Exploiting Diversity and Specialization in Intelligent Systems via Adaptive Combination

Jerónimo Arenas-García

DTSC, Universidad Carlos III de Madrid

http://www.tsc.uc3m.es/~jarenas

Sep 2nd, 2010

Thanks to …

Aníbal R. Figueiras-Vidal (UC3M)

Manel Martínez-Ramón (UC3M)

Luis A. Azpicueta-Ruiz (UC3M)

Vanessa Gómez-Verdejo (UC3M)

Miguel Lázaro-Gredilla (UC3M)

Ali H. Sayed (UCLA)

W. Kellermann (Erlangen-Nürnberg)

Vítor H. Nascimento (U P Sao Paulo)

Magno T. Madeira Silva (U P Sao Paulo)

Jan Larsen (Denmark TU)

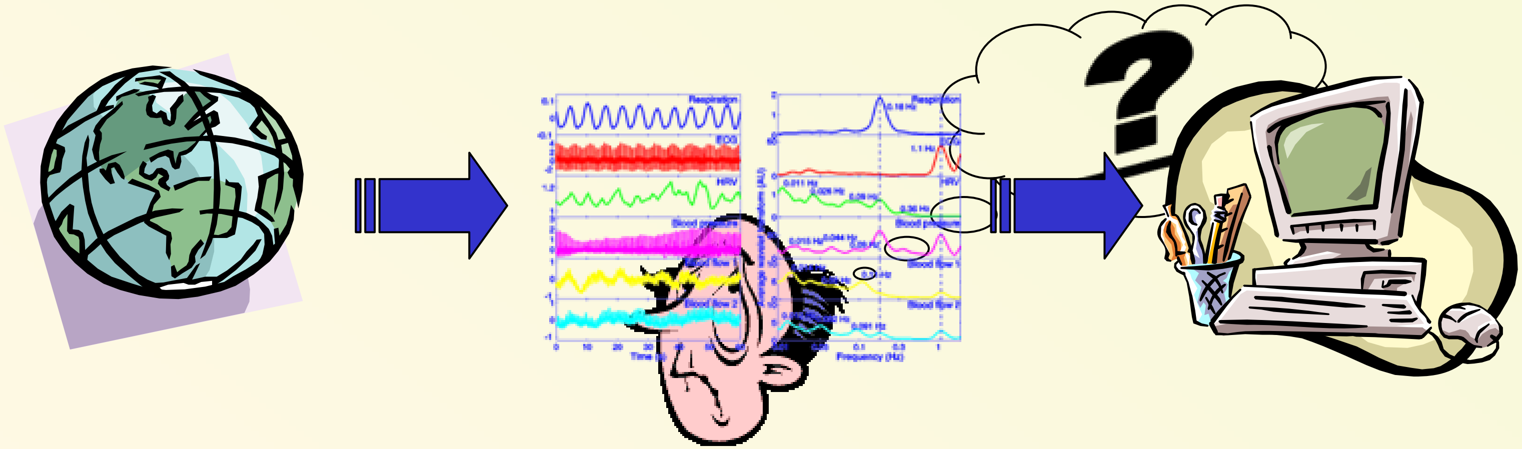
Marcus Zeller (Erlangen-Nürnberg)

Outline

1. Learning from Data
2. Batch vs Sequential Supervised Learning
3. Combination methods in Batch Learning
4. Improving performance in Adaptive Systems via Adaptive Combination
5. Selected Examples
   - Echo cancellation
   - Blind Adaptive Equalization
6. Conclusions and expected future research

Learning from Data

Aim: To extract information contained (in a not obvious way) in noisy data or signals



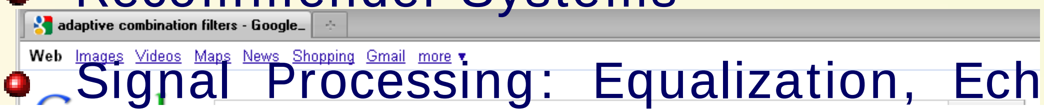
Typical Learning Tasks:

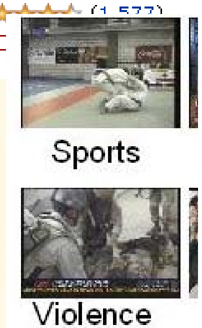
- Supervised (Regression, Classification)
- Unsupervised (Clustering, Novelty Detection, Pdf modelling)
- Other (e.g., Collaborative Filtering, Reinforcement Learning)

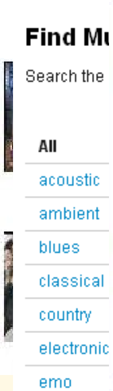
Learning from Data (II): Applications

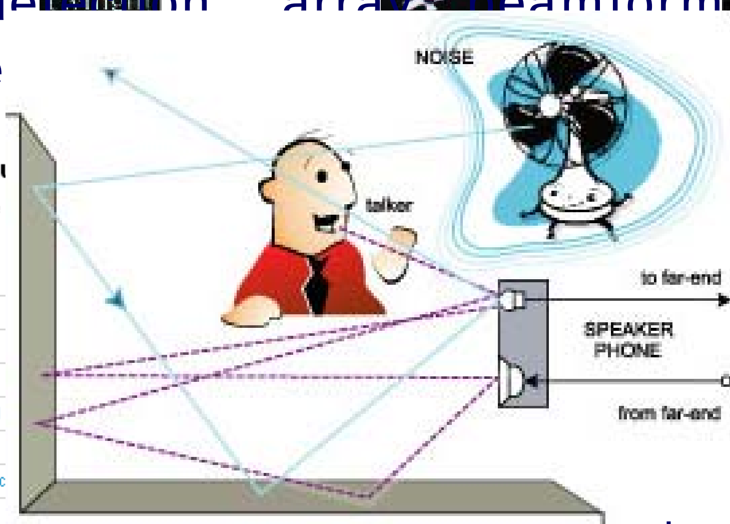
Applications of such Learning Systems are wide, and include, e.g.:

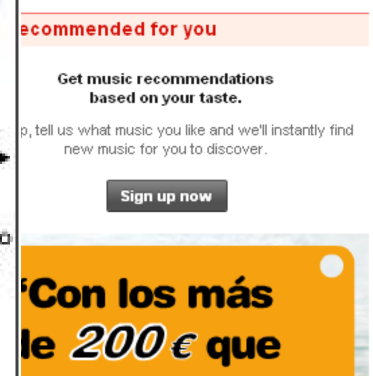
- Text/Video Categorization and Retrieval
- Recommender Systems
- Signal Processing: Equalization, Echo Cancellation, Source Separation, etc. in communication systems (IP telephony, hands-free telephony, ASR)
- Radar detection, array beamforming, soil classification, music ge … y more











www.adaptivedigital.com

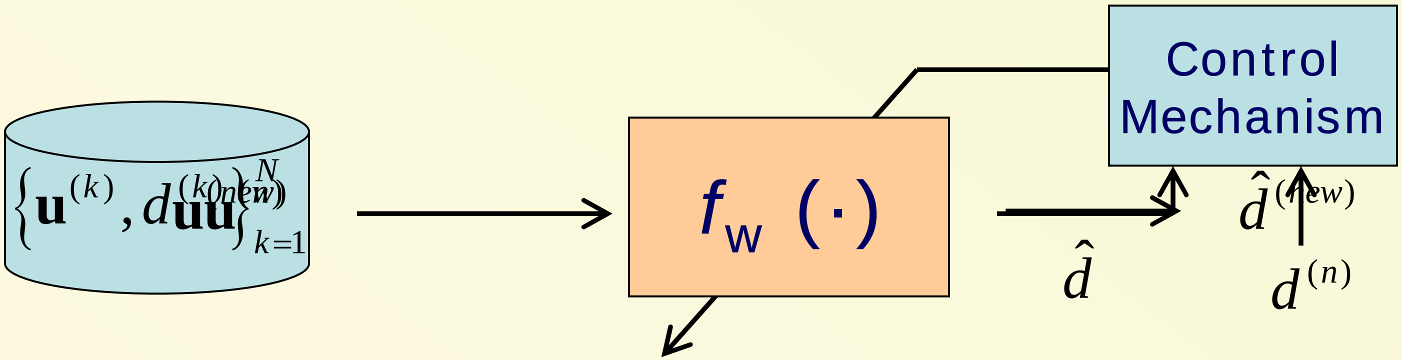
Learning from Data (III)

Keypoints:

- Increasing accessibility to data in digital format
  (including Internet, companies databases ...)
- Cheaper HW (storage, processing, sensing)
- No statistical information required

The bottom line is “Data is everywhere and we now have the necessary resources (algorithms and hardware) to extract and exploit the information they contain)”

Batch vs Sequential Supervised Learning



Batch Learning:

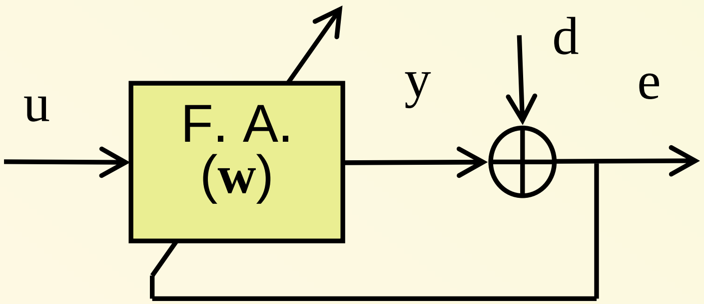
- Training Phase: Available data is used to adjust the weights and free parameters of the system
- Operation Phase: Target values are predicted for new (unseen data)

Sequential Learning:

- Training examples become available over time $\{\mathbf{u}^{(n)}, d^{(n)}\}$, and the system is adapted one sample at a time
- Adaptivity: The system can forget old samples
- Typically, computationally more attractive than batch learning

Adaptive Signal Filtering

- Systems that carry out transformations on some input signal to optimize some performance goal (typically, squared error)



- If the statistics were known, an optimum filter could be design
- When this is not the case, or in time varying situations, we can recur to **ADAPTIVE SCHEMES**

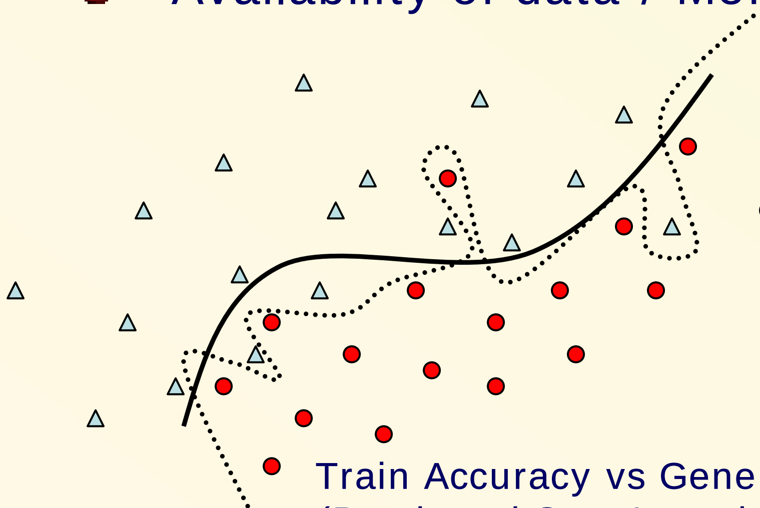
- Traditionally, SW and HW constraints (e.g., energy consumption) limited these systems to linear structures and simple algorithms

- Modern Digital Signal Processing (DSP) tools include
  - Powerful algorithms: Particle Filters, Extended Kalman Filter (EKF), Set-Membership Methods ...
  - Non-linear structures: Kernel AF, Volterra Series, ...

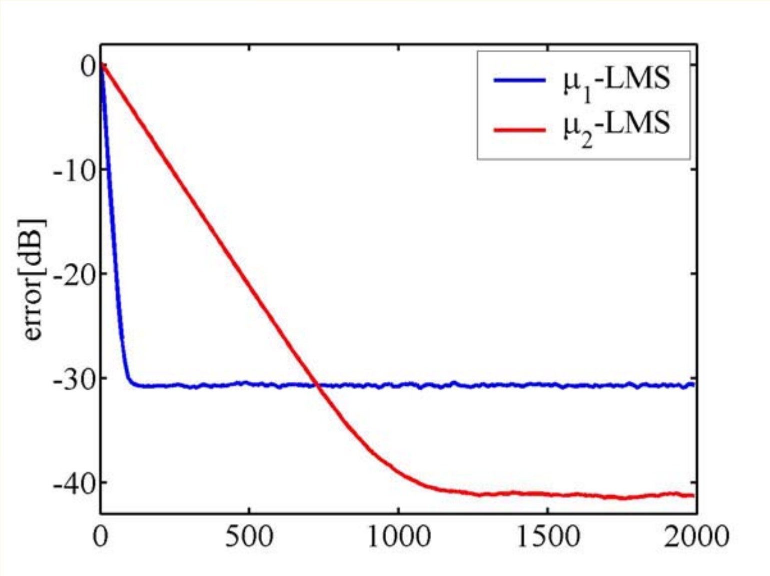
Limitations and Compromises

When designing and applying learning algorithms (both batch and sequential), it is necessary to be aware of several related, and some times conflicting, issues:

- Difficulty of the learning task
- Expressiveness of the method
- Generalization properties
- Availability of data / Memory
- Algorithm complexity
- Convergence + Tracking
- Accurateness of the solution



Train Accuracy vs Generalization
(Batch and Seq. Learning)

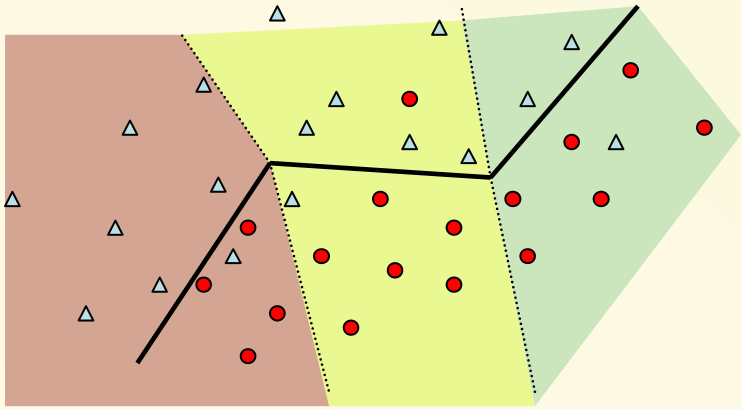


Convergence vs Steady-State error
(Seq. Learning)

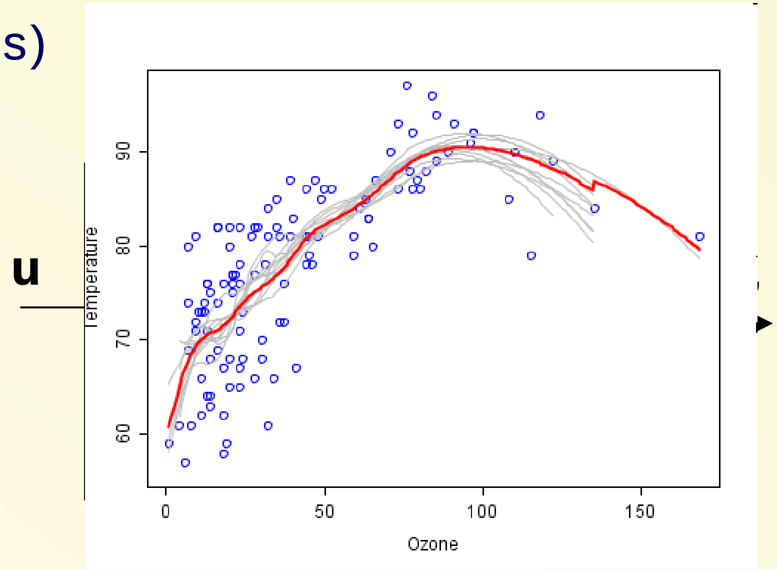
Exploiting diversity + Specialization

Combination methods have been exploited mostly under the batch learning paradigm. By inducing **diversity** and/or **specialization** among several component networks, the obtained solutions:

- Provide superior performance
- Are conceptually simpler
- Are computationally more affordable (besides distributed implementations)
- Can be more easily interpreted



Exploiting Specialization



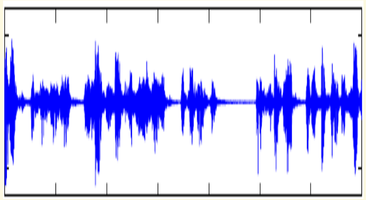
Exploiting Diversity

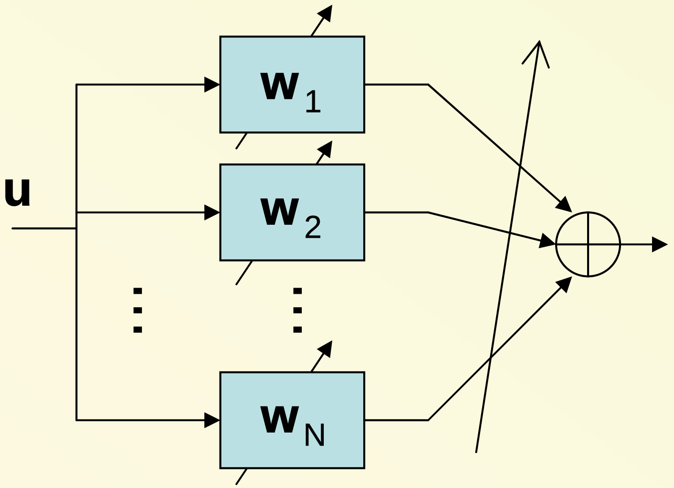
Some Milestones

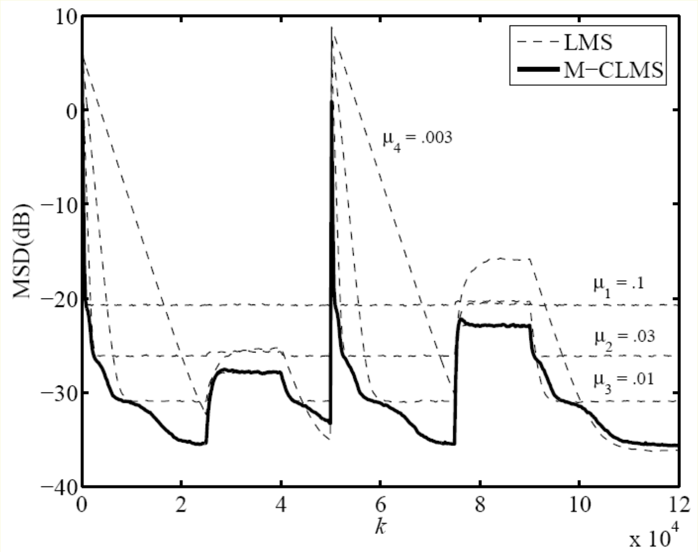
(1990)	R. Schapire: Boosting
(1991)	R. Jacobs et al.: Mixtures of Experts
(1995)	Y. Freund and R. Schapire: Adaboost
(1996)	L. Breiman: Bagging
(2001-2010)	Multi-Classifier Systems Workshop

- Combination methods have been applied to improve the performance of most batch learning methods (NNs, Regression and Classification Trees, SVMs, etc), and in many different applications
- However, apart from the study of some general principles under the sequential learning approach, their application to Adaptive Filtering of signals has mostly been considered over the last five years
- Operation principles are probably sufficiently general to be exported to other engineering fields

Exploiting diversity + Specialization in Adaptive Systems





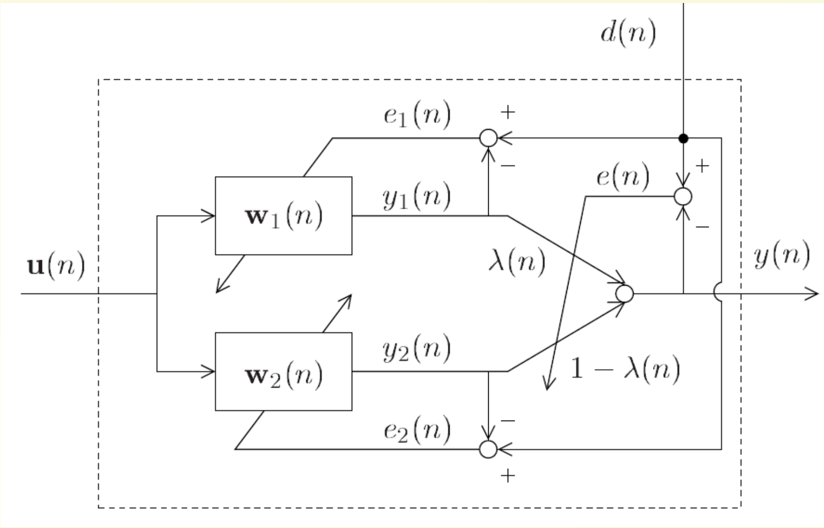


- The performance of AF is typically affected by the selection of free parameters, whose optimal selection would require certain knowledge about the filtering scenario
- Working principles are simple: select complementary components (specialization) and update combination parameters to optimize some overall performance goal

(Singer and Erdogan, 1998) Model Selection

(Martínez-Ramón et al., 2002) Speed vs precision (biological inspiration)

Basic convex combination of two filters



- Component filters are independently adapted according to their own rules and error signals

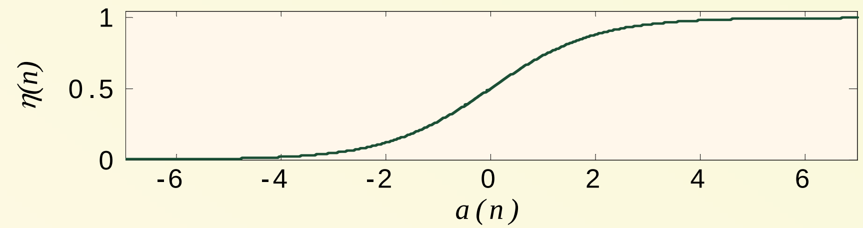
$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mathbf{f}_i[\mathbf{w}(n), \mathbf{u}(n), d(n), \mathbf{q}_i(n)]$$

- Combination output is obtained as

$$y(n) = \lambda(n) y_1(n) + [1 - \lambda(n)] y_2(n)$$

- For the convex combination case $\lambda(n) \in [0, 1]$, it is convenient to define

$$\lambda(n) = \mathrm{sigm}[a(n)] = \frac{1}{1+\exp^{-a(n)}}$$



- Mixing parameter is adapted to minimize overall error, e.g.,:

$$a(n+1) = a(n) - \mu_a \frac{\partial e^2(n)}{\partial a(n)}$$

Theoretical results (Arenas et al., 2006)

It can be shown that, if properly designed, the combination will be operating in one of the following two regimes

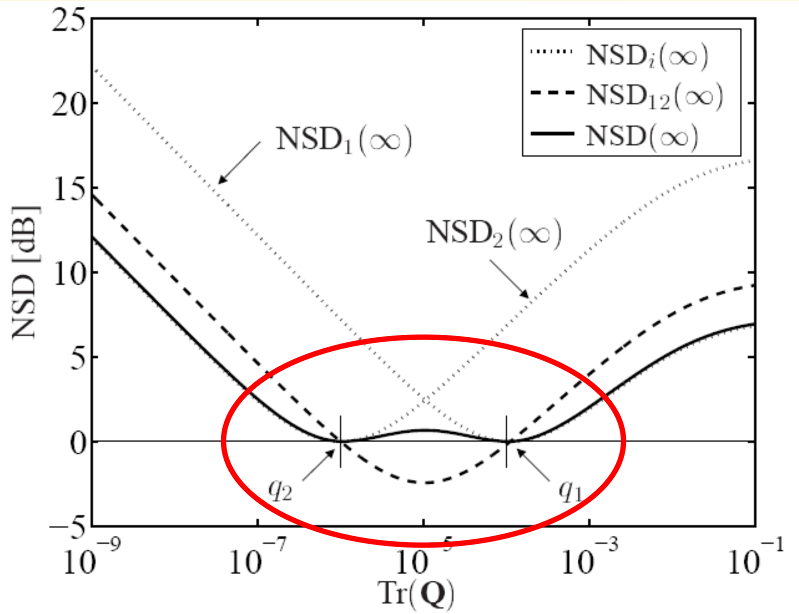
- CASE 1: when the cross-correlation between filter errors is in between mean square errors, then the combination performs like the best operating filter (specialization)
- CASE 2: when the cross-correlation between filter errors is smaller than mean square errors, then the combination outperforms both component filters (diversity)

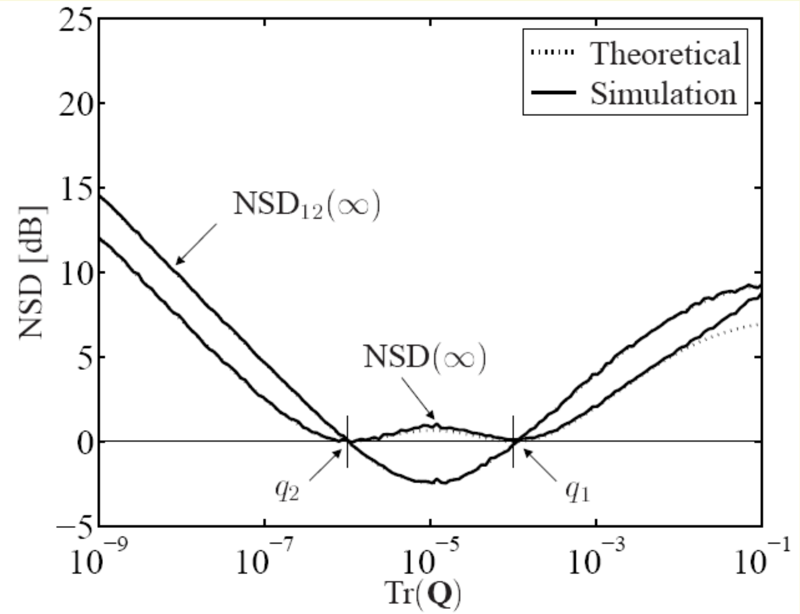
  This “better-than-universal” behavior appears, e.g.,

  - In tracking situations with fast and slow filters
  - When combining heterogeneous filters (e.g., LMS and RLS)
  - Combinations of proportionate filters with different asymmetry factors (echo cancellation applications)
  - ?

Theoretical results (II)

For instance, when combining filters with different memories, the following behavior is observed in a tracking situation (the optimal solution continuously changes with speed Tr(**Q**)):





Combination Methods + Analysis

(Kozat and Singer, 2000)	Unconstrained combination of M filters
(Arenas-García et al., 2006)	Convex combination of 2 filters
(Arenas-García et al., 2005)	Convex combination of M filters
(Bershad et al., 2008)	Affine combination of 2 filters
(Azpicueta-Ruiz et al., 2008)	Normalized convex combination of 2 filters
(Nascimento and Silva, 2008)	Tracking analysis of convex combinations
(Arenas-García et al., 2009)	Block-based combination of M filters
(Nascimento et al., 2010)	Convergence analysis
(Kozat et al., 2010)	Steady-state analysis (unifying framework)

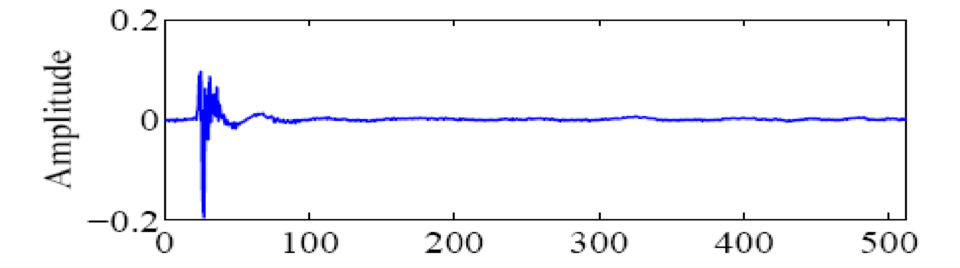
Adaptive Filtering issues that have already been tackled with “general purpose” combinations schemes:

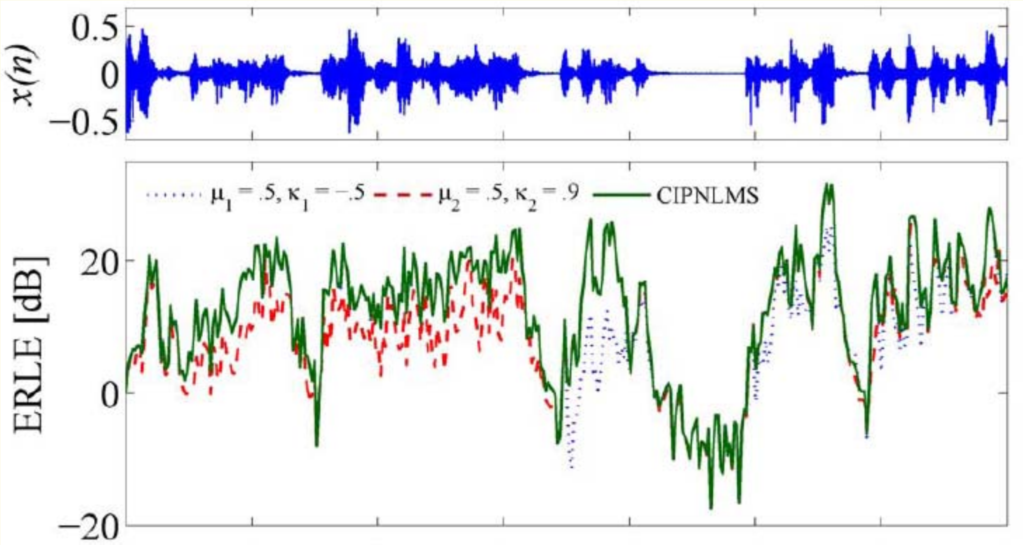
- Model Selection
- Convergence vs steady-state error tradeoff
- Tracking capabilities
- Robustness to non-Gaussian noise
- Stability

Selected examples: acoustic echo cancellation

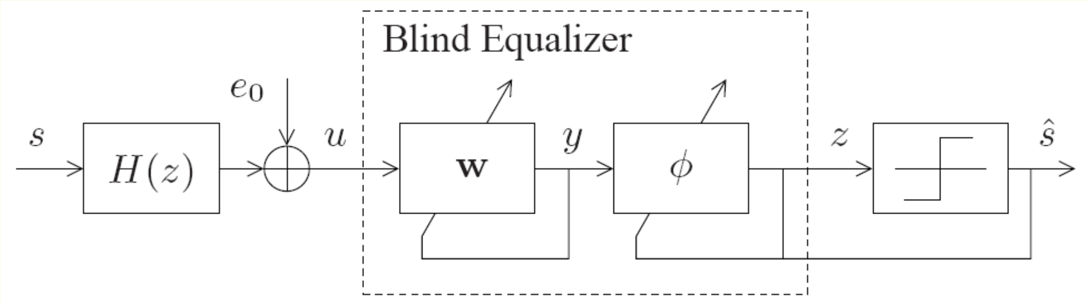
In the context of echo cancellation, combinations have been applied to:

- Improve robustness to unknown or varying SNR
- Improve the identification of sparse channels
- Managing the linear-non-linear cancellation trade-off with Volterra Filters
- Learning the size of Volterra kernels in non-linear echo cancellation



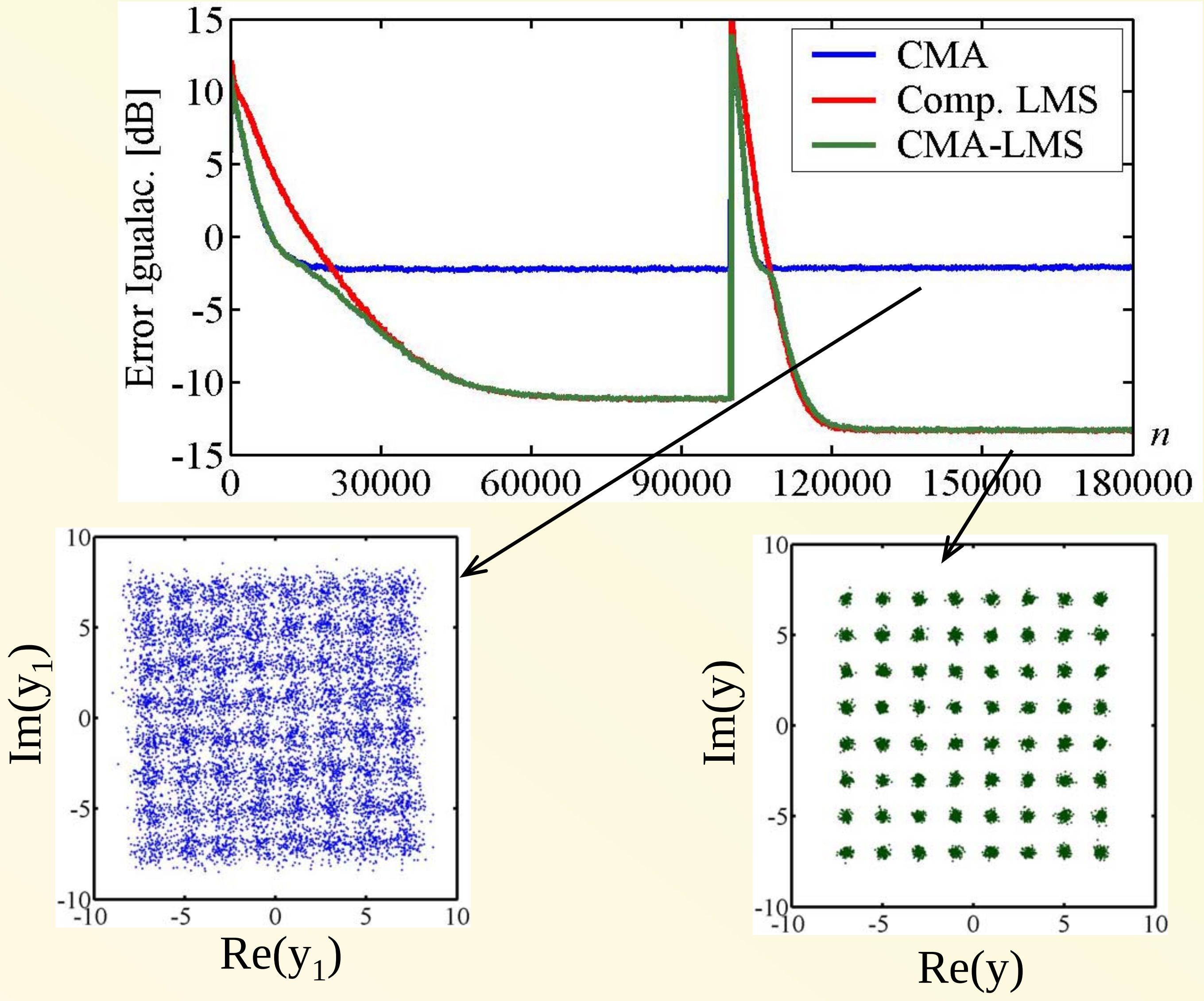


Selected examples: blind equalization



- Essentially a classification problem (which symbol was transmitted?)
- The goal of the equalizer is to compensate the distortion introduced by the channel
- Supervised Adaptive Filters can be applied, but require a training sequence to converge
- Blind Adaptive Filters do not need such sequence, thus saving wideband; on the downside, they incur in larger steady-state error
- Running both kinds of algorithms in parallel and combining their outputs allow no training sequence + accurate solution

Selected examples: blind equalization (II)



Conclusions: Advantages of the approach

We have presented a general approach to improve the properties of virtually any Adaptive Filter or Sequential Learning Scheme:

- No assumptions are imposed on the component filters. Conceptually very simple
- Completely general

  (although "ad hoc" combination strategies can be derived for particular applications)
- "Better than universal" performance is possible
- An effective approach to fight against the lack of knowledge about the learning task

The approach exploits a very general principle:

"Adaptively combining complementary (specialized) and diverse solutions can lead to improved solutions in an easy way"

Conclusions: Next results in AF

Some guesses on what we will see in the near future in Adaptive Filtering:

- Studying combinations based on new algorithmic components, and new combination rules
- Incorporation into other DSP applications
- Enforcing low-correlation among components by design
- Fight other performance trade-offs:
  - Topology selection (e.g., kernel or kernel parameter selection in Kernel Adaptive Filtering, active learning)
  - Generalization control in non-linear adaptive filtering
  - Semi-supervised learning
- Implementation in distributed scenarios (sensor networks)

Thank you

http://www.tsc.uc3m.es/~jarenas

http://www.tsc.uc3m.es/~jarenas/publications