System Identification:

An Inverse Problem in Control

Potentials and Possibilities of Regularization

Lennart Ljung

Linköping University, Sweden

Inverse Problems, MAI, Linköping, April 4, 2013

System Identification






System
Identification
(since 1956)

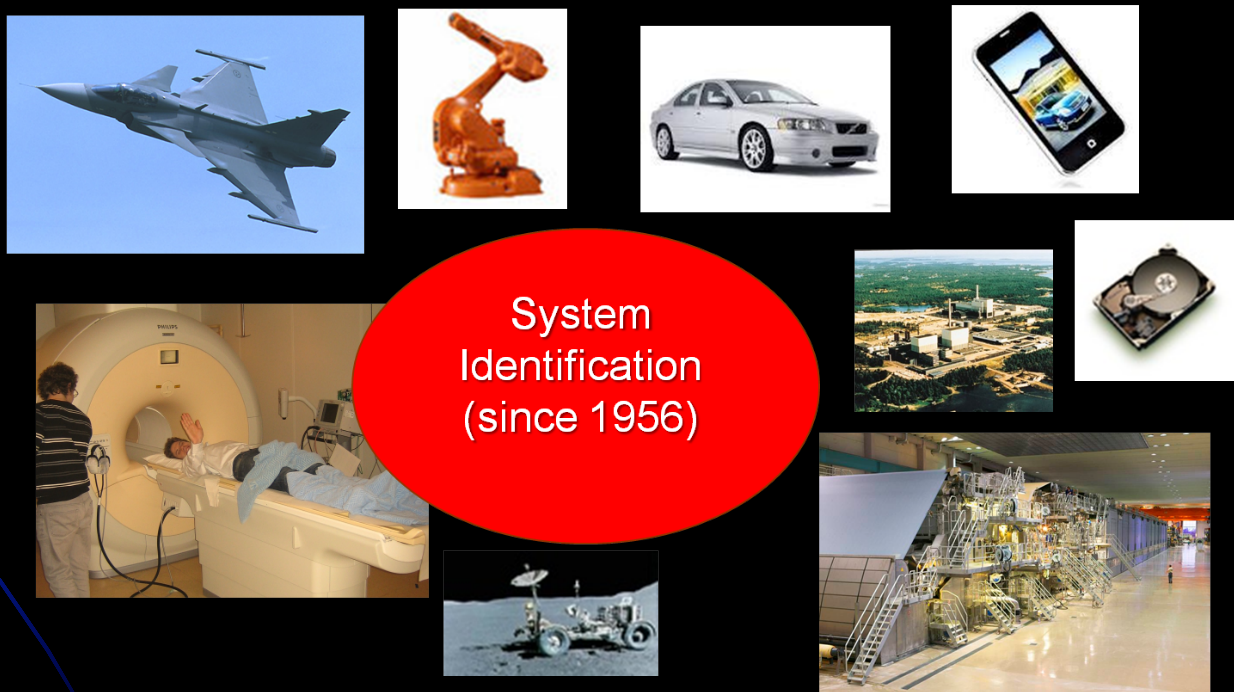


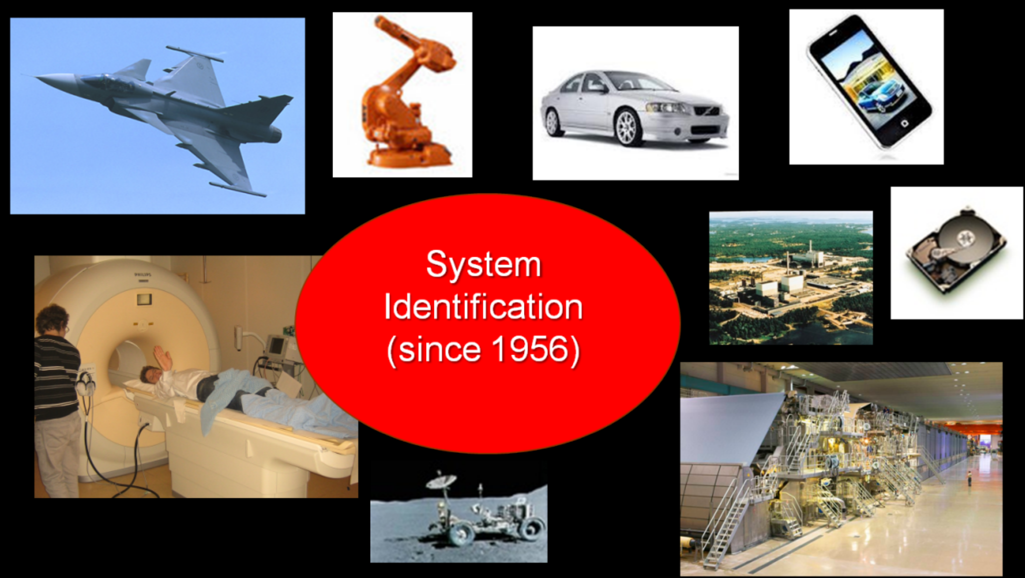





System
Identification
(since 1956)




System
Identification
(since 1956)


System
Identification
(since 1956)

The World Around System Identification

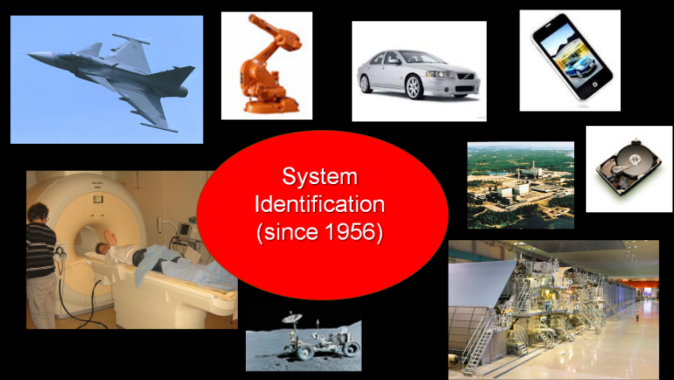
Statistical Learning theory

Manifold learning

Sparsity

Statistics

Machine Learning



Networked systems

Compressed sensing

Particle filters

Abstract

- System Identification is a well established area in Automatic Control
- To find a system that (may) have generated observed input-output signals
- An inverse problem!
- The role of regularization. – Recently re-discovered in the field.

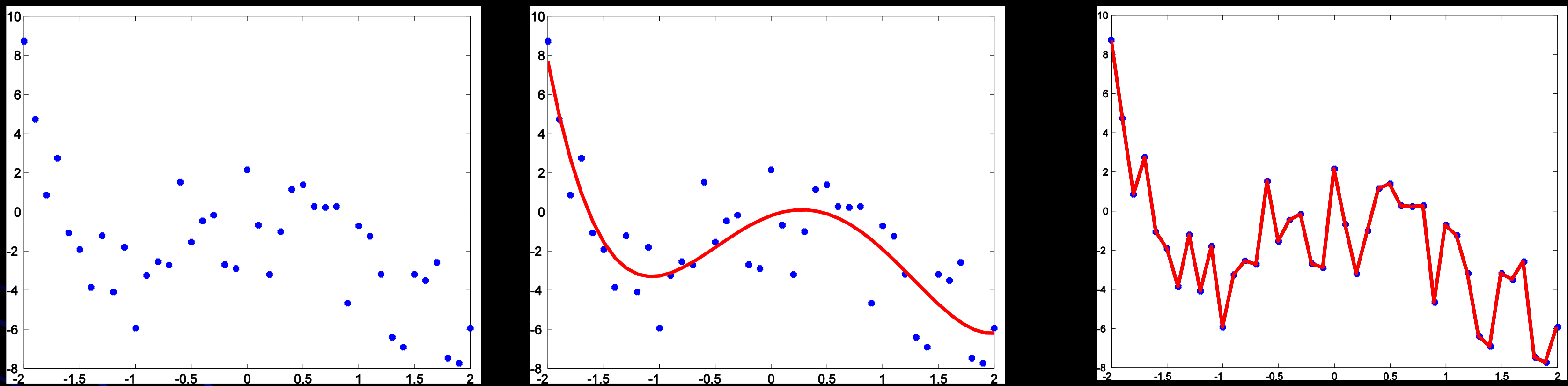
Outline

- Preamble: A quick primer on estimation and system identification
- The standard approach to build a model
- A new algorithm
- .....

A Primer on Estimation

Squeeze out the relevant information in data

But NOT MORE !

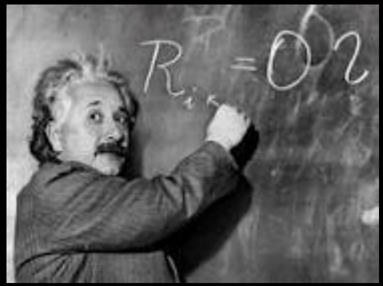


All data contain information and misinformation ("Signal and noise")

So need to meet the data with a prejudice!

Primer: Estimation Prejudices

- Nature is Simple!
  - Occam's razor
    Principle of parsimony



  - God is subtle, but He is not malicious (Einstein)

- So, conceptually, when you build a model:

$$\hat{\mathfrak{m}} = \arg\min_{\mathfrak{m} \in \mathcal{M}} (\text{Fit} + \text{Complexity Penalty})$$

Primer: Bias and Variance

$\mathcal{S}$ – True system $\quad \hat{\mathfrak{m}}$ – Estimate $\quad \mathfrak{m}^* = E\hat{\mathfrak{m}}$

$\hat{\mathfrak{m}} \in \mathcal{M}$: Typically $\mathfrak{m}^*$ is the model closest to $\mathcal{S}$ in $\mathcal{M}$.

$$\mathbf{E}\|\mathcal{S} - \hat{\mathfrak{m}}\|^2 = \|\mathcal{S} - \mathfrak{m}^*\|^2 + \mathbf{E}\|\hat{\mathfrak{m}} - \mathfrak{m}^*\|^2$$

MSE	=	BIAS (B)	+	VARIANCE (V)
Error	=	Systematic	+	Random

As complexity of $\mathcal{M}$ increases, B decreases &V increases

This bias/variance tradeoff is at the heart of estimation!

The best MSE trade-off typically has non-zero bias!

Take Home Messages from the Preamble

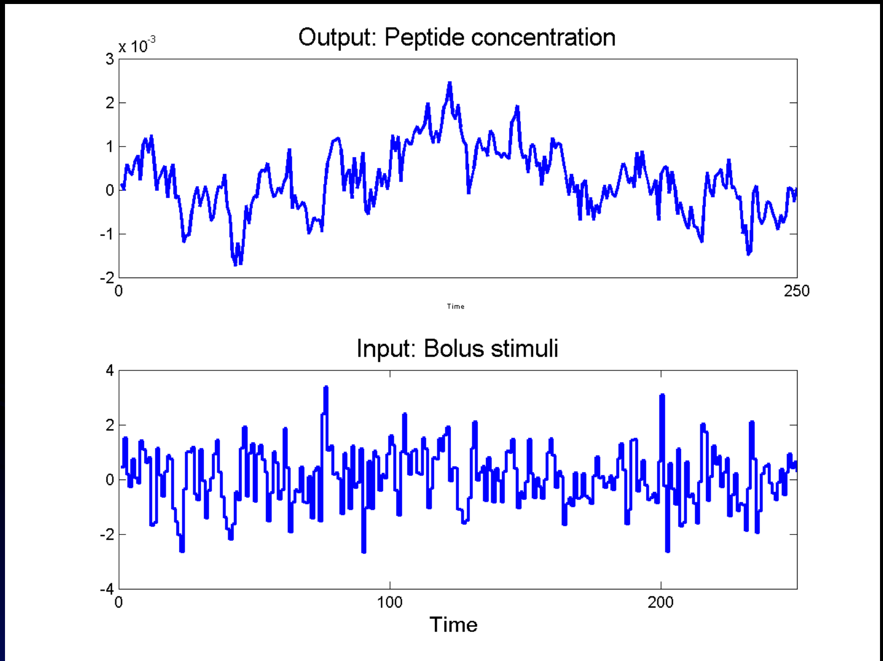
Seek parsimonious models

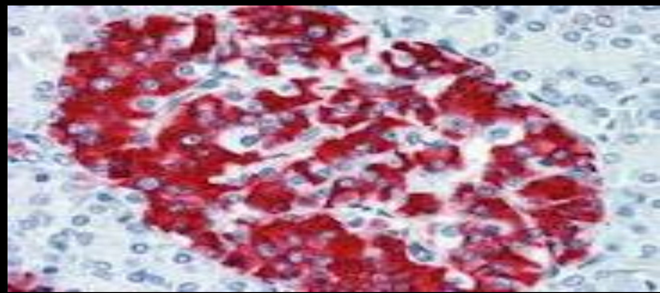
The bias/variance trade-off is at the heart of estimation

Mature area with traces to old history but still open for new encounters

An Eyeopening Encounter

I was given input-output data that mimics the C-peptide dynamics in humans





Find a good estimate of the impulse response (transfer function) of the system

The Traditional Approach

(My) traditional approach (Maximum Likelihood, ML):

Build state-space models of certain order n

(n:th order Difference equations)

by **`pem(data,n)`**

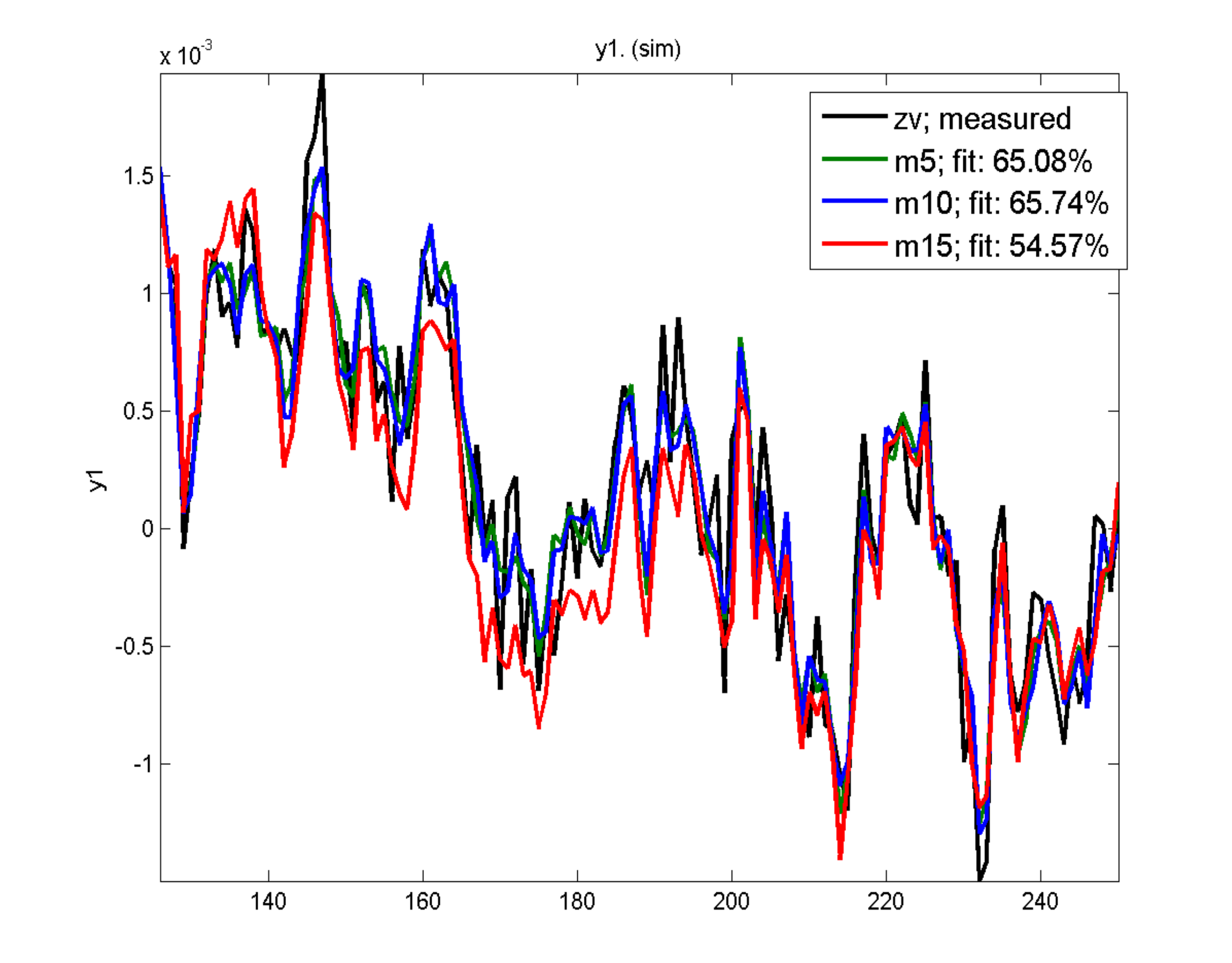
`Use cross validation to find n:`

```
ze=z(1:125); zv=z(126:end);
 m5=pem(ze,5); m10=pem(ze,10);
 m15=pem(ze,15);
```

Model Order Choice

Compare model output with models' simulated outputs

```
compare(zv,m5,m10,m15)
```



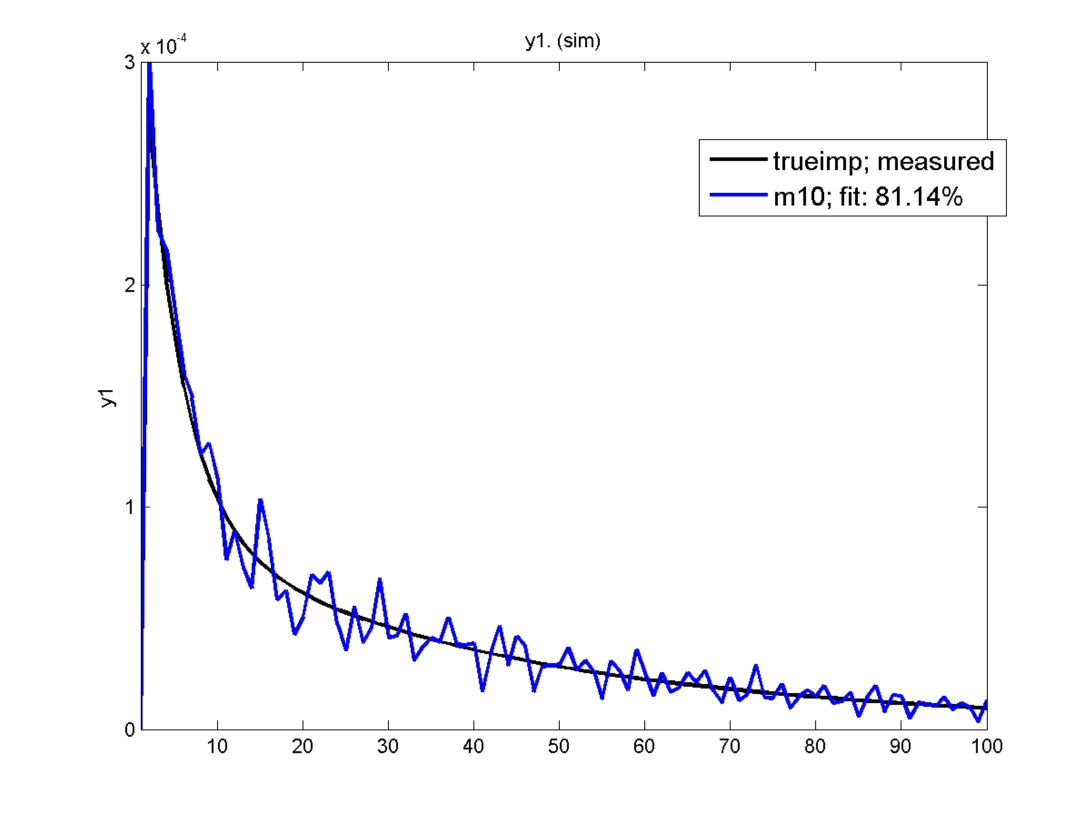
Order 10 is best, reestimate:

```
m10=pem(z,10);
```

Compare with True Impulse Response

Happen to find the true impulse response. How good was my estimate?

```
compare(trueimp,m10,'ini','z')
```



81,1%

Another proposed method: XXX

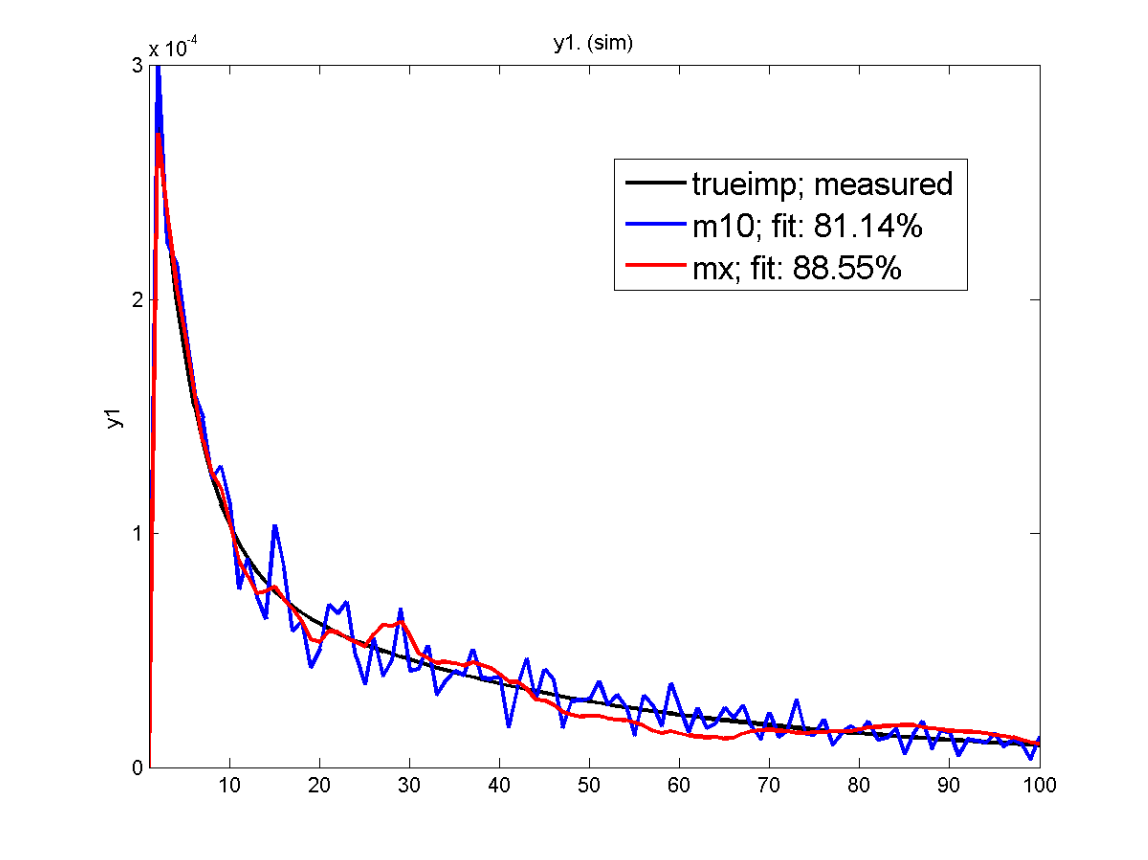
Here is a new approach to system identification: mfile xxx

```
>> help xxx
This is a magic algorithm for
system identification.
  Try me!
  Just do Model = xxx(Data)
```

So, let's do that!

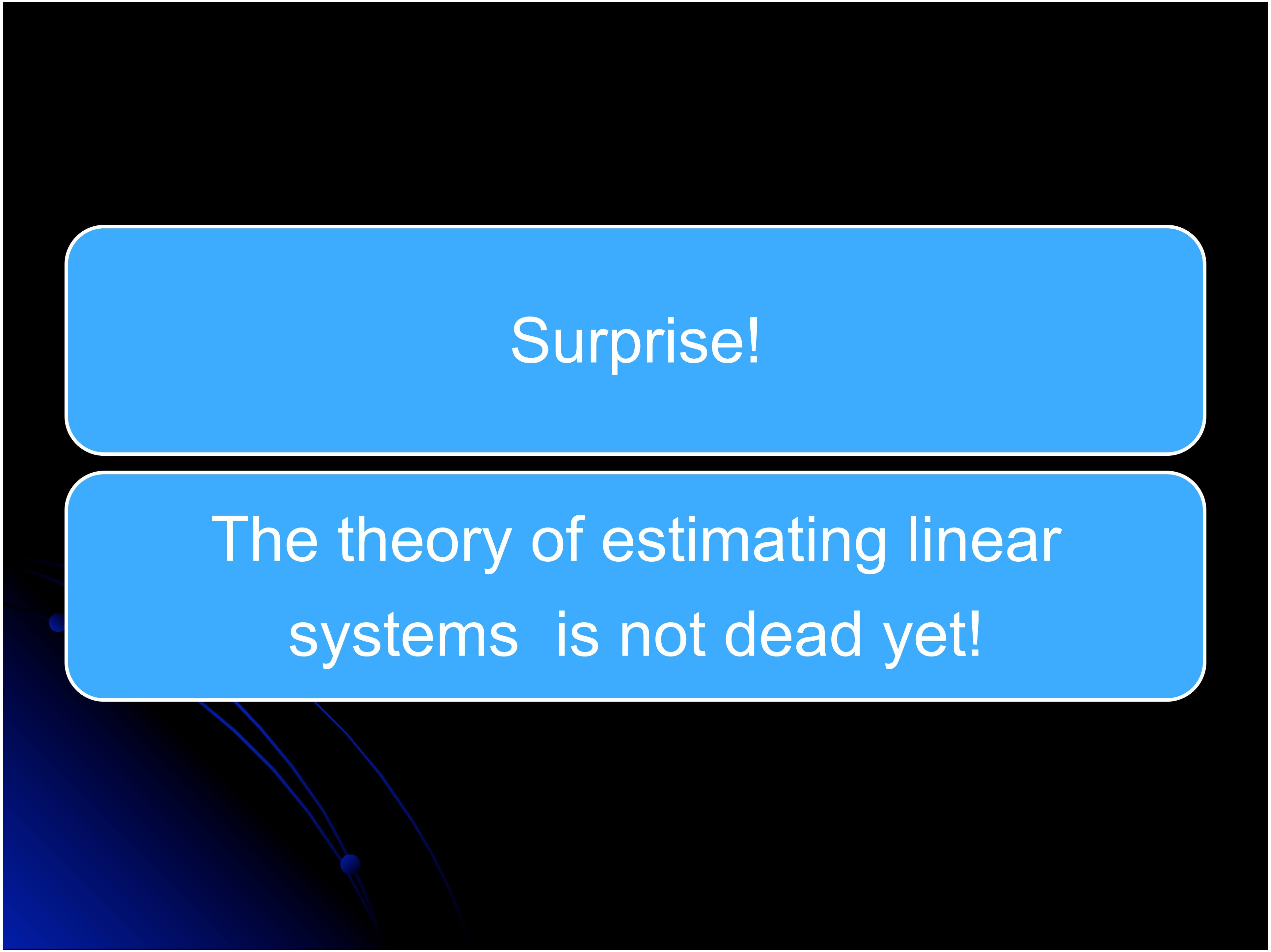
Estimate of the Impulse Response

```
mx = xxx(z); compare(trueimp,mx,m10,'ini','z')
```



Fit mx: 88.6%

Fit m10: 81,1%


Surprise!
The theory of estimating linear systems is not dead yet!

What is the Key Idea?

Regularization:

- Use flexible model structures with (too) many parameters
- Which ones are not quite necessary?
- Put the parameters on leashes and check which ones are most eager in the pursuit for a good fit!
  - Pull parameters towards zero (ℓ_2)
  - Pull parameters to zero (ℓ_1)

Outline for Remainder of Talk

Regularization: Curb the freedom in flexible models.

ℓ_2

- Regularization for bias/variance tradeoff
- Regularization for manifold learning
- Regularization for sparsity and parsimony

ℓ_1

Outline

- Regularization for bias/variance tradeoff
- Regularization for manifold learning
- Regularization for sparsity and parsimony

Regularization

Recall: $\hat{\mathfrak{m}} = \arg\min_{\mathfrak{m}\in\mathcal{M}} (\textbf{Fit} + \textbf{Complexity Penalty})$

E.g. Linear Regression: $\mathbf{Y} = \mathbf{\Phi}\theta + \mathbf{E}$

$$[\mathbf{y}(\mathbf{t}) = \sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{n}} \mathbf{g}(\mathbf{k})\mathbf{u}(\mathbf{t}-\mathbf{k}) + \mathbf{e}(\mathbf{t})].$$

$$\hat{\theta}^{\mathrm{LS}} = \arg\min \|\mathbf{Y} - \mathbf{\Phi}\theta\|^{\mathbf{2}}$$

(Too) many parameters? Put them on leashes!

$$\hat{\theta}^{\mathrm{R}} = \mathbf{arg\ min}\|\mathbf{Y} - \mathbf{\Phi}\theta\|^{\mathbf{2}} + \theta^{\mathbf{T}}\mathbf{P}^{-\mathbf{1}}\theta$$

A Frequentist Perspective

$$\hat{\theta}^{\mathrm{R}} = (\mathbf{R_N} + \mathbf{P}^{-1})^{-1}\mathbf{R_N}\hat{\theta}^{\mathrm{LS}}, \quad \mathbf{R_N} = \mathbf{\Phi^T\Phi}$$

Frequentist (classical) perspective

True parameter θ_0 noise variance $\sigma^2 (= 1)$

$$\mathrm{BIAS:} \quad \mathrm{E}\hat{\theta}^{\mathrm{R}} - \theta_{\mathbf{0}} = -(\mathrm{R_N} + \mathrm{P}^{-1})^{-1}\mathrm{P}^{-1}\theta_{\mathbf{0}}$$

$$\mathbf{MSE:} \quad \mathrm{E}(\hat{\theta}^{\mathrm{R}} - \theta_{\mathbf{0}})(\hat{\theta}^{\mathrm{R}} - \theta_{\mathbf{0}})^{\mathrm{T}} =$$

$$(\mathbf{R_N} + \mathbf{P}^{-1})^{-1}(\mathbf{R_N} + \mathbf{P}^{-1}\theta_{\mathbf{0}}\theta_{\mathbf{0}}^{\mathbf{T}}\mathbf{P}^{-1})(\mathbf{R_N} + \mathbf{P}^{-1})^{-1}$$

$$\textbf{No Regul,} \quad \mathbf{P}^{-1} = \mathbf{0}: \quad \mathrm{BIAS} = \mathbf{0}, \quad \mathbf{MSE} = \mathbf{R_N^{-1}}$$

The choice $\mathrm{P} = \theta_0\theta_0^{\mathrm{T}}$ minimizes the MSE to $(\mathrm{R_N} + \mathrm{P}^{-1})^{-1}$

...

Bayesian Interpretation

θ is a random variable that before observing (a priori) $\mathbf{Y}$ is $\mathbf{N(0,P)}$ i.e. the negative log of its pdf is $\sim \theta^{\mathbf{T}}\mathbf{P}^{-1}\theta$

and its pdf after (a posteriori) is $\sim \|\mathbf{Y} - \Phi\theta\|^2 + \theta^{\mathbf{T}}\mathbf{P}^{-1}\theta$

This is the Regularized LS criterion!

pdf: probability density function

So, the reg. LS estimate $\hat{\theta}^{\mathbf{R}} = (\mathbf{R_N} + \mathbf{P}^{-1})^{-1}\mathbf{R_N}\hat{\theta}^{\mathbf{LS}}$

gives the maximum of this pdf (MAP),

(the Bayesian posterior estimate)

Clue to the choice of *P*!

Estimation of Impulse Response

$$\mathbf{Y} = \mathbf{\Phi}\theta + \mathbf{E}, \qquad \mathbf{y(t)} = \sum_{\mathbf{k=1}}^{\mathbf{n}} \mathbf{g(k)u(t-k)} + \mathbf{e(t)}.$$

A good prior for $\theta \in \mathbf{N}(\mathbf{0}, \mathbf{P})$ describes the behaviour of the typical impulse response *g(k):*

- Exponentially decaying, size *C*, rate λ
- Smooth as a function of *k*, correlation ρ
- $\mathbf{P}(\beta),\ \beta = [\mathbf{C}, \lambda, \rho]$

Estimate (the hyperparameters) β from data

Estimation of Hyperparameters

$$\mathbf{Y} = \mathbf{\Phi}\theta + \mathbf{E}$$

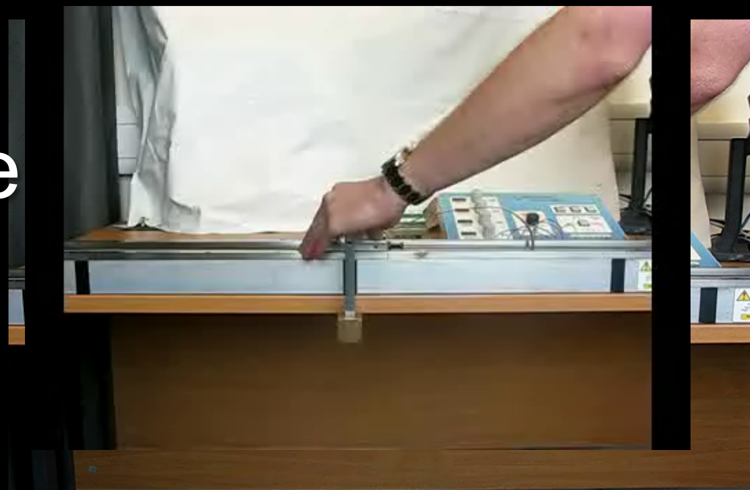
In a Bayesian framework, **Y** is a random variable with a distribution that depends on the hyperparameters. Estimate those by ML!

- "Empirical Bayes" (EB)
- xxx: estimate $\hat{\beta}$ by EB and use $\mathbf{P}(\hat{\beta})$ in regularized LS! (= RFIR)
- Original research and results by Pillonetto, De Nicolao and Chiuso

A Link to Machine Learning ”Gaussian Processes (GP)”

The IR estimation algorithm is a case of GP function estimation,

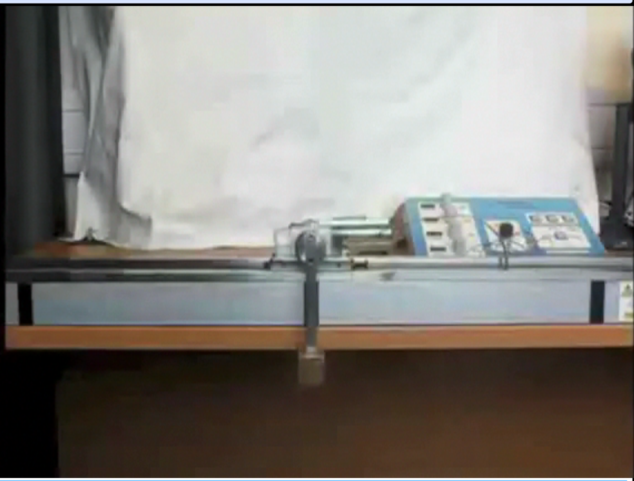
frequently used in Machine Learning.



(Pillonetto et al used this framework to device the XXX algorithm)

GP: Estimate a Function *f(x)*

Observe $y(t), t = 1, \ldots, N$, that are linear functionals of f measured in Gaussian noise $y(t) = L(t, f) + e(t)$



Assume a Gaussian prior for *f*

$Ef(x) = 0, Ef(r)f(s) = K(r, s)$

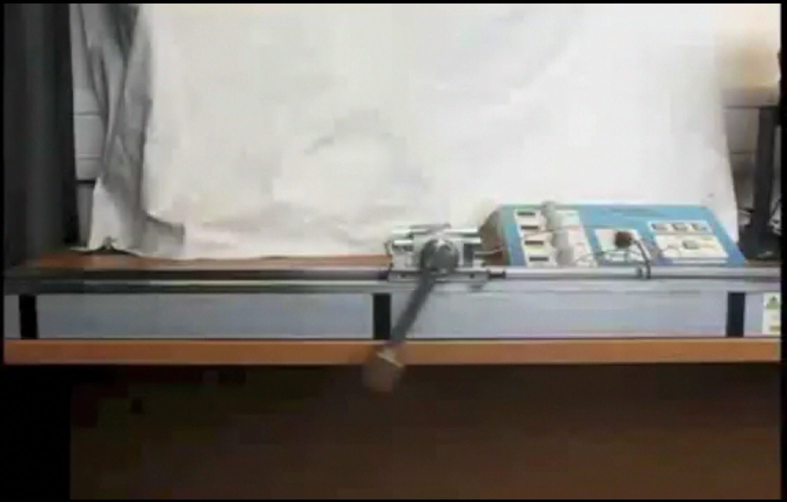
Compute the posterior estimate given the observations

$\hat{f}(x) = E(f(x)|Y_1^N)$

These are the same as the previous Bayesian calculations!

Machine Learning of Dynamic Systems

Carl Rasmussen (Machine Learning Group, Cambridge) has performed quite spectacular experiments by swinging up an inverted pendulum using MPC and a model estimated by GP .



The function estimated is the state transition function

$$\mathbf{x}(\mathbf{t}+1) = \mathbf{f}(\mathbf{x}(\mathbf{t}), \mathbf{u}(\mathbf{t})) \quad \mathrm{f}(\mathrm{x}, \mathrm{u}) \text{ from } \mathrm{R}^5 \text{ to } \mathrm{R}^4$$

GP: Duality with RKHS

Let the prior pdf of the function *f* have a covariance function *K* associated with a Reproducing Kernel Hilbert Space $\mathcal{H}$. Then the Bayesian posterior estimate of *f* is given as

$$\min_{\mathbf{f}} \sum (\mathbf{y}(\mathbf{t}) - \mathbf{L}(\mathbf{t}, \mathbf{f}))^2 + \|\mathbf{f}(\cdot)\|_{\mathcal{H}}^2$$

Compare with the finite dimensional FIR case:

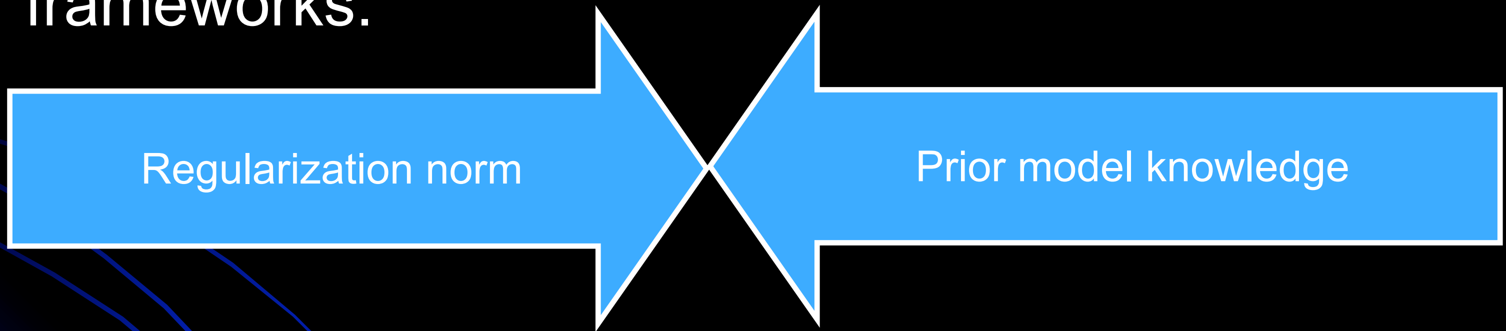
$$\min_\theta \|\mathbf{Y} - \Phi\theta\|^2 + \theta^\mathbf{T}\mathbf{P}^{-1}\theta$$

This is a much studied problem in statistics and machine learning (Wahba, Schölkopf,…)

Summary: Quadratic Norm Regularization

- Regularization for bias/variance tradeoff

1. Symbiosis with Bayesian calculations in Gaussian frameworks.



2. Well tuned regularization norm (e.g. by EB) can give significant improvement in model quality (MSE)

Outline

- Regularization for bias/variance tradeoff
- Regularization for manifold learning
- Regularization for sparsity and parsimony

Tailored Regularization

$$\mathbf{min_f \sum (y(t) - f(x_t))^2 + \lambda R(f(\cdot))}$$

More pragmatic: Known/desired properties of *f(x)* can be expressed in terms of *R*.

Intriguing special case: We want to estimate *f* when the ”regressors” $\mathbf{x_t}$ are confined to an unknown manifold: We need to estimate that manifold at the same time as *f* : ”manifold learning”.

Manifold Learning and (NL)Dimension Reduction

If we know that the regressors *x* in a mapping *y=f(x)* are confined to a lower dimensional manifold, we may write

y= f(g(x)), where *g(x)* are local coordinates (dim *g(x)* < dim *x*) on the manifold. This would give a simpler model.

How to find the manifold *g(x)?* [Linear case: SVD, PCA ,…]

NL case: ISOmap, KPCA, Diffeomap, ..,

LLE (Local Linear Embedding): Find a weight matrix K that describes the local metric of the regressors:

$$\mathrm{x_t} \approx \sum \mathrm{K_{ts} x_s}$$

That matrix can be used to construct the lower dimensional local coordinates.

Function Estimation on Unknown Manifolds

Build a model $\mathbf{y(t) = f(x_t)}, \quad \mathbf{x_t} \in ?$

A weight matrix *K* describing the regressor manifold is constructed by LLE and that is used to penalize non-smoothness over the associated manifold:

$$\mathbf{min}_{\hat{\mathbf{f}}_\mathbf{t}} \sum_{\mathbf{t}} (\mathbf{y(t)} - \hat{\mathbf{f}}_\mathbf{t})^2 + \lambda \sum_{\mathbf{i}} \left(\hat{\mathbf{f}}_\mathbf{i} - \sum_{\mathbf{j}} \mathbf{K_{ij}} \hat{\mathbf{f}}_\mathbf{j} \right)^2$$

$$\hat{\mathbf{f}}_\mathbf{t} = \hat{\mathbf{f}}(\mathbf{x_t}), \quad \mathbf{K_{ij}} = \mathbf{from\ LLE}$$

WDMR: Weight determination by manifold regression

Quadratic in *f*!

Let’s apply it to brain activity analysis (fMRI)!

The Observed Data

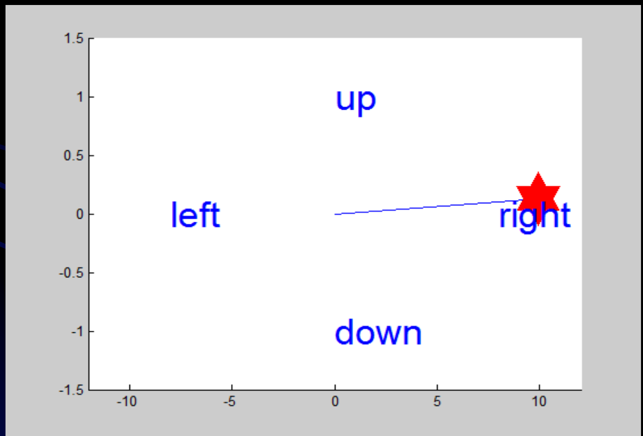
The person in the magnet camera is moving his eye focus in a circle left - right - up – down and his eye focus is measured as $\mathbf{y}(\mathbf{t}) \in [-\pi, \pi]$.

128 voxels in the visual cortex are monitored by fMRI, giving a regression vector $\mathbf{x_t} \in \mathrm{R}^{128}$. Data are sampled every two seconds for five minutes.

The Observed Data

The person in the magnet camera is moving his eye focus in a circle left - right - up – down and his eye focus is measured as $y(t) \in [-\pi, \pi]$.

128 voxels in the visual cortex are monitored by fMRI, giving a regression vector $x_t \in R^{128}$. Data are sampled every two seconds for five minutes.



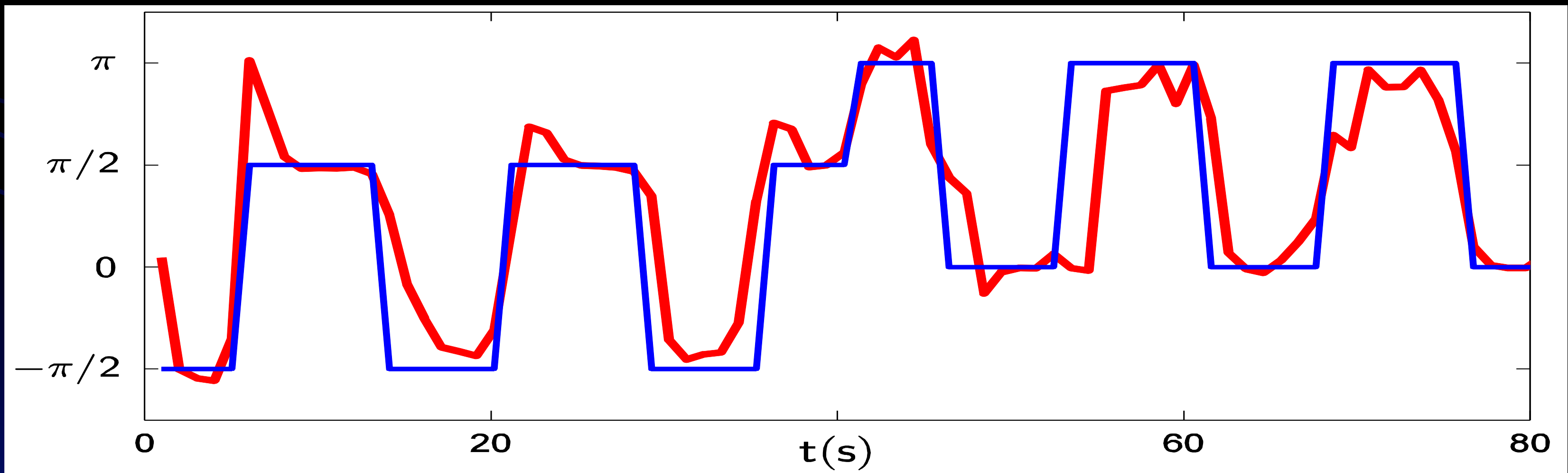
The regressor x_t *is 128-dimensional.* At the same time the "brain activity is 1-dimensional", so the interesting variation in the regressor space should be confined to a one-dimensional manifold

WDMR: Estimated model

Let us apply WDMR to these data!

Build a model using 110 data. Validate it on the remaining 40.

Below we show the predicted y-values ($\text{angles } [-\pi, \pi]$) *(red) for* validation measurements together with the corresponding true angles (blue).



Recall: 110 estimation data in $\mathbb{R}^{128}$!

Summary: Tailored Regularization

- Regularization for manifold learning

1. Added regularization penalties to criteria of fit can be used in an ad hoc manner
2. Constraints on the regressor space can be handled quite well in this way
3. Broader implications for System Identification unclear

Outline

- Regularization for bias/variance tradeoff
- Regularization for manifold learning
- Regularization for sparsity and parsimony

Regularization for Parsimony

Parsimony: Find good model fits, without being wasteful with parameters. Conceptually (model error)

$$\min \sum \varepsilon^2(t,\theta) + \lambda \|\theta\|_0.$$

$\|\cdot\|_0 = \ell_0$–"norm": The number of non-zero elements

Compare with Akaike's AIC!

- OK to solve with "linearly ordered" model families (like FIR models). Combinatorial explosions for richer model structures, like polynomial nonlinearities, or neural networks with 100's of possible parameters.

:

ℓ_1 as Relaxation of ℓ_0

Replace the ℓ_0-"norm" by the ℓ_1 -norm!

$$\min \sum \varepsilon^2(t, \theta) + \lambda \|\theta\|_0$$

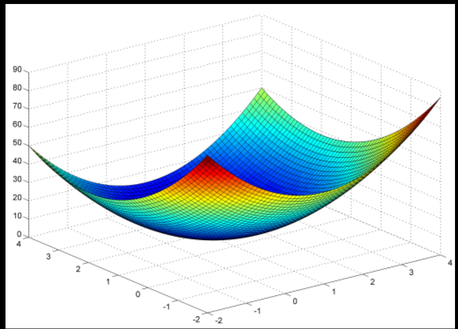
$$\min \sum \varepsilon^2(t, \theta) + \lambda \|\theta\|_1.$$

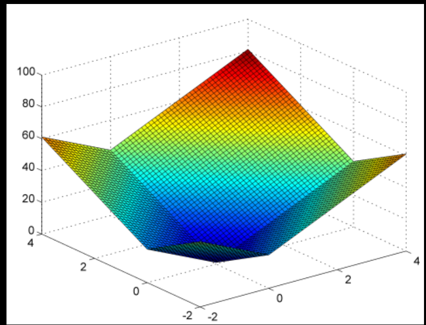
Will this still favor sparse solutions with small $\|\theta\|_0$?

("Sparse" $\approx$ "parsimonious")

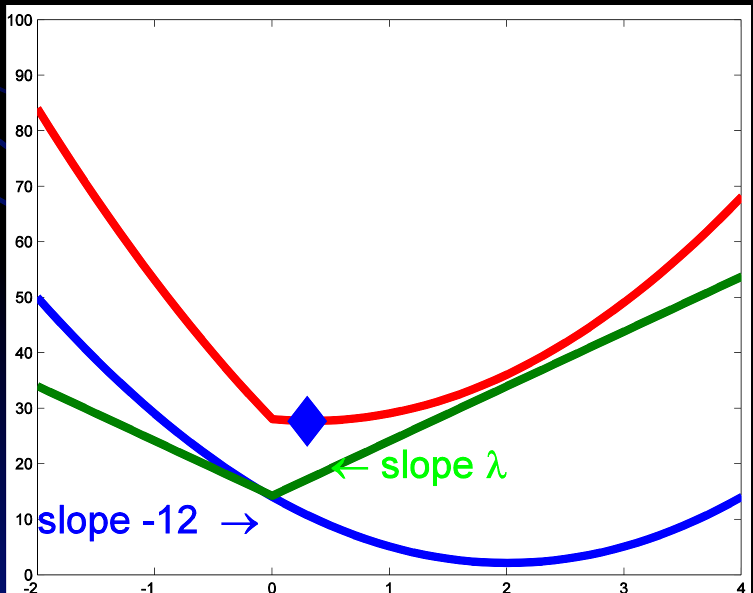
Check Linear Regression

$$\hat{\theta}(\lambda) = \arg\min \|\mathbf{Y} - \mathbf{\Phi}\theta\|^{\mathbf{2}} + \lambda\|\theta\|_{\mathbf{1}}$$





Intersection with plane through the origin:



Blue Curve: Quadratic fit

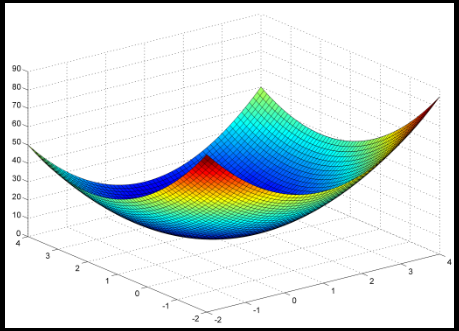
Green Curve: Regularization

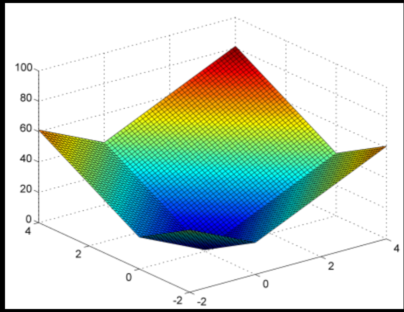
Red curve: Criterion

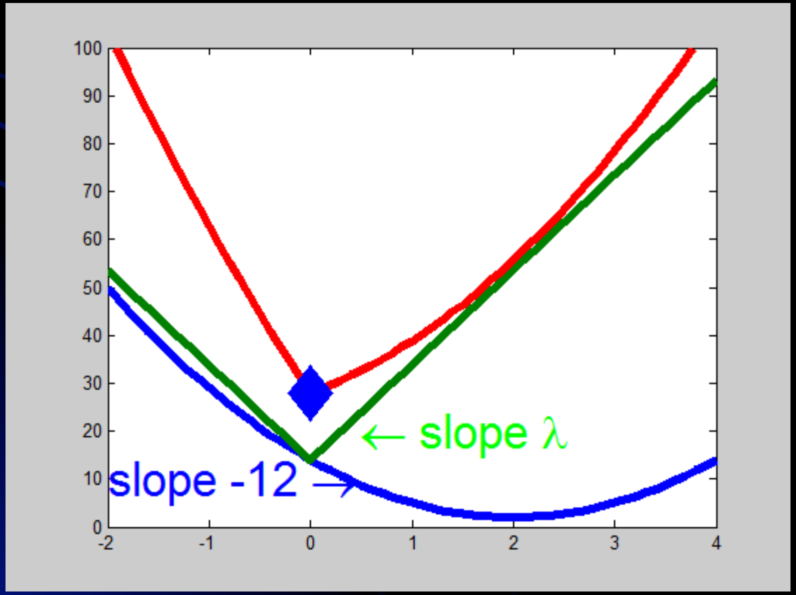
Blue diamond: Minimum

Check Linear Regression

$$\hat{\theta}(\lambda) = \arg\min \|\mathbf{Y} - \mathbf{\Phi}\theta\|^{2} + \lambda\|\theta\|_{1}$$







All zeros for large λ and one by one of the components become non-zero as λ decreases.

$\hat{\theta}(\lambda)$ piecewise linear function of λ

So, the Relaxed Criterion

$$\min \sum \varepsilon^2(t, \theta) + \lambda \|\theta\|_1$$

still favors sparse solutions!

Considarable recent theory around this:

Sparsity and compressed sensing (Candès, Donoho … ~2006)

Regressor selection in linear regression by LASSO (Tibshirani, 1996):

$$\min \|Y - \Phi\theta\|^2 + \lambda \|\theta\|_1$$

Convex problem. Covers many yet unexploited system identification problems

Lasso-like Applications

- Order selection in dynamic models
- Select polynomials terms in NL models
- Find structure in networked systems
- Piecewise affine hybrid models
- Trajectory generation by sparse grid-points
- State smoothing with rare disturbances
- ….

A Standard State Smoothing Problem

$$\mathbf{x}(\mathbf{t}+1) = \mathbf{A}_\mathbf{t}\mathbf{x}(\mathbf{t}) + \mathbf{B}_\mathbf{t}\mathbf{u}(\mathbf{t}) + \mathbf{G}_\mathbf{t}\mathbf{v}(\mathbf{t})$$

$$\mathbf{y}(\mathbf{t}) = \mathbf{C}_\mathbf{t}\mathbf{x}(\mathbf{t}) + \mathbf{e}(\mathbf{t})$$

e is white measurement noise and v is process disturbance.

v is often modelled as white Gaussian noise but in many applications it is mostly zero and strikes only occasionally:

- Control: Load disturbances
- Tracking: Sudden maneuvers
- FDI: Additive system faults
- Parameter estimation: Model segmentation

The Estimation Problem

- Find the jump times $\mathbf{t},\ \mathbf{v}(\mathbf{t}) \neq \mathbf{0}$ and the smoothed state estimates $\hat{\mathbf{x}}_{\mathbf{s}}(\mathbf{t}|\mathbf{N})$
- [Approaches:
  - Willsky-Jones GLR: treat $\mathbf{t}^*,\ \mathbf{v}(\mathbf{t}^*)$ as unknown parameters
  - Treat v as WGN and use Kalman Smoothing
  - IMM: Branch the KF at each time (jump/no jump). Merge/ prune trajectories
  - Treat it as a non-linear smoothing (non-Gaussian noise) by particle techniques]

Treat it as a Sparsity Problem

$$\mathbf{x(t+1) = A_t x(t) + B_t u(t) + G_t v(t)}$$

$$\mathbf{y(t) = C_t x(t) + e(t)}$$

See x as a function of v and optimize the fit with many v(t)=0 by solving

$$\mathbf{min_{v(\cdot)}} \sum \|\mathbf{y(t) - C_t x(t)}\|^2 + \lambda \sum \|\mathbf{v(t)}\|_2$$

(StateSON). Note that

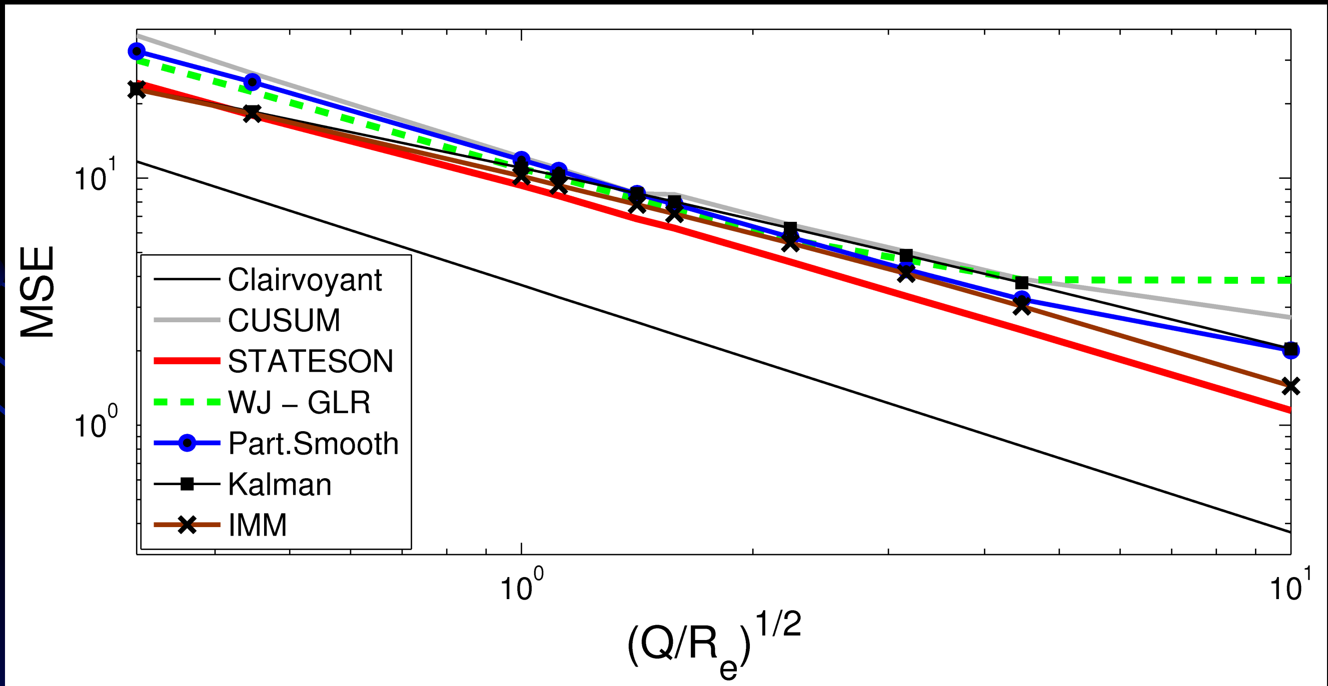
$$\sum \|\mathbf{v(t)}\|_2 = \|\mathbf{V}\|_1, \quad \mathbf{V} = [\|\mathbf{v(1)}\|_2, \ldots, \|\mathbf{v(N)}\|_2]$$

So this is ℓ_1 (sum-of-norm) regularization

Load Disturbances:

DC motor with step load disturbances with probability 0.015. Consider 100 time steps. Varying SNR: Q= jump size, R = noise variance

For each SNR, the RMSE average over time and over 500 MC runs is shown, Many different approaches



StateSON outperforms the established methods!

Segmentation of Systems

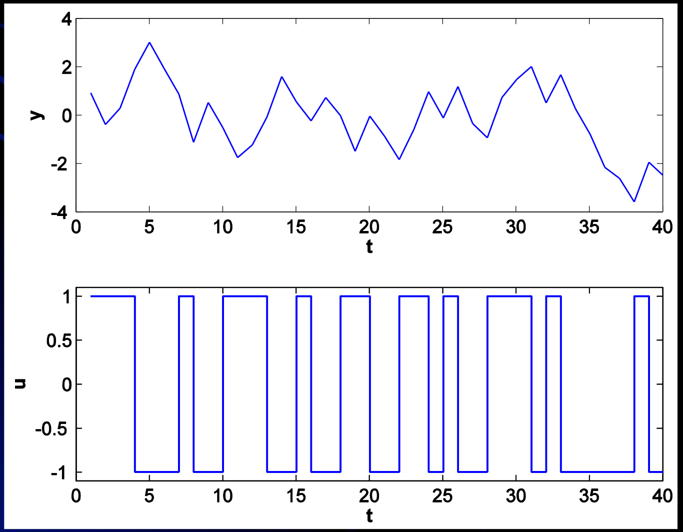
System $\mathbf{y(t) + ay(t-1) = u(t-k) + e(t)}$

k changes from 2 to 1 at time 20

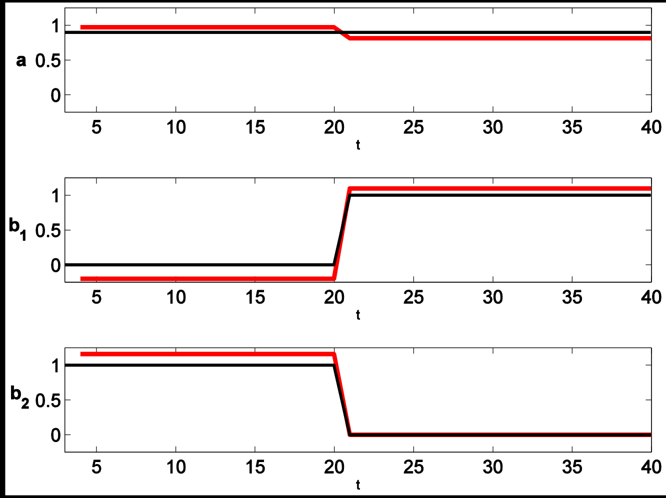
$\mathbf{Model: y(t) + ay(t-1) = b_1 u(t-1) + b_2 u(t-2) + e(t)}$

$\mathbf{written\ as:}\ \theta(\mathbf{t}+1) = \theta(\mathbf{t}) + \mathbf{v(t)}, \quad \mathbf{y(t)} = \varphi(\mathbf{t})\theta(\mathbf{t}) + \mathbf{e(t)}$

Data



Estimate



Red: Estimate

Black: true

Summary: ℓ_1 and Sum-Of-Norms Regularization

- Regularization for sparsity and parsimony

1. ℓ_1 and SoN good proxies for parameter count
2. Valuable tool for structure selection in models
3. Handles rare disturbances/changes
4. Active area of new development: Ideas for nonlinear, hybrid, and LPV model estimation

The World Around System Identification

Statistical Learning theory

Manifold learning

Sparsity

Statistics

Machine Learning

Networked systems

Compressed sensing

Particle filters

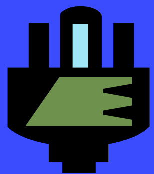
Take Home Messages
Conclusions

Seek Parsimonious Models

ℓ_1 : - regularization is a prime tool for sparsity

The bias/variance trade-off is at the heart of estimation

ℓ_2 : - regularization [well tuned] offers new techniques for robust smaller MSE



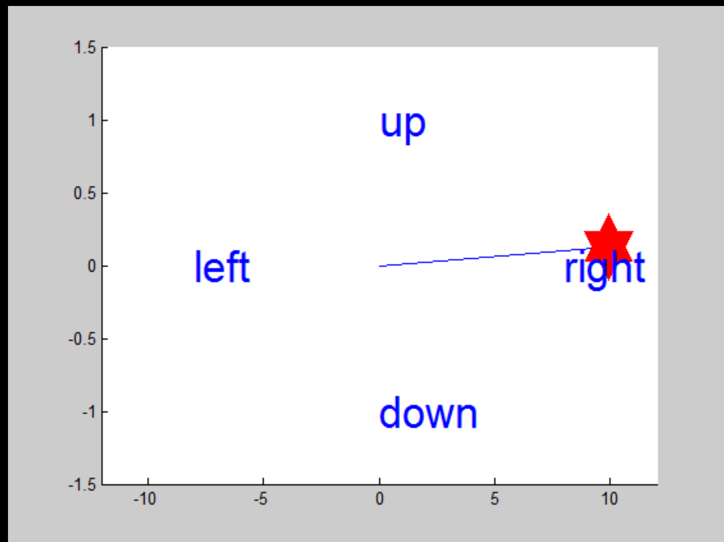
Mature area with traces to old history … but still open for new encounters

Keep vital contacts with other cultures in the world around System Identification

test

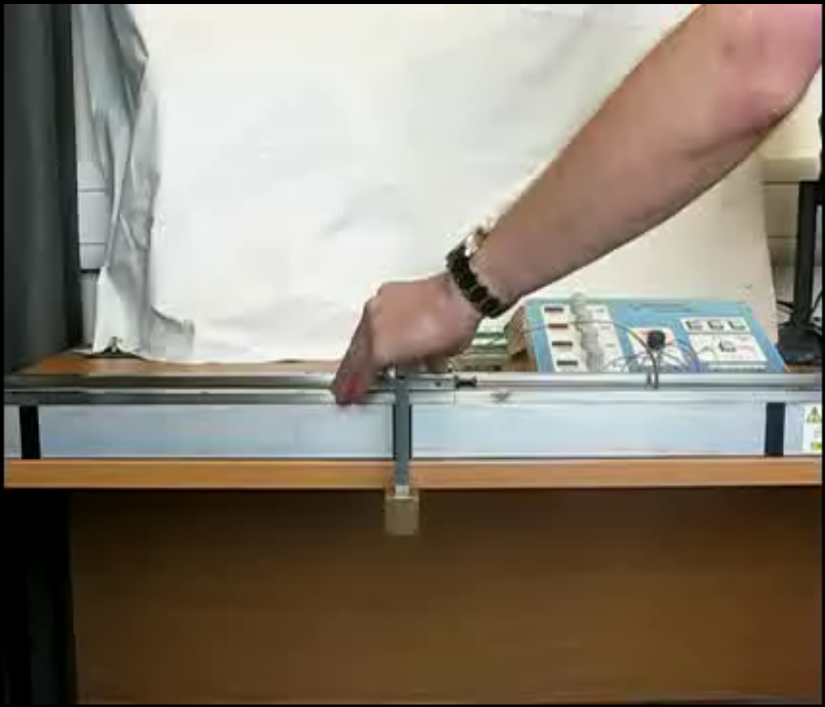
test2

t

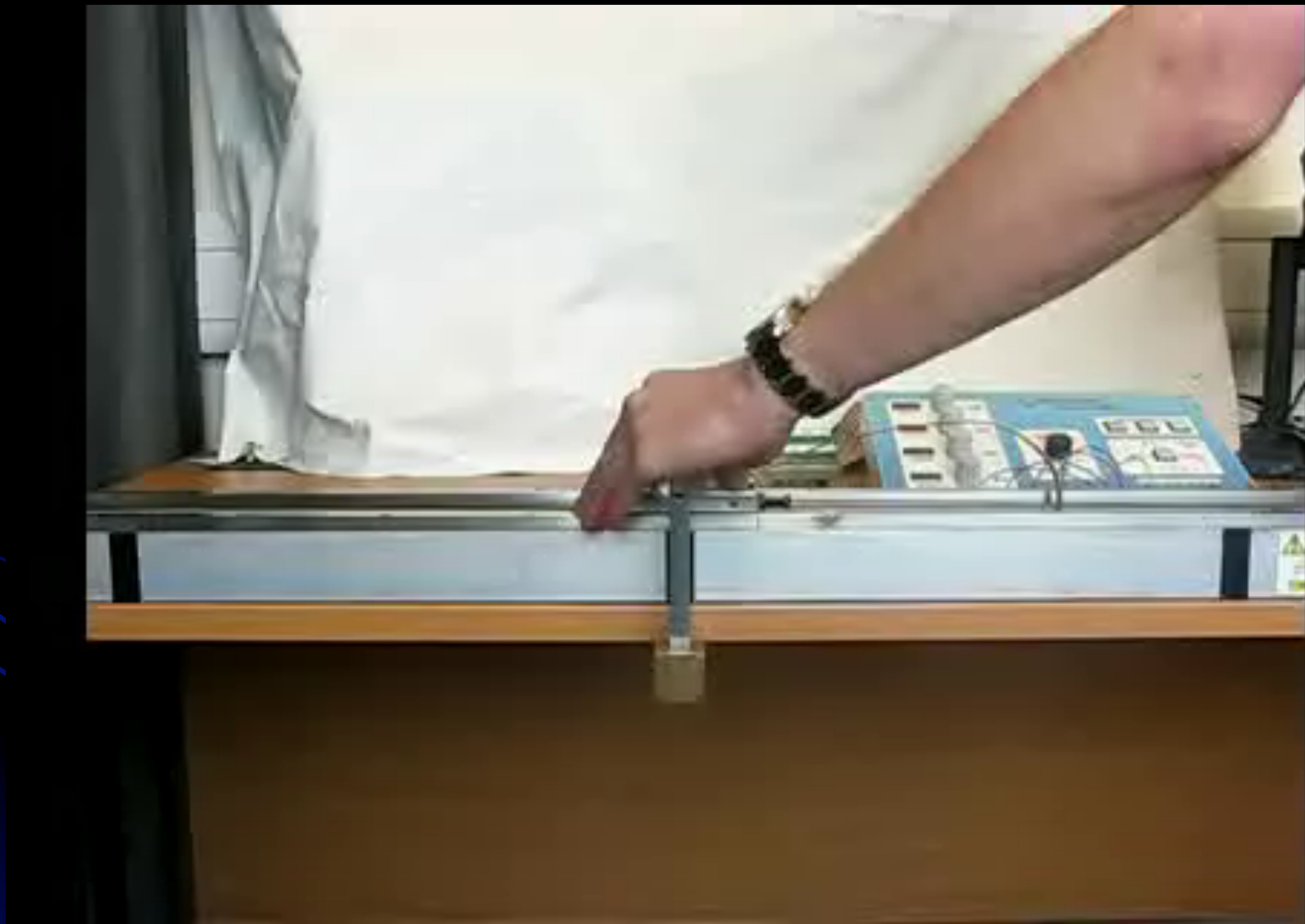


fas

- Text3
- Text1
- Text2



hej



nu

From IR to Model Estimation

The result of the impulse response estimate is a (high order) Finite Impulse Response model (FIR). This can be converted to state space models of any order by model reduction:

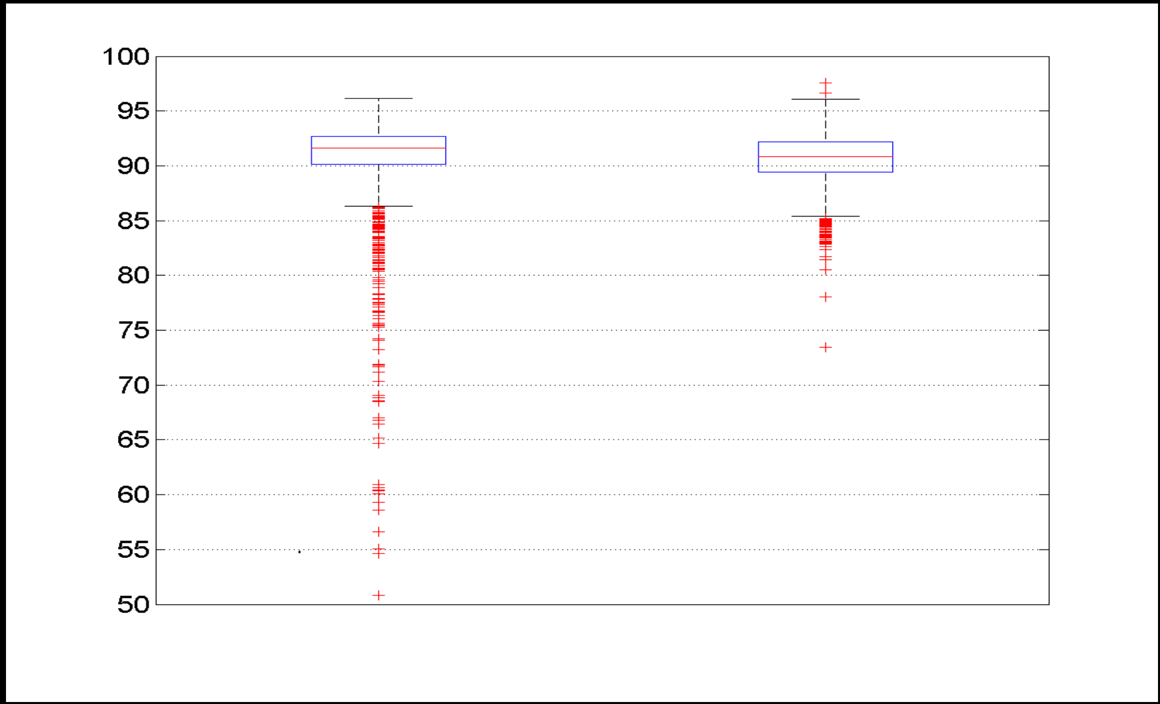
mf = Rfir(data)

m = balred(mf,10)

”Rb-method”

Alt. to ML-method

Box-plots over fits for 2500 different (high order) systems

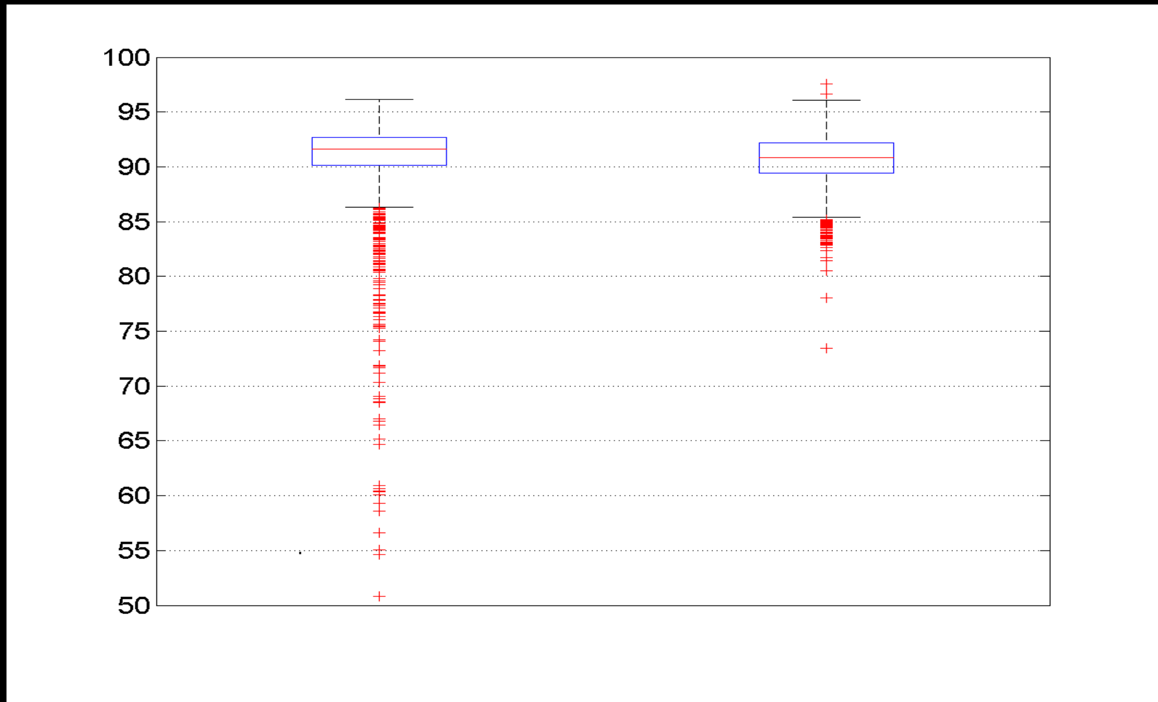


ML 10th order model Rfir+Balred

From IR to Model Estimation

In certain cases Rfir+Balred could be a viable approach to model estimation

Box-plots over fits for 2500 different (high order) systems



ML 10th order model Rfir+Balred