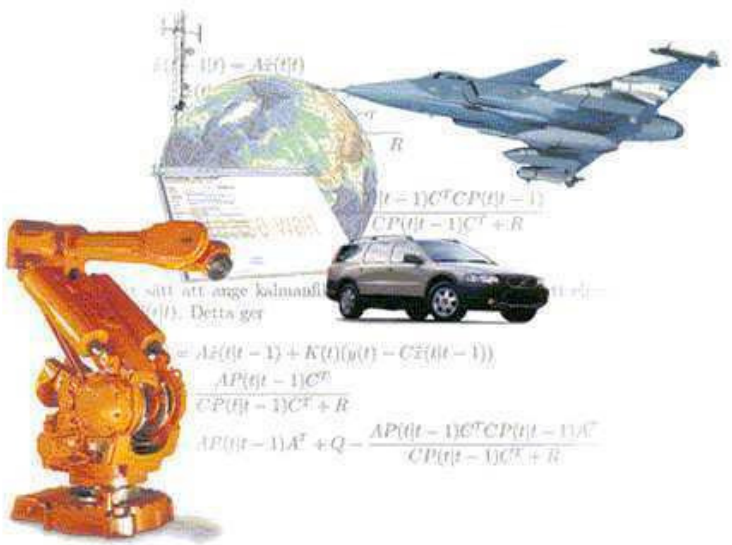
Influences from Machine Learning in Today's System Identification.

Influences from Machine Learning in Today's System Identification.

or

Will Machine Learning Change the Paradigm of System Identification?



Lennart Ljung

Reglerteknik, ISY, Linköpings Universitet

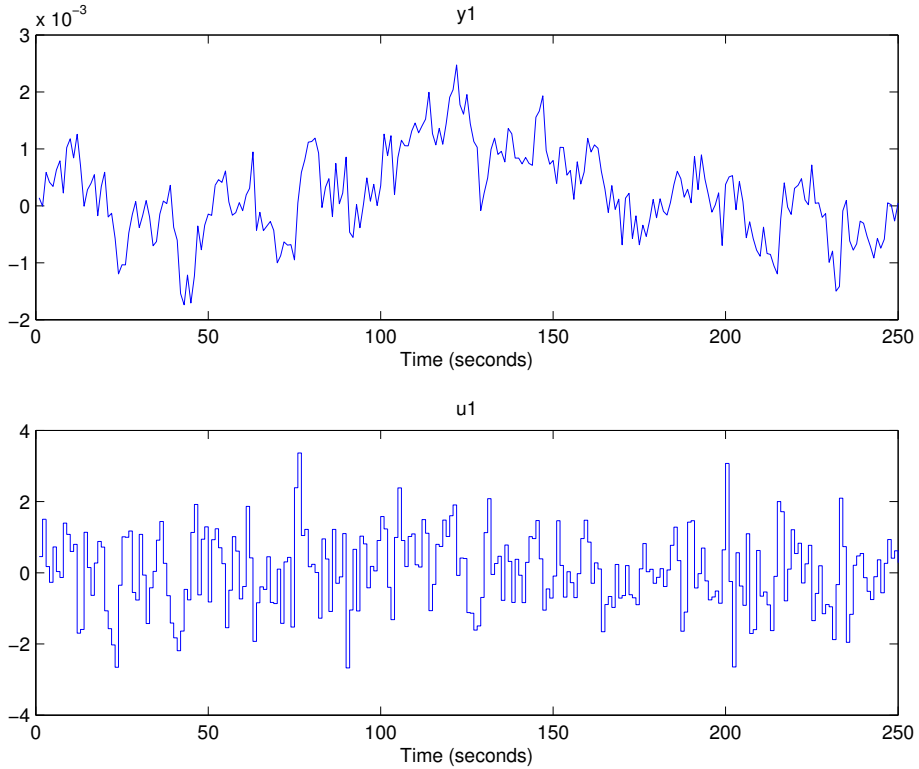
Outline

Will Machine Learning Change the Paradigm of System Identification?

- The paradigm of System Identification
- A function learning approach to estimating impulse response functions
- Using System Identification thinking in more thoughtful way

A Concrete Question

Given the input-Output Data below (250 time points), find the Impulse Response Function (IRF) of the linear system that generated the data! `plot(z)`



A Typical Problem

Given Observed Input-Output Data: Find a Description of the System that Generated the Data [Simulator or Predictor. Linear System: Impulse response or Bode plot].

A Typical Problem

Given Observed Input-Output Data: Find a Description of the System that Generated the Data [Simulator or Predictor. Linear System: Impulse response or Bode plot].

Basic Approach

Find a suitable Model Structure, Estimate its parameters, and compute the response of the resulting model

The System Identification Paradigm

A Typical Problem

Given Observed Input-Output Data: Find a Description of the System that Generated the Data [Simulator or Predictor. Linear System: Impulse response or Bode plot].

Basic Approach

Find a suitable Model Structure, Estimate its parameters, and compute the response of the resulting model

Techniques

Estimate the parameters by ML techniques/PEM (prediction error methods). Find the model structure by AIC, BIC or Cross Validation

The System Identification Paradigm

A Typical Problem

Given Observed Input-Output Data: Find a Description of the System that Generated the Data [Simulator or Predictor. Linear System: Impulse response or Bode plot].

Basic Approach

Find a suitable Model Structure, Estimate its parameters, and compute the response of the resulting model

Techniques

Estimate the parameters by ML techniques/PEM (prediction error methods). Find the model structure by AIC, BIC or Cross Validation

Cross Validation: Estimate a model using part of the data. Evaluate how well that model can reproduce other parts of the data.

Try the Paradigm!

- Let us try Model Structures that are linear state-space models of orders, 1,2, ..., 30
- Estimate them by ML (Maximum Likelihood)
- Select the one that has the best cross validation (CV)

Try the Paradigm!

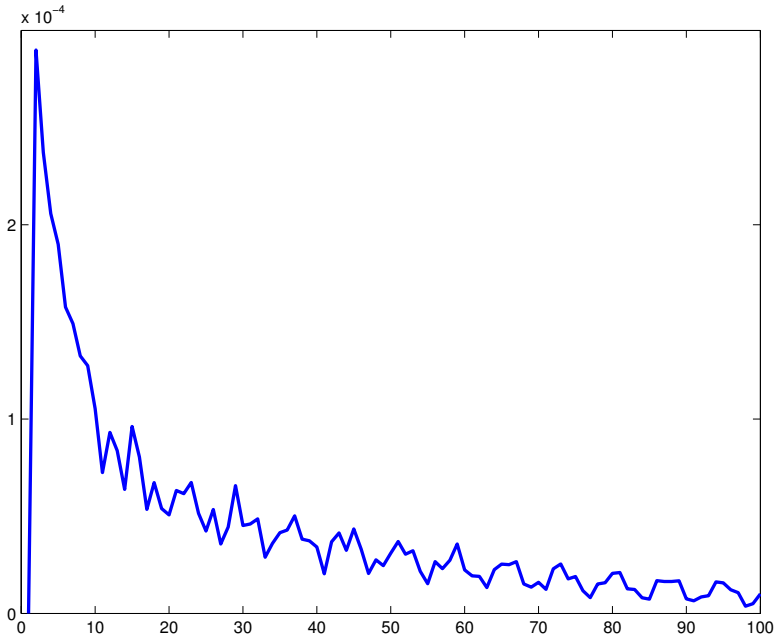
- Let us try Model Structures that are linear state-space models of orders, 1,2, ..., 30
- Estimate them by ML (Maximum Likelihood)
- Select the one that has the best cross validation (CV)

```
for k=1:30
    m{k}= ssest(z(1:125),k);
     (~,fit(k))=compare(z(126:end),m{k});
end
(~,n) = max(fit);%n=9
mss = ssest(z,n);
impulse(mss)
```


Try the Paradigm!

- Let us try Model Structures that are linear state-space models of orders, 1,2, ..., 30
- Estimate them by ML (Maximum Likelihood)
- Select the one that has the best cross validation (CV)

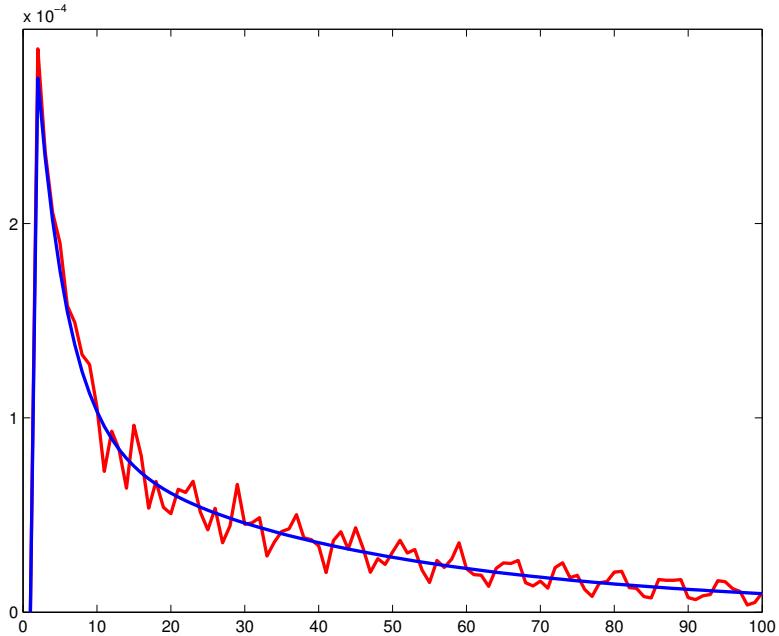
```
for k=1:30
    m{k}= ssest(z(1:125),k);
     (~,fit(k))=compare(z(126:end),m{k});
end
(~,n) = max(fit);%n=9
mss = ssest(z,n);
impulse(mss)
```



How Well Does it Do?

Blue curve: The true impulse response.

Red curve: The selected 9th order SS-model by CV



G : fit: **mss: 83.94%**

Following Pillonetti (et al), Automatica, IEEE AC 2010–2011.
Linear system identification is really about estimating the IRF $g(t)$.
Let us apply function learning to ”learn” the IRF:

Following Pillonetti (et al), Automatica, IEEE AC 2010–2011.
Linear system identification is really about estimating the IRF $g(t)$.
Let us apply function learning to ”learn” the IRF: We measure

$$y(t_k) = \int_0^\infty g(\tau)u(t_k - \tau)d\tau + e(t_k) = L_k^u[g] + e(t_k), k = 1, 2, \ldots, n$$

i.e. functionals of the unknown function in (Gaussian) noise.

Following Pillonetti (et al), Automatica, IEEE AC 2010–2011.
Linear system identification is really about estimating the IRF $g(t)$.
Let us apply function learning to "learn" the IRF: We measure

$$y(t_k) = \int_0^\infty g(\tau)u(t_k - \tau)d\tau + e(t_k) = L_k^u[g] + e(t_k), k = 1, 2, \ldots, n$$

i.e. functionals of the unknown function in (Gaussian) noise.
If we assume the unknown function is a Gaussian Process, we can either employ Gaussian Processes Regression á la Carl Rasmussen or the RKHS (Reproducing Kernel Hilbert Space) function estimation theory of Grace Wahba.

In either case we are lead to a minimization problem

$$\arg\min_{g} \sum_{t=1}^{n} (y(t_k) - L_k^u[g])^2 + \gamma \|g\|_{\mathcal{H}}^2$$

where $\mathcal{H}$ is the Hilbert space reproducing the kernel in the space where g lives.

In either case we are lead to a minimization problem

$$\arg\min_g \sum_{t=1}^{n} (y(t_k) - L_k^u[g])^2 + \gamma \|g\|_{\mathcal{H}}^2$$

where $\mathcal{H}$ is the Hilbert space reproducing the kernel in the space where g lives. An often used kernel in function estimation is the so called cubic smoothing spline kernel corresponding to an assumption that g is integrated Wiener process.

In either case we are lead to a minimization problem

$$\arg\min_{g} \sum_{t=1}^{n} (y(t_k) - L_k^u[g])^2 + \gamma \|g\|_{\mathcal{H}}^2$$

where $\mathcal{H}$ is the Hilbert space reproducing the kernel in the space where g lives. An often used kernel in function estimation is the so called cubic smoothing spline kernel corresponding to an assumption that g is integrated Wiener process.

Pillonetto advocated a modification of this function space, by subjecting the kernel to an exponential transformation, leading to a Stable Splines Kernel.

Function Learning

In either case we are lead to a minimization problem

$$\arg\min_{g} \sum_{t=1}^{n} (y(t_k) - L_k^u[g])^2 + \gamma \|g\|_{\mathcal{H}}^2$$

where $\mathcal{H}$ is the Hilbert space reproducing the kernel in the space where g lives. An often used kernel in function estimation is the so called cubic smoothing spline kernel corresponding to an assumption that g is integrated Wiener process.

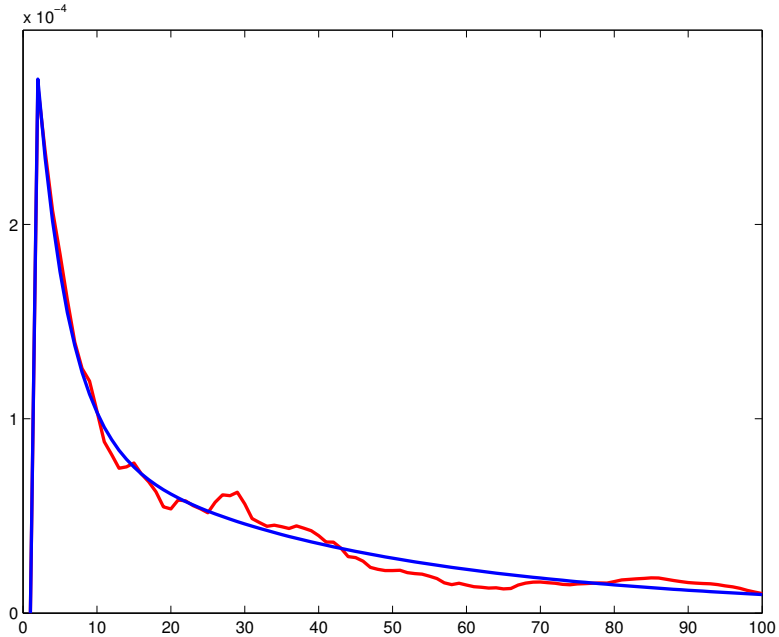
Pillonetto advocated a modification of this function space, by subjecting the kernel to an exponential transformation, leading to a Stable Splines Kernel.

[Technical details will be explained later via reverse engineering.]

How Does it Perform?

Blue curve: The true IRF. Red curves: Model IRFs

Function Learned IRF

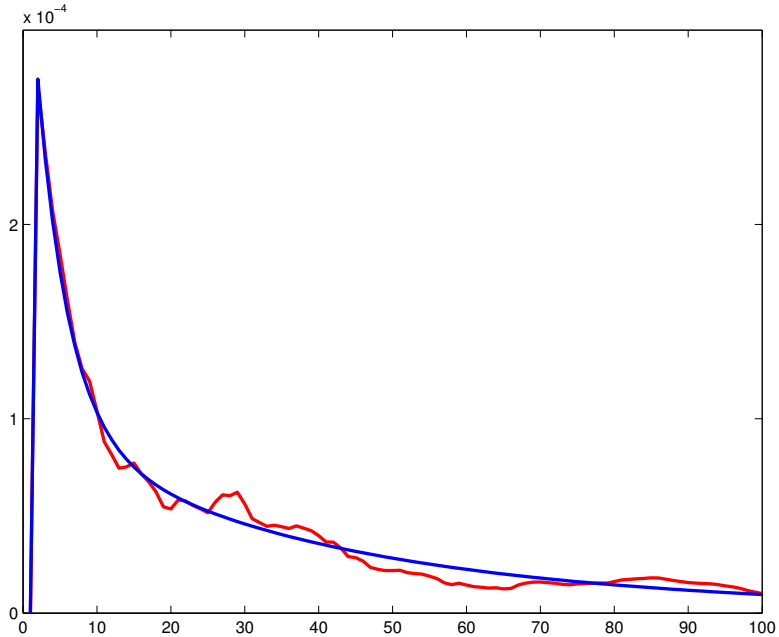


Fit: **mtc: 88.54%**

How Does it Perform?

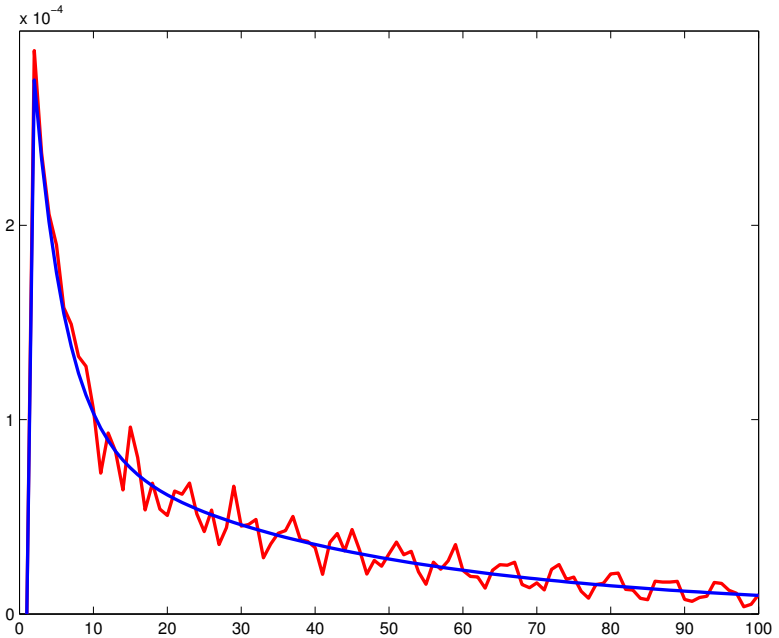
Blue curve: The true IRF. Red curves: Model IRFs

Function Learned IRF



Fit: **mtc: 88.54%**

The SI model 9th order SS



Fit: **mss: 83.94%**

Thought Provoking!

The Machine Learnt model is clearly better than the conventional SS model from the SI paradigm!

Thought Provoking!

The Machine Learnt model is clearly better than the conventional SS model from the SI paradigm!

Can we revisit the System Identification Paradigms and think deeper? Fit to beat: **88.4%**!!

Thought Provoking!

The Machine Learnt model is clearly better than the conventional SS model from the SI paradigm!

Can we revisit the System Identification Paradigms and think deeper? Fit to beat: **88.4%**!!

At the heart of estimation is the Bias-Variance-tradeoff: Choose the model flexibility large enough to allow small bias but not so large that it gives high variance to the parameters.

Thought Provoking!

The Machine Learnt model is clearly better than the conventional SS model from the SI paradigm!

Can we revisit the System Identification Paradigms and think deeper? Fit to beat: **88.4%**!!

At the heart of estimation is the Bias-Variance-tradeoff: Choose the model flexibility large enough to allow small bias but not so large that it gives high variance to the parameters. In System Identification typically is dealt with by the model order selection.

Thought Provoking!

The Machine Learnt model is clearly better than the conventional SS model from the SI paradigm!

Can we revisit the System Identification Paradigms and think deeper? Fit to beat: **88.4%**!!

At the heart of estimation is the Bias-Variance-tradeoff: Choose the model flexibility large enough to allow small bias but not so large that it gives high variance to the parameters. In System Identification typically is dealt with by the model order selection.

In the current situation with short data set and rather complex dynamics it may be even more crucial to deal with this trade-off in a thoughtful and flexible way,

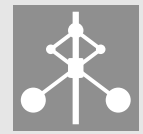
Thought Provoking!

The Machine Learnt model is clearly better than the conventional SS model from the SI paradigm!

Can we revisit the System Identification Paradigms and think deeper? Fit to beat: **88.4%**!!

At the heart of estimation is the Bias-Variance-tradeoff: Choose the model flexibility large enough to allow small bias but not so large that it gives high variance to the parameters. In System Identification typically is dealt with by the model order selection.

In the current situation with short data set and rather complex dynamics it may be even more crucial to deal with this trade-off in a thoughtful and flexible way, since some bias will be beneficial, and then we do not enjoy the optimal variance properties of the unbiased ML-estimates.



$$y(t) = b_1 u(t-1) + b_2 u(t-2) + \ldots + b_{nb} u(t-nb) + e(t);$$
$$t = nb+1, \ldots, N$$

This is a direct discrete-time counterpart of estimating the impulse response function in continuous time.

Use FIR (Finite Impulse Response) model structures

$$y(t) = b_1 u(t-1) + b_2 u(t-2) + \ldots + b_{nb} u(t-nb) + e(t);$$
$$t = nb+1, \ldots, N$$

This is a direct discrete-time counterpart of estimating the impulse response function in continuous time. Write in compact form as

$$Y = \Phi\theta + E \qquad \theta \sim b, \quad \Phi \sim u$$

Use FIR (Finite Impulse Response) model structures

$$y(t) = b_1 u(t-1) + b_2 u(t-2) + \ldots + b_{nb} u(t-nb) + e(t);$$
$$t = nb+1, \ldots, N$$

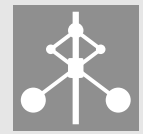
This is a direct discrete-time counterpart of estimating the impulse response function in continuous time. Write in compact form as

$$Y = \Phi\theta + E \qquad \theta \sim b, \quad \Phi \sim u$$

$$Y = [y(nb+1), \ldots, y(N)]^T; \quad \theta = [b_1, \ldots b_{nb}]^T$$

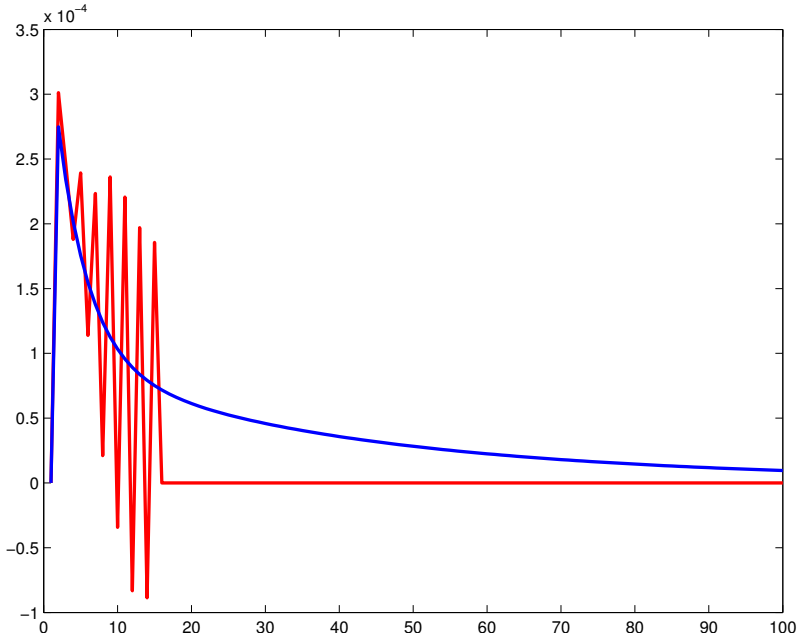
$$\Phi = \begin{bmatrix} u(nb) & u(nb-1) & \ldots & u(1) \\ u(nb+1) & u(nb) & \ldots & u(2) \\ \vdots & \vdots & \ddots & \vdots \\ u(N-1) & u(N-2) & \ldots & u(N-nb) \end{bmatrix}$$

FIR order $nb = 14$ is chosen



FIR Models: Select Order nb Cross-Validation

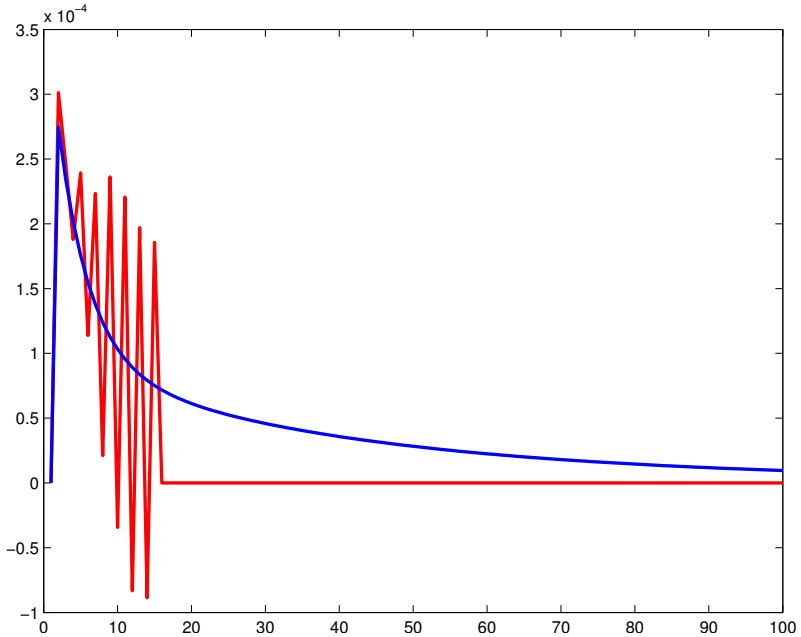
FIR order $nb = 14$ is chosen



fit: **mfir -6%**

FIR Models: Select Order nb Cross-Validation

FIR order $nb = 14$ is chosen



fit: **mfir -6%**

Bias-Variance Trade-off by FIR-order!

Regularization: Ridge Regression

Can we restrain the model flexibility by other means than choosing low order?

Regularization: Ridge Regression

Can we restrain the model flexibility by other means than choosing low order?

Regularization: Add a penalty on the norm of the θ-vector, rather than on its size!

$$\arg\min_{\theta} \|Y - \Phi\theta\|^2 + \lambda\|\theta\|^2$$

Regularization: Ridge Regression

Can we restrain the model flexibility by other means than choosing low order?

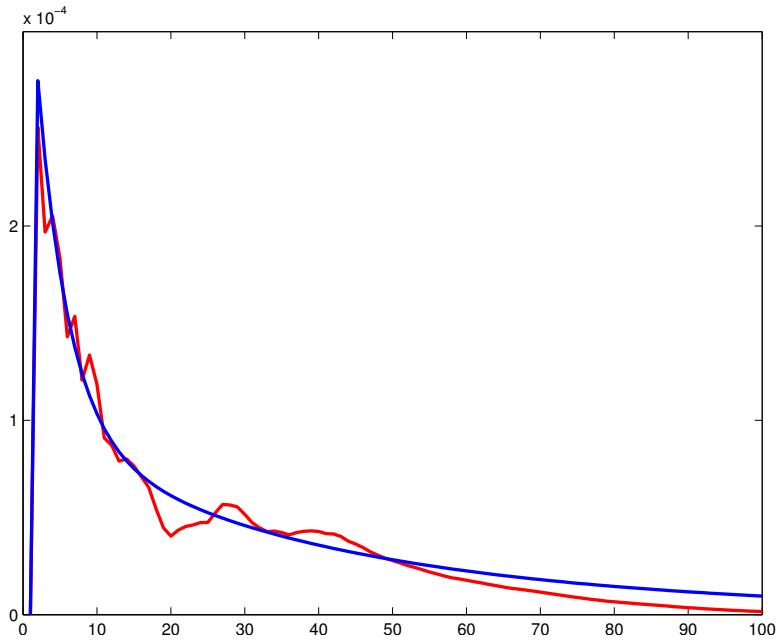
Regularization: Add a penalty on the norm of the θ-vector, rather than on its size!

$$\arg\min_{\theta} \|Y - \Phi\theta\|^2 + \lambda\|\theta\|^2$$

The bias-variance tradeoff (model flexibility) is now controlled by the continuous regularization parameter λ. Clearly a more versatile tool! ("continuous order")

Use Regularization for a FIR model or order 100

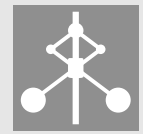
Regularization parameter λ chosen to $\lambda = 27$ by cross-validation



fit: **mfir 81.53%**

More Versatile Regularization Norm

$\arg\min_\theta \|Y - \Phi\theta\|^2 + \lambda\theta^T R\theta;$ R a full $nb|nb$ psd matrix



More Versatile Regularization Norm

$\arg\min_\theta \|Y - \Phi\theta\|^2 + \lambda\theta^T R\theta$; R a full $nb|nb$ psd matrix

How to select the regularization matrix R?

More Versatile Regularization Norm

$\arg\min_\theta \|Y - \Phi\theta\|^2 + \lambda\theta^T R\theta$; R a full $nb|nb$ psd matrix

How to select the regularization matrix R?

The extra term can be interpreted as prior information that $\theta \in N(0, P)$ $P = R^{-1}/\lambda$ (Gaussian with zero mean and covariance P.)

More Versatile Regularization Norm

$\arg\min_\theta \|Y - \Phi\theta\|^2 + \lambda\theta^T R\theta$; R a full $nb|nb$ psd matrix

How to select the regularization matrix R?

The extra term can be interpreted as prior information that $\theta \in N(0, P)$ $P = R^{-1}/\lambda$ (Gaussian with zero mean and covariance P.) We are now maximizing the posterior probability.

More Versatile Regularization Norm

$\arg\min_\theta \|Y - \Phi\theta\|^2 + \lambda\theta^T R\theta$; R a full $nb|nb$ psd matrix

How to select the regularization matrix R?

The extra term can be interpreted as prior information that $\theta \in N(0, P)$ $P = R^{-1}/\lambda$ (Gaussian with zero mean and covariance P.) We are now maximizing the posterior probability.

If we think of P as the covariance matrix of the IRF θ,

$$P = E\theta\theta^T$$

More Versatile Regularization Norm

$\arg\min_\theta \|Y - \Phi\theta\|^2 + \lambda\theta^T R\theta$; R a full $nb|nb$ psd matrix

How to select the regularization matrix R?

The extra term can be interpreted as prior information that $\theta \in N(0, P)$ $P = R^{-1}/\lambda$ (Gaussian with zero mean and covariance P.) We are now maximizing the posterior probability.

If we think of P as the covariance matrix of the IRF θ,

$$P = E\theta\theta^T$$

and we assume that the system is exponentially stable and has a smooth IRF, then

$$P_{jk} = C\mu^{k+j} \cdot \rho^{|k-j|}$$

μ determines the exponential decay and ρ the smoothness of the IRF. $P = P(\alpha)$, $\alpha = [C, \mu, \rho]$, the "hyperparameters".

Tuning the Regularization Matrix

Suppose

$$Y = \Phi\theta + E, \qquad \text{and } \theta \in N(0, P(\alpha)), \quad E \in N(0, \sigma^2 I)$$

Tuning the Regularization Matrix

Suppose

$$Y = \Phi\theta + E, \qquad \text{and } \theta \in N(0, P(\alpha)), \quad E \in N(0, \sigma^2 I)$$

Then

$$Y \in N(0, \Sigma(\alpha)) \qquad \Sigma(\alpha) = \Phi P(\alpha)\Phi^T + \sigma^2 I$$

Tuning the Regularization Matrix

Suppose

$$Y = \Phi\theta + E, \qquad \text{and } \theta \in N(0, P(\alpha)), \quad E \in N(0, \sigma^2 I)$$

Then

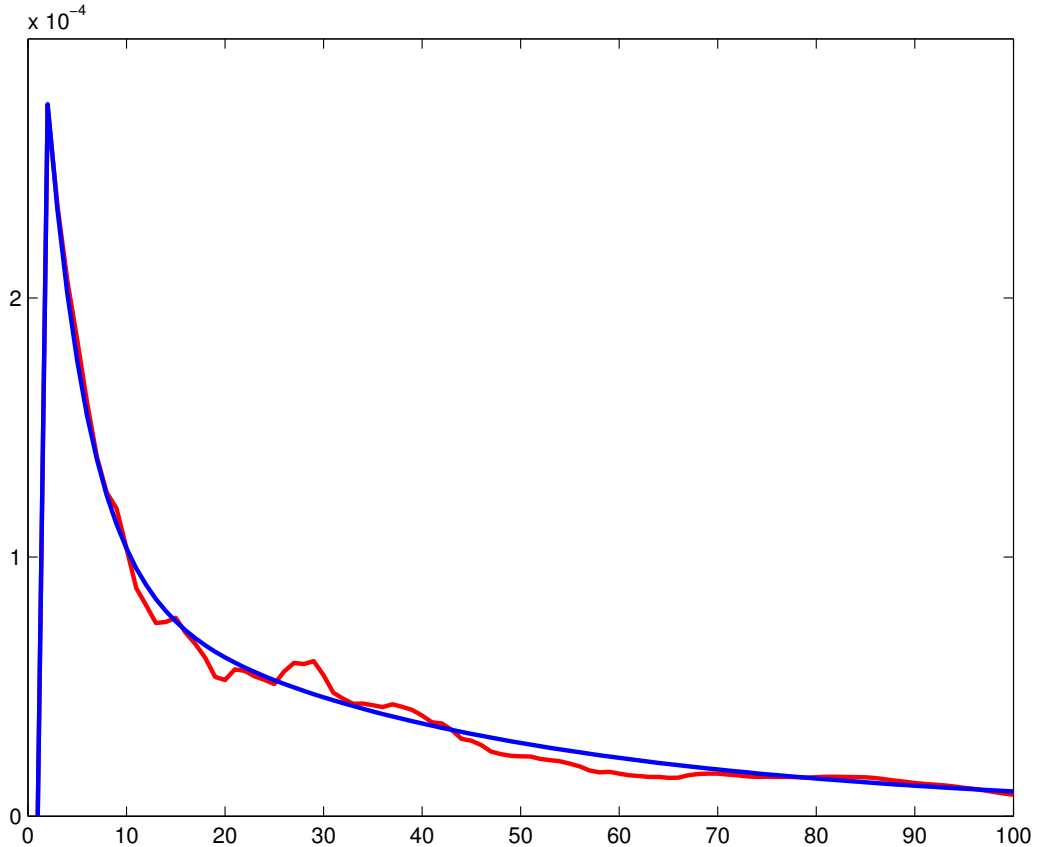
$$Y \in N(0, \Sigma(\alpha)) \qquad \Sigma(\alpha) = \Phi P(\alpha)\Phi^T + \sigma^2 I$$

so we know the pdf of Y up to the parameter α and it can be estimated by Maximum Likelihood:

$$\hat{\alpha} = \arg\min_{\alpha} Y^T \Sigma(\alpha)^{-1} Y + \log\det\Sigma(\alpha)$$

The Tuned Regularized FIR model

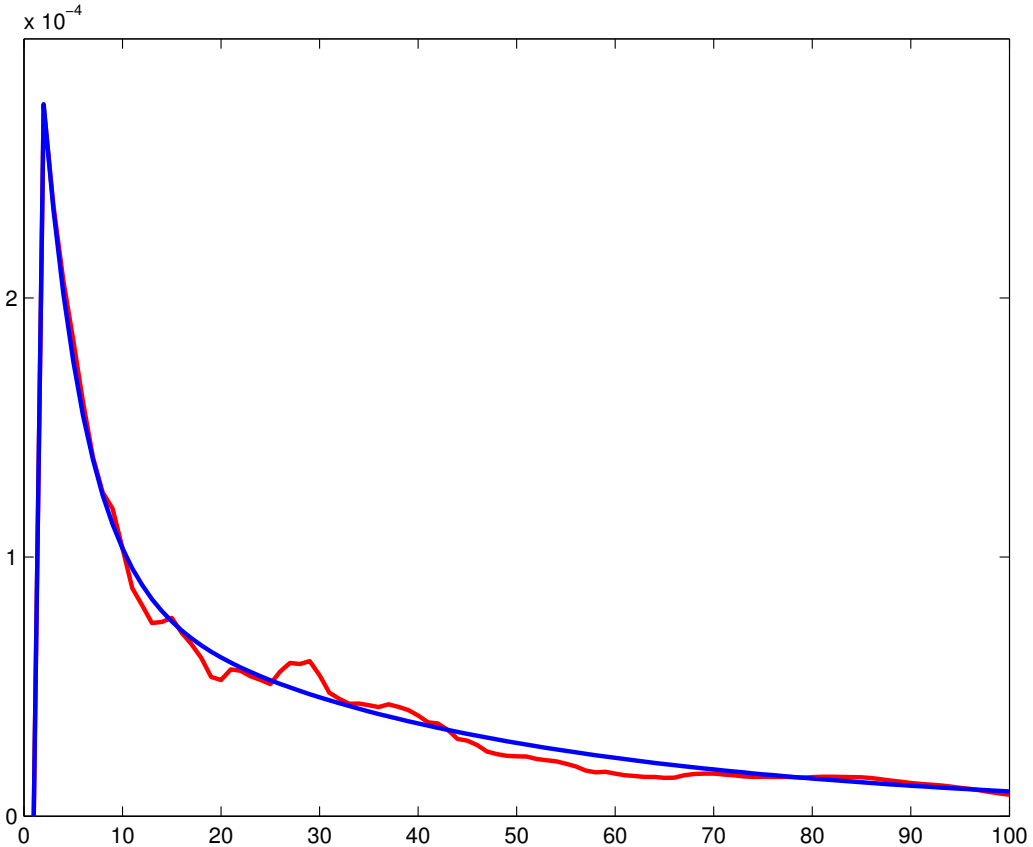
$$\arg\min_\theta \|Y - \Phi\theta\|^2 + \theta^T P^{-1}(\hat{\alpha})\theta$$



Fit: **90.82%**

The Tuned Regularized FIR model

$$\arg\min_\theta \|Y - \Phi\theta\|^2 + \theta^T P^{-1}(\hat{\alpha})\theta$$



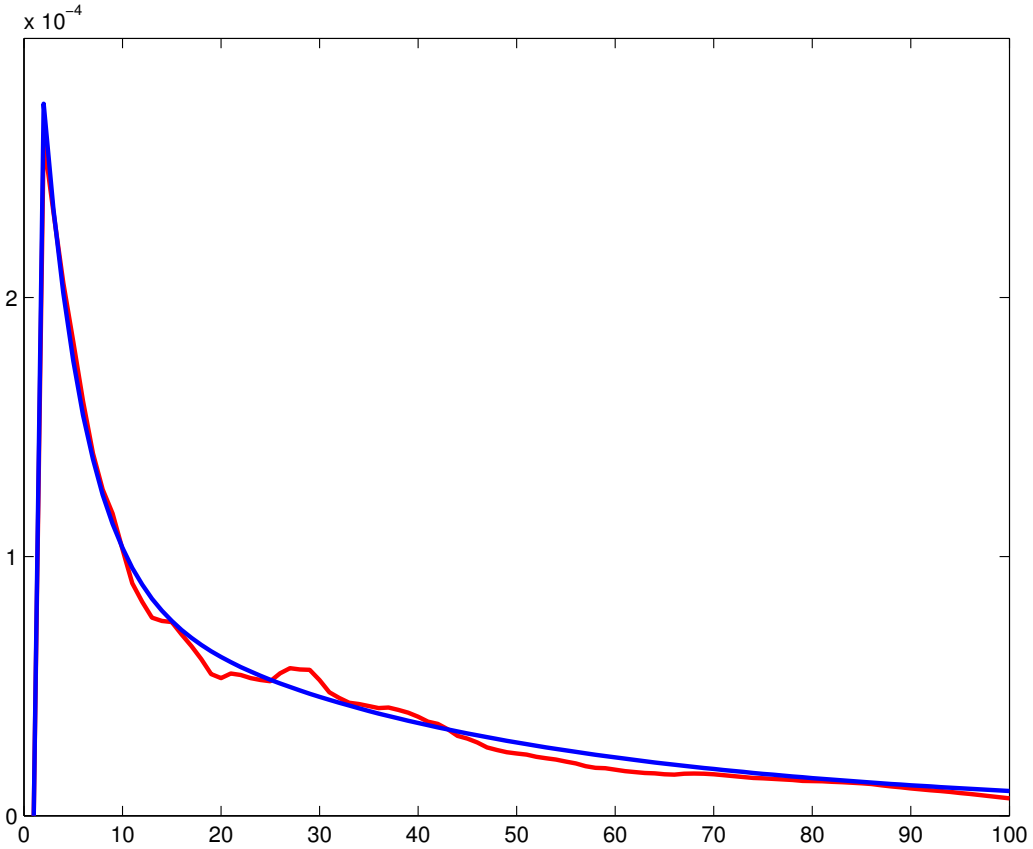
Fit: **90.82%**

The Estimation of the Machine Learning Model follows exactly the same algorithm, but with a slightly different parameterization of $P(\alpha)$: The "stable splines" correspond to the parameterization we used with $\rho = \sqrt{\mu}$.

A Further Tuned Regularized FIR model

$$\arg\min_\theta \|Y - \Phi\theta\|^2 + 1.9 \cdot \theta^T P^{-1}(\hat{\alpha})\theta$$

1.9 chosen by cross-validation.



Fit: **91.72%**

General linear model

$$y(t) = G(\theta, q)u(t) + H(q, \theta)e(t)$$

General linear model

$$y(t) = G(\theta, q)u(t) + H(q, \theta)e(t)$$

The predictor:

$$\hat{y}(t|\theta) = (I - H^{-1})y(t) + H^{-1}Gu(t) = \sum_{k=1}^{\infty} \tilde{h}(k)y(t-k) + \tilde{g}(k)u(t-k)$$

General linear model

$$y(t) = G(\theta, q)u(t) + H(q, \theta)e(t)$$

The predictor:

$$\hat{y}(t|\theta) = (I - H^{-1})y(t) + H^{-1}Gu(t) = \sum_{k=1}^{\infty} \tilde{h}(k)y(t-k) + \tilde{g}(k)u(t-k)$$

Response from y and u are exponentially stable:

General linear model

$$y(t) = G(\theta, q)u(t) + H(q, \theta)e(t)$$

The predictor:

$$\hat{y}(t|\theta) = (I - H^{-1})y(t) + H^{-1}Gu(t) = \sum_{k=1}^{\infty} \tilde{h}(k)y(t-k) + \tilde{g}(k)u(t-k)$$

Response from y and u are exponentially stable:Truncate the sum at n_a and n_b:

$$\hat{y}(t|\theta)) = (I - A(q))y(t) + B(q)u(t);$$

General linear model

$$y(t) = G(\theta, q)u(t) + H(q, \theta)e(t)$$

The predictor:

$$\hat{y}(t|\theta) = (I - H^{-1})y(t) + H^{-1}Gu(t) = \sum_{k=1}^{\infty} \tilde{h}(k)y(t-k) + \tilde{g}(k)u(t-k)$$

Response from y and u are exponentially stable:Truncate the sum at n_a and n_b:

$$\hat{y}(t|\theta)) = (I - A(q))y(t) + B(q)u(t);$$

$\Rightarrow$ The same as an ARX model:

$$A(q)y(t) = B(q)u(t) + e(t)$$

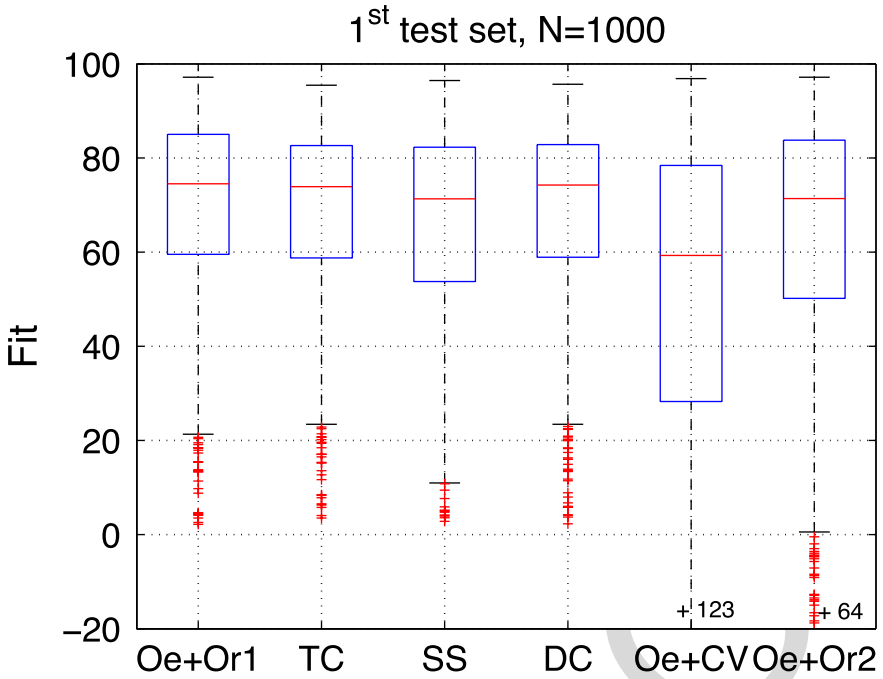
(Functionally the same as a FIR model with n_y+n_u inputs)

So, ARX models are universal approximators of any linear model!

Postulate a “high order” ARX model and apply the same regularization as for FIR models. Different hyperparameters can be used for the various ”inputs” (outputs). The regularized high order estimates can be compared to conventional lower order models (OE, ARMAX, BJ, State Space ...).

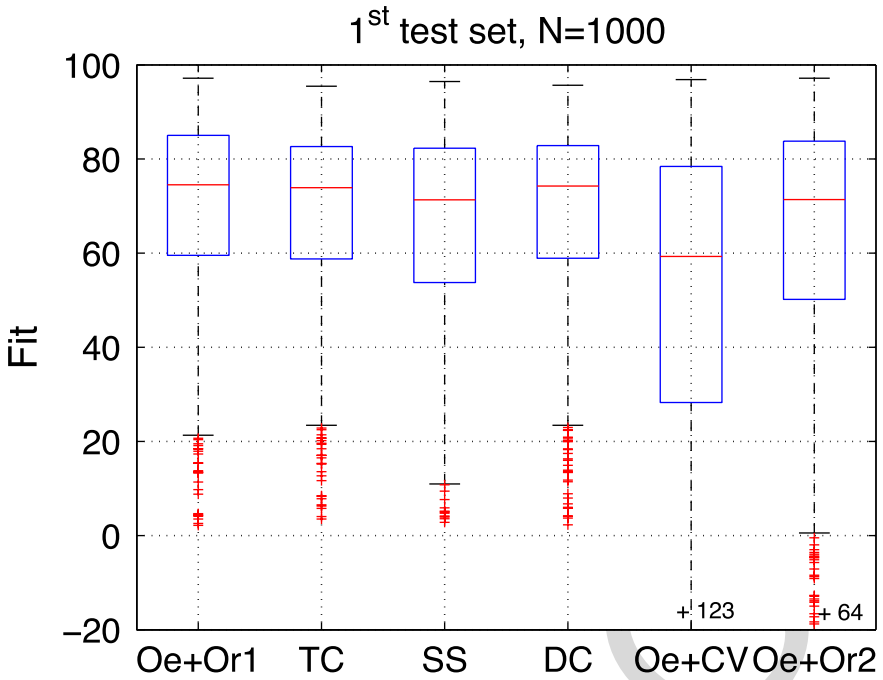
Handling General Linear Models

Postulate a “high order” ARX model and apply the same regularization as for FIR models. Different hyperparameters can be used for the various ”inputs” (outputs). The regularized high order estimates can be compared to conventional lower order models (OE, ARMAX, BJ, State Space ...).



Handling General Linear Models

Postulate a “high order” ARX model and apply the same regularization as for FIR models. Different hyperparameters can be used for the various ”inputs” (outputs). The regularized high order estimates can be compared to conventional lower order models (OE, ARMAX, BJ, State Space ...).



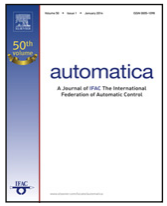
If desired, the high order ARX model can then be transformed to a lower order State Space model by model order reduction.

Automatica 50 (2014) 657–682

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica



Survey Paper

Kernel methods in system identification, machine learning and function estimation: A survey☆

Gianluigi Pillonetto [a,1], Francesco Dinuzzo [b], Tianshi Chen [c], Giuseppe De Nicolao [d], Lennart Ljung [c]

[a] Department of Information Engineering, University of Padova, Padova, Italy
[b] Max Planck Institute for Intelligent Systems, Tübingen, Germany
[c] Division of Automatic Control, Linköping University, Linköping, Sweden
[d] Department of Computer Engineering and Systems Science, University of Pavia, Pavia, Italy

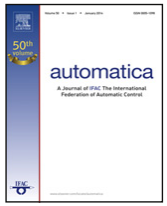
Kernel Methods in System Identification

Automatica 50 (2014) 657–682

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica



Survey Paper

Kernel methods in system identification, machine learning and function estimation: A survey☆

Gianluigi Pillonetto [a,1], Francesco Dinuzzo [b], Tianshi Chen [c], Giuseppe De Nicolao [d], Lennart Ljung [c]

[a] Department of Information Engineering, University of Padova, Padova, Italy
[b] Max Planck Institute for Intelligent Systems, Tübingen, Germany
[c] Division of Automatic Control, Linköping University, Linköping, Sweden
[d] Department of Computer Engineering and Systems Science, University of Pavia, Pavia, Italy

Is this a new Paradigm for SI?

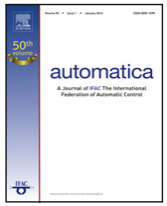
Kernel Methods in System Identification

Automatica 50 (2014) 657–682

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica



Survey Paper

Kernel methods in system identification, machine learning and function estimation: A survey☆

Gianluigi Pillonetto [a,1], Francesco Dinuzzo [b], Tianshi Chen [c], Giuseppe De Nicolao [d], Lennart Ljung [c]

[a] Department of Information Engineering, University of Padova, Padova, Italy
[b] Max Planck Institute for Intelligent Systems, Tübingen, Germany
[c] Division of Automatic Control, Linköping University, Linköping, Sweden
[d] Department of Computer Engineering and Systems Science, University of Pavia, Pavia, Italy

Is this a new Paradigm for SI?

- Well, not really. Regularization is a classical tool in estimation theory. It has not been used extensively in SI, though.

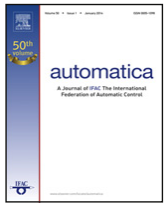
Kernel Methods in System Identification

Automatica 50 (2014) 657–682

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica



Survey Paper

Kernel methods in system identification, machine learning and function estimation: A survey☆

Gianluigi Pillonetto [a,1], Francesco Dinuzzo [b], Tianshi Chen [c], Giuseppe De Nicolao [d], Lennart Ljung [c]

[a] Department of Information Engineering, University of Padova, Padova, Italy
[b] Max Planck Institute for Intelligent Systems, Tübingen, Germany
[c] Division of Automatic Control, Linköping University, Linköping, Sweden
[d] Department of Computer Engineering and Systems Science, University of Pavia, Pavia, Italy

Is this a new Paradigm for SI?

- Well, not really. Regularization is a classical tool in estimation theory. It has not been used extensively in SI, though.
- Also the tuning possibilities of general L2-regularization norms have not been a topic of SI research before.

Conclusions

- The function learning approach to system identification corresponds to regularized FIR-model estimation, with careful tuning of the regularization.

Conclusions

- The function learning approach to system identification corresponds to regularized FIR-model estimation, with careful tuning of the regularization.
- This regularized FIR (or ARX) approach is a valuable complement to the standard paradigm of system identification

Conclusions

- The function learning approach to system identification corresponds to regularized FIR-model estimation, with careful tuning of the regularization.
- This regularized FIR (or ARX) approach is a valuable complement to the standard paradigm of system identification
- It can clearly be derived and explained entirely within the conventional system identification box of tools.

So What Role does Machine Learning Play for SI?

- Important inspiration for estimation of black box models

So What Role does Machine Learning Play for SI?

- Important inspiration for estimation of black box models
- Several other sub-problems in SI: manifold learning, ...

So What Role does Machine Learning Play for SI?

- Important inspiration for estimation of black box models
- Several other sub-problems in SI: manifold learning, ...
- Machine Learning is itself merging with basic statistics

So What Role does Machine Learning Play for SI?

- Important inspiration for estimation of black box models
- Several other sub-problems in SI: manifold learning, ...
- Machine Learning is itself merging with basic statistics
- As, always: be alert to new ideas in neighbouring areas!