Signal Processing by Neural Networks

Aurelio Uncini

INFOCOM Dept.

University of Rome “La Sapienza”

e-mail: aurel@ieee.org

International Joint Conference on Neural Networks, Washington, DC USA, July 18 2001

Agenda

- ⌘ Neural architectures for real-time DSP
  - Non-linear generalizations of FIR-IIR filters by Dynamic Multilayer Perceptron (DMLP) neural networks;
  - Fast adaptive spline neural model for signal processing
- ⌘ Some Applications

linear DSP: it's enough ?

- ⌘ major classical DSP techniques are based on linear models
- ⌘ *in the real world the processes are non-linear*
- ⌘ linear structures could not be able to model them adequately

non-linear DSP: *classical approaches*

specific or generic?

- ⌘ *specific non-linear architectures*
  - particular class of problems
  - efficient but specific
- ⌘ e.g.
  - median and bilinear filters
  - some spectral analysis techniques
  - ...

- ⌘ *generic non-linear architectures*
  - large class of problem
  - general but complex
- ⌘ e.g.
  - Volterra filters
  - non-linear state equations
  - polynomial filters
  - functional links
  - ...

non-linear DSP: *desired approaches*

generic and efficient

- ⌘ *"standard" architecture with scalable complexity;*
- ⌘ *capability to approximate any non-linear (dynamic) behaviour;*

- ⌘ *"standard" design algorithms;*
- ⌘ *good implementation efficiency*
  - use of the available know-how on linear filters;
  - use of the available DSP processors for real-time audio applications.

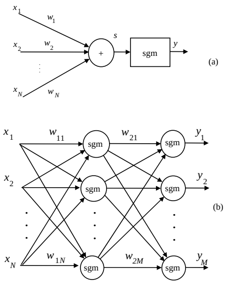
non-linear DSP: *a different approach*

⌘ Dynamic Artificial Neural Networks ***(discrete time ANN)***

⌘ ***dynamic = capability of processing temporal sequences***

- how to get dynamic behaviours from an ANN;
- non-linear FIR/IIR architectures;
- design methods or *learning process*.

ANN: Multi-Layer model



"Artificial neurons" arranged in layers Multi-Layer Perceptron (MLP)

feedforward structure without delays
no dynamics

approximation of any non-linearity
Cibenko '88, Hornik et al. '89
non-linear universal approximator

7

7

Dynamic Multilayer Networks (DMLP)

Delay lines (memory elements) put dynamic in the multilayer model

⌘ *External memory*
the non-linear filter is built using a static MLP inside a FIR/IIR framework

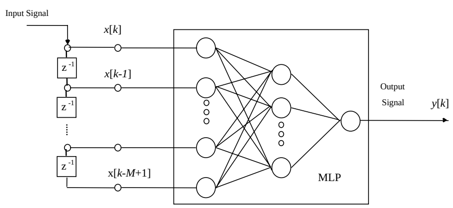
external delay lines ("buffers")
Narendra-Partharasarathy 1990

⌘ *Internal memory*
the non-linear filter is built using a MLP with neurons containing FIR/IIR filters

internal delay lines ("dynamic neuron")
Waibel et al. 1989, Back-Tsoi 1991/94, Frasconi et al. 1992, Campolucci et al. 1999, etc.

8

DMLP with external memory ***FIR scheme***



⌘ allows to use the classical multilayer model;

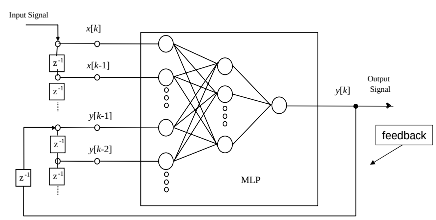
⌘ straightforward generalization of the linear **FIR filters**: linear filter ≡ DMLP with one neuron + (f(.) = identity)

Universal Functional Approximator

9

9

DMLP with external memory ***IIR scheme***

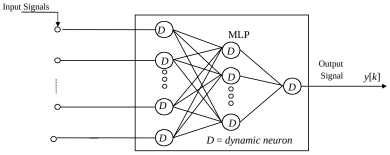


⌘ allows to use the classical multilayer model;

⌘ straightforward generalization of the linear **IIR filters**: linear filter ≡ DMLP with one neuron + f(.) = identity

DMLP with internal memory

multilayer network composed by ***"dynamic" neurons***



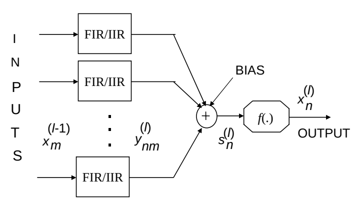
⌘ several different research contributions (already object of studies);

⌘ another generalization of the linear FIR/IIR filters: linear filter ≡ DMLP with one neuron + (f(.)=identity)

11

DMLP with internal memory

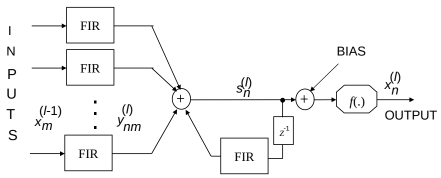
FIR/IIR synapses



12

DMLP with internal memory

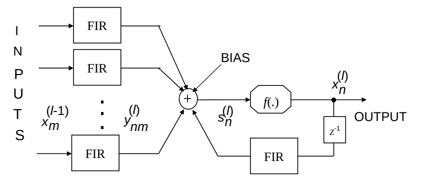
Activation feedback



13

DMLP with internal memory

Output feedback



14

DMLP: design methods

⌘ *Linear filters*

⌘ *Static design*
⌘ desired response in frequency, windowing, equiripple, Deczky, etc.

⌘ *Adaptive*
⌘ desired response in time (or frequency), LMS, RLS, etc.

⌘ *DMLP - ANN*

design ≡ "learning", by examples (supervised or unsupervised), desired response in time.

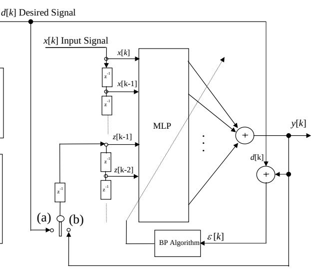
15

DMLP: learning with external memory

FIR/IIR buffers
BackPropagation (Rumelhart 1986), LMS or RLS extension (LS)

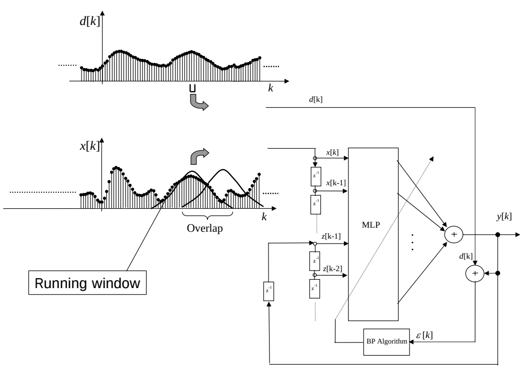
(a) *Equation error* $z[k]=d[k]$
(here "teacher forced")

(b) *Output error* $z[k]=y[k]$



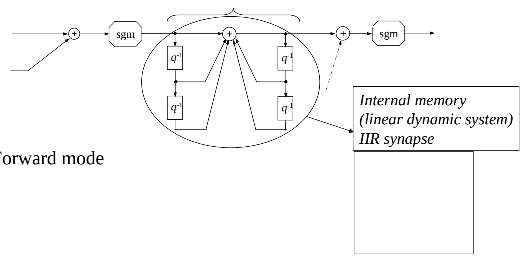
16

External delay-line Learning Example (supervised)



1

DMLP: learning with internal memory



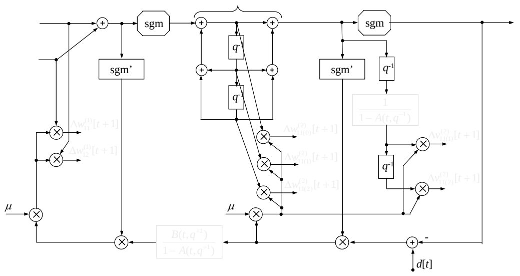
Forward mode

The learning is *non-causal* => batch mode

18

DMLP: Recursive Backpropagation (RBP)

RBP signal flow graph



RBP: **General framework** to derive several *causalised* on-line approximations learning algorithms for DMLP

19

DMLP: complexity, implementation

⌘ *Computational complexity* (fast learning algorithms)

⌘ *Structural complexity* (in terms of number of interconnections)

A ***suitable activation function*** can increase the Neural Network representation property

Fewer neurons needed

- small *structural* complexity
- small *computational* complexity

20

A New Neuron Architecture

⌘Theoretical framework

- Extention of the Cybenko theorem : (Hornik, Stinchcombe, White, 1989)
- neurons which present any "squashing" activation function maintain the **universal approximation characteristics**

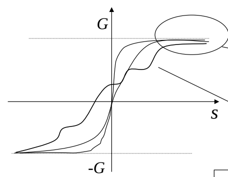
⌘GOALS FOR THE NEW NEURON

- A "more sophisticate" activation function;
- Retains the squashing property of the sigmoid;
- Necessary smoothing characteristic;
- Easy to implement both in hardware and in software
- **Flexible** (by the adaptation of few free parameters).

21

A New Neuron Architecture's Activation Function

⌘Two needed properties



- Squashing characteristic
- Flexible shape

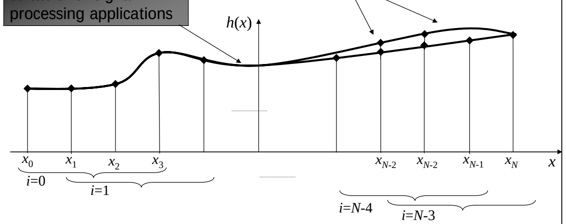
New Neural Network Architecture with ***Flexible Adaptive*** *Activation Function*

22

Adaptive Spline Neural Network (ASNN)

Smooth characteristic: suitable for signal processing applications

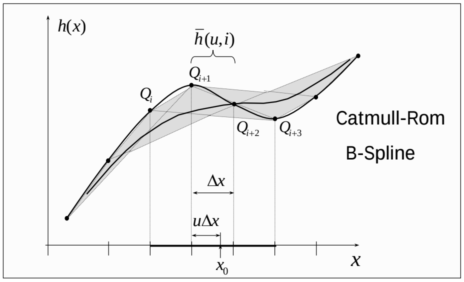
Local shape adaptation



Using *Catmull-Rom* or *B-splines* functions, with uniformly spaced control points, the **shape of the activation function is controlled by few parameters** (3rd degree: four control points).

2

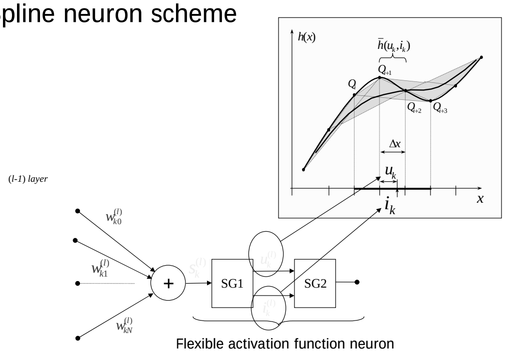
ASNN: Spline Activation Function



Spline activation function is specified as the weighted average of the four equally spaced control points

24

Spline neuron scheme



25

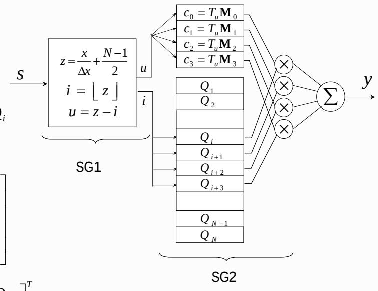
Flexible spline neuron scheme

$$y = \bar{h}(u,i) = \mathbf{T}_u \cdot \mathbf{M} \cdot \mathbf{Q}_i$$

$$\mathbf{T}_u = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix}$$

$$\mathbf{M} = \frac{1}{2}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}$$

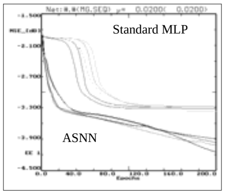
$$\mathbf{Q}_i = \begin{bmatrix} Q_i & Q_{i+1} & Q_{i+2} & Q_{i+3} \end{bmatrix}^T$$

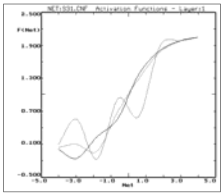


26

Adaptive Spline Neural Networks

⌘Fast performance: *suitable for many non-linear DSP applications*





27

Some Related References

On-Line Learning of DMLP

Paolo Campolucci, Aurelio Uncini, Francesco Piazza, Bhaskar D. Rao, "**On-Line Learning Algorithms for Locally Recurrent Neural Networks**", *IEEE Trans. on Neural Network*, Vol. 10, No. 2, pp.253-271 March 1999.

Paolo Campolucci, Aurelio Uncini and Francesco Piazza "**A Signal-Flow-Graph Approach to On-line Gradient Calculation**", 2000 Massachusetts Institute of Technology *Neural Computation,* Vol. 12, pp. 1901–1927, August 2000.

Adaptive Spline Neural Networks

Lorenzo Vecci, Francesco Piazza, Aurelio Uncini, "**Learning and Approximation Capabilities of Adaptive Spline Activation Function Neural Networks**", *Neural Networks*, Vol.11, No.2, pp 259-270, March 1998.

A. Uncini, L. Vecci, P. Campolucci, F. Piazza, "**Complex-Valued Neural Networks with Adaptive Spline Activation Function for Digital Radio Links Nonlinear Equalization**", *IEEE Trans. on Signal Processing*, Vol. 47, No. 2, February 1999.

Stefano Guarnieri, Francesco Piazza and Aurelio Uncini, "**Multilayer Feedforward Networks with Adaptive Spline Activation Function**", *IEEE Trans. On Neural Network*, Vol. 10, No. 3, pp.672-683, May 1999.

Mirko Solazzi and Aurelio Uncini, "**Artificial Neural Network with Adaptive Multidimensional Spline Activation Functions**", IEEE-INNS-ENNS International Joint Conference on Neural Networks IJCNN2000, Como, Italy, 24-27 July 2000.

28

Some applications

⌘Low cost Loudspeakers linearisation by predistortion

⌘Speech Quality Enhancement

⌘Subbands Audio Signal Recovering using Neural Non-linear Prediction

⌘Non liner models for sound synthesis

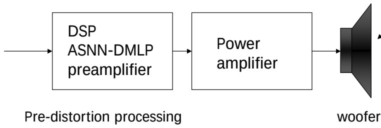
⌘Blind acoustic signal separation/deconvolution

29

Linearisation of a non-linear device

⌘Loudspeaker Predistortion

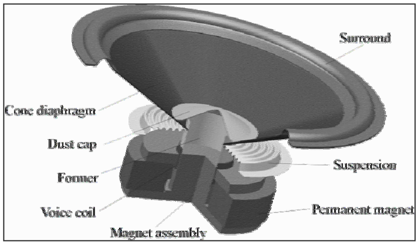
Loudspeaker: non linear dynamic device



ASNN - DMLP with buffers real-time implemented on ARIEL DSP96 (Motorola 96002) with ARIEL Proport 16 (Uncini, Piazza, BIAS-1997).

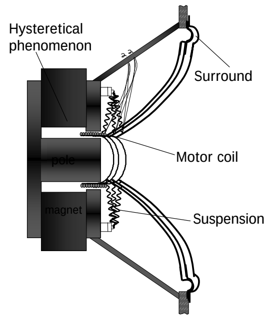
30

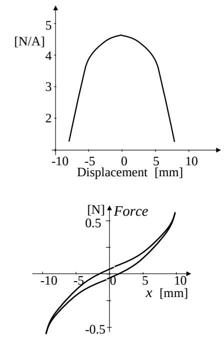
Loudspeaker scheme



3
1

Several ***electrical*** *and* ***mechanical*** *non linearity sources*



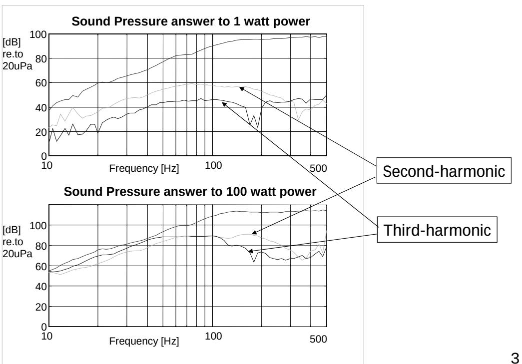


(hard) dynamic non-linear device

Small signal behaviour

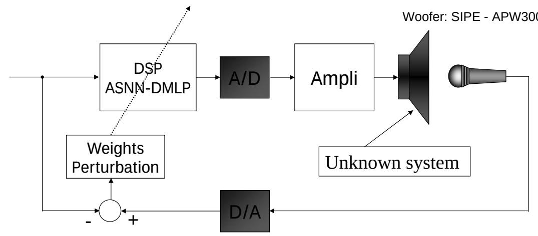
3
2

Frequency response Woofer SIPE - APW300



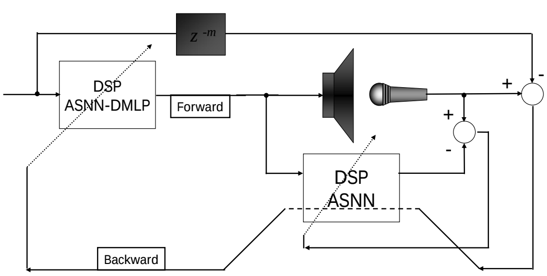
3
3

Predistortion learning scheme 1: *on-line*



3
4

Predistortion learning scheme 2: *on-line*

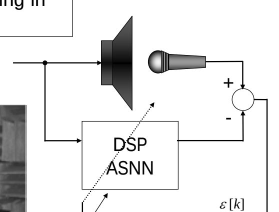


Loudspeaker NN model to performe the *on-line* Backpropagation Through Time

3
5

Learning scheme 2: *on-line*

Neural Network pre-training in anechoic chambre



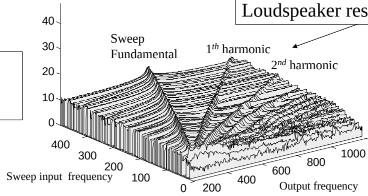
Loudspeaker Model
APW300 acquired in anechoic chambre

3
6

Learning scheme 2: *on-line*

Loudspeaker response

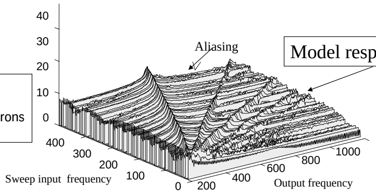
Harmonic Distortion Output to Sweep **Loudspeaker APW300**



Harmonic Distortion Output to Sweep **Neural Model**

ASNN architecture
- 10 input (FIR)
- 1 hidden 2 neurons
- 1 linear output

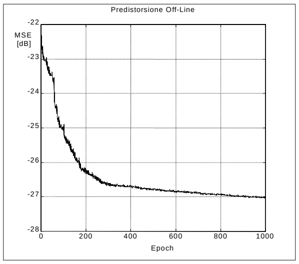
Model response



3 7

Predistortion Results

ASNN architecture
- 20 input (FIR)
- 1 hidden with 2 spline neurons
- 1 linear output



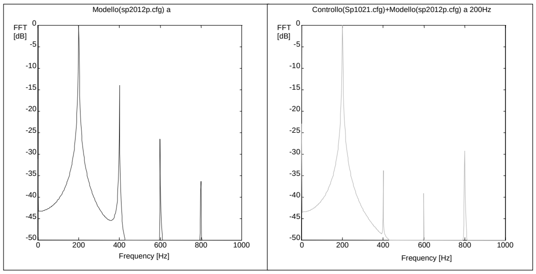
Harmonic distortion

Without predistorter
TDH=20.7 %

With predistorter
TDH=4.15 %
(14dB reduction at 200Hz)

3 8

Predistortion Results



About 20dB of 2nd and 3rd harmonic attenuation at 200 Hz

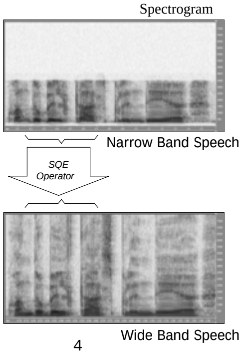
3 9

Speech Quality Enhancement

⌘ The problem is to **recover from narrow band signal** (telephonic quality) **the two missing frequency bands**

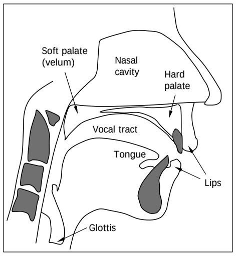
⌘ nominally from 20 Hz to 300 Hz and from 3400 Hz to 8000 Hz.

$$s_{NB}[n] \rightarrow s_{WB}[n]$$



4

Speech Quality Enhancement



⌘ *As supposed by several authors this should be made possible by the human speech production mechanism, which relates the frequency contents of different bands.*

We postulate the existence of a **non-linear operator** for speech enhancement called **Quality Enhancement Operator** (QEO) denoted as Ψ

4 1

Quality Enhancement Operator

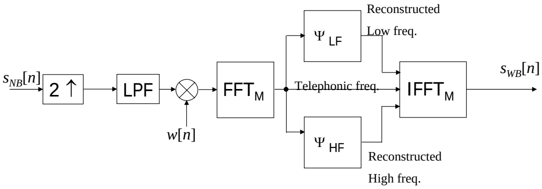
⌘ The proposed Ψ operator works in the **frequency domain** without an excitation signal and does not need any parameter tuning. Is defined as:

⌘ or in term of Short Time Fourier Transform (STFT):

⌘ Implementation of Ψ operator by Dynamic Complex domain Adaptive Spline Neural Network (D-CASNN)

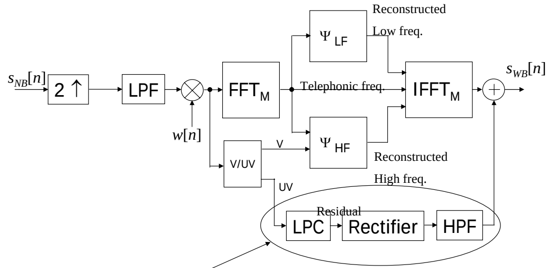
4 2

Proposed Speech Quality Enhancement *scheme 1*



4
3

Proposed Speech Quality Enhancement *scheme 2*



Additional scheme for the recovery of high frequency unvoiced sounds

4
4

Speech Quality Enhancement set-up

64 points Hanning windows with an overlap of 32 points;

64 points complex FFT/IFFT;

a DCASNN1 with 12 inputs and 2 outputs, with $\Delta x=1$ for each neuron for the recovery of the **low band**;

a DCASNN2 with 12 inputs and 19 outputs, with $\Delta x=1$ for the recovery of the **high band voiced sounds**;

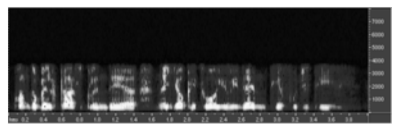
an LPC of order 8 for the recovery of the **high band unvoiced sounds** followed by a rectifier and a high-pass filter.

Very small networks implemented in standard low cost floating-point DSP processor (Texas TMSC30)

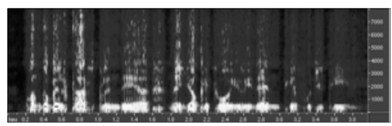
45

Preliminary results (working on)

[redacted]



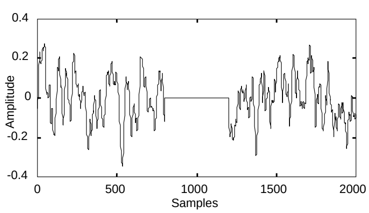
[redacted]



4
6

Subbands Audio Signal Recovering using Neural Non-linear Prediction

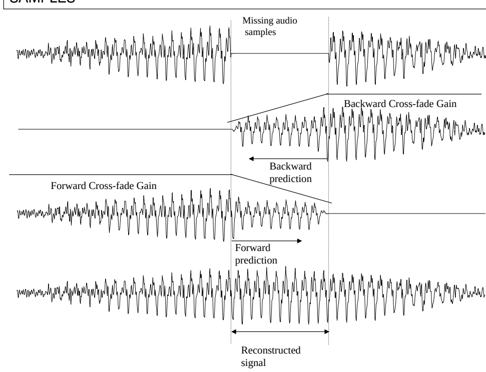
Audio signal recovering is a common problem in digital audio restoration field. The reconstruction of *L* consecutive missing samples in an audio signal may be considered an extrapolation problem



Ex. of 400 samples (8ms) of missing samples of music audio signal

4
7

METHOD: CROSS-FADE OF FORWARD AND BACKWARD PREDICTED SAMPLES



48

Subbands Multirate Approach

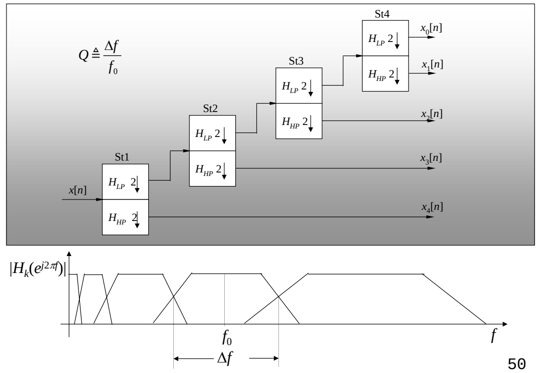
⌘ A multirate approach gives the advantage of a reduced sample rate, and so a decrease of samples number allowing a **longer gap reconstruction**.

⌘ A direct implementation of multirate technique is a filter bank which offers two different advantages

- The signal is split in several components. This is similar to a frequency analysis. The prediction is made easier by slower evolution of subband signal.
- The decimation process which reduces signal length, avoiding numeric problems.

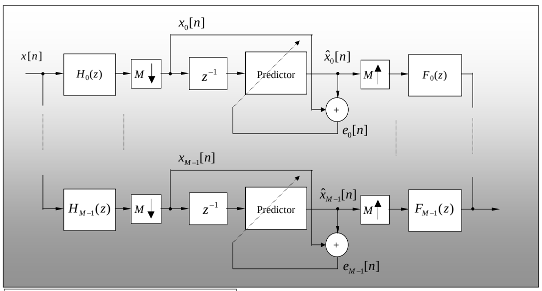
4
9

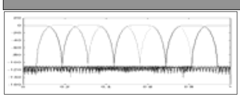
The Octave (Constant-Q) Subband Predictor



50

The uniform M-CHANNEL Subband Predictor



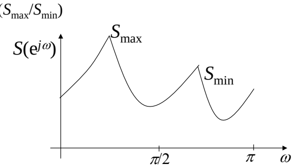


51

Convergence Properties of Subband Prediction Algorithm

⌘ The **stability region** for mean-square convergence of the LMS algorithm is given by $0<\mu<\lambda_{\max}$ (where $\lambda_{\max}$ is the maximum eigenvalue of the correlation matrix $\mathbf{R}$ of the input signal)

⌘ **The performances** of the adaptive system employing LMS-like algorithm is dependent on the **eigenvalues spread** :
$\chi(\mathbf{R}) = (\lambda_{\max}/\lambda_{\min}) \leq (S_{\max}/S_{\min})$



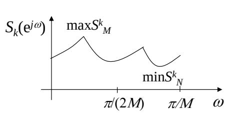
⌘ **The convergence do not take place *uniformly*** and the speed depends on the largest time constant, which is given by $\tau_{\max}=2/\mu\lambda_{\min}$

5
2

Convergence Properties of Subband Prediction Algorithm

⌘ For a subband system we have:

- $\max S^k_M \leq S_{\max}$
  $\min S^k_M \geq S_{\min}$
- $\max \lambda^k_M \leq \lambda_{\max}$
  $\min \lambda^k_M \geq \lambda_{\min}$



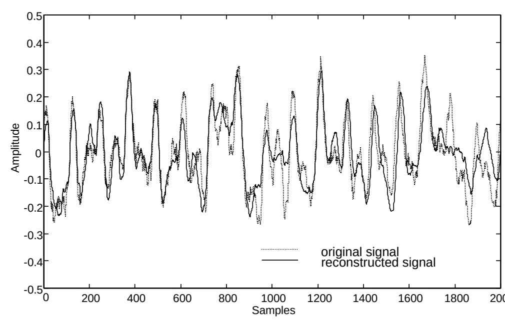
⌘ For a subband algorithm we have a **reduced eigenvalues spread:**
$\chi(\mathbf{R}^S) \leq \chi(\mathbf{R})$

⌘ The convergence is more *uniform*

Best and *fast* convergence performances

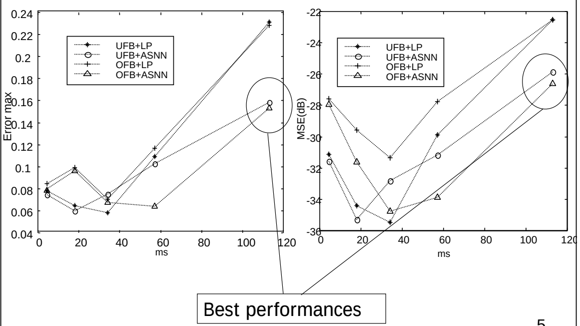
53

Example of reconstruction of 2000 samples (45ms) by octave filter bank and spline neural networks.



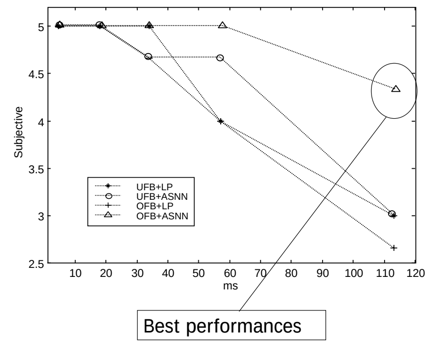
5
4

The MAXERR and MSE vs signal reconstructed length



55

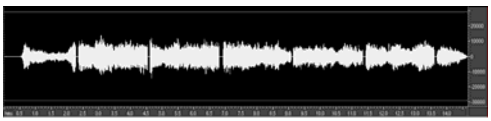
The subjective opinion vs signal reconstructed length



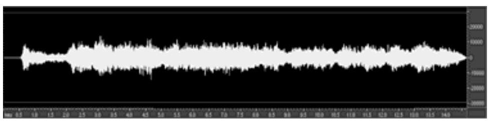
56

Experimental Results

Signal with missing samples



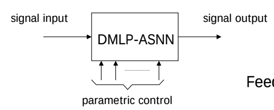
Signal with recovered samples



57

Non liner models for sound synthesis

- *Neural Networks can represent a generalization of several digital sound synthesis techniques*
- *Natural extension of non-linear distortion synthesis technique (dynamic non linearity)*

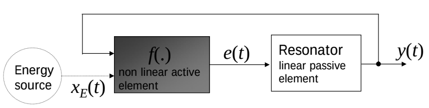


Feed forward scheme

58

E.g. non linear musical oscillator (lumped circuit model)

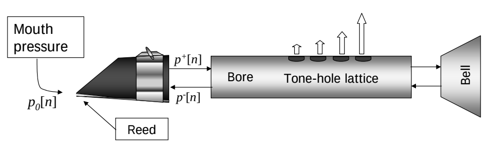
- Fairly idealized general musical oscillator
  - Non linear function + linear filter



- $e(t) = f(y(t)\ ,\ x_E(t))$ = excitation signal
- $x_E(t)$ = external control

59

Example of a single-reed instrument



REED -> External Exitation

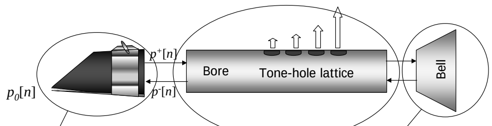
BORE -> Resonator

BELL -> Acoustic impedance adaptation

TONE-HOLE -> Fingers control

60

Single-reed instrument



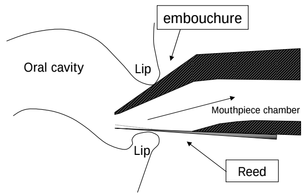
Complex Non Linear Exitation Mechanism (NLEM)

Simple linear filter: two-port Network (reflection req.>1.5KHz)

Resonator implemented by a Digital Delay Line

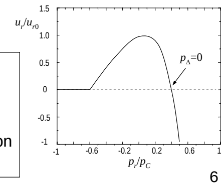
61

Single Reed - Non Linear Excitation Mechanism (NLEM)



The force over the reed is equal to $S_r p_\Delta$ where S_r is effective reed pressure surface, and $p_\Delta = p_{oc} - p_r$.

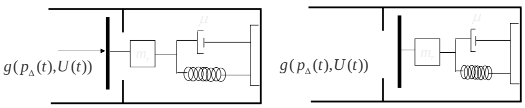
- ⌘ A vibrating reed is a pressure-controlled valve with two possible configuration: blown closed and blown open
- ⌘ The reed vibrations control the air flow on the embouchure



6

Single Reed Model

⌘ The reed can be considered as a **non-linear mechanical oscillator**

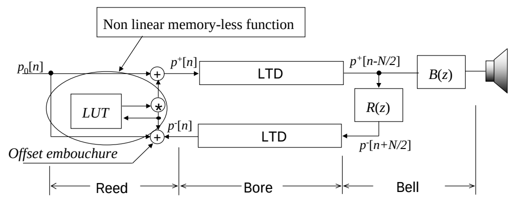


$$m_r\left[\frac{d^2x}{dt^2}+\mu\omega_r\frac{dx}{dt}+\omega_r^2(x-x_0)\right]=g(p_\Delta(t),U(t))$$

m_r represents the reed mass
μ is the dumping factor
ω_r is the mechanical resonance frequency
$U(t)$ is the flow pressure through the reed embouchure
$g(t)$ is a non linear function

63

Single-reed instrument *null-mass* **Physical Model**



$R(z)$ = Linear Filter
A_v = Vibrato ampl. (e.g. 0.03)
f_v = Vibrato freq. (e.g. 5Hz)

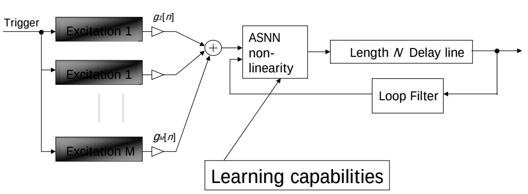
$B(z)$ = High-pass filter (ord. 1) 1.5KHz

[J.O. Smith 1993]

64

Proposal: more general framework

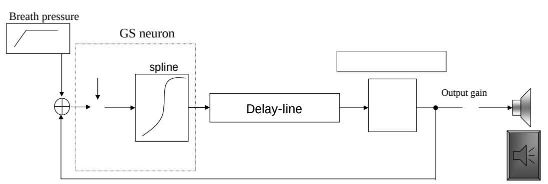
⌘ *General framework for* ***physical-like*** *model synthesis*



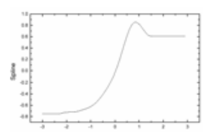
Learning capabilities

65

Single Reed *null-mass* physical-like model

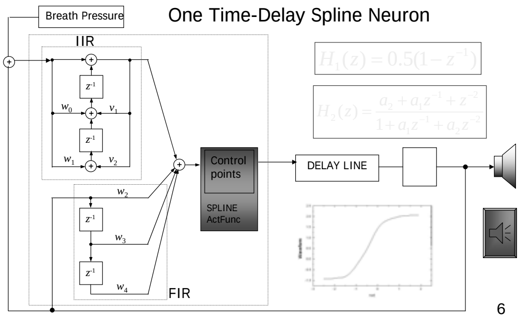


Learning of clarinet sound



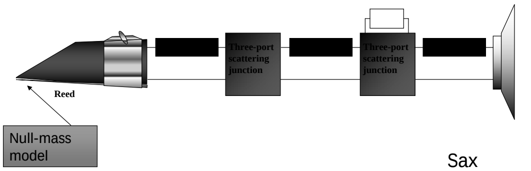
66

Single Reed physical-like model



67

Preliminary results: Sax sound



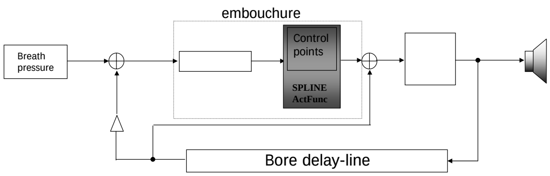
Sax

Learning of clarinet and sax sounds

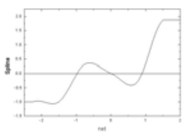
"Chaotic sound" behaviour

68

Preliminary results: Flute Sound

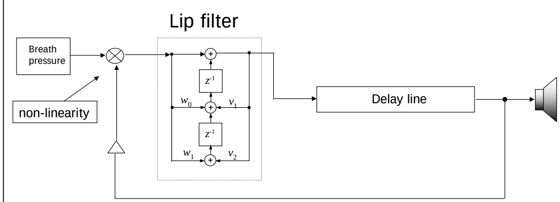


Adaptation of the only non linear function spline parameters



69

Preliminary results: Trumpet sound

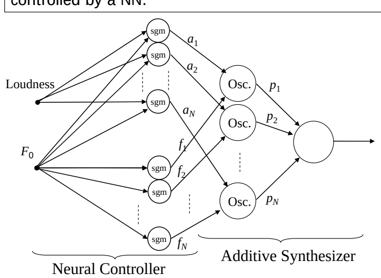


Adaptable parameters
$\alpha \quad w_0 \quad w_1 \quad \upsilon_1 \quad \upsilon_2$

70

Another approach: additive synthesis controlled by ANN[1]

The amplitudes and frequencies (a_k, f_k) of the oscillators are controlled by a NN.



Solo sax

Solo oct2

AOEUI

Suling

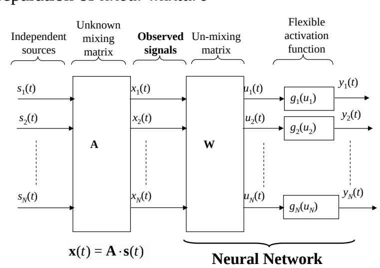
Note: reproduced sounds by record

[1] D. Wessel, C. Drame, M. Wright, CNMAT, Berkeley

71

Audio signal separation

Blind separation of *linear mixture*

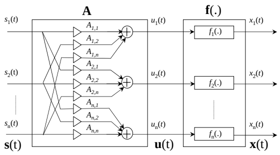


Problem: estimate the un-mixing matrix **W** such that $\mathbf{u}(t) = \mathbf{s}(t)$ (several algorithms exist).

72

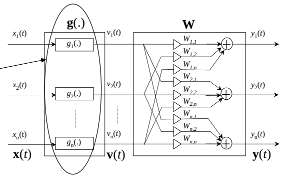
Preliminary: signal separation of post non-linear mixture

post non-linear mixture model



$$x_i(t) = f_i\left(\sum_{j=1}^{N} a_{i,j}\, s_j(t)\right) \quad i = 1, 2, \ldots M$$

Non linear blind separation structure

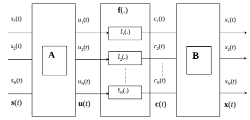


Flexible spline activation functions

7
3

Preliminary: signal separation of non-linear mixing

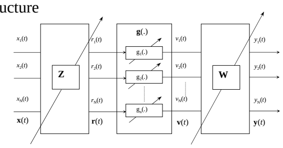
Separation of non-linear mixture



Non linear mixing model

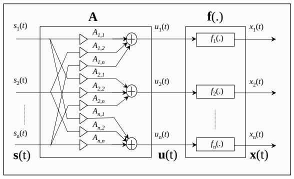
$$x_k(t) = \sum_{i=1}^{N} b_{k,i}\, f_i\left(\sum_{j=1}^{N} a_{i,j}\, s_j(t)\right) \quad k = 1, 2, \ldots M$$

Non linear blind separation structure



7
4

Preliminary Results: signal separation of post non-linear mixing

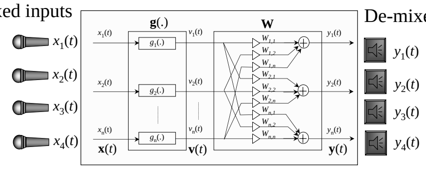


$$\mathbf{A} = \begin{bmatrix} 0.2811 & 0.2926 & -0.1364 & 0.4080 \\ -0.4608 & -0.1232 & -0.2025 & 0.2492 \\ 0.3908 & -0.3373 & -0.3908 & -0.2917 \\ -0.4621 & -0.4660 & -0.4438 & -0.3846 \end{bmatrix}$$

$$x_1(t) = \tanh(u_1(t))$$
$$x_2(t) = \tanh(u_2(t))$$
$$x_3(t) = 0.8u_3(t) + u_3^3(t)$$
$$x_4(t) = \tanh(0.9u_4(t))$$

Mixed inputs

De-mixed outputs



7
5

Some conclusions

- ⌘ Dynamic Neural Networks represent a new class of non-liner DSP algorithms
- ⌘ Fast architectures allow real time applications at high throughput rate
- ⌘ Learning algorithms ensure consistent design methods
- ⌘ A huge application fields in audio/sound processing

⌘ Neural Networks: general framework for non linear digital sound analysis/synthesis methods

7
6