JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

IIR ADAPTIVE FILTERING

IIR filters have one big advantage with respect to FIR designs. They are much more efficient, i.e. a given frequency domain characteristic is obtained with a much smaller filter order.

One of the problems with FIR filters is that the length of the impulse response is coupled to the filter order.

Example:

Suppose we would like to identify a plant that can be modeled by a 3rd order system, but has an impulse response that extends to 100 samples. (The extension is related to the location of the poles and zeros of the plant).

With an FIR design we will have to choose a 100 order filter for perfect identification (an order larger than 3 for a reasonable performance).

When doing so we create several problems for the identification:

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA

904-392-2662
principe@brain.ee.ufl.edu

1- If the signal is noisy, since the FIR has a large number of degrees of freedom, we will be adapting to noise.

2- The convergence process becomes much slower, because we are searching a high dimensionality surface, and the shape of the performance surface may have slowly rolling slopes and steep ridges, which slow down the search.

It is obvious that if we could use an IIR filter (of order 3 for the example), the problem would become one of choosing the appropriate pole/zero locations.

Problem with IIR adaptation

1- During adaptation poles may move outside the unit circle.

2- Performance surfaces are generally nonquadratic

3- Gradient search algorithms become computationally more complex.

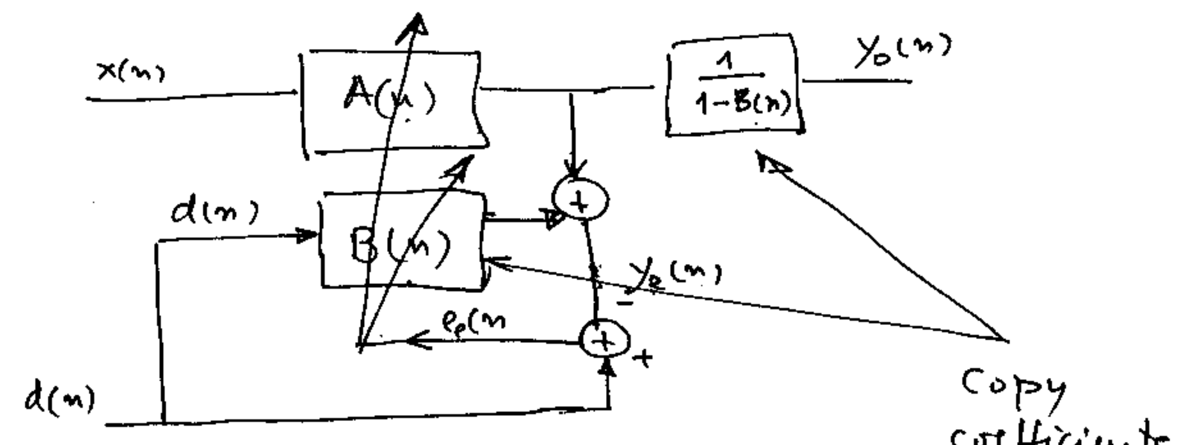
JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

Recursive Adaptive Filters- equation error formulation

Consider the nonrecursive equation

$$y_k = \sum_{n=0}^{L} a_n x_{k-n} + \sum_{n=1}^{L} b_n d_{k-n}$$

It is a two input, single output filter, that does not have feed-back. The output is a LINEAR function of the coefficients.



$$\begin{cases} A(n) = \sum_{n=0}^{L} a_n z^{-n} \\ B(n) = \sum_{n=1}^{L} b_n z^{-n} \end{cases}$$

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

The coefficients of A(n) will be copied to the denominator of the block at left. This makes the equation error formulation IIR.

The appeal of this technique is that the error defined by

$$e_e(n) = d(n) - y_e(n)$$

is also a linear function of the coefficients. Therefore the MSE is a quadratic function with a single minima just like the FIR case. We can expect similar convergence properties.

Stability is easily checked by monitoring the zeros of 1-A(z). If not, projection of the roots to the inside of the unit circle is performed (mirror position method is often utilized).

If we define the vectors

$$U_k = \left[x_k, x_{k-1}, \ldots, x_{k-L}, d_{k-1}, \ldots d_{k-L} \right]^T$$

$$W_k = \left[a_{0k}, a_{1k}, \ldots, a_{Lk}\ b_{1k}, \ldots\ b_{Lk} \right]^T$$

Then the LMS formulation follows.

$$\varepsilon_k = d_k - W_k^T U_k$$

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

$$\hat{\nabla}_k = \frac{\partial \varepsilon^2}{\partial w} = -2\,\varepsilon_k U_k$$

So

$$\vec{W}_{k+1} = \vec{W}_k + 2\mu\,\varepsilon_k \vec{U}_k$$

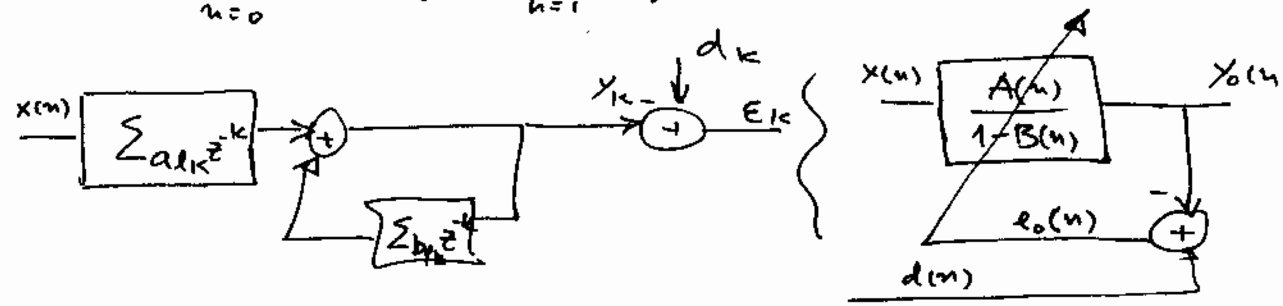
But RLS can also be utilized, as well as others (maximum likelihood, maximum a posteriori.

The major problem with this approach is that the solution may be BIASED. It can be shown that only when A(n) is zero the bias will be zero. The bias comes from the fact that the poles are being estimated in a all -zero form where the eventual input noise plays also a role in the minimization.

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

RECURSIVE ADAPTIVE FILTER- Output error

$$y_k = \sum_{n=0}^{L} a_n x_{k-n} + \sum_{n=1}^{L} b_n y_{k-n}$$



Define new vectors W and U

$$W_k = \left[a_{0k}, a_{1k}, \dots a_{Lk}, b_{1k}, \dots b_{Lk} \right]^T$$

$$U_k = \left[x_k, x_{k-1}, \dots x_{k-L}, y_{k-1}, \dots y_{k-L} \right]^T$$

Notice that U now contains past values of y_k. The output becomes a nonlinear function of the coefficients (sometimes the output of the filter is called pseudo linear regression). Notice that the equation error is a filtered version of the output error. Both methods coincide if A(n).

$$e_e(n) = \left[1 - A(n) \right] e_o(n)$$

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

Due to the fact that y_n is a nonlinear function of the weights, the error defined as $\varepsilon_k = d_k - y_k = d_k - W_k^T U_k$

will also be a nonlinear function of the weights. The error function is NOT quadratic in the weights. It may contain local minima, which means that gradient descent procedures are not guaranteed to converge (search may be caught in local minima).

However the output error formulation is UNBIASED.

It has been shown that for system identification if the adaptive filter has enough degrees of freedom, the input noise is white and the order of the numerator exceeds the order of the plant denominator, there will be NO local minima.

The search may be dependent in the initial conditions. The convergence is much slower than the equation formulation.

It is believed that the error formualtion is THE approach for IIR adaptive filtering.

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

LMS GRADIENT (recursive prediction error)

$$\hat{\nabla}_k = \frac{\partial \varepsilon^2}{\partial W_k} = 2\varepsilon \frac{\partial \varepsilon}{\partial W_k} =$$

$$= -2\varepsilon_k \left[\frac{\partial y_k}{\partial a_{0k}}, \dots \frac{\partial y_k}{\partial a_{Lk}}, \frac{\partial y_k}{\partial b_{1k}}, \dots \frac{\partial y_k}{\partial b_{Lk}} \right]^T$$

Problem now is to compute the derivatives. Using (1)

$$\alpha_{n,k} \triangleq \frac{\partial y_k}{\partial a_n} = x_{k-n} + \sum_{l=1}^{L} b_l \frac{\partial y_{k-l}}{\partial a_n}$$

$$\simeq x_{k-n} + \sum_{l=1}^{L} b_l \alpha_{n,k-l}$$

$$\beta_{n,k} \triangleq \frac{\partial y_k}{\partial b_n} \simeq y_{k-n} + \sum_{l=1}^{L} b_l \beta_{n,k-l}$$

Notice that these are now ORDERED derivatives (i.e. to compute the component for order l, must first compute for order l-1). There is an approximation involved

$$\frac{\partial y_{k-l}}{\partial a_n} \simeq \frac{\partial y_{k-l}}{\partial a_{n-l}} \implies$$ DERIVATIVES CAN BE MADE RECURSIVE.

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

The gradient becomes

$$\hat{\nabla}_k = -2\,\varepsilon_k \left[\alpha_{0k}, \ldots \alpha_{Lk}, \beta_{1k}, \ldots \beta_{Lk} \right]^T$$

$$W_{k+1} = W_k - M\,\hat{\nabla}_k$$

Notice now that the computation of the gradient is no longer O(L). For each term α_{nk} and β_{nk} we need L multiplications. Since we have L elements the computation of the gradient is $O(L^2)$.

Of course if we approximate the gradient by the first terms, then the computation is again O(L). This sometimes is called the <u>approximate gradient or simplified RPE</u>.

Then we can write,

$$W_{k+1} = W_k - M\,\hat{\nabla}_k$$

where M is a diagonal matrix of step sizes, one for each a and b weights.

$$M = \begin{bmatrix} \mu_1 & \cdots & & & 0 \\ & & \mu_L \nu_1, & & \\ 0 & & & \cdots & \nu_L \end{bmatrix}$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

IIR LMS

$$\begin{cases} y_k = W_k^T U_k \\ \alpha_{n,k} = x_{k-n} + \sum_{l=1}^{L} b_{lk}\, \alpha_{n,k-l} & 0 \le n \le L \\ \beta_{n,k} = y_{k-n} + \sum_{l=1}^{L} b_{lk}\, \beta_{n,k-l} & 1 \le n \le L \\ \hat{\nabla}_k = -2\, \varepsilon_k \left[\alpha_{0k}, \ldots \alpha_{Lk}, \beta_{1k}, \ldots \beta_{Lk} \right]^T \\ W_{k+1} = W_k - \mu\, \hat{\nabla}_k \end{cases}$$

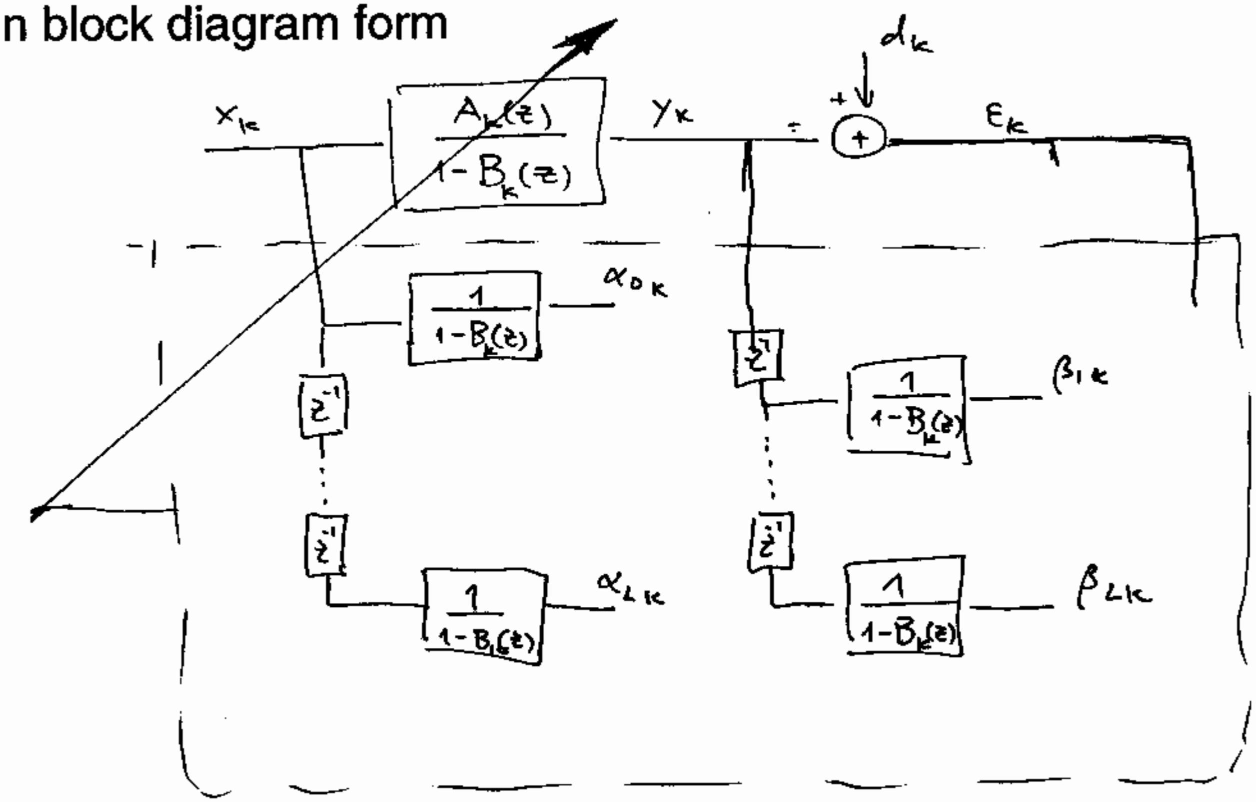
The calculation of the gradient terms can be put into a block diagram form, if we write it in the Z domain (x, y inputs, α, β outputs).

$$B_k(z) = \sum_{l=1}^{L} b_{lk} z^{-l}$$

$$y_k \longrightarrow \boxed{z^{-n}} \longrightarrow \boxed{\frac{1}{1-B_k(z)}} \longrightarrow \beta_{n,k}$$

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

In block diagram form



$$w_{k+1} = w_k + M\left[\frac{1}{1-B_k(z)}\right] U_k \varepsilon_k$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

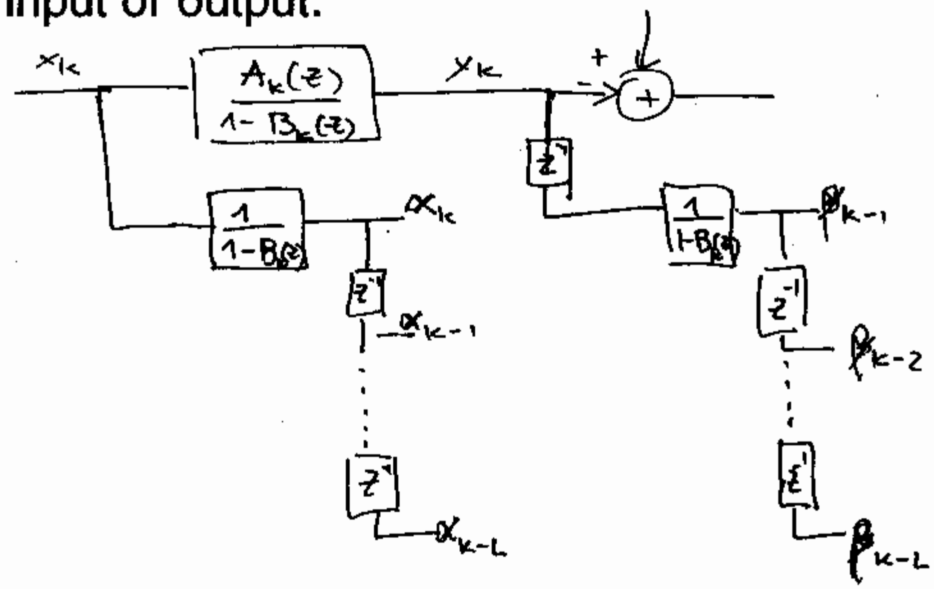
principe@brain.ee.ufl.edu

Simplified RPE algorithms

Just approximate the gradient by the first terms

$$\begin{cases} \alpha_{n,k} \simeq x_{k-n} \\ \beta_{n,k} \simeq y_{k-n} \end{cases}$$

Then each component of the gradient is just the delayed version of the input or output.



This is normally done in practice, with small degradation.

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA

904-392-2662
principe@brain.ee.ufl.edu

SHARF (simplest hyperstable adaptive recursive filter)

The idea of HARF methods is to smooth the error ε_k to guarantee convergence. SHARF is a special case where the truncated gradient is used.

The gradient can be approximated by the signal vector U. This is also sometimes called pseudolinear regression because the output is still a nonlinear function of the coefficients but the gradient ignores dependence on the coefficients.

$$W_{k+1} = W_k + M\, U_k\, \varepsilon_k$$

This algorithm becomes very similar to the RLS implementation of the equation error formulation.

The problem is that the algorithm may not converge, unless the denominator polynomial is strictly positive. Therefore, one normally filter the error to ensure this condition.

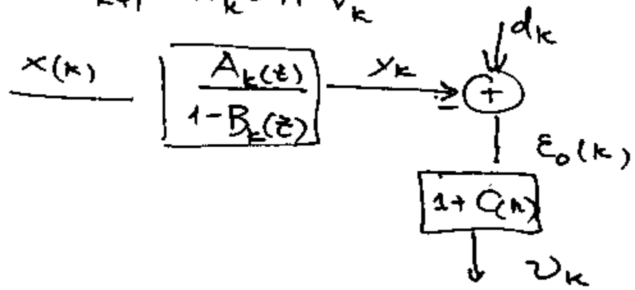
$$R_e\left(\frac{1+C(z)}{1-B^*(z)}\right) - \frac{1}{2} > 0$$

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

Filtered error algorithm or SHARF

$$\left\{\begin{array}{l} y_k = W_k^T U_k \\ \varepsilon_k = d_k - y_k \\ v_k = \varepsilon_k + \sum_{n=1}^{N} c_n \varepsilon_{k-n} \\ \hat{\nabla}_k = -2 v_k \left[x_k, \dots x_{k-L}, y_{k-1}, \dots y_{k-L} \right]^T \end{array}\right.$$

$$W_{k+1} = W_k - \mu \hat{\nabla}_k$$



The problem is to set c_n in general. The filter should be time varying but coefficients are normally set to constant values. Convergence has only been proven for special cases. Algorithm has been utilized in echo cancelling and noise cancelling.

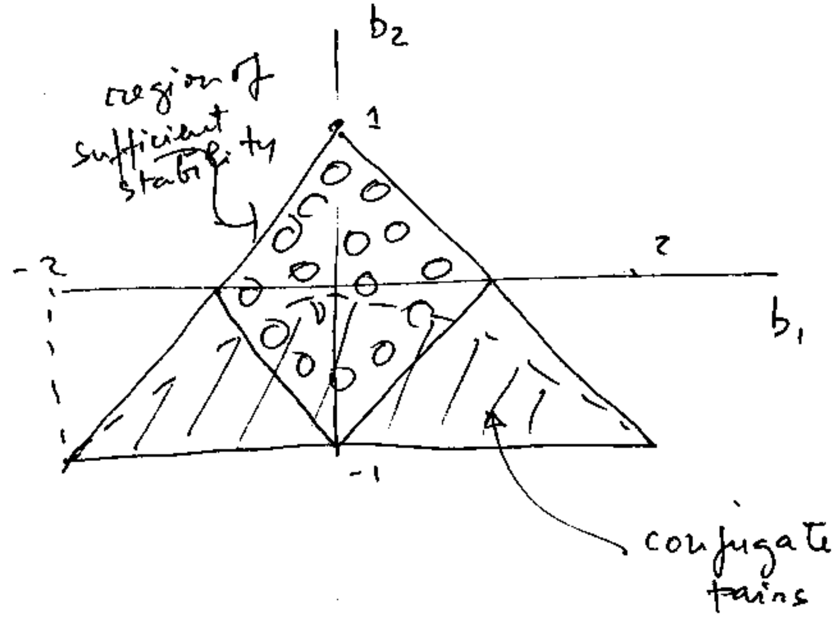
JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

Stability Monitoring

If poles go outside the unit circle (due to the noisy gradient) the filter becomes unstable. The stability triangle is usually utilized for <u>2nd order structures</u>.

Instability can be monitored by testing if the sum of the denominator coefficients are less than 1. Jury's test is less restrictive, but more complex. Any method does not tell which coefficient causes the problem (must factor the polynomial).

Parallel structrues may be the answer, or lattice.



$|b_1| + |b_2| < 1$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

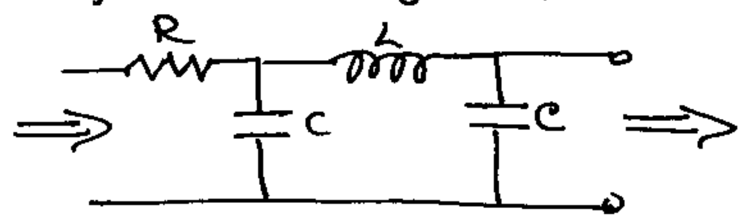
EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

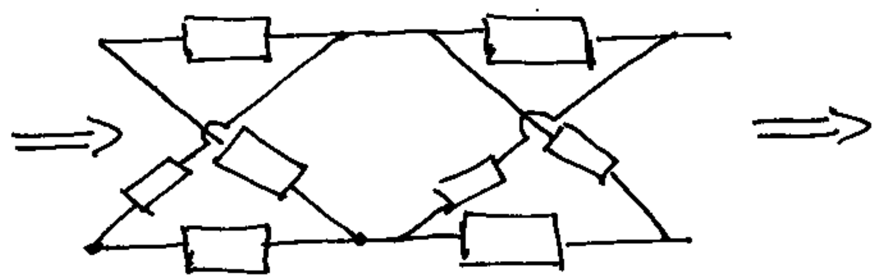
LATTICE STRUCTURES

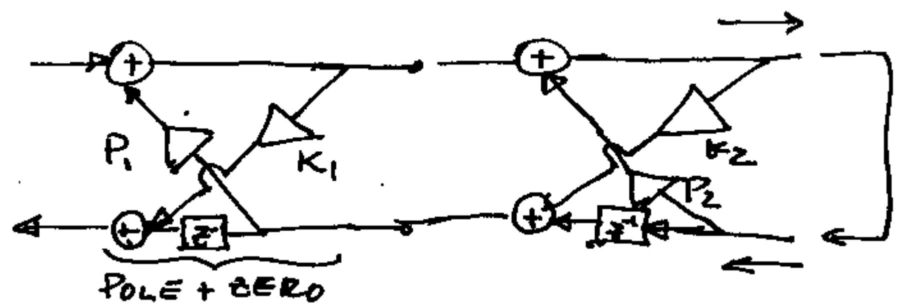
Are motivated by similar analog structures called ladder networks



which are known to have very good properties (insensitive to parameter drift). They require more hardware to realize a given transfer function.

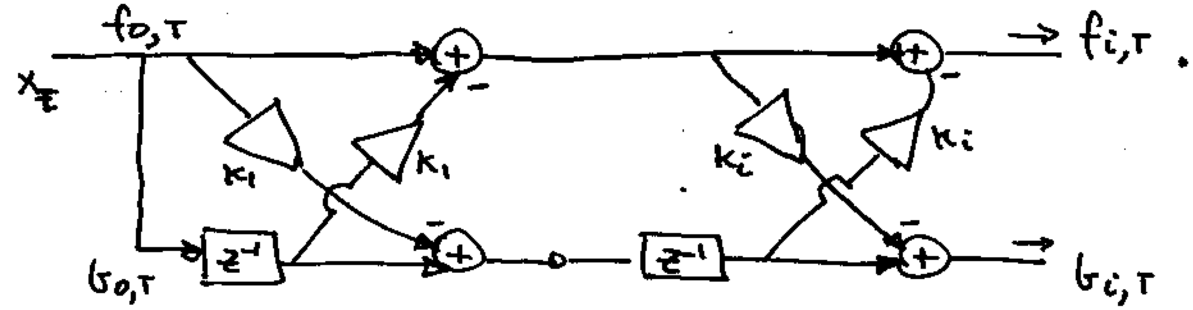
This can be translated into the general ladder network.





JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

For symmetric networks k≅p.We will restrict our attention to feedfoward lattice structures. Theblock diagram becomes



We can see that these structures propagate a forward and backward signal

$$\begin{cases} f_{j+1}(t) = f_j(t) - k_{j+1}\, b_j(t-1) \\ b_{j+1}(t) = b_j(t-1) - k_{j+1}\, f_j(t) \end{cases}$$

k_j are known as the partial correlation coefficients (PARCOR) or reflection coefficients. We can go from a direct structure to a lattice structure (book pp 169).

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA

904-392-2662
principe@brain.ee.ufl.edu

PROPERTIES

- Lattice structures require more operations.
- They are stable if the k_j are less than 1.
- They produce a stage-by-stage orthogonalization of the input signal.
- They model wave propagation in stratified medium.

Most important characteristic for adaptive signal processing is the step by step orthogonalization of the input.The $b_j(t)$ corresponds to a Gram-Schmidt orthogonalization of the delayed version of the input signal.

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EFL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

WHY?

In adaptation using LMS, we are adapting at each step. However the error of the gradient estimate may be correlated near adaptation and produce rattling.

In the lattice this does not happen because we are adapting (changing the k) at each stage with the information locally available. So the residual error that is propagated through the structure is uncoupled (uncorrelated) with the input and previous input stages.

In the case of the lattice structure the mse is minimized stage-by-stage because k_j depends on quantities that are orthogonal between stages.

This does not happen with the linear combiner where the error is

$$\varepsilon_k = d_k - \sum_i w_i x_{k-i}$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

904-392-2662

principe@brain.ee.ufl.edu

COMPUTATION OF THE PREDICTOR COEFFICIENTS

For a predictor of order p, the data sample at time t, i.e. the next x(t) is approximated by a linear combination of the p previous samples (forward prediction) x(t-1),...x(t-p).

$$f_p(t) = \sum_{i=0}^{p} a_i \, x(t-i) \qquad a_0 = 1$$

To minimize the error, the coefficients are such that the error is orthogonal to the data,i.e.

$$E\left[f_p(t) \cdot x(t-j) \right] = 0 \qquad 0 \leq j \leq p$$

A backward predicition $b_p(t-1)$ will similarly predict x(t-p-1) using the samples x(t-1) ....x(t-p).

$$b_p(t-1) = \sum_{i=p-1}^{0} c_i \, x(t-i) \qquad c_{p-1} = 1$$

Increasing the prediction to order p+1 will make x(t) dependent upon x(t-1),.....x(t-p) and x(t-p-1). Notice however that the NEW information x(t-p-1) is contained in $b_p(t-1)$.

$$E\left[b_p(t-1) \cdot x(t-j) \right] = 0 \qquad 1 \leq j \leq p$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

Therefore using the recursion for the lattice,

$$f_{p+1}(t) = f_p(t) - K^f_{p+1} b_p(t-1)$$

where k_{p+1}^f must be determined satisfying the orthogonality condition of the error.

$$E\left[f_{p+1}(t).\, x(t-j)\right] = 0 \qquad 1 \le j \le p+1$$

The only constraint not immediatly satisfied involves x(t-p-1). Writing it as a function of the recursion

$$E\left[f_{p+1}(t).\, x(t-p-1)\right] = 0 \Rightarrow E\left[f_p(t).x(t-p-1)\right] - K^f_{p+1} E\left[b_p(t-1).\, x(t-p-1)\right] = 0$$

and noting that

$$E\left[b_p(t-1).x(t-p-1)\right] = E\left[\left(\sum_{i=p-1}^{0} c_i\, x(t-i)\right).x(t-p-1)\right] = E\left[X(t-p-1)X(t-p-1)\right]$$

then we have, noting that x(t-p-1) is the predicted b_p(t-1).

$$E\left[f_p(t).\, b_p(t-1)\right] = K^f_{p+1} E\left[b_p^2(t-1)\right]$$

Similarly,

$$K^b_{p+1} = \frac{E\left[f_p(t).\, b_p(t-1)\right]}{E\left[f_p^2(t)\right]}$$

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

Extending the prediction to order p+2 requires calculation of 2 more prediction terms f_{p+1} and b_{p+1}. We are constructing the optimization step by step.. When signal is stationary, k_p^f and k_p^b are equal (the final k is a combination of k_p^f and k_p^b).

In the tapped delay line, the w_i would ALL change if the order of the filter was changed.

We can also see why the k are called partial correlation coefficients. The correlation between x(t) and x(t-p-1) after their linear dependence is removed is $E(f_p(t)\ b_p(t))$. When this quantity is normalized by the variance of f_p and b_p, we get the pth order partial correlation. This is exactly the expression for k_p^f and k_p^b.

The backward prediction error are often used as a Gram-Schmidt orthogonalization of the delayed versions of the input.

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

SAMPLE DATA ESTIMATION OF THE REFLECTION COEFFICIENTS

Assume stationarity so,

$$\begin{cases} f_{j+1}(t) = f_j(t) - k_{j+1} b_j(t-1) \\ b_{j+1}(t) = b_j(t-1) - k_{j+1} f_j(t) \end{cases}$$

BLOCK ESTIMATES

One of the first estimates was proposed by Itakura and Saito and follows closely the Durbin algorithm of the BLS. It is the normalized conditional correlation coefficient between x(t) and x(t-j-1) given the intervening samples

$$k_{j+1} = \frac{\sum_{t=1}^{T} f_j(t) b_j(t-1)}{\sqrt{\sum_{t=1}^{T} f_j^2(t) \cdot \sum_{t=1}^{T} b_j^2(t-1)}}$$

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

Another very well known estimate is the harmonic mean of k and k (Burg maximum entropy). It is much simpler to compute.

$$K_{j+1}^{B} = \frac{\sum_{t=1}^{T} f_j(t) \cdot b_j(t-1)}{\frac{1}{2} \sum_{t=1}^{T} \left(f_j^2(t) + b_j^2(t-1) \right)}$$

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

GRADIENT ESTIMATES

Here only the prediction error at the preceding time instant is needed for the gradient. The simplest algorithm uses the forward and backward prediction errors weighted by a constant (Griffiths)

$$k_j(t+1) = k_j(t) + \alpha \{ f_j(t)\, b_{j-1}(t-1) + f_{j-1}(t) \cdot b_j(t) \}$$

This estimate can be improved if alpha is normalized by the power (accumulated square of f and b). This correspondds to the LMS algorithm.

The book (Ch 8, 173) uses a different notation

$$f \to s \qquad t \to k$$
$$b \to s' \qquad j \to \ell$$

Also coefficients start at 0 not at 1.

$$\left\{ \begin{aligned} s_{\ell+1,k} &= s_{\ell k} + k_\ell\, s'_{\ell,k-1} \\ s'_{\ell+1,k} &= k_\ell\, s_{\ell k} + s'_{\ell,k-1} \end{aligned} \right. \iff \left\{ \begin{aligned} f_{j+1}(t) &= f_j(t) - k_{j+1} b_j(t-1) \\ b_{j+1}(t) &= b_j(t-1) - k_{j+1} f_j(t) \end{aligned} \right.$$

904-392-2662
principe@brain.ee.ufl.edu

The book defines the statistics of the forward/backward prediction errors as

$$\begin{cases} \phi_\ell(n) = E\left[s_{\ell k} s_{\ell, k+n}\right] \\ \phi'_\ell(n) = E\left[s'_{\ell k} s_{\ell, k+n}\right] \\ \phi''_\ell(n) = E\left[s'_{\ell k} s'_{\ell, k+n}\right] \end{cases}$$

With this notation k is given by

$$K^b_{p+1} = \frac{E\left[f_p(t) . b_p(t-1)\right]}{E\left[b_p^2(t-1)\right]} \equiv \frac{\phi'_p(1)}{\phi''_p(0)} = \frac{\phi'_p(1)}{\phi_p(0)}$$

(Compare with 8.100). Now if we take the LMS gradient estimate, and use the steepest descent formalism,

$$\frac{\partial \widehat{\phi_{\ell+1}(0)}}{\partial k_\ell} = \frac{\partial s^2_{\ell+1, k}}{\partial k_\ell}$$

$$= 2 s_{\ell+1, k} s'_{\ell, k-1}$$

Notice that this the Griffiths equation with alpha substituted by --2μ.

$$k_{\ell, k+1} = k_{\ell k} - 2\mu_\ell s_{\ell+1, k} s'_{\ell, k-1}$$

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA

904-392-2662
principe@brain.ee.ufl.edu

The convergence of the lattice filters is much faster than the direct II structures (tapped delay line). The lattice orthogonalizes the input signal to each stage, so coefficient estimates are uncoupled, which implies not very sensitive to eigenvalue spread.

(see book).

The SER algorithm can also be applied to these structures.

The Generalized Feedforward Model (1)

A General Memory Model: The Convolution neural model

$$x(n) = \sigma\left(\sum_{m=0}^{n} w(n-m)\,x(m)\right) + I(n) \; ,$$

where $w(n) = 0$ for $n < 0$.

Consider the memory mechanism

$$net(n) \equiv \sum_{m=0}^{n} w(n-m)\,x(m) \, .$$

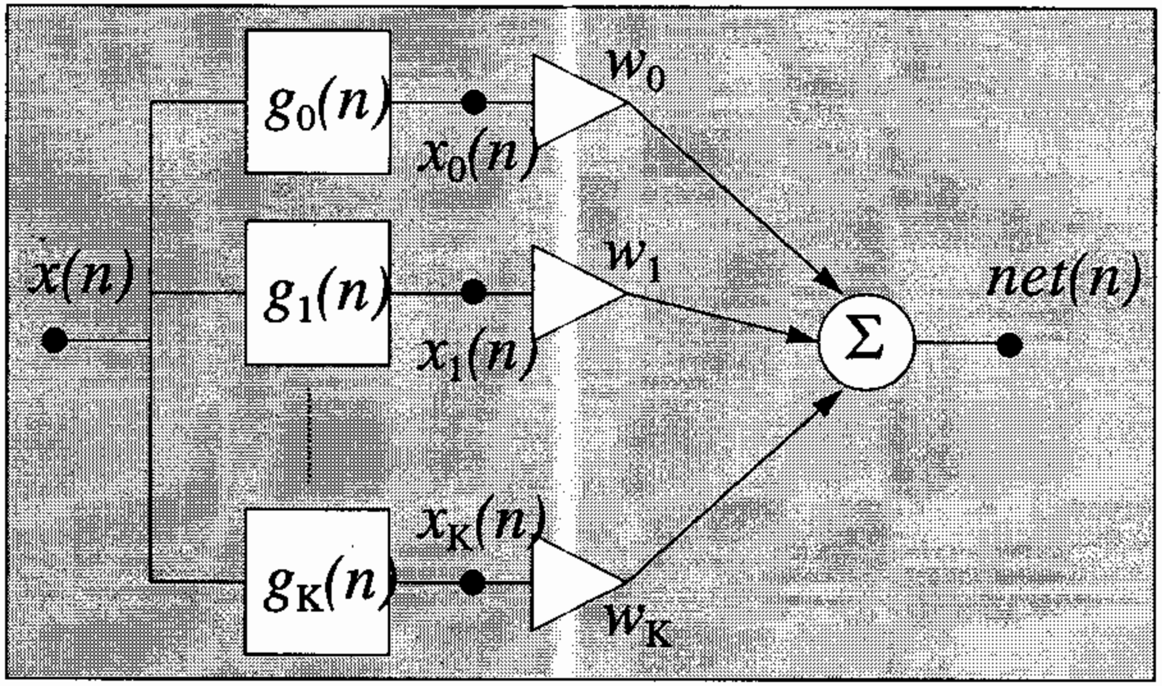
Dimensionality grows linearly with time.

The Generalized Feedforward Model (2)

Simplify by decomposition assumption (Wiener, 1949!)-

$$w(n) = \sum_{1}^{K} w_k g_k(n). \tag{1}$$

This system has fixed (K) dimensionality,but generation of $g_k(n)$ poses problems.



The Generalized Feedforward Model (3)

Fundamental Assumption:

Assume an additive network structure for the generation of $g_k(n)$.

In this study, recursive generation by

$$G_k(z) = G(z)\,G_{k-1}(z), \qquad (2)$$

where $G_k(z) \equiv Z\{g_k(n)\}$.

The Decomposition and Recursive Generation assumption reduces the convolution model to a new structure -

The Generalized Feedforward Filter (GFF)

Linear Memory Filter

Definitions.

A sequence $g(t)$ is the impulse response of a memory filter if the following two conditions hold:

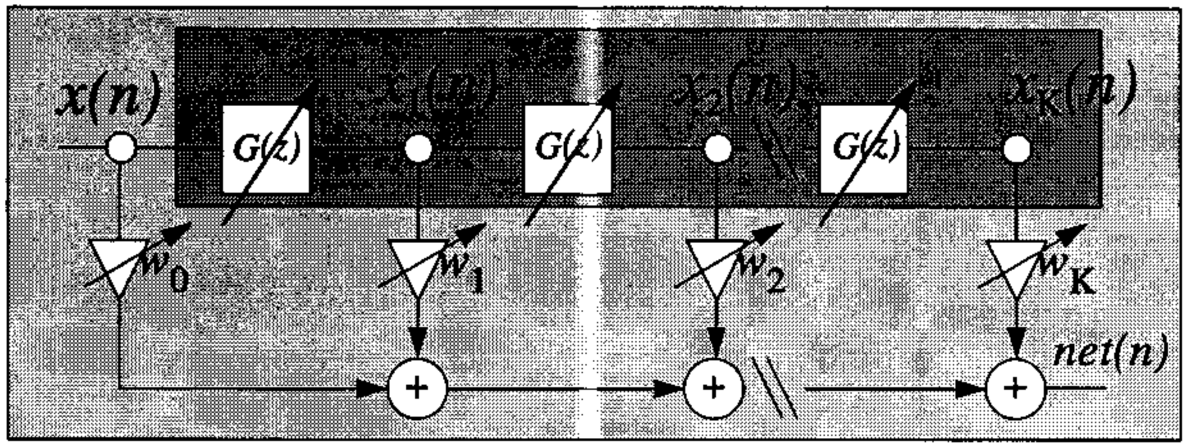
- $g(t)$ is causal, that is, $g(t)=0$ for $t < 0$.
- $g(t)$ is normalized such that $\sum_{t=0}^{\infty} |g(t)| = 1$.

Memory depth: The center of mass of the last tap.

Memory Resolution: The number of taps per unit time.

A memory filter is BIBO stable. ($\sum_{t=0}^{\infty} g(t) < \infty$)

The Generalized Feedforward Filter

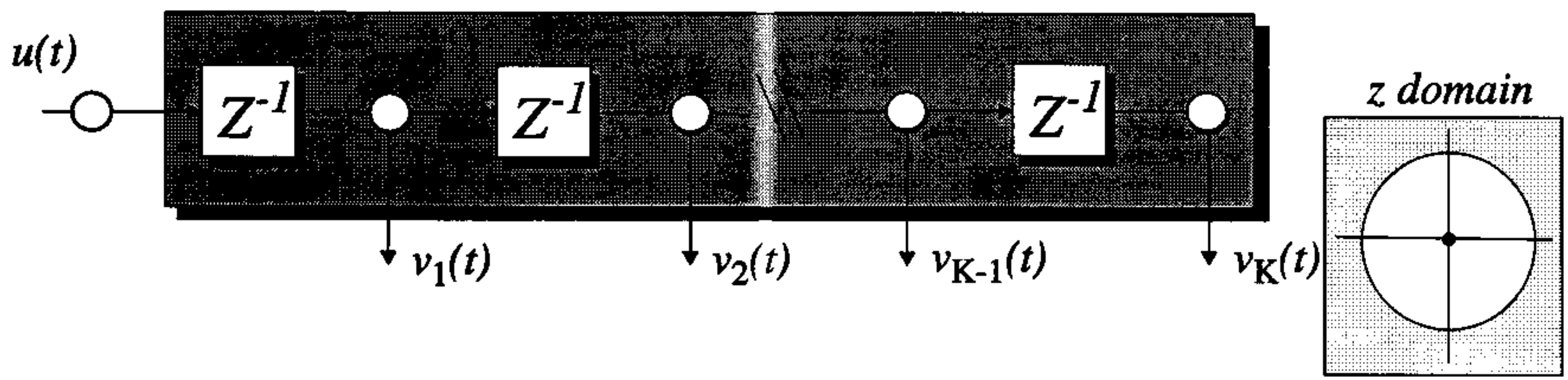


G(z) is the Generalized Delay Operator, an additive linear network.

$$NET(z) = \sum_{1}^{K} w_k X_k(z),$$

$$X_k(z) = G(z) X_{k-1}(z). \tag{3}$$

The Tapped Delay Line



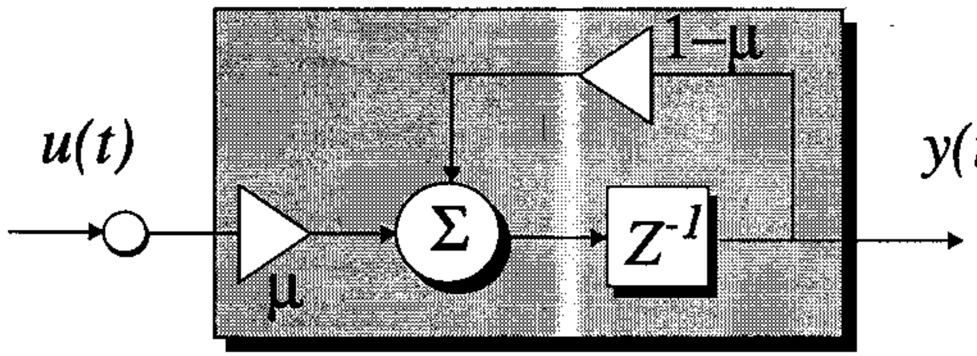
$$\text{Delay Operator } G(z) = z^{-1}.$$

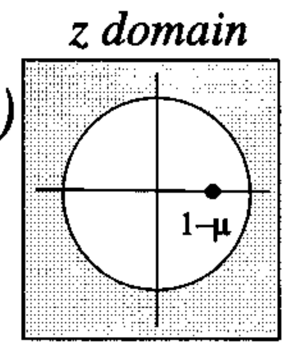
$$\text{Memory Depth } D_k \equiv \sum_{t=0}^{\infty} t g_k(t) = K,$$

$$\text{Resolution} \quad R_k \equiv \frac{K}{D_k} = 1 \quad .$$

<u>Notes</u>. General applications; high resolution, # weights proportional to depth!

The Leaky Integrator





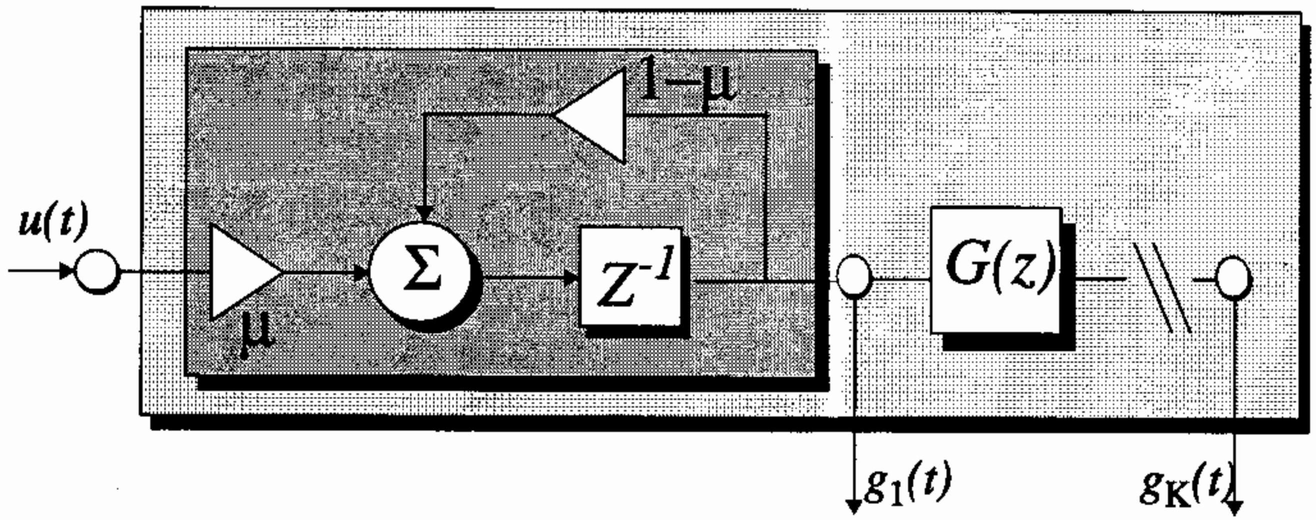
Delay Operator $G(z) = \dfrac{\mu}{z - (1-\mu)}$.

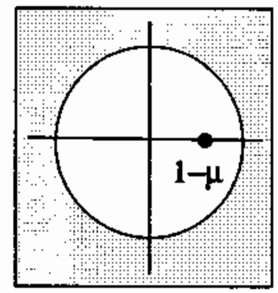
memory Depth $D = \sum_{t=0}^{\infty} t g(t) = \dfrac{1}{\mu}$.

Resolution $R = 1/D = \mu$.

Notes. Also called context units, memory neurons. Apply to problems where *deep memory with low resolution* is needed. Stable for $0 < \mu < 2$.

The Gamma Memory Filter





$$\text{delay operator } G(z) = \frac{\mu}{z - (1-\mu)} \quad .$$

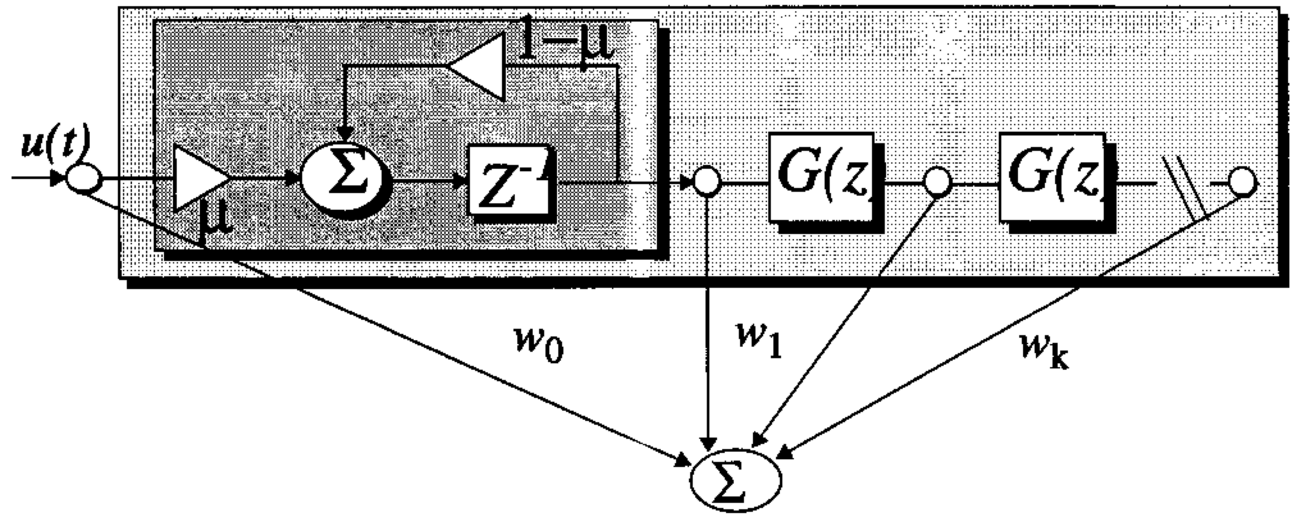
$$\text{depth} \quad D_k = \frac{K}{\mu} \; ,$$

$$\text{resolution} \quad R = K/\left(\frac{K}{1-\mu}\right) = \mu \; .$$

Notes. Gamma filter generalizes tapped delay line and leaky integrator into a single structure (order, μ).

ADALINE (μ) or Gamma Filter

The gamma filter just extends the adaptive linear network with a variable pole, adapted to the input signal statistics.



Learning equations:

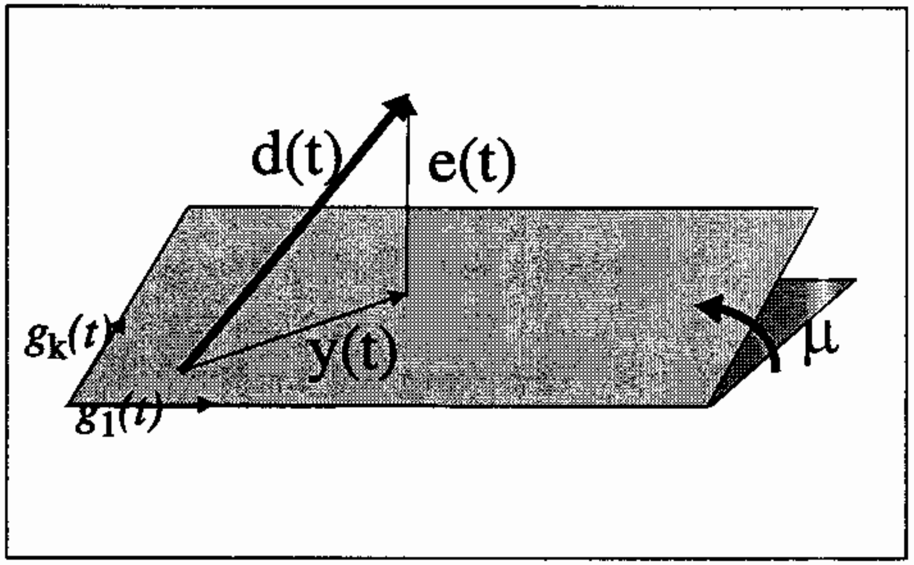
$$\Delta w_k(n) = \eta_1 e(n)\, x_k(n) \quad k = 0, \ldots, L$$

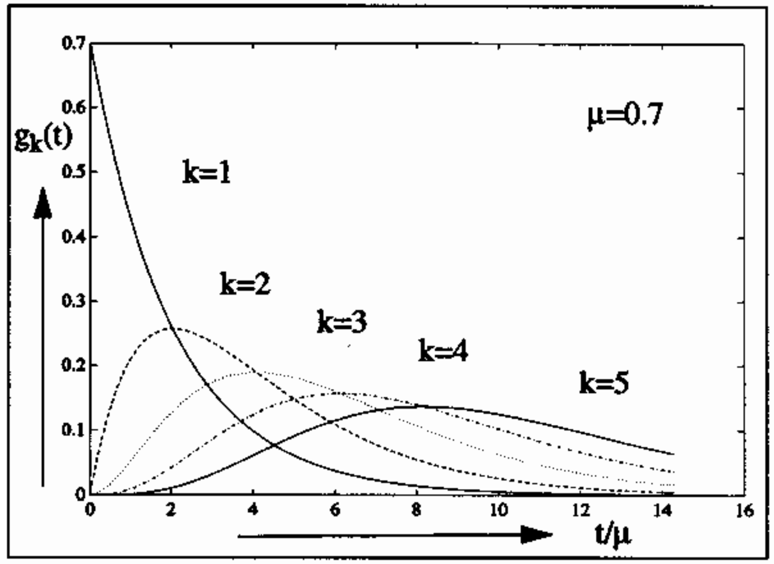
$$\Delta\mu(n) = \eta_2 \sum_{k=0}^{L} e(n)\, w_k \alpha_k(n)$$

where η is step size, e(n) the error and $\alpha_k(n) \equiv \frac{\partial}{\partial\mu} x_k(n)$

Structure of the gamma space

In continuous time the gamma space is a rigid hyperplane when μ varies. Thus, when the mse is minimized, μ works as an extra degree of freedom that changes the angle between the desired signal and the hyperplane.





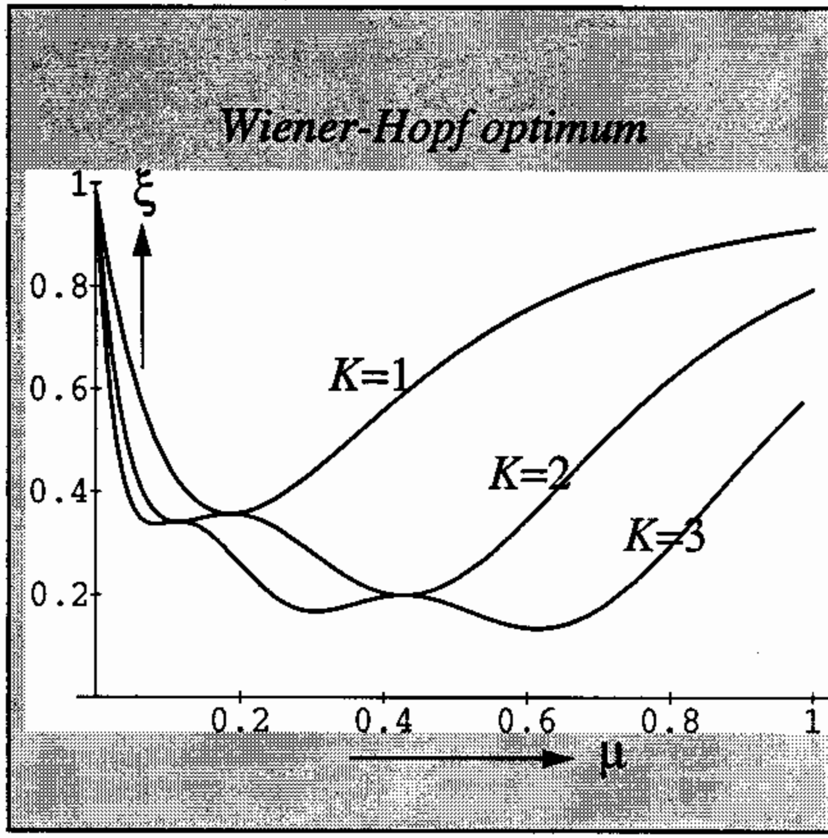
Problem is that the adaptation of μ is non-convex.

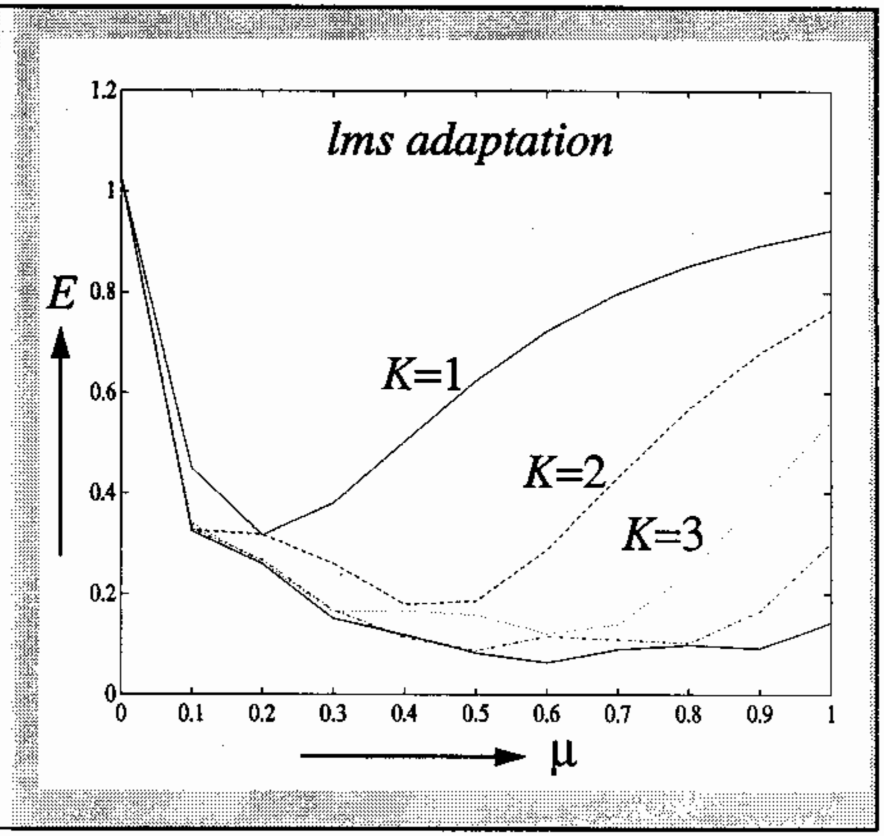
The gamma kernel is complete in L_2.

Memory Review Revisited

K-th ORDER	*feedforward*	*GAMMA*	*recurrent*
Stability	*Always*	*Trivial* $0<\mu<2$	*Difficult*
Adaptation Complexity	$O(K)$	$O(K)$	$O(K^2)$
Depth vs. Order	*Coupled* K	*Un-coupled* K/μ	*Free*
Shape	*Uniform*	*Free*	*Free*

A System Identification Experiment



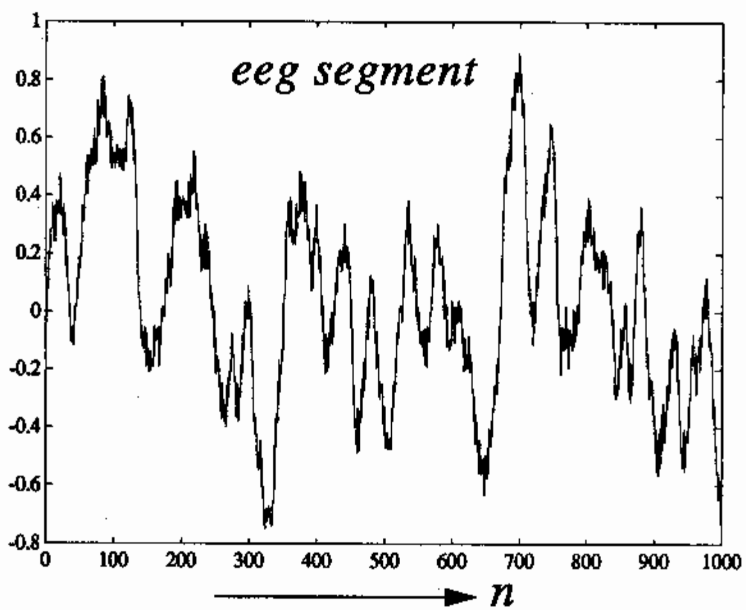


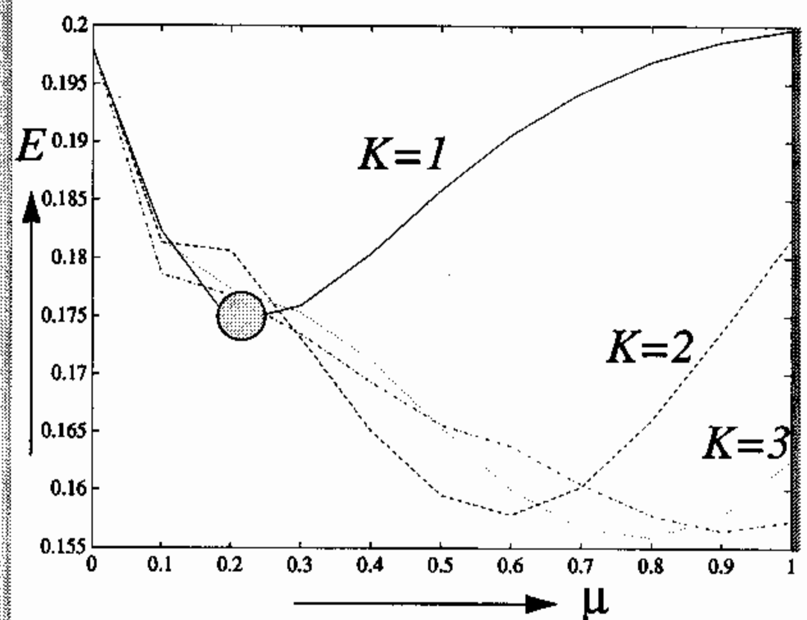
$$H(z) = \frac{0.0563 - 0.0009z^{-1} - 0.0009z^{-2} + 0.0563z^{-3}}{1 - 2.1291z^{-1} + 1.7834z^{-2} - 0.5435z^{-3}}$$

Just Another Set of Time Basis Functions?

Prediction of EEG segment.

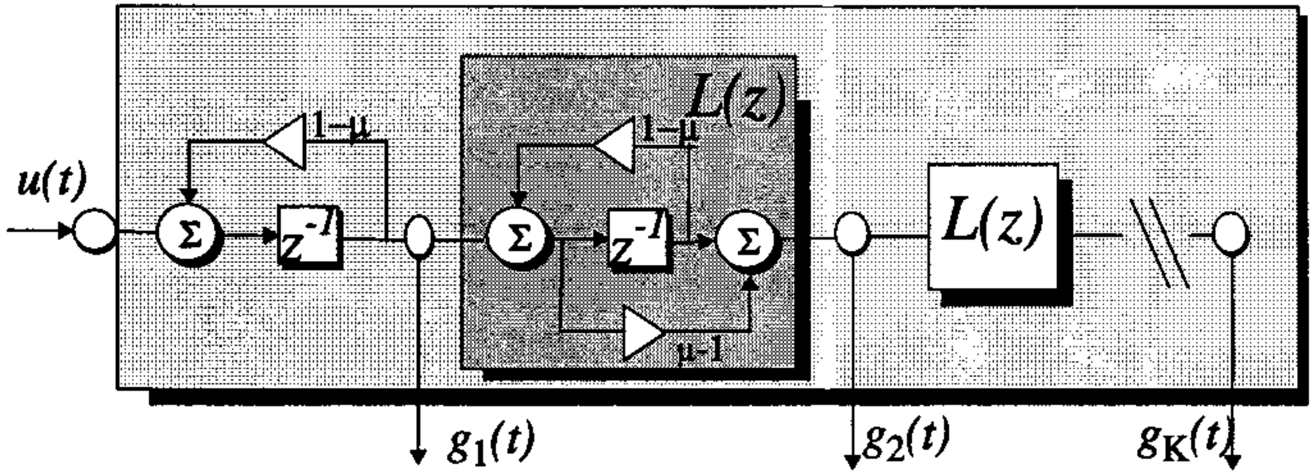
Architecture: $N_{in}=1$, K_{in} and μ parameter, $N_{hid}=5$, $N_{out}=1$, # pred. steps = 5.

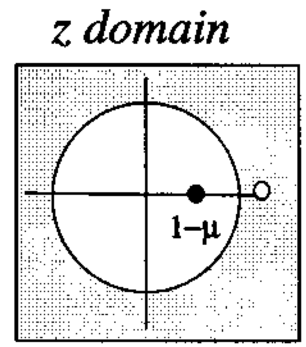




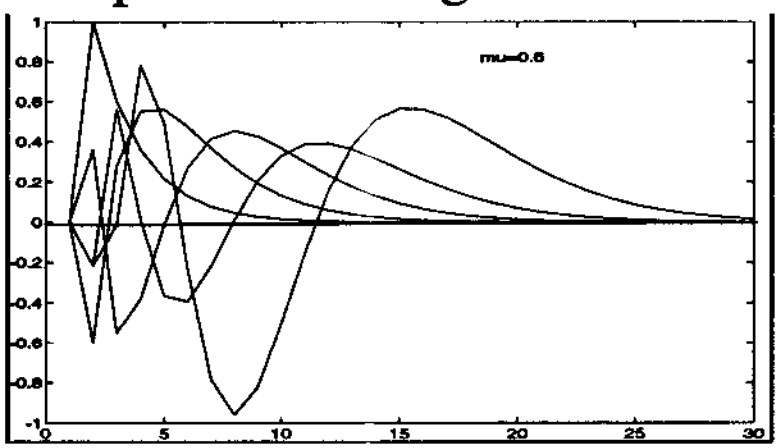
Other TLRN -Laguerre

The Laguerre filters are an orthogonal span of the Gamma space.



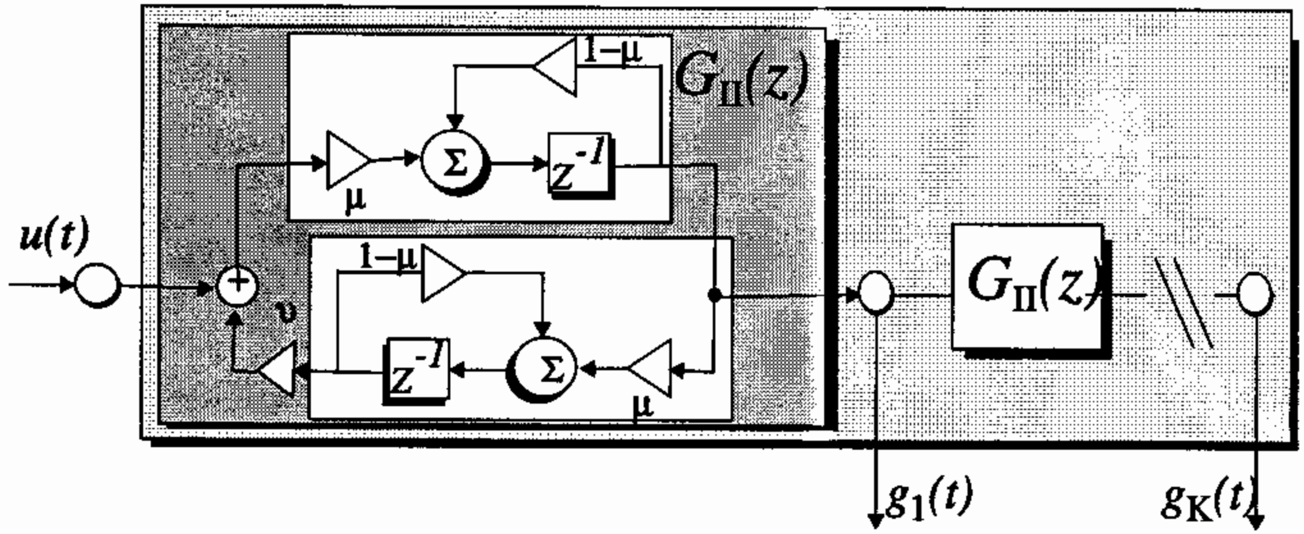


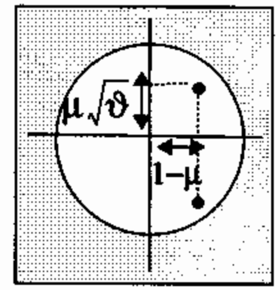
Laguerre should adapt faster than gamma for values of $\mu \sim 0, 2$.



Other TLRN -The Gamma II

Gamma Filter can be extended to complex poles.



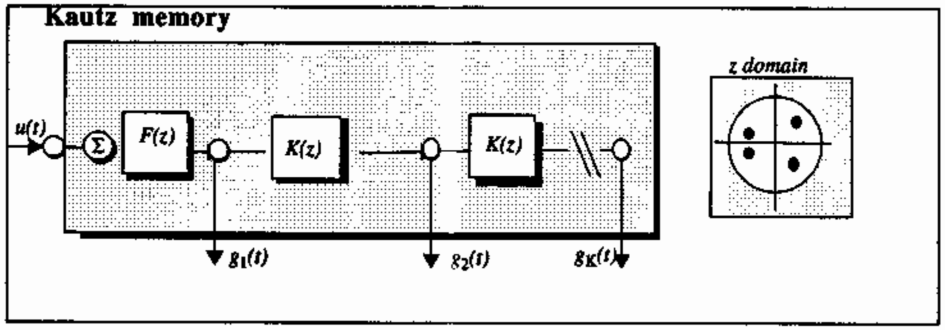


This structure is parametrized by μ and υ. It implements a general frequency dependent delay.

$$\text{Delay operator } G_{II}(z) = \frac{\mu\,[z-(1-\mu)]}{[z-(1-\mu)]^2+\vartheta\mu^2}$$

Most General TLRN- Kautz Memory

The Kautz functions implement general linear systems.



This memory has L poles of multiplicity K

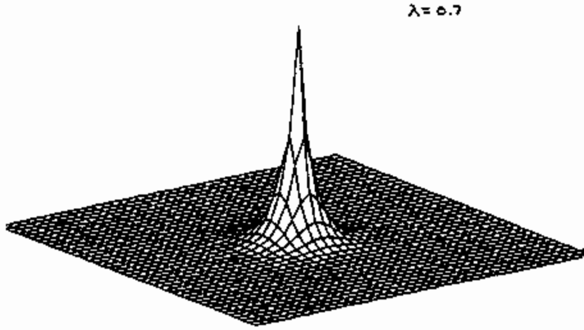
$$F(z,\vec{\mu}) = \frac{f(z,\vec{\mu})}{\prod_{i=1}^{L}(1-(1-\mu_i)z^{-1})} \qquad K(z,\vec{\mu}) = \frac{\prod_{i=1}^{L} z^{-1}-(1-\mu_i)^*}{\prod_{i=1}^{L} 1-(1-\mu_i)z^{-1}}$$

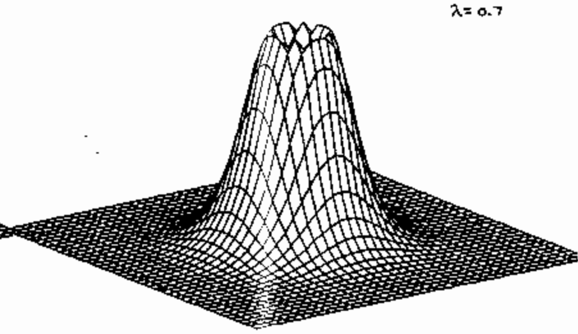
It is basically a vector Laguerre filter with complex poles. F(z) is a set of bandpass filters, and K(z) a set of allpass functions.

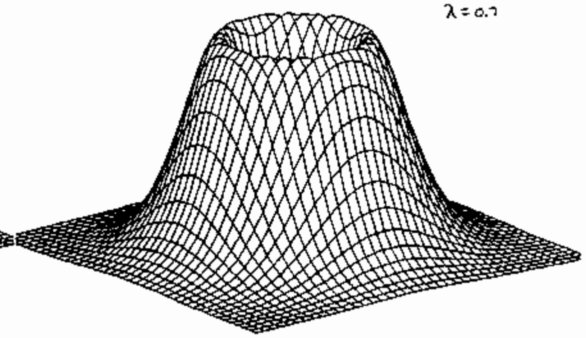
Multi-Dimensional Gamma

Can be considered an extension of radial basis functions.

$$g_j(\|\vec{x}(n) - c_i\|) = \frac{\mu^j}{(j-1)!}[(\vec{x}(n) - c_i)^2]^{0.5(j-1)} e^{-\mu\left[(\vec{x}(n) - c_i)^2\right]^{1/2}}$$







They are a compromise between local and global approximators.