5 Identification using Prediction Error Methods

The previous chapters have shown how the dynamic behaviour of a civil engineering structure can be modelled by a discrete-time ARMAV model or an equivalent stochastic state space realization. The purpose of this chapter is to investigate how an estimate of a *p*-variate discrete-time model can be obtained from measurements of the response $\boldsymbol{y}(t_k)$. It is assumed that $\boldsymbol{y}(t_k)$ is a realization of a Gaussian stochastic process and thus assumed stationary. The stationarity assumption implies that the behaviour of the underlying system can be represented by a time-invariant model. In the following, the general ARMAV(*na*,*nc*), for $na \geq nc$, model will be addressed. In chapter 2, it was shown that this model can be equivalently represented by an *m*-dimensional state space realization of the innovation form, with $m = na \cdot p$. No matter how the discrete-time model is represented, all adjustable parameters of the model are assembled in a parameter vector $\boldsymbol{\theta}$, implying that the transfer function description of the discrete-time model becomes

$$\begin{aligned} \boldsymbol{y}(t_k) &= \boldsymbol{H}(q,\boldsymbol{\theta})\boldsymbol{e}(t_k) \;, \quad k = 1,\ 2,\ \ldots\ ,\ N \\ E\left[\boldsymbol{e}(t_k)\boldsymbol{e}^T(t_{k+s})\right] &= \boldsymbol{\Lambda}(\boldsymbol{\theta})\delta(s) \end{aligned} \tag{5.1}$$

The dimension of the vectors $\boldsymbol{y}(t_k)$ and $\boldsymbol{e}(t_k)$ are $p \times 1$. Since there is no unique way of computing an estimate of (5.1), it is necessary to select an appropriate approach that balances between the required accuracy and computational effort. The approach used in this thesis is known as the Prediction Error Method (PEM), see Ljung [71]. The reason for this choice is that the PEM for Gaussian distributed prediction errors is asymptotic unbiased and efficient. Further, the use of the PEM enables an estimate of the associated uncertainties of the estimated model parameters. The precision of the PEM does not come for free. It is as such not the most computationally fast approach compared to other estimation techniques. However, in applications such as VBI computational time is not the issue compared to estimation accuracy. In such cases, one would probably rather wait for an accurate estimate instead of using a less accurate but computationally fast approach. The PEM algorithm used in this thesis is of the off-line type, which implies that it uses all available data to update the estimates of the model parameters. This is in contrast to the so-called on-line or recursive algorithms that update the estimated model parameters from sample point to sample point. The choice of the off-line algorithm is reasonable, since the measured data are assumed stationary within a given time series, and since the applied models are time-invariant. At the same time, this algorithm does not suffer the difficulties experienced with the use of the on-line algorithm, with regard to e.g. initialization. A description of the on-line PEM algorithms can be found in Goodwin et al. [31], Ljung [71], Norton [82] and Söderström et al. [105]. Applications on the use of recursive ARMA and ARMAV models for on-line identification of time-varying system can be found in Kirkegaard et al. [55], [56] and [60].

All systems are in principle stochastic, which means that the output $\boldsymbol{y}(t_k)$ at the time k cannot be determined exactly from data available at the time k-1. It is thus valuable to know at the time k-1 what the output $\boldsymbol{y}(t_k)$ of the stochastic process is likely to be at time k.

Therefore, it makes sense to determine the model parameter vector $\boldsymbol{\theta}$ so that the prediction error, defined as

$$\boldsymbol{\varepsilon}(t_k,\boldsymbol{\theta}) = \boldsymbol{y}(t_k) - \hat{\boldsymbol{y}}(t_k|t_{k-1};\boldsymbol{\theta}) \tag{5.2}$$

is as small as possible. $\hat{\boldsymbol{y}}(t_k|t_{k-1};\boldsymbol{\theta})$ is the predicted response at the time k based on the parameters $\boldsymbol{\theta}$, and given available data up to and including the time k-1, i.e.

$$\hat{\boldsymbol{y}}(t_k|t_{k-1};\boldsymbol{\theta}) = \boldsymbol{L}(q,\boldsymbol{\theta})\boldsymbol{y}(t_k) \tag{5.3}$$

where $\boldsymbol{L}(q,\boldsymbol{\theta})$ is a p-variate prediction filter having a pure time delay between $\boldsymbol{y}(t_k)$ and $\hat{\boldsymbol{y}}(t_k|t_{k-1};\boldsymbol{\theta})$. This filter can be constructed in a number of ways, but it will always be based on the model parameters $\boldsymbol{\theta}$. Once the model structure and this filter have been chosen, the prediction errors are calculated from (5.2). In this thesis, system identification of a p-variate stochastically excited linear and time-invariant system is either performed using the p-variate ARMAV(na,nc) model, defined in theorem 2.3 as

$$\begin{aligned}\boldsymbol{y}(t_k) = {} & -\boldsymbol{A}_1\boldsymbol{y}(t_{k-1}) - \ldots - \boldsymbol{A}_{na}\boldsymbol{y}(t_{k-na}) + \\ & \boldsymbol{e}(t_k) + \boldsymbol{C}_1\boldsymbol{e}(t_{k-1}) + \ldots + \boldsymbol{C}_{nc}\boldsymbol{e}(t_{k-nc}) \\ E\left[\boldsymbol{e}(t_k)\boldsymbol{e}^T(t_{k+s})\right] = {} & \boldsymbol{\Lambda}(\boldsymbol{\theta})\delta(s)\end{aligned} \tag{5.4}$$

or by using the innovation state space system

$$\begin{aligned}\hat{\boldsymbol{x}}(t_{k+1}|t_k;\boldsymbol{\theta}) = {} & \boldsymbol{A}(\boldsymbol{\theta})\hat{\boldsymbol{x}}(t_k|t_{k-1};\boldsymbol{\theta}) + \boldsymbol{K}(\boldsymbol{\theta})\boldsymbol{e}(t_k) \\ \boldsymbol{y}(t_k) = {} & \boldsymbol{C}(\boldsymbol{\theta})\hat{\boldsymbol{x}}(t_k|t_{k-1};\boldsymbol{\theta}) + \boldsymbol{e}(t_k) \\ E\left[\boldsymbol{e}(t_k)\boldsymbol{e}^T(t_{k+s})\right] = {} & \boldsymbol{\Lambda}(\boldsymbol{\theta})\delta(s)\end{aligned} \tag{5.5}$$

The transfer function of the ARMAV(na,nc) model is defined as

$$\begin{aligned}\boldsymbol{H}(q,\boldsymbol{\theta}) = {} & \boldsymbol{A}^{-1}(q,\boldsymbol{\theta})\boldsymbol{C}(q,\boldsymbol{\theta}) \\ \boldsymbol{A}(q,\boldsymbol{\theta}) = {} & \boldsymbol{I} + \boldsymbol{A}_1 q^{-1} + \ldots + \boldsymbol{A}_{na} q^{-na} \\ \boldsymbol{C}(q,\boldsymbol{\theta}) = {} & \boldsymbol{I} + \boldsymbol{C}_1 q^{-1} + \ldots + \boldsymbol{C}_{nc} q^{-nc}\end{aligned} \tag{5.6}$$

and the transfer function of the innovation state space system is defined as

$$\boldsymbol{H}(q,\theta) = \boldsymbol{C}(\theta)\big(\boldsymbol{I}q - \boldsymbol{A}(\theta)\big)^{-1}\boldsymbol{K}(\theta) + \boldsymbol{I} \tag{5.7}$$

Having chosen the model structure, it is necessary to make the following choices to define a prediction error method for minimization of $\boldsymbol{\varepsilon}(t_k,\theta)$.

- *Choice of prediction filter* $\boldsymbol{L}(q,\theta)$.
- *Choice of a scalar-valued function that can assess the performance of the predictor.*
- *Choice of procedure for minimization of this performance function.*

The first three sections of this chapter describe how to make the three choices listed above. Section 5.4 concerns, organization of the parameters of the chosen model structure in the parameter vector θ, and describes how the gradient of the chosen prediction filter is calculated. In section 5.5, the statistical properties of the prediction error method are analysed. These properties are statistical asymptotic efficiency and consistency of the PEM estimate, and asymptotic distribution of the estimated parameters. Section 5.6 deals with the problems of obtaining reliable initial estimates of the model parameters used in the PEM. Finally, in section 5.7, guidelines for validation of an identified model and model order selection will be provided.

5.1 Choice of Prediction Filter

In this section it is shown how the prediction filter may appear when the model structure is the ARMAV model or the innovation state space system. If the model structure is the *p*-variate ARMAV(*na*,*nc*) model shown in (5.4) then the prediction filter may appear as in the following theorem.

Theorem 5.1 - The Prediction Filter of the ARMAV Model

If the ARMAV model is represented by a transfer function description as in (5.6) then the prediction filter $\boldsymbol{L}(q,\theta)$ in (5.3) is given by

$$\boldsymbol{L}(q,\theta) = \boldsymbol{I} - \boldsymbol{H}^{-1}(q,\theta) \tag{5.8}$$

The filter will only be asymptotically stable if all eigenvalues of $\boldsymbol{H}^{-1}(q,\theta)$ are inside the complex unit circle, see Söderström et al. [105].

Proof:

From (5.1), it is seen that

$$\boldsymbol{y}(t_k) = \boldsymbol{H}(q,\theta)\boldsymbol{e}(t_k) = \big(\boldsymbol{H}(q,\theta) - \boldsymbol{I}\big)\boldsymbol{e}(t_k) + \boldsymbol{e}(t_k) \tag{5.9}$$

By assuming that $\boldsymbol{H}(q,\theta)$ is invertible the knowledge of the response $\boldsymbol{y}(t_s)$, for $s \le k$-1 implies the knowledge $\boldsymbol{e}(t_s)$ for $s \le k$-1. In other words, $\boldsymbol{e}(t_k)$ can be replaced by $\boldsymbol{y}(t_k)$ using (5.1). The

first term of (5.9) is therefore known at the time t_{k-1}. Since the $\boldsymbol{e}(t_k)$ has zero mean the conditional expectation of (5.9) with respect to $\boldsymbol{y}(t_s)$, for $s \le k$-1, is given by

$$\begin{aligned}\hat{\boldsymbol{y}}(t_k|t_{k-1};\boldsymbol{\theta}) &= \big(\boldsymbol{H}(q,\boldsymbol{\theta}) - \boldsymbol{I}\big)\boldsymbol{e}(t_k) \\ &= \big(\boldsymbol{H}(q,\boldsymbol{\theta}) - \boldsymbol{I}\big)\boldsymbol{H}^{-1}(q,\boldsymbol{\theta})\boldsymbol{y}(t_k) \\ &= \big(\boldsymbol{I} - \boldsymbol{H}^{-1}(q,\boldsymbol{\theta})\big)\boldsymbol{y}(t_k) \\ &= \boldsymbol{L}(q,\boldsymbol{\theta})\boldsymbol{y}(t_k)\end{aligned} \tag{5.10}$$

which proves the theorem. □

If the prediction filter (5.8) is combined with (5.2) the predictor of the model appears as

$$\begin{aligned}\boldsymbol{\varepsilon}(t_k,\boldsymbol{\theta}) &= \boldsymbol{y}(t_k) - \hat{\boldsymbol{y}}(t_k|t_{k-1};\boldsymbol{\theta}) \\ &= \boldsymbol{y}(t_k) - \big(\boldsymbol{I} - \boldsymbol{H}^{-1}(q,\boldsymbol{\theta})\big)\boldsymbol{y}(t_k) \\ &= \boldsymbol{H}^{-1}(q,\boldsymbol{\theta})\boldsymbol{y}(t_k)\end{aligned} \tag{5.11}$$

Combining this result with the fact that $\boldsymbol{H}^{-1}(q,\boldsymbol{\theta}) = \boldsymbol{C}^{-1}(q,\boldsymbol{\theta})\boldsymbol{A}(q,\boldsymbol{\theta})$ for the ARMAV(*na*,*nc*) model yields

$$\begin{aligned}\hat{\boldsymbol{y}}(t_k|t_{k-1};\boldsymbol{\theta}) &= \boldsymbol{y}(t_k) - \boldsymbol{\varepsilon}(t_k,\boldsymbol{\theta}) \\ &= -\boldsymbol{A}_1(\boldsymbol{\theta})\boldsymbol{y}(t_{k-1}) - \ldots - \boldsymbol{A}_{na}(\boldsymbol{\theta})\boldsymbol{y}(t_{k-na}) + \\ &\quad \boldsymbol{C}_1(\boldsymbol{\theta})\boldsymbol{\varepsilon}(t_{k-1},\boldsymbol{\theta}) + \ldots + \boldsymbol{C}_{nc}(\boldsymbol{\theta})\boldsymbol{\varepsilon}(t_{k-nc},\boldsymbol{\theta})\end{aligned} \tag{5.12}$$

This relation reveals that the predictor of the ARMAV model is nonlinear, since the prediction errors themselves depend on the parameter vector $\boldsymbol{\theta}$. This indicates that an iterative minimization procedure has to be applied.

Now let (5.1) be represented by the innovation state space system (5.5). The prediction filter may for this representation of (5.1) be given by the following theorem.

Theorem 5.2 - The Prediction Filter of the Innovation State Space System

The predictor of the innovation state space system (5.5) is given by

$$\begin{aligned}\hat{\boldsymbol{x}}(t_{k+1}|t_k;\boldsymbol{\theta}) &= \big(\boldsymbol{A}(\boldsymbol{\theta}) - \boldsymbol{K}(\boldsymbol{\theta})\boldsymbol{C}(\boldsymbol{\theta})\big)\hat{\boldsymbol{x}}(t_k|t_{k-1};\boldsymbol{\theta}) + \boldsymbol{K}(\boldsymbol{\theta})\boldsymbol{y}(t_k) \\ \hat{\boldsymbol{y}}(t_k|t_{k-1};\boldsymbol{\theta}) &= \boldsymbol{C}(\boldsymbol{\theta})\hat{\boldsymbol{x}}(t_k|t_{k-1};\boldsymbol{\theta})\end{aligned} \tag{5.13}$$

Since the input in (5.13) is $\boldsymbol{y}(t_k)$, and the output is $\hat{\boldsymbol{y}}(t_k|t_{k-1};\theta)$, the system is an internal description of $\boldsymbol{L}(q,\theta)$. By introduction of the shift operator q the prediction filter can formally be represented by

$$\boldsymbol{L}(q,\theta) = \boldsymbol{C}(\theta)\big(\boldsymbol{I}q - \boldsymbol{A}(\theta) + \boldsymbol{K}(\theta)\boldsymbol{C}(\theta)\big)^{-1}\boldsymbol{K}(\theta) \tag{5.14}$$

This filter will only be asymptotically stable if the eigenvalues of the matrix $\boldsymbol{A}(\theta)$-$\boldsymbol{K}(\theta)\boldsymbol{C}(\theta)$ lie inside the complex unit circle, see Söderström et al. [105].

Proof:

The innovation state space representation in (5.5) is derived from the steady-state Kalman filter, see theorem 2.2,

$$\begin{aligned}\hat{\boldsymbol{x}}(t_{k+1}|t_k;\theta) &= \boldsymbol{A}(\theta)\hat{\boldsymbol{x}}(t_k|t_{k-1};\theta) + \\ &\quad \boldsymbol{K}(\theta)\big(\boldsymbol{y}(t_k) - \boldsymbol{C}(\theta)\hat{\boldsymbol{x}}(t_k|t_{k-1};\theta)\big) \\ \hat{\boldsymbol{y}}(t_k|t_{k-1};\theta) &= \boldsymbol{C}(\theta)\hat{\boldsymbol{x}}(t_k|t_{k-1};\theta)\end{aligned} \tag{5.15}$$

which by rearrangement of the state equation looks as (5.13). □

In both cases the predictors are mean square optimal if the prediction errors are Gaussian distributed, see e.g. Goodwin et al. [31], Ljung [71] and Söderström et al. [105]. It should be pointed out that $\boldsymbol{\varepsilon}(t_k,\theta)$ is assumed to be Gaussian white noise in the above derivations. However, it is possible to compute and apply $\boldsymbol{L}(q,\theta)$ even if the Gaussian assumption is not satisfied by the data. However, if $\boldsymbol{\varepsilon}(t_k,\theta)$ is no longer Gaussian white noise, then the prediction filter is no longer optimal. In any case, given a model structure and its parameters in a vector θ, it is now possible to construct a prediction filter $\boldsymbol{L}(q,\theta)$. Based on this filter and available data $\boldsymbol{y}(t_k)$ the prediction errors $\boldsymbol{\varepsilon}(t_k,\theta)$ can now be computed. The next step in the derivation of a PEM algorithm is the definition of a criterion function.

5.2 Choice of Criterion Function

The criterion function has to be a scalar-valued function of all prediction errors $\boldsymbol{\varepsilon}(t_k,\theta)$, $k = 1$ to N. The purpose of this function is to assess the performance of the predictor. If the prediction errors $\boldsymbol{\varepsilon}(t_k,\theta)$ are assumed to be Gaussian distributed, an optimal set of parameters θ must minimize the covariance matrix of $\boldsymbol{\varepsilon}(t_k,\theta)$. It is therefore reasonable to base the minimization of the prediction errors on the following function $\boldsymbol{Q}_N(\theta)$, see Söderström et al. [105]

$$\boldsymbol{Q}_N(\theta) = \frac{1}{N}\sum_{k=1}^{N}\boldsymbol{\varepsilon}(t_k,\theta)\boldsymbol{\varepsilon}^T(t_k,\theta) \tag{5.16}$$

However, it is only possible to use $\boldsymbol{Q}_N(\boldsymbol{\theta})$ directly as a criterion function, when the prediction errors constitute a scalar sequence. In the general case, a monotonically increasing function $h(\boldsymbol{Q}_N(\boldsymbol{\theta}))$ that maps $\boldsymbol{Q}_N(\boldsymbol{\theta})$ as a scalar value must be chosen. A choice of a monotonically increasing function is

$$h(\boldsymbol{Q}_N(\boldsymbol{\theta})) = det\left(\boldsymbol{Q}_N(\boldsymbol{\theta})\right) \tag{5.17}$$

This function is preferred, because it corresponds to the criterion function of the maximum likelihood method for Gaussian distributed prediction errors, see Akaike [2], Box et al. [16] and Ljung [71]. The off-line parameters estimate, based on N samples, and returned in $\hat{\boldsymbol{\theta}}_N$ is therefore obtained as the global minimum point of the criterion function (5.17), i.e.

$$\begin{aligned}\hat{\boldsymbol{\theta}}_N &= arg \min_{\boldsymbol{\theta}} det\left(\frac{1}{N}\sum_{k=1}^{N}\boldsymbol{\varepsilon}(t_k,\boldsymbol{\theta})\boldsymbol{\varepsilon}^T(t_k,\boldsymbol{\theta})\right)\\ &= arg \min_{\boldsymbol{\theta}} V_N(\boldsymbol{\theta})\end{aligned} \tag{5.18}$$

In (5.18) *arg* min should be read as the argument that minimizes the criterion function. Notice that the criterion function for short is denoted $V_N(\boldsymbol{\theta})$. The criterion function is sometimes referred to as the loss function, since it is a measure of the loss of information contained in the data.

5.3 Choice of Minimization Procedure

The question now is how to satisfy (5.18), i.e. how to minimize the loss function $V_N(\boldsymbol{\theta})$. As shown in section 5.1, the prediction filter as well as the prediction errors depend on currently estimated parameters, implying that the search for a minimum becomes nonlinear. The applied search-scheme is of the full-Newton or Newton-Raphson type, see e.g. Ljung [71], Söderström et al. [105] and Vanderplaats [108]. It updates the estimate of $n \times 1$ parameter vector $\hat{\boldsymbol{\theta}}_N^i$ of the ith iteration to a new estimate $\hat{\boldsymbol{\theta}}_N^{i+1}$ using the following update procedure

$$\hat{\boldsymbol{\theta}}_N^{i+1} = \hat{\boldsymbol{\theta}}_N^i - \mu_i \ddot{\boldsymbol{V}}_N^{-1}(\hat{\boldsymbol{\theta}}_N^i)\dot{\boldsymbol{V}}_N^T(\hat{\boldsymbol{\theta}}_N^i) \tag{5.19}$$

where $\dot{\boldsymbol{V}}_N(\boldsymbol{\theta})$ is the $1 \times n$ gradient vector and $\ddot{\boldsymbol{V}}_N(\boldsymbol{\theta})$ the $n \times n$ Hessian matrix of the criterion function $V_N(\boldsymbol{\theta})$. In strict theoretical sense the coefficient μ_i is equal to 1. However, in practice the step size is often too long and needs adjustment. Therefore μ_i should be chosen as the argument that minimizes the following relation, Söderström et al. [105]

$$\mu_i = arg \min_{\mu} V\left(\hat{\boldsymbol{\theta}}_N^i - \mu \ddot{\boldsymbol{V}}_N^{-1}(\hat{\boldsymbol{\theta}}_N^i)\dot{\boldsymbol{V}}_N(\hat{\boldsymbol{\theta}}_N^i)\right) \tag{5.20}$$

Besides the use for adjustment of the step size, μ_i can also be used to ensure that the eigenvalues of the prediction filter stay inside the complex unit circle. The computational burden of the search procedure is the calculation of the gradient and of the Hessian matrix of the criterion function, and the only algebraic difference between the estimation of the ARMAV model and the corresponding state space representation is the calculation of the gradient of the predictor.

5.3.1 Gradient and Hessian of the Criterion Function

As shown in (5.19), it is necessary to calculate both the gradient and the Hessian of the criterion function $V_N(\boldsymbol{\theta})$. Straightforward differentiation of $V_N(\boldsymbol{\theta})$ in (5.18) with respect to the parameters in $\boldsymbol{\theta}$ yields, see Söderström et al. [105]

$$\dot{\boldsymbol{V}}_N(\boldsymbol{\theta}) = -2V_N(\boldsymbol{\theta})\frac{1}{N}\sum_{k=1}^{N}\boldsymbol{\varepsilon}^T(t_k,\boldsymbol{\theta})\boldsymbol{Q}_N^{-1}(\boldsymbol{\theta})\boldsymbol{\psi}^T(t_k,\boldsymbol{\theta}) \tag{5.21}$$

where the $n \times p$ matrix $\boldsymbol{\psi}(t_k,\boldsymbol{\theta})$ is the gradient of the predictor $\hat{\boldsymbol{y}}(t_k|t_{k-1};\boldsymbol{\theta})$, defined as

$$\boldsymbol{\psi}(t_k,\boldsymbol{\theta}) = \frac{\partial\hat{\boldsymbol{y}}(t_k|t_{k-1};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} \tag{5.22}$$

It should be noted, that in (5.21) the following differential rule has been used

$$\frac{\partial V_N(\boldsymbol{\theta})}{\partial \boldsymbol{Q}_N(\boldsymbol{\theta})} = V_N(\boldsymbol{\theta})\boldsymbol{Q}_N^{-1}(\boldsymbol{\theta}) \tag{5.23}$$

which only applies to the criterion (5.17). The calculation of $\boldsymbol{\psi}(t_k,\boldsymbol{\theta})$ marks the crucial numerical difference between the estimation of the ARMAV model and the innovation state space representation. An approximate expression for the $n \times n$ Hessian matrix $\ddot{\boldsymbol{V}}_N(\boldsymbol{\theta})$ is given by, Söderström et al. [105]

$$\ddot{\boldsymbol{V}}_N(\boldsymbol{\theta}) \approx 2V_N(\boldsymbol{\theta})\frac{1}{N}\sum_{k=1}^{N}\boldsymbol{\psi}(t_k,\boldsymbol{\theta})\boldsymbol{Q}_N^{-1}(\boldsymbol{\theta})\boldsymbol{\psi}^T(t_k,\boldsymbol{\theta}) \tag{5.24}$$

There are two reasons for this approximation. First of all, by construction, it is guaranteed that $\ddot{\boldsymbol{V}}_N(\boldsymbol{\theta})$ is positive definite. Secondly, it is simple to compute this approximated Hessian. The convergence will only be quadratic close to the minimum point. Far from minimum, it will be linear but with fast convergence, see Söderström et al. [105]. If this approximation is used in (5.19) the Gauss-Newton search scheme is obtained.

Definition 5.1 - The Gauss-Newton Search Scheme

If the gradient (5.21) of the criterion function is inserted into (5.19), together with the approximated Hessian in (5.24), the Gauss-Newton search-scheme

$$\hat{\boldsymbol{\theta}}_N^{i+1} = \hat{\boldsymbol{\theta}}_N^{i} + \mu_i \boldsymbol{R}_N^{-1}(\hat{\boldsymbol{\theta}}_N^{i}) \boldsymbol{F}_N(\hat{\boldsymbol{\theta}}_N^{i}) \tag{5.25}$$

is obtained. The $n \times n$ matrix $\boldsymbol{R}_N(\theta)$ and the $n \times 1$ vector $\boldsymbol{F}_N(\theta)$ given by

$$\begin{aligned} \boldsymbol{R}_N(\boldsymbol{\theta}) &= \sum_{k=1}^{N} \boldsymbol{\psi}(t_k,\boldsymbol{\theta}) \boldsymbol{Q}_N^{-1}(\boldsymbol{\theta}) \boldsymbol{\psi}^T(t_k,\boldsymbol{\theta}) \\ \boldsymbol{F}_N(\boldsymbol{\theta}) &= \sum_{k=1}^{N} \boldsymbol{\psi}(t_k,\boldsymbol{\theta}) \boldsymbol{Q}_N^{-1}(\boldsymbol{\theta}) \boldsymbol{\varepsilon}(t_k,\boldsymbol{\theta}) \end{aligned} \tag{5.26}$$

and μ_i chosen according to (5.20). □

Because the approximated Hessian is guaranteed positive definite, the criterion function will decrease in every iteration by proper choice of μ_i. However, if the model is overparameterized, i.e. has too many parameters compared to what is actually needed, the approximated Hessian might be close to singular and therefore difficult to invert. Various techniques have been proposed in order to overcome this problem. A well-known technique is the Marquardt-Levenberg search scheme, Levenberg [70] and Marquardt [74]. However, if a good initial estimate of $\boldsymbol{\theta}$ is available, it is the experience that the Gauss-Newton search-scheme works very well for properly selected model structures. This search scheme is standard in the *System Identification* toolbox for use with MATLAB, see Ljung [72], and has been used in several identification application, see e.g. Andersen et al. [6], Asmussen et al. [9], Brincker et al. [17], Kirkegaard et al. [56] and [57].

5.4 Calculating the Predictor Gradient

As stated previously, the only significant difference between estimation of the ARMAV model and the innovation state space representation is the way the predictor $\hat{\boldsymbol{y}}(t_k|t_{k-1};\boldsymbol{\theta})$ and its gradient $\boldsymbol{\psi}(t_k,\boldsymbol{\theta})$ are calculated. In (5.3), it was shown that $\hat{\boldsymbol{y}}(t_k|t_{k-1};\boldsymbol{\theta})$ could be obtained by filtering the data through a prediction filter as

$$\hat{\boldsymbol{y}}(t_k|t_{k-1};\boldsymbol{\theta}) = \boldsymbol{L}(q,\boldsymbol{\theta}) \boldsymbol{y}(t_k) \tag{5.27}$$

implying that the n rows of $\boldsymbol{\psi}(t_k,\boldsymbol{\theta})$ are obtained from

$$\boldsymbol{\psi}_i^T(t_k,\boldsymbol{\theta}) = \frac{\partial \boldsymbol{L}(q,\boldsymbol{\theta})}{\partial \theta_i} \boldsymbol{y}(t_k) \ , \quad i = 1,\ 2,\ \ldots\ ,\ n \tag{5.28}$$

However, before the gradient can be calculated, it is necessary to determine how the parameters of the model structures are organized in the parameter vector $\boldsymbol{\theta}$.

5.4.1 The Predictor Gradient of the ARMAV Model

The p-variate ARMAV(na,nc) model was defined in (5.4). For simplicity, it is assumed that all parameters of the $p \times p$ coefficient matrices $\boldsymbol{A}_i$ and $\boldsymbol{C}_i$ are adjustable. The $n \times 1$ parameter vector $\boldsymbol{\theta}$ is organised in the following way

$$\boldsymbol{\theta} = col\left(\left[\boldsymbol{A}_1 \ . \ . \ \boldsymbol{A}_{na} \ , \ \boldsymbol{C}_1 \ . \ . \ \boldsymbol{C}_{nc}\right]\right) \tag{5.29}$$

where *col* means stacking of all columns of the argument matrix. The total number of adjustable parameters in $\boldsymbol{\theta}$ is as such $n = (na+nc)p^2$.

Denote the (i,j)th element of the kth auto-regressive coefficient matrix $\boldsymbol{A}_k$ by $a_{i,j,k}$. This element is recovered as element number $(k\text{-}1)p^2+(j\text{-}1)p+i$ of $\boldsymbol{\theta}$. In a similar manner, the (i,j)th element of the kth moving average coefficient matrix $\boldsymbol{C}_k$, denoted $c_{i,j,k}$, can be recovered as element number $(na+k\text{-}1)p^2+(j\text{-}1)p+i$ of $\boldsymbol{\theta}$.

The predictor of the ARMAV model is given by (5.27), with the prediction filter defined in (5.8). If the definition $\boldsymbol{H}(q,\boldsymbol{\theta}) = \boldsymbol{A}^{-1}(q,\boldsymbol{\theta})\boldsymbol{C}(q,\boldsymbol{\theta})$ is inserted, the predictor can be formulated as

$$\boldsymbol{C}(q,\boldsymbol{\theta})\hat{\boldsymbol{y}}(t_k|t_{k-1};\boldsymbol{\theta}) = \left(\boldsymbol{C}(q,\boldsymbol{\theta}) - \boldsymbol{A}(q,\boldsymbol{\theta})\right)\boldsymbol{y}(t_k) \tag{5.30}$$

Introduce a $p \times p$ matrix $\boldsymbol{Q}_{j,k}$ whose elements are all zero except the element (j,k) that is equal to 1. Further, define *int* $|x|$ as an operator that returns the integral part of the argument x. The ith row of $\boldsymbol{\psi}(t_k,\boldsymbol{\theta})$ can then be found by partial differentiation of (5.30) with respect to $\boldsymbol{\theta}_i$. If $\boldsymbol{\theta}_i$ is one of the auto-regressive parameters then $\boldsymbol{\psi}_i(t_k,\boldsymbol{\theta})$ is given by

$$\begin{aligned}
&\boldsymbol{\psi}_i^T(t_k,\boldsymbol{\theta}) = -\boldsymbol{C}^{-1}(q,\boldsymbol{\theta})\boldsymbol{Q}_{r,c}\boldsymbol{y}(t_{k-d}) \\
&i = 1, \ \ldots \ , \ na\cdot p^2 \\
&d = int\left|\frac{i}{p^2}\right| + 1 \\
&c = int\left|\frac{1 - (d-1)p^2}{p}\right| + 1 \\
&r = 1 - (d-1)p^2 - (c-1)p
\end{aligned} \tag{5.31}$$

However, if θ_i is one of the moving average parameters then $\psi_i(t_k,\theta)$ is given by

$$\psi_i^T(t_k,\theta) = C^{-1}(q,\theta)Q_{r,c}\,\varepsilon(t_{k-d},\theta)$$

$$i = na\cdot p^2 + 1,\ \ldots\ ,\ (na+nc)p^2$$

$$d = int\left|\frac{i}{p^2}\right| - na + 1 \qquad (5.32)$$

$$c = int\left|\frac{1 - (na + d - 1)p^2}{p}\right| + 1$$

$$r = 1 - (na + d - 1)p^2 - (c - 1)p$$

In both cases, it is seen that $\psi_i(t_k,\theta)$ is obtained as the output of a multivariate filter. The computation can as such be done in a fast and efficient manner. It should also be noted that $\psi_i^T(t_k,\theta)$ simply is $\psi_{i+p^2}^T(t_{k-1},\theta)$, i.e. $\psi_i(t_k,\theta)$ is obtained directly from $\psi_{i+p^2}(t_{k-1},\theta)$ without any computation. This means that each of the filters (5.31) and (5.32) only needs to be applied p^2 times, i.e. exactly the number of times that correspond to the number of elements of $Q_{j,k}$.

5.4.2 The Predictor Gradient of the State Space System

If the model structure is of the innovation state space form, it is necessary to decide on the realization for use. For illustrative purposes the observability canonical state space realization of the p-variate ARMAV(na,nc), with $na \geq nc$, will be used, see theorem 2.5. For this particular realization of the innovation state space system (5.5), the matrices $A(\theta)$ and $K(\theta)$ which contain all the adjustable parameters, are given by

$$A(\theta) = \begin{bmatrix} 0 & I & . & . & 0 & 0 \\ 0 & 0 & . & . & 0 & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & . & . & 0 & I \\ -A_{na} & -A_{na-1} & . & . & -A_2 & -A_1 \end{bmatrix}, \quad K(\theta) = \begin{bmatrix} h(1) \\ h(2) \\ . \\ . \\ h(na-1) \\ h(na) \end{bmatrix} \qquad (5.33)$$

The matrix $C(\theta)$ only contains trivial parameters as

$$C(\theta) = C = \begin{bmatrix} I & 0 & . & . & 0 & 0 \end{bmatrix} \qquad (5.34)$$

The dimension of $\boldsymbol{A}(\boldsymbol{\theta})$ is $na \cdot p \times na \cdot p$. All non-trivial and adjustable parameters of this matrix are located in the last p rows. The dimension of $\boldsymbol{K}(\boldsymbol{\theta})$ is $na \cdot p \times p$ and all parameters of this matrix are adjustable. The total number of adjustable parameters of this realization is as such $n = 2na \cdot p^2$. These are assembled in the $n \times 1$ parameter vector $\boldsymbol{\theta}$ as

$$\boldsymbol{\theta} = \begin{bmatrix} col\left(\boldsymbol{\theta}^{\boldsymbol{A}}\right) \\ col\left(\boldsymbol{\theta}^{\boldsymbol{K}}\right) \end{bmatrix} \tag{5.35}$$

where

$$\boldsymbol{\theta}^{\boldsymbol{A}} = \begin{bmatrix} -\boldsymbol{A}_{na} & -\boldsymbol{A}_{na-1} & . & . & -\boldsymbol{A}_2 & -\boldsymbol{A}_1 \end{bmatrix}, \quad \boldsymbol{\theta}^{\boldsymbol{K}} = \begin{bmatrix} \boldsymbol{h}(1) \\ \boldsymbol{h}(2) \\ . \\ . \\ \boldsymbol{h}(na-1) \\ \boldsymbol{h}(na) \end{bmatrix} \tag{5.36}$$

It should be noted, that the first half of the parameters of $\boldsymbol{\theta}$ only relates to $\boldsymbol{A}(\boldsymbol{\theta})$, whereas the second half only relates to $\boldsymbol{K}(\boldsymbol{\theta})$. Having estimated $\boldsymbol{\theta}$ the corresponding ARMAV model can be determined by use of theorem 2.3.

Denote the partial derivative of $\hat{\boldsymbol{x}}(t_k|t_{k-1};\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}_i$ by $\boldsymbol{\Psi}_i(t_k|t_{k-1};\boldsymbol{\theta})$. Consider the parameters of the first half of $\boldsymbol{\theta}$. Introduce the $na \cdot p \times na \cdot p$ matrix $\boldsymbol{Q}^{\boldsymbol{A}}_{r,c}$ whose elements are all zero except element (r,c) which is equal to 1. The gradient $\boldsymbol{\psi}_i(t_k,\boldsymbol{\theta})$, defined in (5.28), is then obtained by differentiation of (5.13) with respect to $\boldsymbol{\theta}_i$ as

$$\begin{aligned} \boldsymbol{\Psi}_i(t_{k+1},\boldsymbol{\theta}) &= \left(\boldsymbol{A}(\boldsymbol{\theta}) - \boldsymbol{K}(\boldsymbol{\theta})\boldsymbol{C}\right)\boldsymbol{\Psi}_i(t_k,\boldsymbol{\theta}) + \boldsymbol{Q}^{\boldsymbol{A}}_{r,c}\hat{\boldsymbol{x}}(t_k|t_{k-1};\boldsymbol{\theta}) \\ \boldsymbol{\psi}_i^T(t_k,\boldsymbol{\theta}) &= \boldsymbol{C}\boldsymbol{\Psi}_i(t_k,\boldsymbol{\theta}) \\ \\ i &= 1, \ \ldots \ , \ na \cdot p^2 \\ c &= int\left|\frac{i-1}{p}\right| + 1 \\ r &= \left(na - c\right)p + i \end{aligned} \tag{5.37}$$

Now consider the parameters of the second half of $\boldsymbol{\theta}$. Introduce the $na \cdot p \times p$ matrix $\boldsymbol{Q}^{\boldsymbol{K}}_{r,c}$ whose elements are all zero except element (r,c) which is equal to 1. The gradient $\boldsymbol{\psi}_i(t_k,\boldsymbol{\theta})$ is then given by

$$\boldsymbol{\Psi}_i(t_{k+1},\boldsymbol{\theta}) = \big(\boldsymbol{A}(\boldsymbol{\theta}) - \boldsymbol{K}(\boldsymbol{\theta})\boldsymbol{C}\big)\boldsymbol{\Psi}_i(t_k,\boldsymbol{\theta}) + \boldsymbol{Q}_{r,c}^{K}\,\boldsymbol{\varepsilon}(t_k,\boldsymbol{\theta})$$

$$\boldsymbol{\psi}_i^T(t_k,\boldsymbol{\theta}) = \boldsymbol{C}\boldsymbol{\Psi}_i(t_k,\boldsymbol{\theta})$$

$$i = na\cdot p^2 + 1,\ldots,2na\cdot p^2 \tag{5.38}$$

$$c = int\left|\frac{i - na\cdot p^2 - 1}{na\cdot p}\right| + 1$$

$$r = \big(1 - c - p\big)na\cdot p + i$$

Each of the state space systems (5.37) and (5.38) has to be applied $na\cdot p^2$ times in order to construct $\boldsymbol{\psi}(t_k,\boldsymbol{\theta})$. By now it is possible to construct a complete PEM algorithm that implements the Gauss-Newton search-scheme. However, to appreciate the PEM approach fully some of the statistical properties associated with it will be considered in the following section. The actual implementation is discussed in chapter 7. The PEM approach using the Gauss-Newton search scheme is also implemented in Ljung [71], for several different model structures, including the ARMA model.

5.5 Properties of the Prediction Error Method

In this section, it will be shown that under a certain assumption the estimated parameters possess some appealing statistical properties. As an introduction to these topics the uniqueness of an estimated ARMAV model will be considered.

5.5.1 Uniqueness of the Model Structure Parametrization

Define the true linear and discrete-time system *S* as

$$\boldsymbol{y}(t_k) = \boldsymbol{H}_0(q)\boldsymbol{e}_0(t_k)\ , \quad E\Big[\boldsymbol{e}_0(t_k)\boldsymbol{e}_0^T(t_{k+s})\Big] = \boldsymbol{\Lambda}_0\,\delta(s) \tag{5.39}$$

Introduce the set $D_T(S,M)$ as

$$D_T(S,M) = \Big\{\boldsymbol{\theta} \mid \boldsymbol{H}_0(q) \equiv \boldsymbol{H}(q,\boldsymbol{\theta}),\ \boldsymbol{\Lambda}_0 \equiv \boldsymbol{\Lambda}(\boldsymbol{\theta})\Big\} \tag{5.40}$$

and let it consist of those parameter vectors $\boldsymbol{\theta}$ for which a chosen model structure *M* gives a perfect description of the system, i.e. $S = M(\boldsymbol{\theta})$. The set might be empty or consist of several points. However, in the following it will be assumed that $D_T(S,M)$ only includes one parameter vector $\boldsymbol{\theta}_0$, that satisfies $S = M(\boldsymbol{\theta}_0)$. If $D_T(S,M)$ consists of several points, it implies that several models satisfy $S = M(\boldsymbol{\theta})$. This situation is

encountered in cases where model structures are overparameterized, see Ljung [71] and Söderström et al. [105]. The question is when a model structure is over-parameterized. To answer this, consider the ARMAV model defined in (5.4)

$$\boldsymbol{A}(q,\boldsymbol{\theta})\boldsymbol{y}(t_k) = \boldsymbol{C}(q,\boldsymbol{\theta})\boldsymbol{e}(t_k)\ , \quad E\left[\boldsymbol{e}(t_k)\boldsymbol{e}^T(t_{k+s})\right] = \boldsymbol{\Lambda}(\boldsymbol{\theta})\delta(s) \tag{5.41}$$

and let the true system be given by

$$\begin{aligned} &\boldsymbol{A}_0(q)\boldsymbol{y}(t_k) = \boldsymbol{C}_0(q)\boldsymbol{e}_0(t_k)\ , \quad E\left[\boldsymbol{e}_0(t_k)\boldsymbol{e}_0^T(t_{k+s})\right] = \boldsymbol{\Lambda}_0\delta(s) \\ &\boldsymbol{A}_0(q) = \boldsymbol{I} + \boldsymbol{A}_{0,1}q^{-1} + \ldots + \boldsymbol{A}_{0,na_0}q^{-na_0} \\ &\boldsymbol{C}_0(q) = \boldsymbol{I} + \boldsymbol{C}_{0,1}q^{-1} + \ldots + \boldsymbol{C}_{0,nc_0}q^{-nc_0} \end{aligned} \tag{5.42}$$

The identities in (5.40) that define the set $D_T(S,M)$ then become

$$\boldsymbol{A}_0^{-1}(q)\boldsymbol{C}_0(q) = \boldsymbol{A}^{-1}(q,\boldsymbol{\theta})\boldsymbol{C}(q,\boldsymbol{\theta})\ , \quad \boldsymbol{\Lambda}_0\delta(s) = \boldsymbol{\Lambda}(\boldsymbol{\theta})\delta(s) \tag{5.43}$$

It is assumed that $\boldsymbol{A}_0(q)$ and $\boldsymbol{C}_0(q)$ are left coprime, i.e. their greatest common divisor is an unimodal matrix, implying that $na_0 \geq nc_0$. For (5.43) to have a solution, it is necessary that $na \geq na_0$ and $nc \geq nc_0$, and $\boldsymbol{A}(q,\boldsymbol{\theta})$ and $\boldsymbol{C}(q,\boldsymbol{\theta})$ must be left coprime, see e.g. Gevers et al. [29] and Kailath [48]. In other words, the degree of $\boldsymbol{A}(q,\boldsymbol{\theta})$ must at least be equal to the degree of $\boldsymbol{A}_0(q)$, and similar, the degree of $\boldsymbol{C}(q,\boldsymbol{\theta})$ must at least be equal to the degree of $\boldsymbol{C}_0(q)$. If both $na = na_0$ and $nc = nc_0$ are fulfilled, the solution will be unique, which implies that only one parameter vector $\boldsymbol{\theta}_0$ will exist.

5.5.2 Convergence and Consistency of the Estimate

It is interesting to analyse what happens when the number of samples N tends to infinity. Since $\boldsymbol{y}(t_k)$ is assumed stationary and ergodic the sampled covariance matrix of the prediction errors $\boldsymbol{Q}_N(\boldsymbol{\theta})$ converges to the corresponding expected values

$$\boldsymbol{Q}_\infty(\boldsymbol{\theta}) = \lim_{N\to\infty} \boldsymbol{Q}_N(\boldsymbol{\theta}) = E\left[\boldsymbol{\varepsilon}(t_k,\boldsymbol{\theta})\boldsymbol{\varepsilon}^T(t_k,\boldsymbol{\theta})\right] \tag{5.44}$$

implying that

$$V_\infty(\boldsymbol{\theta}) = det\left(\boldsymbol{Q}_\infty(\boldsymbol{\theta})\right) = det\left(E\left[\boldsymbol{\varepsilon}(t_k,\boldsymbol{\theta})\boldsymbol{\varepsilon}^T(t_k,\boldsymbol{\theta})\right]\right) \tag{5.45}$$

If the convergence is uniform, see Ljung [71], it follows that the parameters $\hat{\boldsymbol{\theta}}_N$ converge to a minimum point of $V_\infty(\boldsymbol{\theta})$, described by the vector $\hat{\boldsymbol{\theta}}_\infty$, with the probability 1 as N tends to infinity. If $\hat{\boldsymbol{\theta}}_\infty$ coincides with the true parameters $\boldsymbol{\theta}_0$ the

PEM estimate $\hat{\boldsymbol{\theta}}_N$ is said to be consistent, in which case $\boldsymbol{Q}_\infty(\boldsymbol{\theta}) = \boldsymbol{\Lambda}_0$. If $\hat{\boldsymbol{\theta}}_\infty$, on the other hand, does not coincide with $\boldsymbol{\theta}_0$ the estimate will be biased, i.e. non-consistent. The bias $\boldsymbol{\beta}_\theta$ is then given by

$$\boldsymbol{\beta}_\theta = \boldsymbol{\theta}_0 - \hat{\boldsymbol{\theta}}_\infty \tag{5.46}$$

The PEM estimate will be consistent if the true system *S* is contained in the model structure *M*, and if $\boldsymbol{\varepsilon}(t_k,\boldsymbol{\theta})$ is a white noise.

5.5.3 Asymptotic Distribution of the Estimated Parameters

Besides establishing that the PEM estimate converges, it might be desirable to measure its performance against some standard. A standard for estimating covariance is provided by the Cramer-Rao bound, see e.g. Söderström et al. [105]. The bound applies to any unbiased estimate $\hat{\boldsymbol{\theta}}_N$ of the true parameters $\boldsymbol{\theta}_0$ using available data $\boldsymbol{y}(t_k)$. Since noise most certainly is present, at least some of the data are stochastic variables, described by their joint probability density function $p(Y^N,\boldsymbol{\theta}_0)$, with Y^N defined as $Y^N = \{\boldsymbol{y}(t_N), \boldsymbol{y}(t_{N-1}), \ldots, \boldsymbol{y}(t_0)\}^T$. As seen $p(Y^N,\boldsymbol{\theta}_0)$ is influenced by the true parameters $\boldsymbol{\theta}_0$. The covariance $\boldsymbol{P}(\hat{\boldsymbol{\theta}}_N)$ of $\hat{\boldsymbol{\theta}}_N$ cannot be less that the Cramer-Rao lower bound $\boldsymbol{F}^{-1}$

$$\boldsymbol{P}(\hat{\boldsymbol{\theta}}_N) \geq \boldsymbol{F}^{-1} \tag{5.47}$$

where

$$\boldsymbol{F} = E\left[\left(\frac{\partial \ln p(Y^N,\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0}\right)^T\left(\frac{\partial \ln p(Y^N,\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0}\right)\right] \tag{5.48}$$

is the Fisher information matrix, see e.g. Söderström et al. [105]. Without attempting a detailed interpretation of $\boldsymbol{F}$, this matrix associates the lowest potential covariance $\boldsymbol{F}^{-1}$ of an estimate with the most information about it. If the covariance of $\hat{\boldsymbol{\theta}}_N$ in (5.47) equals $\boldsymbol{F}^{-1}$ then the estimator is said to be statistically efficient. This implies that no other unbiased estimate will have lower covariance. The PEM estimate will be statistically efficient if, see Ljung [71] and Söderström et al. [105]:

- ☞ *The true system S is contained in the model structure M*
- ☞ $\boldsymbol{\varepsilon}(t_k,\boldsymbol{\theta})$ *is a white noise.*
- ☞ *The chosen criterion function is* (5.17).

On the basis of these assumptions an estimate of the covariance associated with $\hat{\boldsymbol{\theta}}_N$ is given by, see Söderström et al. [105]

$$\hat{\boldsymbol{P}}(\hat{\boldsymbol{\theta}}_N) = \left[\sum_{k=1}^{N} \boldsymbol{\psi}(t_k,\hat{\boldsymbol{\theta}}_N)\boldsymbol{Q}_N^{-1}(\hat{\boldsymbol{\theta}}_N)\boldsymbol{\psi}^T(t_k,\hat{\boldsymbol{\theta}}_N)\right]^{-1} = \boldsymbol{R}_N^{-1}(\hat{\boldsymbol{\theta}}_N) \tag{5.49}$$

where $\boldsymbol{R}_N(\boldsymbol{\theta})$ is obtained from (5.26).

5.6 Constructing Initial Parameter Estimates

The Gauss-Newton search scheme given in definition 5.1 is started by supplying initial parameter estimates. Even though the convergence rate is fast, it is vital that the initial parameter estimates are accurate to avoid unnecessary iterations. This section investigates how such initial estimates can be obtained in a fast manner.

5.6.1 Stochastic State Space Realization Estimation

Assume that initial parameter estimates of the system matrices $\{\boldsymbol{A}(\boldsymbol{\theta}), \boldsymbol{K}(\boldsymbol{\theta}), \boldsymbol{C}(\boldsymbol{\theta})\}$ of the innovation state space system given in section 2.2.3 are desired. Such estimates can be obtained from several different algorithms, known as stochastic state space estimators. Consider the stochastic state space system

$$\begin{aligned} \boldsymbol{x}(t_{k+1}) &= \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x}(t_k) + \boldsymbol{w}(t_k) \\ \boldsymbol{y}(t_k) &= \boldsymbol{C}(\boldsymbol{\theta})\boldsymbol{x}(t_k) + \boldsymbol{v}(t_k) \end{aligned} \tag{5.50}$$

Given the dimensions of the system and measured system response $\boldsymbol{y}(t_k)$, for $k = 1$ to N, the common output from these algorithms are estimates of the system matrices, denoted $\hat{\boldsymbol{A}}(\boldsymbol{\theta})$ and $\hat{\boldsymbol{C}}(\boldsymbol{\theta})$, and the covariance matrices $\boldsymbol{\Sigma} = E[\boldsymbol{y}(t_k)\boldsymbol{y}^T(t_k)]$ and $\boldsymbol{M} = E[\boldsymbol{x}(t_{k+1})\boldsymbol{y}^T(t_k)]$. The Kalman gain matrix can then be estimated as

$$\hat{\boldsymbol{K}}(\boldsymbol{\theta}) = \left(\boldsymbol{M} - \hat{\boldsymbol{A}}(\boldsymbol{\theta})\boldsymbol{\Pi}\hat{\boldsymbol{C}}^T(\boldsymbol{\theta})\right)\left(\boldsymbol{\Sigma} - \hat{\boldsymbol{C}}(\boldsymbol{\theta})\boldsymbol{\Pi}\hat{\boldsymbol{C}}^T(\boldsymbol{\theta})\right)^{-1} \tag{5.51}$$

by the solution of the algebraic Riccati equation

$$\begin{aligned} \boldsymbol{\Pi} = {} & \hat{\boldsymbol{A}}(\boldsymbol{\theta})\boldsymbol{\Pi}\hat{\boldsymbol{A}}^T(\boldsymbol{\theta}) + \\ & \left(\boldsymbol{M} - \hat{\boldsymbol{A}}(\boldsymbol{\theta})\boldsymbol{\Pi}\hat{\boldsymbol{C}}^T(\boldsymbol{\theta})\right)\left(\boldsymbol{\Sigma} - \hat{\boldsymbol{C}}(\boldsymbol{\theta})\boldsymbol{\Pi}\hat{\boldsymbol{C}}^T(\boldsymbol{\theta})\right)^{-1}\left(\boldsymbol{M} - \hat{\boldsymbol{A}}(\boldsymbol{\theta})\boldsymbol{\Pi}\hat{\boldsymbol{C}}^T(\boldsymbol{\theta})\right)^T \end{aligned} \tag{5.52}$$

The estimated realizations are typically balanced, see section 7.1. However, if the estimated realization is observable, it can be transformed to e.g. the observability canonical form by the similarity transformation of definition 2.2. If the state dimension divided by the number of output equals an integral value n, the realization can also be converted to an ARMAV(n,n) model, see theorem 2.3.

The stochastic state space realization estimators can as such be powerful tools for initialization of the PEM algorithm both for the innovation state space representation and the ARMAV(n,n) model. These estimators can be divided into covariance and data driven algorithms. The parameter estimates of the covariance driven algorithms are obtained by a decomposition of Hankel matrices that consist of sampled lagged covariances $\boldsymbol{\Sigma}(i)$ of the measured response $\boldsymbol{y}(t_k)$. Such algorithms have been studied by e.g. Aoki [11] and Hoen [38] and are known as e.g. the Matrix Block Hankel estimator. The data driven algorithms are of more recent origin. In these algorithms the system matrices are estimated by a decomposition of Hankel matrices that consists of the measured data. This is a more powerful estimation approach, since the estimation of the covariance matrices of the measurements are eliminated. These algorithms have been studied extensively in De Moor et al. [20], Kirkegaard et al. [54] and Van Overschee et al. [107]. Common to the estimators is that they rely on the numerical reliable Singular Value Decomposition (SVD). The performance of both covariance and data-driven stochastic state space estimator for identification of civil engineering structures is studied in e.g. Andersen et al. [6], Hoen [38] and Kirkegaard et al. [54].

5.6.2 Multi-Stage Least-Squares Estimation

This section describes a least squares approach for estimation of an ARMAV(na,nc) model of arbitrary orders na and nc. In the following it is assume that the true system is described by an ARMAV model

$$\boldsymbol{A}_0(q)\boldsymbol{y}(t_k) = \boldsymbol{C}_0(q)\boldsymbol{e}(t_k) \tag{5.53}$$

Since this model can be approximated arbitrarily well by a high-order ARV model

$$\boldsymbol{A}_H(q)\boldsymbol{y}(t_k) = \hat{\boldsymbol{e}}(t_k) \tag{5.54}$$

as the number of samples N tends to infinity, an estimated ARMAV model is obtained from, see Hannan et al. [34], Ljung [71] and Mayne et al. [75]

$$\boldsymbol{A}(q,\boldsymbol{\theta})\boldsymbol{y}(t_k) = \big(\boldsymbol{C}(q,\boldsymbol{\theta}) - \boldsymbol{I}\big)\hat{\boldsymbol{e}}(t_k) + \boldsymbol{e}(t_k) \tag{5.55}$$

Since $\hat{\boldsymbol{e}}(t_k)$ can be determined before hand from (5.54) by simple linear least squares estimation of $\boldsymbol{A}_H(q)$, the model (5.55) is a multivariate ARX model. Such a model can also be solved by a linear least squares technique and estimates of $\boldsymbol{A}(q,\boldsymbol{\theta})$ and $\boldsymbol{C}(q,\boldsymbol{\theta})$ can be extracted. This approach can be refined iteratively by repeating the estimation of the ARX model with an updated external input $\hat{\boldsymbol{e}}(t_k)$. The prediction error $\boldsymbol{\varepsilon}(t_k,\boldsymbol{\theta})$ of (5.55) is obtained from

$$\boldsymbol{\varepsilon}(t_k,\boldsymbol{\theta}) = \boldsymbol{y}(t_k) - \boldsymbol{A}^{-1}(q,\boldsymbol{\theta})\big(\boldsymbol{C}(q,\boldsymbol{\theta}) - \boldsymbol{I}\big)\hat{\boldsymbol{e}}(t_k) \tag{5.56}$$

and the updated external input $\hat{\boldsymbol{e}}(t_k)$ is then given by

$$\hat{\boldsymbol{e}}(t_k) = \boldsymbol{\varepsilon}(t_k, \boldsymbol{\theta}) \tag{5.57}$$

This approach is a fast and reliable way of obtaining initial parameter estimates, since only linear least-squares estimation is involved. The estimates of the parameters of models of moderate size typically converges after 20-40 iterations. However, as stated in Piombo et al. [93], the number of iterations is strongly dependent upon the choice of sampling interval.

5.7 Model Structure Selection and Model Validation

In system identification both the determination of an appropriate model structure $M(\boldsymbol{\theta})$ and model validation are important for obtaining a successful model of the true system *S*, see section 5.5.1. In that section, it was shown that the model structure must be large enough to contain the true system. However, overparameterization of a model structure should be avoided since this leads to unnecessary complicated computations in the estimation process, and increases the uncertainty on the significant parameters of the model. Underparameterization of a model structure should also be avoided, since this means that the true system cannot be fully described by the model. This inadequacy of the model might as such lead to an inaccurate description of the system. The purpose of this section is to quote some basic methods that can be used to obtain an appropriate model structure and to validate the best model pointed out within the chosen structure. To obtain an appropriate and adequate model structure three choices have to be made in a sensible way. These choice are:

- *Choice of number of sensors to use.*
- *Choice of an adequate model structure.*
- *Choice of the dimension of the model.*

The next three sections concern these choices, whereas the fourth section concerns the model validation of the chosen model structures.

5.7.1 Choice of Number of Sensors to Use

The number of sensors to use is usually decided prior to the system identification session. The sensors might e.g. be a build-in part of the structure. In addition, it also depends on the actual application; if mode shapes are desired for some reason it is necessary to use several sensors and to place them on locations that can provide an optimal description of the mode shapes. There might be several reasons for using several sensors. Since the number of independent model parameters of e.g. the ARMAV model depends on the square of the number of sensors, it is desirable to limit the number of sensors as a consequence of the parsimony principle. This

principle states that out of two or more competing models which all explain the data well, the model with the smallest number of independent parameters should be chosen, see also Söderström et al. [105].

The effect of the parsimony principle can be illustrated for a univariate ARMA model by letting N tend to infinity and assume the that true system belongs to the selected class of models. An expression for the expected prediction error variance, where the expectation is with respect to $\hat{\boldsymbol{\theta}}_N$ is given by, see e.g. Söderström et al. [105]

$$E\left[W_N(\hat{\boldsymbol{\theta}}_N) \right] \approx \Lambda\left(1 + \frac{m}{N} \right) \tag{5.58}$$

m is the dimension of the parameter vector and $W_N(\hat{\boldsymbol{\theta}}_N)$ is a scalar assessment measure that must be a smooth function and be minimized when $\hat{\boldsymbol{\theta}}_N = \boldsymbol{\theta}_0$. In the scalar case $W_N(\hat{\boldsymbol{\theta}}_N)$ equals

$$W_N(\hat{\boldsymbol{\theta}}_N) = E\left[\varepsilon^2(t_k, \hat{\boldsymbol{\theta}}_N) \right] \tag{5.59}$$

which is the prediction error variance when the corresponding model to $\boldsymbol{\theta}_0$ is used.

It is seen that this measure of accuracy only depends on the number of data N and the number of adjustable or free parameters in the model m. The model structure $M(\boldsymbol{\theta})$ and the experimental conditions do not affect the result contrary to the asymptotic expression for the estimated covariance of the estimate $\hat{\boldsymbol{\theta}}_N$ given in (5.49) in section 5.5.3. In other words, it is in some cases more relevant to consider (5.58) instead of (5.49) for the accuracy of a system identification problem. Further, (5.58) states that the expected prediction error variance $E[W_N(\hat{\boldsymbol{\theta}}_N)]$ increases with a relative amount $\frac{m}{N}$. Thus, there is a penalty using models with an unnecessarily high number of parameters. This is in conflict with the fact that flexible models are needed to ensure small bias. So clearly a compromise has to be found. Increasing the number of parameters will reduce the bias and increase the variance. The question how to reach a sensible compromise is the key point of the system identification process. Thus, the primary issue of the parsimony principle is to find an adequate model structure that can contain the true system and is described with as few parameters as possible. To illustrate this, consider the following example, showing two ways of modelling the dynamic behaviour of a structural system.

Example 5.1 - Modelling of a Second-Order Noise-Free System

Suppose the true system is a white noise excited second order system that has a total of 10 complex conjugated pairs of eigenvalues. Assume that the measurements are obtained in a noise-free environment. Consider the following two ways of modelling this system.

If the number of observed outputs of the system is $p = 1$, then an appropriate covariance equivalent discrete-time model is an ARMA(20,19) with a total number of parameters $m = (20 + 19) \cdot 1^2 = 39$.

On the other hand, if the number of observed outputs of the system is $p = 10$, then an appropriate covariance equivalent discrete-time model is an ARMAV(2,1) with a total number of parameters $m = (2+1)\cdot 10^2 = 300$. □

This example clearly shows that if the extra information provided by the ARMAV(2,1), such as mode shape information, is unnecessary, and if all eigenvalues are identifiable from the single output of the ARMA(20,19) model, then this model should be chosen according to the principle of parsimony.

So in conclusion:

☞ *The number of sensors should be as large as necessary to describe the dynamic characteristics sought in the actual application, and at the same time as small as possible in order to satisfy the principle of parsimony.*

The optimal number of sensors to use and their optimal location on the structure can be estimated in different ways. This topic has been investigated in e.g. Kirkegaard [52] and Kirkegaard et al. [61].

5.7.2 Choice of an Adequate Model Structure

It is not only a matter of selecting the number of sensors. It is also necessary to apply an appropriate model structure that can describe the true system with as few parameters as possible. In chapters 3-5, it has been established how a civil engineering structure subjected to an unknown ambient excitation might be modelled. The assumptions on which the modelling has been based are:

☞ *The system can be assumed linear and time-invariant.*
☞ *The ambient excitation can be modelled as a stationary Gaussian white noise excited linear and time-invariant shaping filter.*

On the basis of these assumptions the discrete-time ARMAV(n,n-1) model was derived for a noise-free p-variate linear and time-invariant system subjected to a stationary Gaussian distributed excitation. So if the system and the applied excitation fulfil the above assumptions, and if the measurements are sampled perfectly without noise, then the ARMAV(n,n-1) model will be adequate for $m_S \leq np$. m_S is the state dimension of the true system and p is the number of sensors. However, most structures will not behave completely linear and time-invariant. Further, the actual excitation will probably not be completely stationary, and the measurements cannot be perfectly sampled. All this implies that disturbance will always be present. In chapter 4 it was shown how to incorporate the disturbance into the equivalent discrete-time model. This incorporation was based on the following assumption:

☞ *The disturbance of the system can be modelled as Gaussian white noise.*

On the basis of this assumption and the results obtained in chapter 4, the adequate model structure will be the ARMAV(n,n) model instead.

It might of course also be any equivalent stochastic state space realization. In fact, these are the usually assumptions made if system identification using stochastic state space realization estimators is applied.

So in conclusion:

☞ *Since disturbance most certainly always is present in some sense, an adequate way of modelling a linear and time-invariant system excited by a stationary stochastic Gaussian distributed excitation is by using an* ARMAV(*n*,*n*) *model. The order n is determined from the state dimension* m_S *of the true system and the number of sensors p as* $m_S \leq np$.

5.7.3 Choice of the Dimension of the Model

What remains is essential to determine the state dimension m_S of the true system in the light of the number of sensors, the number of dynamic modes of the true system, and the parsimony principle. As stated above, the key issue of the parsimony principle is to determine an adequate model structure that can contain the true system and is described with as few free parameters as possible. There is no general way of choosing the dimension of an adequate model structure, but a large number of methods to assist in the choice exists. These methods can be divided into several categories. They are based on

☞ *A priori knowledge.*
☞ *Preliminary data analysis.*
☞ *Comparison of estimated model structures.*

A Priori Knowledge

A priori knowledge could e.g. originate from a prior identification of the structure in question or a physical understanding of its dynamic behaviour. So, if a priori knowledge exists, it should be used to determine the state dimension m_S .

Preliminary Data Analysis

The state dimension m_S can also be estimated from a preliminary data analysis using e.g. spectral analysis of the measured system response which provides valuable information about the resonance peaks of underdamped damped modes. Thus, assuming that all modes are underdamped m_S is approximately twice the number of resonance peaks. However, there might be modes that are not necessarily under-damped. These originate e.g. from the modelling of the excitation, nonwhite disturbance, or from prefilter characteristics. This implies that the actual state dimension m_S might be larger. However, spectral analysis provides a fast and easy way of establishing a sensible initial guess of m_S. Other robust numerical techniques that utilise the Singular Value Decomposition (SVD), can also be applied to the measured data for estimation of m_S, see e.g. Aoki [11].

Comparison of Estimated Model Structures

A most natural approach to search for a suitable dimension of the model structure is simply to test a number of different ones and then compare the resulting models. However, it is usually only feasible to do with simple models because of the amount of calculation involved in more complicated models. One practical way of doing it is to start an identification session by selecting a low order ARMAV model, such as the ARMAV(2,2) model. If the dimension of this model is inadequate, then an ARMAV(3,3) is tested, and so on. This is the approach used in Pandit [84], and is termed the modelling approach for Data Dependent Systems (DDS). However, the crucial difference between the DDS modelling approach and the above-mentioned approach is that the use of the ARMAV(*n*,*n*-1) models is recommended, see De Roeck et al. [21], Pandit [84] and Pandit et al. [87].

For such a comparison of model structure dimensions a discriminating criterion is needed. The comparison of the model structures can be interpreted as a test for a significant decrease in the minimal values of the criterion function $V_N(\hat{\theta}_N)$ associated with the model structures in question. As a model structure is expanded, e.g. by increasing the state dimension and thereby the number of free parameters, the minimal value of $V_N(\hat{\theta}_N)$ decreases since new degrees of freedom have been added to the minimization problem. The decrease of $V_N(\hat{\theta}_N)$ is a consequence that more flexible model structures give a possibility for better fit to the data. On the other hand, at the moment when a good fit can be obtained there is no reason to increase e.g. the number of free parameters. In fact it follows from (5.58) that there is an inherent penalty for a too flexible model structure.

Considerations of this type has led to e.g. the Akaike Information Criterion (AIC) and the Akaike Final Prediction Error criterion (FPE), see e.g. Akaike [1]

$$AIC = N\log\left(V_N(\hat{\theta}_N)\right) + 2m$$

$$FPE = V_N(\hat{\theta}_N)\frac{1 + \frac{m}{N}}{1 - \frac{m}{N}} \tag{5.60}$$

where *log* is the natural logarithm. The model structure giving the smallest value of one of these criterions will then be the “best” compromise between the decrease of $V_N(\hat{\theta}_N)$ and the increase of model structure complexity, see also Ljung [71] and Söderström et al. [105]. In figure 5.1, the AIC and FPE criterions are plotted together with the loss function itself for eleven different ARMAV models. All models have been estimated using the nonlinear PEM approach presented in this chapter. The measurements used in all estimations consists of data from two sensors. The measured system has five significant underdamped modes.

5.7.4 Model Validation

Model validation is the final stage of the system identification procedure. In fact model validation overlaps with the model structure selection. Since system identification is, an iterative process various stages will not be separated, i.e. models are estimated and the validation results may lead to new models, etc. Model validation involves two basic questions:

- ☞ *What is the best model within the chosen model structure?*
- ☞ *Is the model fit for its purpose?*

The dilemma in model validation is that there are many different ways to determine and compare the quality of the estimated models. The subjective judgement in model validation should be stressed. It is the user that makes the decision based on numerical indicators. The estimated variance of the estimated parameters, i.e. the diagonal elements of $\hat{\boldsymbol{P}}(\hat{\boldsymbol{\theta}}_N)$, is such an indicator. High values indicate a model with a bad fit or overparameterization.

It is also important to verify whether the model is a good fit for the data to which is was estimated. If it is a bad fit, it may e.g. indicate that the parameter estimates represent a local minimum point. In this context it should also be noted that simulation of the response of the model with the prediction error sequence as its input, and comparing the measured output $\boldsymbol{y}(t_k)$ with the simulated $\hat{\boldsymbol{y}}(t_k|t_{k-1};\boldsymbol{\theta})$, can also be used for model validation.

Statistical tests on the prediction errors $\boldsymbol{\varepsilon}(t_k,\boldsymbol{\theta})$ are also typically used numerical indicators for model validation. If the statistical distribution of $\boldsymbol{\varepsilon}(t_k,\boldsymbol{\theta})$ matches the assumed distribution then it can be concluded that the system dynamics is indeed well represented by the model. Any different trend in the statistical characteristics, from which it was originally assumed, is an indication that either the model or the noise is incorrectly assumed or that the parameters are incorrectly estimated. These tests are also referred to as residual tests, and the most commonly used procedure is plotting of the estimated cross- and auto-correlation functions of the prediction errors, see e.g. Ljung [71] and Söderström et al. [105]. The above-mentioned tools for model validation lead to a conclusion as to whether a model is fit for its purpose or not.

5.8 Summary

This chapter has considered the estimation of the multivariate ARMAV model, or equivalently the innovation state space representation, using a nonlinear prediction error method (PEM). It has been discussed how to obtain an optimal predictor for a given model structure, and how to assess the performance of this predictor through a scalar valued criterion function. This criterion function has been selected so it, under the assumption that the prediction errors are independent Gaussian distributed random variables, provides optimal accuracy of the estimated parameters.

The minimization of the criterion function, with respect to the adjustable parameters of the model structure, has then been investigated. A specific search scheme has been considered. This is known as the Gauss-Newton search scheme for iterative improvement of an estimate. The statistical properties of the PEM estimator are then considered. It is shown that if the true system is contained in the chosen model structure, and if the prediction error sequence is a white noise, then the estimator will be consistent and asymptotically efficient. Based on these assumptions it is shown how to obtain an estimate of the covariance of the estimated parameters. Finally, there is the problem of selecting the right model structure and how to validate that this model will actually fulfil its purpose. Practical guidelines for the selection of an appropriate and adequate model structure have been given.