E3.08
ISE3.17

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2009

EEE/ISE PART III/IV: MEng, BEng and ACGI

Corrected Copy Q2

ADVANCED SIGNAL PROCESSING

Friday, 8 May 10:00 am

Time allowed: 3:00 hours

There are FIVE questions on this paper.

Answer TWO of questions 1, 2, 3 and ONE of questions 4, 5.

All questions carry equal marks

Any special instructions for invigilators and information for candidates are on page 1.

Examiners responsible First Marker(s) : D.P. Mandic, D.P. Mandic
Second Marker(s) : M. Petrou, M. Petrou

1) Consider the problem of autoregressive moving average (ARMA) modelling.

a) State the expression for a general second order autoregressive AR(2) process $z[n]$. Explain the role of AR modelling in terms of shaping the spectrum of the driving white Gaussian noise (WGN). [3]

i) Derive the autocorrelation function for this process. Explain how you can obtain the expression for the autocorrelation function directly from the expression for the AR(2) process. Write down and explain the expression for the autocorrelation function for lags $k \geq 2$. [4]

ii) State the set of stability conditions for this process (stability triangle). Explain the bounds on the AR parameters for an AR(2) process to be stable. What are the four possibilities for a general shape of the autocorrelation function and spectrum? [4]

b) Consider the AR(2) process given by

$$z[n] = -1.9z[n-1] + 0.8z[n-2] + w[n]$$

where symbol $w[n]$ denotes samples of white Gaussian noise with finite variance $\sigma_w^2 < \infty$. Comment on the stability of the process $z[n]$ both in terms of the characteristic polynomial and stability triangle. [2]

i) How many data samples do we need to calculate the autocorrelation function, and how many correlation coefficients do we need for Yule-Walker estimation of AR(2) paremeters? Write down the Yule-Walker equations for an AR(2) process. [4]

ii) How many peaks in the spectrum can we expect for an AR(2) process and how many peaks for an AR(3) process. Explain. (Hint: we have an all–pole system) [3]

2) When signals are being observed by real world sensors, they are often corrupted by measurement noise $q[n]$ of variance σ_q^2. Consider the original signal $x[n]$ produced by an autoregressive process of order $p = 1$ (AR(1)), given by

$$x[n] = a_1 x[n-1] + w[n] \qquad w[n] \sim \mathcal{N}(0, \sigma_w^2)$$

which is measured as the noise corrupted process $y[n]$ given by

$$y[n] = x[n] + q[n] \qquad q[n] \sim \mathcal{N}(0, \sigma_q^2)$$

a) Show that the autoregressive parameter $\hat{a}_1$ is estimated from

$$\hat{a}_1 = \frac{r_{yy}(1)}{r_{yy}(0)}$$

where symbols $r_{yy}(1)$ and $r_{yy}(0)$ denote respectively the autocorrelations at lags $k = 1$ and $k = 0$. [4]

b) Show that the relation between the true AR coefficient a_1 and the estimated AR coefficient $\hat{a}_1$ is

$$\hat{a}_1 = a_1 \frac{\eta}{\eta + 1}$$

where $\eta = r_{xx}(0)/\sigma_q^2$ is the signal to noise ratio (SNR). [8]

c) Show that the noisy measurement $y[n]$ can be modelled as an ARMA(1,1) process, where the MA filter parameter b_1 is given as the solution of

$$\frac{1 + b_1^2}{b_1} = \sigma_w^2 + \frac{\sigma_q^2(1 + a_1^2)}{a_1 \sigma_q^2}$$

Correction at 11:30

Comment on the values of b_1 for a large ad small ratio σ_w^2/σ_q^2. [8]

Hint: compare the power spectrum of P_{yy} and the power spectrum of an ARMA(1,1) process $P(z) = \frac{\left(1+b_1 z^{-1}\right)\left(1+b_1 z\right)}{\left(1+a_1 z^{-1}\right)\left(1+a_1 z\right)}$.

3) Consider Minimum Variance Unbiased (MVU) estimation based on a linear data model.

a) Write down the vector-matrix equation for a linear model describing the signal

$$x[n] = A + Bn + w[n], \qquad n = 0, 1, \ldots, N-1, \quad w[n] \sim \mathcal{N}(0, \sigma^2)$$

where and A and B are unknown parameters. [3]

b) Write down the expression for the likelihood function for this data model. Explain the benefits of using the likelihood function over the standard probability density function. [3]

c) Explain the meaning of the regularity condition for the Cramer-Rao estimation. Explain how you would use the information obtained from the regularity condition to derive the MVU estimator for the vector parameter $\boldsymbol{\theta} = [A \;\; B]^T$ in the form

$$\hat{\boldsymbol{\theta}} = g(\mathbf{x}) = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}$$

(there is no need for a full derivation - just provide a brief sketch)
State the statistical condition on the columns of the measurement matrix $\mathbf{H}$ for this estimation problem to be feasible. [6]

d) A linear model from c) is used to estimate the parameters of the AR(2) process given by

$$x[n] = a_1\, x[n-1] + a_2\, x[n-2] + w[n], \qquad w[n] \sim \mathcal{N}(0, \sigma^2)$$

i) Express this process as a vector parameter linear model. For convenience start from $n = 2$, that is, from $x[2] = a_1 x[1] + a_2 x[0] + w[2]$. Are the columns of the measurement matrix orthogonal? [4]

ii) Based on only two data points, $x[2]$ and $x[3]$, and the general solution in c), write down the expression for the linear MVU solution for the AR(2) parameters $\hat{\boldsymbol{\theta}} = [\hat{a}_1 \;\; \hat{a}_2]^T$. Compare with the Yule-Walker solution. [4]

4) Adaptive linear prediction is at the core of adaptive filtering.

a) Sketch the block diagram of the adaptive prediction configuration. Explain the operation of an adaptive predictor. [4]

b) A finite impulse response (FIR) adaptive filter is employed within the adaptive prediction configuration. Derive the learning rate of the normalised least mean square (NLMS) algorithm by expanding the output error $e(k+1)$ of the LMS algorithm using a Taylor series around $e(k)$ and setting $e(k+1) = 0$. [6]

c) The class of sign algorithms is used to reduce the computational complexity of the LMS algorithm. Write down the weight update equations for this class of algorithms. Write down the equations for normalised versions of sign algorithms. [4]

d) Let $x(k)$ be a second order autoregressive (AR) process that is generated according to the difference equation

$$x(k) = 1.4x(k-1) - 0.6x(k-2) + w(k)$$

where $w(k)$ is a zero mean unit variance white Gaussian noise. An adaptive FIR predictor is used to predict process $\{x(k)\}$.

i) Using the result from b) write down the equations for the NLMS weight updates for such an adaptive predictor. [2]

ii) Describe the bound on the step size which ensures the convergence of the class of LMS algorithms. What effect does the value of the step size have on the convergence trajectory on the error performance surface? [2]

iii) What is the minimum achievable mean square error when using an adaptive FIR filter to predict the AR(2) process above? [2]

5) A simple extension of linear finite impulse response (FIR) adaptive filters is a nonlinear FIR filter shown in the Figure 5.1. The nonlinearity Φ is a saturation–type nonlinear function, such as tanh, and the output of this filter is given by $y(k) = \Phi\big(\mathbf{x}^T(k)\mathbf{w}(k)\big)$. In the stochastic gradient setting, the cost function for this filter is based on the minimisation of the squared instantanous output error and is given by

$$E(k) = \frac{1}{2}e^2(k)$$

a) Give the reasons for this nonlinear FIR filter also being called a "dynamical perceptron" or "artificial neuron". [4]

b) Derive the weight update equation for this filter based on the cost function given above and the gradient descent approach.
(Hint: $\Delta\mathbf{w}(k) = -\eta\nabla_{\mathbf{w}}E(k)$) [8]

c) Explain the difference in the way this filter and the standard FIR adaptive filter trained by the least mean square (LMS) algorithm process signals with large dynamical ranges. Which structure do you expect to perform better when filtering nonlinear signals? [2]

d) If the nonlinear function Φ is the tanh function, explain the effect of the saturation on the output of the filter and on the learning process (Hint: Cases when the operating point moves towards the tails of the nonlinearity, where the gradient values are very small). [2]

e) We would like train the nonlinear adaptive filter with a sign-sign algorithm. State the weight update equation and compare this type of learning with LMS based learning. [4]

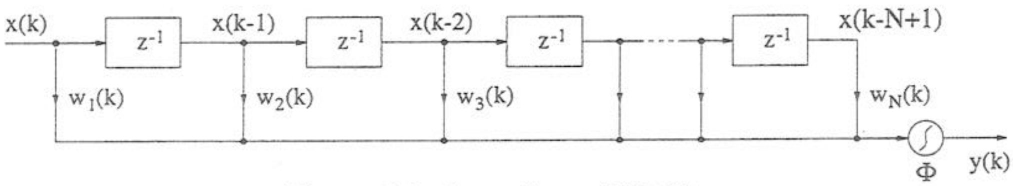

Figure 5.1: A nonlinear FIR filter

Master - April 09

Advanced Signal Processing

E3 08

Solutions

2009

Isc 3.17

1) [**Bookwork and practical application of bookwork**] a)

1/10

$$z[n] = a_1 z[n-1] + a_2 z[n-2] + w[n]$$

where a_1, a_2 are the model parameters and $\{w[n]\}$ is the driving white noise, $w[n] \sim \mathcal{N}(0, \sigma_w^2)$. The all-pole system above, with transfer function $H(z) = \frac{1}{1-a_1 z^{-1} - a_2 z^{-2}}$ shapes the flat spectrum of $w[n]$ and produces a spectrum with peaks determined by the roots of the polynomial in the denominator of $H(z)$.

i) By applying the expectation operator $E\{\cdot\}$ to

$$z[n-k]z[n]$$

we see that the ACF model follows the general form of the AR(2) process. It is often more convenient to consider this relationship in terms of the normalised correlation coefficients $\rho(k) = r(k)/r(0)$. Therefore we have

$$\rho(k) = a_1 \rho(k-1) + a_2 \rho(k-2)$$

where $\rho(0) = 1$. Since $\rho(0)$ and $\rho(1)$ depend on the driving noise terms $w[n]$ and $w[n-1]$, this relationship holds only for $k \geq 2$.

ii) Using the results for AR(1) and extending for AR(2) model, we can derive the bounds on stability for AR(2) processes, which in a convenient way can be put within a "stability triangle" shown in Figure 1.
From the ACF

$$\rho_k = a_1 \rho_{k-1} + a_2 \rho_{k-2} \qquad k > 0$$

- **Real roots:** $\Rightarrow (a_1^2 + 4a_2 > 0)$ ACF = mixture of damped exponentials
- **Complex roots:** $\Rightarrow (a_1^2 + 4a_2 < 0) \Rightarrow$ ACF exhibits a pseudo–periodic behaviour

$$\rho_k = \frac{D^k \sin(2\pi f_0 k + F)}{\sin F}$$

where: D denotes the damping factor of a sine wave with frequency f_0 and phase F.

$$\begin{aligned} D &= \sqrt{-a_2} \\ \cos(2\pi f_0) &= \frac{a_1}{2\sqrt{-a_2}} \\ \tan(F) &= \frac{1+D^2}{1-D^2}\tan(2\pi f_0) \end{aligned}$$

For stationarity we have

$$\begin{aligned} a_1 + a_2 &< 1 \\ a_2 - a_1 &< 1 \\ -1 < a_2 &< 1 \end{aligned}$$

Obviously the stability conditions are $-2 \leq a_1 \leq 2, -1 \leq a_1 \leq 1$, which is illustrated in the stability triangle.

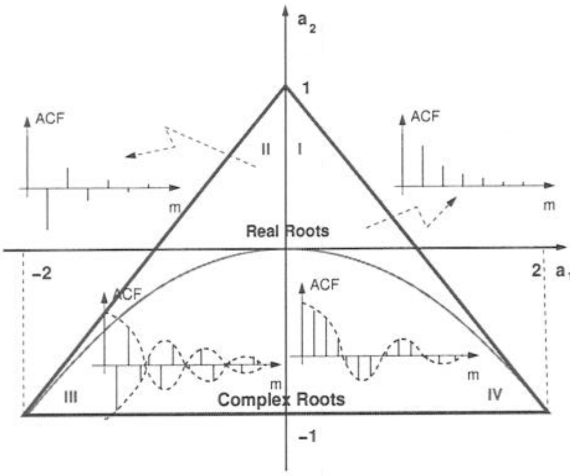

Figure 1: AR(2) Stability triangle. Region 1: Decaying ACF, Region 2: Decaying oscillating ACF, Region 3: Oscilating pseudoperiodic ACF, Region 4: Pseudoperiodic ACF

b) The process is not stable, as by inspection from the above stability triangle the combination of AR parameters lies outside the bounds on stability, and also by showing that the roots of the characteristic polynomial are larger than unity in magnitude, that is $z_1 = -2.2548, \; z_2 = 0.3548$.

i) This is second order variant of the general Yule Walker solution.
Substituting $p = 2$ into the general form of Yule–Walker equations, we have

$$\begin{aligned} \rho_1 &= a_1 + a_2 \rho_1 \\ \rho_2 &= a_1 \rho_1 + a_2 \end{aligned}$$

which when solved for a_1 and a_2 gives

$$a_1 = \frac{\rho_1(1-\rho_2)}{1-\rho_1^2}$$
$$a_2 = \frac{\rho_2-\rho_1^2}{1-\rho_1^2}$$

We thus need three values of the correlation function $\rho(0)$, $\rho(1)$ and $\rho(2)$, however the results are much more accurate if the ACF is first calculated over a longer segment and then the first three coefficients are used within the Yule-Walker method.

ii)

$$\begin{aligned} P_{zz}(f) &= \frac{2\sigma_w^2}{|1-a_1e^{-\jmath 2\pi f}-a_2e^{-\jmath 4\pi f}|^2} \\ &= \frac{2\sigma_w^2}{1+a_1^2+a_2^2-2a_1(1-a_2\cos(2\pi f)-2a_2\cos(4\pi f))}, \qquad 0 \le f \le 1/2 \end{aligned}$$

Due to the all–pole system, the spectrum is dominated by peaks. If the poles come in conjugate pairs (see the stability triangle in Figure 1), then every two poles produce one peak in the spectrum. Thus, the AR(2) power spectrum will have one peak. For an AR(3) model we cannot have two peaks, unless one of the peaks is for $\omega = 0$, that is at DC.

2) a) [**bookwork and new examples**]
The true AR(1) coefficient a_1 is estimated from

$$a_1 = \frac{r_{xx}(1)}{r_{xx}(0)}$$

Since we can only observe the noisy measurement $y[n] = x[n]+q[n]$, the coefficient $\hat{a}_1$ is estimated from

$$\hat{a}_1 = \frac{r_{yy}(1)}{r_{yy}(0)}$$

that is, based on the noisy measurement $y[n]$.

b) Since the measurement noise $q[n]$ is white (e.g. $q[n] \perp q[n+1]$, $x \perp q$ and $E\{q^2(n)\} = \sigma_q^2$), we have

$$\begin{aligned} r_{yy}(0) &= E\{y[n]y[n]\} = E\{(x[n]+q[n])(x[n]+q[n])\} = r_{xx}(0) + \sigma_q^2 \\ r_{yy}(1) &= E\{y[n]y[n+1]\} = E\{(x[n]+q[n])(x[n+1]+q[n+1])\} = r_{xx}(1) \end{aligned}$$

Thus

$$\hat{a}_1 = \frac{r_{xx}(1)}{r_{xx}(0)+\sigma_q^2} = a_1\frac{1}{1+\frac{1}{\eta}} = a_1\frac{\eta}{1+\eta} \qquad \text{where} \qquad \eta = \mathrm{r_{xx}}(0)/\sigma_\mathrm{q}^2$$

Hence, as the SNR reduces, the coefficients of an AR model reduce in magnitude.

c) Since we have $y[n] = x[n] + q[n]$, then the corresponding power spectrum

$$\begin{aligned} P_{yy}(z) &= P_{xx}(z) + P_{qq}(z) = \frac{\sigma_w^2}{(1-a_1z^{-1})(1-a_1z}) + \sigma_q^2 \\ &= \frac{\sigma_w^2 + \sigma_q^2\left(1+a_1z^{-1}\right)\left(1+a_1z\right)}{\left(1+a_1z^{-1}\right)\left(1+a_1z\right)} \end{aligned}$$

Compare with the spectrum of ARMA(1,1) process

$$P(z) = \frac{\left(1+b_1z^{-1}\right)\left(1+b_1z\right)}{\left(1+a_1z^{-1}\right)\left(1+a_1z\right)}$$

and equate the numerator coefficients to yield

$$\begin{aligned} &(1) \qquad \sigma_w^2 + \sigma_q^2(1+a_1^2) = 1 + b_1^2 \\ &(2) \qquad \sigma_q^2 a_1(z+z^{-1}) = b_1(z+z^{-1}) \end{aligned}$$

to yield

$$\frac{\sigma_w^2 + \sigma_q^2(1+a_1^2)}{\sigma_q^2 a_1} = \frac{1+b_1^2}{b_1}$$

Therefore for a low SNR the ratio σ_w^2/σ_q^2 is small and $a_1 \to b_1$. For a high SNR $b_1 \to 0$ and we have the true AR signal.

3) a) [**bookwork**] The model is given by

$$\mathbf{x} = \mathbf{H}\,\boldsymbol{\theta} + \mathbf{w}$$

where $\mathbf{x} = [x[0], x[1], \ldots, x[N-1]]^T$, $\boldsymbol{\theta} = [A \;\; B]^T$, and

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & N-1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{w} = \begin{bmatrix} w[0] \\ w[1] \\ \vdots \\ w[N-1] \end{bmatrix}$$

or in an expanded form

$$\begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & N-1 \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} + \begin{bmatrix} w[0] \\ w[1] \\ \vdots \\ w[N-1] \end{bmatrix}$$

b) The likelihood function is given by

$$lnp(\mathbf{x};\boldsymbol{\theta}) = \left[-\frac{N}{2} ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\mathbf{x} - \mathbf{H}\boldsymbol{\theta})^T(\mathbf{x} - \mathbf{H}\boldsymbol{\theta})\right]$$

It is much more convenient to deal with the log-likelihood function than with the standard joint pdf of data and parameters, and the monotonic nature of the ln function preserves the position of extrema. In addition, the mathematical tractability is much enhanced, as upon applying the log function the terms in the exponentials are brought down and the products become sums.

c) The regularity condition

$$\frac{\partial ln\; p(\mathbf{x};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{I}(\boldsymbol{\theta})\big(g(\mathbf{x}) - \boldsymbol{\theta}\big) = 0$$

simply indicates that there is a solution to the linear MVU estimation problem for which the estimator $g(\mathbf{x}) = \boldsymbol{\theta}$. The derivative of the log-likelihood function wrt the unknown vector parameter $\boldsymbol{\theta}$ is (there is no need to give this level of detail)

$$\frac{\partial p(\mathbf{x};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{1}{\sigma^2}\left[\mathbf{H}^T\mathbf{x} - \mathbf{H}^T\mathbf{H}\boldsymbol{\theta}\right]$$

Upon involving the Fisher Information Matrix

$$\mathbf{I}(\boldsymbol{\theta}) = -\frac{\partial^T}{\partial \boldsymbol{\theta}}\left[\frac{\partial ln\; p(\mathbf{x};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right] = \frac{1}{\sigma^2}\mathbf{H}^T\mathbf{H}$$

we obtain the MVU solution for the linear vector parameter model.

Since the parameter estimation involves the inverse of the measurement matrix $\mathbf{H}$, its columns must be orthogonal for the inverse to take place, and for the matrix $\mathbf{H}^T\mathbf{H}$ to be positive semidefinite.

d) **new example**
i) In a similar way as in a), we can write the AR parameter estimation problem as a linear estimation problem in the form (for convenience, start from $N = 2$)

$$\underbrace{\begin{bmatrix} x[2] \\ x[3] \\ \vdots \\ x[N-1] \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} x[1] & x[0] \\ x[2] & x[1] \\ \vdots & \vdots \\ x[N-2] & x[N-3] \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}}_{\boldsymbol{\theta}} + \underbrace{\begin{bmatrix} w[2] \\ w[3] \\ \vdots \\ w[N-1] \end{bmatrix}}_{\mathbf{w}}$$

Then the AR(2) parameter vector becomes

$$\hat{\boldsymbol{\theta}} = g(\mathbf{x}) = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{x}$$

ii) We would have

$$\begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \end{bmatrix} = \left(\begin{bmatrix} x[1] & x[2] \\ x[0] & x[1] \end{bmatrix} \begin{bmatrix} x[1] & x[0] \\ x[2] & x[1] \end{bmatrix} \right)^{-1} \begin{bmatrix} x[1] & x[2] \\ x[0] & x[1] \end{bmatrix} \quad (1)$$

We can see that this form has similarities with the Yule-Walker form

$$\hat{\boldsymbol{\theta}}\mathbf{R}^{-1}\mathbf{r}$$

where the the produce $\mathbf{H}^T\mathbf{H}$ takes the role of the correlation matrix. For a white Gaussian driving noise both solution should lead to the same result.

4) a) **bookwork and worked example**

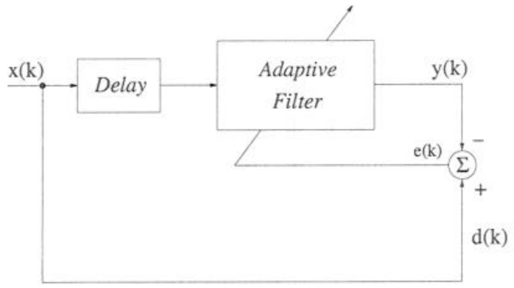

Figure 4.1 Adaptive prediction configuration

The role of the teaching signal is taken by the "forwarded" input $x(k+M)$ where M is the prediction horizon. This configuration can also be used for adaptive line enhancement, as the correlation structure of white noise exhibits very fast decay, and an input shifted by several time instants should not contain the same noise structure as the original input.

b)

$$e(k+1) = e(k) + \sum_{i=1}^{N} \frac{\partial e(k)}{\partial w_i(k)} \Delta w_i(h) + \text{Higher Order Terms}$$

Inserting the partial derivatives from the above, we arrive at

$$e(k+1) = e(k)\left[1 - \eta \parallel \mathbf{x}(k) \parallel_2^2\right]$$

From there the NLMS step size which minimizes the error is

$$\eta_{NLMS} = \frac{1}{\parallel \mathbf{x}(k) \parallel_2^2}$$

c) Sign algorithms apply the sign operator to the error and input signals.
The sign-error algorithm applies the sign operator to the filter error, and is given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \eta sign\big(e(k)\big)\mathbf{x}(k)$$

The sign-regressor algorithm applies the sign operator to the input vector, and is given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \eta e(k) sign\big(\mathbf{x}(k)\big)$$

The sign–sign algorithm combines the sign–error and sign–regressor algorithms, and is given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \eta sign\big(e(k)\big) sign\big(\mathbf{x}(k)\big)$$

It is very convenient to implement in hardware, especially if the step size is chosen to be a multiple of 2. In that case it is extremely fast. The drawback is an increased mean square error as compared to LMS, especially for inputs with low amplitude range.

The normalised versions of these algorithms are obtained upon dividing the weight update by the correspnding tap input powers, that is by $\| \mathbf{x}(k) \|_2^2$ for the sign error algorithm, and by N^2 for the sign-regressor and sign-sign algorithms.

d) **new example** i)

$$w_1(k+1) = w_1(k) + \frac{\mu}{x^2(k-1) + x^2(k-2)} e(k)x(k-1)$$
$$w_2(k+1) = w_2(k) + \frac{\mu}{x^2(k-1) + x^2(k-2)} e(k)x(k-2)$$

where μ is the learning rate.

ii) The step size of the LMS is bounded by the reciprocal of the maximum eigenvalue of the input autoccorelation matrix, that is

$$0 < \mu < \frac{2}{\lambda_{max}}$$

For the NLMS, this bound is $0 < \mu < 1$.

iii) It is convenient to write the teaching signal as

$$d(k) = x(k) + w(k) \qquad \text{where} \qquad \mathrm{w(k)} \sim \mathcal{N}(0, \sigma_{\mathrm{w}}^2)$$

In ARMA modelling, usually $\sigma_w^2 = 1$, that is the WGN is with unit variance. The minimum mean square error for this example is therefore

$$\xi_{min} = \sigma_w^2 = 1$$

Namely after the filter coefficients converge to the true values of the AR parameters, there is still the "unpredictable" part due to the error term in the AR model, which in this case has unit variance.

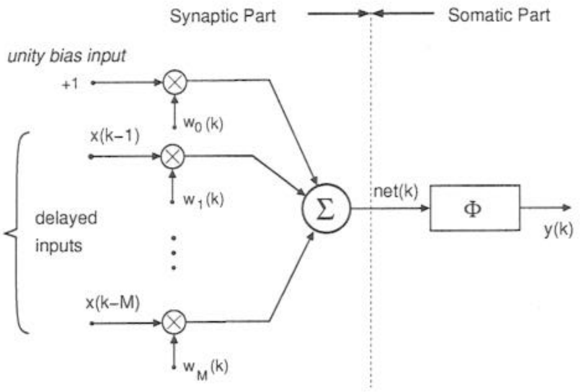

Figure 5.1: Model of Artificial Neuron

5) [**bookwork and intuitive reasoning**]

a) Dynamical perceptron. The structure represents an electrical model of a neuron from the brain. It has its synaptic part (delayed inputs and weihts) and somatic part (summation and nonlinearity), as shown in Figure 5.1.
b) Based on Figure 5.1, this nonlinear FIR filter can be trained by a procedure similar to the LMS, with the exception that we need to account for the nonlinearity within the structure. Thus, based on the cost function $J(k) = \frac{1}{2}e^2(k)$, we have

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \nabla_{\mathbf{w}} J$$

where

$$\nabla_{\mathbf{w}} J = \frac{\partial J}{\partial e(k)} \frac{\partial e(k)}{\partial \Phi(k)} \frac{\partial \Phi(k)}{\partial y(k)} \frac{\partial y(k)}{\partial \mathbf{w}(k)} = e(k)(-1)\Phi'(k)\mathbf{x}(k)$$

The final update then becomes

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \eta e(k)\Phi'(\mathbf{x}^T(k)\mathbf{w}(k))\mathbf{x}(k)$$

c) The FIR filter trained by LMS is linear and is not sensitive to the amplitudes of the signal. This dynamical perceptron has a saturation type nonlinearity and is therefore sensitive to the changes in the signal dynamics.

d) The nonlinearity can be thought of as having a quasilinear range and a saturation range, hence producing nonlinear distortion of input signals, when the signals are linear, but also having the ability to track better nonlinear signals.
The output magnitude range is limited to the range of the nonlinearity, also the traning can be very slow in the tails of the nonlinearity, due to the low value of

10/10

gradient at those points.

e) If we apply the sign-sign algorithm to the weight update of the dynamical perceptron, we obtain

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \text{sign}\big(\Phi(\text{k})\big)\text{sign}\big(\text{e}(\text{k})\big)\text{sign}\big(\mathbf{x}(\text{k})\big)$$

We still do have a source of nonlinearity as the odd symmetric function Φ becomes a hard limiter and also both the error and input have values $\in \{-1, 1\}$. Thus, the new algorithm is less computationally expensive than LMS, but has larger steady state error.