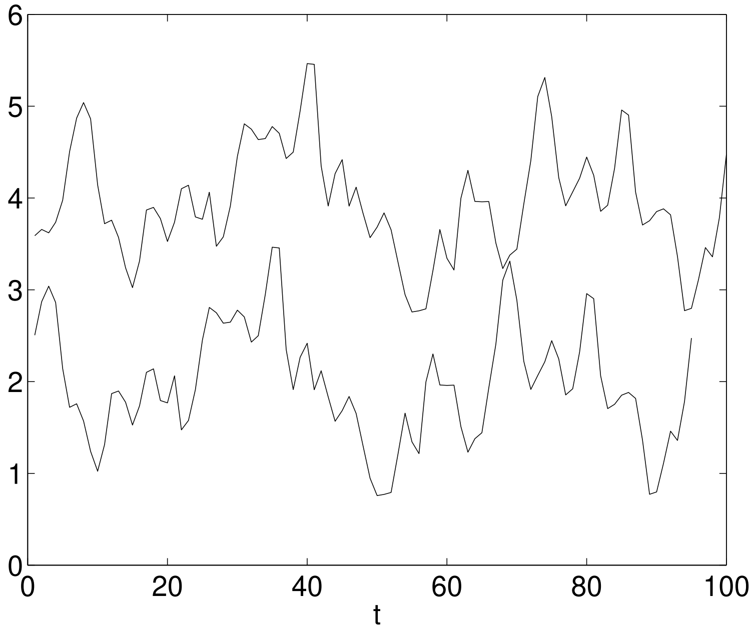
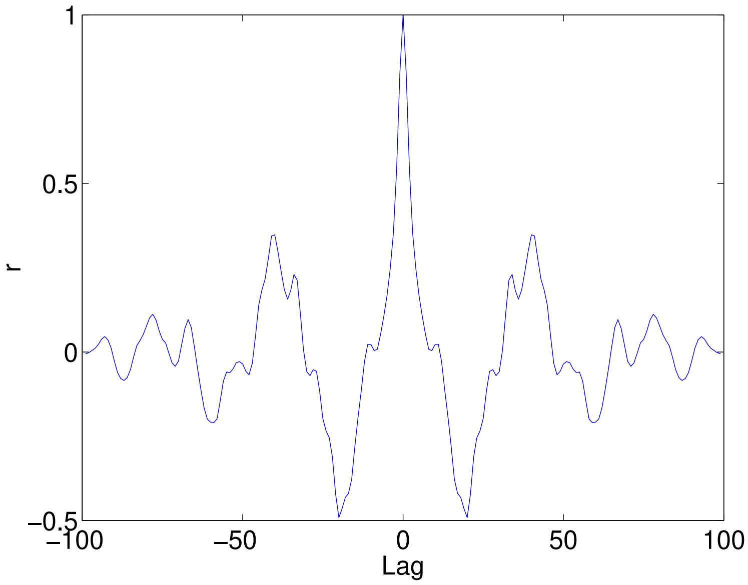


(a)

Figure 6.1: *Signal x_t (top) and x_{t+5} (bottom). The bottom trace* **leads** *the top trace by 5 samples. Or we may say it* **lags** *the top by -5 samples.*



(a)

Figure 6.2: *Autocorrelation function for x_t. Notice the negative correlation at lag 20 and positive correlation at lag 40. Can you see from Figure 6.1 why these should occur ?*

6.3 Autoregressive models

An autoregressive (AR) model *predicts* the value of a time series from previous values. A pth order AR model is defined as

$$x_t = \sum_{i=1}^{p} x_{t-i} a_i + e_t \tag{6.3}$$

where a_i are the *AR coefficients* and e_t is the prediction error. These errors are assumed to be Gaussian with zero-mean and variance σ_e^2. It is also possible to include an extra parameter a_0 to soak up the mean value of the time series. Alternatively, we can first subtract the mean from the data and then apply the zero-mean AR model described above. We would also subtract any trend from the data (such as a linear or exponential increase) as the AR model assumes stationarity (see later).

The above expression shows the relation for a single time step. To show the relation for all time steps we can use matrix notation.

We can write the AR model in matrix form by making use of the *embedding matrix*, $\boldsymbol{M}$, and by writing the signal and AR coefficients as vectors. We now illustrate this for $p = 4$. This gives

$$\boldsymbol{M} = \begin{bmatrix} x_4 & x_3 & x_2 & x_1 \\ x_5 & x_4 & x_3 & x_2 \\ .. & .. & .. & .. \\ x_{N-1} & x_{N-2} & x_{N-3} & x_{N-4} \end{bmatrix} \tag{6.4}$$

We can also write the AR coefficients as a vector $\boldsymbol{a} = [a_1, a_2, a_3, a_4]^T$, the errors as a vector $\boldsymbol{e} = [e_5, e_6, ..., e_N]^T$ and the signal itself as a vector $\boldsymbol{X} = [x_5, x_6, ..., x_N]^T$. This gives

$$\begin{bmatrix} x_5 \\ x_6 \\ .. \\ x_N \end{bmatrix} = \begin{bmatrix} x_4 & x_3 & x_2 & x_1 \\ x_5 & x_4 & x_3 & x_2 \\ .. & .. & .. & .. \\ x_{N-1} & x_{N-2} & x_{N-3} & x_{N-4} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} + \begin{bmatrix} e_5 \\ e_6 \\ .. \\ e_N \end{bmatrix} \tag{6.5}$$

which can be compactly written as

$$\boldsymbol{X} = \boldsymbol{M}\boldsymbol{a} + \boldsymbol{e} \tag{6.6}$$

The AR model is therefore a special case of the multivariate regression model (compare the above equation to that given in the second lecture). The AR coefficients can therefore be computed from the equation

$$\hat{\boldsymbol{a}} = (\boldsymbol{M}^T\boldsymbol{M})^{-1}\boldsymbol{M}^T\boldsymbol{X} \tag{6.7}$$

The AR predictions can then be computed as the vector

$$\hat{\boldsymbol{X}} = \boldsymbol{M}\hat{\boldsymbol{a}} \tag{6.8}$$

and the error vector is then $\boldsymbol{e} = \boldsymbol{X} - \hat{\boldsymbol{X}}$. The variance of the noise is then calculated as the variance of the error vector.

To illustrate this process we analyse our data set using an AR(4) model. The AR coefficients were estimated to be

$$\hat{\boldsymbol{a}} = [1.46, -1.08, 0.60, -0.186]^T \tag{6.9}$$

and the AR predictions are shown in Figure 6.3. The noise variance was estimated to be $\sigma_e^2 = 0.079$ which corresponds to a standard deviation of 0.28. The variance of the original time series was 0.3882 giving a signal to noise ratio of $(0.3882 - 0.079)/0.079 = 3.93$.

6.3.1 Random walks

If $p = 1$ and $a_1 = 1$ then the AR model reduces to a random walk model, an example of which is shown in Figure 6.4.

6.3.2 Relation to autocorrelation

The autoregressive model can be written as

$$x_t = a_1 x_{t-1} + a_2 x_{t-2} + ... + a_p x_{t-p} + e_t \tag{6.10}$$

If we multiply both sides by x_{t-k} we get

$$x_t x_{t-k} = a_1 x_{t-1} x_{t-k} + a_2 x_{t-2} x_{t-k} + ... + a_p x_{t-p} x_{t-k} + e_t x_{t-k} \tag{6.11}$$

If we now sum over t and divide by $N-1$ and assume that the signal is zero mean (if it isn't we can easily make it so, just by subtracting the mean value from every sample) the above equation can be re-written in terms of covariances at different lags

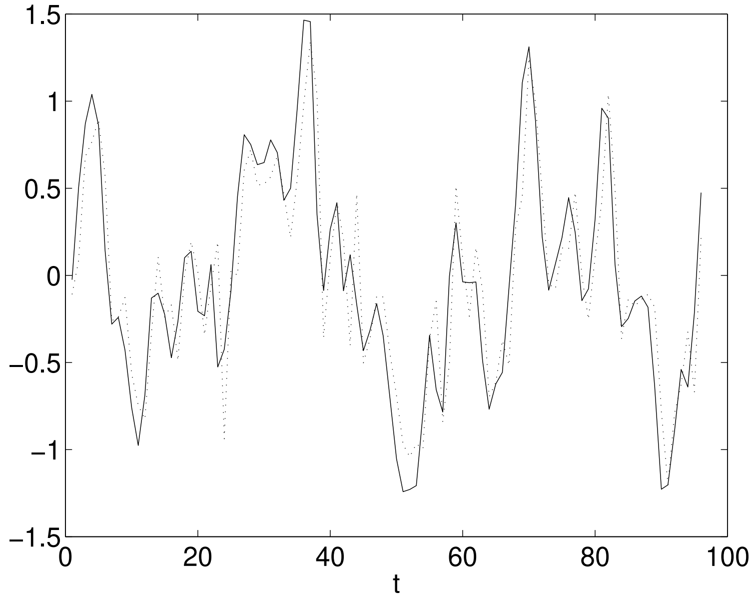
$$\sigma_{xx}(k) = a_1 \sigma_{xx}(k-1) + a_2 \sigma_{xx}(k-2) + ... + a_p \sigma_{xx}(k-p) + \sigma_{e,x} \tag{6.12}$$

where the last term $\sigma_{e,x}$ is the covariance between the noise and the signal. But as the noise is assumed to be independent from the signal $\sigma_{e,x} = 0$. If we now divide every term by the signal variance we get a relation between the correlations at different lags

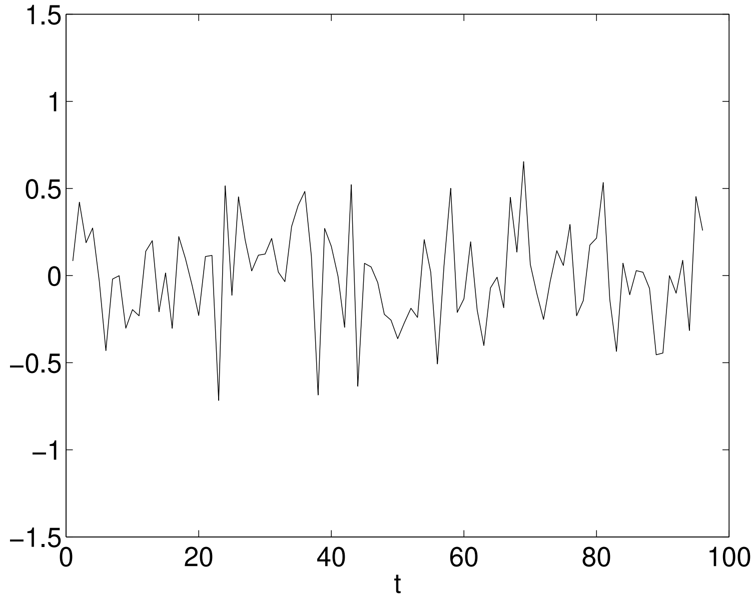
$$r_{xx}(k) = a_1 r_{xx}(k-1) + a_2 r_{xx}(k-2) + ... + a_p r_{xx}(k-p) \tag{6.13}$$

This holds for all lags. For an AR(p) model we can write this relation out for the first p lags. For $p = 4$

$$\begin{bmatrix} r_{xx}(1) \\ r_{xx}(2) \\ r_{xx}(3) \\ r_{xx}(4) \end{bmatrix} = \begin{bmatrix} r_{xx}(0) & r_{xx}(-1) & r_{xx}(-2) & r_{xx}(-3) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(-1) & r_{xx}(-2) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & r_{xx}(-1) \\ r_{xx}(3) & r_{xx}(2) & r_{xx}(1) & r_{xx}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \tag{6.14}$$



(a)



(b)

Figure 6.3: *(a) Original signal (solid line),* $\boldsymbol{X}$*, and predictions (dotted line),* $\hat{\boldsymbol{X}}$*, from an AR(4) model and (b) the prediction errors,* $\boldsymbol{e}$*. Notice that the variance of the errors is much less than that of the original signal.*

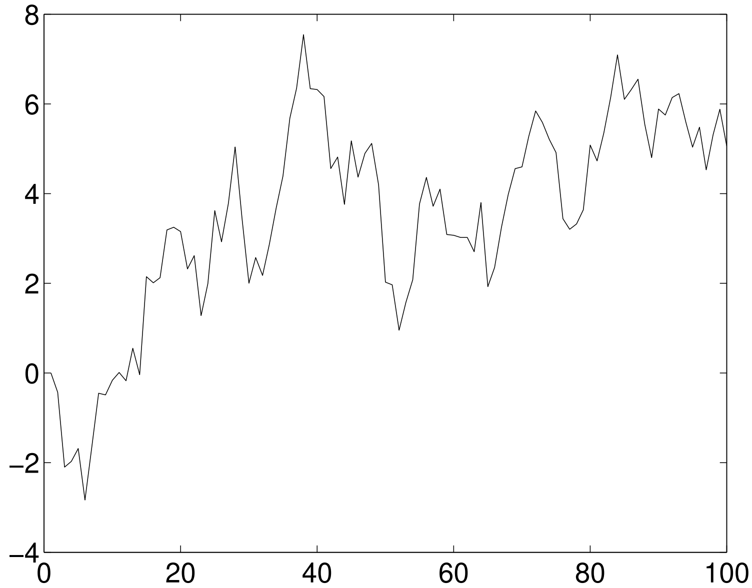


Figure 6.4: *A random walk.*

which can be compactly written as

$$\boldsymbol{r} = \boldsymbol{R}\boldsymbol{a} \tag{6.15}$$

where $\boldsymbol{r}$ is the autocorrelation vector and $\boldsymbol{R}$ is the autocorrelation matrix. The above equations are known, after their discoverers, as the Yule-Walker relations. They provide another way to estimate AR coefficients

$$\boldsymbol{a} = \boldsymbol{R}^{-1}\boldsymbol{r} \tag{6.16}$$

This leads to a more efficient algorithm than the general method for multivariate linear regression (equation 6.7) because we can exploit the structure in the autocorrelation matrix. By noting that $r_{xx}(k) = r_{xx}(-k)$ we can rewrite the correlation matrix as

$$\boldsymbol{R} = \begin{bmatrix} 1 & r_{xx}(1) & r_{xx}(2) & r_{xx}(3) \\ r_{xx}(1) & 1 & r_{xx}(1) & r_{xx}(2) \\ r_{xx}(2) & r_{xx}(1) & 1 & r_{xx}(1) \\ r_{xx}(3) & r_{xx}(2) & r_{xx}(1) & 1 \end{bmatrix} \tag{6.17}$$

Because this matrix is both symmetric and a Toeplitz matrix (the terms along any diagonal are the same) we can use a recursive estimation technique known as the Levinson-Durbin algorithm [1].

6.3.3 Relation to partial autocorrelation

The partial correlation coefficients (see lecture 2) in an AR model are known as *reflection coefficients*. At lag m, the partial correlation between x_{t-m} and x_t, is

[1] This algorithm actually operates on the autocovariance matrix, although some authors, eg. Pardey et al. [45], call it the autocorrelation matrix. What we refer to as autocorrelation, they refer to as normalised autocorrelation.

written as k_m; the mth reflection coefficient. It can be calculated as the relative reduction in prediction error

$$k_m = \frac{E_{m-1} - E_m}{E_{m-1}} \tag{6.18}$$

where E_m is the prediction error from an $AR(m)$ model [2]. The reflection coefficients are to the AR coefficients what the correlation is to the slope in a univariate AR model; if the mth reflection coefficient is significantly non-zero then so is the mth AR coefficient. And vice-versa.

The Levinson-Durbin algorithm computes reflection coefficients as part of a recursive algorithm for computing the AR coefficients. It finds k_1 and from it calculates the AR coefficient for an $AR(1)$ model, a_1. It then computes k_2 and from it calculates the AR coefficients for an AR(2) model (a_2 is computed afresh and a_1 is re-estimated from a_1 for the $AR(1)$ model - as it will be different). The algorithm continues by calculating k_m and the coefficients for $AR(m)$ from $AR(m-1)$. For details, see Pardey *et al.* [45].

6.3.4 Model order selection

Because an AR model is a special case of multivariate regression we can use the same significance testing procedure (see earlier lecture) to determine the relevance or otherwise of our variables. To recap, (i) we compute our coefficients for the AR(p) model, (ii) we estimate the standard deviation of each AR coefficient (see second lecture), (iii) we then perform a double-sided t-test to see if the smallest coefficient is significantly different from zero. If it is, then we might try a larger model order and repeat the process. If it isn't then we might try a smaller model order. We can either start with a model order of 1 and gradually increase it (stepwise forward selection) or start with a very high model order and gradually decrease it (stepwise backward selection), performing the significance test as we increase/decrease the model order.

We note that the above statistical test is identical to seeing whether or not the pth reflection coefficient is significantly non-zero (see earlier lecture).

For our data set both SFS and SBS, with a significance level of 0.05, chose $p = 3$ as the optimal model order. For SFS, for example, when $p = 4$ the smallest coefficient is $a_4 = -0.186$ and the corresponding standard deviation is $\sigma_4 = 0.103$. This gives a t-statistic of $t = -1.8006$ which under a double-sided test gives a probability of 0.0749. We therefore cannot reject the null hypothesis that the coefficient is zero at the 0.05 significance level; the SFS procedure therefore stops at a model order of 3.

Alternatively, we could use other model selection criteria eg. the Minimum Description Length (MDL) (see Lecture 4)

$$MDL(p) = \frac{N}{2} \log \sigma_e^2(p) + \frac{p}{2} \log N \tag{6.19}$$

[2] We have also written $E_m = \sigma_e^2(m)$.

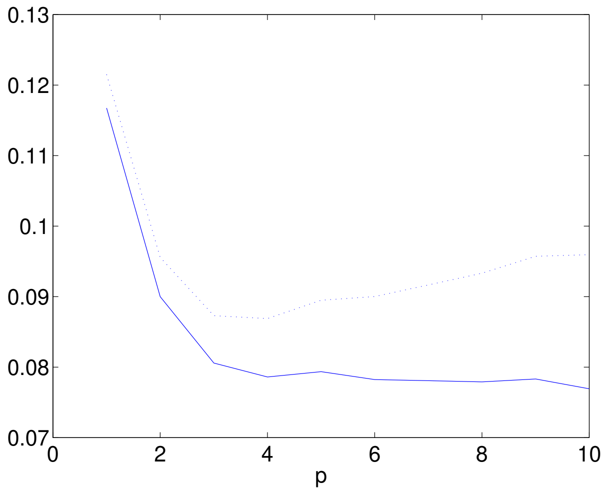


Figure 6.5: *Error variance, $\sigma_e^2(p)$, (solid line) and Final Prediction Error (FPE) (dotted line) versus AR model order, p.*

Another example is the Final Prediction Error

$$FPE(p) = \left(\frac{N+p+1}{N-p-1} \right) \sigma_e^2(p) \tag{6.20}$$

where N is the number of samples and $\sigma_e^2(p)$ is the error variance for model order p. Applying this to our data gives the results shown in Figure 6.5 showing an optimal moder order of 3 or 4.

6.3.5 Example: Sleep EEG

As a subject falls from wakefulness into a deep sleep the EEG increases in amplitude and decreases in the frequency of its oscillations. The optimal AR model order also decreases indicating a decrease in complexity. Using FPE Pardey *et al.* show, for example, that wakefulness, REM sleep and deep sleep have typical optimal model orders of 6, 5 and 3 respectively. It should be noted that these are averages and the optimal order has a high variance. Waking EEG shows the highest variance and deep sleep the least.

6.3.6 Discussion

For a comprehensive introduction to AR modelling see Pardey *at al.* [45]. This paper also contains details of other methods for estimating AR coefficients such as the Burg algorithm, which minimises both a *forwards* prediction error and a *backwards* prediction error.

6.4 Moving Average Models

A Moving Average (MA) model of order q is defined as

$$x_t = \sum_{i=0}^{q} b_i e_{t-i} \tag{6.21}$$

where e_t is Gaussian random noise with zero mean and variance σ_e^2. They are a type of FIR filter (see last lecture). These can be combined with AR models to get Autoregressive Moving Average (ARMA) models

$$x_t = \sum_{i=1}^{p} a_i x_{t-i} + \sum_{i=0}^{q} b_i e_{t-i} \tag{6.22}$$

which can be described as an ARMA(p,q) model. They are a type of IIR filter (see last lecture).

Usually, however, FIR and IIR filters have a set of fixed coefficients which have been chosen to give the filter particular frequency characteristics. In MA or ARMA modelling the coefficients are tuned to a particular time series so as to capture the spectral characteristics of the underlying process.

6.5 Spectral Estimation

Autoregressive models can also be used for spectral estimation. An $AR(p)$ model predicts the next value in a time series as a linear combination of the p previous values

$$x_t = -\sum_{k=1}^{p} a_k x_{t-k} + e_t \tag{6.23}$$

where a_k are the AR coefficients and e_t is IID Gaussian noise with zero mean and variance σ^2.

The above equation can be *solved* by assuming that the solution is in the form of an exponential

$$x_t = z^t \tag{6.24}$$

where z is, generally, a complex number. This form of solution has the property that $x_{t-i} = z^{t-i}$; effectively z^{-i} acts as a delay operator denoting a delay of i time steps. This allows the equation to be written

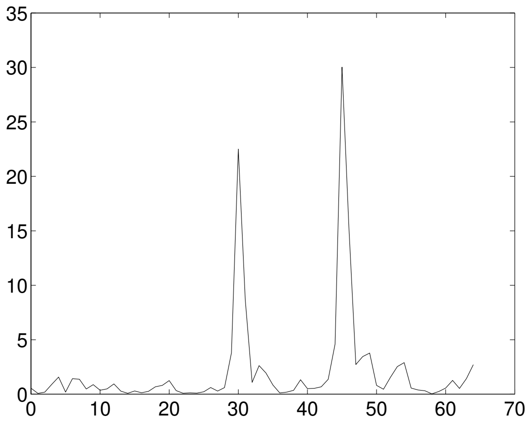
$$a_p z^{t-p} + a_{p-1} z^{t-(p-1)} + ... + z^t = e_t \tag{6.25}$$

It can then be rewritten

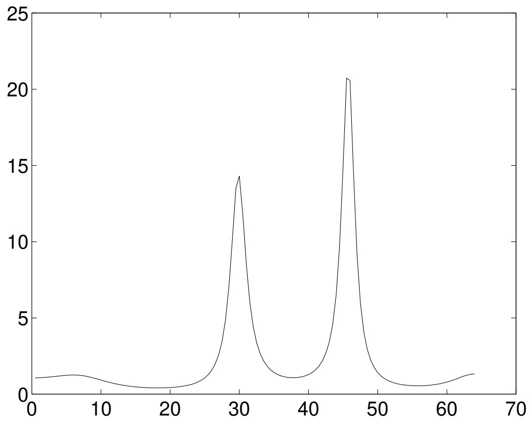
$$z^t = \frac{e_t}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{6.26}$$

Given that any complex number can be written in exponential form

$$z = \exp(i 2\pi f T_s) \tag{6.27}$$



(a)



(b)

Figure 6.6: *Power spectral estimates of two sinwaves in additive noise using (a) Welch's periodogram method and (b) Autoregressive spectral estimation.*

where f is frequency and T_s is the sampling period we can see that the frequency domain characteristics of an AR model are given by (also see Pardey *et al.* [45])

$$P(f) = \frac{\sigma_e^2 T_s}{|1 + \sum_{k=1}^{p} a_k \exp(-ik2\pi f T_s)|^2} \tag{6.28}$$

An AR(p) model can provide spectral estimates with $p/2$ peaks; therefore if you know how many peaks you're looking for in the spectrum you can define the AR model order. Alternatively, AR model order estimation methods should automatically provide the appropriate level of smoothing of the estimated spectrum.

AR spectral estimation has two distinct advantages over methods based on the periodogram (last lecture) (i) power can be estimated over a continuous range of frequencies (not just at fixed intervals) and (ii) the power estimates have less variance.