POLITECNICO DI MILANO

Time-domain output-error identification

Marco Lovera

Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano

The output-error method

Model class:

$$\mathcal{M}(\theta): \quad \begin{aligned} \dot{x} &= f(x, u; \theta) \\ y &= g(x, u; \theta) \end{aligned}$$

Assumptions:

- y is a scalar measurement
- u(t) piece-wise constant with period T_s
- $\mathcal{S} \in \mathcal{M}(\theta)$ or equivalently $\exists \theta^\star$: $\mathcal{S} = \mathcal{M}(\theta^\star)$
- $\theta \in \mathbb{R}^{n_\theta}$.

The output-error method

Measurement model:

- Measurements are discrete
- Sampling is uniform and defined by

$$t_k = t_0 + kT_s, \quad k = 1, \ldots, K.$$

- Measurement equation:

$$y_m(k) = y(k) + v(k)$$

  - $y(k) = y(kT_s)$
  - $v(k) = G(0, \sigma^2), \quad E[v(i)v(j)] = 0, \quad i \neq j$

The output-error method

- Under the previous assumptions the samples of the measured output are such that

$$E[y_m(k)] = E[y(k) + v(k)] = E[y(k)] + E[v(k)] = y(k)$$

as *y*(*k*) is a deterministic sequence.

- In terms of variance we have

$$E[(y_m(k) - E[y_m(k)])^2] = E[(y_m(k) - y(k))^2] = E[v^2(k)] = \sigma^2$$

- Therefore

$$y_m(k) = G(y(k), \sigma^2)$$

The output-error method

- We have to check independence of the measurements, which, under Gaussianity assumptions, reduces to checking incorrelation:

$$E[y_m(i)y_m(j)] = E[(y(i)+v(i))(y(j)+v(j))]$$

- Expanding the product:

$$E[y_m(i)y_m(j)] = E[y(i)y(j)] + E[y(i)v(j)] + E[y(j)v(i)] + E[v(i)v(j)]$$

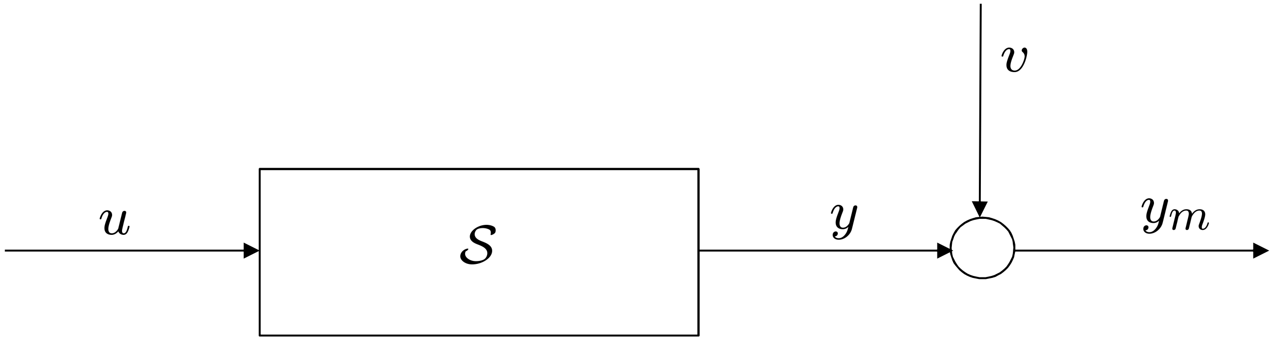
- And recalling that samples of *y* are deterministic:

$$E[y_m(i)y_m(j)] = y(i)y(j) + y(i)E[v(j)] + y(j)E[v(i)] + E[v(i)v(j)]$$

- Therefore $E[y_m(i)y_m(j)] = y(i)y(j) = E[y_m(i)]E[y_m(j)]$.

The output-error method

- Block diagram:



- Hence the name output-error for this set-up:
  - Only measurement noise is considered
  - No disturbances acting on the plant are included in the model.
  - In other words, the map from *u* to *y* is deterministic.

The output-error method

The joint density of the data can then be written as

$$f(q_1, q_2, \ldots, q_K|\theta) = f_1(q_1|\theta) f_2(q_2|\theta) \ldots f_K(q_K|\theta)$$

$$f(q_1, q_2, \ldots, q_K|\theta) = \frac{1}{(\sigma\sqrt{2\pi})^K} e^{-\frac{(q_1 - y(1;\theta))^2}{2\sigma^2}} \ldots e^{-\frac{(q_K - y(K;\theta))^2}{2\sigma^2}}$$

$$f(q_1, q_2, \ldots, q_K|\theta) = \frac{1}{(\sigma\sqrt{2\pi})^K} e^{-\frac{\sum_{k=1}^K (q_k - y(k;\theta))^2}{2\sigma^2}}$$

so the logarithm of the joint density is

$$\log f(q_1, q_2, \ldots, q_K|\theta) = -\log(\sigma\sqrt{2\pi})^K - \frac{\sum_{k=1}^K (q_k - y(k;\theta))^2}{2\sigma^2}$$

The output-error method

- Therefore the log-likelihood can be obtained by plugging the measurements in place of the running variables, to get

$$\begin{aligned}\log L(y_m(1),\ldots,y_m(K)|\theta) &= -\log(\sigma\sqrt{2\pi})^K - \frac{\sum_{k=1}^{K}(y_m(k)-y(k;\theta))^2}{2\sigma^2}\\ &= -\log(\sigma\sqrt{2\pi})^K - J(\theta).\end{aligned}$$

- Note now that maximizing the log-likelihood is equivalent to the minimisation of

$$J(\theta) = \frac{1}{2\sigma^2}\sum_{k=1}^{K}(y_m(k)-y(k;\theta))^2$$

The output-error method

- Note that defining

$$e(k;\theta) = y_m(k) - y(k;\theta)$$

we have that

$$J(\theta) = \frac{1}{2\sigma^2}\sum_{k=1}^{K}(y_m(k) - y(k;\theta))^2 = \frac{1}{2\sigma^2}\sum_{k=1}^{K} e(k;\theta)^2.$$

- Therefore the cost function is equal to the sum of the squares of the deviations between the measured outputs and the model outputs.
- This is a particular case of ML estimation known as Least Squares (LS) estimation.

The output-error method

- Note further that under the assumption $\mathcal{S} = \mathcal{M}(\theta^\star)$ we have that

$$\begin{aligned} e(k,\theta) &= y_m(k) - y(k,\theta) = \\ &= y(k;\theta^\star) + v(k) - y(k;\theta) = \\ &= v(k) + (y(k;\theta^\star) - y(k;\theta)). \end{aligned}$$

- Therefore if $\theta \to \theta^\star$ then $e(k,\theta) \to v(k)$ and the cost converges to

$$J(\theta^\star) = \frac{1}{2\sigma^2}\sum_{k=1}^{K} v^2(k).$$

- The optimal cost is zero only in the noise-free case.

An optimization scheme for the OE method

Consider a starting value θ_0 for the parameter and a perturbation $\Delta\theta$

$$J(\theta_0 + \Delta\theta) = J(\theta_0) + \frac{\partial J}{\partial \theta}|_{\theta=\theta_0}\Delta\theta + \frac{1}{2}\Delta\theta^T \frac{\partial^2 J}{\partial\theta\partial\theta^T}|_{\theta=\theta_0}\Delta\theta + \dots$$

Taking a second order approximation imposing stationarity

$$\frac{\partial J(\theta_0 + \Delta\theta)}{\partial\Delta\theta} = 0$$

we get

$$\frac{\partial J(\theta_0 + \Delta\theta)}{\partial\Delta\theta} = \frac{\partial J}{\partial \theta}|_{\theta=\theta_0} + \frac{\partial^2 J}{\partial\theta\partial\theta^T}|_{\theta=\theta_0}\Delta\theta = 0$$

An optimization scheme for the OE method

Solving for the increment in the parameter we get

$$\Delta\theta = -\left[\frac{\partial^2 J}{\partial\theta\partial\theta^T}|_{\theta=\theta_0}\right]^{-1}\left[\frac{\partial J(\theta_0+\Delta\theta)}{\partial\theta}|_{\theta=\theta_0}\right].$$

Therefore if the cost is truly quadratic then starting from the initial guess we find the minimum in one iteration.

An optimization scheme for the OE method

If the cost is not quadratic we can use this result to set up an iterative optimisation scheme:

$$
\begin{aligned}
\theta^0 &= \theta_0 \\
\theta^{k+1} &= \theta^k - \left[\frac{\partial^2 J}{\partial\theta\partial\theta^T}|_{\theta=\theta^k}\right]^{-1}\left[\frac{\partial J(\theta_0+\Delta\theta)}{\partial\Delta\theta}|_{\theta=\theta^k}\right].
\end{aligned}
$$

Iteration is repeated until convergence of the cost function

$$J(\theta^{k+1}) \simeq J(\theta^k)$$

and/or convergence of the parameter

$$\theta^{k+1} \simeq \theta^k$$

is reached.

An optimization scheme for the OE method

This simple iterative scheme is known as the Newton-Raphson method.

Other possible convergence criteria include:

- Relative rather than absolute changes in cost and/or parameters.
- Gradient of the cost sufficiently close to zero.

An optimization scheme for the OE method

How do we compute the gradient and the hessian of the cost?

Recall that

$$J(\theta) = \frac{1}{2\sigma^2}\sum_{k=1}^{K}(y_m(k) - y(k;\theta))^2 = \frac{1}{2\sigma^2}\sum_{k=1}^{K} e(k;\theta)^2.$$

and therefore

$$\frac{\partial J(\theta)}{\partial \theta} = \frac{1}{\sigma^2}\sum_{k=1}^{K}\frac{\partial e(k;\theta)}{\partial \theta}e(k;\theta) = -\frac{1}{\sigma^2}\sum_{k=1}^{K}\frac{\partial y(k;\theta)}{\partial \theta}e(k;\theta).$$

An optimization scheme for the OE method

This is a vector with components given by

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\frac{1}{\sigma^2}\sum_{k=1}^{K}\frac{\partial y(k;\theta)}{\partial \theta_j}e(k;\theta).$$

As for the second derivative, element-wise we get

$$\frac{\partial^2 J(\theta)}{\partial \theta_j \partial \theta_i} = \frac{1}{\sigma^2}\sum_{k=1}^{K}\frac{\partial y(k;\theta)}{\partial \theta_j}\frac{\partial y(k;\theta)}{\partial \theta_i} - \frac{1}{\sigma^2}\sum_{k=1}^{K}\frac{\partial^2 y(k;\theta)}{\partial \theta_j \partial \theta_i}e(k;\theta).$$

An optimization scheme for the OE method

For the sake of simplicity in the expression of the second derivative

$$\frac{\partial^2 J(\theta)}{\partial\theta_j \partial\theta_i} = \frac{1}{\sigma^2}\sum_{k=1}^{K} \frac{\partial y(k;\theta)}{\partial\theta_j}\frac{\partial y(k;\theta)}{\partial\theta_i} - \frac{1}{\sigma^2}\sum_{k=1}^{K} \frac{\partial^2 y(k;\theta)}{\partial\theta_j \partial\theta_i} e(k;\theta).$$

the second term is often neglected (this avoids the need to compute the second derivative of the model output), so that

$$\frac{\partial^2 J(\theta)}{\partial\theta_j \partial\theta_i} \simeq \frac{1}{\sigma^2}\sum_{k=1}^{K} \frac{\partial y(k;\theta)}{\partial\theta_j}\frac{\partial y(k;\theta)}{\partial\theta_i}.$$

Note that the approximate hessian is still symmetric.

The resulting approximate algorithm is called Gauss-Newton.

An optimization scheme for the OE method

To complete the definition of the method we need a scheme to compute the sensitivities of the model output with respect to the parameters.

This can be done either numerically or analytically.

The numerical approach is unavoidable whenever nonlinear models are considered.

Sensitivities can be computed

- Using forward differences
- Using central differences.

An optimization scheme for the OE method

Using forward differences we get

$$\frac{\partial y(k;\theta)}{\partial \theta_j} = \frac{y(k;\theta+\delta\theta_j) - y(k;\theta)}{\delta\theta_j}, \qquad j = 1, \ldots, n_\theta.$$

the perturbation should be small – general guideline: 1% of the current value of the parameter component.

Clearly the computation of the vector of sensitivities requires

$$n_\theta + 1$$

simulations of the response of the model to the sampled input.

An optimization scheme for the OE method

Using central differences instead we get

$$\frac{\partial y(k;\theta)}{\partial \theta_j} = \frac{y(k;\theta+\delta\theta_j) - y(k;\theta-\delta\theta_j)}{2\delta\theta_j}, \qquad j = 1, \ldots, n_\theta.$$

In this case the computation of the vector of sensitivities requires

$$2n_\theta$$

simulations of the response of the model to the sampled input, but the computed sensitivities are significantly more accurate.

An optimization scheme for the OE method

The analytical approach, on the other hand, starts from the model equations:

$$\dot{x} = f(x, u; \theta), \qquad x(0) = x_0$$
$$y = g(x, u; \theta).$$

Differentiating with respect to a component of the parameter vector we get for the state equation

$$\frac{\partial}{\partial \theta_j}\frac{dx}{dt} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial \theta_j} + \frac{\partial f}{\partial u}\frac{\partial u}{\partial \theta_j} + \frac{\partial f}{\partial \theta_j}, \qquad \frac{\partial x(0)}{\partial \theta_j} = 0$$

and note that

$$\frac{\partial f}{\partial u}\frac{\partial u}{\partial \theta_j} = 0.$$

An optimization scheme for the OE method

Interchanging derivatives on the left had side we get:

$$\frac{d}{dt}\left(\frac{\partial x}{\partial \theta_j}\right) = \left(\frac{\partial f}{\partial x}\right)\frac{\partial x}{\partial \theta_j} + \frac{\partial f}{\partial \theta_j}, \qquad \frac{\partial x(0)}{\partial \theta_j} = 0, \qquad j = 1, \ldots, n_\theta$$

and similarly for the output equation

$$\left(\frac{\partial y}{\partial \theta_j}\right) = \left(\frac{\partial g}{\partial x}\right)\frac{\partial x}{\partial \theta_j} + \frac{\partial g}{\partial \theta_j}, \qquad j = 1, \ldots, n_\theta.$$

Therefore, it is possible to compute the required sensitivities by simulating the state space models defined by the above state and output equations.

Local and global minima

- The developed optimisation scheme is *local,* in the sense that at each iteration only point-wise information on the derivatives of the cost are used.

- This means that in general the algorithm may converge to a different solution depending on the initial guess for the parameters.

- This is a key issue with the OE method:
  - a reliable initial guess for the parameters is necessary
  - careful inspection of the computed estimates is also necessary, to ensure they are physically meaningful.

Asymptotic variance of the OE estimates

ML estimators are efficient, so we expect that

$$Var[\widehat{\theta}_K] \underset{K\to\infty}{\rightarrow} M^{-1}$$

where

$$M = -E[\frac{\partial^2 \log f}{\partial\theta^2}].$$

How can we evaluate *M*?

We reason as follows:

- The estimate has been chosen so as to maximise the likelihood of the data;
- Therefore maximal probability (corresponding to the expected value) should be attained at the optimal likelihood.

Asymptotic variance of the OE estimates

As a consequence we can make the following approximation:

$$M = -E[\frac{\partial^2 \log f}{\partial\theta^2}] \simeq -\frac{\partial^2 \log L}{\partial\theta^2}|_{\theta=\theta^\star}.$$

Note that for the OE problem

$$M = -E[\frac{\partial^2 \log f}{\partial\theta^2}] = -E[\frac{\partial^2 \left(-\log(\sigma\sqrt{2\pi})^K - J(\theta)\right)}{\partial\theta^2}] \simeq$$
$$\simeq \frac{1}{\sigma^2}\sum_{k=1}^{K} \frac{\partial y(k;\theta)}{\partial\theta}\left(\frac{\partial y(k;\theta)}{\partial\theta}\right)^T$$

Asymptotic variance of the OE estimates

Therefore we can evaluate *M* as

$$M \simeq \frac{1}{\sigma^2} \sum_{k=1}^{K} \frac{\partial y(k;\theta)}{\partial \theta} \left(\frac{\partial y(k;\theta)}{\partial \theta} \right)^T |_{\theta=\theta^\star}.$$

Finally, note that the noise variance is also needed.

If it is not known, it can be estimated using the sample variance, as in the preliminary examples on ML estimation:

$$\hat{\sigma}_K^2 = \frac{1}{K} \sum_{i=1}^{K} e^2(k) = \frac{1}{K} \sum_{i=1}^{K} (y_m(k) - y(k;\theta^\star))^2.$$

Confidence intervals for the estimates

- The theory of ML estimation ensures that, asymptotically, estimates are unbiased and achieve the C-R variance bound.

- Therefore, asymptotically

$$\widehat{\theta}_N \underset{N\to\infty}{\to} G(\theta, M^{-1}) \quad \text{in distribution.}$$

- As a consequence, having obtained an estimate from a given dataset, we can define *confidence intervals* using properties of Gaussian densities.

Confidence intervals for the estimates

- More precisely, letting

$$C = M^{-1} = \{c_{ij}\}$$

- We have, element-wise, that

$$\widehat{\theta}_{iK} \underset{K\to\infty}{\rightarrow} G(\theta_i, c_{ii}) \quad \text{in distribution} \quad i = 1, \dots, n_\theta.$$

- And in terms of probabilities:

$$\begin{aligned}
-\sqrt{c_{ii}} < & \widehat{\theta}_{iK} - \theta_i < +\sqrt{c_{ii}} \quad \text{with prob. } 68\ \% \\
-2\sqrt{c_{ii}} < & \widehat{\theta}_{iK} - \theta_i < +2\sqrt{c_{ii}} \quad \text{with prob. } 95\ \% \\
-3\sqrt{c_{ii}} < & \widehat{\theta}_{iK} - \theta_i < +3\sqrt{c_{ii}} \quad \text{with prob. } 99.7\ \%
\end{aligned}$$

Multiple output case

- So far only the case of a scalar measurement has been considered, for the sake of simplicity.

- In real problems, however, vectors of measurements must be used, so the output equation is

$$y = g(x, u; \theta), \qquad y \in \mathbb{R}^p$$

- And the measurement model becomes

$$y_m(k) = y(k) + v(k)$$

$$y(k) = y(kT_s)$$

$$v(k) = G(0, R), \qquad E[v(i)v(j)^T] = 0, \qquad i \neq j$$

Multiple output case

- Depending on the specific problem, the noise variance R can range from
  - A diagonal matrix, in the case of uncorrelated measurements (individual components of the output provided by p different sensors)
  - A block-diagonal matrix, in the case of partially correlated measurements (subvectors of the output provided by different sensors)
  - Full matrix, in the case of fully correlated measurements.
- The first two cases are the most common in practice.

Multiple output case

- Therefore, the density of the measurement noise is given by

$$f_v(q) = \frac{1}{(\sqrt{\mathrm{det} R})(2\pi)^{p/2}} e^{-\frac{1}{2}q^T R^{-1} q}, \qquad q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_p \end{bmatrix}$$

and following the same derivation as in the scalar output case, the density of the measurements results

$$f_{y_m(k)}(q) = \frac{1}{(\sqrt{\mathrm{det} R})(2\pi)^{p/2}} e^{-(q-y(k;\theta))^T R^{-1}(q-y(k;\theta)}$$

Multiple output case

- The likelihood is constructed as before, to get

$$L(y_m(1), \dots, y_m(K); \theta) =$$

$$= \frac{1}{\left[(\sqrt{\mathrm{det} R})(2\pi)^{p/2}\right]^K} e^{-\frac{1}{2}\sum_{k=1}^{K}(y_m(k)-y(k;\theta))^T R^{-1}(y_m(k)-y(k;\theta))}$$

and the cost function to be minimised becomes

$$J(\theta) = \frac{1}{2}\sum_{k=1}^{K}(y_m(k) - y(k;\theta))^T R^{-1}(y_m(k) - y(k;\theta))$$

Multiple output case

- Note that if *R* is diagonal the cost reduces to the sum of *p* costs for each component of the output:

$$J(\theta) = \frac{1}{2}\sum_{k=1}^{K}(y_m(k) - y(k;\theta))^T R^{-1}(y_m(k) - y(k;\theta)) =$$

$$= J_1(\theta) + J_2(\theta) + \ldots + J_p(\theta)$$

where

$$J_i(\theta) = \frac{1}{2\sigma_i^2}\sum_{k=1}^{K} e_i(k;\theta)^2, \quad i = 1, \ldots, p.$$

- This highlights the importance of proper scaling of the measurements in the formulation of the problem.

Multiple output case

- Finally, the gradient and the hessian of the cost ans the Fischer information matrix can be generalised as

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\sum_{k=1}^{K} \left(\frac{\partial y(k;\theta)}{\partial \theta_j} \right)^T R^{-1} e(k;\theta).$$

$$\frac{\partial^2 J(\theta)}{\partial \theta_j \partial \theta_i} \simeq \sum_{k=1}^{K} \left(\frac{\partial y(k;\theta)}{\partial \theta_j} \right)^T R^{-1} \left(\frac{\partial y(k;\theta)}{\partial \theta_i} \right).$$

$$M \simeq \frac{1}{\sigma^2} \sum_{k=1}^{K} \frac{\partial y(k;\theta)}{\partial \theta} \left(\frac{\partial y(k;\theta)}{\partial \theta} \right)^T |_{\theta=\theta^\star}.$$