Nonparametric kernel-based system identification

Giulio Bottegal

Eindhoven University of Technology

System identification – DISC course 2020

disc

Outline

disc

Part I:

- Least squares
- Regularized least squares
- Bias, variance and mean square error

Part II:

- Kernels for system identification
- Connection with Bayesian estimation
- Tuning the hyperparameters
- Examples

disc

PART I

Nonparametric vs parametric

disc

Parametric system identification

1. Choose model structure (system order)
2. Identify parameters (PEM)
3. If uncertain on model structure, validate your choice
   $\hookrightarrow$ can be a tedious/costly step

Nonparametric system identification

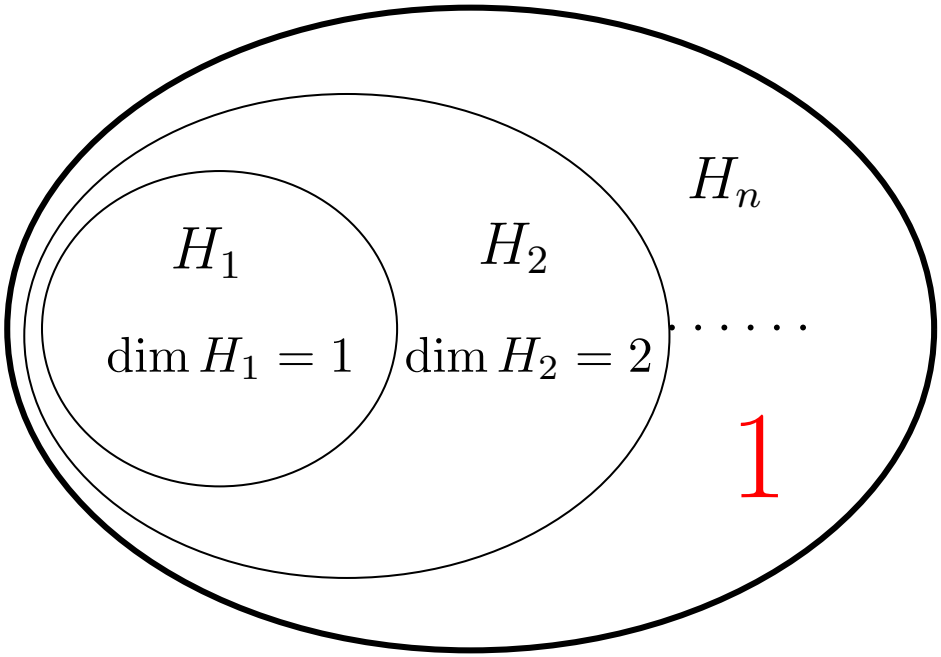
Estimate your transfer function/impulse response pointwise

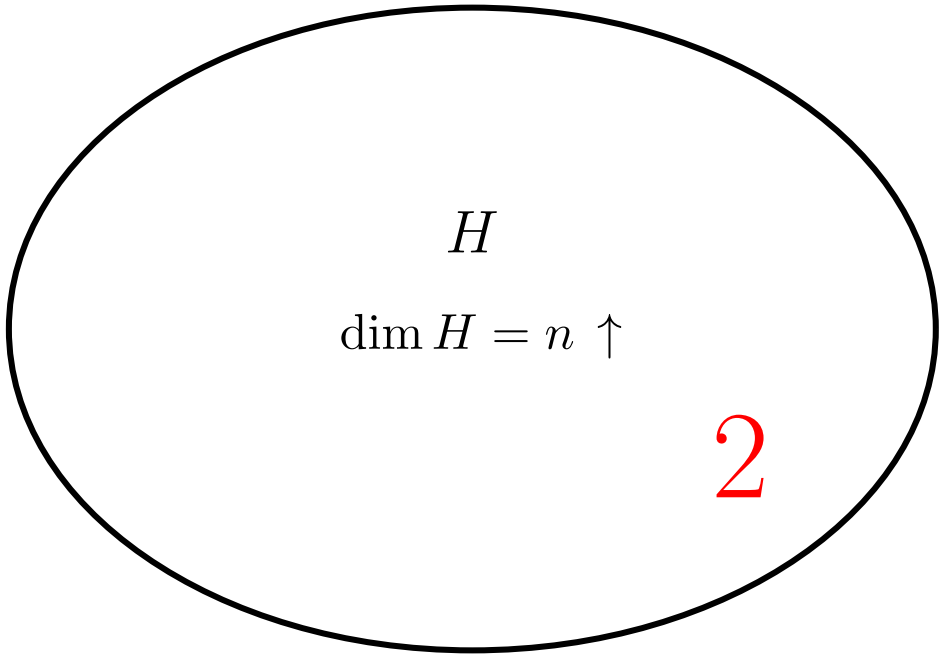
We may avoid some of the tedious/costly steps of parametric identification... but is it just as easy?

Nonparametric vs parametric

disc

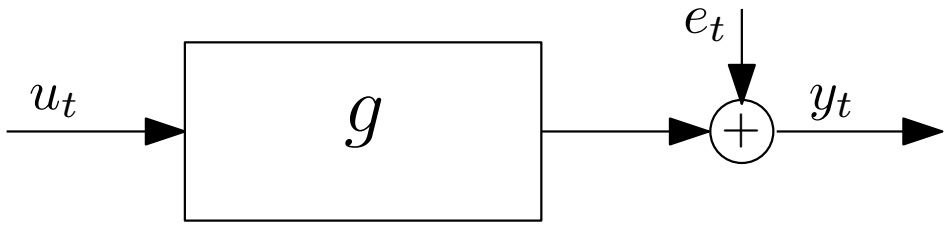
1. Parametric approach: $G_0 \in \mathcal{M}$ (lot of prior information!)
   $\hookrightarrow$ may require model order selection (choice of $\mathcal{M}$)
2. Nonparametric approach: Pointwise description of G_0
   $\hookrightarrow$ Large hypothesis space $\longrightarrow$ is that good?





Nonparametric time-domain system identification

disc



We focus on SISO output-error models

$$y(t) = G_0(z)u(t) + e(t) \qquad e(t) \text{ white noise with variance } \sigma^2$$

rewritten in the time domain

$$y(t) = \sum_{k=1}^{+\infty} g_0(k)u(t-k) + e(t)$$

Assumption

The system is BIBO stable.

$\hookrightarrow$ for n large enough, $g_0(n+1) \simeq 0$

$\hookrightarrow$ we shall focus on $g_0(1), \ldots, g_0(n)$

Time-domain nonparametric identification

disc

We assume that we have collected N input-output measurements. Define:

$$\phi^T(t) = [u(t-1) \ \ldots \ u(t-n)] \qquad \text{(assume } u(t) = 0 \text{ for } t \leq 0)$$

$$y = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} \quad U = \begin{bmatrix} \phi^T(1) \\ \vdots \\ \phi^T(N) \end{bmatrix} \quad g_0 = \begin{bmatrix} g_0(1) \\ \vdots \\ g_0(n) \end{bmatrix} \quad e = \begin{bmatrix} e(1) \\ \vdots \\ e(N) \end{bmatrix}$$

Then:

$$y = Ug_0 + e$$

(essentially, a long FIR model that approximates the true system)

Least squares identification

disc

How to identify the system? Least squares ($\equiv$ PEM in the FIR case)

$$\begin{aligned}\hat{g} &= \arg\min_{g} \|y - Ug\|_2^2 \\ &= (U^T U)^{-1} U^T y\end{aligned}$$

Properties (efficient estimator)

- It is unbiased (neglecting $g_0(n+1), \ldots, g_0(\infty)$):

$$\mathbf{E}[\hat{g}] = g_0$$

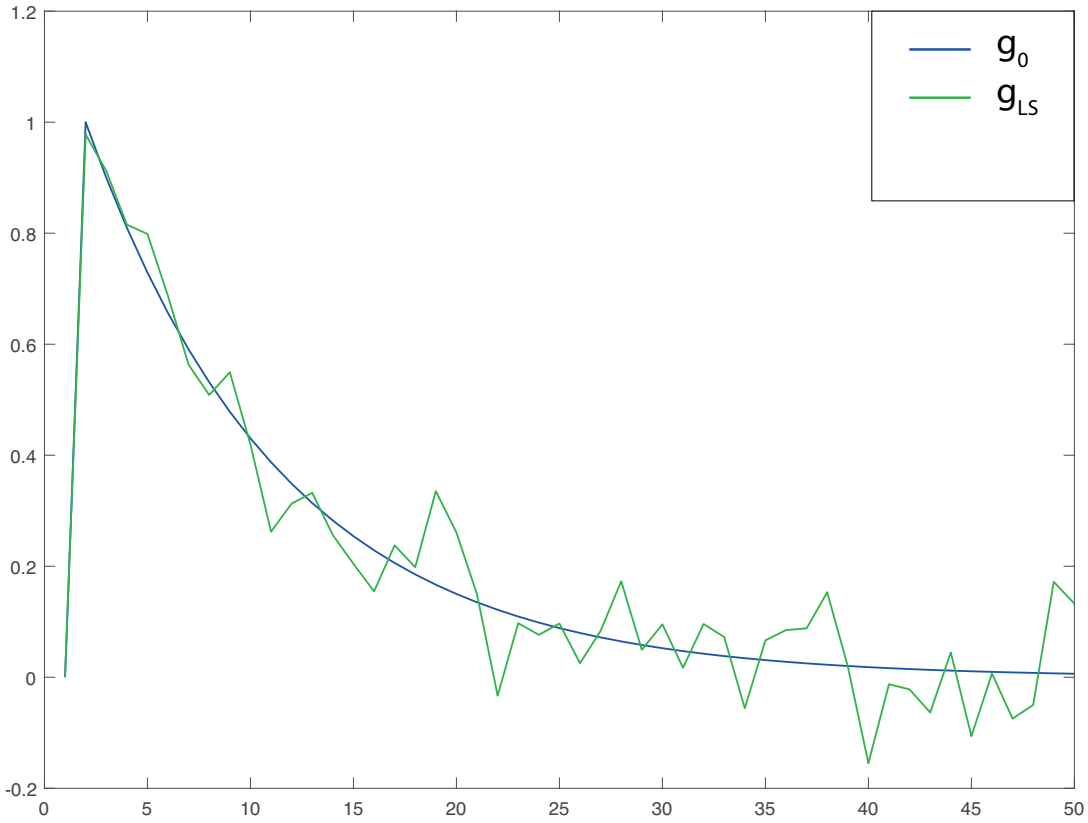
- Its covariance achieves the Cramer-Rao lower bound:

$$\mathbf{E}[(\hat{g} - \mathbf{E}[\hat{g}])(\hat{g} - \mathbf{E}[\hat{g}])^T] = \sigma^2 (U^T U)^{-1}$$

Are we happy with least squares?

disc

Least squares with different realizations of noise ($1\times$)

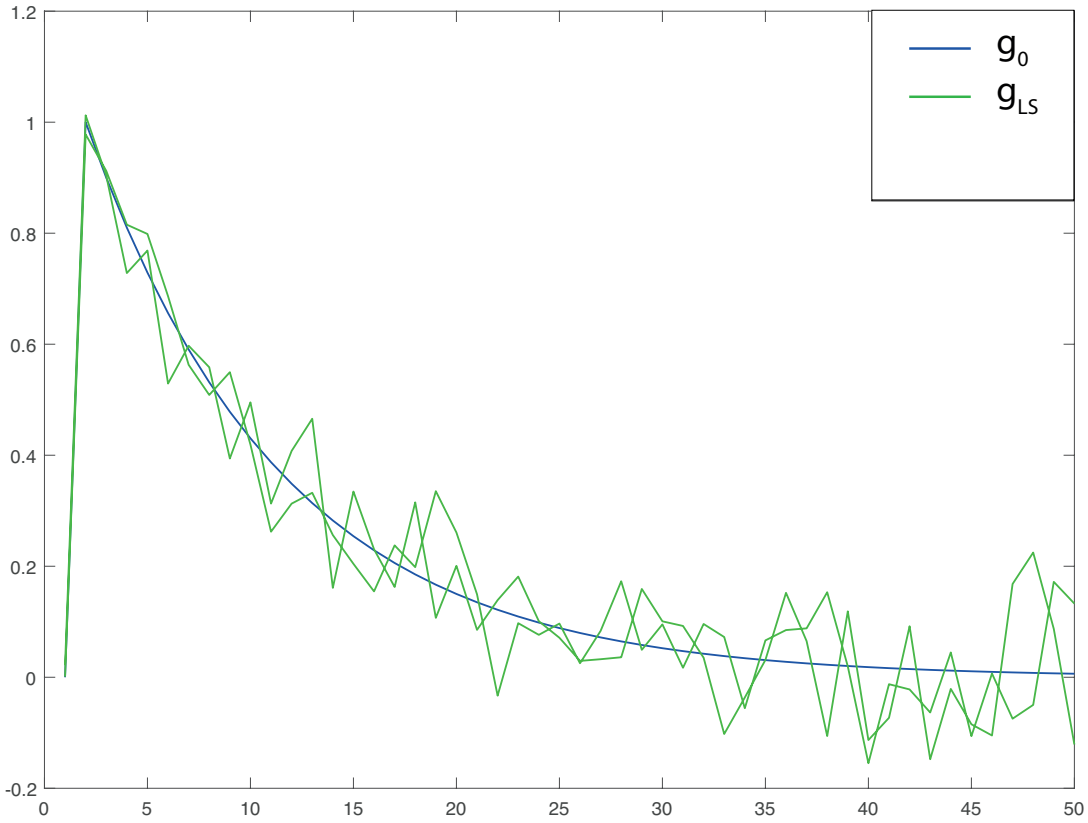


$u =$`randn(N,1)`, $N = 150$, $n = 50$, $\sigma^2 = 0.5$

Are we happy with least squares?

disc

Least squares with different realizations of noise ($2\times$)

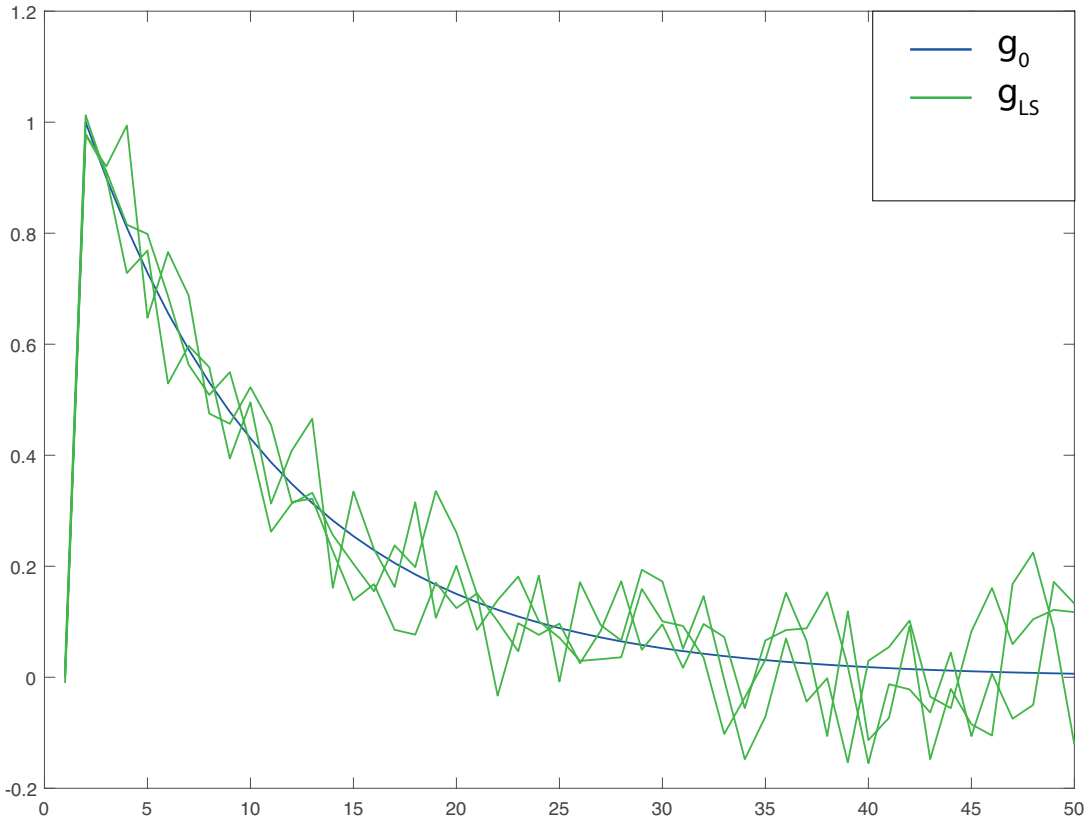


$u =$`randn(N,1)`, $N = 150$, $n = 50$, $\sigma^2 = 0.5$

Are we happy with least squares?

disc

Least squares with different realizations of noise ($3\times$)

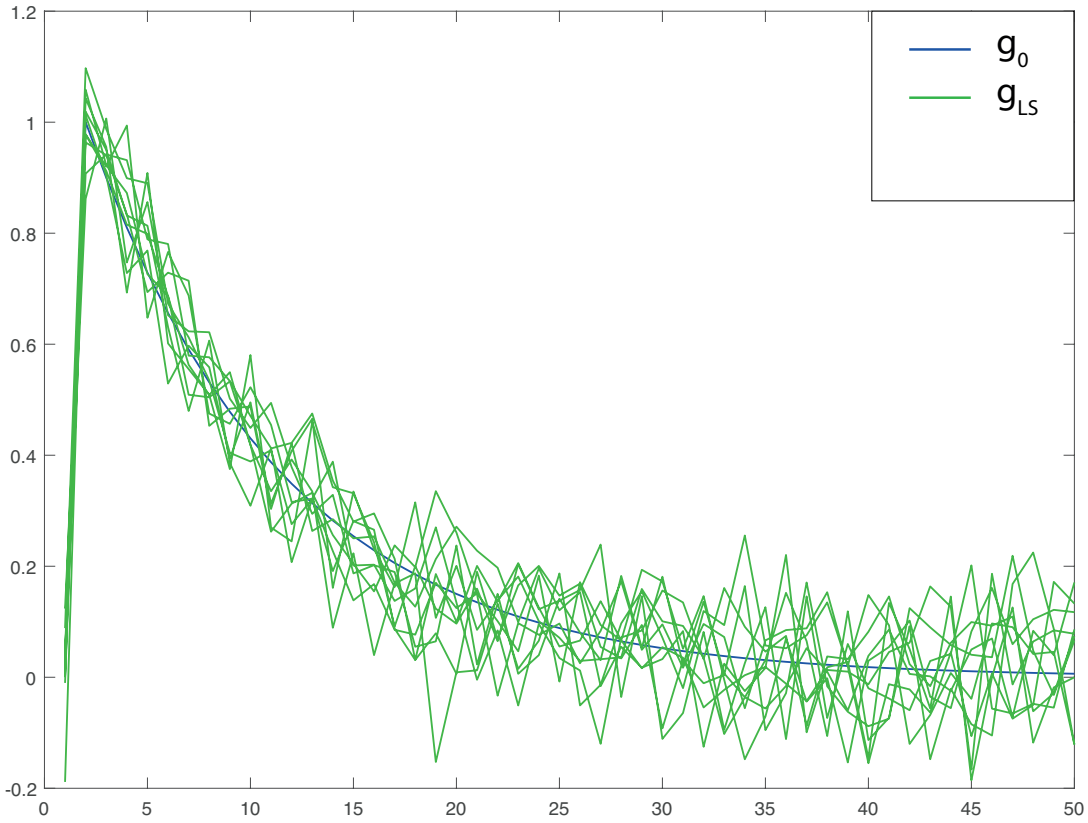


$u =$`randn(N,1)`, $N = 150$, $n = 50$, $\sigma^2 = 0.5$

Are we happy with least squares?

disc

Least squares with different realizations of noise ($10\times$)



$u =$`randn(N,1)`, $N = 150$, $n = 50$, $\sigma^2 = 0.5$

Are we happy with least squares?

disc

Key point: Least squares estimates may have high variance

Example

Let $u(t)$ be white noise with variance ρ. Then the variance ($= \mathrm{tr}[\mathrm{Cov}]$) is

$$\mathbf{E}[(\hat{g} - \mathbf{E}[\hat{g}])^T(\hat{g} - \mathbf{E}[\hat{g}])] = \mathrm{tr}[\sigma^2(U^T U)^{-1}] \simeq \sigma^2 \mathrm{tr}[(\rho N I_n)^{-1}] = \frac{\sigma^2}{\rho}\frac{n}{N}$$

Therefore, we have unsatisfactory results when:

1. The SNR is low
2. The length of the impulse response is high
3. The data length is short

Reducing the variance

disc

Can we modify least squares to have lower variance?

Let's try, for instance, a new cost function

$$J(g) = \|y - Ug\|_2^2 + \gamma \|g\|_{K^{-1}}^2$$

Regularized least squares:

- $\|y - Ug\|_2^2 \quad \longrightarrow$ least squares term
- $\|g\|_{K^{-1}}^2 = g^T K^{-1} g \quad \longrightarrow$ regularization term, K is a called a kernel
- $\gamma > 0 \quad \longrightarrow$ regularization parameter

γ and K are user parameters (we will see later how to choose them)

Regularized least squares

Reducing the variance

disc

Our estimates is now

$$\begin{aligned}\hat{g} &= \arg\min_g J(g) \\ &= (U^T U + \gamma K^{-1})^{-1} U^T y\end{aligned}$$

How do we get this solution?

$$\frac{\partial J}{\partial g} = -U^T 2(y - Ug) + 2\gamma K^{-1} g$$

The solution follows setting the derivative to 0 and solving for g.

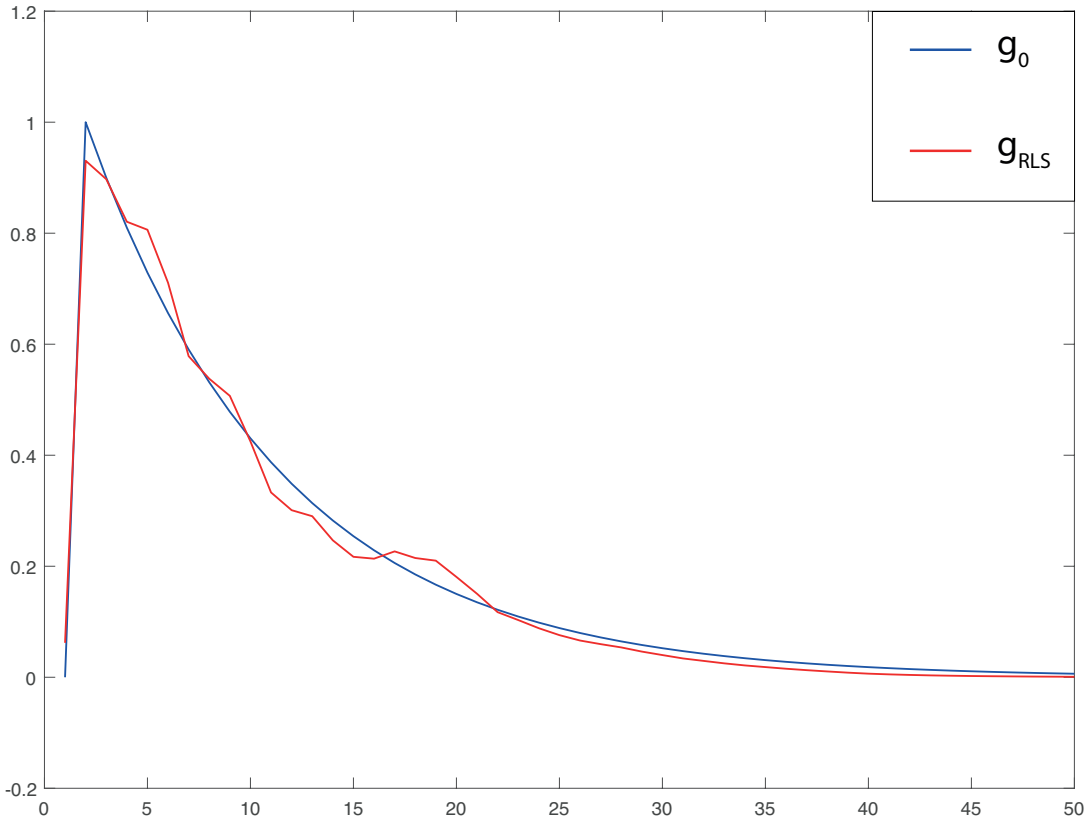
Alternatively, solve by LS

$$\begin{bmatrix} y \\ 0 \end{bmatrix} = \begin{bmatrix} U \\ \sqrt{\gamma} K^{-(1/2)} \end{bmatrix} g + e$$

Are we happy with this estimator?

disc

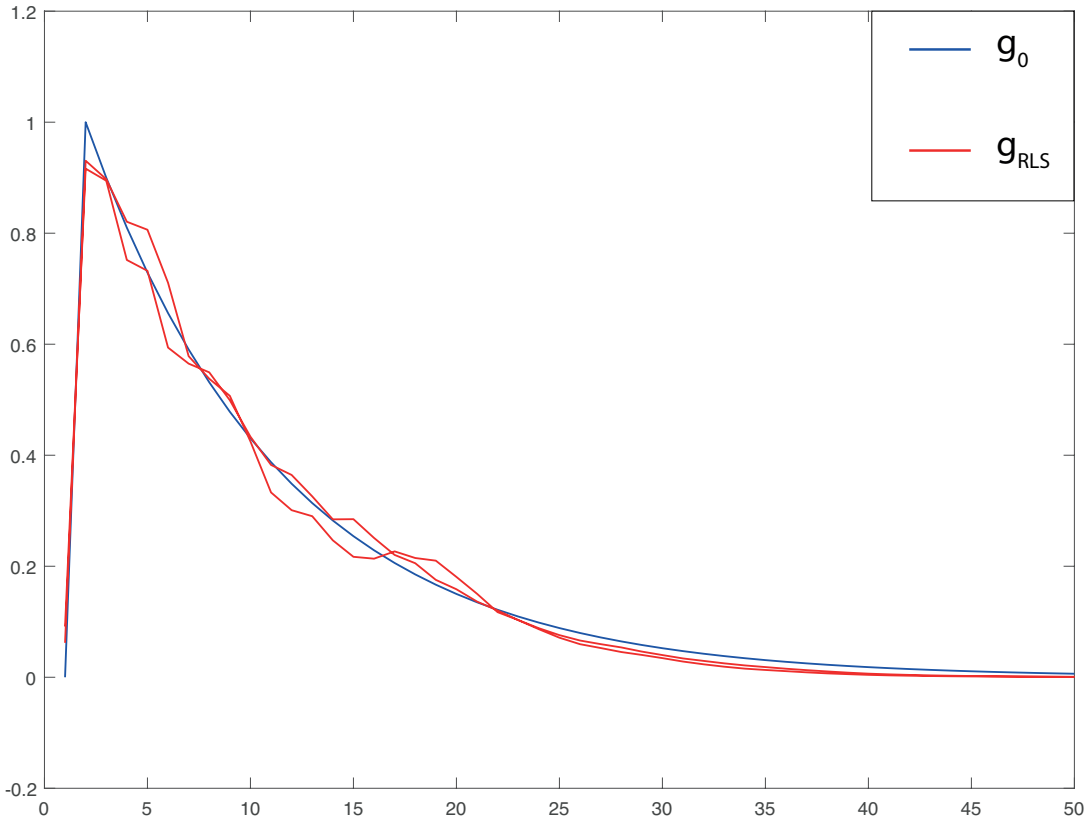
Regularized least squares with different realizations of noise ($1\times$)



$u =$`randn(N,1)`, $N = 150$, $n = 50$, $\sigma^2 = 0.5$

Are we happy with this estimator?

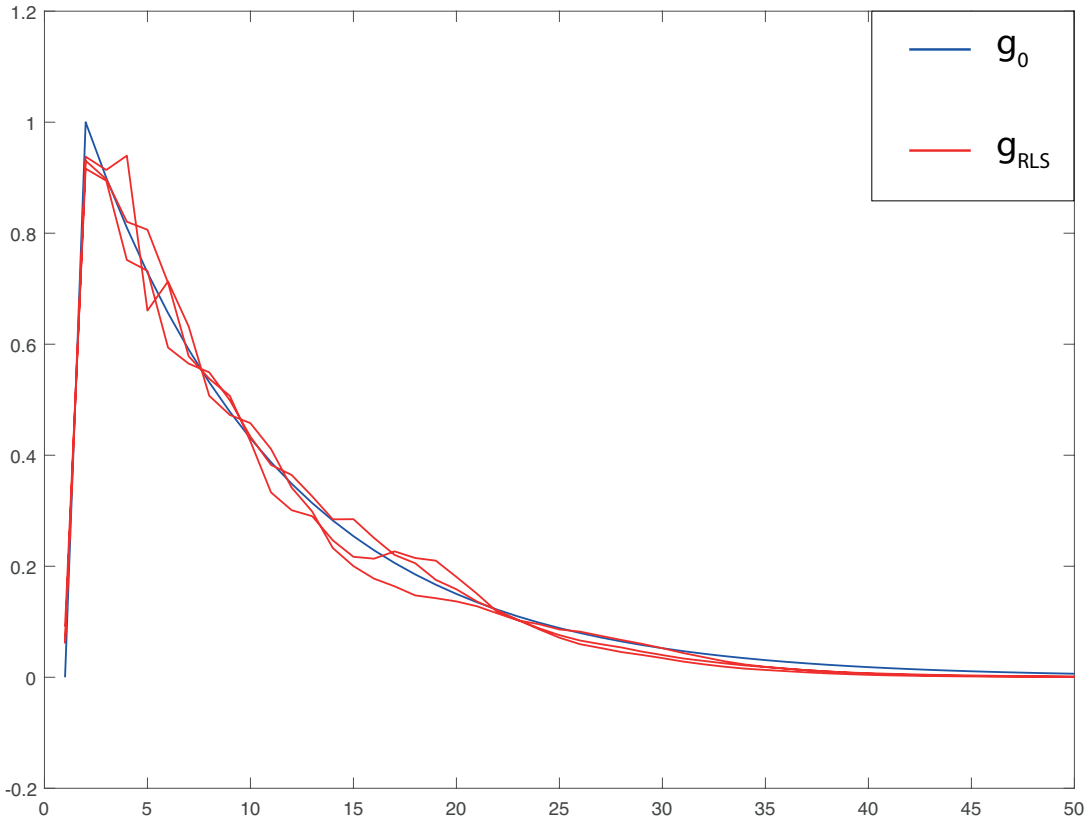
Regularized least squares with different realizations of noise ($2\times$)



$u = \texttt{randn(N,1)}$, $N = 150$, $n = 50$, $\sigma^2 = 0.5$

Are we happy with this estimator?

Regularized least squares with different realizations of noise ($3\times$)

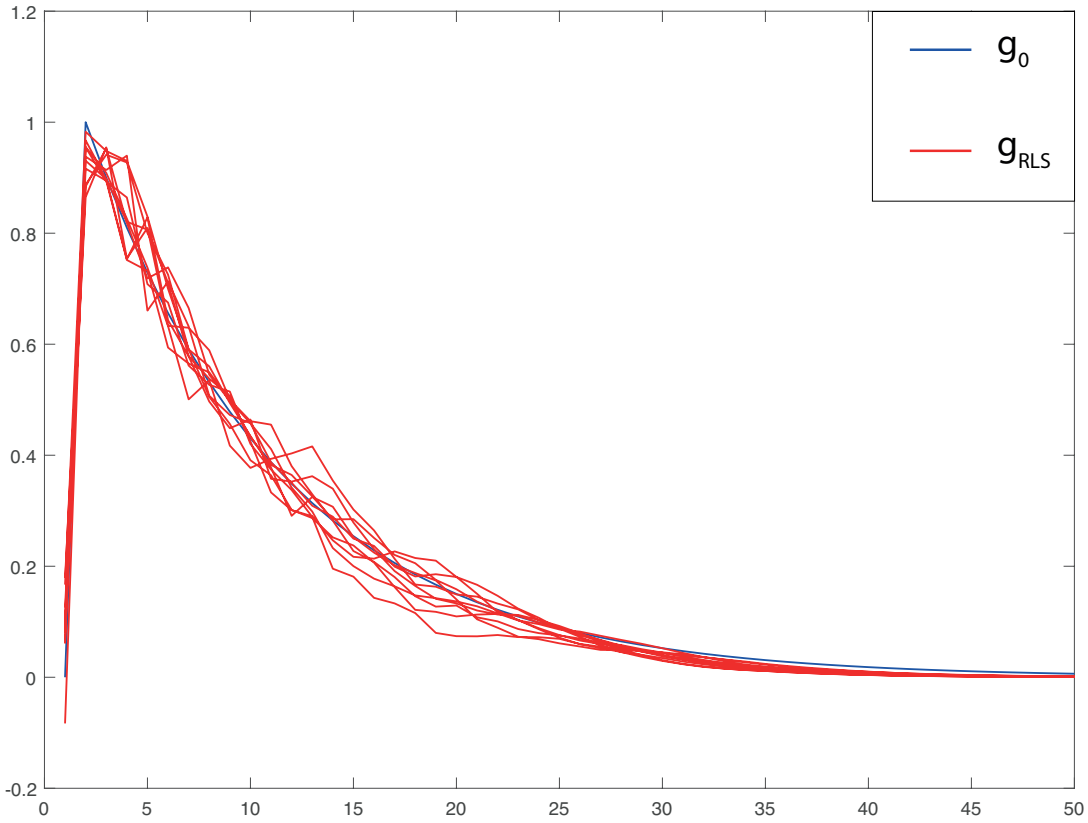


$u =$`randn(N,1)`, $N = 150$, $n = 50$, $\sigma^2 = 0.5$

Are we happy with this estimator?

disc

Regularized least squares with different realizations of noise ($10\times$)

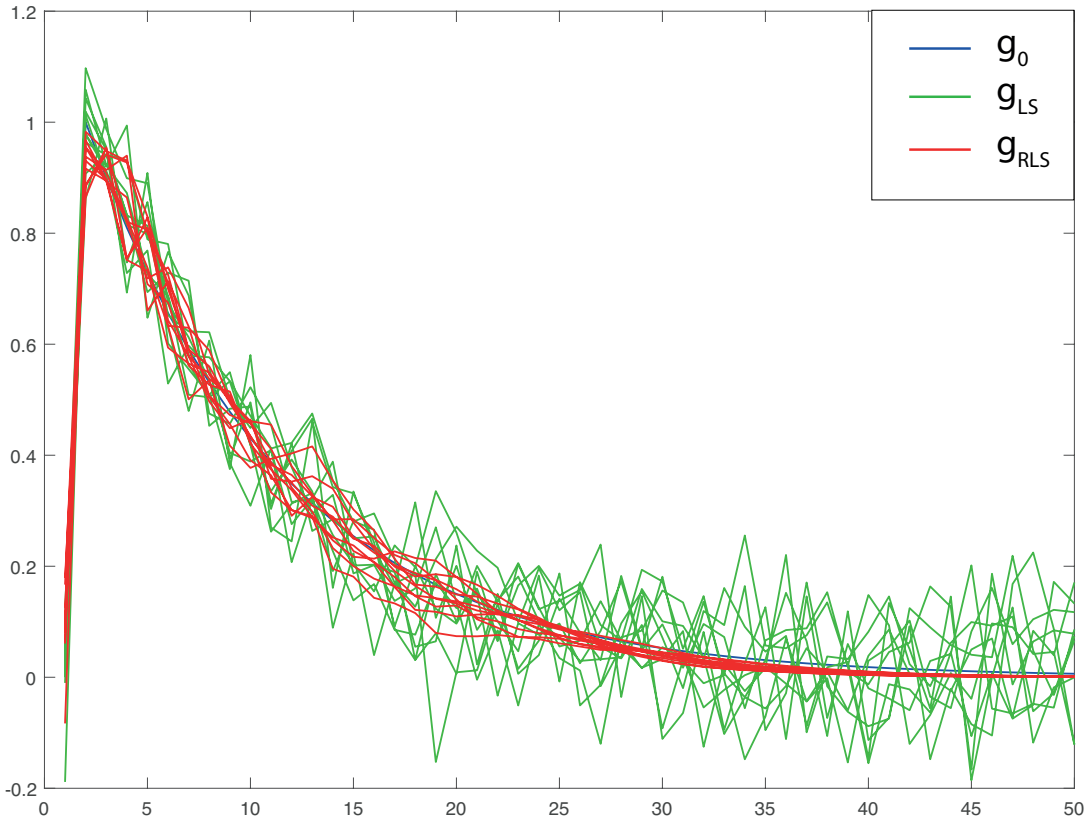


$u =$`randn(N,1)`, $N = 150$, $n = 50$, $\sigma^2 = 0.5$

Comparison

disc

Regularized least squares vs Least Squares (10×)



$u =$`randn(N,1)`, $N = 150$, $n = 50$, $\sigma^2 = 0.5$

What do we observe?

disc

- Regularized least squares seem to be less sensitive to the dataset
- Although we have different realizations of noise, the estimates remain quite close to each other (and also very close to the true impulse response)
- This property is particularly visible in the tail of the impulse response, where we know that the impulse response decays to zero
- Somehow, we have injected into the estimator some a priori information

Properties of this estimator

disc

- It is biased:

$$\mathbf{E}[\hat{g}]-g_0=\mathbf{E}[(U^TU+\gamma K^{-1})^{-1}U^Ty]-g_0=((U^TU+\gamma K^{-1})^{-1}U^TU-I_n)g_0$$

- Its covariance is

$$\mathbf{E}[(\hat{g}-\mathbf{E}[\hat{g}])(\hat{g}-\mathbf{E}[\hat{g}])^T]=\sigma^2(U^TU+\gamma K^{-1})^{-1}U^TU(U^TU+\gamma K^{-1})^{-T}$$

Example

Let $u(t)$ be white noise with variance ρ. Take $K=I_n$. Then the variance is

$$\mathbf{E}[(\hat{g}-\mathbf{E}[\hat{g}])^T(\hat{g}-\mathbf{E}[\hat{g}])]=\sigma^2\mathrm{tr}[(U^TU+\gamma I_n)^{-1}U^TU(U^TU+\gamma I_n)^{-T}]\simeq\frac{\rho\sigma^2Nn}{(\rho N+\gamma)^2}$$

which is smaller than the variance of LS for every $\gamma>0$.

Discussion

disc

- We have found a new estimation criterion that differs from LS/PEM for the presence of a regularization term
- This estimator is biased, meaning that it always commits an error, independently of the quality of the data
- On the other hand, the estimates have a lower variance than LS, i.e., it is less sensitive to the data variation

We may accept to have a systematic error (bias), if this helps us obtaining a better model!

Can we quantify the notion of better model?

The Mean-Square Error (MSE)

disc

Mean-Square Error (MSE) of an estimator

$$\text{MSE} := \mathbf{E}[(\hat{g} - g_0)^T(\hat{g} - g_0)]$$

$$\begin{aligned}
\text{MSE} &= \mathbf{E}[(\hat{g} - g_0)^T(\hat{g} - g_0)] \\
&= \mathbf{E}[(\hat{g} - \mathbf{E}[\hat{g}] + \mathbf{E}[\hat{g}] - g_0)^T(\hat{g} - \mathbf{E}[\hat{g}] + \mathbf{E}[\hat{g}] - g_0)] \\
&= \mathbf{E}[(\hat{g} - \mathbf{E}[\hat{g}])^T(\hat{g} - \mathbf{E}[\hat{g}])] + 2\mathbf{E}[(\hat{g} - \mathbf{E}[\hat{g}])^T\mathbf{E}[(\mathbf{E}[\hat{g}] - g_0)] \\
&\quad + \mathbf{E}[(\mathbf{E}[\hat{g}] - g_0)^T(\mathbf{E}[\hat{g}] - g_0)] \\
&= \mathbf{E}[(\hat{g} - \mathbf{E}[\hat{g}])^T(\hat{g} - \mathbf{E}[\hat{g}])] + (\mathbf{E}[\hat{g}] - g_0)^T(\mathbf{E}[\hat{g}] - g_0) \\
&= \qquad \text{VARIANCE} \qquad + \qquad \text{BIAS}^2
\end{aligned}$$

Note that the decomposition $\text{MSE} = \text{VARIANCE} + \text{BIAS}^2$ holds in general (not only for the estimator under study)

Bias-Variance trade off

disc

The MSE measures the average (squared) error between g_0 and $\hat{g}$

- It is a good measure of the performance of the estimator
- Achieving a low MSE is an interplay between bias and variance

Recall:

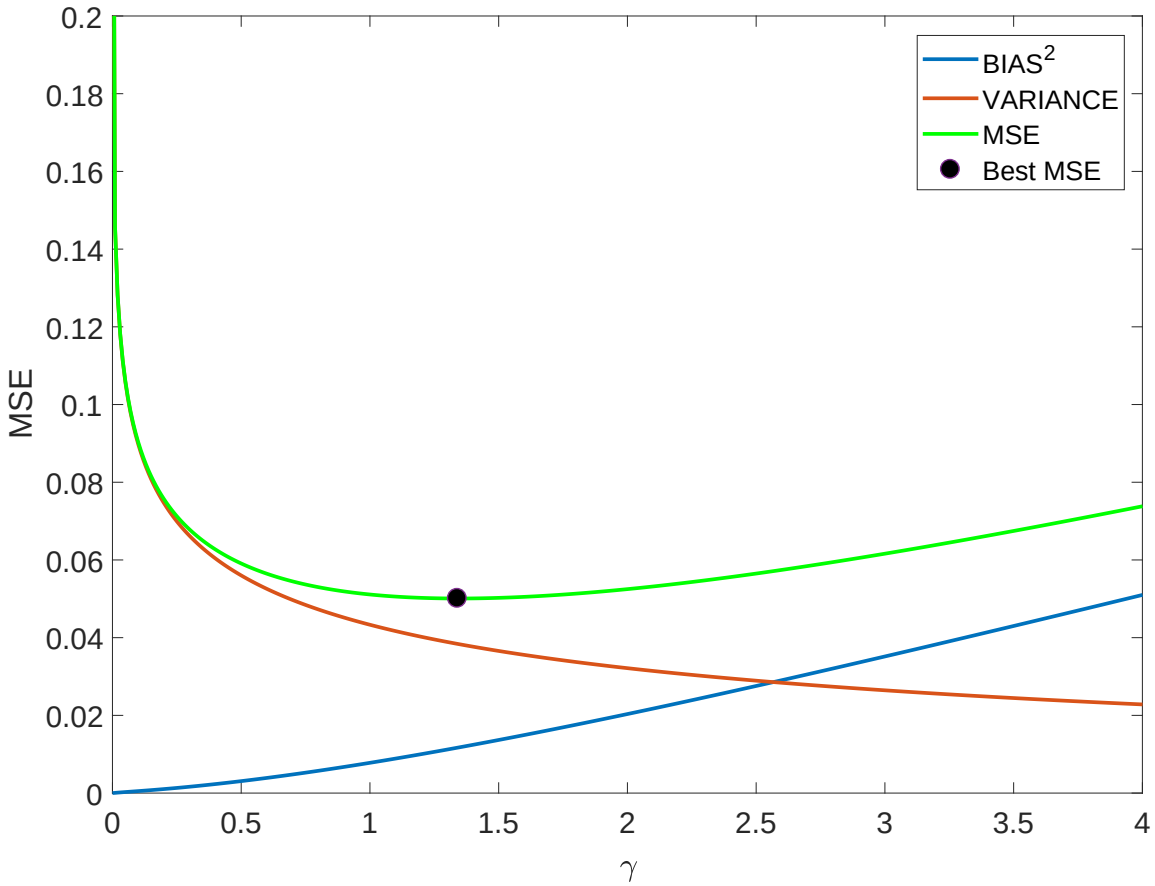
$$J(g) = \underbrace{\|y - Ug\|_2^2}_{\text{VARIANCE}} + \gamma \underbrace{\|g\|_{K^{-1}}^2}_{\text{BIAS}^2}$$

Let K be fixed. Playing with γ, we can trade off bias and variance

- $\gamma = 0 \quad \longrightarrow$ no bias, but high variance (LS)
- $\gamma \uparrow \quad \longrightarrow$ increasing bias, but decreasing variance (RLS)
- $\gamma = \infty \quad \longrightarrow$ only bias, no variance ($\hat{g} = 0$ always)

MSE in our example

MSE as a function of regularization parameter γ



$u =$`randn(N,1)`, $N = 150$, $n = 50$, $\sigma^2 = 0.5$

Another example

disc

We consider a FIR system

$$y = Ug + e,$$

where:

- $g_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
- $N = 20$, $\sigma^2 = 5$
- $u =$`randn(N,1)`

We compare:

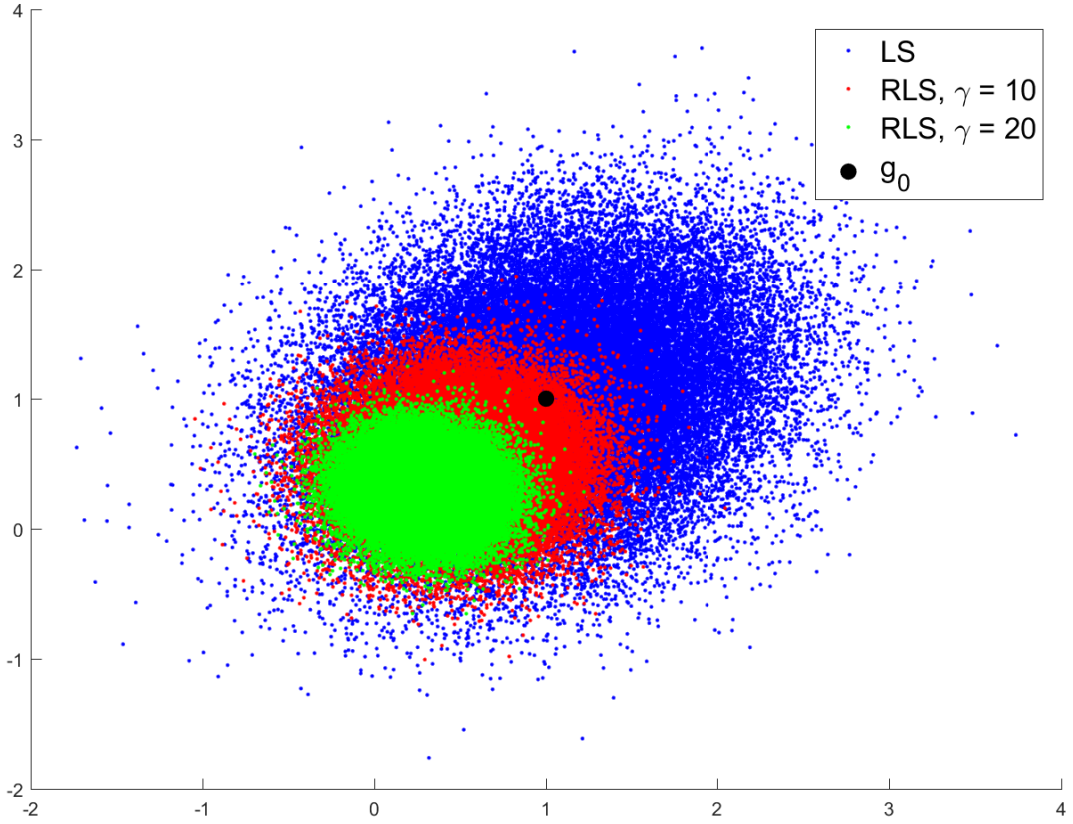
- LS
- RLS with $K = I_2$ and $\gamma = 10$
- RLS with $K = I_2$ and $\gamma = 20$

We repeat the estimation 50k times, each time with a different noise realization.

Another example

disc

Location of the estimates in the plane $(g(1), g(2))$



What do we observe?

Wrap up and take home

disc

- Using LS/PEM in nonparametric methods may lead to unsatisfactory results
- The reason lies in the high variance of LS
- Introducing RLS, we can reduce the variance, at the price of having some bias
- Since MSE $=$ BIAS^2 $+$ VARIANCE, if the variance reduction is high compared to the bias increase, then we can have great improvement in the estimation performance
- The bias/variance trade off depends on the structure of RLS, namely on K (and γ)

How should we choose K?

disc

PART II

Adjusting the bias

disc

Concept of bias (in this framework) $\longleftrightarrow$ Concept of prior knowledge

- We need to inject bias reflecting our belief/knowledge about the unknown quantity, otherwise we do not get a good bias-variance trade off
- Hereby we deal with systems that are:
  - BIBO stable $\longrightarrow$ reasonable assumption in sysid ($\star$)
  - with smooth impulse response $\longrightarrow$ because we deal with sampled versions of continuous time physical systems

Adjusting the bias $\leftrightarrow$ Choosing a suitable K

Another example (cont'd)

disc

- Recall: $g_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
- We try 4 different kernels:

$$K_1 = I \qquad K_2 = \begin{bmatrix} 1 & .1 \\ .1 & 3 \end{bmatrix} \qquad K_3 = \begin{bmatrix} 1 & 0 \\ 0 & .1 \end{bmatrix}$$

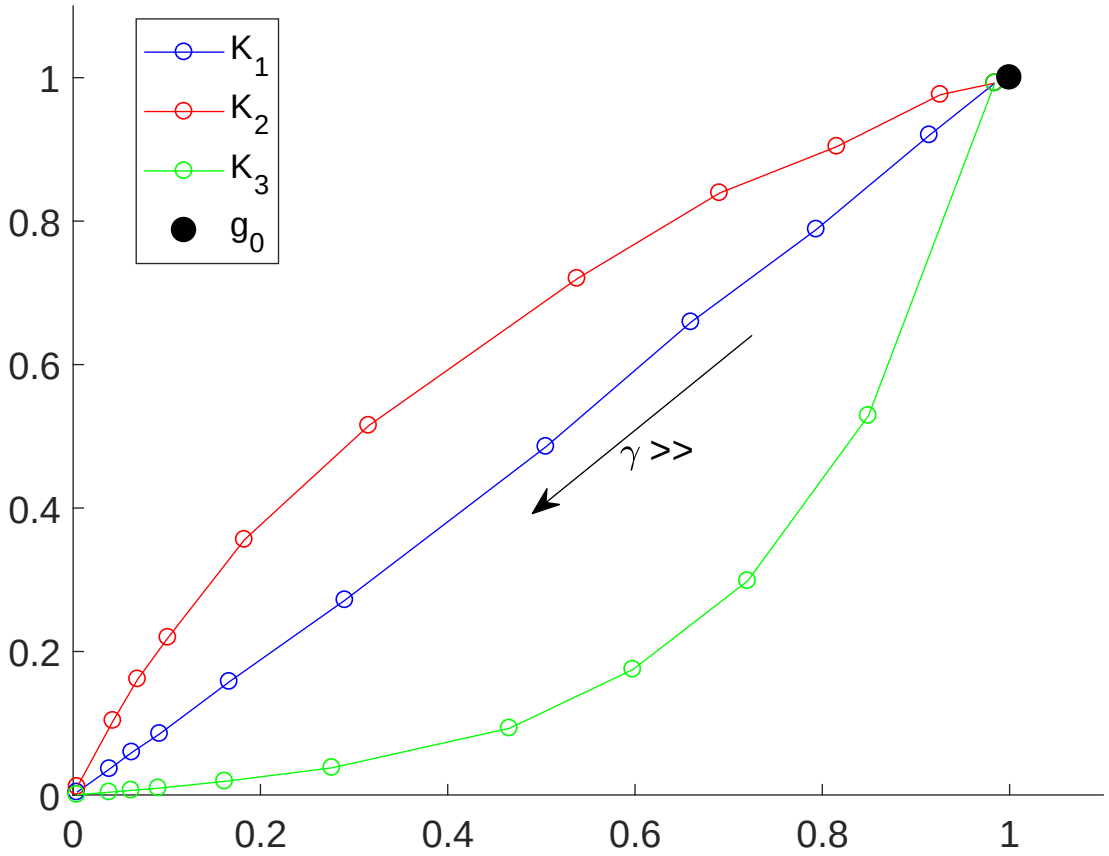
- We test different values of the regularization parameter:

$$\gamma = (0, 2, 5, 10, 20, 50, 100, 200, 300, 500, 5000)$$

- We perform 2000 simulations and average out

Another example (cont'd)

Location of the estimates in the plane $(g(1), g(2))$



K determines how you move towards 0

Towards the design of K

disc

Recall

$$\hat{g} = \arg\min_g \|y - Ug\|_2^2 + \gamma g^T K^{-1} g$$

We want to steer $\hat{g}$ towards certain properties

$\hookrightarrow$ $g^T K^{-1} g$ should penalize "undesired" solutions

$\hookrightarrow$ K should reflect the "desired" behavior

Why is K called a kernel

$$\hat{g} = (U^T U + \gamma K^{-1})^{-1} U^T y = K \underbrace{(U^T UK + \gamma I_n)^{-1} U^T y}_{:=c} = \sum_{i=1}^{n} c_i K(:, i)$$

Hence $\hat{g}$ is a linear combination of the columns of K

Encoding stability

disc

Properties of stable systems

- LTI system is BIBO stable $\Longrightarrow g_0 \in \ell_1$ $\left(\sum_k |g_0(k)| < \infty\right)$
- LTI system of finite (unknown) order $\Longrightarrow g_0$ decays exponentially fast

Therefore, K should have diagonal elements that decay exponentially.

Example:

$$K = \begin{bmatrix} \alpha & 0 & \dots & 0 \\ 0 & \alpha^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \alpha^n \end{bmatrix}$$

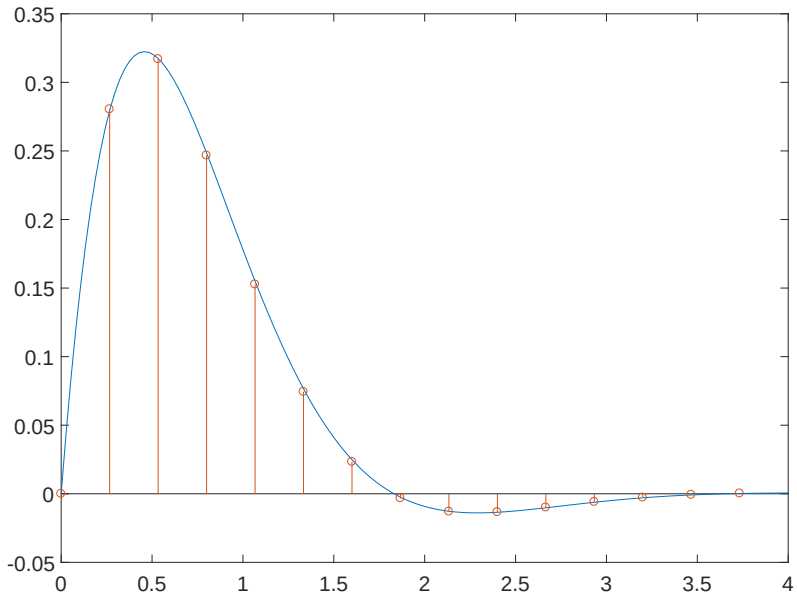
- $0 \leq \alpha < 1$ is an hyperparameter that regulates the decay rate of $\hat{g}$
- We could leave each diagonal element free (n hyperparameters), but then the choice of these elements would be as hard as estimating g_0!

Enconding smoothness

disc

Impulse response samples are likely to decay in a "smooth" way

$\hookrightarrow$ We can expect that $g_0(k+1)$ does not deviate too much from $g_0(k)$



Sampled impulse response of a second-order system

Encoding smoothness

disc

We can exploit the off-diagonal elements of K to encode smoothness

- $K(i,j)$ gives information on the correlation between $g_0(i)$ and $g_0(j)$
- and we know that probably $|g_0(k+1)| < |g_0(k)|$

Good choice: $K(i,j) = \alpha^{\max(i,j)}$

Why is this a good choice?

- It well encodes the above considerations
- K is symmetric positive-definite. This is desirable because
  - $J(g)$ is convex
  - We can interpret RLS in a Bayesian way (later)

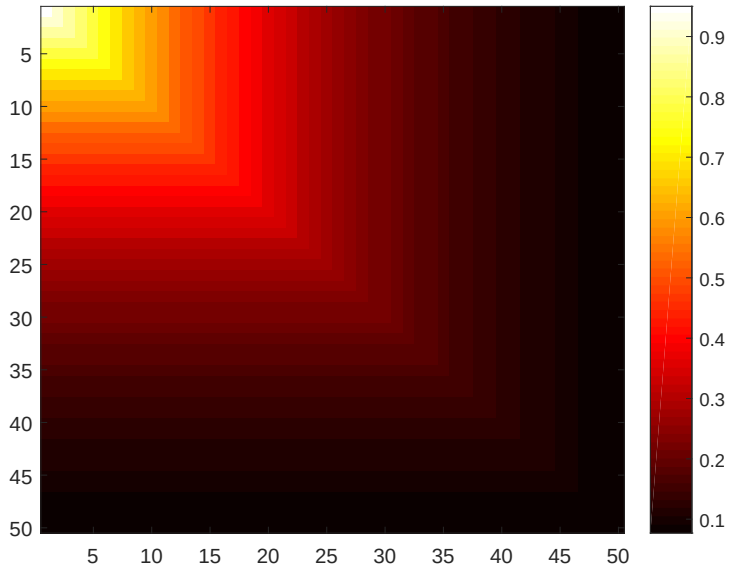
Putting the pieces together

disc

Our choice of kernel is then:

$$K(i,j) = \alpha^{\max(i,j)}$$

a.k.a. first-order stable spline kernel [Pillonetto & De Nicolao, 2010] or TC kernel [Chen et al., 2012]



Heatmap of the TC kernel for $\alpha = 0.95$

Other kernels for system identification

disc

Other kernels have been proposed for impulse response estimation. Examples:

- DC kernel:

$$K(i,j) = \alpha^{(k+j)/2}\rho^{|j-k|}$$

- SS kernel:

$$K(i,j) = \frac{\alpha^{(k+j)+\max(k,j)}}{2} - \frac{\alpha^{3\max(k,j)}}{6}$$

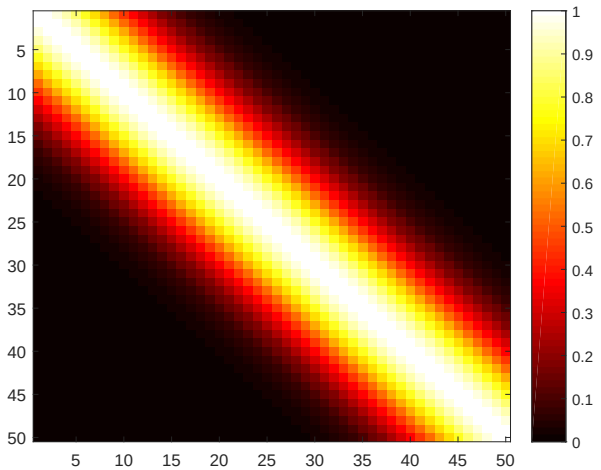
In general, kernel for system identification (in time domain) share the feature to be exponentially decaying along row and column indices.

Kernels not suitable for sysid

disc

Kernel-based RLS is very popular for learning tasks. Common kernels are:

- Gaussian RBF kernel: $K(i,j) = \exp[-(i-j)^2/\eta]$
- Laplacian RBF kernel: $K(i,j) = \exp[-|i-j|/\eta]$
- Ridge regression (kernel): $K = I_n$



Heatmap of the Gaussian kernel

These kernels are called stationary := their argument depends on $|i-j|$

$\hookrightarrow$ They do not capture stability (no decay along diagonal)

When is a kernel suitable for sysid?

disc

Consider the case $n \to \infty$. Then

$$g \text{ is BIBO stable} \Leftrightarrow \sum_k |g(k)| < \infty \ (\hat{g} \in \ell_1)$$

Can we ensure that $\hat{g}$ is BIBO stable?

Theorem (BIBO-stable kernels) [Pillonetto et al., 2014]

If $\sum_i \sum_j |K(i,j)| < \infty$, then $\hat{g} \in \ell_1$.

Simple trick to get BIBO-stable kernels [Chen, 2018]:

1. Take a stationary kernel $K_s(i,j)$;
2. Take $b(k) \in \ell_1$ (e.g. $b(k) = \rho^k, \ 0 < \rho < 1$);
3. Apply $K(i,j) = b(i)b(j)K_s(i,j)$.

Optimal kernel

disc

A natural question now arises: What is the best kernel that we can use?

Theorem [Chen et al., 2012]

Let $\gamma = \sigma^2$. The kernel that minimizes the MSE is $K = g_0 g_0^T$.

- Intuitive explanation: $\hat{g} = K(U^T UK + \sigma^2 I_n)^{-1} U^T y = g_0 \underbrace{g_0^T (U^T U g_0 g_0^T + \sigma^2 I_n)^{-1} U^T y}_{:=\beta} = \beta g_0$
- The solution is aligned to $g_0 \longrightarrow$ we are giving a very specific bias
- Not employable in practice, kernel depends on the true system!

Bayesian interpretation of regularization

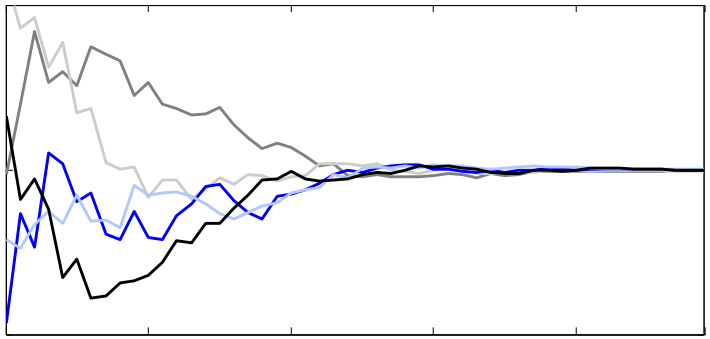
disc

Recall: $y = Ug + e$

- We assume: $e \sim \mathcal{N}(0, \sigma^2 I_N)$ (white Gaussian noise)
- We postulate a random generation mechanism: $g \sim \mathcal{N}(0, \lambda K)$, $\lambda > 0$

What does this mean?

- We "believe" that g has a specific behavior (described through K)
- λ tells how weak is our belief (quantifies uncertainty of the prior)



Realizations from the prior with $K =$ TC kernel

Probabilistic model

disc

We can write a joint Gaussian model for g and y

$$\begin{bmatrix} g \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \lambda K & \Sigma_{gy} \\ \Sigma_{yg} & \Sigma_y \end{bmatrix} \right)$$

where $\Sigma_{gy} = \Sigma_{yg}^T = \lambda K U^T$ and $\Sigma_y = \lambda U K U^T + \sigma^2 I_N$

Bayesian minimum mean-square error (MMSE) estimate

$$\hat{g} = \mathbb{E}\left[g|y\right] = \int_{\mathbb{R}^n} g\, p(g|y)\, dg$$

For Gaussian model it becomes easy: $\mathbb{E}\left[g|y\right] = \Sigma_{gy}\Sigma_y^{-1} y$

Connection with regularized least-squares

disc

$$
\begin{aligned}
\mathbb{E}[g|y] &= \Sigma_{gy}\Sigma_y^{-1}y \\
&= \lambda K U^T \left(\lambda U K U^T + \sigma^2 I_N\right)^{-1} y \\
&= [A(I+BA)^{-1} = (I+AB)^{-1}A] \\
&= \left(U^T U + \frac{\sigma^2}{\lambda} K^{-1}\right)^{-1} U^T y
\end{aligned}
$$

Result

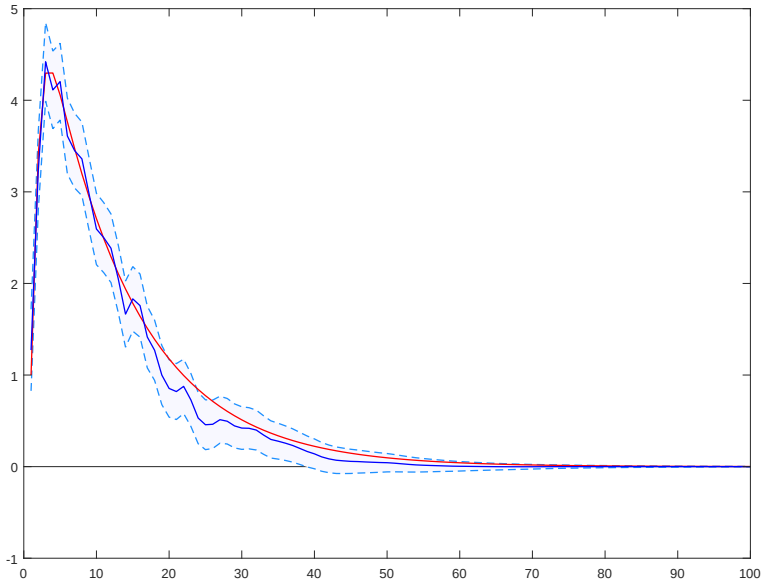
The RLS estimate of g with regularization term $\gamma\|g\|^2_{K^{-1}}$ corresponds to the Bayesian MMSE estimate of g where it is postulated that $g \sim \mathcal{N}(0,\, \lambda K)$, with $\gamma = \sigma^2/\lambda$.

Posterior density of the system

disc

The Bayesian interpretation allows to compute the posterior density of g given the measurements. We have:

$$p(g|y) = \mathcal{N}(\hat{g},\, P) \quad , \quad P = \sigma^2 \left(U^T U + \gamma K^{-1} \right)^{-1}$$



True impulse response (red), its estimate (blue), and 95% confidence intervals ($= \pm 1.96\sqrt{P_{ii}}$)

Maximum-a-posteriori estimate

disc

There is also a correspondence with the MAP estimator of g:

$$\hat{g}_{MAP} = \arg\max_g p(g|y)$$

$$= \arg\max_g p(y|g)\ p(g) \qquad \longrightarrow p(g|y) = \frac{p(y|g)p(g)}{p(y)}$$

Taking the minus-log and discarding constant terms we get the RLS.

Note:

- the MAP estimate is a point estimate $\rightarrow$ no confidence intervals are associated with it
- the equivalence RLS $\equiv$ MMSE $\equiv$ MAP holds only for Gaussian prior + Gaussian noise case

The hyperparameters

disc

- So far, we have determined how the structure of the kernels should be in order to have a good identification performance
- This structure is flexible, because it depends on (one or more) hyperparameters

  $\hookrightarrow$ hyperparameters permit a fine tuning that adapt the kernel to the specific unknown system
- On the other hand, we have to choose these hyperparameters

Hyperparamater tuning can be based on the data

Hyperparameter tuning

disc

Consider the TC kernel $K(i,j) = \alpha^{\max(i,j)}$. Then we have to estimate:

- $\alpha \longrightarrow$ tunes the decay velocity of $\hat{g}$
- $\gamma \longrightarrow$ regulates the strength of the kernel effect (trade off bias/variance)

Popular methods for hyperparameter tuning

- Cross validation $\longrightarrow$ popular in machine learning
- Marginal likelihood $\longrightarrow$ popular in system identification

Cross validation

disc

- First, note that $\hat{g} = (U^T U + \gamma K^{-1})^{-1} U^T y = \hat{g}(\alpha, \gamma)$
- Idea: check for which (α, γ) the estimated $\hat{g}(\alpha, \gamma)$ best tracks some new data

Basic CV

1. Split the data set in a training set and a validation set (usually 70%/30%)
2. Compute $\hat{g}(\alpha, \gamma)$ on the training set
3. set

$$(\hat{\alpha}, \hat{\gamma}) = \arg\min_{\alpha,\gamma} \|y_{\text{val}} - \hat{y}_{\text{val}}\|^2$$

where $\hat{y}_{\text{val}} = U_{\text{val}} \hat{g}(\alpha, \gamma)$

Cross validation

disc

In some cases $\hat{g}(\alpha, \gamma)$ cannot be computed in closed form

Basic CV - sampled hyperparameters

1. Define a grid of candidate hyperparameters $(\alpha_i, \gamma_i) \in \Gamma$
2. Split the data set in a training set and a validation set (usually 70%/30%)
3. For each $(\alpha_i, \gamma_i) \in \Gamma$ compute $\hat{g}(\alpha_i, \gamma_i)$ on the training set
4. Evaluate $S(\alpha_i, \gamma_i) = \|y_{\text{val}} - U_{\text{val}}\hat{g}(\alpha_i, \gamma_i)\|^2$
5. Choose (α_i, γ_i) minimizing S

The marginal likelihood

disc

Marginal likelihood: Evidence of the parameters on data based on the postulated Bayesian model. Introduce the output marginal density

$$p(y;\alpha,\lambda) = \int_{\mathbb{R}^n} p(y,g;\alpha,\lambda)dg$$

Marginal likelihood criterion

$$(\hat{\alpha},\hat{\lambda}) = \arg\max_{(\alpha,\lambda)} p(y;\alpha,\lambda)$$

The marginal likelihood

disc

In the Gaussian case, the marginal distribution of y is a Gaussian random vector with zero mean and covariance

$$\begin{aligned}\mathbf{E}[yy^T] &= \mathbf{E}[(Ug+e)(Ug+e)^T] \\ &= \mathbf{E}[(Ug)(Ug)^T] + \mathbf{E}[(Ug)e^T] + \mathbf{E}[ee^T] \\ &= U\lambda K U^T + \sigma^2 I_N := \Sigma_y\end{aligned}$$

So

$$p(y;\alpha,\lambda) = \frac{1}{\sqrt{(2\pi)^N \det[\Sigma_y]}} \exp\left(-\frac{1}{2} y^T \Sigma_y^{-1} y\right)$$

Final form after taking minus-log and discarding constants

$$(\hat{\alpha},\hat{\lambda}) = \arg\min_{(\alpha,\lambda)} \log\left(\det[U\lambda K U^T + \sigma^2 I_N]\right) + y^T (U\lambda K U^T + \sigma^2 I_N)^{-1} y$$

CV or ML?

disc

CV

+ Very flexible: works for any type of noise/regularization
- May require:
  - large data sets with i.i.d. samples
  - more computationally expensive CV schemes (leave-one-out,...)

ML

+ Very accurate under Gaussian/quadratic assumptions
- May be not robust if these assumptions are violated (e.g., outliers...)

Procedure recap:

1. Tune hyperparameters via ML or CV
2. Compute $\hat{g} = (U^T U + \gamma K^{-1})^{-1} U^T y$

Some simulations

disc

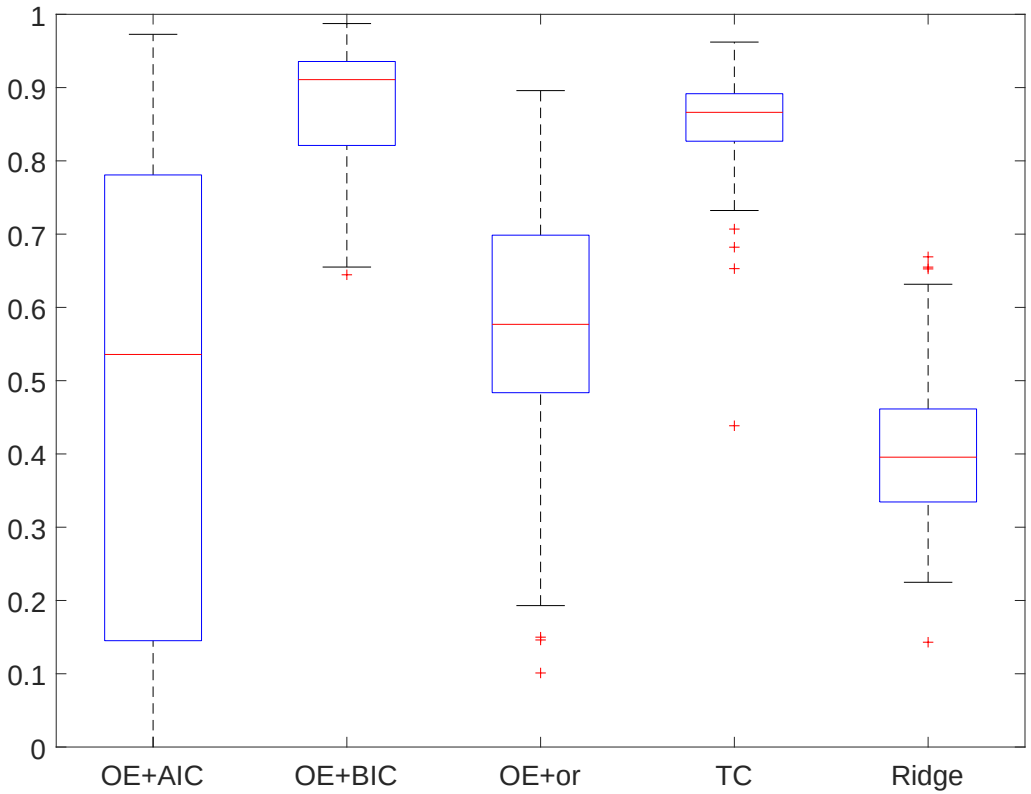
We want to understand when nonparametric kernel-based methods may be an attractive choice to identify LTI systems. We use simulation benchmarks with the following features:

- Randomly generated continuous-time systems (`rss` in Matlab)
- Sampled at $3\times$ bandwidth, SNR $= 20$
- $u(t)$ is filtered with noise (through a low-pass filter)
- Fit $= 1 - \|g_0 - \hat{g}\|_2 / \|g_0\|_2$

Tested methods:

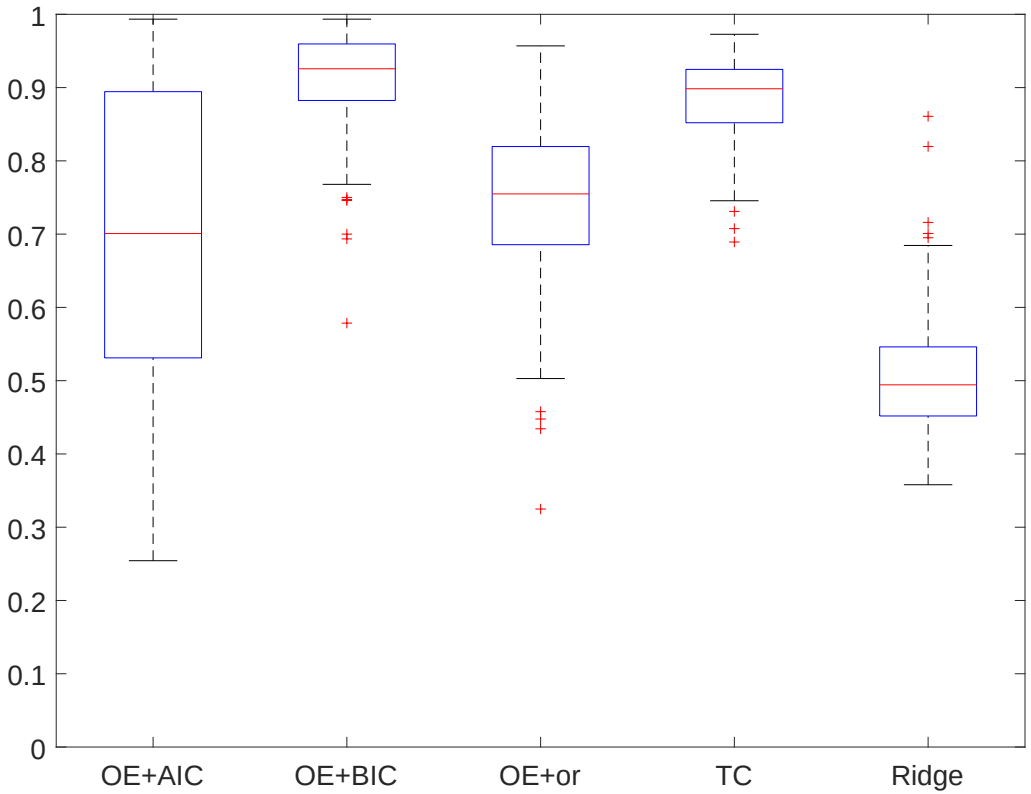
- OE+AIC
- OE+BIC
- OE+true model order
- regularization with TC kernel (and marginal likelihood), $n = 100$
- regularization with ridge regression kernel (and ML), $n = 100$

Simulation 1



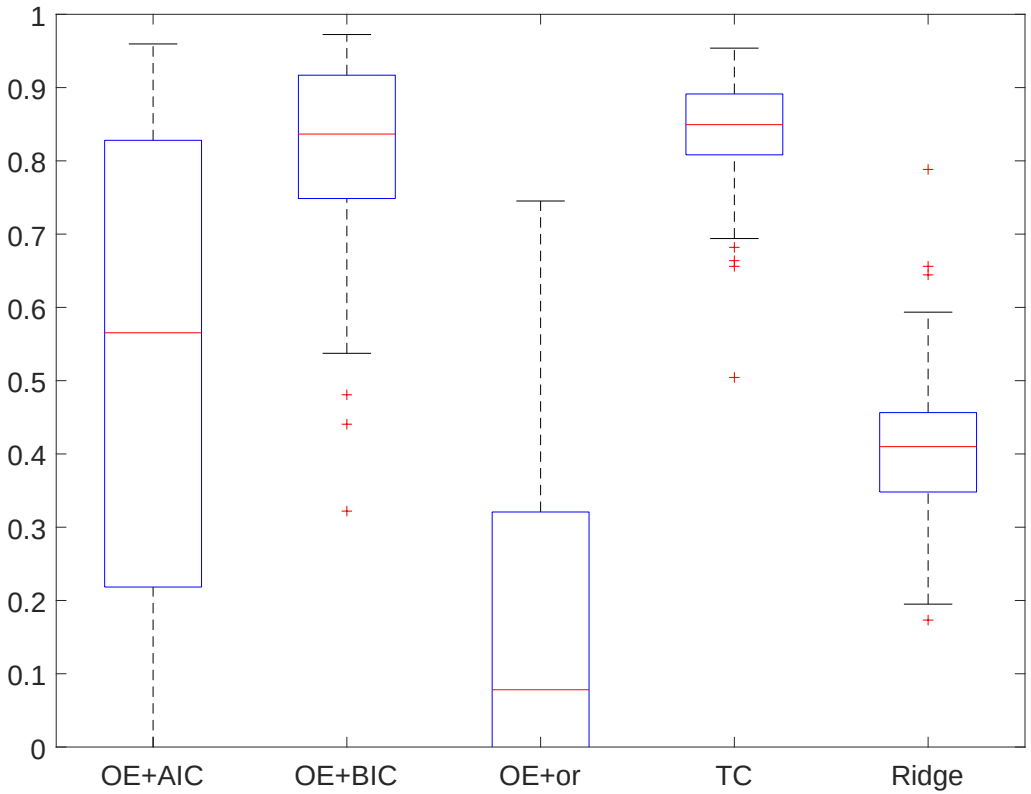
Results for system order 6 and $N = 200$

Simulation 2



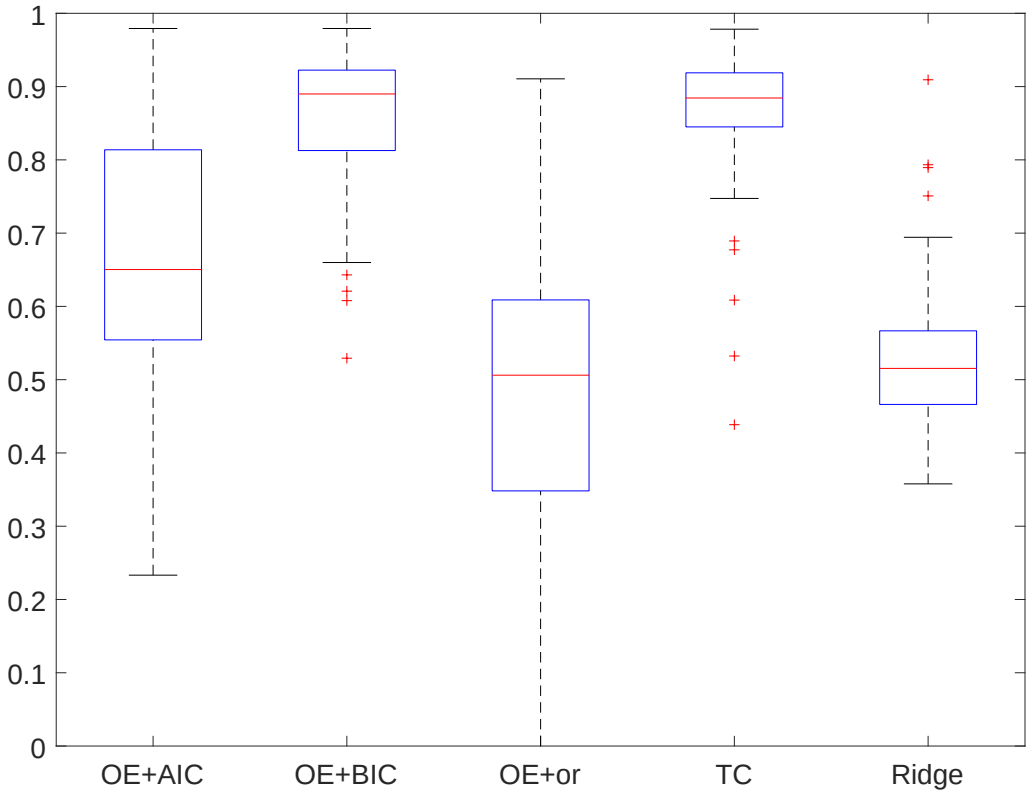
Results for system order 6 and $N = 500$

Simulation 3



Results for system order 20 and $N = 200$

Simulation 4



Results for system order 20 and $N = 500$

Comments

disc

When regularized nonparametric methods are a valid altenative?

- No clue on the physics of the system
  $\hookrightarrow$ Hard to determine model order
- High-order system
  $\hookrightarrow$ Many parameters to estimate
- Short data sets
  $\hookrightarrow$ ML/PEM suffer from high variance
- In Matlab, you can use `impulseest` to apply regularization.
- Extension to colored noise can be achieved using the predictor form of the system [Pillonetto et al., 2011]

References

1. Pillonetto, G., Dinuzzo, F., Chen, T., De Nicolao, G., and Ljung, L. (2014). Kernel methods in system identification, machine learning and function estimation: A survey. Automatica, 50(3), 657-682.
2. Chen, T., Ohlsson, H., and Ljung, L. (2012). On the estimation of transfer functions, regularizations and Gaussian processes-Revisited. Automatica, 48(8), 1525-1535.
3. Chen, T. (2018). On kernel design for regularized LTI system identification. Automatica, 90, 109-122.
4. Friedman, J., Hastie, T., and Tibshirani, R. (2001). The elements of statistical learning (Vol. 1). Springer, Berlin: Springer series in statistics [Ch. 3, Ch. 7].
5. Rasmussen, C.E., and Williams, C. (2006). Gaussian Processes for Machine Learning, the MIT Press.
6. Pillonetto, G., Chiuso, A., and De Nicolao, G.. (2011). Prediction error identification of linear systems: a nonparametric Gaussian regression approach. Automatica, 47(2), 291-305.