Lecture 2

Non-recursive (Off-line) Identification Methods

- Linear Regression and the Least Squares (LS) Methods
- Prediction Error Methods (PEM)
- Instrumental Variable Methods (IVM)

The Least Squares (LS) Methods

- Linear regression
- The least squares method
- Properties of the (deterministic) least squares method
- BLUE
- Computational aspects

Linear Regression

System Identification (SI) procedure: Collect data, *choose a model class*, *find the best model in the model class*, validation.

- Linear regression models. Models that are linearly parametrized.
  - The simplest type of *parametric* model.
  - Computationally simple.
  - Simple to implement.
  - Low memory consumption.
  - Common in signal processing, e.g. echo cancellation.
- Original work by Gauss 1809 for calculating orbits of the planets.
- Starting point of system identification.

Linear Regression (Cont'd)

Model structure ($\mathcal{M}$):

$$y_m(t) = \boldsymbol{\varphi}^T(t)\boldsymbol{\theta}, \quad t = 1, \ldots, N \tag{1}$$

where $y_m(t)$ is the model output, $\boldsymbol{\varphi}(t) \in \mathbb{R}^{n\times 1}$ is a vector of known quantities and $\boldsymbol{\theta} \in \mathbb{R}^{n\times 1}$ is a vector of unknown quantities. The elements of vector $\boldsymbol{\varphi}(t)$ is called as *regression variables* or *regressors*, and vector $\boldsymbol{\theta}$ is known as *parameter vector.*

The model (1) can be compactly written as

$$\boldsymbol{Y}_m = \boldsymbol{\Phi\theta}, \quad \boldsymbol{Y}_m = \begin{bmatrix} y_m(1) \\ \vdots \\ y_m(N) \end{bmatrix}, \boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\varphi}^T(1) \\ \vdots \\ \boldsymbol{\varphi}^T(N) \end{bmatrix} \tag{2}$$

Linear Regression and Least Squares (Optimization)

Problem: Find an estimate of $\boldsymbol{\theta}$ for given measurement $y(1), \boldsymbol{\varphi}(1), \ldots, y(N), \boldsymbol{\varphi}(N)$.

Solution: Introduce the *equation error*

$$\varepsilon(t) = y(t) - y_m(t) = y(t) - \boldsymbol{\varphi}^T(t)\boldsymbol{\theta}, \quad t = 1, \ldots, N$$

or compactly

$$\boldsymbol{\varepsilon} = \boldsymbol{Y} - \boldsymbol{Y}_m = \boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{\theta}$$

Least squares method: Choose $\boldsymbol{\theta}$ such that $\varepsilon^2(t)$ is small for all t:

$$\hat{\boldsymbol{\theta}}_{LS} = \arg\min_{\boldsymbol{\theta}} V(\boldsymbol{\theta})$$

$$V(\boldsymbol{\theta}) = \frac{1}{2}\sum_{t=1}^{N} \varepsilon^2(t) = \frac{1}{2}\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} = \frac{1}{2}(\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{\theta})^T(\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{\theta})$$

Results: Assume that $\boldsymbol{\Phi}^T\boldsymbol{\Phi}$ is invertible. Then the solution of the above optimization is given by solving $\frac{\partial}{\partial \boldsymbol{\theta}} V(\boldsymbol{\theta}) = 0$, which leads to

$$\hat{\boldsymbol{\theta}}_{LS} = \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{Y} = \left(\sum_{t=1}^{N} \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t)\right)^{-1}\sum_{t=1}^{N}\boldsymbol{\varphi}(t)y(t)$$

Note: The above LS algorithm is also referred as to Block/Batch LS.

Weighted least squares (WLS) estimate:

$$\hat{\boldsymbol{\theta}}_{WLS} = \arg\min_{\boldsymbol{\theta}} V(\boldsymbol{\theta}), \quad V(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\varepsilon}^T\boldsymbol{W}\boldsymbol{\varepsilon}$$

$$\Rightarrow \quad \hat{\boldsymbol{\theta}}_{WLS} = \left(\boldsymbol{\Phi}^T\boldsymbol{W}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{W}\boldsymbol{Y}$$

where $\boldsymbol{W}$ is symmetric ($\boldsymbol{W}^T = \boldsymbol{W}$) and positive definite.

Remark: $\boldsymbol{W} = \boldsymbol{I} \Rightarrow \hat{\boldsymbol{\theta}}_{WLS} = \hat{\boldsymbol{\theta}}_{LS}$.

Question: Can you derive/prove the above WLS?

Least Squares (Statistical Properties)

To explore the properties of the least squares estimate we need to specify the system, *i.e.*, we need to make some assumptions about generating data.

Assumptions:

- $\boldsymbol{\varphi}(t)$ is deterministic and known. (Quite restrictive assumption!)
- System: $y(t) = \boldsymbol{\varphi}^T(t)\boldsymbol{\theta}_0 + e(t)$, where $e(t)$ is a sequence of random variables, $\mathrm{E}\, e(t) = 0$ and $\mathrm{E}\, e(t)e(s) = R_{ts}$. Compactly written as

$$\boldsymbol{Y} = \boldsymbol{\Phi}\boldsymbol{\theta}_0 + \boldsymbol{e}, \qquad \mathrm{E}\, \boldsymbol{e} = \boldsymbol{0}, \quad \mathrm{E}\, \boldsymbol{e}\boldsymbol{e}^T = \boldsymbol{R}$$

Remark: If $\boldsymbol{R} = \lambda^2 \boldsymbol{I}$ then $e(t)$ is white noise with variance λ^2.

Least Squares (Statistical Properties) - Results

- The (weighted) least squares estimate is *unbiased*:
  - $\mathrm{E}\,\hat{\boldsymbol{\theta}}_{WLS} = \boldsymbol{\theta}_0$
- Covariance matrix, $\operatorname{cov}\hat{\boldsymbol{\theta}} = \mathrm{E}\,(\hat{\boldsymbol{\theta}} - \mathrm{E}\,\hat{\boldsymbol{\theta}})(\hat{\boldsymbol{\theta}} - \mathrm{E}\,\hat{\boldsymbol{\theta}})^T$:
  - $\operatorname{cov}\hat{\boldsymbol{\theta}}_{WLS} = [\boldsymbol{\Phi}^T \boldsymbol{W} \boldsymbol{\Phi}]^{-1} \boldsymbol{\Phi}^T \boldsymbol{W} \boldsymbol{R} \boldsymbol{W} \boldsymbol{\Phi} [\boldsymbol{\Phi}^T \boldsymbol{W} \boldsymbol{\Phi}]^{-1}$
  - $\operatorname{cov}\hat{\boldsymbol{\theta}}_{LS} = [\boldsymbol{\Phi}^T \boldsymbol{\Phi}]^{-1} \boldsymbol{\Phi}^T \boldsymbol{R} \boldsymbol{\Phi} [\boldsymbol{\Phi}^T \boldsymbol{\Phi}]^{-1}$
  - $\boldsymbol{R} = \lambda^2 \boldsymbol{I} \Rightarrow$
    $\operatorname{cov}\hat{\boldsymbol{\theta}}_{LS} = \frac{\lambda^2}{N}[\frac{1}{N}\boldsymbol{\Phi}^T \boldsymbol{\Phi}]^{-1} = \frac{\lambda^2}{N}\left[\frac{1}{N}\sum_{t=1}^{N} \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t)\right]^{-1}$
- If $e(t)$ is Gaussian distributed $e(t) \sim N(0, \boldsymbol{R})$, then $\hat{\boldsymbol{\theta}}_{WLS} \sim N(\boldsymbol{\theta}_0, \operatorname{cov}\hat{\boldsymbol{\theta}}_{WLS})$. (Holds for finite N)
- $\hat{\boldsymbol{\theta}}_{WLS}$ is *consistent*: $\hat{\boldsymbol{\theta}}_{WLS} \to \boldsymbol{\theta}_0$, $N \to \infty$.

Definition: The estimate $\hat{\boldsymbol{\theta}}_1$ is statistically more efficient than $\hat{\boldsymbol{\theta}}_2$ if

$$\operatorname{cov}\hat{\boldsymbol{\theta}}_1 \leq \operatorname{cov}\hat{\boldsymbol{\theta}}_2$$

Question: Which choice of $\boldsymbol{W}$ will minimize $\operatorname{cov}\hat{\boldsymbol{\theta}}_{WLS}$?

Result: The choice $\boldsymbol{W} = \boldsymbol{R}^{-1}$ yields optimal accuracy:

- $\hat{\boldsymbol{\theta}}_{WLS} = \left(\boldsymbol{\Phi}^T \boldsymbol{R}^{-1} \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^T \boldsymbol{R}^{-1} \boldsymbol{Y}$
- $\operatorname{cov}\hat{\boldsymbol{\theta}}_{WLS} = [\boldsymbol{\Phi}^T \boldsymbol{R}^{-1} \boldsymbol{\Phi}]^{-1}$

In this case $\hat{\boldsymbol{\theta}}_{WLS}$ is known as the BLUE (best linear unbiased estimator) or the Gauss-Markov estimate.

BLUE

- BLUE = Best Linear Unbiased Estimator.
- White noise, $\boldsymbol{R} = \lambda^2 \boldsymbol{I}$. BLUE yields the same estimate as the deterministic least squares method.
- If $e(t)$ is Gaussian, then BLUE yields the best possible estimate! If $e(t)$ is non-Gaussian, then there might exist better non-linear estimates.
- BLUE can be derived also for singular $\boldsymbol{R}$.

Computational Aspects

The least squares solution ($\boldsymbol{\Phi} \in \mathbb{R}^{N\times n}$)

- $\hat{\boldsymbol{\theta}}_{LS} = \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{Y} = \left(\sum_{t=1}^N \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t)\right)^{-1}\sum_{t=1}^N \boldsymbol{\varphi}(t)y(t)$

is unsuitable for numerical implementation.

Alternatives: Avoid the inverse!

- The normal equations: $(\boldsymbol{\Phi}^T\boldsymbol{\Phi})\hat{\boldsymbol{\theta}}_{LS} = \boldsymbol{\Phi}^T\boldsymbol{Y}$.
- Solve an overdetermined linear system of equations: $\boldsymbol{Y} = \boldsymbol{\Phi}\hat{\boldsymbol{\theta}}_{LS}$. (Recall that $\boldsymbol{Y} - \boldsymbol{Y}_m = \boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{\theta}$ should be small.)
  - QR factorizations
  - SVD factorizations

QR factorization: Let $\boldsymbol{\Phi} = \boldsymbol{QR}$, where $\boldsymbol{Q} \in \mathbb{R}^{N \times N}$ is orthogonal ($\boldsymbol{Q}^T \boldsymbol{Q} = \boldsymbol{I}$) and $\boldsymbol{R} \in \mathbb{R}^{N \times n}$ is upper triangular. Then, instead of solving

$$\boldsymbol{Y} = \boldsymbol{\Phi}\boldsymbol{\theta}$$

we can equally well solve

$$\boldsymbol{Q}^T \boldsymbol{Y} = \boldsymbol{Q}^T \boldsymbol{\Phi}\boldsymbol{\theta} = \boldsymbol{R}\boldsymbol{\theta}$$

which is easy due to the structure of $\boldsymbol{R}$.

- Requires more computations than solving the normal equations.
- Less sensitive to rounding errors.

Summary on Least Squares Methods

- Regression models describes a large class of dynamic systems (linear w.r.t the parameters).
- Linear regression can be used also for certain non-linear models.
- The least squares method is fundamental in system identification, and can be derived from various starting points.
- We have assumed that $\mathbf{\Phi}$ is a known and deterministic matrix. Problems when this matrix is a function of $u(t)$ and $y(t)$ (e.g. ARX-model – $A(q^{-1})y(t) = B(q^{-1})u(t) + \varepsilon(t)$).

Issues with the Least-Squares Method

- Up to now, the least squares method has been applied to static (deterministic) linear regression models ($\boldsymbol{\varphi}(t)$ deterministic).
- What happens when we consider dynamic models?

$$\begin{aligned} A(q^{-1}, \boldsymbol{\theta}) y(t) &= B(q^{-1}, \boldsymbol{\theta}) u(t) + e(t) \\ \Rightarrow \qquad y(t) &= \boldsymbol{\varphi}^T(t)\boldsymbol{\theta} + e(t) \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{\varphi}(t) &= [-y(t-1) \ \dots \ -y(t-n_a) \ u(t-1) \ \dots \ u(t-n_b)]^T \\ \boldsymbol{\theta} &= [a_1 \ \dots \ a_{n_a} \ b_1 \ \dots \ b_{n_b}]^T \end{aligned}$$

Properties of the least squares estimate

$$\hat{\boldsymbol{\theta}}_{LS} = \left(\frac{1}{N} \sum_{t=1}^{N} \boldsymbol{\varphi}(t) \boldsymbol{\varphi}^T(t) \right)^{-1} \frac{1}{N} \sum_{t=1}^{N} \boldsymbol{\varphi}(t) y(t)$$

Properties: Assume that the true system can be described as

$$y(t) = \boldsymbol{\varphi}^T(t)\boldsymbol{\theta}_0 + e(t)$$

Results: The estimate $\hat{\boldsymbol{\theta}}_{LS}$ will be consistent ($\hat{\boldsymbol{\theta}} \rightarrow \boldsymbol{\theta}_0$ as $N \rightarrow \infty$) if

(i) $E\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t)$ is nonsingular.

(ii) $E\boldsymbol{\varphi}(t)e(t) = \mathbf{0}$.

The first condition will be satisfied in most cases. A few exceptions:

- The input is not persistently exiting of order n_b.
- The data is noise-free $e(t) \equiv 0$ and the model order is chosen too high (which implies that $A(q^{-1})$ and $B(q^{-1})$ have common factors).

The second condition is in most cases *not* satisfied. A notable exception is when $e(t)$ is white noise.

Modifications of the Least-Squares Method

To relax the second constraint, we will in the following examine two different ways to modify the least-squares method:

(i) Prediction error methods (PEM). Model the noise as well!

(ii) The instrumental variables methods (IVM) – modifying the normal equations associated with the least-squares estimate.

Lecture 2

Prediction Error Methods (PEM)

Prediction Error Methods (PEM)

Main Idea:

- Model the noise as well ⇒ stochastic models, *i.e.*, the outputs from the models are not deterministic.

- Minimize the prediction errors $\varepsilon(t, \boldsymbol{\theta}) = y(t) - \hat{y}(t|t-1, \boldsymbol{\theta})$. The least-squares method is a special case of this approach; consider the prediction error

$$\varepsilon(t, \boldsymbol{\theta}) = y(t) - \hat{y}(t|t-1, \boldsymbol{\theta}) = y(t) - \boldsymbol{\varphi}^T(t)\boldsymbol{\theta}$$

A general methodology applicable to a wide range of model structures.

Examples

Find the optimal predictor, $\hat{y}(t|t-1)$ for the following systems assuming $Ee(t) = 0$, $Ee(t)e(s) = \delta_{s,t}\lambda^2$.

Notice that $\hat{y}(t|t-1)$ is a function of $\{y(s), u(s)\}_{s=-\infty}^{t-1}$.

a) $y(t) = e(t)$

b) $(1 - 0.1q^{-1})y(t) = -0.5q^{-1}u(t) + e(t)$

c) $(1 - 0.1q^{-1})y(t) = -0.5q^{-1}u(t) + (1 - 0.8q^{-1})e(t)$

Predictions

A predictor can be described as a filter that predicts the output of a dynamic system given old measured outputs and inputs. Design the predictor by

(i) *Choosing the model structure* of $y(t)$, *e.g.*, ARX ($A(q^{-1})y(t) = B(q^{-1})u(t) + \varepsilon(t)$), OE ($y(t) = \frac{B(q^{-1})}{F(q^{-1})}u(t) + \varepsilon(t)$), or ARMAX ($A(q^{-1})y(t) = B(q^{-1})u(t) + C(q^{-1})\varepsilon(t)$).

(ii) *Choosing the predictor*, $\hat{y}(t|t-1,\boldsymbol{\theta})$. A general predictor can be viewed as

$$\hat{y}(t|t-1,\boldsymbol{\theta}) = L_1(q^{-1},\boldsymbol{\theta})y(t) + L_2(q^{-1},\boldsymbol{\theta})u(t)$$

where $L_1(q^{-1},\boldsymbol{\theta})$ and $L_2(q^{-1},\boldsymbol{\theta})$ are constrained such that $\hat{y}(t|t-1,\boldsymbol{\theta})$ depends on past data.

(iii) *Choosing the cost function*, $V_N(\boldsymbol{\theta})$, to be minimized.

Optimal Prediction

Optimal Prediction means that minimization of the variance of the prediction error is used for derivation.

We will here consider the general model structure

$$y(t) = G(q^{-1}, \boldsymbol{\theta})u(t) + H(q^{-1}, \boldsymbol{\theta})e(t)$$

where $E[e(t)e^T(s)] = \boldsymbol{\Lambda}(\boldsymbol{\theta})\delta_{t,s}$ and $G(0, \boldsymbol{\theta}) = 0$.

Goal: Find the optimal mean square predictor $\hat{y}(t|t-1, \boldsymbol{\theta})$, *i.e.*, solve

$$\min_{\hat{y}(t|t-1)} E\varepsilon(t)\varepsilon^T(t)$$

where $\varepsilon(t) = y(t) - \hat{y}(t|t-1)$ is the prediction error, and $\hat{y}(t|t-1)$ depends on $\{y(s), u(s)\}_{s=-\infty}^{t-1}$.

Results:

Under the assumptions that

(i) $y(t)$ only depends on past measurements

(ii) $u(t)$ and $e(s)$ are uncorrelated for $t < s$

then

$$\hat{y}(t|t-1,\boldsymbol{\theta}) = H^{-1}(q^{-1},\boldsymbol{\theta})G(q^{-1},\boldsymbol{\theta})u(t) + \left[I - H^{-1}(q^{-1},\boldsymbol{\theta})\right] y(t)$$

is the optimal mean square predictor, and $e(t)$ the prediction error,

$$\begin{aligned} \varepsilon(t,\boldsymbol{\theta}) &= y(t) - \hat{y}(t|t-1,\boldsymbol{\theta}) \\ &= H^{-1}(q^{-1},\boldsymbol{\theta})\left[y(t) - G(q^{-1},\boldsymbol{\theta})u(t)\right] \\ &= e(t) \end{aligned}$$

Hence,

$$E\varepsilon(t,\boldsymbol{\theta})\varepsilon^T(t,\boldsymbol{\theta}) = \boldsymbol{\Lambda}(\boldsymbol{\theta})$$

Optimal Prediction for State Space Models

As an alternative to the model structure:

$$y(t) = G(q^{-1}, \boldsymbol{\theta})u(t) + H(q^{-1}, \boldsymbol{\theta})e(t),$$

it is often common to use state-space models:

$$\begin{aligned} x(t+1) &= F(\boldsymbol{\theta})x(t) + B(\boldsymbol{\theta})u(t) + v(t) \\ y(t) &= C(\boldsymbol{\theta})x(t) + e(t) \end{aligned}$$

where $v(t)$ and $e(t)$ are uncorrelated white noise sequences with zero mean and covariance matrices $R_1(\boldsymbol{\theta})$ and $R_2(\boldsymbol{\theta})$.

In this case the optimal mean square predictor is given by the **Kalman filter**.

Cost Function

How do we find the best model in the model structure?

- Minimize the prediction errors $\varepsilon(t, \boldsymbol{\theta})$ for all t. How?
- Choose a criterion function $V_N(\boldsymbol{\theta})$ to minimize:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} V_N(\boldsymbol{\theta})$$

where $V_N(\boldsymbol{\theta})$ depends on $\varepsilon(t, \boldsymbol{\theta})$ in a suitable manner.

Depending on the choice of model structure, predictor filters and criterion function, the minimization of the loss function is more or less difficult.

In general, the cost function is chosen as

$$V_N(\boldsymbol{\theta}) = h(\mathbf{R}_N(\boldsymbol{\theta}))$$

where $h(\cdot)$ is a scalar-valued monotonically increasing function, and $\mathbf{R}_N(\boldsymbol{\theta})$ is the sample covariance matrix of the prediction errors,

$$\mathbf{R}_N(\boldsymbol{\theta}) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon(t, \boldsymbol{\theta})\varepsilon^T(t, \boldsymbol{\theta}).$$

Example: $h(\cdot) =$tr$(\cdot)$, or $h(\cdot) = \det(\cdot)$.

For single-output systems the following criterion function is most often used

$$V_N(\boldsymbol{\theta}) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon^2(t, \boldsymbol{\theta})$$

A PEM Algorithm

To define a PEM the user has to make the following choices:

- *Choice of model structure.* How should $G(q^{-1}, \boldsymbol{\theta})$, $H(q^{-1}, \boldsymbol{\theta})$ and $\Lambda(\boldsymbol{\theta})$ be parameterized?
- *Choice of predictor* $\hat{y}(t|t-1, \boldsymbol{\theta})$. Usually the optimal mean square predictor is used.
- *Choice of criterion function* $V(\boldsymbol{\theta})$. A scalar-valued function of all the prediction errors $\varepsilon(1, \boldsymbol{\theta}), \ldots, \varepsilon(N, \boldsymbol{\theta})$, which will assess the performance of the predictor used.

Computational Aspects

I. Analytical solution exists

If the predictor is a linear function of the unknown parameters,

$$\hat{y}(t|t-1,\boldsymbol{\theta}) = \boldsymbol{\varphi}^T(t)\boldsymbol{\theta},$$

and the criterion function $V_N(\boldsymbol{\theta})$ is simple enough, a closed form solution can be found. For example, when

$$V_N(\boldsymbol{\theta}) = \frac{1}{N}\sum_{t=1}^{N}\varepsilon^2(t,\boldsymbol{\theta}) = \frac{1}{N}\sum_{t=1}^{N}\left(y(t)-\boldsymbol{\varphi}^T(t)\boldsymbol{\theta}\right)^2,$$

it is clear that the PEM is equivalent to linear regression (the least squares method). This holds for FIR $(y(t) = B(q^{-1})u(t) + \varepsilon(t))$ or ARX models but **not** for ARMAX and OE models.

II. No analytical solution exist

For general criterion functions, and predictors that depend non-linearly on the data, a numerical search algorithm is required to find the $\boldsymbol{\theta}$ that minimizes $V_N(\boldsymbol{\theta})$.

Numerical minimization:

- Nonlinear $\Rightarrow$ local minima.
- Time consuming (convergence rate) and computationally complex.
- Initialization.

Different (standard) methods available:

- The **Newton-Raphson** algorithm

$$\hat{\boldsymbol{\theta}}^{(k+1)} = \hat{\boldsymbol{\theta}}^{(k)} - \alpha_k [V''(\hat{\boldsymbol{\theta}}^{(k)})]^{-1} V'(\hat{\boldsymbol{\theta}}^{(k)})^T$$

  The derivatives of the loss function can be computationally complex to evaluate. Fast convergence.

- The **Gauss-Newton** algorithm is a computationally less intensive algorithm with a theoretically lower rate of convergence which can be used as an alternative.

$$\hat{\boldsymbol{\theta}}^{(k+1)} = \hat{\boldsymbol{\theta}}^{(k)} + \alpha_k \left[\sum_{t=1}^{N} \psi(t, \hat{\boldsymbol{\theta}}^{(k)}) H \psi^T(t, \hat{\boldsymbol{\theta}}^{(k)}) \right]^{-1} \left[\sum_{t=1}^{N} \psi(t, \hat{\boldsymbol{\theta}}^{(k)}) H \varepsilon(t, \hat{\boldsymbol{\theta}}^{(k)}) \right]$$

- **Gradient based** methods are simpler to apply, but has a slow convergence rate.

Summary on PEM

- The PEM is a general method to obtain a parametric model of a dynamic system. The following choices define a prediction error method:
  - choice of model structure;
  - choice of predictor;
  - choice of criterion function.
- The PEM principle is to minimize the prediction errors given a certain model structure and predictor.
- The PEM principle leads to parameter estimates that have several nice properties (in general, consistent and statistically efficient estimates).

- Approximation. The PEM is useful also for under-parameterized models. The model-fit can be controlled by pre-filtering the data, or by choosing an appropriate input.
- If the prediction errors depend linearly on the parameter vector the PEM estimates are obtained through linear regression (e.g., ARX and FIR models).
- In the case of more complicated model structures a nonlinear search algorithm is required to obtain the PEM estimates (e.g., ARMAX, OE, etc.).