Outline:

Basically, to de-mystify multidimensional adaptive estimation

- Multidimensional and multichannel sensors – a unified approach to adaptive filtering of such signals
- Circularity – a unique signature of bivariate signals ⇨ second order circularity (properness) ⇨ augmented complex statistics
- Duality between the processing in $\mathbb{R}^2$ and $\mathbb{C}$ (isomorphism)
- The issue of complex gradient ⇨ $\mathbb{CR}$ calculus
- Covariance, pseudocovariance, and widely linear models
- Complex Wiener filter, complex least mean square (CLMS) and augmented CLMS (ACLMS)
- Magnitude-only, phase-only, and least mean magnitude phase (LMMP) approaches
- Multivariate adaptive filters (any number of data channels)
- Applications: communications, radar and sonar, target tracking, renewable energy, smart grid
We can use multidimensional filters within our usual adaptive filtering configurations!

**System Identification**

\[ y(k) = \sum e(k) \]

**Noise Cancellation**

\[ y(k) = e(k) \]

**Adaptive Prediction**

\[ d(k) = \sum x(k) \]

**Inverse System Modelling**

\[ y(k) = e(k) \]
Multidimensional 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
Why modelling in $\mathbb{C}$?

- Complex signals by design (communications, analytic signals, equivalent baseband representation 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 ↦ but careful: Bedrossian and Nutall theorems

Consider

◦ Amplitude modulated signal \( x(t) = m(t) \cos(\omega_0 t) \) → \( m(t) \) ↦ envelope
◦ Phase modulated signal \( x(t) = a \cos(\Phi(t)) \) → \( \Phi(t) \) ↦ 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 \( z(t) = x(t) + j\mathcal{H}(x(t)) = a(t)e^{j\Phi(t)} \)

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

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

**Remark #3:** For \( x(t) = a(t) \cos \Phi(t) \) ⇒ \( \mathcal{H}\{x(t)\} = a(t) \sin \Phi(t) \)

**Remark #4:** From instantaneous phase \( \Phi(t) \) → instantaneous frequency

\[
f(t) = \frac{d\Phi(t)}{dt}
\]

so we have an excellent resolution and do not depend on stationarity
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**
  \[
  C = E\{xx^T\} \quad \leadsto \quad C = E\{zz^H\}
  \]

- **Autoregressive model and Wiener solution**
  \[
  \mathbf{w} = \mathbf{R}^{-1}\mathbf{p} \quad \leadsto \quad \mathbf{w}^* = \mathbf{R}^{-1}\mathbf{p}
  \]

- **Least mean square (LMS)** \( \leadsto \) complex LMS [Widrow *et al.* 1975]
  \[
  \mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)x(k) \quad \leadsto \quad \mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)z^*(k)
  \]

  \[
  \mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\boldsymbol{\Pi}(k) \quad \leadsto \quad \mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\boldsymbol{\Pi}^*(k)
  \]

This is, however, valid only for circular complex random processes
Property of division algebras ⇔ complex (non)circularity

Noncircularity of the wind distribution

\[ v(k) = |v(k)|e^{j\Phi(k)} \]

Deterministic vs. Stochastic nature
Linear vs. Nonlinear nature

Determinism
Nonlinearity

Chaos
(a)

Wind speed
Wind direction

N
E

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

Real world signals are denoted by ’???’

○ ∃ a unique signature of complex signals?

○ ⇔ degree of noncircularity
Circular vs noncircular complex random variables

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

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

Pseudocovariance is \( p = E\{zz^T\} = E\{z^2\} \)

Circularity quotient: \( \rho = \frac{p}{c} \)

Circularity coefficient: \( |\rho| = \frac{|p|}{c} \)

Circularity quotient and coefficient quantify the degree of non-circularity
Some observations

Although a formalism similar to that used for real valued adaptive filters can also be used for complex valued adaptive filters, notice that in this case the cost function $J(k) = \frac{1}{2}|e(k)|^2 = \frac{1}{2}e(k)e^*(k) = \frac{1}{2}(e_x^2 + e_y^2)$ is a real valued function of complex variable.

- Standard complex differentiability is based on the Cauchy–Riemann (C–R) equations and imposes a stringent structure on complex holomorphic functions;

- Cost functions are real functions of complex variable, that is $J : \mathbb{C} \mapsto \mathbb{R}$, and so they are not differentiable in the complex sense, the Cauchy–Riemann equations do not apply, and we need to develop alternative, more general and relaxed, ways of calculating their gradients (CR-calculus);

- It is also desired that these generalised gradients are formally equivalent to standard complex gradients (their generic extensions) when applied to holomorphic (analytic) functions.
The Wiener solution can be obtained by minimising

\[ J(w) = E[e(k)e^*(k)] = E[(d(k) - w^H x(k))(d(k) - w^H x(k))^*] \]

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

\[ = \sigma_d^2 - w^H p - p^H w + w^H R w \quad \rightarrow \quad \nabla_w^* J(w) = Rw - p = 0 \]

The optimum solution \( w_o \) and the minimum mean square error \( J_{min} \)

\[ w_o = \arg \min_w J(w) = R^{-1} p \quad \quad J_{min} = J(w_o) = \sigma_d^2 - p^H R^{-1} p \]

Also, \( J(w) \) can be expressed as

\[ J(w) = J_{min} + (w - w_o)^H R (w - w_o) = J_{min} + v^H R v \]

by noticing that \( d(k) = w_o^H x(k) + q(k) \) and \( e(k) = (w_o - w(k))^H x_k + q(k) = v^H(k)x(k) \),

where \( q(k) \sim N(0, \sigma_q^2) \).

Notice that \( J(w) \) is quadratic in \( w \) and has a global minimum for \( w = w_o \), with \( J_{min} = \sigma_q^2 \).
Derivative of the cost function $\frac{1}{2}e(k)e^*(k)$ and CLMS

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

$$\mathbb{R} - \text{der: } \frac{\partial}{\partial z} = \frac{1}{2} \left[ \frac{\partial}{\partial x} - j \frac{\partial}{\partial y} \right]$$

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

and the gradient

$$\nabla_w J = \frac{\partial J(e, e^*)}{\partial w} = \left[ \frac{\partial J(e, e^*)}{\partial w_1}, \ldots, \frac{\partial J(e, e^*)}{\partial w_N} \right]^T = 2 \frac{\partial J}{\partial w^*} = \frac{\partial J}{\partial w^r} + j \frac{\partial J}{\partial w^i} \text{ (pseudogradient)}$$

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

$$y(k) = w^H(k)x(k)$$

e(k) = d(k) - y(k) = d(k) - w^H(k)x(k) \quad e^*(k) = d^*(k) - x^H(k)w(k)$$

and

$$\nabla_w J = \nabla_{w^*} J \quad \text{(conjugate gradient direction)}$$

$$w(k+1) = w(k) - \mu \frac{\partial e(k)}{\partial w^*(k)} e^*(k) = w(k) + \mu e^*(k)x(k)$$

Thus, no need for tedious computations ⇝ the CLMS is derived in one line
Which derivative to we choose to compute the gradient?

\( R \)-derivative vs. \( R^* \)-derivative?

Simulation for the CLMS derived using \( R \)-der. and \( R^* \)-der. \((w_o = 3)\)

Both \( R \)-der. & \( R^* \)-der follow the same trajectory...

...but \( R \)-der. leads to instability

\( \nabla w^* \) Gradient
\( \nabla w \) Gradient
Orthogonality principle & alternative forms for the CLMS

Consider the expected value for the CLMS update

$$E[w(k + 1)] = E[w(k)] + \mu E[e(k)x^*(k)]$$

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

This is achieved for $$E[d^*(k)x(k)] - E[x(k)x^H(k)]w(k) = 0 \iff Rw_o = p$$

that is, for the Wiener solution, $$w_o = R^{-1}p$$.

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

The following formulations for the CLMS produce identical results:

1. $$y(k) = x^T(k)w(k) = w^T(k)x(k) \rightarrow w(k + 1) = w(k) + \mu e(k)x^*(k)$$
2. $$y(k) = w^H(k)x(k) = x^T(k)w^*(k) \rightarrow w(k + 1) = w(k) + \mu e^*(k)x(k)$$
3. $$y(k) = x^H(k)w(k) = w^T(k)x^*(k) \rightarrow w(k + 1) = w(k) + \mu e(k)x(k)$$
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 \( w(k) \rightarrow w_o \) when \( k \rightarrow \infty \);

- **Convergence in the mean square**, in order to establish whether the variance of the weight error vector \( v(k) = w(k) - w_o(k) \) approaches \( J_{\text{min}} \) 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 \( k \neq l \).
Convergence of CLMS in the Mean

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

\[
e(k) = x^H(k)w_o + q(k) - x^H(k)w(k)
\]

\[
w(k + 1) = w(k) + \mu x(k)x^H w_o - \mu x(k)x^H(k)w(k) + \mu q(k)x(k)
\]

Subtract \( w_o \) from both sides, and define the weight error vector

\[
v(k) \overset{\text{def}}{=} w(k) - w_o
\]

\[
v(k + 1) = v(k) - \mu x(k)x^H(k)v(k) + \mu q(k)x(k)
\]

Applying the statistical expectation operator \( E\{\cdot\} \) gives the mean weight error recursion

\[
E[v(k + 1)] = (I - \mu E[x(k)x^H(k)])E[v(k)] + \mu E[q(k)x(k)] = 0
\]

\[
\implies E[v(k + 1)] = (I - \mu R)E[v(k)]
\]

**Challenge:** Unless the correlation matrix \( R \) is diagonal, there will be cross-coupling between the coefficients of the weight error vector.
Convergence of CLMS in the Mean (contd.)

**Solution:** Since the eigenvalue decomposition of $\mathbf{R} = \mathbf{Q}\Lambda\mathbf{Q}^H$, the rotation of the weight error vector $\mathbf{v}(k)$ by the eigenvector matrix $\mathbf{Q}$, that is, $\mathbf{v}'(k) \overset{\text{def}}{=} \mathbf{Q}^H \mathcal{E}\{\mathbf{v}(k)\}$, decouples the evolution of its coefficients.

**Proof:** Pre-multiply both sides of mean weight vector error recursion by $\mathbf{Q}^H$

$$
\mathbf{Q}^H \mathcal{E}[\mathbf{v}(k+1)] = \mathbf{Q}^H (\mathbf{I} - \mu \mathbf{Q}\Lambda \mathbf{Q}^H) \mathcal{E}[\mathbf{v}(k)] = (\mathbf{I} - \mu \Lambda) \mathbf{Q}^H \mathcal{E}[\mathbf{v}(k)]
$$

Since the eigenvector matrix is unitary, i.e. $\mathbf{Q}^H \mathbf{Q} = \mathbf{I}$, the modes of convergence are

$$
\mathbf{v}'(k+1) = (\mathbf{I} - \mu \Lambda) \mathbf{v}'(k)
$$

This recursion will be stable if the diagonal elements of $(\mathbf{I} - \mu \Lambda)$ satisfy

$$
|1 - \mu \lambda_i| < 1, \quad \text{for } i = 1, \ldots, N \quad \Rightarrow \quad 0 < \mu < \frac{2}{\lambda_{\text{max}}} < \frac{2}{\text{tr}[\mathbf{R}]}
$$

Since $\text{tr}[\mathbf{R}] = N \mathcal{E}\{|x(k)|^2\}$, an easier to estimate bound $0 < \mu < \frac{2}{N \mathcal{E}[|x(k)|^2]}$. 
Convergence of CLMS in the Mean Square

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

$$\xi(k) = J_{\text{min}} + \xi_{\text{EMSE}}(k) \quad \Rightarrow \quad \xi(k) = \sigma_d^2 + E[v^H(k)Rv(k)] = \sigma_q^2 + \text{tr}[RK(k)]$$

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

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

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

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

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

$$0 < \mu < 2/\text{tr}[R] \quad \leftrightarrow \quad 0 < \mu < 2/(\sigma_x^2N).$$
Does Circularity Influence Estimation in \( \mathbb{C} \)?

Voltage Sag: A magnitude and/or phase imbalance

- For balanced systems, \( v(k) = A(k)e^{j\omega k\Delta T} \rightarrow \) circular trajectory.

- Unbalanced systems, \( v(k) = A(k)e^{j\omega k\Delta T} + B(k)e^{-j\omega k\Delta T} \) are influenced by the “conjugate” component.

- We need the complex conjugate when the modelling the signal.

Circularity Diagram

Strictly linear model yields biased estimates when system is unbalanced
What are we doing wrong ⇝ 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

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

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

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

$$\text{thus} \quad \hat{y} = E[y_r|x_r, x_i] + jE[y_i|x_r, x_i]$$

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

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

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

$$y = h^T x + g^T x^*$$

We can now process general (noncircular) complex signals!
Widely linear autoregressive modelling in \( \mathbb{C} \)

Standard AR model of order \( n \) is given by

\[
z(k) = a_1 z(k - 1) + \cdots + a_n z(k - n) + q(k) = a^T z(k) + q(k),
\]

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

\[
a^* = C^{-1} c
\]

\[
\begin{bmatrix}
a_1^*
\quad a_2^*
\quad \vdots
\quad a_n^*
\end{bmatrix}
= \begin{bmatrix}
c(0) & c^*(1) & \cdots & c^*(n-1)

\quad c(1) & c(0) & \cdots & c^*(n-2)

\quad \vdots & \vdots & \ddots & \vdots

\quad c(n-1) & c(n-2) & \cdots & c(0)
\end{bmatrix}^{-1}
\begin{bmatrix}
c(1) \\
c(2) \\
\vdots \\
c(n)
\end{bmatrix}
\]

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

Widely linear model

Widely linear normal equations

\[
y(k) = h^T(k) x(k) + g^T(k) x^*(k) + q(k)
\]

\[
\begin{bmatrix}
h^* \\
g^*
\end{bmatrix}
= \begin{bmatrix}
C & P
\end{bmatrix}^{-1}
\begin{bmatrix}
c \\
p^*
\end{bmatrix}
\]

where \( h \) and \( g \) are coefficient vectors and \( x \) 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

© D. P. Mandic
Adaptive Signal Processing & Machine Intelligence 23
The Augmented (widely linear) CLMS (ACLMS)

Wide linear model

\[ y(k) = h^T(k)z(k) + g^T(k)z^*(k) \]

\[ h(k + 1) = h(k) - \mu \nabla h^*J \quad \Rightarrow \quad \nabla h^*J = -e(k)x^*(k) \]

\[ g(k + 1) = g(k) - \mu \nabla g^*J \quad \Rightarrow \quad \nabla g^*J = -e(k)x(k) \]

Therefore, the ACLMS update

\[ h(k + 1) = h(k) + \mu e(k)x^*(k) \]

\[ g(k + 1) = g(k) + \mu e(k)x(k) \]

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

\[ w^a(k + 1) = w^a(k) + \eta e^a(k)x^a^*(k) \]

where \( \eta = \mu_h = \mu_g \), \( w^a(k) = [h^T(k), g^T(k)]^T \), \( x^a(k) = [x^T(k), x^H(k)]^T \),

\[ e^a(k) = d(k) - x^a^T(k)w^a(k) \]
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

(a) Modulus of complex wind over one day

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

Different wind regimes \(\leadsto\) different dynamics,
\[
v(k) = |v(k)|e^{j\Phi(k)}, \quad |v| - \text{speed, } \Phi - \text{direction}
\]

Different dynamics \(\leadsto\) different circularity properties \(\leadsto\) impact of ACLMS
Real bivariate or complex: Isomorphism between $\mathbb{C}$ and $\mathbb{R}^2$
(also serves as a basis for the CR calculus)

\[
\begin{bmatrix}
  z \\
  z^*
\end{bmatrix}
\leftrightarrow
\begin{bmatrix}
  \frac{1}{2} & j \\
  j & -\frac{1}{2}
\end{bmatrix}
\begin{bmatrix}
  x \\
  y
\end{bmatrix}
\]

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

\[
\begin{bmatrix}
  z \\
  z^*
\end{bmatrix}
\leftrightarrow
\begin{bmatrix}
  \begin{bmatrix} I & jI \\ I & -jI \end{bmatrix} \\
  \begin{bmatrix} I & jI \\ I & -jI \end{bmatrix}
\end{bmatrix}
\begin{bmatrix}
  x \\
  y
\end{bmatrix}
\]

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

\[
v = [z_1, z_1^*, \ldots, z_N, z_N^*]^T
\]

\[
v = Aw, \quad w = [x_1, y_1, \ldots, x_N, y_N]^T
\]

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

Desired signal: \( \hat{d}(k) = \hat{d}_r(k) + j\hat{d}_i(k) \)
Input: \( x(k) = x_r(k) + jx_i(k) \)

The bivariate (dual channel) real filter:
\[
\begin{bmatrix}
\hat{d}_r(k) \\
\hat{d}_i(k)
\end{bmatrix}
= 
\begin{bmatrix}
 a^T(k) & b^T(k) \\
 c^T(k) & d^T(k)
\end{bmatrix}
\begin{bmatrix}
 x_r(k) \\
 x_i(k)
\end{bmatrix}
\]

The strictly linear model \( \hat{d}(k) = w^H(k)x(k) \) \( (w(k) = w_r(k) + jw_i(k)) \) in the CLMS is highly restrictive since it imposes the condition
\[ a(k) = d(k) = w_r(k) \quad \text{and} \quad b(k) = -c(k) = w_i(k) \]

Augmented CLMS:
\[
\begin{bmatrix}
\hat{d}_r(k) \\
\hat{d}_i(k)
\end{bmatrix}
= 
\begin{bmatrix}
 (h_r(k) + g_r(k))^T & (h_i(k) - g_i(k))^T \\
 -(h_i(k) + g_i(k))^T & (h_r(k) - g_r(k))^T
\end{bmatrix}
\begin{bmatrix}
 x_r(k) \\
 x_i(k)
\end{bmatrix}
\]

Sufficient degrees of freedom to model general complex signals!

The bound on the step-size which preserves convergence: \( 0 < \mu < \frac{2}{\lambda_{\max}} \)

The bivariate and augmented complex correlation matrices related as
\[
C^a = E[zz^aH] = E[A\omega\omega^TA^H] = AWA^H
\]

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

ACLMS and DCRLMS converge at the same speed when \( \mu_r = 2\mu_{aclms} \).
Duality: Simulations

AR(4) and Ikeda signal

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>AR4</th>
<th>Ikeda</th>
<th>Wind</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_p ) for DCRLMS</td>
<td>5.8423</td>
<td>3.9733</td>
<td>13.2604</td>
</tr>
<tr>
<td>( R_p ) for CLMS</td>
<td>6.6380</td>
<td>2.4278</td>
<td>14.2941</td>
</tr>
<tr>
<td>( R_p ) for ACLMS</td>
<td>6.6096</td>
<td>4.0330</td>
<td>14.8926</td>
</tr>
<tr>
<td>( R_p ) for DCRLMS (double ( \mu ))</td>
<td>6.6096</td>
<td>4.0330</td>
<td>14.8926</td>
</tr>
</tbody>
</table>
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)
Part 2: Advanced learning algorithms and applications of Complex-Valued Adaptive Filters
Consider the problem of using a zero-mean r.v. $z \in \mathbb{C}$ to estimate its complex conjugate, that is

$$\hat{z}^* = w^* z$$

find an estimate of $w$ that minimizes

$$J_{MSE} = E\{|e|^2\} = E\{|z^* - \hat{z}^*|^2\}$$

The Wiener solution

$$w_{opt} = c^{-1}r = \frac{E\{z^2\}}{E\{|z|^2\}} = \frac{p}{c}$$

→ Circularity Quotient!!!
Real time tracker of complex circularity

Idea: Use an **adaptive filter** to estimate the complex conjugate of a signal using the original signal as the input.

Since the complex least mean square (CLMS) [Widrow 1975] estimates the Wiener solution, we can configure it with input signal $z_k$ and the desired signal $z_k^*$. The one-tap filter weight is the estimate of the circularity quotient $\rho$.

**Adaptive filtering configuration**

**CLMS algorithm**

$$\hat{z}_k^* = w_k^* z_k$$
$$e_k = z_k^* - \hat{z}_k^*$$
$$w_{k+1} = w_k + \mu e_k^* z_k$$
Real time tracker of complex circularity

Simulations on synthetic white Gaussian data

The real and imaginary parts of the evolution of the CLMS weights when tracking circularity.

Synthetic signal was generated by concatenating three segments of zero-mean white Gaussian signals, $z_{i,k}$, with different properties, where

$$z_{i,k} = x_{i,k} + jy_{i,k} \quad z_i \sim \mathcal{N}(0, c, p_i)$$

$$i = \{1, 2, 3\}$$

Each segment:

| Sample, $k$ | $c$   | $p_i$   | $|\rho_i|$ |
|-------------|-------|---------|-----------|
| 1 – 1000    | 1     | 0.8j    | 0.8       |
| 1001 – 2000 | 1     | 0.6 + 0.4j | 0.72     |
| 2001 – 3000 | 1     | 0       | 0         |
Real time tracker of complex circularity

Simulations on wind sensor data

The circularity diagram of wind speeds in the low, medium and high dynamic regimes.

The estimate of circularity coefficient for wind signals in the low, medium and high dynamic regimes using the proposed algorithm.

Wind signal modelled as wind a complex number

\[ s = s_E + js_N \]
Circular vs. Noncircular: Fast convergence vs. accuracy
Best of both worlds ⇔ a Hybrid Filter

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

\[ x \left(1 - \lambda\right) + \lambda x \]

Convexity \(\Rightarrow\) existence and uniqueness of solution

Let **Filter1** be trained by CLMS and **Filter2** by ACLMS
Adaptation of the mixing parameter $\lambda$

To preserve their inherent characteristics, subfilters, $\text{Filter}_1$ and $\text{Filter}_2$ updated based on their own errors:

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

The convex mixing parameter $\lambda$ is updated based on

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

and the stochastic gradient based update of $\lambda$ becomes

$$\lambda(k+1) = \lambda(k) + \mu_\lambda \left[ e(k) \left( y_{aclms}(k) - y_{clms}(k) \right)^* + e^*(k) \left( y_{clms}(k) - y_{aclms}(k) \right) \right]$$

**We must ensure that the value of $\lambda(k)$ belongs to $0 \leq \lambda(k) \leq 1$.**
Performance of hybrid filters – prediction setting

consider an LMS/GNGD hybrid – GNGD is fast, LMS with small $\mu$ has good $M$.

Hybrid attempts to follow the subfilter with better performance. If one of the subfilters diverges, hybrid filters still converges.

Learn. curves for pred.: Left $\leftrightarrow$ linear signal  Right $\leftrightarrow$ nonlinear signal
The hybrid CLMS ↔ 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 convergence and excellent steady state properties!

© D. P. Mandic

Adaptive Signal Processing & Machine Intelligence 39
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*. 
On phase-only cost functions

The least mean phase (LMP) learning is based on the cost function

\[ J_p(k) = (\angle d_k - \angle y_k)^2 \]

where \( e(k) = \angle d_k - \angle y(k) \), and \( y(k) = w^H(k)x(k) \).

Then,

\[ J_p(k) = (\angle d_k - \angle y_k)^2 = \left[ \angle d_k - \arctan \frac{\Im(y_k)}{\Re(y_k)} \right]^2 = \left[ \angle d_k - \arctan \frac{\Im(w^H_opt x_k)}{\Re(w^H_opt x_k)} \right]^2 \]

and \( J_{p,\min} \) is achieved for \( w_k = w_{opt} \), to give

\[ J_{p,\min} = J_p(w_{opt}) = \left[ \angle d_k - \arctan \frac{\Im(w^H_{opt} x_k)}{\Re(w^H_{opt} x_k)} \right]^2 = \left[ \angle d_k - \arctan \frac{\Im(\alpha w^H_{opt} x_k)}{\Re(\alpha w^H_{opt} x_k)} \right]^2 \]

Therefore \( J_p(w_{opt}) = J_p(\alpha w_{opt}) \), for any \( \alpha \) and there is no unique minimum.
Decomposing the squared error cost function

Decompose the well-known squared error cost function:

\[ J_{LMS} = |e(k)|^2 = |d(k) - y(k)|^2 \]

\[ = |d(k)|^2 + |y(k)|^2 - [d(k)y^*(k) + d(k)y^*(k)] \]

\[ = (|d(k)| - |y(k)|)^2 + 2|d(k)||y(k)| - [d(k)y^*(k) + d(k)y^*(k)] \]

\[ = (|d(k)| - |y(k)|)^2 + 2|d(k)||y(k)| [1 - \cos(\theta_d - \theta_y)] \]

The Least Mean-Magnitude Phase (LMMP) algorithm can be derived as

\[ w(k+1) = w(k) - \mu_m \nabla w^* J_{Mag} - \mu_p \nabla w^* J_{Phase} \]

\[ = w(k) + \mu_m (|d(k)| - |y(k)|) \frac{y_k}{|y_k|} x_k^* + \mu_p \left( d(k) - |d(k)| \frac{y_k}{|y_k|} \right) x_k^* \]

If \( \mu_p = \mu_m \), the LMMP simplifies into the CLMS!
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 $v[n]$. 

$$\frac{1}{H_c(f)} = 1$$
Example: Digital communications ➔ continued

Input signal constellation

Re{s(n)}
Im{s(n)}

Input signal constellation

Impulse Response

Transmission channel

Impulse Response

Amplitude

Transmission channel

Received signal x[n]

Re{x[n]}
Im{x[n]}

Received signal x[n]

Equalized signal y[n]

Re{y[n]}
Im{y[n]}

Equalized signal y[n]
Beamforming example

- Beamforming Model: ULA, $\lambda/2$ spacing
- Sources: BPSK, BPSK, QPSK, QPSK
- Angle of Arrival: $-45^\circ, 8^\circ, -13^\circ, 30^\circ$
- Number of Antenna Elements: 3
- Algorithms: 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
Output signal constellations

CCLMS

\[ \text{NMSE} = -10.1832 \text{dB} \]

ACLMS

\[ \text{NMSE} = -20.477 \text{dB} \]

ACLMS has lower MSE because of non-circular binary sources
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

- Dual character of load/supply $\leadsto f^+$ 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.

\[
v_a(k) = V_a(k) \cos(\omega k \Delta T + \phi)
\]

\[
v_b(k) = V_b(k) \cos(\omega k \Delta T + \phi - \frac{2\pi}{3})
\]

\[
v_c(k) = V_c(k) \cos(\omega k \Delta T + \phi + \frac{2\pi}{3})
\]

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

\[
\begin{bmatrix}
v_0 \\
v_\alpha(k) \\
v_\beta(k)
\end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix}
\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\
1 & -\frac{1}{2} & -\frac{1}{2} \\
0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2}
\end{bmatrix} \begin{bmatrix}
v_a(k) \\
v_b(k) \\
v_c(k)
\end{bmatrix}
\]

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

\[
v(k) = v_\alpha(k) + jv_\beta(k) = A(k) e^{j(\omega k \Delta T + \phi)} + B(k) e^{-j(\omega k \Delta T + \phi)}
\]

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

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 → a degree of noncircularity

Some types of voltage sags
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
Using the widely linear model for frequency estimation

The widely linear model is able to estimate the frequency for both circular (balanced) and noncircular (unbalanced) voltages.
Introduction to Distributed Adaptive Filters

Multiple adaptive filters collaborating in a network

The \textit{Diffusion} ACLMS at node $i$

Input: \( z_i(k) = [x_i^T(k), x_i^H(k)]^T \),

Weights: \( w_i(k) = [h_i^T(k), g_i^T(k)]^T \)

\[
\begin{align*}
y_i(k) &= w_i^H(k)z_i(k) \\
e_i(k) &= d_i(k) - y_i(k) \\
\psi_i(k + 1) &= w_i(k) + \mu_i e_i^*(k)z_i(k)
\end{align*}
\]

\[
w_i(k + 1) = \sum_{\ell=1}^{N} a_{\ell i} \psi_\ell(k + 1)
\]

The weighting coefficients satisfy

\[
\sum_{\ell=1}^{N} a_{\ell i} = 1 \implies a_{ii} = 1 - \sum_{\ell \neq i} a_{\ell i}
\]
Performance of the Diffusion-CLMS and Diffusion-ACLMS in the smart grid

Frequency estimate at a randomly selected node with multiple frequency events between 0.1 s and 0.5 s

The average MSE of the frequency estimate of D-ACLMS and D-CLMS under a Type D sag.
Conclusions

○ Adaptive processing of noncircular complex signals

○ These arise e.g. due to the nonlinearity of transivevers (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 modelling


- 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 nonlinear IIR filters
- Adaptive stepsizes, dynamical range reduction, collaborative adaptive filters, statistical tests for the validity of complex representations
Some back-up material
Appendix: Transform domain signal processing

Usually complex valued (e.g. Fourier Domain)

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.

\[ x[n] = a_{-0.8} x[n-1] + w[n] \]

\[ x[n] = -0.8 x[n-1] + w[n] \]

\[ a < 0 \rightarrow \text{highpass}, \ a > 0 \rightarrow \text{lowpass} \]

**Highpass signal:** fast changing in time \(\leftrightarrow\) however, smooth spectrum
Appendix: The CLMS algorithm

Step-by-step derivation

Consider a complex valued FIR filter.

The weight update equation for a real valued filter is

\[ w(k + 1) = w(k) + \eta e(k)x(k) \]

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

Hence

\[ e(k) = e_r(k) + je_i(k) \]
\[ d(k) = d_r(k) + jd_i(k) \]
\[ x(k) = x_r(k) + jx_i(k) \]
\[ w(k) = w_r(k) + jw_i(k) \]
\[ y(k) = x^T(k)w(k) \]
Appendix: 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

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

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 for the original LMS, except that the rules of complex algebra must be observed.

Notice that $(x^T(k)w(k))^* = (x^*(k))^T w^*(k)$, e.g.

$$x \times w = [(x_r + jx_i)(w_r + jw_i)] = [x_r w_r - x_i w_i + j(x_r w_i + x_i w_r)]$$

After conjugation we have $[x_r w_r - x_i w_i - j(x_r w_i + x_i w_r)]$. On the other hand

$$x^* w^* = [(x_r - jx_i)(w_r - jw_i)] = [x_r w_r - x_i w_i - j(x_r w_i + x_i w_r)]$$

which is the same as when we conjugate the whole expression.
Appendix: The CLMS Derivation

\[ e^*(k) = d^*(k) - (x^T(k))^* w^*(k) \]

Therefore, for a GD adaptation we have

\[ w(k + 1) = w(k) - \frac{1}{2} \eta \nabla_w (e(k)e^*(k)) \]

where

\[ \nabla(e(k)e^*(k)) = \nabla_r(e(k)e^*(k)) + j \nabla_i(e(k)e^*(k)). \]

Now,

\[ \nabla_r(e(k)e^*(k)) = \left[ \begin{array}{c} \frac{\partial (e(k)e^*(k))}{\partial w_{1r}} \\ \frac{\partial (e(k)e^*(k))}{\partial w_{2r}} \\ \vdots \\ \frac{\partial (e(k)e^*(k))}{\partial w_{Nr}} \end{array} \right] = e(k) \nabla_r(e^*(k)) + e^*(k) \nabla_r(e(k)) \]
Appendix: The CLMS derivation – contd.

Notice that (in a simplified way)

\[ e(k) = d(k) - x(k)w(k) = d(k) - \]

\[ - [x_r(k)w_r(k) - x_i(k)w_i(k) + j(x_r(k)w_i(k) + x_i(k)w_r(k))] \]

\[ e^*(k) = d(k) - [x_r(k)w_r(k) + x_i(k)w_i(k) - j(x_r(k)w_i(k) + x_i(k)w_r(k))] \]

The partial derivatives wrt to \(w_r\) and \(w_i\) are

\[
\frac{\partial e(k)}{\partial w_r(k)} = -[x_r(k) + jx_i(k)] = -x(k)
\]

\[
\frac{\partial e(k)}{\partial w_i(k)} = -[-x_i(k) + jx_r(k)] = -jx(k)
\]

\[
\frac{\partial e^*(k)}{\partial w_r(k)} = -[x_r(k) - jx_i(k)] = -x^*(k)
\]

\[
\frac{\partial e^*(k)}{\partial w_i(k)} = -[x_i(k) - jx_r(k)] = -[-jx^*(k)]
\]
Appendix: The CLMS Derivation – Complex Gradients

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

\[ \nabla_r(e(k)e^*(k)) = e(k)\nabla_r(e^*(k)) + e^*(k)\nabla_r(e(k)) \]
\[ = e(k)(-x^*(k)) + e^*(k)(-x(k)) \]
\[ \nabla_i(e(k)e^*(k)) = e(k)\nabla_i(e^*(k)) + e^*(k)\nabla_i(e(k)) \]
\[ = e(k)(jx^*(k)) + e^*(k)(-jx(k)) \]

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

\[ w_r(k+1) = w_r(k) - \frac{1}{2}\eta\nabla_r(e(k)e^*(k)) \]
\[ w_i(k+1) = w_i(k) - \frac{1}{2}\eta\nabla_i(e(k)e^*(k)) \]
Appendix: The CLMS Derivation – Filter Update

Having in mind that \( w(k + 1) = w_r(k + 1) + jw_i(k + 1) \), we have

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

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

\[
\nabla_r (e(k)e^*(k)) + j\nabla_i (e(k)e^*(k)) = e(k)(-x^*(k)) + e^*(k)(-x(k)) + j\left[ e(k)(jx^*(k)) + e^*(k)(-jx(k)) \right] = -e_r x^* - je_i x^* - e_r x + je_i x - e^*_r x - je^*_i x + e_r x - je_i x = -2e_r x^* - 2je_i x^* = -2e(k)x^*(k)
\]

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

\[
w(k + 1) = w(k) + \eta e(k)x^*(k)
\]
Notes

○
Notes
Notes