Ill conditioning

Contents map

developed content units	taxonomy levels
ill conditioning	u1, e1
ill posedness	u1, e1

prerequisite content units	taxonomy levels
least squares regression	u1, e1

Main ILO of sub-module "Ill conditioning"

Describe what ill conditioning and ill posedness mean, in the context of system identification

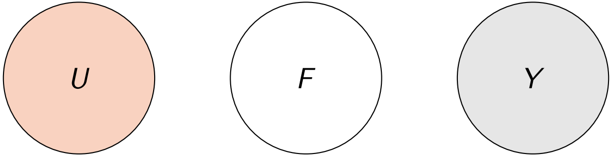
Recognize when ill conditioning may happen in practice

Starting point: system identification

starting from

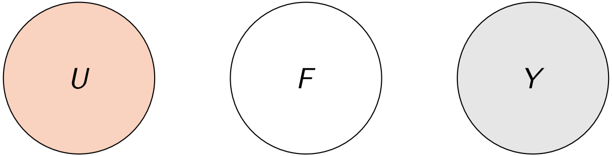
$$y[k] = f\Big(u[k], u[k-1], \dots\Big) + d[k] \qquad \mathcal{D} = \{u[k], y[k]\}_{k \in \mathcal{K}}$$

identify the model $f(\cdot)$



Definition of ill-posedness and ill-conditioning

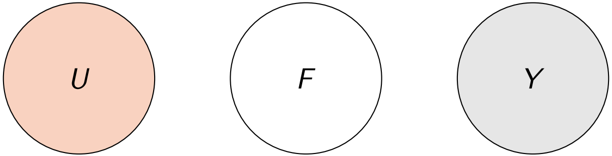
$$y[k] = f\Big(u[k], u[k-1], \ldots\Big) + d[k] \qquad \mathcal{D} = \{u[k], y[k]\}_{k \in \mathcal{K}}$$



- *ill-posed problem (in the Hadamard sense):* solution is either not unique or does not depend continuously on the data

Definition of ill-posedness and ill-conditioning

$$y[k] = f\Big(u[k], u[k-1], \dots\Big) + d[k] \qquad \mathcal{D} = \{u[k], y[k]\}_{k \in \mathcal{K}}$$

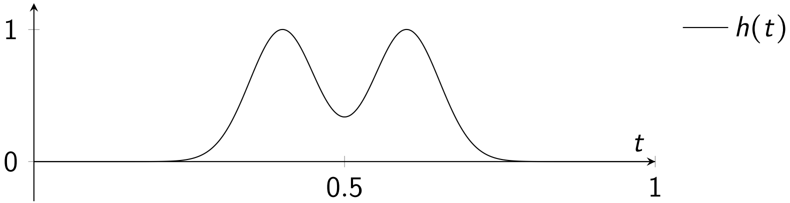


- *ill-posed problem (in the Hadamard sense):* solution is either not unique or does not depend continuously on the data
- *ill-conditioned problem:* solution is very sensitive to the data

Example: the Hunt reconstruction problem

(continuous-time LTI with sampled output)

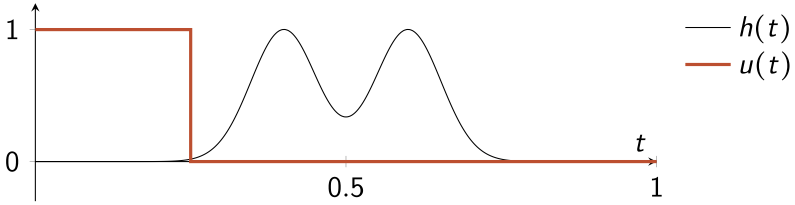
$$h(t) = \exp\left(-\left(\frac{t-0.4}{0.075}\right)^2\right) + \exp\left(-\left(\frac{t-0.6}{0.075}\right)^2\right)$$



Example: the Hunt reconstruction problem

(continuous-time LTI with sampled output)

$$h(t) = \exp\left(-\left(\frac{t-0.4}{0.075}\right)^2\right) + \exp\left(-\left(\frac{t-0.6}{0.075}\right)^2\right) \qquad u(t) = \begin{cases} 1 & \text{if } 0 \le t \le 0.25 \\ 0 & \text{otherwise} \end{cases}$$

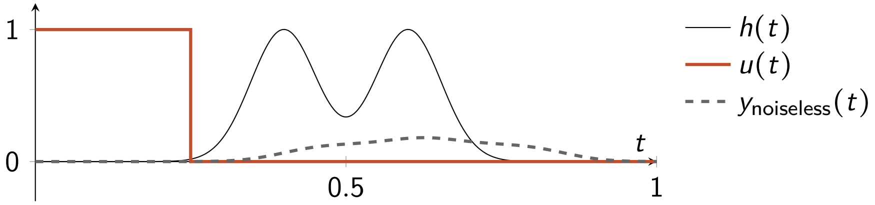


Example: the Hunt reconstruction problem

(continuous-time LTI with sampled output)

$$h(t) = \exp\left(-\left(\frac{t-0.4}{0.075}\right)^2\right) + \exp\left(-\left(\frac{t-0.6}{0.075}\right)^2\right) \qquad u(t) = \begin{cases} 1 & \text{if } 0 \le t \le 0.25 \\ 0 & \text{otherwise} \end{cases}$$

$$y_{\text{noiseless}}(t) = \int_0^{+\infty} h(\tau)u(t-\tau)d\tau$$

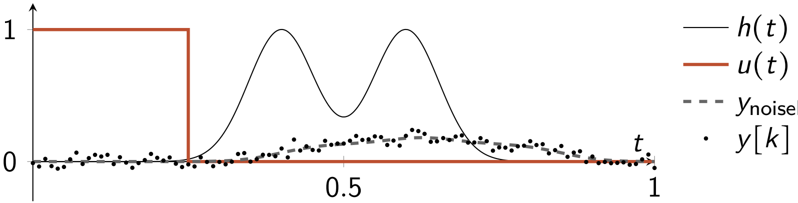


Example: the Hunt reconstruction problem

(continuous-time LTI with sampled output)

$$h(t) = \exp\left(-\left(\frac{t-0.4}{0.075}\right)^2\right) + \exp\left(-\left(\frac{t-0.6}{0.075}\right)^2\right) \qquad u(t) = \begin{cases} 1 & \text{if } 0 \le t \le 0.25 \\ 0 & \text{otherwise} \end{cases}$$

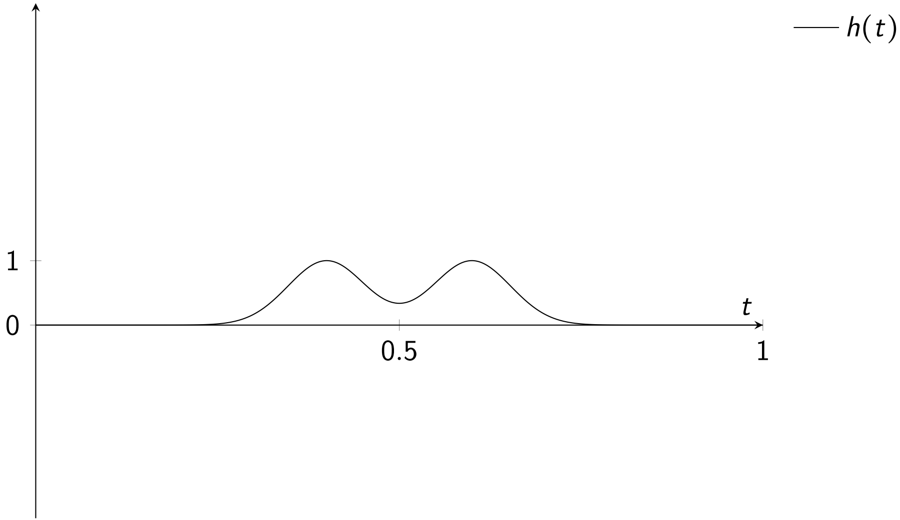
$$y_{\text{noiseless}}(t) = \int_0^{+\infty} h(\tau)u(t-\tau)d\tau \qquad y[k] = y_{\text{noiseless}}[k] + v(k)$$



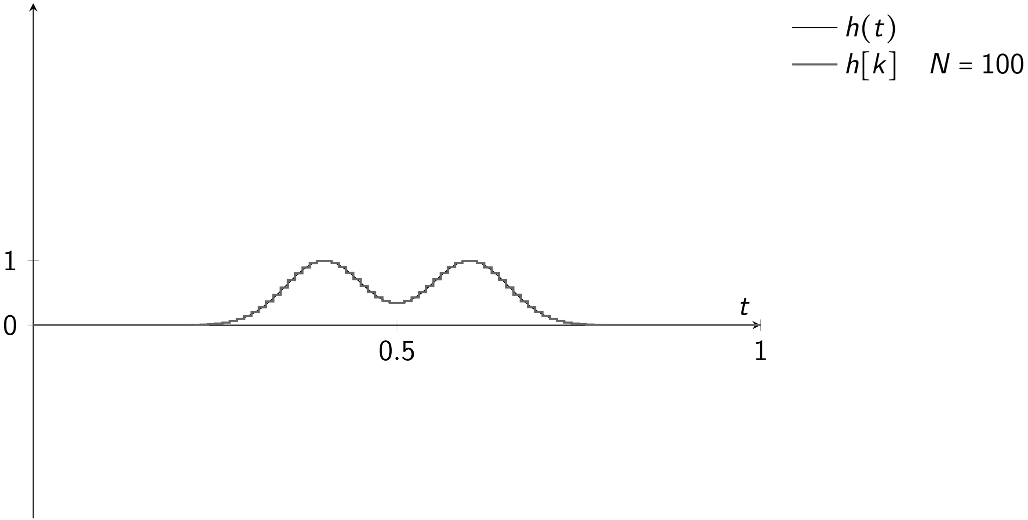
This problem can be solved with linear algebra!

$$\boldsymbol{y} = U\boldsymbol{h} + \boldsymbol{d} \quad \Rightarrow \quad \widehat{\boldsymbol{h}} = \left(U^T U\right)^{-1} U^T \boldsymbol{y}$$

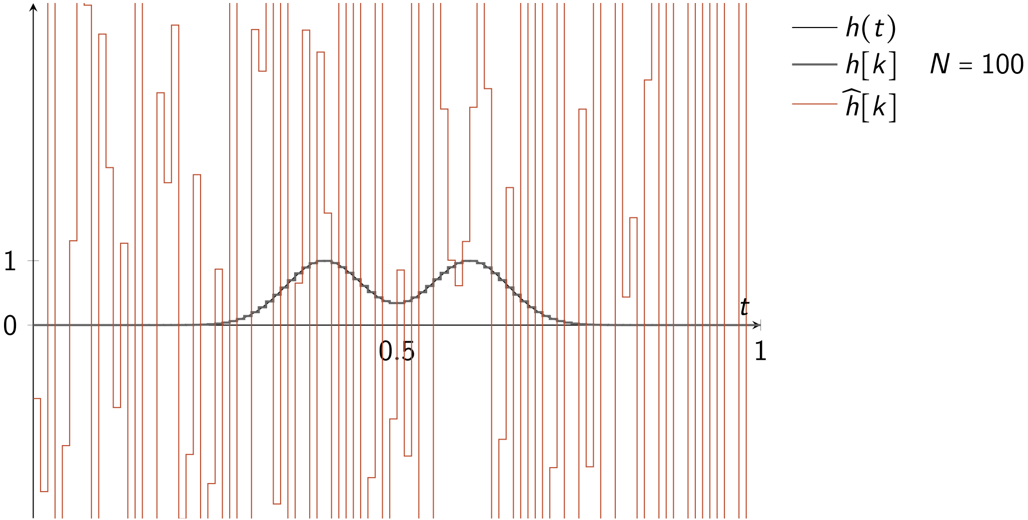
Is the Hunt reconstruction problem well defined?



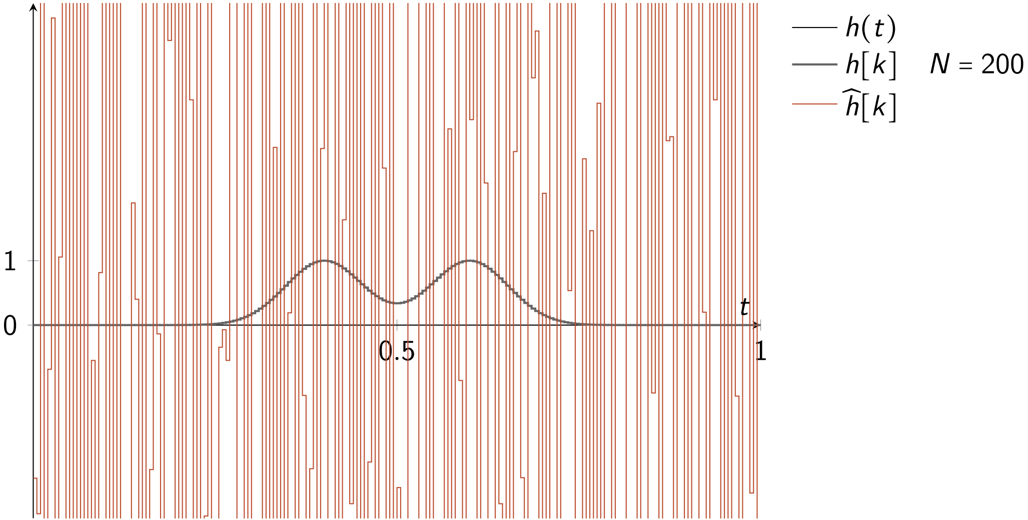
Is the Hunt reconstruction problem well defined?



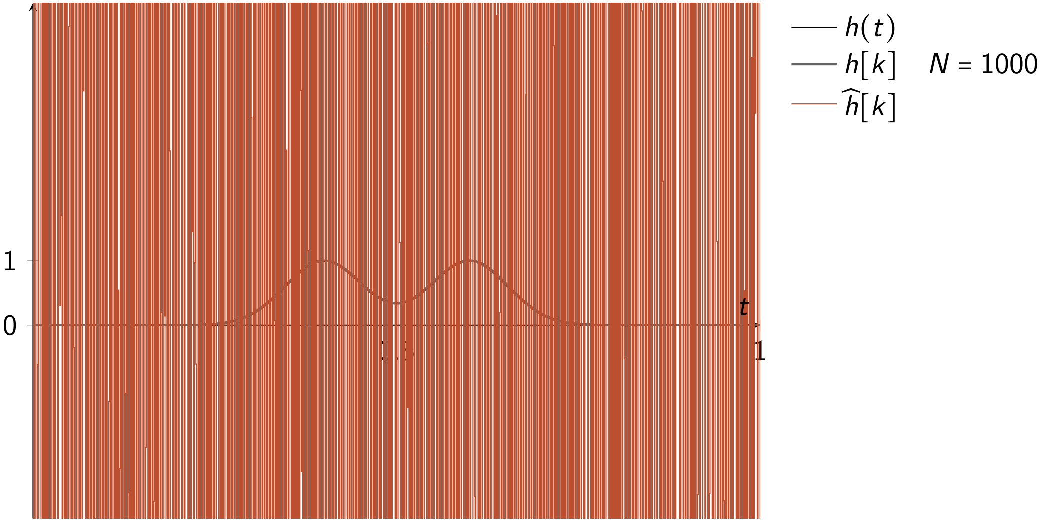
Is the Hunt reconstruction problem well defined?



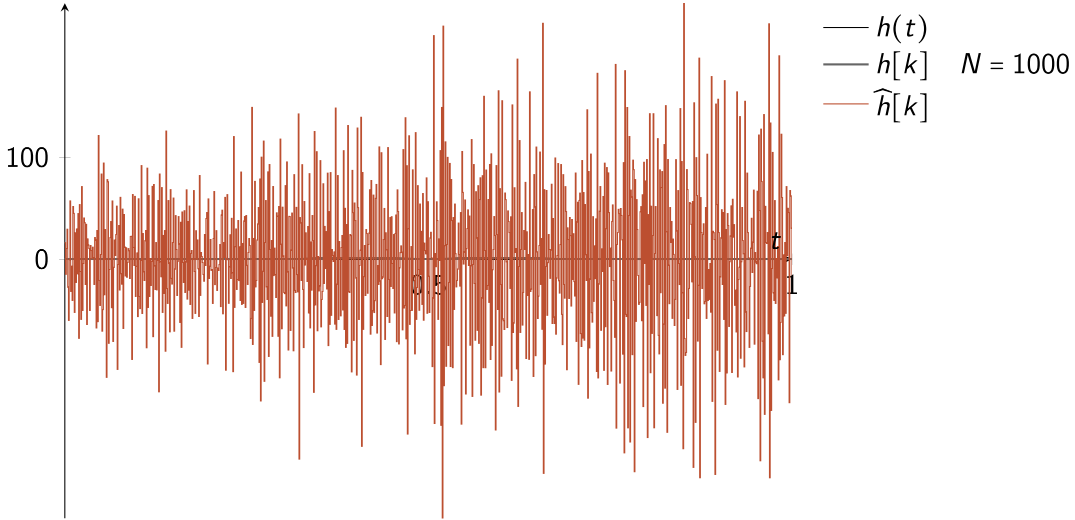
Is the Hunt reconstruction problem well defined?



Is the Hunt reconstruction problem well defined?



Is the Hunt reconstruction problem well defined?



What is happening?

$$\boldsymbol{e} = \boldsymbol{h} - \widehat{\boldsymbol{h}} = U^{-1}\boldsymbol{d}$$

$$\frac{\|\boldsymbol{e}\|}{\|\boldsymbol{h}\|} \leq \frac{\sigma_{\max}(U)}{\sigma_{\min}(U)} \frac{\|\boldsymbol{d}\|}{\|U\boldsymbol{h}\|}$$

- the slower u the higher $\frac{\sigma_{\max}(U)}{\sigma_{\min}(U)}$
- the faster Δ the higher $\frac{\sigma_{\max}(U)}{\sigma_{\min}(U)}$

how can we improve our estimates?

$\mapsto$ regularization

Summarizing

Describe what ill conditioning and ill posedness mean, in the context of system identification

Recognize when ill conditioning may happen in practice

- TODO

Most important python code for this sub-module

Linear algebra tools

- `numpy.linalg.solve`
- `numpy.linalg.inv`

Self-assessment material

Question 1

Which of the following best describes the difference between an ill-posed and an ill-conditioned problem in system identification?

Potential answers:

I: Ill-conditioned problems have no solution, while ill-posed problems have too many.

II: Ill-posed problems may lack uniqueness or continuous dependence on the data, while ill-conditioned problems are extremely sensitive to small changes in data.

III: Ill-posed problems always have unstable solutions, while ill-conditioned ones always diverge.

IV: Ill-conditioning is due to randomness in the input, while ill-posedness is due to measurement noise.

V: I do not know

Question 2

Why does the Hunt reconstruction problem become ill-conditioned as the length of the input increases?

Potential answers:

I: Because more data always makes the system overdetermined.

II: Because slow or non-diverse input signals lead to poor numerical conditioning of the matrix U.

III: Because increasing the number of samples reduces the noise-to-signal ratio.

IV: Because the model structure becomes nonlinear with large N.

V: I do not know

Question 3

In the context of system identification, what does the condition number $\frac{\sigma_{\max}(U)}{\sigma_{\min}(U)}$ represent?

Potential answers:

I: The maximum amplification of relative errors in the data to the estimation error.
II: The rate of convergence of the optimization algorithm used.
III: The ratio between input and output power in the system.
IV: The likelihood that a model is nonlinear.
V: I do not know

Question 4

What is a practical way to reduce ill-conditioning in system identification?

Potential answers:

I: Use richer or faster-varying input signals during data collection.
II: Use fewer data points to simplify the estimation problem.
III: Reduce the noise artificially in the measurements after data collection.
IV: Make the input signal constant over time to ensure stability.
V: I do not know

Question 5

Why is regularization used when solving ill-conditioned system identification problems?

Potential answers:

I: To make the inverse of U exactly equal to zero.

II: To stabilize the solution by penalizing large parameter values or enforcing smoothness.

III: To reduce the condition number by artificially shrinking the data.

IV: To avoid computing the inverse of the matrix altogether.

V: I do not know

Recap of sub-module “Ill conditioning”

- Ill-posed problems may lack a solution, have multiple solutions, or be highly sensitive to small changes in data
- Ill-conditioned problems have a solution, but it is numerically unstable and highly sensitive to input errors
- The condition number of a matrix quantifies the degree of ill-conditioning; a high condition number indicates poor numerical stability
- In system identification, slowly varying or insufficiently rich input signals can lead to ill-conditioning
- Regularization techniques can mitigate the effects of ill-conditioning by introducing stability through additional constraints
- Choosing appropriate input signals is critical to ensuring well-posed and well-conditioned identification problems
- Understanding the structure and properties of the data matrix (e.g., U in least squares problems) is essential to diagnose ill-conditioning

?