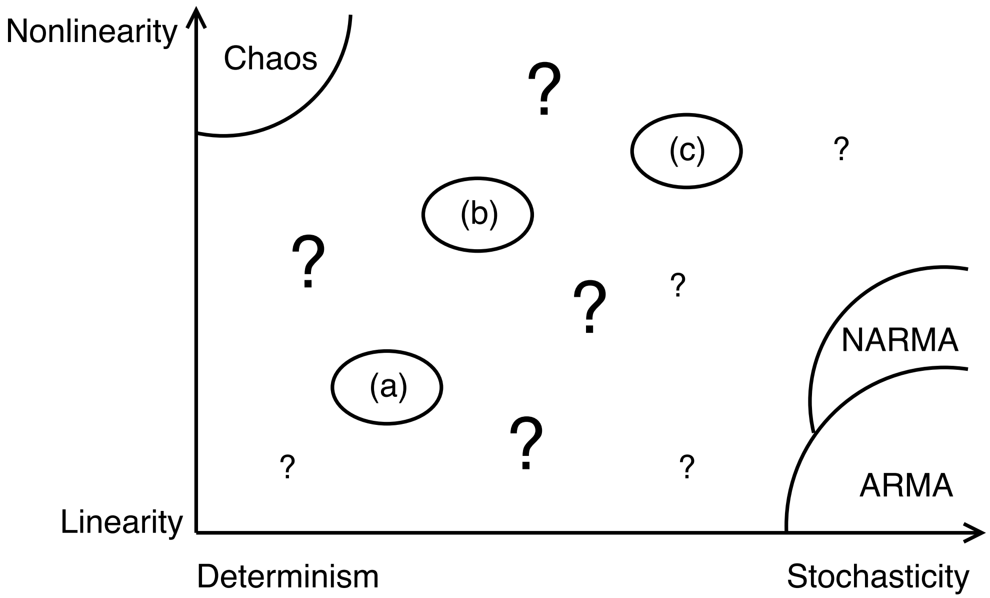
This is however valid only for circular complex random processes

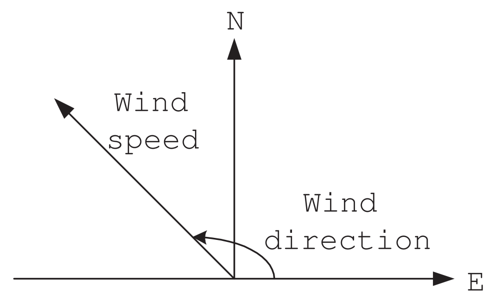
Property of division algebras $\looparrowright$ complex (non)circularity

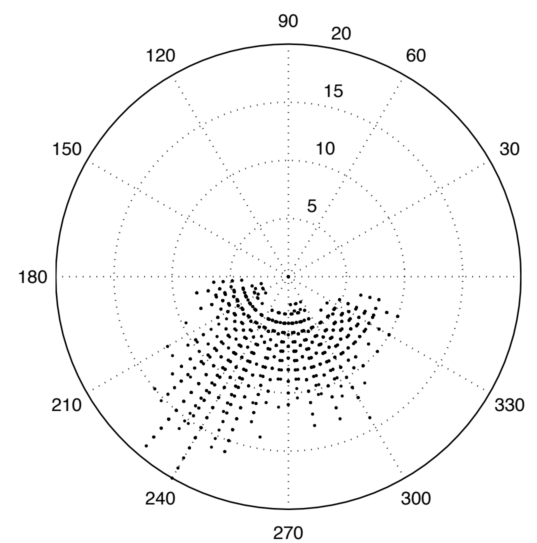
Noncircularity of the wind distribution $\quad v(k) = |v(k)|e^{\jmath\Phi(k)}$

Deterministic vs. Stochastic nature

Linear vs. Nonlinear nature







Change in signal modality can indicate e.g. health hazard (fMRI, HRV)

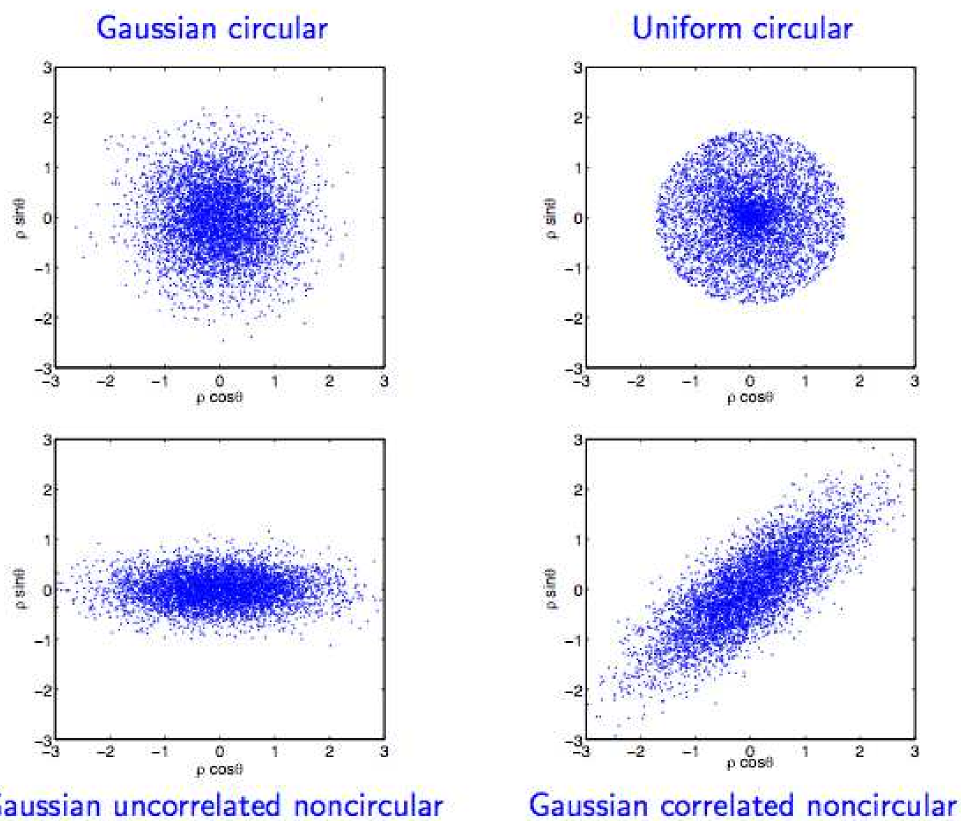
Real world signals are denoted by '???'

- ∘ $\exists$ a unique signature of complex signals?
- ∘ $\looparrowright$ **degree of noncircularity**

Circular vs Noncircular Complex Random Variables

Circularity = Rotation invariant distribution $p(\rho, \theta) = p(\rho, \theta - \phi)$

circular variable = construct $Z = \rho\cos(\theta) + j\rho\sin(\theta)$, $\theta \sim \mathcal{U}[0, 2\pi)$, $\rho \sim$ any pdf



- Geometric view of circularity (propriety) via real-imaginary scatter plots of zero-mean complex doubly-white Gaussian and uniform distributions
- The degrees of noncircularity of complex signals have a direct effect on the second order estimation performance of algorithms (LMS, RLS, Kalman filter, etc)

Bottom left: Circ.coef. = 0.8, corr.coef. =0 **Bottom right:** Circ.coef. = 0.8, corr.coef. =0.8

$\looparrowright$ **we can discriminate even between many Gaussian signals!**

Isomorphism between $\mathbb{C}$ and $\mathbb{R}^2$

(also serves as a basis for the $\mathbb{CR}$ calculus)

$$z \rightarrow z^a \quad \leftrightarrow \quad \begin{bmatrix} z \\ z^* \end{bmatrix} = \begin{bmatrix} 1 & \jmath \\ 1 & -\jmath \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

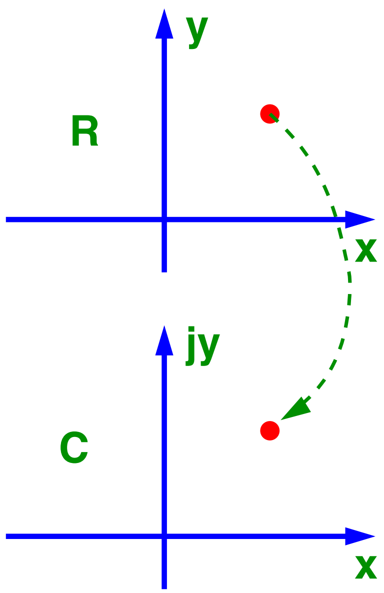
whereas in the case of complex–valued signals, we have

$$\mathbf{z} \rightarrow \mathbf{z}^a \quad \leftrightarrow \quad \begin{bmatrix} \mathbf{z} \\ \mathbf{z}^* \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \jmath\mathbf{I} \\ \mathbf{I} & -\jmath\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$$

For convenience, the "augmented" complex vector $\mathbf{v} \in \mathbb{C}^{2N\times 1}$ can be introduced as

$$\begin{aligned} \mathbf{v} &= [z_1, z_1^*, \ldots, z_N, z_N^*]^T \\ \mathbf{v} &= \mathbf{A}\mathbf{w}, \qquad \mathbf{w} = [x_1, y_1, \ldots, x_N, y_N]^T \end{aligned}$$

where matrix $\mathbf{A} = diag(\mathbf{J}, \ldots, \mathbf{J}) \in \mathbb{C}^{2N\times 2N}$ is block diagonal and transforms the **composite** real vector $\mathbf{w}$ into the augmented complex vector $\mathbf{v}$.



What are we doing wrong $\looparrowright$ the Widely Linear Model

Consider the MSE estimator of a signal y in terms of another observation x

$$\hat{y} = E[y|x]$$

For zero mean, jointly normal y and x, the solution is

$$\hat{y} = \mathbf{h}^T \mathbf{x}$$

In standard MSE in the complex domain $\hat{y} = \mathbf{h}^H \mathbf{x}$, however

$$\hat{y}_r = E[y_r|x_r, x_i] \quad \& \quad \hat{y}_i = E[y_i|x_r, x_i]$$

$$thus \qquad \hat{y} = E[y_r|x_r, x_i] + \jmath E[y_i|x_r, x_i]$$

Upon employing the identities $x_r = (x + x^*)/2$ and $x_i = (x - x^*)/2\jmath$

$$\hat{y} = E[y_r|x, x^*] + \jmath E[y_i|x, x^*]$$

and thus arrive at the **widely linear** estimator for general complex signals

$$y = \mathbf{h}^T \mathbf{x} + \mathbf{g}^T \mathbf{x}^*$$

We can now process general (noncircular) complex signals!

Augmented Complex Statistics

'properness' is a second order property and 'circularity' is a property of the pdf

widely linear estimator: $\hat{y}_{wl} = \mathbf{h}^T\mathbf{z} + \mathbf{g}^T\mathbf{z}^*$

For $\mathbf{z} = \mathbf{x} + \jmath\mathbf{y}$, using 'augmented' vectors $\mathbf{w}^a = [\mathbf{h}, \mathbf{g}]^T$ and $\mathbf{z}^a = [\mathbf{z}, \mathbf{z}^*]^T$

$$y_{wl} = \mathbf{w}^{aT}\mathbf{z}^a$$

'augmented' covariance matrix: $\mathcal{C}^a = E\begin{bmatrix}\mathbf{z}\\ \mathbf{z}^*\end{bmatrix}\begin{bmatrix}\mathbf{z}^H \mathbf{z}^T\end{bmatrix} = \begin{bmatrix}\mathcal{C} & \mathcal{P}\\ \mathcal{P}^* & \mathcal{C}^*\end{bmatrix}$

Remark#1: In general, the covariance matrix $\mathcal{C} = E\left[\mathbf{z}\mathbf{z}^H\right]$ **does not** completely describe the second order statistics of $\mathbf{z}$ $\;\looparrowright\;$ $c = \sigma_x^2 + \sigma_y^2$

Remark#2: General complex data are **improper**, requiring also the *pseudocovariance* $\mathcal{P} = E\left[\mathbf{z}\mathbf{z}^T\right]$ $\;\looparrowright\;$ $p = E[zz^T] = \sigma_x^2 - \sigma_y^2 + 2\jmath\rho_{xy}$

Widely linear AR solution

$$\begin{bmatrix}\mathbf{h}^*\\ \mathbf{g}^*\end{bmatrix} = \begin{bmatrix}\mathcal{C} & \mathcal{P}\\ \mathcal{P}^* & \mathcal{C}^*\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{c}\\ \mathbf{p}^*\end{bmatrix}$$

Augmented CLMS (ACLMS)

$$\begin{aligned}\mathbf{h}(\mathbf{k}+1) &= \mathbf{h}(\mathbf{k}) + \mu\mathbf{e}(\mathbf{k})\mathbf{z}^*(\mathbf{k})\\ \mathbf{g}(\mathbf{k}+1) &= \mathbf{g}(\mathbf{k}) + \mu\mathbf{e}(\mathbf{k})\mathbf{z}(\mathbf{k})\end{aligned}$$

Stability of this algorithm: Mandic and Xia, Douglas and Mandic 2008, 2009, 2010

Widely Linear Autoregressive Modelling in $\mathbb{C}$

Standard AR model of order n is given by

$$z(k) = a_1 z(k-1) + \cdots + a_n z(k-n) + q(k) = \mathbf{a}^T \mathbf{z}(k) + q(k),$$

Using the Yule-Walker equations the AR coefficients are found from

$$\mathbf{a}^* = \mathcal{C}^{-1}\mathbf{c}$$

$$\begin{bmatrix} a_1^* \\ a_2^* \\ \vdots \\ a_n^* \end{bmatrix} = \begin{bmatrix} c(0) & c^*(1) & \ldots & c^*(n-1) \\ c(1) & c(0) & \ldots & c^*(n-2) \\ \vdots & \vdots & \ddots & \vdots \\ c(n-1) & c(n-2) & \ldots & c(0) \end{bmatrix}^{-1} \begin{bmatrix} c(1) \\ c(2) \\ \vdots \\ c(n) \end{bmatrix}$$

where $\mathbf{c} = [c(1), c(2), \ldots, c(n)]^T$ is the time shifted correlation vector.

Widely linear model

$$y(k) = \mathbf{h}^T(k)\mathbf{x}(k) + \mathbf{g}^T(k)\mathbf{x}^*(k) + q(k)$$

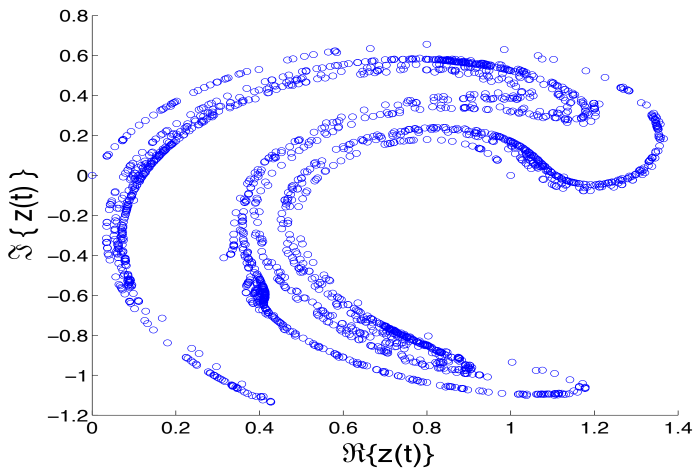
Widely linear normal equations

$$\begin{bmatrix} \mathbf{h}^* \\ \mathbf{g}^* \end{bmatrix} = \begin{bmatrix} \mathcal{C} & \mathcal{P} \\ \mathcal{P}^* & \mathcal{C}^* \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{c} \\ \mathbf{p}^* \end{bmatrix}$$

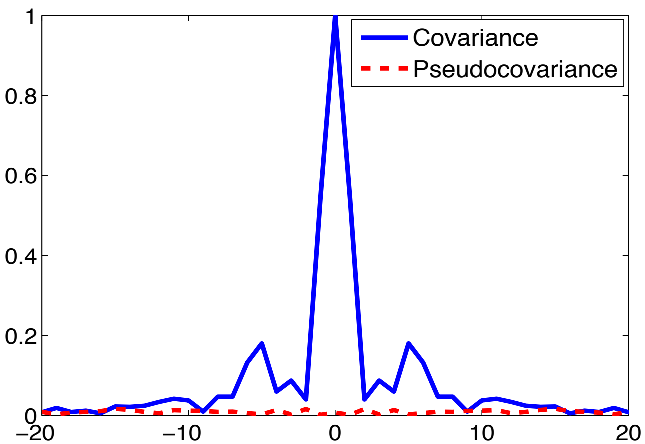
where $\mathbf{h}$ and $\mathbf{g}$ are coefficient vectors and $\mathbf{x}$ the regressor vector.

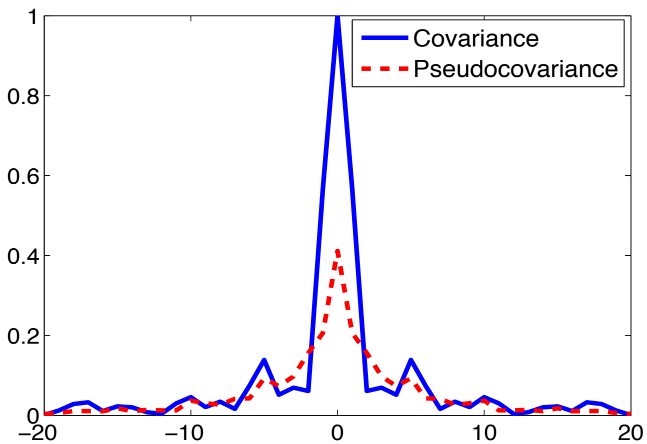
This is a rigorous way to model general complex signals!

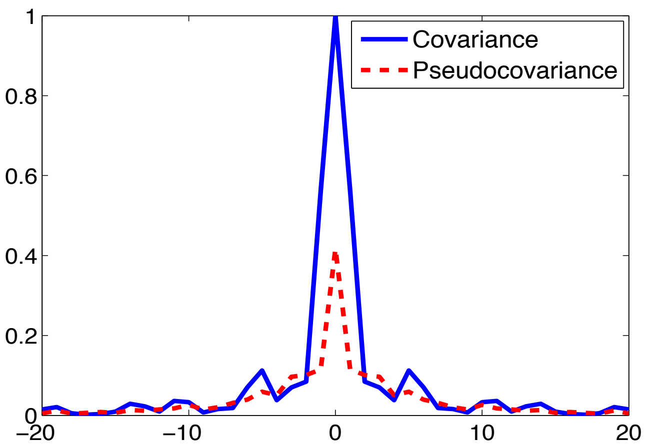
Circularity for Ikeda map



AR model of Ikeda signal

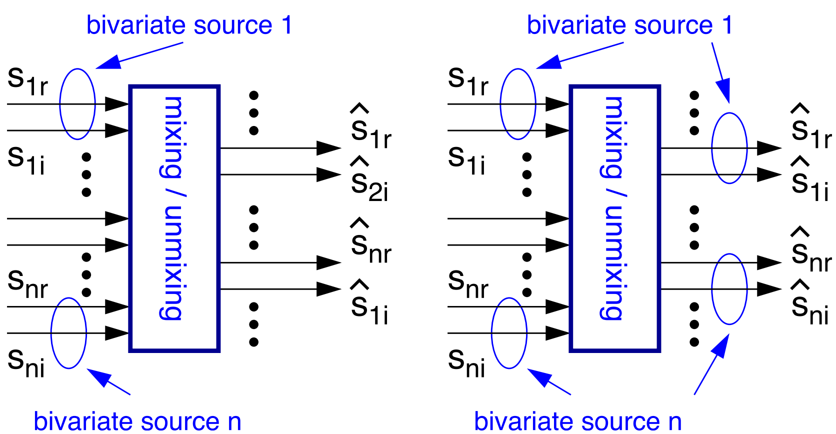


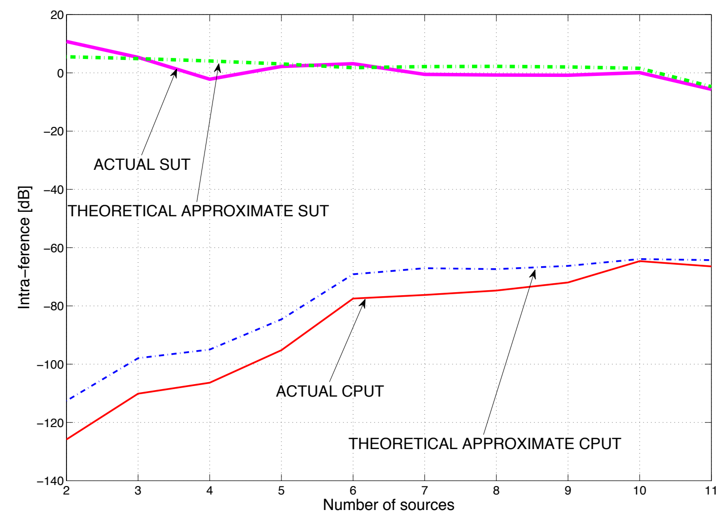




Covariances: Original Ikeda | Widely linear AR of Ikeda

Food for thought: Intra–ference *cf.* inter–ference
preserving the integrity of bivariate sources





- Current BSS (real and complex) decorrelate both the bivariate data channels from one another and the components of such channels
- For the **integrity** of bivarate sources to be preserved, we need to keep the correlation properties within data channels
- This is achieved through the pseudocovariance (phase) of complex data
- The Correlation Preserving Transform makes this possible (DPM 2014)

Some intuition: Cauchy–Riemann equations for $f(z) = u(x,y) + \jmath v(x,y)$

$$\frac{\partial u(x,y)}{\partial x} = \frac{\partial v(x,y)}{\partial y}, \qquad \frac{\partial v(x,y)}{\partial x} = -\frac{\partial u(x,y)}{\partial y}$$

The Jacobian matrix of a complex function $f(z) = u + \jmath v$, is given by

$$\mathbf{J} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix} \qquad \Leftrightarrow \qquad \begin{bmatrix} '1' & '1' \\ '-1' & '1' \end{bmatrix}$$

Thus, $f(z) = z^*$ is not analytic as its Jacobian $\mathbf{J} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$.

Functions which depend on both $z = x + \jmath y$ and $z^* = x - \jmath y$ are not analytic

$$J(z,z^*) = zz^* = x^2+y^2 \quad \Rightarrow \quad \mathbf{J} = \begin{bmatrix} 2x & 2y \\ 0 & 0 \end{bmatrix} \quad \Leftrightarrow \quad \frac{\partial u}{\partial x} \neq \frac{\partial v}{\partial y} \quad \frac{\partial v}{\partial x} \neq -\frac{\partial u}{\partial y}$$

One typical example is the cost function $J = \frac{1}{2}e(k)e^*(k) = \frac{1}{2}|e(k)|^2$.

The Key: $\mathbb{CR}$-derivatives

- If a function $f = f(z, z^*) = g(x, y) = u(x, y) + \jmath v(x, y)$ is holomorphic, then the Cauchy–Riemann conditions are satisfied, that is

$$\frac{\partial u(x,y)}{\partial x} = \frac{\partial v(x,y)}{\partial y} \text{ and } \frac{\partial v(x,y)}{\partial x} = -\frac{\partial u(x,y)}{\partial y}$$

$$\mathbb{R} - \text{derivative} \qquad \frac{1}{2}\left[\frac{\partial f}{\partial x} - \jmath\frac{\partial f}{\partial y}\right] = \frac{1}{2}\left[2\frac{\partial u}{\partial x} + 2\jmath\frac{\partial v}{\partial x}\right] = f'(z)$$

$$\mathbb{R}^* - \text{derivative} \qquad \frac{1}{2}\left[\frac{\partial f}{\partial x} + \jmath\frac{\partial f}{\partial y}\right] = 0$$

  $\Rightarrow$ **for holomorphic functions the $\mathbb{R}^*$-derivative vanishes and the $\mathbb{R}$-derivative is equivalent to the standard complex derivative $f'(z)$**

- $\rightsquigarrow$ if $f(z, z^*)$ is independent of z^*, then the $\mathbb{R}$-derivative of $f(z)$ is equivalent to the standard $\mathbb{C}$-derivative;

- Examples: $f(z) = z = x + \jmath y \quad \Rightarrow \quad \mathbb{R} - \mathrm{der} = 1 \quad \& \quad \mathbb{R}^* - \mathrm{der} = 0,$
  $f(z) = z^* = x - \jmath y \Rightarrow \mathbb{R} - \mathrm{der} = 0 \quad \& \quad \mathbb{R}^* - \mathrm{der} = 1$

Stochastic gradient optimisation: Complex gradient

For the filter error $e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k)$

$$J(e,e^*) = ee^* \Rightarrow \nabla_{\mathbf{w}} J = \frac{\partial J(e,e^*)}{\partial \mathbf{w}} = \left[\frac{\partial J(e,e^*)}{\partial w_1}, \ldots, \frac{\partial J(e,e^*)}{\partial w_N}\right]^T$$

For the minima

$$\frac{\partial J(e,e^*)}{\partial \mathbf{w}} = \mathbf{0} \qquad \text{and} \qquad \frac{\partial J(e,e^*)}{\partial \mathbf{w}^*} = \mathbf{0}$$

The first term of Taylor series expansion (since $J(e,e^*)$ is real).

$$\Delta J(e,e^*) = \left[\frac{\partial J}{\partial \mathbf{w}}\right]^T \Delta\mathbf{w} + \left[\frac{\partial J}{\partial \mathbf{w}^*}\right]^T \Delta\mathbf{w}^* = 2\Re\left\{\left[\frac{\partial J}{\partial \mathbf{w}}\right]^H \Delta\mathbf{w}^*\right\} = 2\Re\left\{\left[\frac{\partial J}{\partial \mathbf{w}^*}\right]^T \Delta\mathbf{w}^*\right\}$$

The scalar product

$$< \partial J/\partial \mathbf{w}, \Delta\mathbf{w}^* >= \left[\frac{\partial J}{\partial \mathbf{w}}\right]^H \Delta\mathbf{w}^* = \| \partial J/\partial \mathbf{w} \| \| \Delta\mathbf{w}^* \| \cos\angle(\partial J/\partial \mathbf{w}, \Delta\mathbf{w}^*)$$

achieves its maximum value when $\frac{\partial J}{\partial \mathbf{w}} \parallel \Delta\mathbf{w}^*$.

Thus, **the maximum change of the gradient of the cost function is in the direction of the conjugate weight vector**, and

$$\nabla_{\mathbf{w}} J = \nabla_{\mathbf{w}^*} J \qquad \text{Brandwood 1984}$$

The derivative of a cost function $\frac{1}{2}e(k)e^*(k)$ and CLMS

As $\mathbb{C}$-derivatives are not defined for real functions of complex variable

$$\mathbb{R}-\text{der:}\quad \frac{\partial}{\partial \mathbf{z}} = \frac{1}{2}\left[\frac{\partial}{\partial \mathbf{x}} - \jmath\frac{\partial}{\partial \mathbf{y}}\right] \qquad \mathbb{R}^*-\text{der:}\quad \frac{\partial}{\partial \mathbf{z}^*} = \frac{1}{2}\left[\frac{\partial}{\partial \mathbf{x}} + \jmath\frac{\partial}{\partial \mathbf{y}}\right]$$

and the gradient

$$\nabla_{\mathbf{w}} J = \frac{\partial J(e,e^*)}{\partial \mathbf{w}} = \left[\frac{\partial J(e,e^*)}{\partial w_1},\ldots,\frac{\partial J(e,e^*)}{\partial w_N}\right]^T = 2\frac{\partial J}{\partial \mathbf{w}^*} = \underbrace{\frac{\partial J}{\partial \mathbf{w}^r} + \jmath\frac{\partial J}{\partial \mathbf{w}^i}}_{pseudogradient}$$

The standard Complex Least Mean Square (CLMS) (Widrow *et al.* 1975)

$$\begin{aligned}
y(k) &= \mathbf{x}^T(k)\mathbf{w}(k) \\
e(k) = d(k) - y(k) & \qquad e^*(k) = d^*(k) - \mathbf{x}^*(k)\mathbf{w}^*(k) \\
\text{and} &\quad \nabla_{\mathbf{w}} J = \nabla_{\mathbf{w}^*} J \\
\mathbf{w}(k+1) &= \mathbf{w}(k) - \mu\frac{\partial \frac{1}{2}e(k)e^*(k)}{\partial \mathbf{w}^*(k)} = \mathbf{w}(k) + \mu e(k)\mathbf{x}^*(k)
\end{aligned}$$

☞ **no need for tedious derivations $\hookrightarrow$ the CLMS is derived in one line**

The Augmented (widely linear) CLMS (ACLMS)

Widely linear model $\quad y(k) = \mathbf{h}^{\mathsf{T}}(k)\mathbf{z}(k) + \mathbf{g}^{\mathsf{T}}(k)\mathbf{z}^*(k)$

$$\mathbf{h}(k+1) = \mathbf{h}(k) - \mu\nabla_{\mathbf{h}^*}J \qquad \Rightarrow \qquad \nabla_{\mathbf{h}^*}J = -e(k)\mathbf{x}^*(k)$$

$$\mathbf{g}(k+1) = \mathbf{g}(k) - \mu\nabla_{\mathbf{g}^*}J \qquad \Rightarrow \qquad \nabla_{\mathbf{g}^*}J = -e(k)\mathbf{x}(k)$$

Therefore, the ACLMS update

$$\begin{aligned}
\mathbf{h}(\mathbf{k+1}) &= \mathbf{h}(\mathbf{k}) + \mu\mathbf{e}(\mathbf{k})\mathbf{x}^*(\mathbf{k}) \\
\mathbf{g}(\mathbf{k+1}) &= \mathbf{g}(\mathbf{k}) + \mu\mathbf{e}(\mathbf{k})\mathbf{x}(\mathbf{k})
\end{aligned}$$

or in a more compact form (using augmented input and weight vectors)

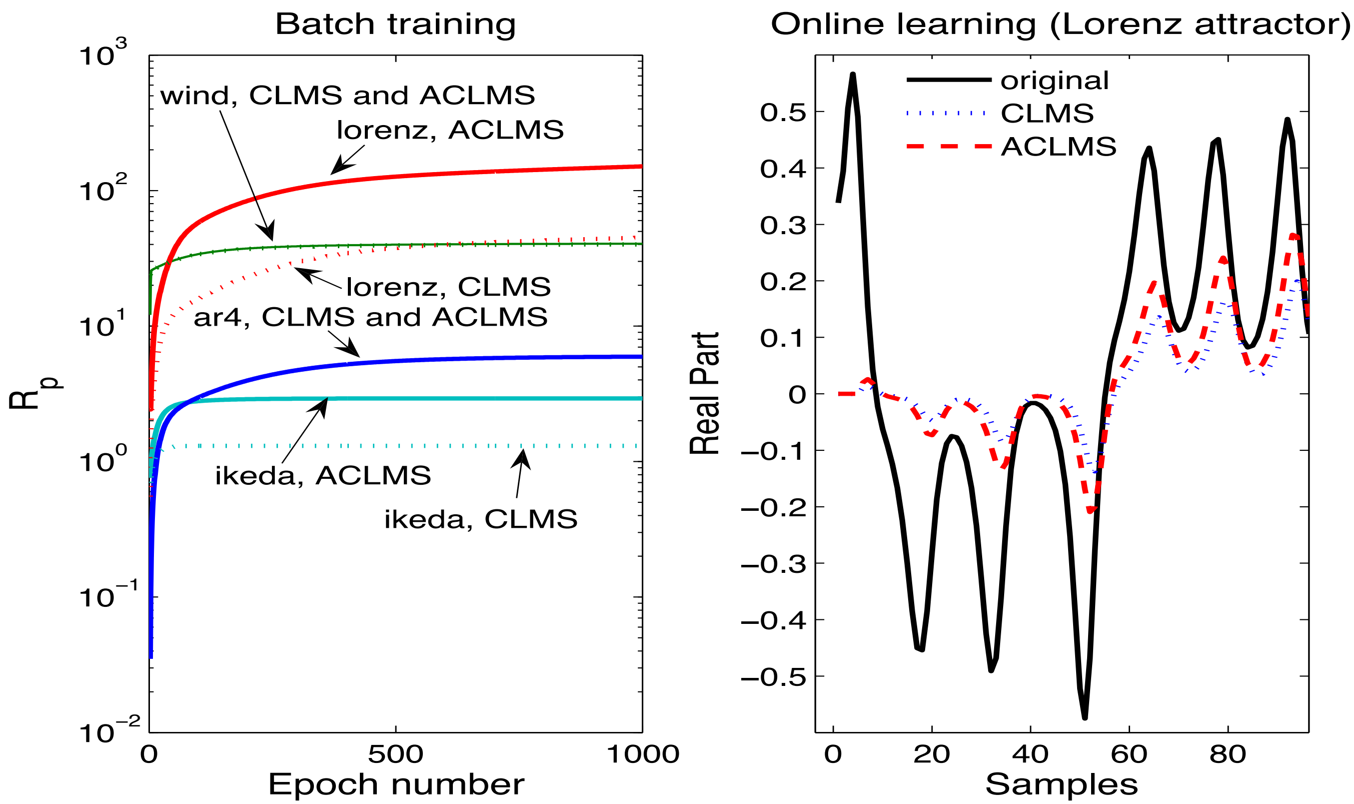
$$\mathbf{w^a}(\mathbf{k+1}) = \mathbf{w^a}(\mathbf{k}) + \eta\mathbf{e^a}(\mathbf{k})\mathbf{x^{a^*}}(\mathbf{k})$$

where $\eta = \mu_h = \mu_g$, $\mathbf{w^a}(\mathbf{k}) = [\mathbf{h^T}(\mathbf{k}),\ \mathbf{g^T}(\mathbf{k})]^{\mathbf{T}}$, $\mathbf{x^a}(\mathbf{k}) = [\mathbf{x^T}(\mathbf{k}),\ \mathbf{x^H}(\mathbf{k})]^{\mathbf{T}}$, $e^a(k) = d(k) - \mathbf{x^{a^T}}(\mathbf{k})\mathbf{w^a}(\mathbf{k})$

[For more detail see: Mandic *et al.* 2008, 2009, 2010].

Performance of the ACLMS

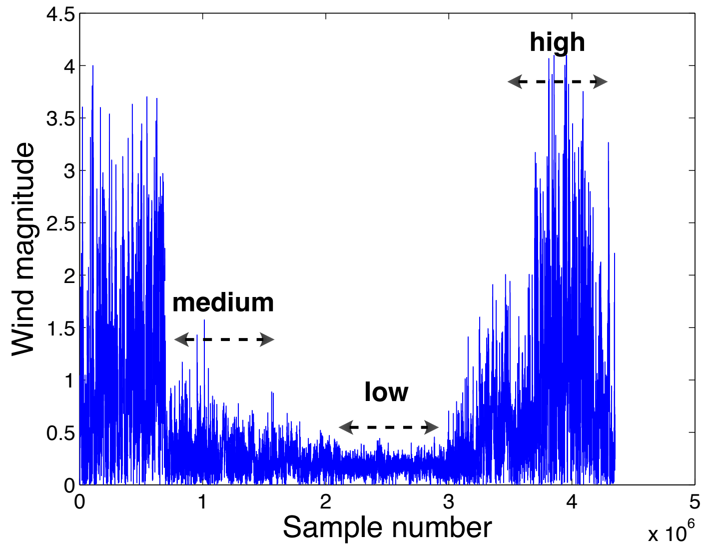
Evaluated for both second order circular (proper) and improper signals.



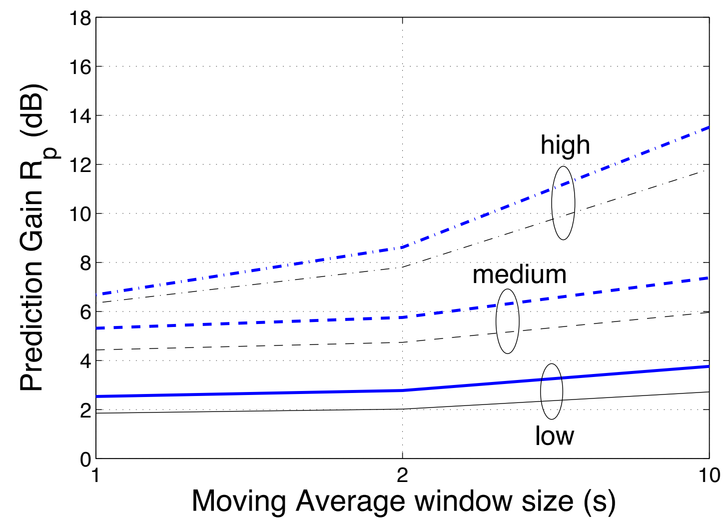
The ACLMS outperforms CLMS for second order noncircular signals.

Wind modelling: Performance vs dynamics vs circularity

Data recorded in an urban environment over one day



(e) Modulus of complex windover one day



(f) CLMS vs ACLMS for different wind regimes. CLMS - black, ACLMS - blue

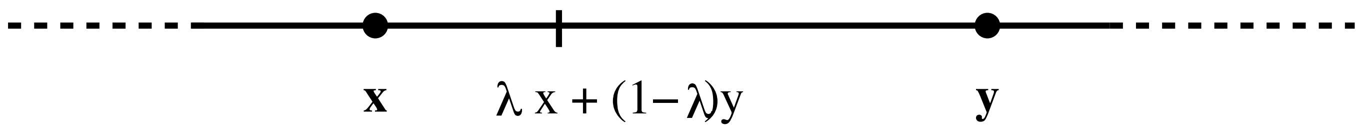
Different wind regimes $\rightsquigarrow$ different dynamics,

$$v(k) = |v(k)|e^{\jmath\Phi(k)}, \; |v| \text{ - speed, } \Phi \text{ - direction}$$

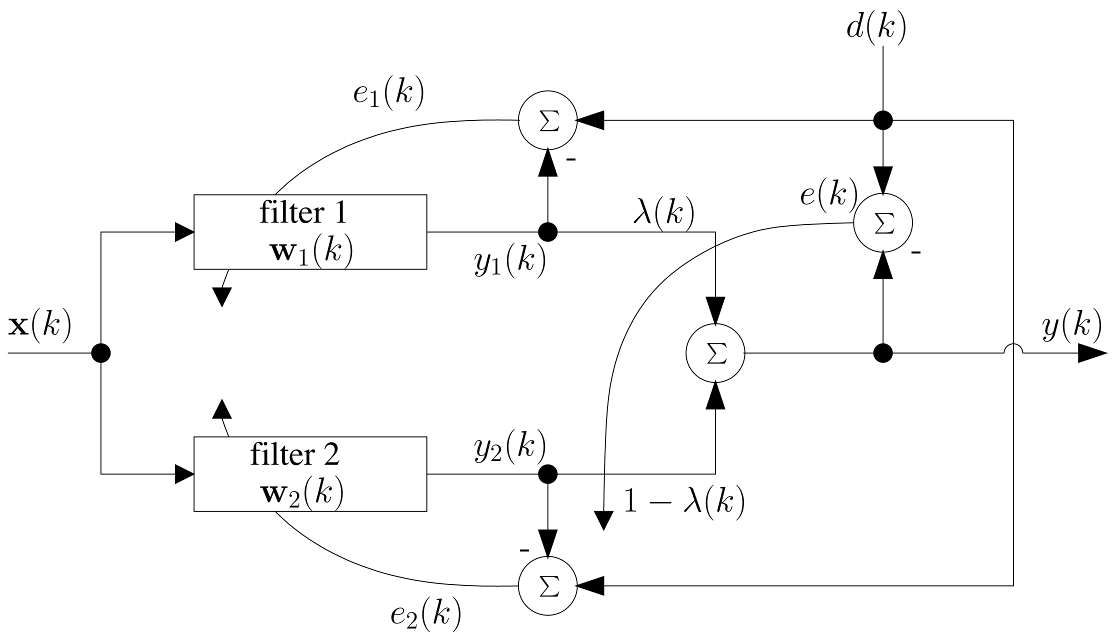
Different dynamics $\rightsquigarrow$ different circularity properties $\rightsquigarrow$ impact of ACLMS

Circular vs. Noncircular: That is the question
Best of both worlds ↬ a Hybrid Filter

Virtues of Convex Combination ($\lambda \in [0,1]$)



Convexity ⇒ existence and uniqueness of solution ☺



Let Filter1 be trained by CLMS and Filter2 by ACLMS

Adaptation of the mixing parameter λ

To preserve their inherent characteristics, subfilters, $Filter1$ and $Filter2$ updated based on their own errors:

- Linear $e_{clms}(k)$

- Widely linear $e_{aclms}(k)$

The convex mixing parameter λ is updated based on based on

$$J(k) = \frac{1}{2}e(k)e^*(k) = \frac{1}{2}|e(k)|^2 \quad \rightsquigarrow \quad \nabla_\lambda J(k)_{\lambda=\lambda(k)} = e(k)\frac{\partial e^*(k)}{\partial \lambda(k)} + e^*(k)\frac{\partial e(k)}{\partial \lambda(k)}$$

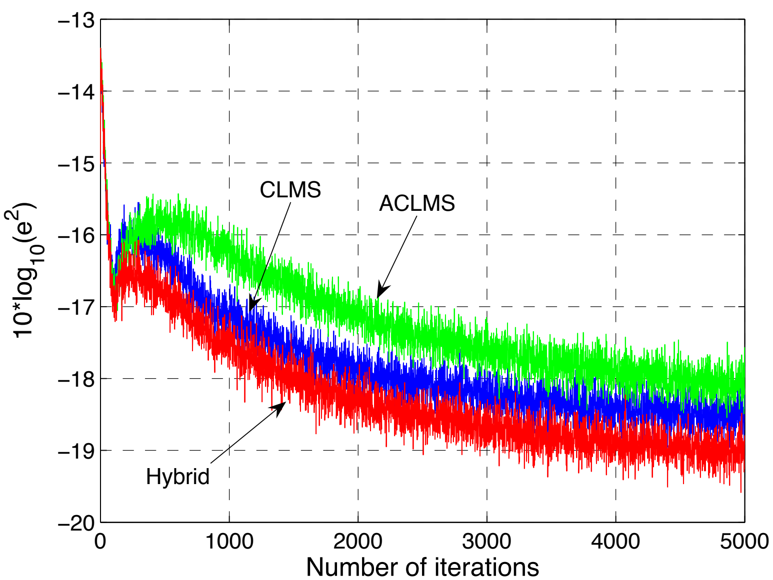
and the stochastic gradient based update of λ becomes

$$\begin{aligned} \lambda(k+1) &= \lambda(k) + \mu_\lambda \Big[e(k)\big(y_{aclms}(k) - y_{clms}(k)\big)^* \\ &+ e^*(k)\big(y_{clms}(k) - y_{aclms}(k)\big)\Big] \end{aligned}$$

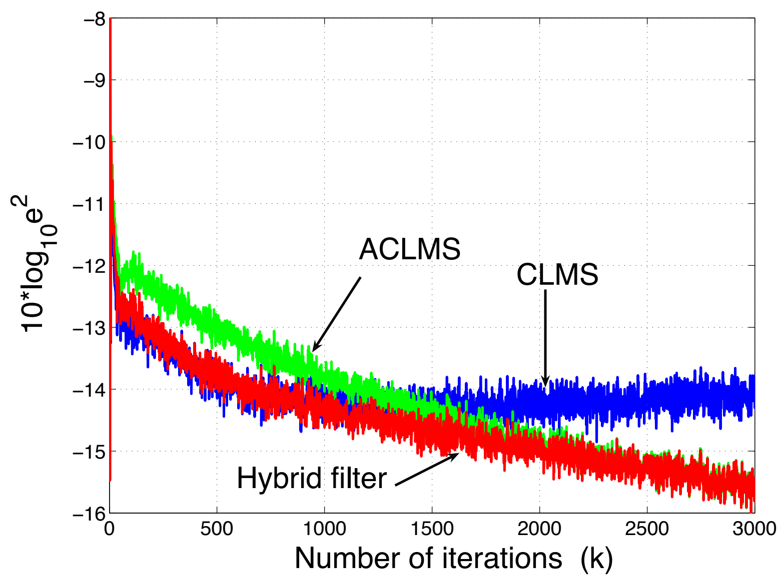
We must ensure that the value of $\lambda(k)$ belongs to $0 \le \lambda(k) \le 1$.

The hybrid CLMS $\leftrightarrow$ ACLMS filter (prediction setting)

Left: circular AR(4) process



Right: Noncircular Ikeda map



- ◦ The CLMS has half the number of parameters of ALMS
- ◦ Hence, it initially converges faster for all test signals

⊛ Both filters perform similarly for proper data in terms of the steady state

⊛ ACLMS has superior steady state properties for the improper Ikeda map

⊛ Hybrid filter: both fast covergence and excellent steady state properties!

Duality between bivariate real and complex filters

The bivariate (dual channel) real filter:

$$\begin{aligned}\hat{x}(k) &= \mathbf{a}^T(k)\mathbf{x}(k) + \mathbf{b}^T(k)\mathbf{y}(k)\\ \hat{y}(k) &= \mathbf{c}^T(k)\mathbf{x}(k) + \mathbf{d}^T(k)\mathbf{y}(k)\end{aligned}$$

Augmented CLMS:

$$\begin{aligned}\hat{x}(k) &= (\mathbf{h}_r(k) + \mathbf{g}_r(k))^T(k)\mathbf{x}(k) + (\mathbf{g}_i(k) - \mathbf{h}_i(k))^T(k)\mathbf{y}(k)\\ \hat{y}(k) &= (\mathbf{h}_i(k) + \mathbf{g}_i(k))^T(k)\mathbf{x}(k) + (\mathbf{h}_r(k) - \mathbf{g}_r(k))^T(k)\mathbf{y}(k)\end{aligned}$$

The bound on the stepsize which preserves convergence: $0 < \mu < \frac{2}{\lambda_{max}}$

The bivariate and augmented complex correlation matrices related as

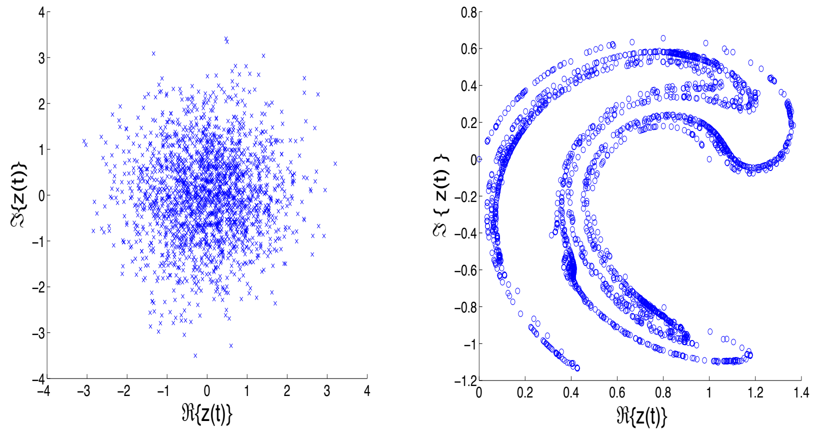
$$\mathcal{C}^a = E[\mathbf{z}^a\mathbf{z}^{aH}] = E\big[\mathbf{A}\boldsymbol{\omega}\boldsymbol{\omega}^T\mathbf{A}^H\big] = \mathbf{AWA}^H$$

It then follows that $\lambda^a = 2\lambda^\omega$.

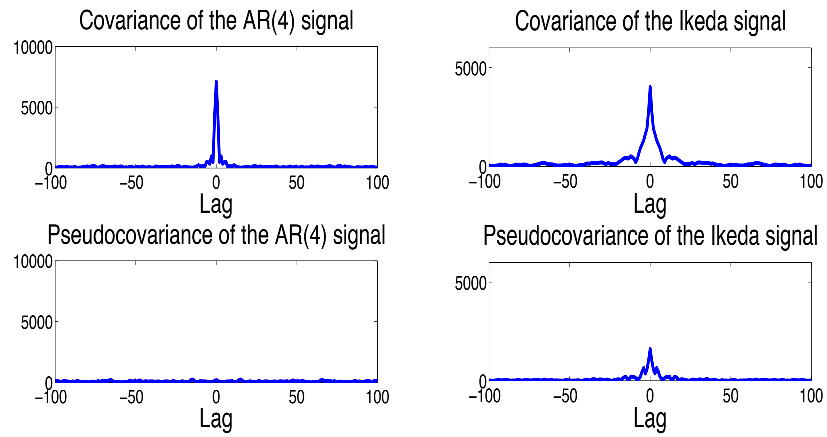
$\Rightarrow$ ACLMS and DCRLMS converge at the same speed when $\mu_r = 2\mu_{aclms}$.

More detail can be found in the follow up papers with S. Douglas and S. Javidi

Duality: Simulations



AR(4) and Ikeda signal



Covariances and pseudocovariances

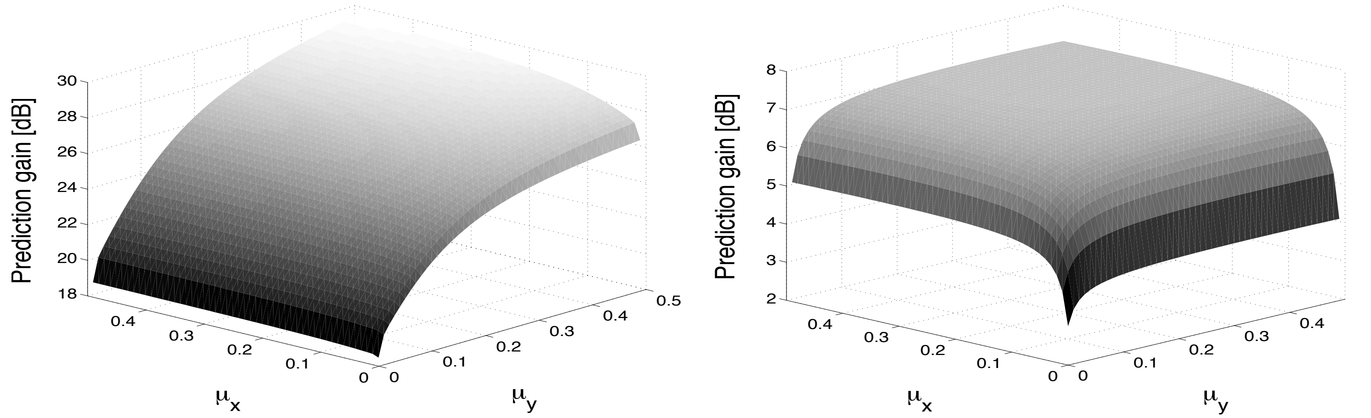
$$R_p = 10 \log \frac{\sigma_z^2}{\sigma_e^2}$$

Algorithm	AR4	Ikeda	Wind
R_p for DCRLMS	5.8423	3.9733	13.2604
R_p for CLMS	6.6380	2.4278	14.2941
R_p for ACLMS	6.6096	4.0330	14.8926
R_p for DCRLMS (double μ)	6.6096	4.0330	14.8926

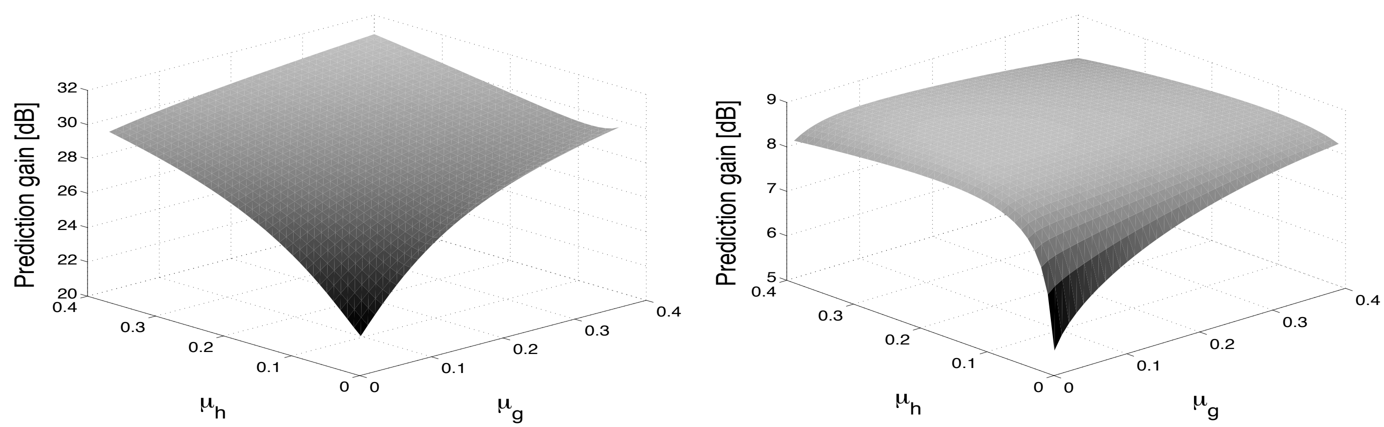
Duality simulations: Dependence on the parameters

(observe the possibility for better tuning using the complex ACLMS)

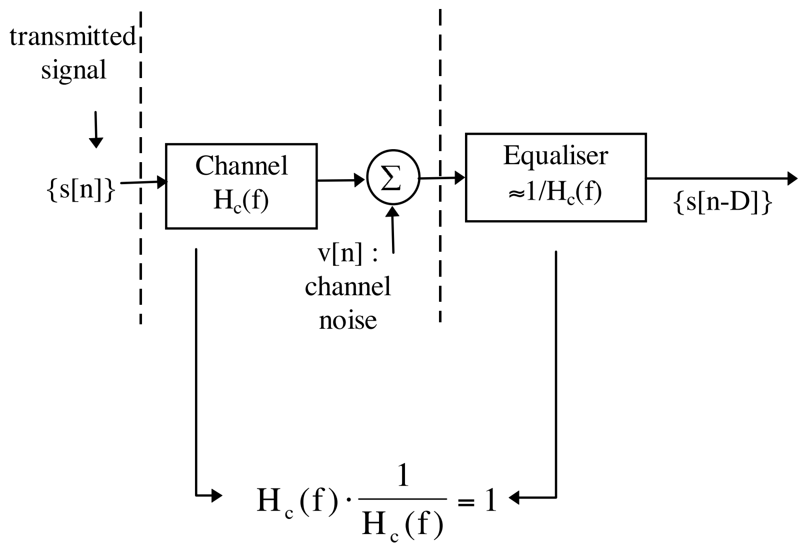
DCRLMS Lorenz (left) and wind signal (right)



ACLMS Lorenz (left) and wind signal (right)



Example: Channel equalisation in digital communications



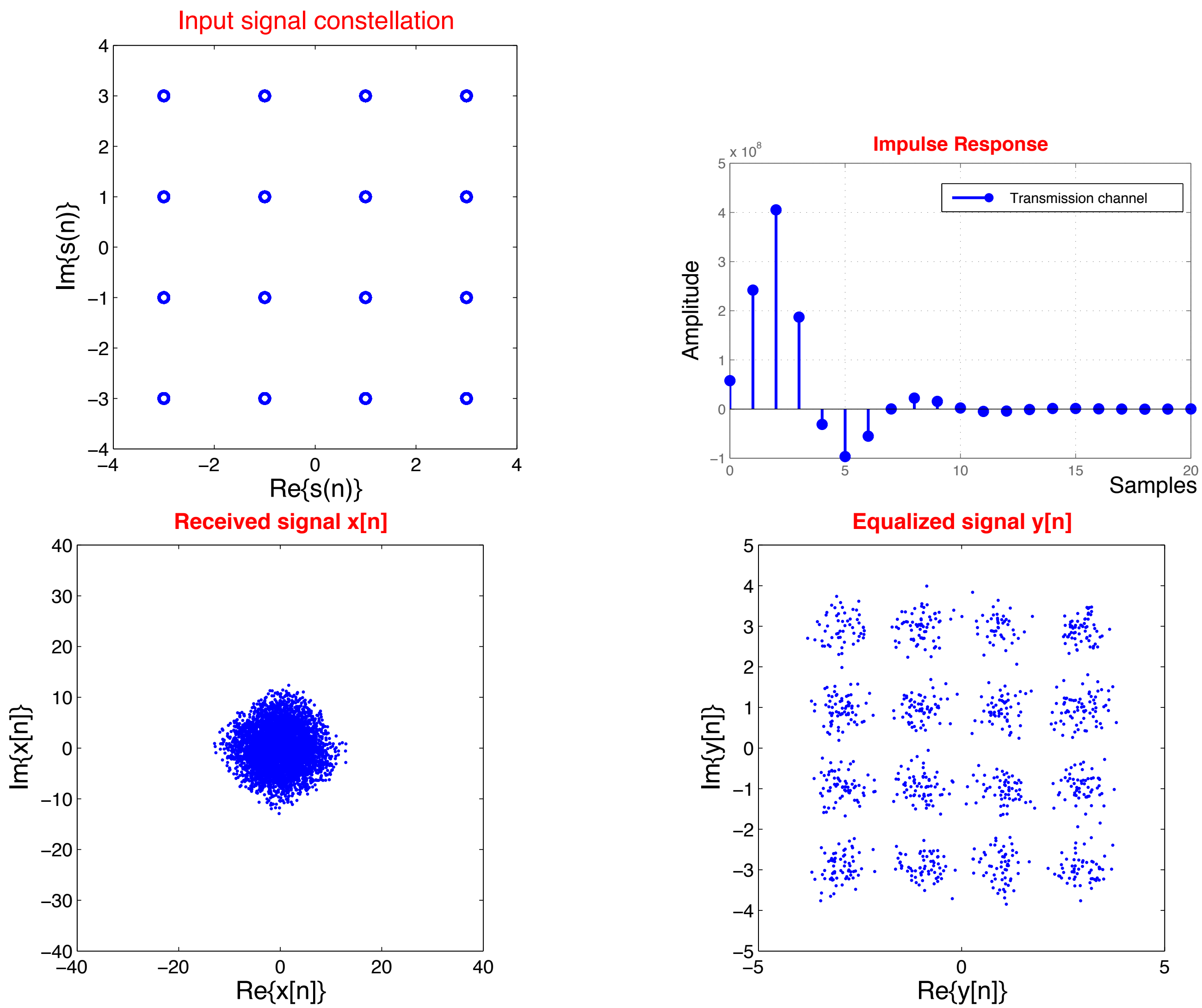
□ Channel equalization is a simple way of mitigating the detrimental effects caused by a frequency-selective and/or dispersive communication link between sender and receiver.

□ During the training phase of channel equalization, a digital signal $s[n]$ that is known to both the transmitter and receiver is sent by the transmitter to the receiver

□ The received signal $x[n]$ contains two signals: the signal $s[n]$ filtered by the channel impulse response, and an unknown broadband noise signal $v[n]$

□ The goal is to filter $x[n]$ to remove the inter-symbol interference (ISI) caused by the dispersive channel and to minimize the effect of the additive noise $v[n]$

Example: Digital communications $\looparrowright$ continued



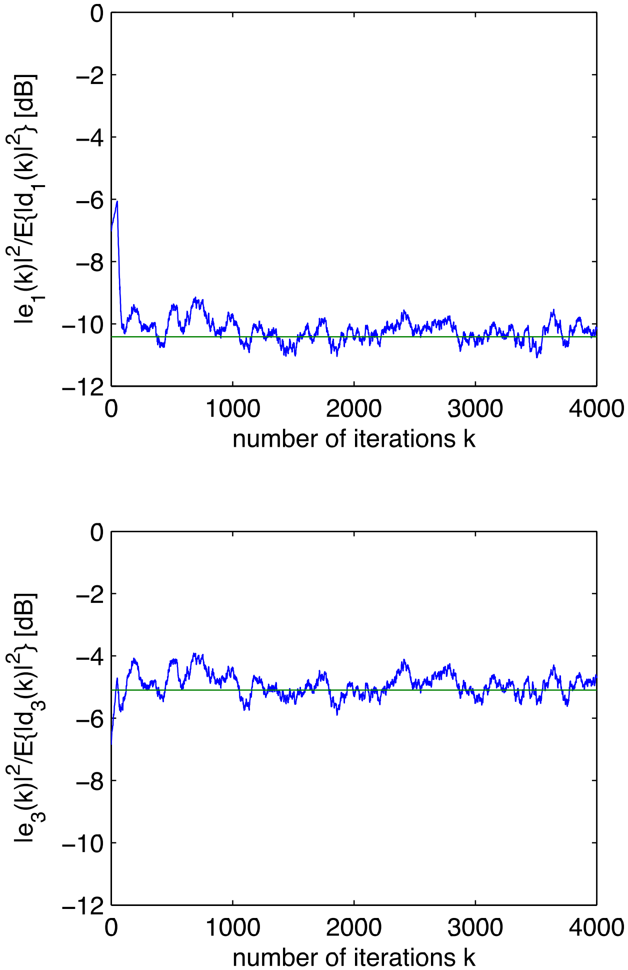
Beamforming example

- Beamforming Model: ULA, $\lambda/2$ spacing
- Sources: BPSK, BPSK, QPSK, QPSK
- Angle of Arrival: $-45°, 8°, -13°, 30°$
- Number of Antenna Elements: 3
- Algorthims: ACLMS and CCLMS
- Desired Signal: $d(k) = s_p^*(k)$, $1 \leq p \leq 4$
- Step Size: $\mu = 0.0001$.

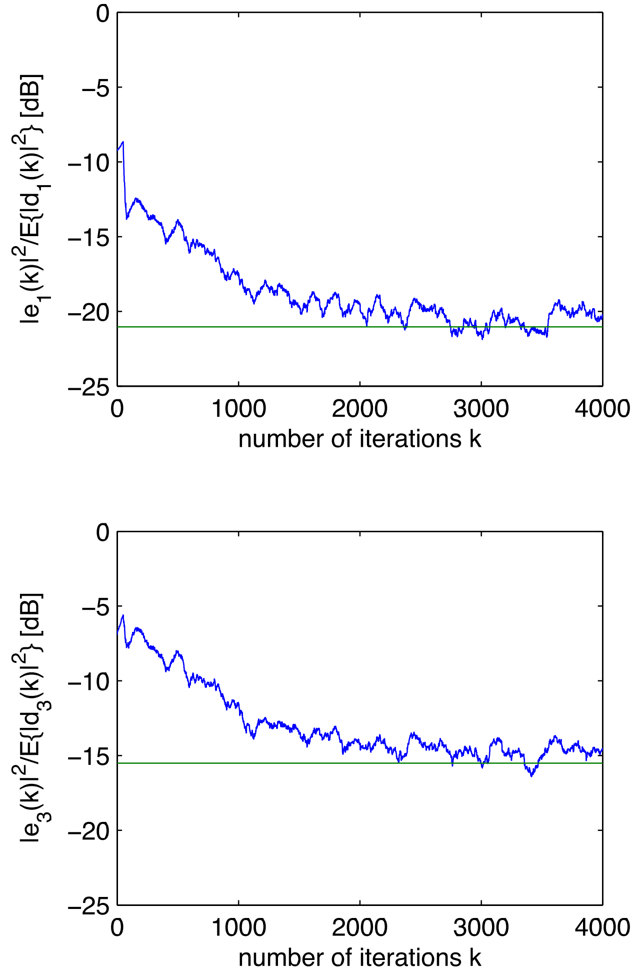
Compare: Convergence Rates, Steady State MSE

Convergence of the CLMS and ACLMS algorithms

CCLMS



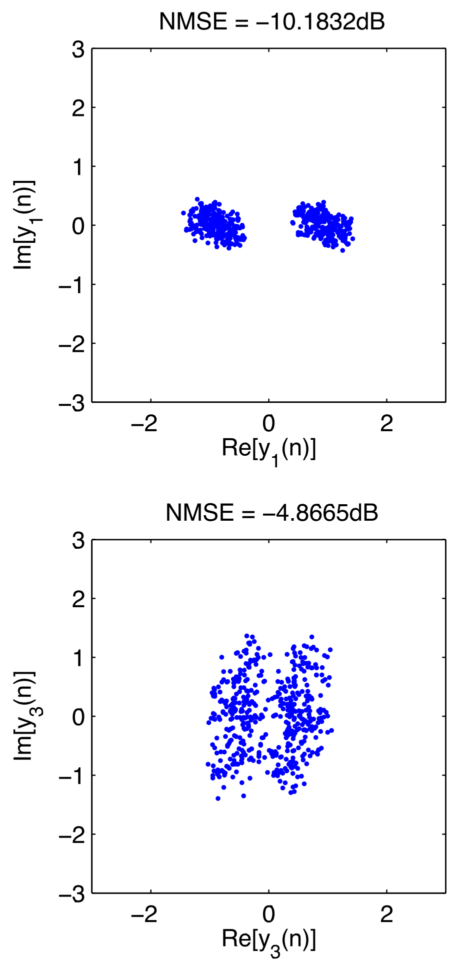
ACLMS



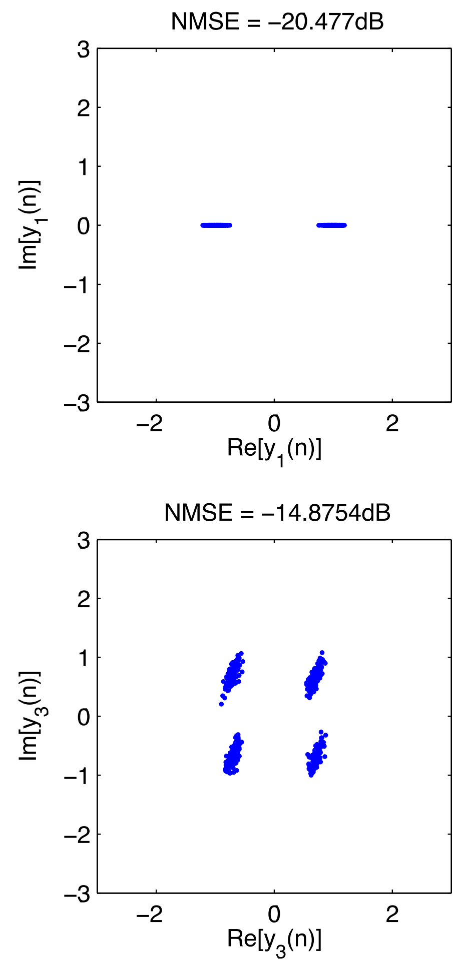
ACLMS takes longer to converge due to input signal non-circularity.

Output signal constellations

CCLMS

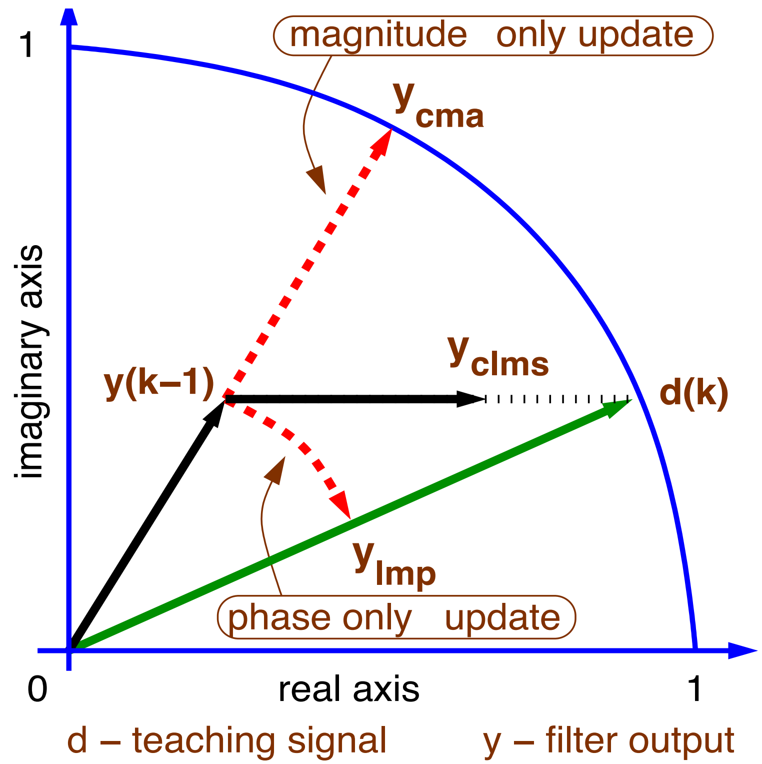


ACLMS



ACLMS has lower MSE because of non-circular binary sources

A continuum between phase-only and magnitude-only cost functions (when does phase error matter?)



So, when does the (complex) phase error matter to the overall performance?

- **Answer #1:** Constant Modulus Channel Estimator (CMCE) [Rupp 1998] can estimate time-varying channels better *if phase error is ignored*.

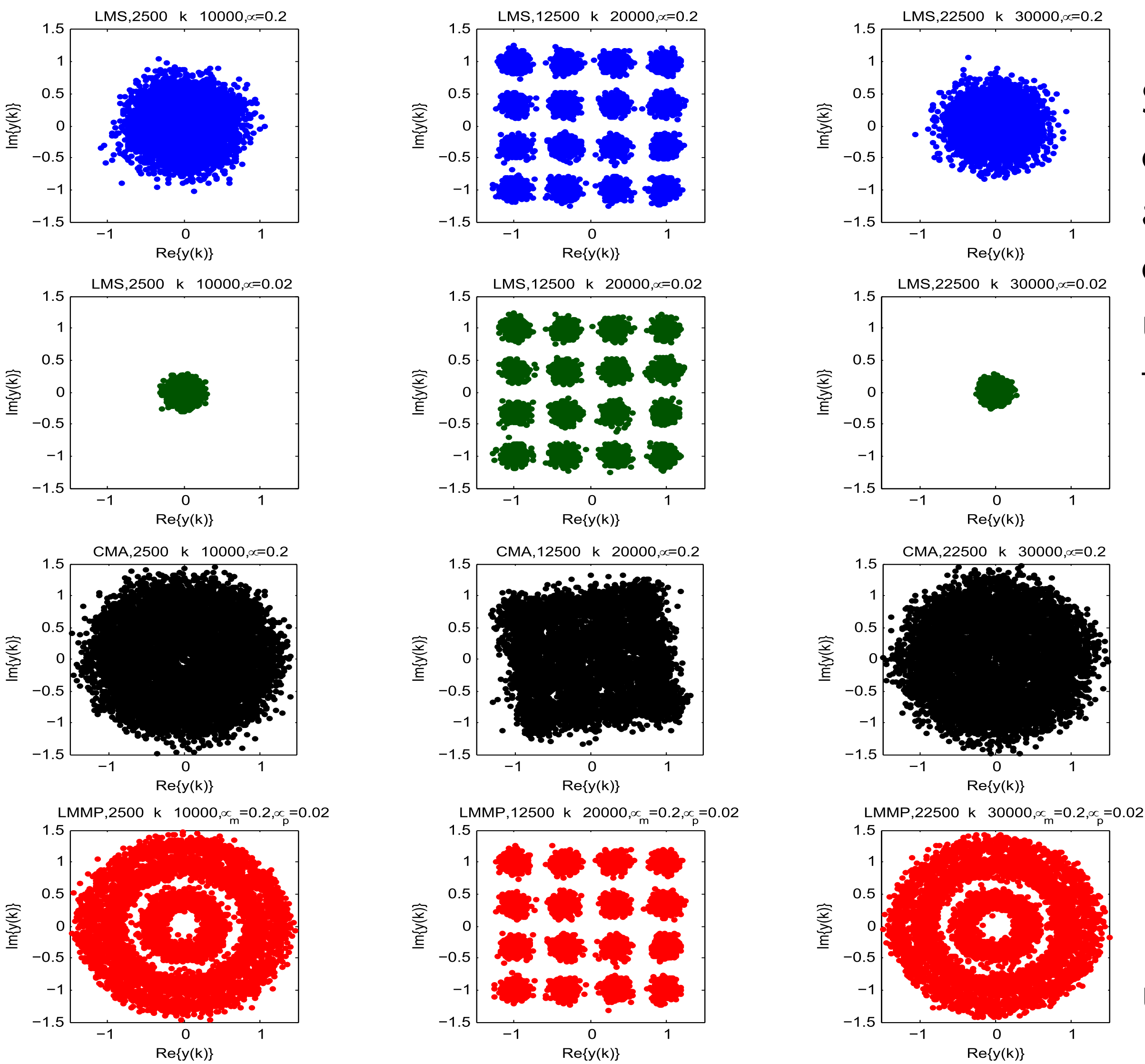
- **Answer #2:** Least Mean Phase (LMP) [Tarighat/Sayed 2004] can estimate complex symbols better if *phase error term is added to update*.

- **Answer #3:** Least Mean Magnitude Phase (LMMP) [Douglas/Mandic ICASSP 2011] employs a combined magnitude- and phase-based criterion *best performance with stable behavior*.

Signal Constellations: Channel Equalization

Channel: 3-tap with frequency offset   Source phases: time-varying

Performance meas.: average inter-symbol interfer., ISI (not dependent on phase)

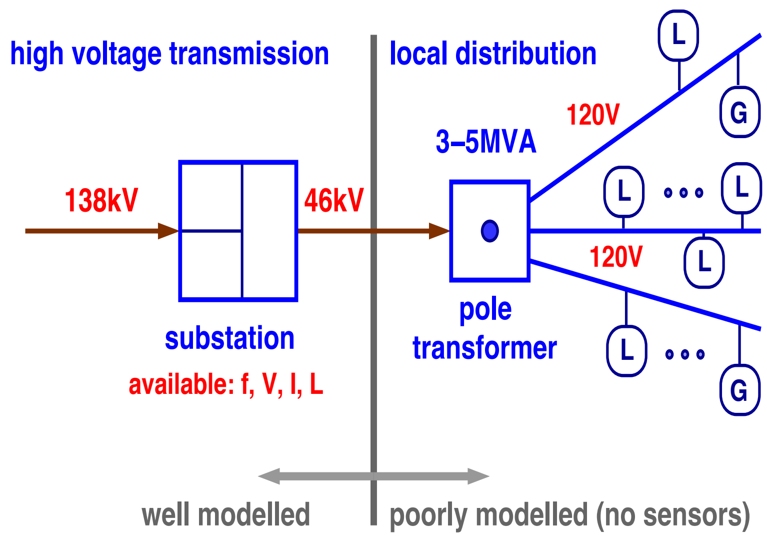


Moral:
Signal phase uncertainty can harm channel amplitude and phase estimation performance, unless it is mitigated within the algorithm itself.

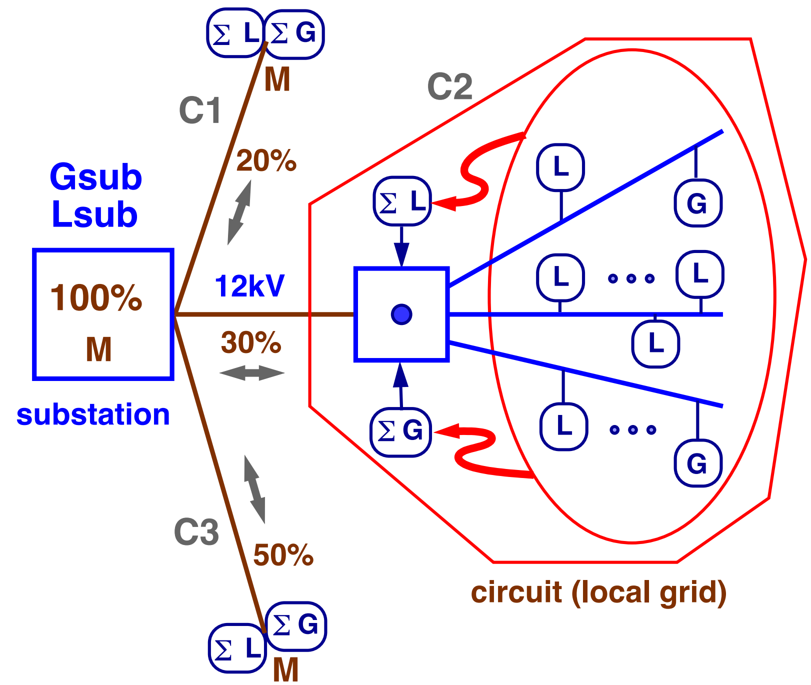
LMMP in red addresses this uncertainty explicitly.

Case Study: Frequency estimation in smart grid
sources of frequency deviation

Transmission: well-modelled
distribution side is not



Block diagram of power grid



Nodal estimation

- Dual character of load/supply $\rightsquigarrow$ $f+$ for $G > L$ and $f-$ for $G < L$
- Harmonics and freq. drifts from loads with nonlinear $V - I$ properties
- Transient stability issues cause inaccurate frequency estimates, also switching on/off the shunt capacitors in reactive power compensation

The Three-Phase Power System and $\alpha\beta$ Transformation

Three-phase system where $V_a(k), V_b(k), V_c(k)$ are the peak values.

$$v_{\mathsf{a}}(k) = V_a(k)\cos(\omega k \triangle \mathsf{T} + \phi)$$

$$v_{\mathsf{b}}(k) = V_b(k)\cos(\omega k \triangle \mathsf{T} + \phi - \frac{2\pi}{3})$$

$$v_{\mathsf{c}}(k) = V_c(k)\cos(\omega k \triangle \mathsf{T} + \phi + \frac{2\pi}{3})$$

The $\alpha\beta$ transform - a complex signal which carries the same information

$$\begin{bmatrix} v_0 \\ v_\alpha(k) \\ v_\beta(k) \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} v_{\mathsf{a}}(k) \\ v_{\mathsf{b}}(k) \\ v_{\mathsf{c}}(k) \end{bmatrix}$$

For **balanced systems** $v_0 = 0$, and thus the complex Clarke's voltage

$$v(k) = v_\alpha(k) + \jmath v_\beta(k) = A(k)e^{\jmath(\omega k \Delta T + \phi)} + B(k)e^{-\jmath(\omega k \Delta T + \phi)}$$

$$A(k) = \frac{\sqrt{6}(V_a(k)+V_b(k)+V_c(k))}{6} \quad \text{and} \quad B(k) = \frac{\sqrt{6}(2V_a(k)-V_b(k)-V_c(k))}{12} - \frac{\sqrt{2}(V_b(k)-V_c(k))}{4}\jmath$$

Clearly, unbalanced systems are noncircular!

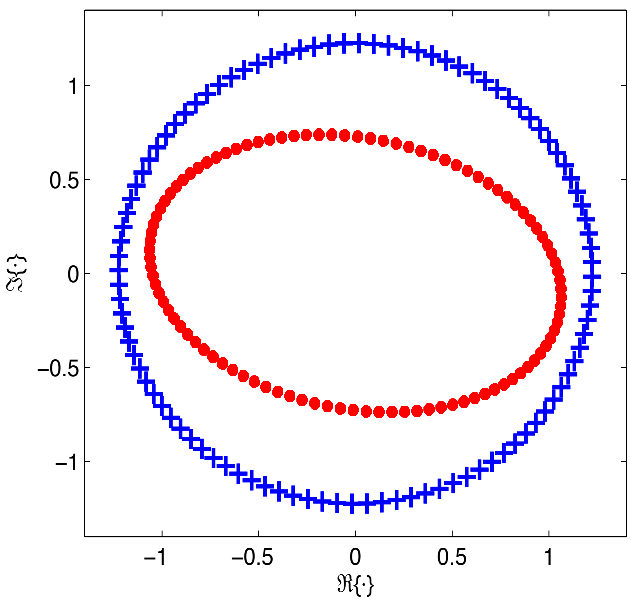
Noncircularity in unbalanced voltage conditions

○ **Balanced system:** $V_a(k) = V_b(k) = V_c(k)$, $A(k) = \text{const}$, $B(k) = 0$, and $v(k)$ is on a circle

○ **Unbalanced system:** $V_a(k)$, $V_b(k)$, $V_c(k)$ are not identical

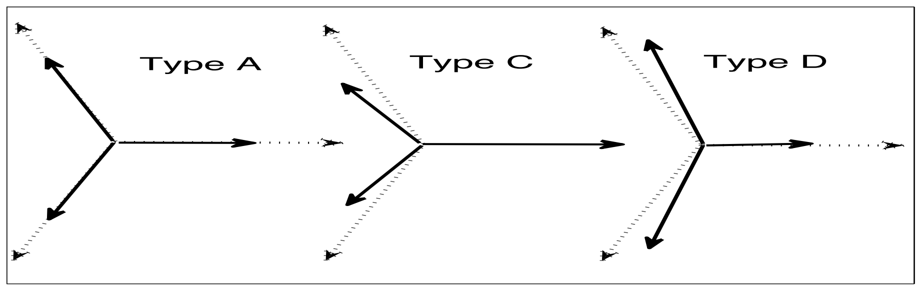
⊛ $A(k)$ is no longer constant, $B(k) \neq 0$

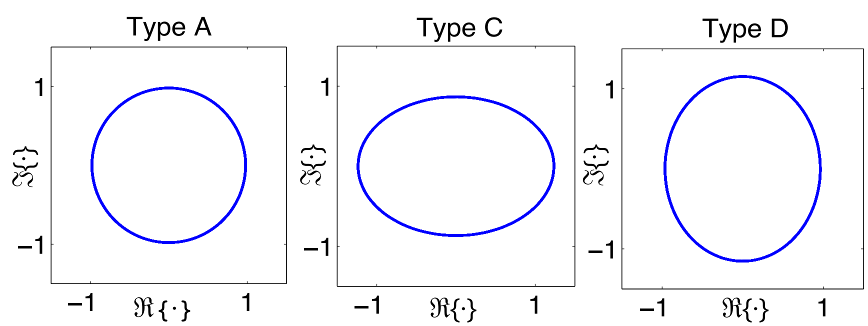
⊛ $v(k)$ is not on a circle $\rightarrow$ **a degree of noncircularity**



balanced and unbalanced system

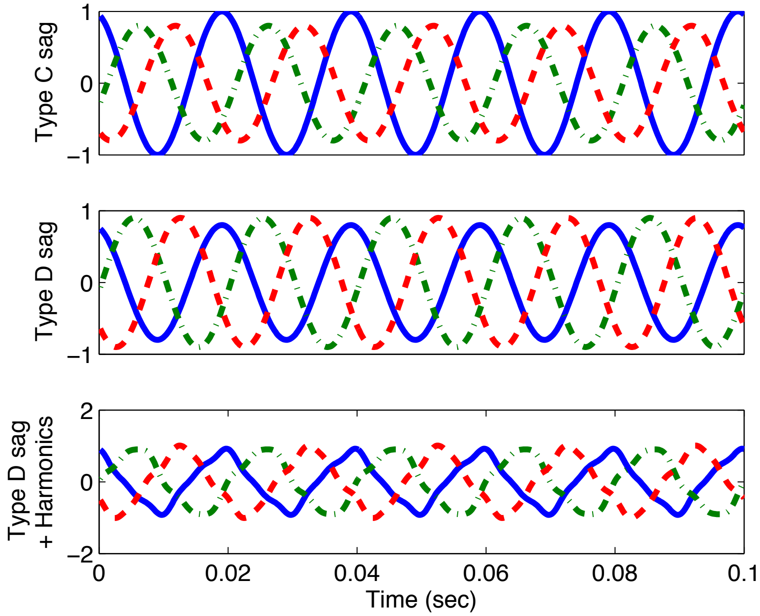
Some types of voltage sags

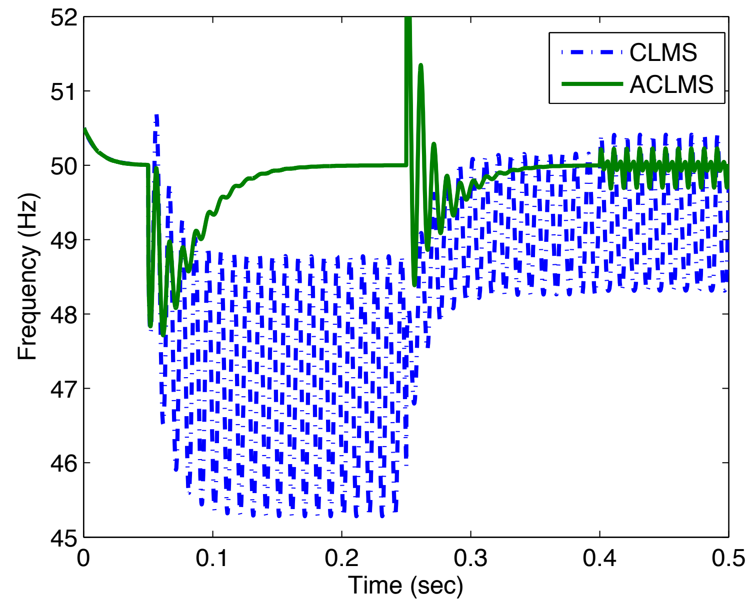




Their circularity diagrams

Simulations: Several successive sags





Phase voltages for different sags Linear and widely linear frequency estimation

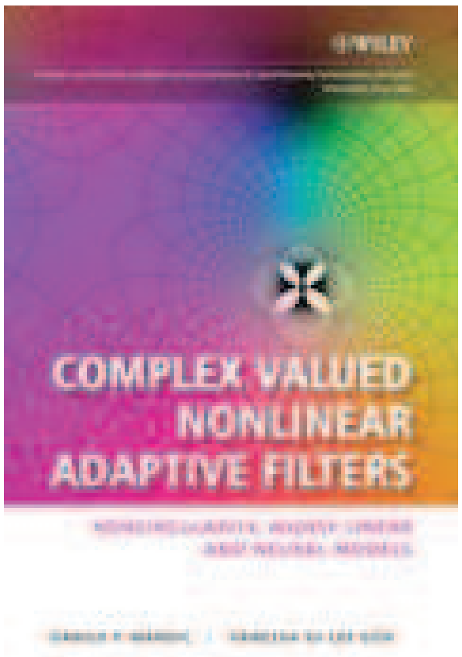
- Initially, the power system (50Hz) was operating normally and both CLMS and ACLMS converged to 50Hz
- The widely linear ACLMS had advantage in the subsequent type C and D sags and under harmonics

Conclusions

- Adaptive processing of noncircular complex signals
- These arise e.g. due to the nonlinearity of transievers (I/Q imbalance mitigation), multipath, in some modulation schemes (BPSK, GSMK, PAM, offset-QPSK) widely used in practical communications systems
- Standard solutions assume second order circularity of signal distributions and are inadequate when the signals are observed through nonlinear sensors, mixtures of sources, or noise model which is not doubly white
- This is achieved based on *augmented complex statistics* and *widely linear modelling*
- The complex LMS (CLMS) and augmented CLMS (ACLMS) introduced
- Convergence of CLMS and ACLMS – from the booklet provided (Chapter 6)
- This promises enhanced practical solutions in a variety of applications (interference supression, DoA estimation, blind estimation)

A Comprehensive Account of Widely Linear Modeling

D.Mandic and V. Goh, "Complex Valued Nonlinear Adaptive FIlters: Noncircularity, Widely Linear and Neural Models", Wiley 2009.



- Unified approach to the design of complex valued adaptive filters and neural networks
- Augmented learning algorithms based on widely linear models
- Suitable for processing both second order circular (proper) and noncircular (improper) complex signals
- ACLMS, augmented Kalman filters, augmented CRTRL, linear and nonlinar IIR filters
- Adaptive stepsizes, dynamical range reduction, collaborative adaptive filters, statitical tests for the validity of complex representations

Some back–up material

App: The RLS algorithms and its variants

From $\quad \mathbf{w}(n+1) = \mathbf{R}^{-1}(n+1)\mathbf{p}(n+1)$, we have

$$\mathbf{w}(n+1) = \left[\mathbf{R}^{-1}(k) - \frac{\mathbf{R}^{-1}(k)\mathbf{x}(k+1)\mathbf{x}^T(k+1)\mathbf{R}^{-1}(k)}{\mathbf{x}^T(k+1)\mathbf{R}^{-1}(k)\mathbf{x}(k+1)+1}\right]\left[\mathbf{p}(n) + d(n)\mathbf{x}(n)\right]$$

After some grouping of the terms above, we arrive at

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mathbf{R}^{-1}(n)}{1 + \mathbf{x}^T(n)\mathbf{R}^{-1}(n)\mathbf{x}(n)}\, e(n)\mathbf{x}(n)$$

∘ the term $\mathbf{R}^{-1}$ "filters" the direction and length of the data vector $\mathbf{x}$

∘ The denominator $1 + \mathbf{x}^T(n)\mathbf{R}^{-1}(n)$ is a measure of input signal power, normalised by $\mathbf{R}^{-1}$.

∘ This normalisation makes the power proportional to the data length N and not to the actual signal level $\looparrowright$ it also decorrelates the data.

Related algorithms:

On the next slide.

App: The RLS algorithms and its variants

Accelerated LMS:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{C}(n)\mathbf{x}(n), \quad \mathbf{C} \text{ a chosen matrix}$$

Normalised LMS:

$$\mu(n) = \frac{\mu}{\varepsilon + \mathbf{x}^T(n)\mathbf{x}(n)}, \quad 0 < \mu < 2$$

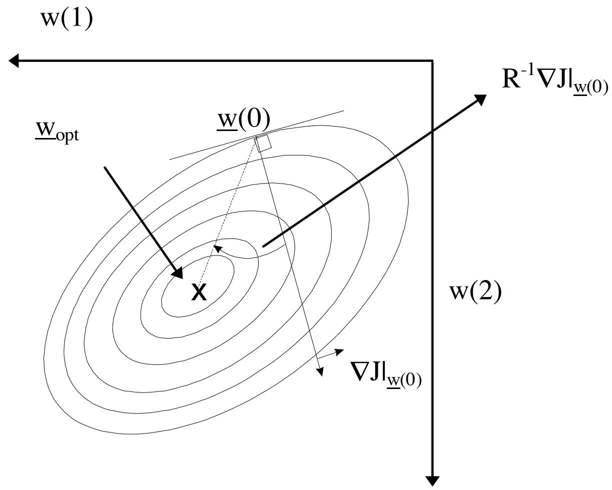
Exponentially weighted RLS (RLS with the forgetting factor)

$$J(n) = \sum_{i=0}^{n} \lambda^{n-i} \big| d(i) - y(i) \big|^2, \quad \lambda \text{ is a forgetting factor}$$

- Forgetting factor $\varepsilon \in (0,1]$, but typically > 0.95
- The forgetting factor introduces an effective window length of $\frac{1}{1-\varepsilon}$

App: Comparison between LMS & RLS

The role of $\mathbf{R}^{-1}$ in RLS is to rotate the direction of descent of the LMS algorithm towards the minimum of the cost function independent of the nature, or colouration, of the input.



The operation $\mathbf{R}^{-1}\mathbf{x}[k]$ is a **pre-whitening operation**, compare with transform domain adaptive filtering.

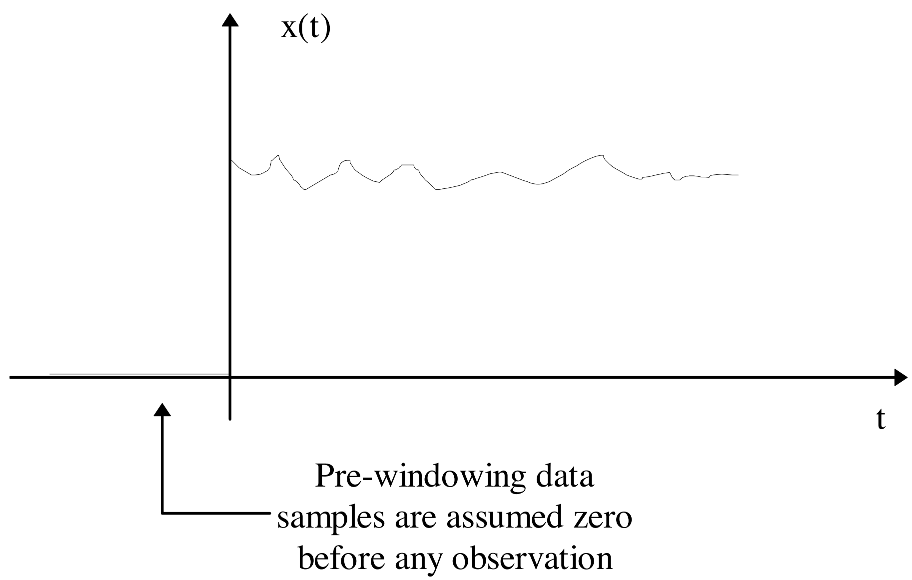
The convergence of the RLS in a high SNR environment ($> 10dB$) is of an order $\mathcal{O}(2p)$, whereas for LMS $\mathcal{O}(10p)$.

The misadjustment performance of RLS is essentially zero because it is deterministic and matches the data where $\lambda = 1$.

App: Pre–windowing

- To initialise RLS we need to make some assumptions about the input data

- The most common one is that of prewindowing, that is $\forall k < 0, x(k) = 0$



App: Complete RLS algorithm

- Initialisation $\mathbf{w}(0) = \mathbf{0}$, $\mathbf{P}(0) = \delta\mathbf{I}$, $\mathbf{P}(k) = \mathbf{R}^{-1}$

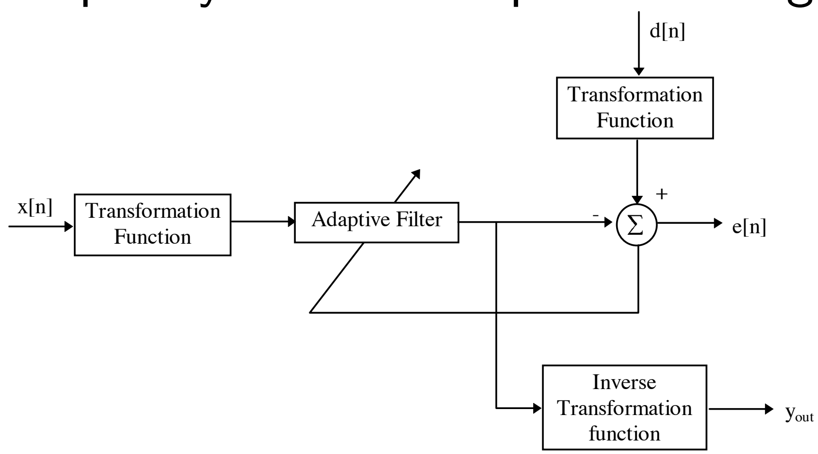
- For each k

$$
\begin{aligned}
\gamma(k+1) &= \mathbf{x}^T(k+1)\mathbf{P}(k)\mathbf{x}(k+1) \\
\mathbf{k}(k+1) &= \frac{\mathbf{P}(k)\mathbf{x}(k+1)}{1+\gamma(k+1)} \\
\mathbf{e}(k+1) &= \mathbf{d}(k+1) - \mathbf{x}^T(k+1)\mathbf{w}(k+1) \\
\mathbf{w}(k+1) &= \mathbf{w}(k)+\mathbf{k}(k+1)\mathbf{e}(k+1) = \mathbf{w}(k)+\mathbf{R}^{-1}(k+1)\mathbf{x}(k+1)\mathbf{e}(k+1) \\
\mathbf{P}(k+1) &= \mathbf{P}(k) - \mathbf{k}(k+1)\mathbf{x}^T(k)\mathbf{P}(k)
\end{aligned}
$$

- Computational complexity of RLS is an $\mathcal{O}(p^2)$ in terms of multiplications and additions as compared to LMS and NLMS which are $\mathcal{O}(2N)$

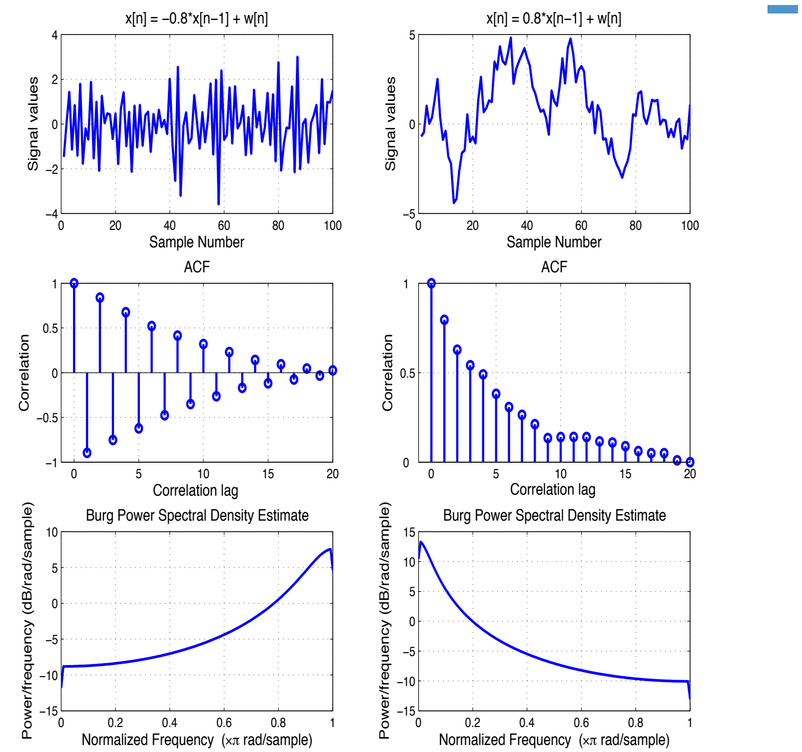
App: Transform domain signal processing: Usually complex valued

Frequency domain adaptive filtering



A transform can be applied to the inputs of an adaptive filter in order to maximise the performance of the LMS algorithm, i.e. a white input with equal eigenvalues.

Various techniques can be used such as Lattice Filters, the FFT which requires the Complex LMS algorithm, the Discrete Cosine or Wavelet transforms, or sub-band filters.



$a < 0 \rightarrow$ highpass, $a > 0 \rightarrow$ lowpass

$$x(k) = a_1 x(k-1) + w(k)$$

Highpass signal: fast changing in time $\looparrowright$ however, smooth spectrum ☺

App: Forms of CLMS and Complex Wiener Filter

The following formulations produce identical results

$$y(k) = \mathbf{x}^T(k)\mathbf{w}(k) = \mathbf{w}^T(k)\mathbf{x}(k) \quad \rightarrow \quad \mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\mathbf{x}^*(k)$$

$$y(k) = \mathbf{w}^H(k)\mathbf{x}(k) = \mathbf{x}^T(k)\mathbf{w}^*(k) \quad \rightarrow \quad \mathbf{w}(k+1) = \mathbf{w}(k) + \mu e^*(k)\mathbf{x}(k)$$

$$y(k) = \mathbf{x}^H(k)\mathbf{w}(k) = \mathbf{w}^T(k)\mathbf{x}^*(k) \quad \rightarrow \quad \mathbf{w}(k+1) = \mathbf{w})k) + \mu e(k)\mathbf{x}(k)$$

We have mainly used the first formulation. For instance, to obtain the **Wiener solution** we can use the second formulation to minimize

$$J(\mathbf{w}) = E\big[e(k)e^*(k)\big] = E\Big[\big(d(k) - \mathbf{w}^H\mathbf{x}(k)\big)\big(d(k) - \mathbf{w}^H\mathbf{x}(k)\big)^*\Big]$$

$$J(\mathbf{w}) = E\big[|d(k)|^2\big] - \mathbf{w}^H E\big[\mathbf{x}(k)d^*(k)\big] - \mathbf{w}^T E\big[\mathbf{x}^*(k)d(k)\big] + \mathbf{w}^H E\big[\mathbf{x}(k)\mathbf{x}^H(k)\big]\mathbf{w}$$

$$= \quad \sigma_d^2 - \mathbf{w}^H\mathbf{p} - \mathbf{p}^H\mathbf{w} + \mathbf{w}^H\mathbf{R}\mathbf{w} \quad \rightarrow \quad \nabla_{\mathbf{w}} J(\mathbf{w}) = 2\mathbf{R}\mathbf{w} - 2\mathbf{p} = \mathbf{0}$$

The optimum solution $\mathbf{w}_o$ and the minimum mean square error $J_{\min}$

$$\mathbf{w}_o = arg\min_{\mathbf{w}} J(\mathbf{w}) = \mathbf{R}^{-1}\mathbf{p} \qquad\qquad J_{min} = J(\mathbf{w}_o) = \sigma_d^2 - \mathbf{p}^H\mathbf{R}^{-1}\mathbf{p}$$

To emphasise that $J(\mathbf{w})$ is quadratic in $\mathbf{w}$ and has a global minimum for $\mathbf{w} = \mathbf{w}_o$,

$$J(\mathbf{w}) = J_{min} + (\mathbf{w} - \mathbf{w}_o)^H\,\mathbf{R}\,(\mathbf{w} - \mathbf{w}_o)$$

App: Convergence of CLMS

Consider the expected value for the CLMS update

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu E\big[e(k)\mathbf{x}^*(k)\big] \tag{1}$$

It has converged when $\mathbf{w}(k+1) = \mathbf{w}(k) = \mathbf{w}(\infty)$ and thus the weight update $\Delta\mathbf{w}(k) = \mu E[e(k)\mathbf{x}^*(k)] = \mathbf{0}$.

This is achieved for

$$\mathbf{0} = E\big[d^*(k)\mathbf{x}(k)\big] - E\big[\mathbf{x}(k)\mathbf{x}^H(k)\big]\mathbf{w}(k) \quad \Leftrightarrow \quad \mathbf{R}\mathbf{w} = \mathbf{p} \tag{2}$$

that is, for the Wiener solution.

The condition $E[e(k)\mathbf{x}^*(k)] = \mathbf{0}$ is called the "orthogonality condition" and states that the output error of the filter and the tap input vector are orthogonal ($e \perp \mathbf{x}$) when the filter has converged to the optimal solution.

App: Types of Convergence of CLMS

It is of greater interest, however, to analyse the evolution of the weights in time. As with any other estimation problem, we need to analyse the "bias" and "variance" of the estimator, that is:

- Convergence in the mean, to ascertain whether $\mathbf{w}(k) \rightarrow \mathbf{w}_o$ when $k \rightarrow \infty$;
- Convergence in the mean square, in order to establish whether the variance of the weight error vector $\mathbf{v}(k) = \mathbf{w}(k) - \mathbf{w}_o(k)$ approaches J_{min} as $k \rightarrow \infty$.

The analysis of convergence of linear adaptive filters is made mathematically tractable if we use so called **independence assumptions**, such that the filter coefficients are

⊛ statistically independent of the data currently in filter memory, and

⊛ $\{d(l), x(l)\}$ is independent of $\{d(k), x(k)\}$ for $k \neq l$.

App: Convergence of CLMS in the Mean

Assume $d(k) = \mathbf{x}^H(k)\mathbf{w}_o + q(k),\ q(k) \sim \mathcal{N}(0, \sigma_q^2$. Then

$$
\begin{aligned}
e(k) &= \mathbf{x}^H(k)\mathbf{w}_o + q(k) - \mathbf{x}^H(k)\mathbf{w}(k) \\
\mathbf{w}(k+1) &= \mathbf{w}(k) + \mu\mathbf{x}(k)\mathbf{x}^H\mathbf{w}_o - \mu\mathbf{x}(k)\mathbf{x}^H(k)\mathbf{w}(k) + \mu q(k)\mathbf{x}(k)
\end{aligned}
$$

Subtract $\mathbf{w}_o$ from both sides, the weight error vector $\mathbf{v}(k) = \mathbf{w}(k) - \mathbf{w}_o$

$$
\begin{aligned}
\mathbf{v}(k+1) &= \mathbf{v}(k) - \mu\mathbf{x}(k)\mathbf{x}^H(k)\mathbf{v}(k) + \mu q(k)\mathbf{x}(k) \\
E[\mathbf{v}(k+1)] &= \big(\mathbf{I} - \mu E[\mathbf{x}(k)\mathbf{x}^H(k)]\big)E[\mathbf{v}(k)] + \mu E[q(k)\mathbf{x}(k)] = \big(\mathbf{I} - \mu\mathbf{R}\big)E[\mathbf{v}(k)]
\end{aligned}
$$

Unless the correlation matrix $\mathbf{R}$ is diagonal, there will be cross–coupling between the coefficients of the weight error vector. Rotating the weight error vector $\mathbf{v}(k)$ by the eigenmatrix $\mathbf{Q}$, that is, $\mathbf{v}'(k) = \mathbf{Q}\mathbf{v}(k)$, decouples the evolution of its coefficients, so that the **modes of convergence**

$$
\mathbf{v}'(k+1) = (\mathbf{I} - \mu\boldsymbol{\Lambda})\,\mathbf{v}'(k) \qquad \overset{|1-\mu\lambda_n|<1}{\Longrightarrow} \qquad 0 < \mu < \frac{2}{\lambda_{max}} \approx \frac{2}{\text{tr}[\mathbf{R}]}
$$

Since $\mathrm{tr}\mathbf{R} = \mathrm{NE}[|\mathrm{x}(\mathrm{k})|^2|$, an easier to estimate bound $0 < \mu < \frac{2}{NE[|x(k)|^2|]}$.

App: Convergence of CLMS in the Mean Square

As the filter coefficients converge in the mean, they fluctuate around their optimum values $\mathbf{w}_o$. As a result, the mean square error $\xi(k) = E[|e(k)|^2]$ exceeds the minimum mean square error J_{min} by an amount referred to as the **excess mean square error**, denoted by $\xi_{EMSE}(k)$, that is

$$\xi(k) = J_{min} + \xi_{EMSE}(k) \quad \overset{J_{min}=\sigma_d^2}{\Longrightarrow} \quad \xi(k) = \sigma_q^2 + E\big[\mathbf{v}^H(k)\mathbf{R}\mathbf{v}(k)\big] = \sigma_q^2 + \text{tr}[\mathbf{R}\mathbf{K}(k)]$$

We have used the identity $E[\mathbf{v}^H(k)\mathbf{R}\mathbf{v}(k)] = \text{tr}[\mathbf{R}\mathbf{K}(k)] = \text{tr}[\mathbf{K}(k)\mathbf{R}]$, where $\mathbf{K}(k) = E[\mathbf{v}(k)\mathbf{v}^H(k)]$.

The excess mean square error depends on second order statistical properties of $d(k), \mathbf{x}(k), \mathbf{w})k), e(k)$. **The plot showing time evolution of the mean square error is called the *learning curve*.**

For convergence in the **mean square** the misadjustment

$$\mathcal{M} = \frac{\xi_{EMSE}(\infty)}{\xi_{min}} = \frac{\xi_{EMSE}(\infty)}{\sigma_q^2} \approx \frac{1}{2}\mu\sigma_x^2 N$$

must be bounded and positive, that is, $(1 - \frac{1}{2}\mu\text{tr}[\mathbf{R}] > 0)$, and therefore

$$0 < \mu < 2/\text{tr}[\mathbf{R}] \quad \leftrightarrow \quad 0 < \mu < 2/(\sigma_{\text{x}}^2 \text{N}).$$

App: The Complex LMS Algorithm - Step by Step

Consider a complex valued FIR filter.

The weight update equation for a real valued filter is

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \eta e(k)\mathbf{x}(k)$$

Now, the weights and errors and teaching signal and input are complex valued.

Hence

$$\begin{aligned} e(k) &= e_r(k) + \jmath e_i(k) \\ d(k) &= d_r(k) + \jmath d_i(k) \\ \mathbf{x}(k) &= \mathbf{x}_r(k) + \jmath \mathbf{x}_i(k) \\ \mathbf{w}(k) &= \mathbf{w}_r(k) + \jmath\, \mathbf{w}_i(k) \\ y(k) &= \mathbf{x}^T(k)\mathbf{w}(k) \end{aligned}$$

App: The CLMS Cost function

The complex LMS should simultaneously adapt the real and imaginary part, minimising in some sense both $e_r(k)$ and $e_i(k)$, with respect to the average total error power, given by

$$E\left[e(k)e^*(k)\right] = \frac{1}{2}E\left[e_r^2(k) + e_i^2(k)\right] = \frac{1}{2}E\left[e_r^2(k)\right] + \frac{1}{2}E\left[e_i^2(k)\right]$$

Since the two components of the error are in quadrature relative to each other, they cannot be minimised independently.

The derivation of the complex LMS is similar to the derivation fo the original LMS, except that the rules of complex algebra must be observed.

Notice that

$$\left(\mathbf{x}^T(k)\mathbf{w}(k)\right)^* = \left(\mathbf{x}^*(k)\right)^T \mathbf{w}^*(k),$$

e.g.

$$\left[(x_r + \jmath x_i)(w_r + \jmath w_i)\right] = \left[x_r w_r - x_i w_i + \jmath(x_r w_i + x_i w_r)\right]$$

App: The CLMS Cost function

After conjugation we have

$$[x_r w_r - x_i w_i - \jmath(x_r w_i + x_i w_r)].$$

On the other hand

$$x^* w^* = [(x_r - \jmath x_i)(w_r - \jmath w_i] = [x_r w_r - x_i w_i - \jmath(x_r w_i + x_i w_r)],$$

which is the same as when we conjugate the whole expression.

App: The CLMS Derivation

$$e^*(k) = d^*(k) - \left(\mathbf{x}^T(k)\right)^* \mathbf{w}^*(k)$$

Therefore, for a GD adaptation we have

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{1}{2}\eta\nabla_{\mathbf{w}}\left(e(k)e^*(k)\right)$$

where $$\nabla(e(k)e^*(k)) = \nabla_r(e(k)e^*(k)) + \jmath\nabla_i(e(k)e^*(k)).$$

Now,

$$\nabla_r(e(k)e^*(k)) = \begin{bmatrix} \frac{\partial(e(k)e^*(k))}{\partial w_{1r}} \\ \frac{\partial(e(k)e^*(k))}{\partial w_{2r}} \\ \vdots \\ \frac{\partial(e(k)e^*(k))}{\partial w_{Nr}} \end{bmatrix} = e(k)\nabla_r(e^*(k)) + e^*(k)\nabla_r(e(k))$$

App: The CLMS Derivation – contd

Notice that (in a simplified way)

$$\begin{aligned} e(k) &= d(k) - x(k)w(k) = d(k) - \\ &- [x_r(k)w_r(k) - x_i(k)w_i(k) + \jmath(x_r(k)w_i(k) + x_i(k)w_r(k))] \\ e^*(k) &= d(k) - [x_r(k)w_r(k) + x_i(k)w_i(k) - \jmath(x_r(k)w_i(k) + x_i(k)w_r(k))] \end{aligned}$$

The partial derivatives wrt to w_r and w_i are

$$\begin{aligned} \frac{\partial e(k)}{\partial w_r(k)} &= -[x_r(k) + \jmath x_i(k)] = -\mathbf{x}(k) \\ \frac{\partial e(k)}{\partial w_i(k)} &= -[-x_i(k) + \jmath x_r(k)] = -\jmath\mathbf{x}(k) \\ \frac{\partial e^*(k)}{\partial w_r(k)} &= -[x_r(k) - \jmath x_i(k)] = -\mathbf{x}^*(k) \\ \frac{\partial e^*(k)}{\partial w_i(k)} &= -[x_i(k) - \jmath x_r(k)] = -[-\jmath\mathbf{x}^*(k)] \end{aligned}$$

App: The CLMS Derivation – Complex Gradients

The instantaneous gradient with respect to its real and imaginary component becomes

$$
\begin{aligned}
\nabla_r(e(k)e^*(k)) &= e(k)\nabla_r(e^*(k)) + e^*(k)\nabla_r(e(k)) \\
&= e(k)(-\mathbf{x}^*(k)) + e^*(k)(-\mathbf{x}(k)) \\
\nabla_i(e(k)e^*(k)) &= e(k)\nabla_i(e^*(k)) + e^*(k)\nabla_i(e(k)) \\
&= e(k)\left(\jmath\mathbf{x}^*(k)\right) + e^*(k)\left(-\jmath\mathbf{x}(k)\right)
\end{aligned}
$$

Now, applying the method of steepest descent, to the real and imaginary part of the weights we have

$$
\begin{aligned}
\mathbf{w}_r(k+1) &= \mathbf{w}_r(k) - \frac{1}{2}\eta\nabla_r\left(e(k)e^*(k)\right) \\
\mathbf{w}_i(k+1) &= \mathbf{w}_i(k) - \frac{1}{2}\eta\nabla_i\left(e(k)e^*(k)\right)
\end{aligned}
$$

App: The CLMS Derivation – Filter Update

Having in mind that $\mathbf{w}(k+1) = \mathbf{w}_r(k+1) + \jmath\mathbf{w}_i(k+1)$, we have

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{1}{2}\eta\left[\nabla_r\left(e(k)e^*(k)\right) + \jmath\nabla_i\left(e(k)e^*(k)\right)\right]$$

If the gradients are now substituted in the above equation, we have

$$\begin{aligned}
\nabla_r\left(e(k)e^*(k)\right) + \jmath\nabla_i\left(e(k)e^*(k)\right) &= \\
e(k)(-\mathbf{x}^*(k)) + e^*(k)(-\mathbf{x}(k)) + \jmath\left[e(k)\left(\jmath\mathbf{x}^*(k)\right) + e^*(k)\left(-\jmath\mathbf{x}(k)\right)\right] &= \\
-e_r\mathbf{x}^* - \jmath e_i\mathbf{x}^* - e_r\mathbf{x} + \jmath e_i\mathbf{x} - e_r\mathbf{x}^* - \jmath e_i\mathbf{x}^* + e_r\mathbf{x} - \jmath e_i\mathbf{x} &= \\
-2e_r\mathbf{x}^* - 2\jmath e_i\mathbf{x}^* = -2e(k)\mathbf{x}^*(k)
\end{aligned}$$

Therefore, the complex form of the LMS algorithm is given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \eta e(k)\mathbf{x}^*(k)$$

App: The Multivariate Complex Normal Distribution

Recall, the relationships like "$<$" or "$\geq$" make no sense in $\mathbb{C}$.

$$\mathbf{V} = cov(\mathbf{v}) = E[\mathbf{v}\mathbf{v}^H] = \mathbf{A}\mathbf{W}\mathbf{A}^H$$

Using the result by Vanden Bos 1995

$$\begin{aligned} \mathbf{w} &= \mathbf{A}^{-1}\mathbf{v} = \frac{1}{2}\mathbf{A}^H\mathbf{v} \\ det(\mathbf{W}) &= \left(\frac{1}{2}\right)^{2N} det(\mathbf{V}) \\ \mathbf{w}^T\mathbf{W}^{-1}\mathbf{w} &= \mathbf{v}^H\mathbf{V}^{-1}\mathbf{v} \end{aligned}$$

The multivariate *generalised complex normal distribution* (GCND) can now be expressed as

$$f(\mathbf{v}) = \frac{1}{\pi^N\sqrt{det(\mathbf{V})}}\, e^{-\frac{1}{2}\mathbf{v}^H\mathbf{V}^{-1}\mathbf{v}}$$

and has been derived without any restriction.

The $\mathbb{CR}$ calculus

Based on our earlier examples of nonanalytic functions $f(z) = z^*$ and $f(z) = |z|^2 = zz^*$, observe that:

- A function $f(z)$ can be non–holomorphic in the complex variable $z = x + \jmath y$, but still be analytic in real variables x and y, as for instance, $f(z) = z^*$ and $f(z) = zz^* = x^2 + y^2$;
- Both $f(z) = z^*$ and $f(z) = zz^*$ are holomorphic in z for $z^* = const$, and are also holomorphic in z^* when $z = const$.

The main idea behind both Wirtinger calculus and Brandwood's result, is to introduce so called *conjugate coordinates*

$$f(z) = f(z, z^*) = g(x, y) = \Re\{f\} + \jmath\Im\{f\} = u(x, y) + \jmath v(x, y)$$

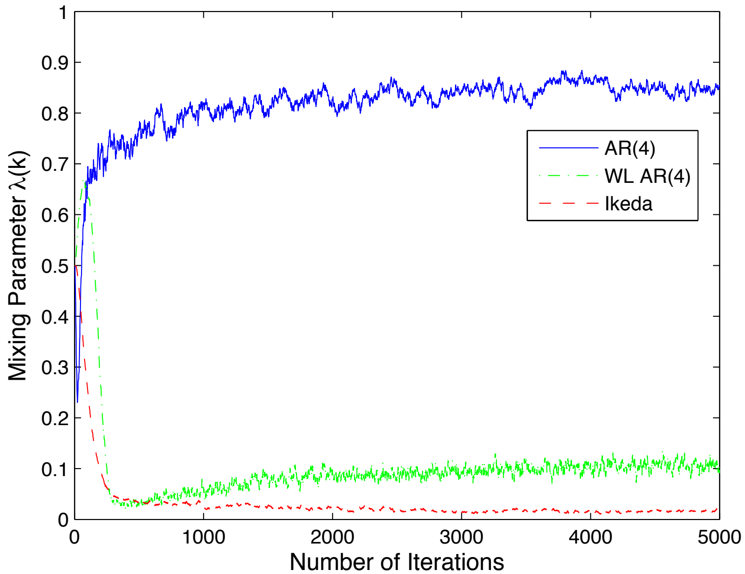
Then

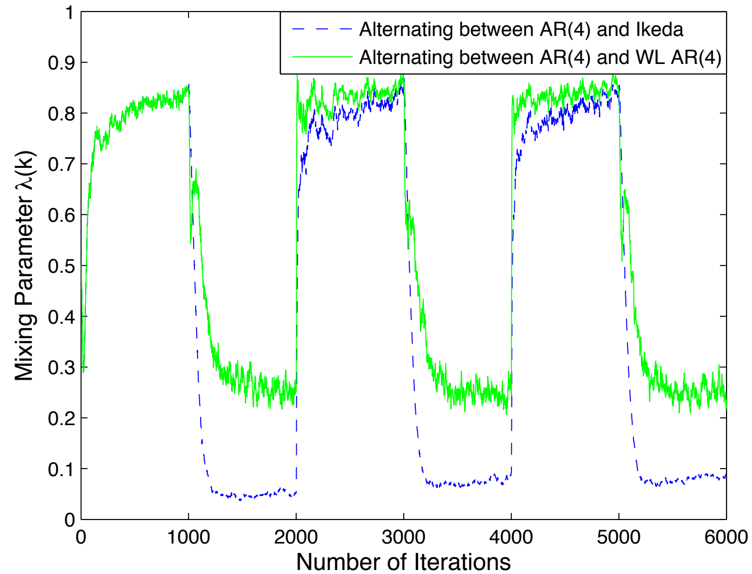
$$f + \frac{\partial f}{\partial \boldsymbol{\omega}^T}\boldsymbol{\omega} + \frac{1}{2!}\boldsymbol{\omega}^T \frac{\partial^2 f}{\partial \boldsymbol{\omega} \partial \boldsymbol{\omega}^T}\boldsymbol{\omega} + \cdots \quad \Rightarrow \quad \frac{\partial f}{\partial \mathbf{v}} = \frac{1}{2}\mathbf{A}^* \frac{\partial f}{\partial \boldsymbol{\omega}} \quad \Rightarrow \quad \frac{\partial f}{\partial \boldsymbol{\omega}} = \mathbf{A}^T \frac{\partial f}{\partial \mathbf{v}}$$

For an excellent overview see the web material by Kenneth Kreutz-Delgado

Identification of Noncircular Signals (prediction setting)

Left: dynamics of λ (proper vs improper) Right: Tracking proper vs improper





- For circular $AR(4)$, $\lambda(k) \rightarrow 1$
- For noncircular $AR(4)$ and Ikeda signals, $\lambda(k) \rightarrow 0$

◦ This is used to track the occurrence of noncircular signals in real time

◦ Also, a rough indication of the degree of noncircularity

⊛ Potentially useful in practical applications, in combination with other algorithms $\rightsquigarrow$ provides an additional degree of freedom

App: Notes