The Augmented Complex Gaussian Kernel LMS Algorithm

P. Bouboulis[1] S. Theodoridis[1] M Mavroforakis[2]

[1]Department of Informatics and Telecommunications
University of Athens
Athens, Greece
[2]Department of Computer Science
Computational Biomedicine Lab
University of Houston
Texas, U.S.A.

29-09-2011

Outline

Outline

Processing in RKHS

Processing in Reproducing Kernel Hilbert Spaces is gaining in popularity within the Machine Learning and Signal Processing Communities:

Processing in RKHS

Processing in Reproducing Kernel Hilbert Spaces is gaining in popularity within the Machine Learning and Signal Processing Communities:

Basic Steps:

Processing in RKHS

Processing in Reproducing Kernel Hilbert Spaces is gaining in popularity within the Machine Learning and Signal Processing Communities:

Basic Steps:

1. Map the finite dimensionality input data from the input space F into a higher dimensionality RKHS $\mathcal{H}$.

Processing in RKHS

Processing in Reproducing Kernel Hilbert Spaces is gaining in popularity within the Machine Learning and Signal Processing Communities:

Basic Steps:

1. Map the finite dimensionality input data from the input space F into a higher dimensionality RKHS $\mathcal{H}$.
2. Perform a linear processing (e.g., adaptive filtering) on the mapped data in $\mathcal{H}$.

Processing in RKHS

Processing in Reproducing Kernel Hilbert Spaces is gaining in popularity within the Machine Learning and Signal Processing Communities:

Basic Steps:

1. Map the finite dimensionality input data from the input space F into a higher dimensionality RKHS $\mathcal{H}$.
2. Perform a linear processing (e.g., adaptive filtering) on the mapped data in $\mathcal{H}$.

This procedure is equivalent with a non linear processing in F.

Reproducing Kernel Hilbert Spaces.

Consider a linear class $\mathcal{H}$ of real (complex) valued functions f defined on a set X (in particular $\mathcal{H}$ is a Hilbert space), for which there exists a function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}(\mathbb{C})$ with the following two properties:

Reproducing Kernel Hilbert Spaces.

Consider a linear class $\mathcal{H}$ of real (complex) valued functions f defined on a set X (in particular $\mathcal{H}$ is a Hilbert space), for which there exists a function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}(\mathbb{C})$ with the following two properties:

1. For every $x \in \mathcal{X}$, $\kappa(x, \cdot)$ belongs to $\mathcal{H}$.

Reproducing Kernel Hilbert Spaces.

Consider a linear class $\mathcal{H}$ of real (complex) valued functions f defined on a set X (in particular $\mathcal{H}$ is a Hilbert space), for which there exists a function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}(\mathbb{C})$ with the following two properties:

1. For every $x \in \mathcal{X}$, $\kappa(x, \cdot)$ belongs to $\mathcal{H}$.
2. κ has the so called reproducing property, i.e.,

$$f(x) = \langle f, \kappa(x, \cdot)\rangle_{\mathcal{H}}, \text{ for all } f \in \mathcal{H}, \ x \in \mathcal{X}. \tag{1}$$

Kernel Trick

In particular, If

Kernel Trick

In particular, If
$\mathcal{X} \ni x \to \Phi(x) := \kappa(x, \cdot) \in \mathcal{H}$
$\mathcal{X} \ni y \to \Phi(y) := \kappa(y, \cdot) \in \mathcal{H}$,
then the inner product in $\mathcal{H}$ is given as a function computed on $\mathcal{X}$:

$$\boxed{\kappa(x, y) = \langle \kappa(x, \cdot), \kappa(y, \cdot) \rangle_{\mathcal{H}}} \quad \text{kernel trick}$$

Advantages

Advantages of kernel-based signal processing:

Advantages

Advantages of kernel-based signal processing:

- The original nonlinear task is transformed into a linear one.

Advantages

Advantages of kernel-based signal processing:

- The original nonlinear task is transformed into a linear one.
- Different types of nonlinearities can be treated in a unified way.

Developing Algorithms in RKHS

- The black box approach.

Developing Algorithms in RKHS

- The black box approach.
  - Develop the Algorithm in $\mathcal{X}$.

Developing Algorithms in RKHS

- The black box approach.
  - Develop the Algorithm in $\mathcal{X}$.
  - Express it, **if possible**, in inner products.

Developing Algorithms in RKHS

- The black box approach.
  - Develop the Algorithm in $\mathcal{X}$.
  - Express it, **if possible**, in inner products.
  - Replace inner products with kernel evaluations according to the kernel trick.

Developing Algorithms in RKHS

- The black box approach.
  - Develop the Algorithm in $\mathcal{X}$.
  - Express it, **if possible**, in inner products.
  - Replace inner products with kernel evaluations according to the kernel trick.
- Work directly in the RKHS, assuming that the data have been mapped and live in the RKHS $\mathcal{H}$, i.e.,

$$\mathcal{X} \ni \boldsymbol{x} \to \Phi(\boldsymbol{x}) := \kappa(\boldsymbol{x}, \cdot) \in \mathcal{H}.$$

Outline

The complex domain

- Complex-valued signals arise frequently in applications as diverse as communications, biomedicine, radar, e.t.c.

The complex domain

- Complex-valued signals arise frequently in applications as diverse as communications, biomedicine, radar, e.t.c.
- The complex domain not only provides a convenient and elegant representation for these signals, but also a natural way to preserve their characteristics and to handle transformations that need to be performed.

The complex domain

- Complex-valued signals arise frequently in applications as diverse as communications, biomedicine, radar, e.t.c.
- The complex domain not only provides a convenient and elegant representation for these signals, but also a natural way to preserve their characteristics and to handle transformations that need to be performed.
- In the more traditional setting, one usually assumes that the signal is *circular*.

Circularity

- Circularity is intimately related to the rotation in the geometric sense.

Circularity

- Circularity is intimately related to the rotation in the geometric sense.
- A complex random variable is$Z$ is called circular, if for any angle ϕ both Z and $Ze^{i\phi}$ (i.e., the rotation of Z by angle ϕ) follow the same probability distribution.

Circularity

- Circularity is intimately related to the rotation in the geometric sense.
- A complex random variable is$Z$ is called circular, if for any angle ϕ both Z and $Ze^{i\phi}$ (i.e., the rotation of Z by angle ϕ) follow the same probability distribution.
- Naturally this assumption limits the area for applications, as many practical signals exhibit non-circular characteristics.

Circularity

- Circularity is intimately related to the rotation in the geometric sense.
- A complex random variable is$Z$ is called circular, if for any angle ϕ both Z and $Ze^{i\phi}$ (i.e., the rotation of Z by angle ϕ) follow the same probability distribution.
- Naturally this assumption limits the area for applications, as many practical signals exhibit non-circular characteristics.
- Widely linear filters are able to efficiently treat such signals, as they capture the full second order statistics of any given complex-valued data sequence.

Outline

Complex RKHS

- Although the theory of RKHS holds for complex spaces too, most of the kernel-based techniques were designed to process real data only.

Complex RKHS

- Although the theory of RKHS holds for complex spaces too, most of the kernel-based techniques were designed to process real data only.
- Moreover, in the related literature the complex kernel functions have been ignored.

Complex RKHS

- Although the theory of RKHS holds for complex spaces too, most of the kernel-based techniques were designed to process real data only.
- Moreover, in the related literature the complex kernel functions have been ignored.
- Recently, however, a unified kernel-based framework, which is able to treat complex signals, has been presented.

Complex RKHS

- Although the theory of RKHS holds for complex spaces too, most of the kernel-based techniques were designed to process real data only.
- Moreover, in the related literature the complex kernel functions have been ignored.
- Recently, however, a unified kernel-based framework, which is able to treat complex signals, has been presented.
- This machinery transforms the input data into complex RKHS and employs the Wirtinger's Calculus to derive the respective gradients.

Feature Map

- In the case of the pure complex kernels, we map directly the data to the complex RKHS using the corresponding feature map, i.e.,

$$\boldsymbol{\Phi}(\boldsymbol{z}) = \kappa_{\mathbb{C}}(\cdot, \boldsymbol{z}).$$

Outline

1. Introduction
   - Reproducing Kernel Hilbert Spaces
   - Complex valued signals
   - Complex RKHS
   - The Problem
2. Widely Linear Estimation Filters
   - Definition
   - Why widely-linear?
3. Linear and Augmented Complex Kernel LMS Algorithms
   - Complex Kernel LMS (CKLMS)
   - Augmented Complex Kernel LMS (ACKLMS)
4. Experiments
   - Soft non linear channel
   - Hard non linear channel
5. Future Research

The problem

The major tasks of this research:

The problem

The major tasks of this research:

1. Development of widely-linear kernel-based adaptive filters

The problem

The major tasks of this research:

1. Development of widely-linear kernel-based adaptive filters
2. Determining why widely-linear estimation is better than traditional linear estimation.

Outline

1. Introduction
   - Reproducing Kernel Hilbert Spaces
   - Complex valued signals
   - Complex RKHS
   - The Problem
2. Widely Linear Estimation Filters
   - Definition
   - Why widely-linear?
3. Linear and Augmented Complex Kernel LMS Algorithms
   - Complex Kernel LMS (CKLMS)
   - Augmented Complex Kernel LMS (ACKLMS)
4. Experiments
   - Soft non linear channel
   - Hard non linear channel
5. Future Research

Definition

- Consider the complex input/output data $(\boldsymbol{z}(n), y(n))$, $n = 1, 2, \ldots, N$.

Definition

- Consider the complex input/output data $(\boldsymbol{z}(n), y(n))$, $n = 1, 2, \ldots, N$.
- In A typical linear complex filter the output at time instance n is estimated as

$$\boxed{\hat{d}(n) = \boldsymbol{w}^H \boldsymbol{z}(n).}$$

Definition

- Consider the complex input/output data $(\boldsymbol{z}(n), y(n))$, $n = 1, 2, \ldots, N$.
- In A typical linear complex filter the output at time instance n is estimated as

$$\hat{d}(n) = \boldsymbol{w}^H \boldsymbol{z}(n).$$

- In widely-linear filters, we estimate the filter's output as:

$$\tilde{d}(n) = \boldsymbol{w}^H \boldsymbol{z}(n) + \boldsymbol{v}^H \boldsymbol{z}^*(n).$$

Outline

Equivalent notations

If we set $\boldsymbol{w} = \boldsymbol{w}_r + i\boldsymbol{w}_i$, $\boldsymbol{v} = \boldsymbol{v}_r + i\boldsymbol{v}_i$ and $\boldsymbol{z}(n) = \boldsymbol{x}(n) + i\boldsymbol{y}(n)$ we take that:

Equivalent notations

If we set $\boldsymbol{w} = \boldsymbol{w}_r + i\boldsymbol{w}_i$, $\boldsymbol{v} = \boldsymbol{v}_r + i\boldsymbol{v}_i$ and $\boldsymbol{z}(n) = \boldsymbol{x}(n) + i\boldsymbol{y}(n)$ we take that:

$$\hat{d}(n) = \boldsymbol{w}_r^T\boldsymbol{x}(n) + \boldsymbol{w}_i^T\boldsymbol{y}(n) + i\left(\boldsymbol{w}_r^T\boldsymbol{y}(n) - \boldsymbol{w}_i^T\boldsymbol{x}(n)\right)$$

Equivalent notations

If we set $\boldsymbol{w} = \boldsymbol{w}_r + i\boldsymbol{w}_i$, $\boldsymbol{v} = \boldsymbol{v}_r + i\boldsymbol{v}_i$ and $\boldsymbol{z}(n) = \boldsymbol{x}(n) + i\boldsymbol{y}(n)$ we take that:

$$\hat{d}(n) = \boldsymbol{w}_r^T\boldsymbol{x}(n) + \boldsymbol{w}_i^T\boldsymbol{y}(n) + i\left(\boldsymbol{w}_r^T\boldsymbol{y}(n) - \boldsymbol{w}_i^T\boldsymbol{x}(n)\right)$$

$$\begin{aligned}\tilde{d}(n) = &\left((\boldsymbol{w}_r^T + \boldsymbol{v}_r^T)\boldsymbol{x} + (\boldsymbol{w}_i^T - \boldsymbol{v}_i^T)\boldsymbol{y}\right)\\ &+ i\left(-(\boldsymbol{w}_i^T + \boldsymbol{v}_i^T)\boldsymbol{x} + (\boldsymbol{w}_r^T - \boldsymbol{v}_r^T)\boldsymbol{y})\right).\end{aligned}$$

The DRC approach

The real essence behind a linear complex filter operation:

The DRC approach

The real essence behind a linear complex filter operation:

Given two real input vectors $\boldsymbol{x}(n)$ and $\boldsymbol{y}(n)$ compute linear filters in order to estimate two real processes, $d_r(n)$ and $d_i(n)$, in an optimal way that jointly cares for both $d_r(n)$ and $d_i(n)$

The DRC approach

We express the problem in its multichannel formulation:

$$\begin{pmatrix} \bar{d}_r(n) \\ \bar{d}_i(n) \end{pmatrix} = \begin{pmatrix} \boldsymbol{u}_{1,1}^T & \boldsymbol{u}_{1,2}^T \\ \boldsymbol{u}_{2,1}^T & \boldsymbol{u}_{2,2}^T \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{x}(n) \\ \boldsymbol{y}(n) \end{pmatrix} \equiv \boldsymbol{U} \cdot \begin{pmatrix} \boldsymbol{x}(n) \\ \boldsymbol{y}(n) \end{pmatrix}.$$

The DRC approach

We express the problem in its multichannel formulation:

$$\begin{pmatrix} \bar{d}_r(n) \\ \bar{d}_i(n) \end{pmatrix} = \begin{pmatrix} \boldsymbol{u}_{1,1}^T & \boldsymbol{u}_{1,2}^T \\ \boldsymbol{u}_{2,1}^T & \boldsymbol{u}_{2,2}^T \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{x}(n) \\ \boldsymbol{y}(n) \end{pmatrix} \equiv \boldsymbol{U} \cdot \begin{pmatrix} \boldsymbol{x}(n) \\ \boldsymbol{y}(n) \end{pmatrix}.$$

This is the Dual Real Channel (DRC) formulation of the problem.

The DRC approach

We express the problem in its multichannel formulation:

$$\begin{pmatrix} \bar{d}_r(n) \\ \bar{d}_i(n) \end{pmatrix} = \begin{pmatrix} \boldsymbol{u}_{1,1}^T & \boldsymbol{u}_{1,2}^T \\ \boldsymbol{u}_{2,1}^T & \boldsymbol{u}_{2,2}^T \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{x}(n) \\ \boldsymbol{y}(n) \end{pmatrix} \equiv \boldsymbol{U} \cdot \begin{pmatrix} \boldsymbol{x}(n) \\ \boldsymbol{y}(n) \end{pmatrix}.$$

This is the Dual Real Channel (DRC) formulation of the problem.
In complex notation the DRC formulation gives:

The DRC approach

We express the problem in its multichannel formulation:

$$\begin{pmatrix} \bar{d}_r(n) \\ \bar{d}_i(n) \end{pmatrix} = \begin{pmatrix} \boldsymbol{u}_{1,1}^T & \boldsymbol{u}_{1,2}^T \\ \boldsymbol{u}_{2,1}^T & \boldsymbol{u}_{2,2}^T \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{x}(n) \\ \boldsymbol{y}(n) \end{pmatrix} \equiv \boldsymbol{U} \cdot \begin{pmatrix} \boldsymbol{x}(n) \\ \boldsymbol{y}(n) \end{pmatrix}.$$

This is the Dual Real Channel (DRC) formulation of the problem.
In complex notation the DRC formulation gives:

$$\bar{d}(n) = (\boldsymbol{u}_{1,1}^T \boldsymbol{x} + \boldsymbol{u}_{1,2}^T \boldsymbol{y}) + i(\boldsymbol{u}_{2,1}^T \boldsymbol{x} + \boldsymbol{u}_{2,2}^T \boldsymbol{y}).$$

Different Formulations

Let's Summarize:

Linear filter: $$\hat{d}(n) = \boldsymbol{w}_r^T \boldsymbol{x}(n) + \boldsymbol{w}_i^T \boldsymbol{y}(n) + i\left(\boldsymbol{w}_r^T \boldsymbol{y}(n) - \boldsymbol{w}_i^T \boldsymbol{x}(n)\right).$$

Widely-linear filter: $$\tilde{d}(n) = \left((\boldsymbol{w}_r^T + \boldsymbol{v}_r^T)\boldsymbol{x} + (\boldsymbol{w}_i^T - \boldsymbol{v}_i^T)\boldsymbol{y}\right) + i\left(-(\boldsymbol{w}_i^T + \boldsymbol{v}_i^T)\boldsymbol{x} + (\boldsymbol{w}_r^T - \boldsymbol{v}_r^T)\boldsymbol{y}\right).$$

DRC filter: $$\bar{d}(n) = (\boldsymbol{u}_{1,1}^T \boldsymbol{x} + \boldsymbol{u}_{1,2}^T \boldsymbol{y}) + i(\boldsymbol{u}_{2,1}^T \boldsymbol{x} + \boldsymbol{u}_{2,2}^T \boldsymbol{y}).$$

Remarks

1. It easy to see that the DRC approach (which is what we wanted to emulate) adopts a richer representation than that of the traditional linear filter.

Remarks

1. It easy to see that the DRC approach (which is what we wanted to emulate) adopts a richer representation than that of the traditional linear filter.
2. The same is true for the Widely-linear filter.

Remarks

1. It easy to see that the DRC approach (which is what we wanted to emulate) adopts a richer representation than that of the traditional linear filter.
2. The same is true for the Widely-linear filter.
3. It is not difficult to prove that the DRC and the widely-linear estimation functions are equivalent.

A useful Proposition

In general we can prove the following.

Proposition

Consider a real Hilbert space $\mathcal{H}$, the real Hilbert space $\mathcal{H}^2$ and the complex Hilbert space $\mathbb{H} = \mathcal{H} + i\mathcal{H}$. Then any linear function $\boldsymbol{T} : \mathcal{H}^2 \to \mathbb{R}^2$ can be expressed in complex notation as

$$\boldsymbol{T}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{T}(\boldsymbol{x} + i\boldsymbol{y}) = \boldsymbol{T}(\boldsymbol{z}) = \langle \boldsymbol{z}, \boldsymbol{w} \rangle_{\mathbb{H}} + \langle \boldsymbol{z}^*, \boldsymbol{v} \rangle_{\mathbb{H}},$$

for some $\boldsymbol{w}, \boldsymbol{v} \in \mathbb{H}$, where $\langle \cdot, \cdot, \rangle_{\mathbb{H}}$ is the respective inner product of $\mathbb{H}$.

Remarks

In view of the proposition, one understands that the original formulation of the complex linear filters (such as the complex LMS) was rather "unorthodox", as it excludes a large class of linear functions from being considered in the estimation process.

Remarks

In view of the proposition, one understands that the original formulation of the complex linear filters (such as the complex LMS) was rather “unorthodox”, as it excludes a large class of linear functions from being considered in the estimation process.

The linearity with respect to the field of complex numbers is restricted, compared to the linearity that underlies the DRC approach, which is more natural. Thus, the correct complex linear estimation is $\boldsymbol{T}(\boldsymbol{z}) = \langle \boldsymbol{z}, \boldsymbol{w} \rangle_{\mathbb{H}} + \langle \boldsymbol{z}^*, \boldsymbol{v} \rangle_{\mathbb{H}}$ rather than $\boldsymbol{T}(\boldsymbol{z}) = \langle \boldsymbol{z}, \boldsymbol{w} \rangle_{\mathbb{H}}$.

Outline

Complex Kernel LMS (CKLMS)

Consider the sequence of examples

$$(\boldsymbol{z}(1), d(1)), (\boldsymbol{z}(2), d(2)), \dots, (\boldsymbol{z}(N), d(N)),$$

where $d(n) \in \mathbb{C}$, $\boldsymbol{z}(n) \in \mathbb{C}^{\nu}$,
$\boldsymbol{z}(n) = \boldsymbol{x}(n) + i\boldsymbol{y}(n)$,
$\boldsymbol{x}(n), \boldsymbol{y}(n) \in \mathbb{R}^{\nu}$, for $n = 1, \dots, N$.

Complex Kernel LMS (CKLMS)

Consider the sequence of examples

$$(\boldsymbol{z}(1), d(1)), (\boldsymbol{z}(2), d(2)), \ldots, (\boldsymbol{z}(N), d(N)),$$

where $d(n) \in \mathbb{C}$, $\boldsymbol{z}(n) \in \mathbb{C}^\nu$,
$\boldsymbol{z}(n) = \boldsymbol{x}(n) + i\boldsymbol{y}(n)$,
$\boldsymbol{x}(n), \boldsymbol{y}(n) \in \mathbb{R}^\nu$, for $n = 1, \ldots, N$.

Consider, also, a real RKHS $\mathcal{H}$, the real Hilbert space $\mathcal{H}^2$ and the complex Hilbert space $\mathbb{H} = \mathcal{H} + i\mathcal{H}$.

Complex Kernel LMS (CKLMS)

We model the linear filter as:

$$\hat{d}(n) = \langle \boldsymbol{\Phi}(\boldsymbol{z}(n)), \boldsymbol{w} \rangle.$$

Complex Kernel LMS (CKLMS)

We model the linear filter as:

$$\hat{d}(n) = \langle \boldsymbol{\Phi}(\boldsymbol{z}(n)), \boldsymbol{w} \rangle.$$

The objective of the CKLMS is to estimate $\boldsymbol{w}$, so that to minimize $E(\mathcal{L}_n(\boldsymbol{w}))$, at each time instance n, where

$$\mathcal{L}_n(\boldsymbol{w}) = |e(n)|^2 = \left| d(n) - \hat{d}(n) \right|^2.$$

Complex Kernel LMS (CKLMS)

Applying the rules of Wirtinger's Calculus in complex RKHS, we can easily deduce that the gradient step update of the CKLMS is

$$\boldsymbol{w}(n) = \boldsymbol{w}(n-1) + \mu \boldsymbol{\Phi}(\boldsymbol{z}(n)) e^*(n).$$

Complex Kernel LMS (CKLMS)

Applying the rules of Wirtinger's Calculus in complex RKHS, we can easily deduce that the gradient step update of the CKLMS is

$$\boldsymbol{w}(n) = \boldsymbol{w}(n-1) + \mu \boldsymbol{\Phi}(\boldsymbol{z}(n)) e^*(n).$$

Assuming that $\boldsymbol{w}(0) = 0$, the repeated application of the weight-update equations gives:

$$\boldsymbol{w}(n-1) = \mu \sum_{k=1}^{n-1} \boldsymbol{\Phi}(\boldsymbol{z}(n)) e^*(n)$$

and

$$\hat{d}(n) = \mu \sum_{k=1}^{n-1} e(k) \langle \boldsymbol{\Phi}(\boldsymbol{z}(n)), \boldsymbol{\Phi}(\boldsymbol{z}(k)) \rangle_{\mathbb{H}}$$

Outline

Augmented Complex Kernel LMS (ACKLMS)

We model the widely-linear (augmented) filter as:

$$\tilde{d}(n) = \langle \boldsymbol{\Phi}(\boldsymbol{z}(n)), \boldsymbol{w} \rangle + \langle \boldsymbol{\Phi}^*(\boldsymbol{z}(n)), \boldsymbol{v} \rangle.$$

Augmented Complex Kernel LMS (ACKLMS)

We model the widely-linear (augmented) filter as:

$$\tilde{d}(n) = \langle \mathbf{\Phi}(\boldsymbol{z}(n)), \boldsymbol{w} \rangle + \langle \mathbf{\Phi}^*(\boldsymbol{z}(n)), \boldsymbol{v} \rangle.$$

The objective of the CKLMS is to estimate $\boldsymbol{w}$ and $\boldsymbol{v}$, so that to minimize $E(\mathcal{L}_n(\boldsymbol{w}))$, at each time instance n, where

$$\mathcal{L}_n(\boldsymbol{w}) = |e(n)|^2 = \left| d(n) - \hat{d}(n) \right|^2.$$

Augmented Complex Kernel LMS (ACKLMS)

Applying the rules of Wirtinger's Calculus in complex RKHS, we can easily deduce that the gradient step updates of the CKLMS are

$$\boldsymbol{w}(n) = \boldsymbol{w}(n-1) + \mu \boldsymbol{\Phi}(\boldsymbol{z}(n)) e^*(n).$$

$$\boldsymbol{v}(n) = \boldsymbol{v}(n-1) + \mu \boldsymbol{\Phi}^*(\boldsymbol{z}(n)) e^*(n).$$

Augmented Complex Kernel LMS (ACKLMS)

Assuming that $\boldsymbol{w}(0) = 0$ and $\boldsymbol{v}(0) = 0$, the repeated application of the weight-update equations gives:

$$\boldsymbol{w}(n-1) = \mu \sum_{k=1}^{n-1} \boldsymbol{\Phi}(\boldsymbol{z}(n)) e^*(n),$$

and

$$\boldsymbol{v}(n-1) = \mu \sum_{k=1}^{n-1} \boldsymbol{\Phi}^*(\boldsymbol{z}(n)) e^*(n),$$

Augmented Complex Kernel LMS (ACKLMS)

The filter's output becomes:

$$\begin{aligned}\tilde{d}(n) =& \mu \sum_{k=1}^{n-1} e(k) \langle \boldsymbol{\Phi}(\boldsymbol{z}(n)), \boldsymbol{\Phi}(\boldsymbol{z}(k)) \rangle_{\mathbb{H}} \\ &+ \mu \sum_{k=1}^{n-1} e(k) \langle \boldsymbol{\Phi}^*(\boldsymbol{z}(n)), \boldsymbol{\Phi}^*(\boldsymbol{z}(k)) \rangle_{\mathbb{H}}.\end{aligned}$$

Augmented Complex Kernel LMS (ACKLMS)

The filter's output becomes:

$$\begin{aligned}\tilde{d}(n) =&\mu\sum_{k=1}^{n-1} e(k)\langle\boldsymbol{\Phi}(\boldsymbol{z}(n)),\boldsymbol{\Phi}(\boldsymbol{z}(k))\rangle_{\mathbb{H}}\\ &+\mu\sum_{k=1}^{n-1} e(k)\langle\boldsymbol{\Phi}^*(\boldsymbol{z}(n)),\boldsymbol{\Phi}^*(\boldsymbol{z}(k))\rangle_{\mathbb{H}}.\end{aligned}$$

If we replace the inner products with the respective complex kernel function we take:

$$\tilde{d}(n) =\mu\sum_{k=1}^{n-1} e(k)\kappa_{\mathbb{C}}(\boldsymbol{z}(n),\boldsymbol{z}(k)) + \mu\sum_{k=1}^{n-1} e(k)\kappa_{\mathbb{C}}^*(\boldsymbol{z}(n),\boldsymbol{z}(k)).$$

Outline

Equalization

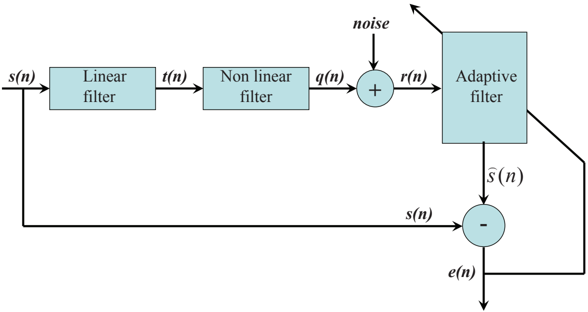


Figure: The equalization problem.

Soft non linear channel

- $t(n) = (-0.9 + 0.8i) \cdot s(n) + (0.6 - 0.7i) \cdot s(n-1)$

Soft non linear channel

- $t(n) = (-0.9 + 0.8i) \cdot s(n) + (0.6 - 0.7i) \cdot s(n-1)$
- $q(n) = t(n) + (0.1 + 0.15i) \cdot t^2(n) + (0.06 + 0.05i) \cdot t^3(n)$

Soft non linear channel

- $t(n) = (-0.9 + 0.8i) \cdot s(n) + (0.6 - 0.7i) \cdot s(n-1)$
- $q(n) = t(n) + (0.1 + 0.15i) \cdot t^2(n) + (0.06 + 0.05i) \cdot t^3(n)$
- $s(n) = 0.70(\sqrt{1-\rho^2}X(n) + i\rho Y(n))$, where $X(n)$ and $Y(n)$ are gaussian random variables.

Soft non linear channel

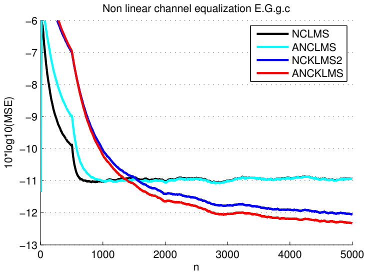
- $t(n) = (-0.9 + 0.8i) \cdot s(n) + (0.6 - 0.7i) \cdot s(n-1)$
- $q(n) = t(n) + (0.1 + 0.15i) \cdot t^2(n) + (0.06 + 0.05i) \cdot t^3(n)$
- $s(n) = 0.70(\sqrt{1-\rho^2}X(n) + i\rho Y(n))$, where $X(n)$ and $Y(n)$ are gaussian random variables.
  1. This input is circular for $\rho = \sqrt{2}/2$

Soft non linear channel

- $t(n) = (-0.9 + 0.8i) \cdot s(n) + (0.6 - 0.7i) \cdot s(n-1)$
- $q(n) = t(n) + (0.1 + 0.15i) \cdot t^2(n) + (0.06 + 0.05i) \cdot t^3(n)$
- $s(n) = 0.70(\sqrt{1-\rho^2}X(n) + i\rho Y(n))$, where $X(n)$ and $Y(n)$ are gaussian random variables.
  1. This input is circular for $\rho = \sqrt{2}/2$
  2. highly non-circular if ρ approaches 0 or 1.

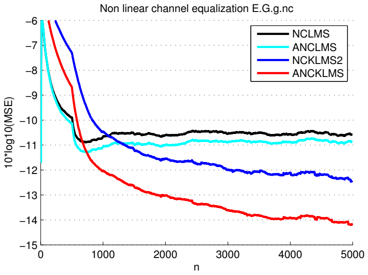
Circular Data

Learning curves for NCLMS, WL-NCLMS, NCKLMS and NACKLMS (filter length $L = 5$, delay $D = 2$) for the soft nonlinear channel equalization, for the **circular** input case.



Non Circular Data

Learning curves for NCLMS, WL-NCLMS, NCKLMS and NACKLMS (filter length $L = 5$, delay $D = 2$) for the soft nonlinear channel equalization, for the **non circular** input case.



Outline

Hard non linear channel

- $t(n) = (-0.9 + 0.8i) \cdot s(n) + (0.6 - 0.7i) \cdot s(n-1) + (-0.4 + 0.3i) \cdot s(n-2) + (0.3 - 0.2i) \cdot s(n-3) + (-0.1i - 0.2i) \cdot s(n-4)$

Hard non linear channel

- $t(n) = (-0.9 + 0.8i) \cdot s(n) + (0.6 - 0.7i) \cdot s(n-1) + (-0.4 + 0.3i) \cdot s(n-2) + (0.3 - 0.2i) \cdot s(n-3) + (-0.1i - 0.2i) \cdot s(n-4)$
- $q(n) = t(n) + (0.2 + 0.25i) \cdot t^2(n) + (0.12 + 0.09i) \cdot t^3(n)$

Hard non linear channel

- $t(n) = (-0.9 + 0.8i) \cdot s(n) + (0.6 - 0.7i) \cdot s(n-1) + (-0.4 + 0.3i) \cdot s(n-2) + (0.3 - 0.2i) \cdot s(n-3) + (-0.1i - 0.2i) \cdot s(n-4)$
- $q(n) = t(n) + (0.2 + 0.25i) \cdot t^2(n) + (0.12 + 0.09i) \cdot t^3(n)$
- $s(n) = 0.70(\sqrt{1-\rho^2}X(n) + i\rho Y(n))$, where $X(n)$ and $Y(n)$ are gaussian random variables.

Hard non linear channel

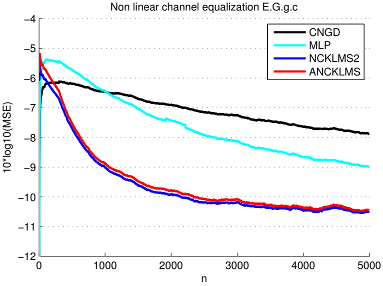
- $t(n) = (-0.9 + 0.8i) \cdot s(n) + (0.6 - 0.7i) \cdot s(n-1) + (-0.4 + 0.3i) \cdot s(n-2) + (0.3 - 0.2i) \cdot s(n-3) + (-0.1i - 0.2i) \cdot s(n-4)$
- $q(n) = t(n) + (0.2 + 0.25i) \cdot t^2(n) + (0.12 + 0.09i) \cdot t^3(n)$
- $s(n) = 0.70(\sqrt{1-\rho^2}X(n) + i\rho Y(n))$, where $X(n)$ and $Y(n)$ are gaussian random variables.
  1. This input is circular for $\rho = \sqrt{2}/2$

Hard non linear channel

- $t(n) = (-0.9 + 0.8i) \cdot s(n) + (0.6 - 0.7i) \cdot s(n-1) + (-0.4 + 0.3i) \cdot s(n-2) + (0.3 - 0.2i) \cdot s(n-3) + (-0.1i - 0.2i) \cdot s(n-4)$
- $q(n) = t(n) + (0.2 + 0.25i) \cdot t^2(n) + (0.12 + 0.09i) \cdot t^3(n)$
- $s(n) = 0.70(\sqrt{1-\rho^2}X(n) + i\rho Y(n))$, where $X(n)$ and $Y(n)$ are gaussian random variables.
    1. This input is circular for $\rho = \sqrt{2}/2$
    2. highly non-circular if ρ approaches 0 or 1.

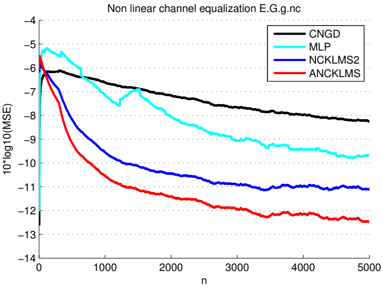
Circular Data

Learning curves for CNGD, MLP (50 nodes), NCKLMS and NACKLMS (filter length $L = 5$, delay $D = 2$) for the hard nonlinear channel equalization, for the **circular** input case.



Non Circular Data

Learning curves for CNGD, MLP (50 nodes), NCKLMS and NACKLMS (filter length $L = 5$, delay $D = 2$) for the hard nonlinear channel equalization, for the **non circular** input case.



Complex Kernel RLS

Development of the sliding-window complex Kernel RLS.

Complex Kernel RLS

Development of the sliding-window complex Kernel RLS.

- Consider the set of samples $\{(\boldsymbol{z}(1), y(1)), \ldots, (\boldsymbol{z}(n), y(n))\}$.

Complex Kernel RLS

Development of the sliding-window complex Kernel RLS.

- Consider the set of samples $\{(\boldsymbol{z}(1), y(1)), \ldots, (\boldsymbol{z}(n), y(n))\}$.
- Let $\tilde{d}(n) = \langle \boldsymbol{\Phi}(\boldsymbol{z}(n)), \boldsymbol{w} \rangle + \langle \boldsymbol{\Phi}^*(\boldsymbol{z}(n)), \boldsymbol{v} \rangle$, $e(n) = \tilde{d}(n) - y(n)$.

Complex Kernel RLS

Development of the sliding-window complex Kernel RLS.

- Consider the set of samples $\{(\boldsymbol{z}(1), y(1)), \ldots, (\boldsymbol{z}(n), y(n))\}$.
- Let $\tilde{d}(n) = \langle \boldsymbol{\Phi}(\boldsymbol{z}(n)), \boldsymbol{w}\rangle + \langle \boldsymbol{\Phi}^*(\boldsymbol{z}(n)), \boldsymbol{v}\rangle$, $e(n) = \tilde{d}(n) - y(n)$.
- The LMS algorithm minimizes $C(w, v) = E[|e(n)|^2]$ at each time instant n.

Complex Kernel RLS

Development of the sliding-window complex Kernel RLS.

- Consider the set of samples $\{(\boldsymbol{z}(1), y(1)), \dots, (\boldsymbol{z}(n), y(n))\}$.
- Let $\tilde{d}(n) = \langle \boldsymbol{\Phi}(\boldsymbol{z}(n)), \boldsymbol{w}\rangle + \langle \boldsymbol{\Phi}^*(\boldsymbol{z}(n)), \boldsymbol{v}\rangle$, $e(n) = \tilde{d}(n) - y(n)$.
- The LMS algorithm minimizes $C(w, v) = E[|e(n)|^2]$ at each time instant n.
- The RLS algorithm minimizes

$$C(w, v) = \sum_{k=n-l+1}^{n} \beta_{k-n+l} |e(k)|^2,$$

for a suitable choice of weights $\beta_1, \dots, \beta_l$.

Complex Kernel RLS

Thinks to do:

Complex Kernel RLS

Thinks to do:

- Compute the Wirtinger gradient and the update scheme.

Complex Kernel RLS

Thinks to do:

- Compute the Wirtinger gradient and the update scheme.
- Develop the algorithm.

Complex Kernel RLS

Thinks to do:

- Compute the Wirtinger gradient and the update scheme.
- Develop the algorithm.
- Implement a sparsification strategy.