Extrinsic Geometrical Methods for Neural Blind Deconvolution

Simone Fiori

fiori@deit.univpm.it

Dipartimento di Elettronica, Intelligenza Artificiale e Telecomunicazioni

Università Politecnica delle Marche

Ancona (Italy, EU)

Talk overview

- Blind deconvolution (BD): Signal model, basic assumptions, Bayesian (‘Bussgang’-type) BD.

Talk overview

- Blind deconvolution (BD): Signal model, basic assumptions, Bayesian (‘Bussgang’-type) BD.
- Automatic gain control (AGC): Source of geometrical structure of the parameter space.

Talk overview

- Blind deconvolution (BD): Signal model, basic assumptions, Bayesian ('Bussgang'-type) BD.
- Automatic gain control (AGC): Source of geometrical structure of the parameter space.
- Algorithms: Geodesic-based and projection-based.

Talk overview

- Blind deconvolution (BD): Signal model, basic assumptions, Bayesian ('Bussgang'-type) BD.
- Automatic gain control (AGC): Source of geometrical structure of the parameter space.
- Algorithms: Geodesic-based and projection-based.
- Numerical experiments and comparison.

BD: Channel Model

Channel output signal model:

$$x_n = \boldsymbol{h}^T \boldsymbol{s}_n + \nu_n \quad ,$$

- $\boldsymbol{s}_n \stackrel{\text{def}}{=} [s_n \ s_{n-1} \ s_{n-2} \ \ldots \ s_{n-L_h+1}]^T$ is the system's input vector-stream at time $n = 1, \ldots, N$,

BD: Channel Model

Channel output signal model:

$$x_n = \boldsymbol{h}^T \boldsymbol{s}_n \ + \ \nu_n \ \ ,$$

- $\boldsymbol{s}_n \stackrel{\text{def}}{=} [s_n \ s_{n-1} \ s_{n-2} \ \ldots \ s_{n-L_h+1}]^T$ is the system's input vector-stream at time $n = 1, \ldots, N$,
- s_n denotes the sampled source signal,

BD: Channel Model

Channel output signal model:

$$x_n = \boldsymbol{h}^T \boldsymbol{s}_n + \nu_n \,,$$

- $\boldsymbol{s}_n \stackrel{\text{def}}{=} [s_n \ s_{n-1} \ s_{n-2} \ \ldots \ s_{n-L_h+1}]^T$ is the system's input vector-stream at time $n = 1, \ldots, N$,
- s_n denotes the sampled source signal,
- ν_n represents a zero-mean white measurement disturbance independent of the source signal,

BD: Channel Model

Channel output signal model:

$$x_n = \boldsymbol{h}^T \boldsymbol{s}_n \; + \; \nu_n \quad ,$$

- $\boldsymbol{s}_n \stackrel{\text{def}}{=} [s_n \; s_{n-1} \; s_{n-2} \; \ldots \; s_{n-L_h+1}]^T$ is the system's input vector-stream at time $n = 1, \ldots, N$,
- s_n denotes the sampled source signal,
- ν_n represents a zero-mean white measurement disturbance independent of the source signal,
- L_h denotes the length of system impulse response $\boldsymbol{h}$.

BD: Filter model

FIR filter output signal mode:

$$z_{m,n} = \boldsymbol{w}_m^T \boldsymbol{x}_n \ ,$$

- $\boldsymbol{w}_m = [w_0 \ w_2 \ w_3 \ \ldots \ w_{L_w-1}]^T$ denotes the filter's impulse response at learning-iteration $m = 1, \ldots, M$,

BD: Filter model

FIR filter output signal mode:

$$z_{m,n} = \boldsymbol{w}_m^T \boldsymbol{x}_n \ ,$$

- $\boldsymbol{w}_m = [w_0 \ w_2 \ w_3 \ \ldots \ w_{L_w-1}]^T$ denotes the filter's impulse response at learning-iteration $m = 1, \ldots, M$,
- $\boldsymbol{x}_n \overset{\text{def}}{=} [x_n \ x_{n-1} \ x_{n-2} \ \ldots \ x_{n-L_w+1}]^T$ denotes the filter input samples at time $n = 1, \ldots, N$,

BD: Filter model

FIR filter output signal mode:

$$z_{m,n} = \boldsymbol{w}_m^T \boldsymbol{x}_n \ ,$$

- $\boldsymbol{w}_m = [w_0 \ w_2 \ w_3 \ \ldots \ w_{L_w-1}]^T$ denotes the filter's impulse response at learning-iteration $m = 1, \ldots, M$,
- $\boldsymbol{x}_n \stackrel{\text{def}}{=} [x_n \ x_{n-1} \ x_{n-2} \ \ldots \ x_{n-L_w+1}]^T$ denotes the filter input samples at time $n = 1, \ldots, N$,
- L_w denotes the length of the filter impulse response.

BD: Channel-Filter Model

Channel-filter cascade output model:

$$z_{m,n} = c_m s_{n-\delta_m} + \mathcal{N}_{m,n} \quad ,$$

where:

- c_m denotes instantaneous amplitude distortion,

BD: Channel-Filter Model

Channel-filter cascade output model:

$$z_{m,n} = c_m s_{n-\delta_m} + \mathcal{N}_{m,n} \quad ,$$

where:

- c_m denotes instantaneous amplitude distortion,
- δ_m instantaneous group delay,

BD: Channel-Filter Model

Channel-filter cascade output model:

$$z_{m,n} = c_m s_{n-\delta_m} + \mathcal{N}_{m,n} \ ,$$

where:

- c_m denotes instantaneous amplitude distortion,
- δ_m instantaneous group delay,
- $\mathcal{N}_{m,n}$ denotes so-termed deconvolution noise:

BD: Channel-Filter Model

Channel-filter cascade output model:

$$z_{m,n} = c_m s_{n-\delta_m} + \mathcal{N}_{m,n} ,$$

where:

- c_m denotes instantaneous amplitude distortion,
- δ_m instantaneous group delay,
- $\mathcal{N}_{m,n}$ denotes so-termed deconvolution noise:
  - zero-mean, white, Gaussian random process,

BD: Channel-Filter Model

Channel-filter cascade output model:

$$z_{m,n} = c_m s_{n-\delta_m} + \mathcal{N}_{m,n} ,$$

where:

- c_m denotes instantaneous amplitude distortion,
- δ_m instantaneous group delay,
- $\mathcal{N}_{m,n}$ denotes so-termed deconvolution noise:
  - zero-mean, white, Gaussian random process,
  - incorrelated with the source signal.

BD: Basic Hypotheses

- Channel's impulse response satisfies $\boldsymbol{h}^T\boldsymbol{h} = 1$ and its inverse has finite energy.

BD: Basic Hypotheses

- Channel's impulse response satisfies $\boldsymbol{h}^T\boldsymbol{h} = 1$ and its inverse has finite energy.
- Channel is time-invariant or slowly time-varying.

BD: Basic Hypotheses

- Channel's impulse response satisfies $\boldsymbol{h}^T \boldsymbol{h} = 1$ and its inverse has finite energy.
- Channel is time-invariant or slowly time-varying.
- Source stream s_n is a stationary, ergodic, independent identically distributed (IID) random process with mean $\mathbb{E}_s[s_n] = 0$ and variance $\mathbb{E}_s[s_n^2] = 1$.

BD: Basic Hypotheses

- Channel's impulse response satisfies $\boldsymbol{h}^T\boldsymbol{h} = 1$ and its inverse has finite energy.
- Channel is time-invariant or slowly time-varying.
- Source stream s_n is a stationary, ergodic, independent identically distributed (IID) random process with mean $\mathbb{E}_s[s_n] = 0$ and variance $\mathbb{E}_s[s_n^2] = 1$.
- The probability density function $p_s(s)$ of the source signal is symmetric around zero and **non-Gaussian**.

BD: Applications

- Equalization of communication channels.

BD: Applications

- Equalization of communication channels.
- Optomagnetic memory-support storage and retrieval enhancement.

BD: Applications

- Equalization of communication channels.
- Optomagnetic memory-support storage and retrieval enhancement.
- Image deblurring.

BD: Applications

- Equalization of communication channels.
- Optomagnetic memory-support storage and retrieval enhancement.
- Image deblurring.
- Geophysical measurements analysis.

BD: Bussgang filtering

The model reveals that the relationship between $z_{m,n}$ and $c_m s_{n-\delta_m}$ is deterministic but for the deconvolution noise.

BD: Bussgang filtering

The model reveals that the relationship between $z_{m,n}$ and $c_m s_{n-\delta_m}$ is deterministic but for the deconvolution noise.

$$\Downarrow$$

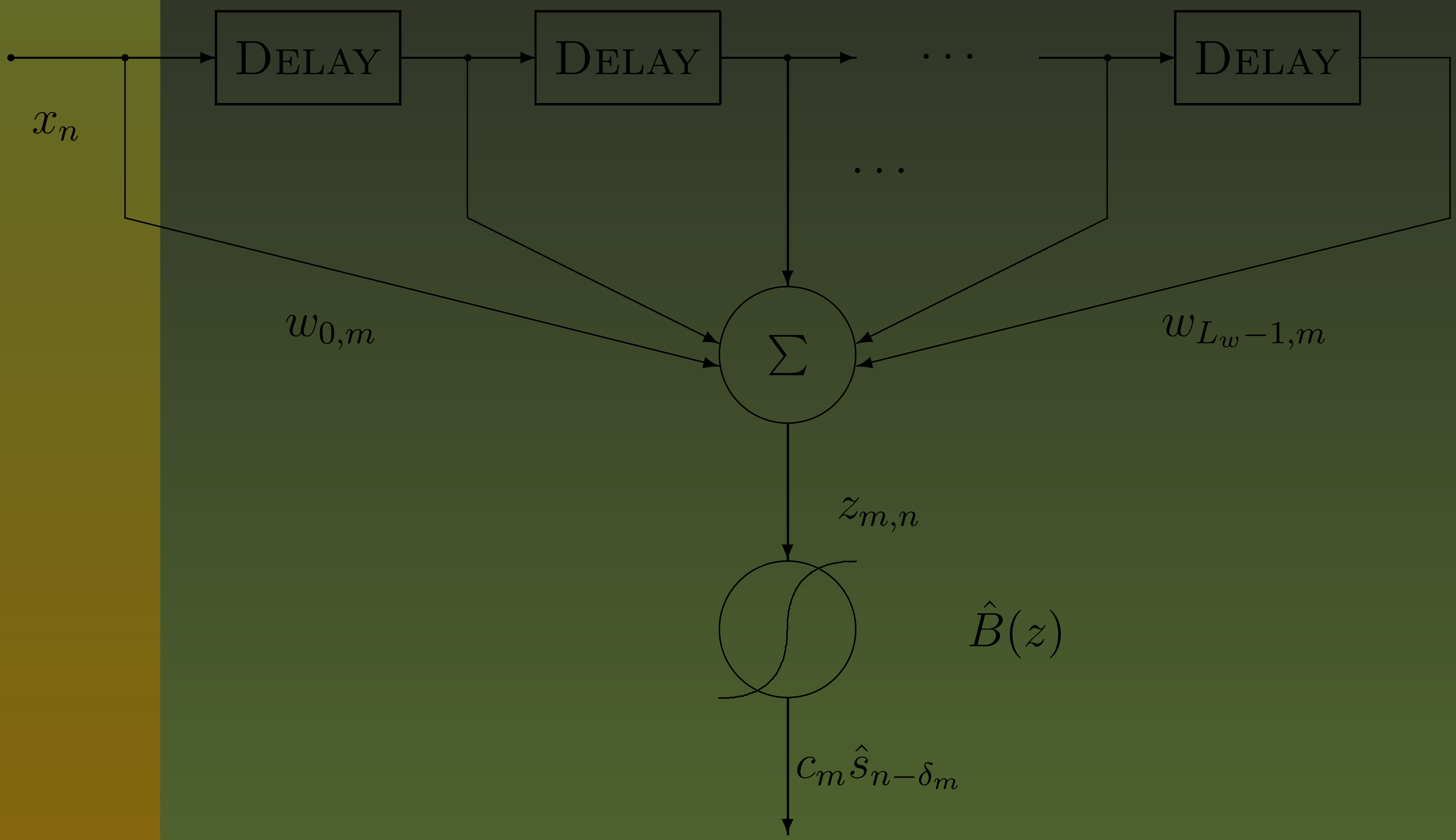
BD: Bussgang filtering

The model reveals that the relationship between $z_{m,n}$ and $c_m s_{n-\delta_m}$ is deterministic but for the deconvolution noise.

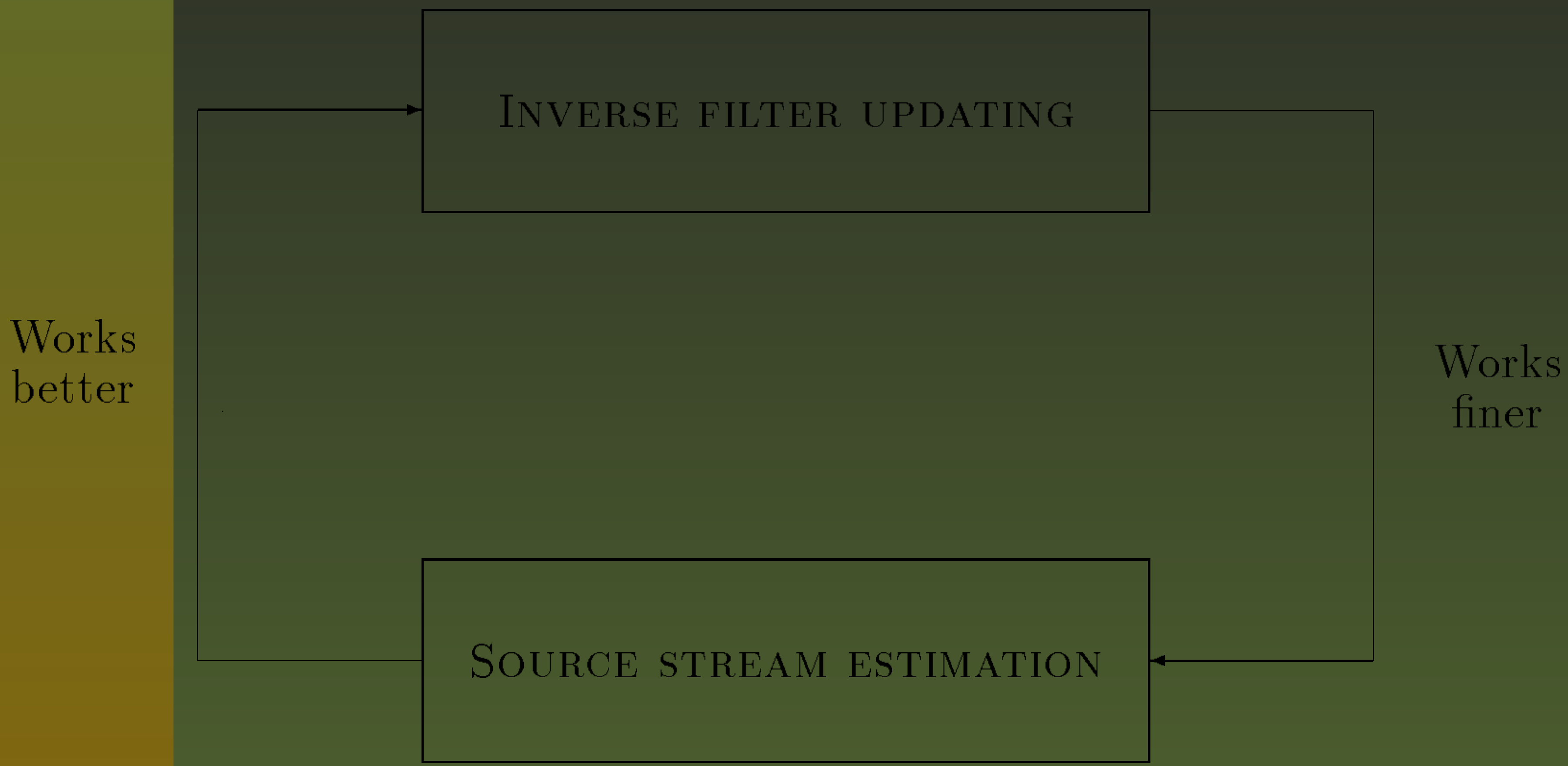
$$\Downarrow$$

An estimator of the source sequence having form $B(z_{m,n})$ can be designed according to Bayesian estimation theory.

BD: Filter Structure



A ‘Virtuous Cycle’



BD as Optimization Problem

On the basis of the available Bayesian estimator, the error criterion may be minimized:

$$C(\boldsymbol{w}_m) \stackrel{\text{def}}{=} \frac{1}{2}\mathbb{E}_{\mathcal{N}_{m,n}}[\mathcal{N}_{m,n}^2] = \frac{1}{2}\mathbb{E}_{z_{m,n}}\left[(z_{m,n} - B(z_{m,n}))^2\right] .$$

Thanks to ergodicity, the ensemble average $\mathbb{E}[\cdot]$ is estimated by:

$$\mathbb{E}_{z_{m,n}}[\Phi(z_{m,n})] \approx \frac{1}{N}\sum_{n=1}^{N}\Phi(z_{m,n}) ,$$

for any vector-valued function $\Phi : \mathcal{R} \to \mathcal{R}^p$.

BD: Automatic Gain Control

For practical reasons, it is customary to set the energy-contraint:

$$w_0^2 + w_2^2 + w_3^2 + \cdots + w_{L_w-1}^2 = 1 \,.$$

Namely, the filter's impulse response should belong – at any time – to the unit hyper-sphere:

$$S^{p-1} \stackrel{\text{def}}{=} \{\boldsymbol{v} \in \mathcal{R}^p | \boldsymbol{v}^T \boldsymbol{v} = 1\} \,.$$

Geometry of S^{p-1}

- At every point $\boldsymbol{v} \in S^{p-1}$, the tangent space has structure:

$$T_{\boldsymbol{v}} S^{p-1} \stackrel{\text{def}}{=} \{\boldsymbol{u} \in \mathcal{R}^p | \boldsymbol{u}^T \boldsymbol{v} = 0\} \, .$$

- If $S^{p-1} \hookrightarrow \mathcal{R}^p$, which is equipped with the standard Euclidean metric, then the normal space has structure:

$$N_{\boldsymbol{v}} S^{p-1} \stackrel{\text{def}}{=} \{\lambda \boldsymbol{v} | \lambda \in \mathcal{R}\} \, .$$

Riemannian Gradient on S^{p-1}

Riemannian gradient of a smooth function $f : S^{p-1} \to \mathcal{R}$ is a vector $\nabla_{\boldsymbol{v}}^{S^{p-1}} f$ that satisfies:

- Tangency: $\nabla_{\boldsymbol{v}}^{S^{p-1}} f \in T_{\boldsymbol{v}} S^{p-1}$,
- Compatibility: $\langle \nabla_{\boldsymbol{v}}^{S^{p-1}} f, \boldsymbol{u} \rangle_{\boldsymbol{v}} = \left(\frac{\partial f}{\partial \mathbf{v}} \right)^T \boldsymbol{u}, \forall \boldsymbol{u} \in T_{\boldsymbol{v}} S^{p-1}$.

With the above setting:

$$\nabla_{\boldsymbol{v}}^{S^{p-1}} f = (\boldsymbol{I}_p - \boldsymbol{v}\boldsymbol{v}^T) \frac{\partial f}{\partial \mathbf{v}} .$$

Geodesics on S^{p-1}

A geodesic $\boldsymbol{v}(t) = G(t, \boldsymbol{v}_0, \boldsymbol{g})$ is a curve on which a particle, amanating from $\boldsymbol{v}_0$ with velocity $\boldsymbol{g}$, slides with constant scalar speed $\|\boldsymbol{g}\|$.

$$\begin{cases} \ddot{\boldsymbol{v}} \in N_{\boldsymbol{v}} S^{p-1} , \\ \boldsymbol{v}(0) = \boldsymbol{v}_0 \in S^{p-1} , \\ \dot{\boldsymbol{v}}(0) = \boldsymbol{g} \in T_{\boldsymbol{v}_0} S^{p-1} . \end{cases}$$

The solution is:

$$G(t, \boldsymbol{v}_0, \boldsymbol{g}) = \cos(\|\boldsymbol{g}\| t) \boldsymbol{v}_0 + \sin(\|\boldsymbol{g}\| t) \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|} .$$

$\nabla_{\boldsymbol{v}}^{S^{p-1}}$-based Optimization

As an optimization law for searching for the minimum (or local minima) of a regular function $f : S^{p-1} \to \mathcal{R}$ over S^{p-1}, we may use the Riemannian-gradient based rule:

$$\begin{cases} \frac{d\boldsymbol{v}}{dt} = -\nabla_{\boldsymbol{v}}^{S^{p-1}} f \,, \\ \boldsymbol{v}(0) = \boldsymbol{v}_0 \in S^{p-1} \,. \end{cases}$$

BD: Geodesic-based Rule

The general-purpose differential equation may be customized as:

$$\frac{d\boldsymbol{w}}{dt} = -(\boldsymbol{I}_p - \boldsymbol{w}\boldsymbol{w}^T)\frac{\partial C(\boldsymbol{w})}{\partial \boldsymbol{w}} \ ,$$

with $p = L_w$ and:

$$\begin{cases} \frac{\partial C(\boldsymbol{w})}{\partial \boldsymbol{w}} = \mathbb{E}_{\boldsymbol{x}}[\gamma(z)\boldsymbol{x}] \ , \\ \gamma(z) \stackrel{\text{def}}{=} (B(z) - z)(B'(z) - 1) \ . \end{cases}$$

BD: Geodesic-based Algorithm

It is suggested to approximate the exact flow of the differential equation on a manifold via piece-wise geodesic arcs:

$$\boldsymbol{w}_m = G\left(t_m, \boldsymbol{w}_{m-1}, -\nabla^{S^{p-1}}_{\boldsymbol{w}_{m-1}} C(\boldsymbol{w})\right) \, , \; m \in \{1, \ldots, M\} \, .$$

where:

- t_m denotes an appropriate sequence of adaptation stepsizes,

BD: Geodesic-based Algorithm

It is suggested to approximate the exact flow of the differential equation on a manifold via piece-wise geodesic arcs:

$$\boldsymbol{w}_m = G\left(t_m, \boldsymbol{w}_{m-1}, -\nabla^{S^{p-1}}_{\boldsymbol{w}_{m-1}} C(\boldsymbol{w})\right) \, , \; m \in \{1, \ldots, M\} \, .$$

where:

- t_m denotes an appropriate sequence of adaptation stepsizes,
- $\boldsymbol{w}_0 \in S^{p-1}$ is arbitrarily selected.

BD: Geodesic-based Algorithm

It is suggested to approximate the exact flow of the differential equation on a manifold via piece-wise geodesic arcs:

$$\boldsymbol{w}_m = G\left(t_m, \boldsymbol{w}_{m-1}, -\nabla^{S^{p-1}}_{\boldsymbol{w}_{m-1}} C(\boldsymbol{w})\right)\ ,\ m \in \{1, \ldots, M\}\ .$$

where:

- t_m denotes an appropriate sequence of adaptation stepsizes,
- $\boldsymbol{w}_0 \in S^{p-1}$ is arbitrarily selected.
- *Up to numerical error*, $\boldsymbol{w}_m \in S^{p-1}$ at every iteration.

BD: Projection-based algorithm

By the embedding $S^{p-1} \hookrightarrow \mathcal{R}^p$, updates along the Euclidean gradient direction:

$$\boldsymbol{w}_m = \Pi\left(\boldsymbol{w}_{m-1} - t_m \left.\frac{\partial C(\boldsymbol{w})}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}=\boldsymbol{w}_{m-1}}\right), \quad \Pi(\boldsymbol{v}) \stackrel{\text{def}}{=} \frac{\boldsymbol{v}}{\sqrt{\boldsymbol{v}^T\boldsymbol{v}}}.$$

where:

- t_m denotes an appropriate sequence of adaptation stepsizes,

BD: Projection-based algorithm

By the embedding $S^{p-1} \hookrightarrow \mathcal{R}^p$, updates along the Euclidean gradient direction:

$$\boldsymbol{w}_m = \Pi \left(\boldsymbol{w}_{m-1} - t_m \left. \frac{\partial C(\boldsymbol{w})}{\partial \boldsymbol{w}} \right|_{\boldsymbol{w}=\boldsymbol{w}_{m-1}} \right), \quad \Pi(\boldsymbol{v}) \stackrel{\text{def}}{=} \frac{\boldsymbol{v}}{\sqrt{\boldsymbol{v}^T \boldsymbol{v}}}.$$

where:

- t_m denotes an appropriate sequence of adaptation stepsizes,
- $\boldsymbol{w}_0 \in S^{p-1}$ is arbitrarily selected,

BD: Projection-based algorithm

By the embedding $S^{p-1} \hookrightarrow \mathcal{R}^p$, updates along the Euclidean gradient direction:

$$\boldsymbol{w}_m = \Pi\left(\boldsymbol{w}_{m-1} - t_m \left.\frac{\partial C(\boldsymbol{w})}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}=\boldsymbol{w}_{m-1}}\right), \quad \Pi(\boldsymbol{v}) \stackrel{\text{def}}{=} \frac{\boldsymbol{v}}{\sqrt{\boldsymbol{v}^T\boldsymbol{v}}}.$$

where:

- t_m denotes an appropriate sequence of adaptation stepsizes,
- $\boldsymbol{w}_0 \in S^{p-1}$ is arbitrarily selected,
- $\Pi : \mathcal{R}^p \to S^{p-1}$ is the selected back-projector.

Short Geodesic Arcs

If the time to within the geodesic is extended is short enough, the geodesic-based algorithm traces the Riemannian-gradient flow.

In fact, for t small enough, the S^{p-1}-geodesic may be approximated as:

$$G(t, \boldsymbol{v}_0, \boldsymbol{g}) \approx \left(1 - \frac{\|\boldsymbol{g}\|^2 t^2}{2}\right)\boldsymbol{v}_0 + \boldsymbol{g}t \, ,$$

which gives rise to the expression:

$$\frac{\boldsymbol{w}_m - \boldsymbol{w}_{m-1}}{t} \approx -\frac{\|\nabla^{S^{p-1}}_{\boldsymbol{w}_{m-1}} C(\boldsymbol{w})\|^2 t}{2}\boldsymbol{w}_{m-1} - \nabla^{S^{p-1}}_{\boldsymbol{w}_{m-1}} C(\boldsymbol{w}) \, .$$

Pre-whitening

The source stream s_n is IID. After passing through the channel, the samples gain second-order statistical correlation.
Second-order correlation is easy to remove by data pre-whitening. Let us define:

$$\boldsymbol{R}_{\boldsymbol{xx}} \stackrel{\text{def}}{=} \mathbb{E}_{x_n}[\boldsymbol{x}_n \boldsymbol{x}_n^T] \ .$$

Whitened filter-input vector-stream:

$$\hat{\boldsymbol{x}}_n \stackrel{\text{def}}{=} \boldsymbol{R}_{\boldsymbol{xx}}^{-\frac{1}{2}} \boldsymbol{x}_n \ .$$

BD Algorithms at a Glance

- Collect the filter-input stream and build-up the multivariate stream $\boldsymbol{x}_n$.

BD Algorithms at a Glance

- Collect the filter-input stream and build-up the multivariate stream $\boldsymbol{x}_n$.
- Whiten the multivariate signal $\boldsymbol{x}_n$.

BD Algorithms at a Glance

- Collect the filter-input stream and build-up the multivariate stream $\boldsymbol{x}_n$.
- Whiten the multivariate signal $\boldsymbol{x}_n$.
- Choose a starting point for the inverse filter impulse response $\boldsymbol{w}_0$ and learning parameters.

BD Algorithms at a Glance

- Collect the filter-input stream and build-up the multivariate stream $\boldsymbol{x}_n$.
- Whiten the multivariate signal $\boldsymbol{x}_n$.
- Choose a starting point for the inverse filter impulse response $\boldsymbol{w}_0$ and learning parameters.
- Compute the final inverse filter impulse response $\boldsymbol{w}_M$ by the geodesic-based algorithm or the projection-based algorithm applied to the whitened input stream.

Figures of Performance

- Residual inter-symbol interference (ISI):

$$\mathrm{ISI}_m \stackrel{\mathrm{def}}{=} \frac{\boldsymbol{T}_m^T \boldsymbol{T}_m - T_{m,\max}^2}{T_{m,\max}^2} ,$$

where $\boldsymbol{T}_m \stackrel{\mathrm{def}}{=} \boldsymbol{h} \otimes \boldsymbol{w}_m$ and $T_{m,\max}$ denotes the component of $\boldsymbol{T}_m$ having maximal absolute value.

Figures of Performance

- Residual inter-symbol interference (ISI):

$$\mathrm{ISI}_m \stackrel{\text{def}}{=} \frac{\boldsymbol{T}_m^T \boldsymbol{T}_m - T_{m,\max}^2}{T_{m,\max}^2},$$

  where $\boldsymbol{T}_m \stackrel{\text{def}}{=} \boldsymbol{h} \otimes \boldsymbol{w}_m$ and $T_{m,\max}$ denotes the component of $\boldsymbol{T}_m$ having maximal absolute value.

- Elapsed run-time on a 1.86GHz – 512MB platform.

Figures of Performance

- Residual inter-symbol interference (ISI):

$$\mathrm{ISI}_m \stackrel{\text{def}}{=} \frac{\boldsymbol{T}_m^T \boldsymbol{T}_m - T_{m,\max}^2}{T_{m,\max}^2} ,$$

  where $\boldsymbol{T}_m \stackrel{\text{def}}{=} \boldsymbol{h} \otimes \boldsymbol{w}_m$ and $T_{m,\max}$ denotes the component of $\boldsymbol{T}_m$ having maximal absolute value.

- Elapsed run-time on a 1.86GHz – 512MB platform.
- Flops (counted by Matlab© 5.3).

Source and Bayesian Estimator

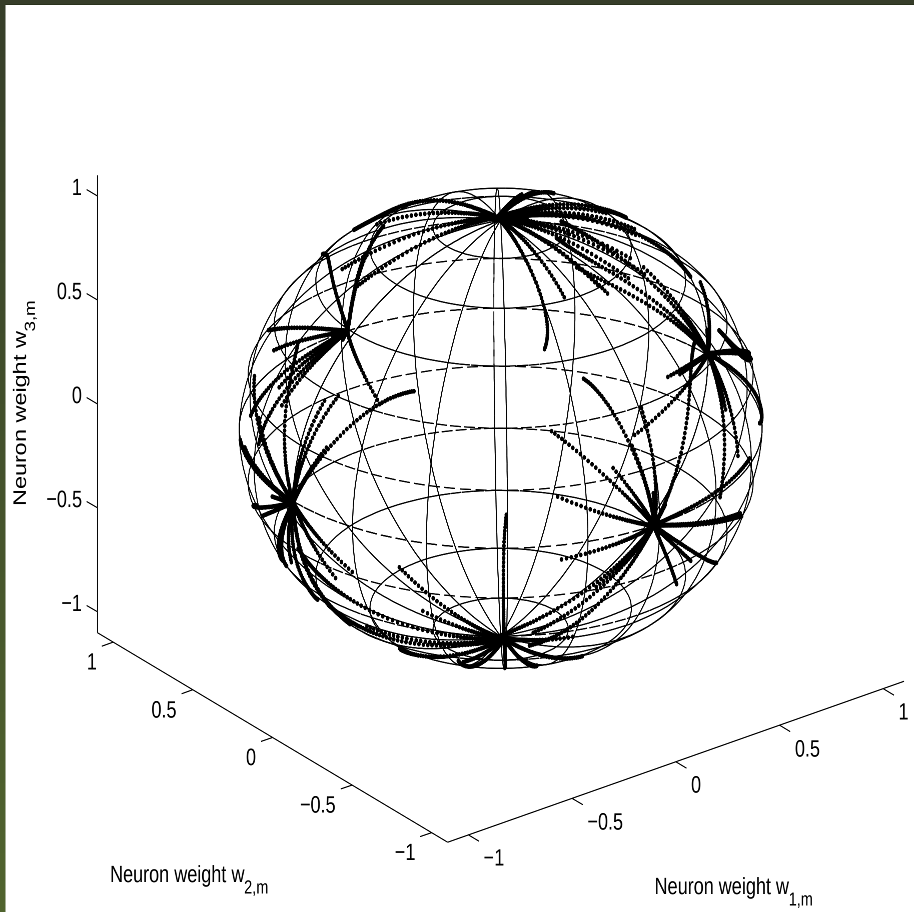
- It is assumed that s_n is a white random signal, uniformly distributed within $[-\sqrt{3}, +\sqrt{3}]$, counting $N = 5,000$ samples.
- In this case, a suitable Bayesian estimator is $\hat{B}(z) = \kappa \tanh(\lambda z)$.
- Parameters κ and λ may be pre-learnt on the basis, e.g., of the procedure introduced in S. FIORI, *Analysis of modified 'Bussgang' algorithms (MBA) for channel equalization*, IEEE Trans. on Circuits and Systems - Part I, Vol. 51, No. 8, pp. 1552 – 1560, August 2004.

Experiments on a Toy Channel

- The channel's impulse response is $\boldsymbol{h} = [1]$ ($L_h = 1$) and the base manifold is S^2 ($L_w = 3$).
- In this experiment, the channel-filter-cascade impulse response $\boldsymbol{T}_m = \boldsymbol{h} \otimes \boldsymbol{w}_m = \boldsymbol{w}_m$.
- If we let the learning trajectories depart from randomly generated $\boldsymbol{w}_0 \in S^2$, they should eventually converge to one of the six attractors $[\pm 1\ 0\ 0]^T$, $[0\ \pm 1\ 0]^T$ or $[0\ 0\ \pm 1]^T$.

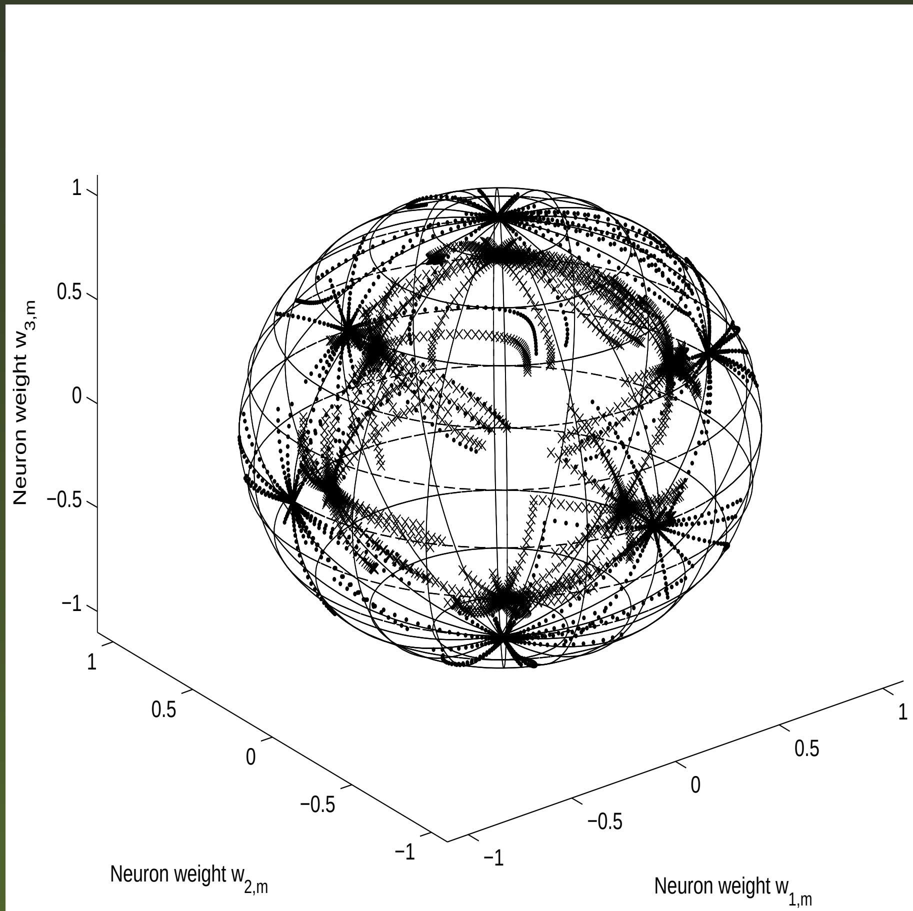
Toy Channel – Geodesic

Numerical results on 100 independent trials, $M = 100$ learning iterations per trial, learning stepsize 0.5.



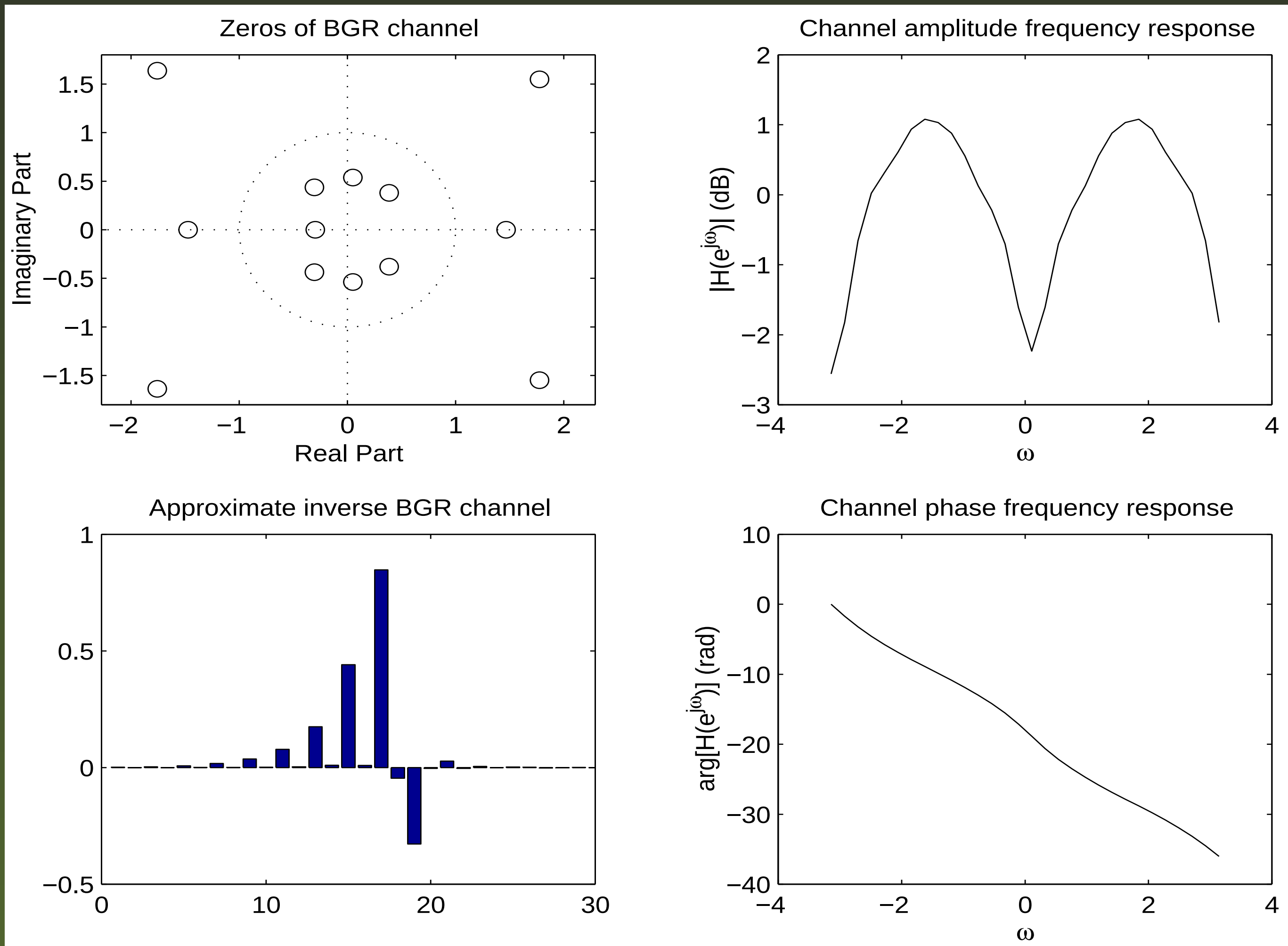
Toy Channel – Projection

Numerical results on 100 independent trials, $M = 100$ learning iterations per trial, learning stepsize 0.9.



Experiments on Telephonic Channel

Sampled telephonic channel having duration $L_h = 14$.



Experiments on BGR: Data

- Filter of length $L_w = 14$.

Experiments on BGR: Data

- Filter of length $L_w = 14$.
- $\boldsymbol{w}_0 = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$.

Experiments on BGR: Data

- Filter of length $L_w = 14$.
- $\boldsymbol{w}_0 = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$.
- Noiseless channel (i.e., with $\nu_n \equiv 0$ identically).

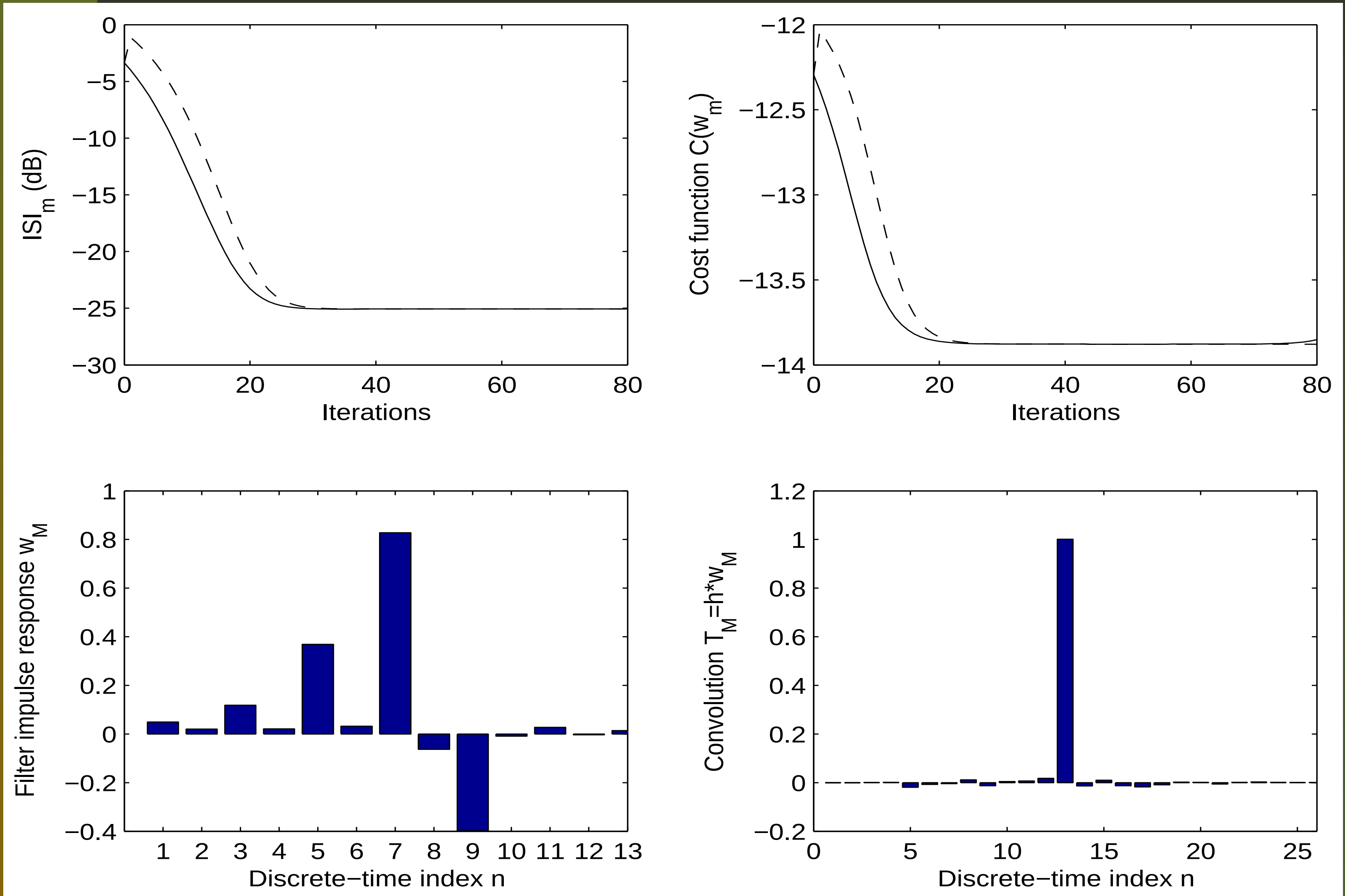
Experiments on BGR: Data

- Filter of length $L_w = 14$.
- $\boldsymbol{w}_0 = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$.
- Noiseless channel (i.e., with $\nu_n \equiv 0$ identically).
- Learning stepsize: 1 for the geodesic-based algorithm and 0.9 for the projection-based algorithm.

Experiments on BGR: Data

- Filter of length $L_w = 14$.
- $\boldsymbol{w}_0 = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$.
- Noiseless channel (i.e., with $\nu_n \equiv 0$ identically).
- Learning stepsize: 1 for the geodesic-based algorithm and 0.9 for the projection-based algorithm.
- Learning iterations: $M = 80$ for both algorithms.

Experiments on BGR: Results



Numerical Complexity Comparison

- Algorithms were run on the same batch of $5,000$ channel output samples.

Numerical Complexity Comparison

- Algorithms were run on the same batch of $5,000$ channel output samples.
- learning iterations: $M = 50$.

Numerical Complexity Comparison

- Algorithms were run on the same batch of $5,000$ channel output samples.
- learning iterations: $M = 50$.
- The flops count refers to the number of floating point operations required by the implemented code to run, averaged over the total number of samples passing by $(5,000 \times 50)$.

Numerical Complexity Comparison

- Algorithms were run on the same batch of $5,000$ channel output samples.
- learning iterations: $M = 50$.
- The flops count refers to the number of floating point operations required by the implemented code to run, averaged over the total number of samples passing by $(5,000 \times 50)$.
- The time count refers to the total time required by each algorithm to run on the specified platform.

Complexity Comparison: Results

Results of computational-complexity comparison of the geodesic-based algorithm and the projection-based algorithm.

ALGORITHM	ISI (dB)	Flops	Time (sec.s)
Geodesic-based	–25.057	80.594	0.328
Projection-based	–25.056	81.582	0.313

Summary

- Both algorithms are well-behaving.

Summary

- Both algorithms are well-behaving.
- The deconvolution performances are comparable for the two algorithms.

Summary

- Both algorithms are well-behaving.
- The deconvolution performances are comparable for the two algorithms.
- The geodesic-based algorithm may exhibit steadier convergence.

Summary

- Both algorithms are well-behaving.
- The deconvolution performances are comparable for the two algorithms.
- The geodesic-based algorithm may exhibit steadier convergence.
- The projection-based algorithm may be slightly lighter from a computational point of view.

Many thanks to...

- The organizers and E.T. Jaynes Foundation!

Many thanks to...

- The organizers and E.T. Jaynes Foundation!
- Everybody for the kind attention!

Many thanks to...

- The organizers and E.T. Jaynes Foundation!
- Everybody for the kind attention!
- The Italian team for winning the World Cup!!!!